HIROLS
HIGH RESOLUTION DYNAMICS LIMB SOUNDER

Originator: Thomas J. Schapp, J.-P. Riley     Date: 2003-06-12

Subject / Title: SAIL Test Report

Contents / Description / Summary:

Keywords: SAIL, IPU

Purpose (20 characters maximum): Report on Test of SAIL

Science Algorithm Implementation Language

Reference:

Approach:

/s/ J. Drake, Flight Software Manager
/s/ J.-P. Riley, Software Engineer

/s/ Karen Kao, ATC SQE
/s/ N. Koepp-Baker, C.S.E

Advanced Technology Center, CAGE code 65113
Lockheed Martin Missiles & Space
3250 Hanover St.
Palo Alto, CA 94304-1191
United States of America

EOS

WARNING

ESD sensitive components are contained in the assemblies called out in this procedure. All work must be done at an approved ESD station with wrist straps worn at all times. At the beginning of each day the wrist strap(s) used during this test shall be tested and logged or have a current verification sticker.
<table>
<thead>
<tr>
<th>Rev. Date</th>
<th>Section</th>
<th>Change Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002-06-18</td>
<td>All</td>
<td>Added results of Software Acceptance Regression Test 1, Including Appendix of data files for the event, update to table 1-1 adding this test event.</td>
</tr>
<tr>
<td>2002-07-12</td>
<td>VCRM</td>
<td>Updated VCRM status column based on review held today and subsequent update of HIRDLS Comprehensive VCRM by SQE regarding requirements verification.</td>
</tr>
<tr>
<td>2002-07-22</td>
<td>Appendix</td>
<td>Added the text of all SAIL tasks used in Software Acceptance Regression Test 1. Added a Section 6 which contains all inspections. Updated the VCRM to trace to inspection reports.</td>
</tr>
<tr>
<td>2002-07-25</td>
<td>TOC, VCRM, List of Acronyms</td>
<td>Table of Contents, page viii, a “List of Tables” was added. In the VCRM (pages 48 - 54), the following requirements have been modified: SAIL290, 465, 524, and 595. Note that SAIL290 was accepted with a waiver, hence that entry reads “SQE Accepted with Waiver W097”. In the Acronym List (page 55), the acronym IPU was repeated, the second occurrence was removed. In the Acronym List (page 55) the following acronyms have been added: SQE (Software Quality Engineer), VCRM (Verification Cross Reference Matrix).</td>
</tr>
<tr>
<td>2002-08-13</td>
<td>VCRM</td>
<td>Added requirements trace to all test case and task headers. Reworked VCRM to trace only to test case.</td>
</tr>
<tr>
<td>2003-06-12</td>
<td>All</td>
<td>Added results of SAIL Acceptance Regression Test 2.</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

1.1 Scope

1.2 Test Report Format Description

1.3 Verification Events

2 APPLICABLE DOCUMENTS

3 REQUIRED EQUIPMENT

3.1 NASA Test Equipment

3.2 Test Layout

4 SECTION ONE: GROUND TESTING AND PROCEDURES

4.1 Software Acceptance Regression Test 1 (2002-06-17)

4.1.1 Compiling Tasks

4.1.1.1 Test Results

4.1.2 SCERR Test Case (SAIL170, SAIL190, SAIL330)

4.1.2.1 Test Results

4.1.3 SNSTERR Test Case

4.1.3.1 Test Results

4.1.4 SASM Test Case (SAIL540)

4.1.4.1 Test Results

4.1.5 SCMP Test Case (SAIL130)

4.1.5.1 Test Results

4.1.6 SFERR Test Case

4.1.6.1 Test Results

4.2 SAIL Acceptance Regression Test 2 (2003-06-05)

4.2.1 SCERR Test Case

4.2.1.1 Test Results

4.2.2 SNSTERR Test Case

4.2.2.1 Test Results

4.2.3 SASM020 Test Case

4.2.3.1 Test Results

4.2.4 SCMP Test Case

4.2.4.1 Test Results

4.2.5 SFERR Test Case

4.2.5.1 Test Results

4.2.6 Transferring and Converting the test files

5 SECTION TWO: RUNNING THE SAIL TASKS

5.1 Software Acceptance Test (2002-04-10 – 2002-04-12)

5.1.1 Starting the Hardware

5.1.1.1 Test Results

5.1.2 Starting the IEGSE

5.1.2.1 Test Results

5.1.3 Starting the Software

5.1.3.1 Test Results

5.1.4 Transferring and Converting the test files

5.2 Debugging the Hardware

5.2.1 Test Results

5.3 Debugging the Software

5.3.1 Test Results

5.4 Software Acceptance Test 2 (2003-06-05 – 2003-06-06)

5.4.1 Test Results

5.5 Software Acceptance Test 3 (2003-06-07 – 2003-06-07)

5.5.1 Test Results

5.6 Software Acceptance Test 4 (2003-06-08 – 2003-06-08)

5.6.1 Test Results

5.7 Software Acceptance Test 5 (2003-06-09 – 2003-06-09)

5.7.1 Test Results

5.8 Software Acceptance Test 6 (2003-06-10 – 2003-06-10)

5.8.1 Test Results

5.9 Software Acceptance Test 7 (2003-06-11 – 2003-06-11)

5.9.1 Test Results

5.10 Software Acceptance Test 8 (2003-06-12 – 2003-06-12)

5.10.1 Test Results


5.11.1 Test Results

5.12 Software Acceptance Test 10 (2003-06-14 – 2003-06-14)

5.12.1 Test Results


5.13.1 Test Results
<table>
<thead>
<tr>
<th>Section</th>
<th>Test Case</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.4</td>
<td>Test Results</td>
<td>8</td>
</tr>
<tr>
<td>5.1.5</td>
<td>Test Results</td>
<td>9</td>
</tr>
<tr>
<td>5.1.6.1</td>
<td>Test Results</td>
<td>9</td>
</tr>
<tr>
<td>5.1.6.2</td>
<td>Test Results</td>
<td>10</td>
</tr>
<tr>
<td>5.1.6.3</td>
<td>Test Results</td>
<td>11</td>
</tr>
<tr>
<td>5.1.6.4</td>
<td>Test Results</td>
<td>11</td>
</tr>
<tr>
<td>5.1.6.5</td>
<td>Test Results</td>
<td>12</td>
</tr>
<tr>
<td>5.1.6.6</td>
<td>Test Results</td>
<td>12</td>
</tr>
<tr>
<td>5.1.6.7</td>
<td>Test Results</td>
<td>13</td>
</tr>
<tr>
<td>5.1.6.8</td>
<td>Test Results</td>
<td>13</td>
</tr>
<tr>
<td>5.1.6.9</td>
<td>Test Results</td>
<td>14</td>
</tr>
<tr>
<td>5.1.6.10</td>
<td>Test Results</td>
<td>15</td>
</tr>
<tr>
<td>5.1.6.11</td>
<td>Test Results</td>
<td>15</td>
</tr>
<tr>
<td>5.1.6.12</td>
<td>Test Results</td>
<td>16</td>
</tr>
<tr>
<td>5.1.6.13</td>
<td>Test Results</td>
<td>16</td>
</tr>
<tr>
<td>5.1.6.14</td>
<td>Test Results</td>
<td>17</td>
</tr>
<tr>
<td>5.1.6.15</td>
<td>Test Results</td>
<td>18</td>
</tr>
<tr>
<td>5.1.6.16</td>
<td>Test Results</td>
<td>18</td>
</tr>
<tr>
<td>5.1.6.17</td>
<td>Test Results</td>
<td>19</td>
</tr>
<tr>
<td>5.1.6.18</td>
<td>Test Results</td>
<td>19</td>
</tr>
<tr>
<td>5.1.6.19</td>
<td>Test Results</td>
<td>19</td>
</tr>
<tr>
<td>5.1.6.20</td>
<td>Test Results</td>
<td>20</td>
</tr>
<tr>
<td>5.1.6.21</td>
<td>Test Results</td>
<td>21</td>
</tr>
<tr>
<td>5.1.6.22</td>
<td>Test Results</td>
<td>21</td>
</tr>
<tr>
<td>5.1.6.23</td>
<td>Test Results</td>
<td>22</td>
</tr>
<tr>
<td>5.1.6.24</td>
<td>Test Results</td>
<td>22</td>
</tr>
<tr>
<td>5.1.6.25</td>
<td>Test Results</td>
<td>22</td>
</tr>
<tr>
<td>5.1.6.26</td>
<td>Test Results</td>
<td>23</td>
</tr>
<tr>
<td>5.1.6.27</td>
<td>Test Results</td>
<td>23</td>
</tr>
<tr>
<td>5.1.6.28</td>
<td>Test Results</td>
<td>24</td>
</tr>
</tbody>
</table>
Software Acceptance Regression Test 1 (2002-06-17)

5.2 Starting the Hardware
5.2.1 Starting the Hardware
5.2.2 Starting the IEGSE
5.2.3 Aliveness Test
5.2.4 Record Software Version Numbers
5.2.5 Start and Setup SAIL
5.2.6 Test Cases/Tasks

5.2.6.1 SARR Test Case (SAIL550, SAIL250)
5.2.6.1.1 SAIL Task SARR010 (SAIL250)
5.2.6.1.2 SAIL Task SARR011 (SAIL250)
5.2.6.1.3 SAIL Task SARR012 (SAIL250)
5.2.6.1.4 SAIL Task SARR013 (SAIL250)
5.2.6.1.5 SAIL Task SARR014 (SAIL250)
5.2.6.1.6 SAIL Task SARR020 (SAIL550)
5.2.6.1.7 SAIL Task SARR030

5.2.6.2 SCMP Test Case (SAIL280, SAIL290, SAIL300)
5.2.6.2.1 SAIL Task SCMP020 (Flight Portion) (SAIL280, SAIL290, SAIL300)

5.2.6.3 SCOMS Test Case (SAIL010, SAIL020, IPS1130)
5.2.6.3.1 SAIL Task SCOMS010 (SAIL010)
5.2.6.3.2 SAIL Task SCOMS020 (SAIL020, IPS1130)

5.2.6.4 SEXEC Test Case (SAIL480, SAIL490, SAIL580)
5.2.6.4.1 SAIL Tasks SEXEC000, - SEXEC016 (SAIL480, SAIL490, SAIL580)
5.2.6.4.2 SAIL Tasks SEXEC020, - SEXEC035

5.2.6.5 SFERR Test Case (SAIL610)
5.2.6.5.1 SAIL Task SFERR010
5.2.6.5.2 SAIL Task SFERR020
5.2.6.5.3 SAIL Task SFERR030
5.2.6.5.4 SAIL Task SFERR040 (SAIL610)
5.2.6.5.5 SAIL Task SFERR050

5.2.6.6 SFUN Test Case (SAIL310, SAIL320, SAIL400)
5.2.6.6.1 SAIL Task SFUN010 (SAIL310, SAIL320)
5.2.6.6.2 SAIL Task SRTL010 (SAIL400)

5.2.6.7 SIERR010 Test Case (SAIL570)
5.2.6.7.1 SAIL Task SIERR010 (SAIL570)

5.2.6.8 SMEM Test Case (SAIL520, SAIL590, SAIL640)
5.2.6.8.1 SAIL Task SMEM010 (SAIL520, SAIL590, SAIL640)

5.2.6.9 SNST Test Case
5.2.6.9.1 SAIL Task SNST010

5.2.6.10 SOP Test Case (SAIL350, SAIL360, SAIL370, SAIL380, SAIL390, SAIL410, SAIL420, SAIL430)
5.2.6.10.1 SAIL Task SOP010 (SAIL350, SAIL360, SAIL370, SAIL380, SAIL390, SAIL410, SAIL420, SAIL430)
5.2.6.10.2 SAIL Task SOP020

5.2.6.11 SOVR Test Case (SAIL450, SAIL460)
5.2.6.11.1 SAIL Task SOVR010 (SAIL450, SAIL460)

5.2.6.12 SST Test Case (SAIL050, SAIL070, SAIL080, SAIL090, SAIL100, SAIL160, SAIL260, SAIL340, SAIL470)
5.2.6.12.1 SAIL Task SST010 (SAIL050, SAIL090, SAIL100, SAIL160, SAIL260, SAIL340, SAIL470)
5.2.6.12.2 SAIL Task SDATA010 (SAIL070, SAIL080)
5.2.6.13 SAIL Test Case (SAIL200, SAIL210, SAIL220, SAIL230, SAIL240, SAIL440)

5.2.6.13.1 SAIL Task STYPE010 (SAIL200, SAIL210, SAIL220, SAIL230, SAIL240)

5.2.6.14 Test Case WAIT (SAIL600, SAIL640)

5.2.6.14.1 SAIL Tasks WAIT010 – WAIT020 (SAIL600, SAIL640)

5.2.6.15 STOVR Test Case (SAIL510)

5.2.6.15.1 SAIL Task STOVR010 (SAIL510)

5.2.6.16 Test Case SASM020 (SAIL540)

5.2.6.16.1 SAIL Task SASMsizeLimit (SAIL540)

5.2.6.16.2 SAIL Task SASMoverSizeLimit (SAIL540)

5.3 SAIL Acceptance Regression Test 2 (2003-06-05)

5.3.1 Test Results

5.3.1.1 SARR Test Case

5.3.1.1.1 SAIL Task SARR010

5.3.1.1.2 SAIL Task SARR011

5.3.1.1.3 SAIL Task SARR012

5.3.1.1.4 SAIL Task SARR013

5.3.1.1.5 SAIL Task SARR014

5.3.1.1.6 SAIL Task SARR020

5.3.1.1.7 SAIL Task SARR030

5.3.1.2 SCMP Test Case

5.3.1.2.1 SAIL Task SCMP020 (Flight Portion)

5.3.1.3 SCOMS Test Case

5.3.1.3.1 SAIL Task SCOMS010

5.3.1.3.2 SAIL Task SCOMS020

5.3.1.4 SEXEC Test Case

5.3.1.4.1 SAIL Tasks SEXEC000 - SEXEC016

5.3.1.5 SFERR Test Case

5.3.1.5.1 SAIL Task SFERR010

5.3.1.5.2 SAIL Task SFERR020

5.3.1.5.3 SAIL Task SFERR030

5.3.1.5.4 SAIL Task SFERR040

5.3.1.5.5 SAIL Task SFERR050

5.3.1.6 SFUN Test Case

5.3.1.6.1 SAIL Task SFUN010

5.3.1.6.2 SAIL Task SRTL010

5.3.1.6.3 SAIL Task SRTL020

5.3.1.7 SIERR010 Test Case

5.3.1.7.1 SAIL Task SIERR010

5.3.1.8 SMEM Test Case

5.3.1.8.1 SAIL Task SMEM010

5.3.1.9 SNST Test Case

5.3.1.9.1 SAIL Task SNST010

5.3.1.10 SOP Test Case

5.3.1.10.1 SAIL Task SOP010

5.3.1.10.2 SAIL Task SOP020

5.3.1.11 SOVR Test Case

5.3.1.11.1 SAIL Task SOVR010

5.3.1.12 SST Test Case

5.3.1.12.1 SAIL Task SST010

5.3.1.12.2 SAIL Task SDATA010
5.3.2.17 SPEC Test Case 65
5.3.2.17.1 SAIL Task sw_loc_117_3_1_1_01 65
5.3.2.17.2 SAIL Task sw_loc_117_3_1_1_02 66
5.3.2.17.3 SAIL Task sw_loc_117_3_1_2_01 66
5.3.2.17.4 SAIL Task sw_loc_117_3_1_2_02 67
5.3.2.17.5 SAIL Task sw_loc_117_3_2_1_01 67
5.3.2.17.6 SAIL Task sw_loc_117_3_3_1_2_1_01 67
5.3.2.17.7 SAIL Task sw_loc_117_3_3_1_2_1_02 68
5.3.2.17.8 SAIL Task sw_loc_117_3_3_1_2_2_01 68
5.3.2.17.9 SAIL Task sw_loc_117_3_3_1_2_3_01 69
5.3.2.17.10 SAIL Task sw_loc_117_3_3_1_2_4_01 69
5.3.2.17.11 SAIL Task sw_loc_117_3_3_1_2_5_01 69
5.3.2.17.12 SAIL Task sw_loc_117_3_3_1_3_01 70
5.3.2.17.13 SAIL Task sw_loc_117_3_3_1_4_01 70
5.3.2.17.14 SAIL Task sw_loc_117_3_3_2_1_01 71
5.3.2.17.15 SAIL Task sw_loc_117_3_3_2_2_01 71
5.3.2.17.16 SAIL Task sw_loc_117_3_4_1_1_01 71
5.3.2.17.17 SAIL Task sw_loc_117_3_4_1_2_01 72
5.3.2.17.18 SAIL Task sw_loc_117_3_5_2_01 72
5.3.2.17.19 SAIL Task sw_loc_117_6_01 72
5.3.2.17.20 SAIL Task sw_loc_117_6_02 73
5.3.2.17.21 SAIL Task sw_loc_117_6_03 73

6 INSPECTIONS AND ANALYSES

6.1 Code Inspection Session 1 (2002-06-22)
6.1.1 IR_SAI_1000 Dynamic Memory Inspection (SAIL180) 76
6.1.2 IR_SAI_1010 Pointer Access Prohibited Inspection (SAIL270) 76
6.1.3 IR_SAI_1020 Non-Optimizing Compiler Inspection (SAIL270) 76
6.1.4 IR_SAI_1030 Memory Management Inspection (SAIL560, SAIL570) 76
6.1.5 IR_SAI_1040 Sail Task Starting (SAIL600) 76
6.1.6 IR_SAI_1050 Sail tasks and data (SAIL640) 76

7 VERIFICATION CROSS-REFERENCE MATRIX

8 LIST OF ACRONYMS

9 APPENDIX
9.1.4.2.5 SAIL TASK SCERR050.sai 140
9.1.4.2.6 SAIL TASK SCERR060.sai 141
9.1.4.3 SCMP Test Case 142
  9.1.4.3.1 SAIL TASK SCMP010.sai 142
  9.1.4.3.2 SAIL TASK SCMP020.sai 143
9.1.4.4 SCOMS Test Case 146
  9.1.4.4.1 SAIL TASK SCOMS010.sai 146
  9.1.4.4.2 SAIL TASK SCOMS020.sai 149
9.1.4.5 SST Test Case 149
  9.1.4.5.1 SAIL TASK SDATA010.sai 149
  9.1.4.5.2 SAIL TASK SST010.sai 158
9.1.4.6 SEXEC Test Case 163
  9.1.4.6.1 SAIL TASK SEXEC000.sai 163
  9.1.4.6.2 SAIL TASK SEXEC001.sai 164
  9.1.4.6.3 SAIL TASK SEXEC002.sai 164
  9.1.4.6.4 SAIL TASK SEXEC003.sai 165
  9.1.4.6.5 SAIL TASK SEXEC004.sai 166
  9.1.4.6.6 SAIL TASK SEXEC005.sai 167
  9.1.4.6.7 SAIL TASK SEXEC006.sai 167
  9.1.4.6.8 SAIL TASK SEXEC007.sai 168
  9.1.4.6.9 SAIL TASK SEXEC008.sai 169
  9.1.4.6.10 SAIL TASK SEXEC009.sai 170
  9.1.4.6.11 SAIL TASK SEXEC010.sai 171
  9.1.4.6.12 SAIL TASK SEXEC011.sai 171
  9.1.4.6.13 SAIL TASK SEXEC012.sai 172
  9.1.4.6.14 SAIL TASK SEXEC013.sai 173
  9.1.4.6.15 SAIL TASK SEXEC014.sai 174
  9.1.4.6.16 SAIL TASK SEXEC015.sai 174
  9.1.4.6.17 SAIL TASK SEXEC016.sai 175
  9.1.4.6.18 SAIL TASK SEXEC020.sai 176
9.1.4.7 SFERR Test Case 177
  9.1.4.7.1 SAIL TASK SFERR010.sai 177
  9.1.4.7.2 SAIL TASK SFERR020.sai 178
  9.1.4.7.3 SAIL TASK SFERR030.sai 180
  9.1.4.7.4 SAIL TASK SFERR040.sai 182
  9.1.4.7.5 SAIL TASK SFERR050.sai 183
9.1.4.8 SFUN Test Case 185
  9.1.4.8.1 SAIL TASK SFUN010.sai 185
  9.1.4.8.2 SAIL TASK SRTL010.sai 189
9.1.4.9 SIERR Test Case 192
  9.1.4.9.1 SAIL TASK SIERR010.sai 192
9.1.4.10 SMEM Test Case 194
  9.1.4.10.1 SAIL TASK SMEM010.sai 194
9.1.4.11 SNST Test Case 196
  9.1.4.11.1 SAIL TASK SNST010.sai 196
  9.1.4.11.2 SAIL TASK SNSTERR.sai 204
9.1.4.12 SOP Test Case 211
  9.1.4.12.1 SAIL TASK SOP010.sai 211
  9.1.4.12.2 SAIL TASK SOP020.sai 221
9.1.4.13 SOVR Test Case 225
  9.1.4.13.1 SAIL TASK SOVR010.sai 225
9.1.4.14 STOVR Test Case 227
  9.1.4.14.1 SAIL TASK STOVR010.sai 227
9.1.4.15 STYPE Test Case 228
  9.1.4.15.1 SAIL TASK STYPE010.sai 228
9.1.4.16 WAIT Test Case 230
  9.1.4.16.1 SAIL TASK WAIT010.sai 230
  9.1.4.16.2 SAIL TASK WAIT020.sai 231
9.2 SAIL Regression Test 2 (2003-06-10) 232
List of Figures

Figure 1  Test Configuration ................................ ................................ ................................ ................. 3

List of Tables

TABLE 1-1 VERIFICATION EVENTS 1
1 Introduction

1.1 Scope
The purpose of this test is to provide confidence in the SAIL implementation including the Compiler, Assembler, Executive and Interpreter. This test will also provide confidence in SAIL's interface to the IPU application code.

As the SAIL test code is first sent through the compiler some outputs will need to be verified. Then after each of the test programs is assembled, converted, uploaded and run, the telemetry will be analysed using the displays and the data_read program.

1.2 Test Report Format Description
Each test case report will begin by listing the test event.

The Software Acceptance Test occurred 2002-04-10 through 2002-04-12

The verification is provided by the test procedure document through test conductor initials. The verification document for this Software Acceptance Test event is TP-LOC-803, dated 2002-04-09.

Each test case will state pass/fail, following the list of verification items.

For each step, Success/Failure/Partial Failure will indicate the result of the step.

Following the verification list, will be a list of any redlines formatted as noted below.

1.3 Verification Events
All Verification Events which provide requirements verification or re-verification due to regression testing will be listed here. Each test/analysis/inspection case and appendix data file will have a sub-section noting the associated test event.

<table>
<thead>
<tr>
<th>Test Event Associated Test Plan</th>
<th>Test Event</th>
<th>Associated Test Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Acceptance Test TP-LOC-854, Initial Release 2002-04-10 – 2002-04-12</td>
<td>Initial</td>
<td>6.0.0 2.3</td>
</tr>
<tr>
<td>Software Acceptance Regression Test 1 TP-LOC-854A 2002-06-17 6.1.0 2.4</td>
<td>Initial</td>
<td>6.1.0 2.4</td>
</tr>
<tr>
<td>Code Inspection Session 1 TP-LOC-854B 2002-06-22 6.1.2 2.4</td>
<td>Initial</td>
<td>6.1.2 2.4</td>
</tr>
<tr>
<td>SAIL Acceptance Regression Test 2 TP-LOC-854C 2003-06-05 6.3.0 2.5</td>
<td>Initial</td>
<td>6.3.0 2.5</td>
</tr>
</tbody>
</table>

Table 1-1 Verification Events

2 Applicable Documents

<table>
<thead>
<tr>
<th>Doc. Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW-LOC-117</td>
<td>SAIL High-Level Specification</td>
</tr>
<tr>
<td>SW-LOC-127</td>
<td>SAIL Low-Level Specification</td>
</tr>
<tr>
<td>SW-HIR-147</td>
<td>SAIL Requirements Document</td>
</tr>
<tr>
<td>SW-LOC-854</td>
<td>SAIL Software Test Plan</td>
</tr>
<tr>
<td>TP-LOC-822C</td>
<td>SAIL Test Procedure</td>
</tr>
</tbody>
</table>
3 Required Equipment

3.1 HIRDLS Test Equipment

<table>
<thead>
<tr>
<th>Description</th>
<th>Mfr./Model #</th>
<th>LMMS Tag #</th>
<th>Cal Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEUs connected to IPU</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSA card running IPU</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC, with SAIL compiler, assembler, and network interface</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power supplies</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wrist straps</td>
<td>IEGSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timing board</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2 Test Layout

This test has two independent sections. First, the test code gets compiled and assembled on a PC. Some error outputs are
checked and one of the assembler files is changed and re-assembled. The second section requires IPU interfaced to the
IEGSE. One of the test cases also requires connection to the TEU.
WARNING

ESD sensitive components are contained in the assemblies called out in this procedure. All work done to ensure an ESD safe environment shall be done by an approved ESD technician wearing ESD wrist straps during the test and with ESD sensitive devices in a static controlled environment.

Figure 1  Test Configuration

Ground Tools Running Under Windows

Assembler  Compiler  

Ground Tools Running Under UNIX

Loader  I/O  Displays  Read

SAIL Control Core

IEGSE  Displays  Data Read

PC

Assembler  Compiler  Loader  I/O  Displays

Hardware

WARNING

ESD sensitive components are contained in the assemblies called out in this procedure. All work done to ensure an ESD safe environment shall be done by an approved ESD technician wearing ESD wrist straps during the test and with ESD sensitive devices in a static controlled environment.
SECTION ONE: GROUND TESTING AND PROCEDURES

4.1 Software Acceptance Regression Test 1 (2002-06-17)

4.1.1 Compiling Tasks

On the PC start a DOS command interpreter, enter the test code directory:
F:\SAIL\SAIL_CSCI\[SAIL_VERSION]\SAILtasks\SAIL_SWACC, replacing [SAIL_VERSION] with the version number of SAIL that is under test. Run the test code through the compiler and assembler by typing
"MAKE_SAIL_SWACC > SAIL_SWACC_[DATE].txt", replacing [DATE] with the date of the test event in the format YYYY_MM_DD, in the DOS command interpreter. This file will contain the version number of all SAIL tasks used for the SAIL SWACC, and must be appended to the SAIL test report.

The associated SAIL source file for each of the below error files in sections 4.11 – 4.1.3 contains a list of the expected errors in comments. Verify that only the expected errors are present in each error file. The column and line numbers for the error messages may not agree, and should be ignored. The error type and error messages will be identical.

4.1.1.1 Test Results

Success

Verification Source: SAIL_SWACC_2002_06_17.txt
SAIL_SWACC_2002_06_17.err

Test Conductor: JPR 2002-06-17, Day 168

4.1.2 SCERR Test Case (SAIL170, SAIL190, SAIL330)

List all error files with the command "dir *.err" from the DOS prompt while in the directory F:\SAIL\SAIL_CSCI\[SAIL_VERSION]\SAILtasks\SAIL_SWACC. The following error files should have a non-zero size:
SCERR010.err
SCERR040.err  (SAIL170, SAIL190)
SCERR050.err
SCERR060.err  (SAIL330)

Test Engineer

4.1.2.1 Test Results

Success

Test Conductor: JPR 2002-06-17, Day 168

4.1.3 SNSTERR Test Case

List all error files with the command "dir *.err" from the DOS prompt while in the directory F:\SAIL\SAIL_CSCI\[SAIL_VERSION]\SAILtasks\SAIL_SWACC. The following error file should have a non-zero size:
SNSTERR.err

Test Engineer

4.1.3.1 Test Results

Success

Test Conductor: JPR 2002-06-17, Day 168

4
4.1.3 Test Results

Test Conductor: JPR 2002-06-17, Day 168

4.1.3.1 Test Results

4.1.4 SASM Test Case (SAIL540)

List all error files with the command "dir *.err" from the dos prompt while in the directory "F\SAIL\SAIL_CSCI\[SAIL_VERSION]\SAILtasks\SAIL_SWACC". The following error file should have a non-zero size:

SASMoverSizeLimit.err

(Note: No associated SAIL source file, expected error: "Code size exceeds 64K at line 65539")

Test Conductor: JPR 2002-06-17, Day 168

4.1.4.1 Test Results

4.1.5 SCMP Test Case (SAIL130)

Verify that the comments have been removed in SCMP010.p1 by checking it against SCMP010.sai

Comments removed OK

Test Conductor: JPR 2002-06-17, Day 168

4.1.5.1 Test Results

4.1.6 SFERR Test Case

Edit the file SFERR050.a created in the "F\SAIL\SAIL_CSCI\[SAIL_VERSION]\SAILtasks\SAIL_SWACC" directory. Find the code section:

```plaintext
TESTEQ errortype err_call_during_error _it0
IFFALSE _it0 _Label8@error
JMP _Label9@error
_Label8@error:
_Label9@error:
```

Add the line CALL function between the IFFALSE line and the JMP line that follows. Assemble SFERR050a by typing

```
SC /ASM SFERR050
```

Test Conductor: JPR 2002-06-17, Day 168

4.1.6.1 Test Results

Test Conductor: JPR 2002-06-17, Day 168
4.2 SAIL Acceptance Regression Test 3 (2003-06-05)

4.2.1 SCERR Test Case
List all error files with the command “dir *.err” from the dos prompt while in the directory “\SAIL\SAIL_CSCI\[SAIL_VERSION]\SAILtasks\SAIL_SWACC”. The following error files should have a non-zero size:

- SCERR010.err
- SCERR040.err
- SCERR050.err
- SCERR060.err

Test Engineer

4.2.1.1 Test Results
Success
Test Conductor: JPR 2003-06-10

Procedural Redline:
Line 323 of scerr040.sai needs updated to indicate that this error is now a warning instead of an error.

4.2.2 SNSTERR Test Case
List all error files with the command “dir *.err” from the dos prompt while in the directory “\SAIL\SAIL_CSCI\[SAIL_VERSION]\SAILtasks\SAIL_SWACC”. The following error file should have a non-zero size:

- SNSTERR.err

Test Engineer

4.2.2.1 Test Results
Success
Test Conductor: JPR 2003-06-10

4.2.3 SASM020 Test Case
List all error files with the command “dir *.err” from the dos prompt while in the directory “\SAIL\SAIL_CSCI\[SAIL_VERSION]\SAILtasks\SAIL_SWACC”. The following error file should have a non-zero size:

- SASM020.err (Note: No associated SAIL source file, expected error: “Code size exceeds 64K at line 65539”)

Test Engineer

4.2.3.1 Test Results
Success
Test Conductor: JPR 2003-06-10

4.2.4 SCMP Test Case
Verify that the comments have been removed in SCMP010.p1 by checking it against SCMP010.sai
Comments removed OK

Test Conductor: JPR 2003-06-10

4.2.4.1 Test Results
Success
Test Conductor: JPR 2003-06-10

Comments removed OK
4.2.4 Test Results

4.2.5 SFERR Test Case

Add the line:

```
CALL function
```

between the IFALSE line and the JMP line that follows. Assemble SFERR050 by typing:

```
SC /ASM SFERR050
```

4.2.6 Transferring and Converting the test files

Open an FTP session to nimbus.spasci.com as the user hirdls, from the PC directory f:SAIL:SAIL_CSCI[SAIL_VERSION]:SAILtasks:SAIL_SWACC. Once logged into nimbus, change to the sail/testcode directory from the hirdls home directory on nimbus. FTP all .C and .d files to the sail/testcode directory on nimbus. From the DOS prompt on the PC, in the directory.

f:SAIL:SAIL_CSCI[SAIL_VERSION]:SAILtasks:SAIL_SWACC,
type the command "CLEAN_SAIL_SWACC [DATE]", replacing [DATE] with the date of the test event in the format YYYY_MM_DD. The command will generate a file named SAIL_SWACC_[DATE].err. This file must be placed into the appendix of the test report.

From fraser, open an FTP session to nimbus.spasci.com as the user hirdls. FTP all .C and .d files from sail/testcode to your dev/upload/MAKESTOL directory. Run the script ~tsadmin/rel/scripts/sailCheckOut to check out all sail testcode files. Run the script ~tsadmin/rel/scripts/makeSailSeg. This script will convert all sail files into seg files, and check them into RCS. Now, log in as user tsadmin. Change to the dev/upload/MAKESTOL directory from the tsadmin home directory. Run the script releaseSailSeg. Now, log in as user hirdls. Run the script releaseSailSeg once more. To run the releaseSailSeg script as user hirdls, the IEGSE must be running (see section 5.2 Starting the IEGSE). This script releases all seg files as user tsadmin, and compiles them into stol procedures as user hirdls. There are two additional shell scripts needed:

```
em_sai_exec_tst
em_sai_diag_tst
```
SECTION TWO: RUNNING THE SAIL TASKS

5.1 Software Acceptance Test (2002-04-10 - 2002-04-12)

5.1.1 Starting the Hardware

Power on the instrument - ops order will direct which side to use

Start hir_qba_on  (a side)
Start hir_qbb_on  (b side)

5.1.1.1 Test Results

Success

Test Conductor: JPR 2002/100

Procedural Redline:
@hir_bip_eeread 0
@hir_bip_boot

Reason/Resolution
Current configuration did not allow for autoboot, must boot manually.

5.1.2 Starting the IEGSE

Start a session on the IEGSE computer. In the menu, clean and start the IEGSE by typing

start_clean; start_testdisplay

Start the SAIL display.

5.1.2.1 Test Results

Success

Test Conductor: JPR 2002/100

5.1.3 Aliveness Test

Power on the IPU and obtain data in the IEGSE displays.

5.1.4 Record Software Version Numbers

SAIL version number
IPU Application code version number
IEGSE version number

5.1.4.1 Test Results

Failure

Test Conductor: JPR 2002/100

Failure Redline:
SAIL version number is 0

Reason/Resolution
There is no SAIL version number, SAIL shares the version number of the IPU. Telemetry point is not connected.
5.1.5 Start and Setup SAIL
Send the following set of commands:

```
@hir_tlm_scifmt 1,0,0,0,0,0,0,0,0,1,1,0,1,x#1FFFFFF,1,1
```

**5.1.5.1 Test Results**
Success

Test Conductor: JPR 2002/100

Procedural Redlines:

```
@hir_tlm_diagfmt 1,0,0,0,0
@hir_tlm_diagfmt 11,0,0,0,0
@hir_tlm_diagfmt 14,0,0,0,0
```

**Reason/Resolution**
Current version of IPU CSCI enabled diagnostic items 1, 11, 14 by default. Test procedure was created under the assumption all diagnostic items should be off by default.

5.1.6 Test Cases
There is a stol proc for each of the test cases, most start by loading the code into SAIL memory block #4 and data into SAIL memory block #128. Next SAIL task #0 is created from those memory blocks and then run. The stol proc is halted waiting for input so the operator can verify that the task completed correctly. After verification the operator hits the Enter key to continue the stol proc which will kill the SAIL task and delete the used SAIL memory blocks, making SAIL ready for the next test case. Then hit Ctrl-C to close the stol proc window.

To start the stol proc:

1) Input start
2) Use TASK_STATUS for task 0 of the SAIL display to verify that the task ran, counter incremented and the task suspended. Here are the task status meanings.

- Halted = 0
- Ready = 1
- Running = 2
- Suspended = 3
- Wait = 4

This status nibble can be found in the lower nibble of the TASK_STATUS word. The upper nibble is a counter, which counts up to 0xF and then starts over with zero. This counter will change any time a task waits or ends.

3) Set memory dump to the start of task's data memory. Use data_read to verify that the output array is correctly loaded. The contents of these arrays will vary from one test case to the next. The dump locations must be found in the diagnostic area for SAIL memory in the science packet. Somewhere in that block will be the sequence of numbers.

5.1.6.1 Test Case SARR010
Use the "sarr010" stol proc to load and run sarr010.

Task ran and suspended itself
If it is not already set, set memory dump to the beginning of the task data memory with the following command:
```
@hir_tlm_diagfmt 5,1,258,128,0
```
Run data_read
Verify that in the first SAIL task data memory block, the data array is loaded with 0's.

Set memory dump to the end of the task data memory with the following command:
```
@hir_tlm_diagfmt 5,1,258,255,0
```
Run data_read
Verify that in the last SAIL task data memory block, the data array is loaded with 0x01.

Date 5.1.6.1.1 Test Results
Success
Test Conductor: JPR 2002/100

5.1.6.2 Test Case SARR011
Use the "sarr011" stol proc to load and run sarr011.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command:
```
@hir_tlm_diagfmt 5,1,258,128,0
```
Run data_read
Verify that in the first SAIL task data memory block, the data array is loaded with 0xFFs.

Set memory dump to the end of the task data memory with the following command:
```
@hir_tlm_diagfmt 5,1,258,255,0
```
Run data_read
Verify that in the last SAIL task data memory block, the data array is loaded with 0xFFs.

Date 5.1.6.2.1 Test Results
Success
Test Conductor: JPR 2002/100

Failure Redline:
The last 2 words of sail memory did not contain 0xFF, it contained 0xFFFFFFFF
Reason/Resolution:
The array was sized such that the size of the array plus the size of the local variables was one short of total available sail memory. The ability of an array to use all available memory successfully demonstrated in SARR010, no requirements impact as an array of logical values is functionally equivalent in memory usage to an array of ints.

5.1.6.3 Test Case SARR012
Use the “sarr012” stol proc to load and run sarr012.  
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command
When dump_diagn 5,1,258,128,0
Run data_read
Verify that in the first SAIL task data memory block the data array is loaded with 0xFFFFs

Set memory dump to the end of the task data memory with the following command
When dump_diagn 5,1,258,255,0
Run data_read to Verify that in the last SAIL task data memory block the data array is loaded with 0xFFFFs.

5.1.6.3.1 Test Results
Success
Test Conductor: JPR 2002/100

Failure Redline:
The last 2 words of sail memory did not contain 0xFFFF, it contained 0xFFFFFFFF
Reason/Resolution:
The array was sized such that the size of the array plus the size of the local variables was one short of total available sail memory. The ability of an array to use all available memory successfully demonstrated in SARR010, no requirements impact as an array of logical values is functionally equivalent in memory usage to an array of logical values.

5.1.6.4 Test Case SARR013
Use the “sarr013” stol proc to load and run sarr013.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command
When dump_diagn 5,1,258,128,0
Run data_read
Verify that in the first SAIL task data memory block the data array is loaded with numbers counting up starting from 0.

Set memory dump to the end of the task data memory with the following command
When dump_diagn 5,1,258,255,0
Run data_read to Verify that in the last SAIL task data memory block the data array is loaded with numbers counting up starting from 0.
5.1.6.4.1 Test Results

Success

Test Conductor: JPR 2002/100

5.1.6.4.5 Test Case SARR014

Use the “sarr014” stol proc to load and run sarr014.

Task ran and suspended itself

Test Engineer

Set memory dump to the beginning of the task data memory with the following command

```
@hir_tlm_diagfmt 5,1,258,128,0
```

Run data_read

 Verify that in the first SAIL task data memory block the data array is loaded with 7FEF FFFF FFFF FFFB

Test Engineer

Set memory dump to the end of the task data memory with the following command

```
@hir_tlm_diagfmt 5,1,258,255,0
```

Run data_read

Verify that in the last SAIL task data memory block the data array is loaded with

```
1.797693134862315E+308s.
```

Test Engineer

5.1.6.5.1 Test Results

Success

Test Conductor: JPR 2002/100

5.1.6.6 Test Case SARR020

This test case should generate an error. Use the “sarr020” stol proc to try to load sarr020.

Task did not load and SCCMD_REJCT inermented with Result Code 0x1FCD (F_SAI_SEG_TOO_BIG)

Test Engineer

Date

5.1.6.6.1 Test Results

Success

Test Conductor: JPR 2002/100

Procedural Redline:

```
@hir_ipu_faultclr
```

Reason/Resolution:

SARR020 generates a known error, the procedure incorrectly does not issue @hir_ipu_faultclr to clear the error flag.
5.1.6.7 Test Case SARR030

Use the "sarr030" stol proc to load and run sarr030.

Task ran and suspended itself

An memory dump to the beginning of the task data memory with the following command:

@hir_tlm_diagfmt 5,1,258,128,0

Task ran and suspended itself

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x18

Test Engineer

5.1.6.7.1 Test Results

Success

Test Conductor: JPR 2002/100

5.1.6.8 Test Case SCMP020

Use the "scmp020" stol proc to load and run scmp020.

Task ran and suspended itself

An memory dump to the beginning of the task data memory with the following command:

@hir_tlm_diagfmt 5,1,258,128,0

Task ran and suspended itself

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x45

Test Engineer

5.1.6.8.1 Test Results

Success

Test Conductor: JPR 2002/100

5.1.6.9 Test Case SCOMS010

Use the "scoms010" stol proc to load and run scoms010. This task will attempt to send a command which should be rejected the first time because the command has not been allowed. The operator must then allow the command and resume the task so that it can successfully send the command.

Task ran and suspended itself

SAIL command reject event occurred

Test Engineer

5.1.6.9.1 Test Results

Success

Test Conductor: JPR 2002/100
Send the following commands:

@hir_sai_command 1, 39
@hir_sai_resum 0

This will allow SAIL to send command #39 and resume task 0. The SAIL task should then successfully send the command. Use the spu_serpts display to verify that the detector delay for #3 was set to 0x1234.

Verify Command received.

Set memory dump to the beginning of the task data memory with the following command:

@hir_tlm_diagfmt 5, 1, 258, 128, 0

Run data_read

Verify that in the SAIL task data memory, intArray counts from 0x00 to 0x07.

---

5.1.6.9.5 Test Results:

Success

---

5.1.6.10 Test Case SCOMS020

Use the "scoms020" stol proc to load and run scoms020. This task will attempt to overflow the IPU sail command queue by sending the HIR_SSH_DOORUNPROT command as fast as possible. This task is run once with control of the sunshield enabled and once with control disabled.

Before starting scoms020, send the following commands:

@hir_sai_command 1, 46
@hir_sai_control 1, 0

start scoms020

Task ran and suspended itself

Use data_read to verify that int_array counts up to 40 and is terminated by a 1.

After the task fails, send the following commands:

@hir_sai_control 0, 0

start scoms020

Verify that in the SAIL display, the CMD_LAST_RC is 0x1fe5
5.1.6.11 Test Case SEXEC000 - SEXEC016

Use the "em_sai_exec_tst" stol proc to load 17 tasks and run 16 of them. Use TASK_STATUS for each of the 16 tasks in the SAIL display to verify that the task run counter increments and the task status is ready, running or waiting or 1, 2 or 4 until each task ran 15 times then all tasks should suspend.

All 16 tasks ran interleafed (not sequentially) 15 times and suspended.

Test Engineer

5.1.6.11.1 Test Results

Success

Test Conductor: JPR 2002/100

5.1.6.12 Test Case SEXEC020 - SEXEC035

Use the "em_sai_diag_chk" stol proc to load and run the 16 tasks. Use TASK_STATUS for each of the 16 tasks in the SAIL display to verify that the task run counter incremented and each task suspended.

All 16 tasks ran and suspended.

Test Engineer

5.1.6.11.1 Test Results

Success

Test Conductor: JPR 2002/100

All memory dump in the first two blocks of SAIL shared memory with the following command:

@hir_tlm_diagfmt 5,1,258,0,0

Run data_read

Verify that the first block of SAIL shared memory count upwards from 0x00 to 32 bit dwords.

@hir_tlm_diagfmt 5,1,258,1,0

Run data_read

Verify that the second block of SAIL shared memory finishes with 0xFF in 32 bit dwords.
Test Engineer

Turn on diagnostic item #0 using the following commands:

@hir_tlm_diagfmt 5,0,0,0,0
@hir_tlm_diagfmt 0,1,34,0xffff,0

Verify that SAIL task parameters in diagnostic item #0 count from 0x00 to 0xFF over 8 frames.

Turn off diagnostic item #0 and turn the SAIL memory dump back on.

@hir_tlm_diagfmt 0,0,0,0,0
@hir_tlm_diagfmt 5,1,258,128,0

Verify that SAIL shared memory parameters in the SAIL display count from 0x100 to 0x107 and SAIL engineering parameters count from 0x8 to 0x15.

Test Engineer

Date

5.1.6.13 Test Results

Success

5.1.6.13 Test Case: SFERR100

This test case should generate errors. Use the "sferr100" script to load and run sferr100.

Task did not run to completion, did not increment the run counter, and killed itself.

Verify correct error generation.

CURRENT ERROR

TASK NUM= 0
ERR TYPE = 253
LOCATON = 0x19

Continue after "SAILTASK_00_STAT bits 4-7 did not increment".
Continue after the first error for @hir_sai_kill
Send fault clear command
@hir_ipu_faultclr
Continue after the second error for @hir_sai_kill

5.1.6.13.1 Test Results

Success
### Test Case SFERR020

This test case should generate errors. Use the "sferr020" stol proc to load and run sferr020.

Task did not run to completion, did not increment the run counter, and killed itself.

<table>
<thead>
<tr>
<th>Test Engineer</th>
<th>Verify correct error generation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CURRENT ERROR</td>
</tr>
<tr>
<td></td>
<td>TASK NUM = 0</td>
</tr>
<tr>
<td></td>
<td>ERR TYPE = 253</td>
</tr>
<tr>
<td></td>
<td>LOCATION = 0x0f</td>
</tr>
</tbody>
</table>

Test Engineer Date

Continue after "SAILTASK_00_STAT bits 4-7 did not increment".
Continue "Hit <ENTER> when SAIL program completed".

Next command will fail two times.
Continue after the first error for @hir_sai_kill
Send fault clear command @hir_ipu_faultclr
Continue after the second error for @hir_sai_kill

### Test Results

Success

### Test Case SFERR030

This test case should generate errors. Use the "sferr030" stol proc to load and run sferr030.

Task did not run to completion, did not increment the run counter, and killed itself.

<table>
<thead>
<tr>
<th>Test Engineer</th>
<th>Verify correct error generation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CURRENT ERROR</td>
</tr>
<tr>
<td></td>
<td>TASK NUM = 0</td>
</tr>
<tr>
<td></td>
<td>ERR TYPE = 253</td>
</tr>
<tr>
<td></td>
<td>LOCATION = 0x25</td>
</tr>
</tbody>
</table>

Test Engineer Date

Continue after "SAILTASK_00_STAT bits 4-7 did not increment".
Continue "Hit <ENTER> when SAIL program completed".

Next command will fail two times.
Continue after the first error for @hir_sai_kill
Send fault clear command @hir_ipu_faultclr
Continue after the second error for @hir_sai_kill

Test Results

Success

[1]
Continue after the second error for @hir_ipu_faultclr

5.1.6.15 Test Results
Success

5.1.6.16 Test Case SFERR040
This test case should generate errors. Use the "sferr040" stol proc to load and run sferr040.
Task did not run to completion, did not increment the run counter, and killed itself.

Verify correct error generation

<table>
<thead>
<tr>
<th>Previous Error</th>
<th>Current Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK NUM = 0</td>
<td>TASK NUM = 0</td>
</tr>
<tr>
<td>ERR TYPE = 253</td>
<td>ERR TYPE = 234</td>
</tr>
<tr>
<td>LOCATION = 0x2B</td>
<td>LOCATION = 0xFD</td>
</tr>
</tbody>
</table>

Test Engineer

Continue after "SAILTASK_00_STAT bits 4-7 did not increment".

Continue "Hit <ENTER> when SAIL program completed".

Next command will fail two times.

Send fault clear command
@hir_ipu_faultclr

Continue after the second error for @hir_sai_kill

5.1.6.16.1 Test Results
Success

5.1.6.17 Test Case SFERR050
This test case should generate errors. Use the "sferr050" stol proc to load and run sferr050.
Task did not run to completion, did not increment the run counter, and killed itself.

Verify correct error generation

<table>
<thead>
<tr>
<th>Previous Error</th>
<th>Current Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK NUM = 0</td>
<td>TASK NUM = 0</td>
</tr>
<tr>
<td>ERR TYPE = 253</td>
<td>ERR TYPE = 231</td>
</tr>
<tr>
<td>LOCATION = 0x31</td>
<td>LOCATION = 0xDE</td>
</tr>
</tbody>
</table>

Test Engineer

Continue after "SAILTASK_00_STAT bits 4-7 did not increment".

Continue after the second error for @hir_ipu_faultclr

Send fault clear command
@hir_ipu_faultclr

Continue after the second error for @hir_sai_kill

5.1.6.17.1 Test Results
Success

Test Conductor: JPR 2002/100
Continue "Hit <ENTER> when SAIL program completed".
Next command will fail two times.
Continue after the first error for @hir_sai_kill
Send fault clear command @hir_ipu_faultclr
Continue after the second error for @hir_sai_kill

5.1.6.17 Test Results

Success
Test Conductor: JPR 2002/100

5.1.6.18 Test Case SFUN010

Use the "sfun010" stol proc to load and run sfun010.
Task ran and suspended itself
Set memory dump to the beginning of the task data memory with the following command:
@hir_tlm_diagfmt 5,1,258,128,0
Run data_read
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x16

Test Results
Success
Test Conductor: JPR 2002/100

5.1.6.19 Test Case SIERR010

This test case should generate errors. Use the "sierr010" stol proc to load and run sierr010.
Task ran and suspended itself
Set memory dump to the beginning of the task data memory with the following command:
@hir_tlm_diagfmt 5,1,258,128,0
Run data_read
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x06

Test Results
Success
Test Conductor: JPR 2002/100
5.1.6.20 Test Case SMEM010

Use the "smem010" stol proc to load and run smem010.

Task ran and suspended itself

Test Engineer

Set memory dump to the beginning of the task data memory with the following command

@hir_tlm_diagfmt 5,1,258,128,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x40

Test Engineer

Verify that in the SAIL task data memory block MemoryArray counts down from 0x12345678 for 15 counts

Test Engineer

Set memory dump to the beginning of the shared memory with the following command

@hir_tlm_diagfmt 5,1,258,0,0

Run data_read

Verify that the SAIL shared memory block starts with 0x12345678 and counts down for 15 counts

Test Engineer

Set memory dump to the beginning of the common shared memory with the following command

@hir_tlm_diagfmt 5,1,258,2,0

Run data_read

Verify that the SAIL shared memory block starts with 0x12345678 and counts down for 15 counts

Test Engineer

5.1.6.21 Test Results

Success

Test Conductor: JPR 2002/100

5.1.6.21 Test Case SNS/ST010

Use the "sntv010" stol proc to load and run sntv010.

Task ran and suspended itself

Test Engineer

Set memory dump to the beginning of the task data memory with the following command

@hir_tlm_diagfmt 5,1,258,0,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x3E

Test Engineer
5.1.6.21 Test Results

Task ran and suspended itself

5.1.6.22 Test Case SOP010

Use the "sop010" stol proc to load and run sop010.

Task ran and suspended itself

5.1.6.22.1 Test Results

Success

Test Conductor: JPR 2002/100

5.1.6.23 Test Case SOP020

Use the "sop020" stol proc to load and run sop020.

Task ran and suspended itself

5.1.6.23.1 Test Results

Success

Test Conductor: JPR 2002/100
5.1.6.23.1 Test Results
Success
Test Conductor: JPR 2002/100

5.1.6.23 Test Case SOVR010
This test case should generate errors. Use the "sovr010" stol proc to load and run sovr010.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command
@hir_tlm_diagfmt 5,1,258,128,0
Run data_read
Verify that in the SAIL task data memory intArray counts from 0x00 to 0x12

5.1.6.23.1 Test Results
Success
Test Conductor: JPR 2002/100

5.1.6.24 Test Case SRTL010
Use the "srtl010" stol proc to load and run srtl010.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command
@hir_tlm_diagfmt 5,1,258,128,0
Run data_read
Verify that in the SAIL task data memory intArray counts from 0x00 to 0x0D

5.1.6.24.1 Test Results
Success
Test Conductor: JPR 2002/100

5.1.6.25 Test Case SRTL010
Use the "srtl010" stol proc to load and run srtl010.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command
@hir_tlm_diagfmt 5,1,258,128,0
Run data_read
Verify that in the SAIL task data memory intArray counts from 0x00 to 0x0D

5.1.6.25.1 Test Results
Success
Test Conductor: JPR 2002/100
5.1.6.26 Test Case SST010
Use the “sst010” stol proc to load and run sst010.
Task ran and suspended itself

Test Engineer
Set memory dump to the beginning of the task data memory with the following command
@hir_tlm_diagfmt 5,1,258,128,0
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x3A

Success
Test Conductor: JPR 2002/100

5.1.6.27 Test Case SDATA010
Note: This is part of SST and requires that the TEU be interfaced with the IPU.
Use the “sdata010” stol proc to load and run sdata010
Task ran and suspended itself

Test Engineer
Set memory dump to the beginning of the task data memory with the following command
@hir_tlm_diagfmt 5,1,258,128,0
@hir_tlm_scifmt 1,0,1,1,1,0,0,0,0, 1,1,6,1,0x1fffff, 1,1
Run data_read
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x1700

Success
Test Conductor: JPR 2002/100

Procedural Redline:
The hir_tlm_scifmt command was issued prior to the issuance of the sdata010 proc.
Reason/Resolution:
Need to have the test pattern set up before the test case begins running since the test case is comparing expected pattern data to the test data, which must be commanded to the pattern in order for a successful test.
The diagfmt command was changed to use cyclic dumping instead of fixed dumping.

Reason/Resolution:
The data for this test spans multiple SAIL blocks, and therefore multiple packets. A rolling dump is necessary to sequentially verify this data using the command @hir_tlm_diagfmt 5,1,258,0,1.

The verification states that intArray counts to 0x1700, this is wrong.

Reason/Resolution:
Typo, 0x1700 should be 0x1080

5.1.6.28 Test Case STYPE010

Use the "stype010" stol proc to load and run stype010.

Task ran and suspended itself

Test Engineer

Set memory dump to the beginning of the task data memory with the following command
@hir_tlm_diagfmt 5,1,258,128,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x08

Test Engineer

Success

5.1.6.28.1 Test Results

Task ran

Test Conductor: JPR 2002/100

5.1.6.29 Test Case WAIT010 – WAIT020

Use the "wait010" stol proc to load and run wait010.

Task is running or waiting

Test Engineer

Suspend the task with the following command
@hir_sai_supend 0

Verify that the task is suspended

Resume the task with the following command
@hir_sai_resume 0

Load and run wait020 as task #1 using the "wait20" stol proc

Verify that task #1 ran and suspended.

Verify that task #0 was suspended.
Resume the task with the following command:
@hir_sai_resume 0

Kill the task with the following:
@hir_sai_kill 0

Verify that the task was killed

Date

Test Engineer

Enter "go" in the CSTOL window after the wait020 stol proc completes erasing task 1.

Ignore errors in the wait010 stol proc, choosing "y" to continue after each error.

Clear all error flags with the following:
@hir_ipu_faultclr

5.1.6.29 Test Results

Success

Procedural Redline:
The procedure calls for a wait020 stol proc, but no such proc exists.

Reason/Resolution:
Unknown why the proc does not exist. Manually issued the following commands which accomplished the task of loading, creating, and resuming the wait020 SAIL task:

start wait020_d
start wait020_c
@hir_sai_create 1, 5, 129, 0
@hir_sai_resume 1

5.1.6.30 Test Case STOR010

Use the "wait010" stol proc to load wait010 into task 0 and then start it running.

Task 0 is running or waiting.

Procedural Redline:
Wait010 stol proc did not exist. Issued following commands to kill the task:
@hir_sai_kill 0
@hir_sai_erase 129
@hir_sai_erase 5

Reason/Resolution:
Wait010 stol proc did not exist. Issued following commands to kill the task:
@hir_sai_kill 0
@hir_sai_erase 129
@hir_sai_erase 5

5.1.6.39 Test Case STOR030

Use the "wait030" stol proc to load wait030 into task 0 and then start it running.

Task 0 is running or waiting.
Use the "stovr010" stol proc to load stovr010 into task 1 and then start it running.

Task 1 suspended with error

Verify correct error generation

task = 1
type = 238

Verify task 0 is still running

Verify task 1 suspended

5.1.6.30.1 Test Results

Success

Test Conductor: JPR 2002/100

Procedural Redline:
The procedure calls for a wait030 stol proc, but no such proc exists.

Reason/Resolution:
Unknown why the proc does not exist, manually issued the following commands which accomplished the task of
loading, creating, and resuming the wait030 SAIL task:

start wait030_d
start wait030_c
@hir_sai_create 2, 6, 130, 0
@hir_sai_resume 2

Procedural Redline:
The procedure calls for a stovr010 stol proc, but no such proc exists.

Reason/Resolution:
Unknown why the proc does not exist, manually issued the following commands which accomplished the task of
loading, creating, and resuming the stovr010 SAIL task:

start stovr010_d
start stovr010_c
@hir_sai_create 1, 5, 129, 0
@hir_sai_resume 1
5.1.6.31 Test Case SASM
Use the "sasm" stol proc to load sasm.
Verify that the task was loaded

5.1.6.31.1 Test Results
Success
Test Conductor: JPR 2002/100

Procedural Redline:
Task 2 still running.
Reason/Resolution:
Not killed from previous test case, redlines:
@hir_sai_stop 0xdead
@hir_sai_start 4

Procedural Redline:
This Test case was incompletely entered into the test procedure.
Reason/Resolution
Unknown why it was not entered completely, outlined steps followed
start sasmoversizelimit_c
start sasmsizelimit_c

5.2 Software Acceptance Regression Test 1 (2003-06-17)

5.2.1 Starting the Hardware
Power on the instrument, ops order will direct which side to use:
Start hir_qba_on  (a side)
Start hir_qbb_on  (b side)

5.2.1.1 Test Results
Success
Test Conductor: JPR 2002-06-17, Day 168

5.2.2 Starting the IEGSE
Start a xterm on the IEGSE computer. In the xterm clean and start the IEGSE by typing:

start_testdisplay $IEGSE_DISPLAYS/sail

5.2.2.1 Test Results
Success
Test Conductor: JPR 2002-06-17, Day 168
5.2.3 Aliveness Test

Power on the IPU and obtain data in the IEGSE displays.

5.2.4 Record Software Version Numbers

- IPU Application code version number
- IEGSE version number

5.2.4.1 Test Results

Test Conductor: JPR 2002-06-17, Day 168

Success

5.2.5 Start and Setup SAIL

Send the following set of commands:

- `@hir_sai_start`
- `@hir_tlm_scifmt 1,0,0,0,0,0,0,0,0,1,1,0,1,x#1FFFFFF,1,1` // need to have room for one SAIL block 128*2 16bit words
- `@hir_tlm_diagfmt 1,0,0,0,0`
- `@hir_tlm_diagfmt 11,0,0,0,0`
- `@hir_tlm_diagfmt 14,0,0,0,0`
- `@hir_tlm_diagfmt 5,1,258,128,0`

5.2.5.1 Test Results

Test Conductor: JPR 2002-06-17, Day 168

Success

Procedural Redline

The `hir_sai_start` STOL proc started the SAIL processor then unexpectedly loaded and started two additional unknown tasks.

Reason/Resolution

This proc was modified since the prior acceptance test event. The proc was cancelled, and the SAIL processor was stopped and restarted manually with the following two commands:

- `@hir_sai_stop 0xdead`
- `@hir_sai_start 4`

The above two commands were sent prior to sending the `@hir_tlm_scifmt` command.

5.2.6 Test Cases/Tasks

There is a stol proc for each of the SAIL tasks, most start by loading the code into SAIL memory block #4 and data into SAIL memory block #128. Next SAIL task #0 is created from those memory blocks and then run. The stol proc is halted waiting for input so the operator can verify that the task completed correctly. After verification, hit the Enter key to continue the stol proc which will kill the SAIL task and delete the used SAIL memory blocks, making SAIL ready for the next task. Then hit Ctrl-C to close the stol proc window.

To start the stol proc:

1. Input `start` the stol proc's name to the IEGSE.
2. Use `TASK_STATUS` for task 0 of the SAIL display to verify that the task ran, counter incremented and the task suspended. Here are the task status meanings:

   - Halted = 0
   - Ready = 1
   - Running = 2
This status nibble can be found in the lower nibble of the TASK_STATUS word. The upper nibble is a counter which counts up to 0xF (15) before resetting with zero. This counter will change any time a task waits or ends.

1) Set memory dump to the start of each task's data memory. Use data_read to Verify that the output array is correctly loaded. Verify that the Task Status nibble (Suspended and Wait) for the task is set. The status is represented in the upper nibble of the TASK_STATUS word. This field will be the sequence of transitions.

2) Set memory dump to the start of task's data memory. Use data_read to Verify that the output array is correctly loaded. The contents of these arrays will not be in an exact location, but all that is required is to find the sequence somewhere in the diagnostic area for SAIL memory in the science packet. Somewhere in that block will be the sequence of transitions.

5.2.6.1 SAIL Test Case (SAIL550, SAIL250)

5.2.6.1.1 SAIL Task SARR010 (SAIL250)

Use the "sarr010" stol proc to load and run sarr010.

Test Engineer

Task ran and suspended itself.

If it is not already set, set memory dump to the beginning of the task data memory with the following command

\texttt{@hir_tlm_diagfmt 5,1,258,128,0}

Run data_read.

Verify that in the first SAIL task data memory block the data array is loaded with 1’s.

Test Engineer

Set memory dump to the end of the task data memory with the following command

\texttt{@hir_tlm_diagfmt 5,1,258,255,0}

Use data_read to Verify that in the last SAIL task data memory block the data array is loaded with 1’s.

Test Engineer

5.2.6.1.1.1 Test Results

Success.

Test Conductor: JPR 2002-06-17, Day 168

5.2.6.1.2 SAIL Task SARR011 (SAIL250)

Use the "sarr011" stol proc to load and run sarr011.

Task ran and suspended itself.

Set memory dump to the beginning of the task data memory with the following command

\texttt{@hir_tlm_diagfmt 5,1,258,128,0}

Run data_read.

Verify that in the first SAIL task data memory block the data array is loaded with 0xFFs.

Test Engineer

Set memory dump to the end of the task data memory with the following command

\texttt{@hir_tlm_diagfmt 5,1,258,255,0}

Use data_read to Verify that in the last SAIL task data memory block the data array is loaded with 0xFFs.

Test Engineer

5.2.6.1.1.1 Test Results

Success.

Test Conductor: JPR 2002-06-17, Day 168
Set memory dump to the end of the task data memory with the following command

@hir_tlm_diagfmt 5,1,258,255,0

Use data_read to verify that in the last SAIL task data memory block the data array is loaded with 0xFFs.

5.2.6.1.2 Test Results
Success
Test Conductor: JPR 2002-06-17, Day 368

5.2.6.1.3 SAIL Task SARR012 (SAIL250)
Use the “sarr012” stol proc to load and run sarr012.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command

@hir_tlm_diagfmt 5,1,258,128,0

Run data_read
Verify that in the first SAIL task data memory block the data array is loaded with 0xFFFFs.

Set memory dump to the end of the task data memory with the following command

@hir_tlm_diagfmt 5,1,258,255,0

Use data_read to verify that in the last SAIL task data memory block the data array is loaded with 0xFFFFs.

5.2.6.1.3.1 Test Results
Success
Test Conductor: JPR 2002-06-17, Day 368

5.2.6.1.4 SAIL Task SARR013 (SAIL250)
Use the “sarr013” stol proc to load and run sarr013.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command

@hir_tlm_diagfmt 5,1,258,128,0

Run data_read
Verify that in the first SAIL task data memory block the data array is loaded with numbers counting up starting from 0.
Test Engineer

Set memory dump to the end of the task data memory with the following command:

```
@hir_tlm_diagfmt 5,1,258,255,0
```

Use `data_read` to verify that in the last SAIL task data memory block the data array is loaded with numbers counting up ending with 16349 or 0x3FDD.

---

Test Results:

5.2.6.1.4.1

Test: Test Conductor: JPR 2002-06-17, Day 168

Success

---

5.2.6.1.5

SAIL Task SARR014 (SAIL250)

Use the "sarr014" stol proc to load and run SARR014.

Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command:

```
@hir_tlm_diagfmt 5,1,258,128,0
```

Run `data_read` to verify that in the first SAIL task data memory block the data array is loaded with 7FEF FFFF FFFF FFFB.

Set memory dump to the end of the task data memory with the following command:

```
@hir_tlm_diagfmt 5,1,258,255,0
```

Use `data_read` to verify that in the last SAIL task data memory block the data array is loaded with 1.797693134862315E+308s.

---

5.2.6.1.5.1

Test Results:

Test Conductor: JPR 2002-06-17, Day 168

Success

---

5.2.6.1.6

SAIL Task SARR020 (SAIL550)

This test case should generate an error. Use the "sarr020" stol proc to try to load SARR020.

Task did not load and SCCMD_REJCT increment with Result Code 0x1FCD (F_SAI_SEG_TOO_BIG)

Clear the error flag after stol proc is finished running.

@hir_ipu_faultclr

---

5.2.6.1.6.1

SAIL Task SARR030 (SAIL550)

This test case should generate an error. Use the "sarr030" stol proc to try to load SARR030.

Task did not load and SCCMD_REJCT increment with Result Code 0x1FCD (F_SAI_SEG_TOO_BIG)

Clear the error flag after stol proc is finished running.

@hir_ipu_faultclr
5.2.6.1.6.1 Test Results

Success
Test Conductor: JPR 2002-06-17, Day 368
Verification Source: sarr020_2002_06_17_verification.txt

Reason/Resolution
Upon examining the data post test, the result code 0x1FCD did correctly appear when the upload was initially rejected. This canceled the upload state such that subsequent upload commands generated the result code F_UPL_BLKNUM_SEQR which is used when upload commands are out of order, which is expected after the upload is canceled.

5.2.6.1.7 SAIL Task SARR030

Use the "sarr030" stol proc to load and run SARR030.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command
@hir_tlm_diagfmt 5,1,258,128,0
Run data_read
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x18

5.2.6.2 SCMP Test Case (SAIL280, SAIL290, SAIL300)

5.2.6.2.1 SCMP Task SCMP020 (Flight Portion) (SAIL280, SAIL290, SAIL300)

Use the "scmp020" stol proc to load and run SCMP020.
Task ran and suspended itself
Set memory dump to the beginning of the task data memory with the following command:
@
hir_tlm_diagfmt 5,1,258,128,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x07

5.2.6.3.1 Test Results:

Date: 2003-06-17

Success

Task ran and suspended itself

SAIL task stayed and sent command count incremented

Send the following commands:
@
hir_sai_command 1,39
@
hir_sai_resum 0

This will allow SAIL to send command #39 and resume task 0. The SAIL task should then successfully send the command. Use the spu_serpts display to verify that the detector delay for #3 was set to 0x1234.

Verify Command executed.

Set memory dump to the beginning of the task data memory with the following command:
@
hir_tlm_diagfmt 5,1,258,128,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x45

5.2.6.3.1.1 Test Results:

Success
5.2.6.3.2 SAIL Task (SAIL020) (SAIL020, IPS1130)

This task will attempt to overflow the IPU sail command queue by sending the HIR_SSH_DOORUNPROT command as fast as possible. This task is run once with control of the sunshield enabled and once with control disabled.

Before starting scoms020, send the following commands:

```
@hir_sai_command 1, 46
@hir_sai_control 1, 0
start scoms020
```

Task ran and suspended itself.

Use data_read to verify that int_array counts up to 40 and is terminated by a 1.

```
@hir_sai_control 0, 0
start scoms020
```

Verify that in the SAIL display, the CMD_LAST_RC is 0x1fe5

5.2.6.3.2.1 Test Results:

Success

Test Conductor: JPR 2002-06-17, Day 168

5.2.6.4 SEXEC Test Case (SAIL480, SAIL490, SAIL580)

5.2.6.4.1 SAIL Tasks (SEXEC000, SEXEC016) (SAIL480, SAIL490, SAIL580)

Use the "em_sai_exec_tst" stol proc to load 17 tasks and run 16 of them. Use TASK_STATUS for each of the 16 tasks in the SAIL display to verify that the task run counter increments and the task status is ready, running or waiting, 1, 2 or 4 until each task ran 15 times then all tasks should suspend.

All 16 tasks ran interleafed (not sequentially) 15 times and suspended.

Verify that the new task ran and suspended.

5.2.6.4.1.1 Test Results:

Success

Test Conductor: JPR 2002-06-17, Day 168
5.2.6.4.2 SAIL Tasks SEXEC020 - SEXEC035

Test the "em_sai_diag_chk" stol proc to load and run the 16 tasks. Use TASK_Status for each of the 16 tasks in the SAIL display to verify that the task run counter incremented and each task suspended.

All 16 tasks ran and suspended

Test Engineer

Set memory dump to the first two blocks of SAIL shared memory with the following commands:

@hir_tlm_diagfmt 5,1,258,0,0
Run data_read
Verify that the first block of SAIL shared memory contains upwards from 0x00 to 32 bit dwords.

@hir_tlm_diagfmt 5,1,258,1,0
Run data_read
Verify that the second block of SAIL shared memory finishes with 0xFF in 32 bit dwords.

Test Engineer

Turn off SAIL memory dump and turn on diagnostic item #0 using the following commands:

@hir_tlm_diagfmt 5,0,0,0,0
@hir_tlm_diagfmt 0,1,34,0xffff,0
Verify that SAIL task parameters in diagnostic item #0 count from 0x00 to 0xFF over 8 frames.

Test Engineer

Turn off diagnostic item #0 and turn the SAIL memory dump back on.

@hir_tlm_diagfmt 0,0,0,0,0
@hir_tlm_diagfmt 5,1,258,128,0
Verify that SAIL shared memory parameters in the SAIL display count from 0x100 to 0x107 and SAIL engineering parameter count from 8 to 15.

Test Engineer

5.2.6.5 SFERR Test Case (SAIL610)

5.2.6.5.1 SAIL Task SFERR010

5.2.6.1.1 Test Results

Success

Test Conductor: JPR 2002-06-17, Day 168

5.2.6.5 SFERR Test Case (SAIL641)

5.2.6.5.5 SAIL Task SFERR010
This test case should generate errors. Use the "sferr010" stol proc to load and run sferr010.
Task did not run to completion, did not increment the run counter, and killed itself.

Verify correct error generation

Current error

TASK NUM = 0
ERR TYPE = 253
LOCATION = 0x19

Test Engineer

Date

Continue after "SAILTASK_00_STAT bits 4-7 did not increment".
Continue "Hit <ENTER> when SAIL program completed".
Next command will fail two times.
Continue after the first error for @hir_sai_kill
Send fault clear command
@hir_ipu_faultclr
Continue after the second error for @hir_sai_kill

5.2.6.5.1 Test Results

Continue...

5.2.6.7 SAIL Task SFERR020

This test case should generate errors. Use the "sferr020" stol proc to load and run sferr020.
Task did not run to completion, did not increment the run counter, and killed itself.

Verify correct error generation

Current error

PREVIOUS ERROR

TASK NUM = 0
ERR TYPE = 253
LOCATION = 0xbf

Test Engineer

Date

Continue after "SAILTASK_00_STAT bits 4-7 did not increment".
Continue "Hit <ENTER> when SAIL program completed".
Next command will fail two times.
Continue after the first error for @hir_sai_kill
Send fault clear command
@hir_ipu_faultclr
Continue after the second error for @hir_sai_kill

56
5.2.6.5.2 Test Results

Success

Test Conductor: JPR 2002-06-17, Day 168

5.2.6.5.3 SAIL Task SFERR030

This test case should generate errors. Use the "sferr030" stol proc to load and run sferr030.

Task did not run to completion, did not increment the run counter, and killed itself.

Test Engineer: Verify correct error generation

PREVIOUS ERROR          CURRENT ERROR
TASK NUM = 0 TASK NUM = 0
ERR TYPE = 253 ERR TYPE = 230
LOCATION = 0x25 LOCATION = 0xED

Test Engineer: Date

Continue after "SAILTASK_00_STAT bits 4-7 did not increment".
Continue "Hit <ENTER> when SAIL program completed".
Send fault clear command

5.2.6.5.3.1 Test Results

Success

Test Conductor: JPR 2002-06-17, Day 168

5.2.6.5.4 SAIL Task SFERR040 (SAIL610)

This test case should generate errors. Use the "sferr040" stol proc to load and run sferr040.

Task did not run to completion, did not increment the run counter, and killed itself.

Test Engineer: Verify correct error generation

PREVIOUS ERROR          CURRENT ERROR
TASK NUM = 0 TASK NUM = 0
ERR TYPE = 253 ERR TYPE = 234
LOCATION = 0x2B LOCATION = 0xFD

Test Engineer: Date
Continue after "SAILTASK_00_STAT bits 4-7 did not increment".
Continue "Hit <ENTER> when SAIL program completed".
Next command will fail two times.
Continue after the first error for @hir_sai_kill
Send fault clear command
@hir_ipu_faultclr
Continue after the second error for @hir_sai_kill

5.2.6.5.4 Test Results
Success
Test Conductor: JPR 2002-06-17, Day 168

5.2.6.5.5 SAIL Task SFERR050
This test case should generate errors. Use the "sferr050" stol proc to load and run sferr050.
Task did not run to completion, did not increment the run counter, and killed itself.

Verify correct error generation

<table>
<thead>
<tr>
<th>PREVIOUS ERROR</th>
<th>CURRENT ERROR</th>
<th>ERR TYPE</th>
<th>LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK NUM = 0</td>
<td>TASK NUM = 0</td>
<td>ERR TYPE = 253</td>
<td></td>
</tr>
<tr>
<td>LOCATION = 0x31</td>
<td>LOCATION = 0xDE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Test Engineer

5.2.6.6 SFUN Test Case (SAIL310, SAIL320, SAIL400)

5.2.6.6.1 SAIL Task SFUN010 (SAIL310, SAIL320)
Use the "sfun010" stol proc to load and run sfun010.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command
@hir_tlm_diagfmt 5,1,258,128,0
5.2.6.6.1 Test Results

Set memory dump to the beginning of the task data memory with the following command:
@hir_tlm_diagfmt 5,1,258,128,0
Task ran and suspended itself

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x0D

Test Engineer

Date

5.2.6.6.2 SAIL Task SIERR010 (SAIL400)

Set memory dump to the beginning of the task data memory with the following command:
@hir_tlm_diagfmt 5,1,258,128,0
Task ran and suspended itself

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x06

Test Engineer

Date

5.2.6.7 SIERR010 Test Case (SAIL570)

5.2.6.7.1 SAIL Task SIERR010 (SAIL570)

This test case should generate errors. Use the “sierr010” stol proc to load and run the test case.
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command:
@hir_tlm_diagfmt 5,1,258,128,0
Task ran and suspended itself

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x0D

Test Engineer

Date

5.2.6.7.1 Test Results
5.2.6.8 SMEM Test Case (SAIL520, SAIL590, SAIL640)

5.2.6.8.1 SAIL Task SMEM010 (SAIL520, SAIL590, SAIL640)

Use the “smem010” stol proc to load and run smem010.

Task ran and suspended itself

Test Engineer

Set memory dump to the beginning of the task data memory with the following command:

@hir_tlm_diagfmt 5,1,258,128,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x40

Test Engineer

Verify that in the SAIL task data memory block MemoryArray counts down from 0x12345678 for 15 counts

Test Engineer

Set memory dump to the beginning of the shared memory with the following command:

@hir_tlm_diagfmt 5,1,258,0,0

Run data_read

Verify that the SAIL shared memory block starts with 0x12345678 and counts down for 15 counts

Test Engineer

Set memory dump to the beginning of the common shared memory with the following command:

@hir_tlm_diagfmt 5,1,258,2,0

Run data_read

Verify that the SAIL shared memory block starts with 0x12345678 and counts down for 15 counts

5.2.6.8.1.1 Test Results

Success

Test Conductor: JPR 2002-06-17, Day 168

5.2.6.9 SNST Test Case

5.2.6.9.1 SAIL Task SNST010

Use the “snst010” stol proc to load and run snst010.
Set memory dump to the beginning of the task data memory with the following command:

@hir_tlm_diagfmt 5,1,258,128,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x3E.

Set memory dump to the beginning of the task data memory with the following command:

@hir_tlm_diagfmt 5,1,258,129,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x5F to 0x60.

5.2.6.11.0 Test Results:

Success

Test Conductor: JPR 2002-06-17, Day 168.
5.2.6.10.2 SAIL Task SOP020

Use the "sop020" stol proc to load and run sop020.

Task ran and suspended itself

Test Engineer

Set memory dump to the beginning of the task data memory with the following command

@hir_tlm_diagfmt 5,1,258,128,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x1D

Test Engineer

5.2.6.10.2.1 Test Results

Success

Test Conductor: JPR 2002-06-17, Day 308

5.2.6.11 SOVR Test Case (SAIL450, SAIL460)

5.2.6.11.1 SAIL Task SOVR010 (SAIL450, SAIL460)

This test case should generate errors. Use the "sovr010" stol proc to load and run sovr010.

Task ran and suspended itself

Test Engineer

Set memory dump to the beginning of the task data memory with the following command

@hir_tlm_diagfmt 5,1,258,128,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x0D

Test Engineer

Date

@hir_ipu_faultclr to clear SAIL error telemetry

5.2.6.11.1.1 Test Results

Success

Test Conductor: JPR 2002-06-17, Day 308

5.2.6.12 SST Test Case (SAIL050, SAIL090, SAIL100, SAIL160, SAIL260, SAIL340, SAIL470)

5.2.6.12.1 SAIL Task SST010 (SAIL050, SAIL090, SAIL100, SAIL160, SAIL260, SAIL340, SAIL470)

Use the "sst010" stol proc to load and run sst010.

Test Engineer

When "end" is entered error telemetry is cleared.

5.2.6.12.1.1 Test Results

Success

Test Conductor: JPR 2002-06-17, Day 308

5.2.6.12.2 SST Test Case (SAIL010, SAIL030, SAIL050, SAIL090, SAIL100, SAIL160, SAIL260, SAIL340, SAIL470)

5.2.6.12.2.1 SAIL Task SIT010 (SAIL010, SAIL030, SAIL050, SAIL090, SAIL100, SAIL160, SAIL260, SAIL340, SAIL470)

Use the "siti" stol proc to load and run siti.

Date
Task ran and suspended itself

Set memory dump to the beginning of the task data memory with the following command:
@hir_tlm_diagfmt 5,1,258,128,0
Run data_read
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x3A

5.2.6.12.2.1 Test Results:
Success
Test Conductor: JPR 2002-06-17, Day 168

5.2.6.12.2.2 SAIL Task SDATA010 (SAIL070, SAIL080)

Turn on the SAIL test pattern
@hir_tlm_scifmt 1,0,1,1,1,0,0,0,0, 1,1,6,1,0x1fffff, 1,1

Use the “sdata010” stol proc to load and run sdata010
Task ran and suspended itself
Set memory dump to the beginning of the task data memory with the following command:
@hir_tlm_diagfmt 5,1,258,0,1
Run data_read
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x1080

5.2.6.12.2.2.1 Test Results:
Success
Test Conductor: JPR 2002-06-17, Day 168

5.2.6.13 STYPE Test Case (SAIL200, SAIL210, SAIL220, SAIL230, SAIL240, SAIL440)

Use the “stype010” stol proc to load and run stype010
Task ran and suspended itself

Test Engineer

Set memory dump to the beginning of the task data memory with the following command:
@hir_tlm_diagfmt 5,1,258,128,0

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x08

5.2.6.13.1 Test Results:

Date

5.2.6.14 Test Case WAIT (SAIL600, SAIL640)

Suspend the task with the following command:
@hir_sai_suspend 0

Verify that the task is suspended

Resume the task with the following command:
@hir_sai_resume 0

Load and run wait020 as task #1 with the following commands:
start wait020_d
start wait020_c
@hir_sai_create 1, 5, 129, 0
@hir_sai_resume 1

Verify that task #1 ran and suspended.

Verify that task #0 was suspended.

Resume the task with the following command:
@hir_sai_resume 0

Kill the task with the following command:
Verify that the task was killed.

Scan error in the wait010 wait proc, choosing "y" to continue when such error.
Clear all error flags with the following:
@hir_ipu_faultclr

Erase task 1:
@hir_sai_kill 1
@hir_sai_erase 129
@hir_sai_erase 5

5.2.6.14.1 Test Results
Success
Test Conductor: JPR 2002-06-17, Day 368

5.2.6.15 STOVR Test Case (SAIL510)

5.2.6.15.1 Task STOVR010 (SAIL510)

Use the "wait010" stol proc to load wait010 into task 0 and then start it running.
Task 0 is running or waiting

Load wait030 into task 2 and then start it running with the following commands.
start wait030_d
start wait030_c
@hir_sai_create 2, 6, 130, 0
@hir_sai_resume 2
Task 2 is running or waiting

Load stovr010 into task 1 and then start it running with the following commands.
start stovr010_d

SUCCESS
5.2.6.15.1.1 Test Results:  
Pass.  
Test Conductor: JPR 2002-06-17, Day 168 

5.2.6.16 Test Case SASM_BIG (SAIL540):  
5.2.6.16.1 SAIL Task SASMsizeLimit (SAIL540)  
Kill and restart sail  
@hir_sai_stop 0xdead  
@hir_sai_start 4  
Load task sasmsizelimit_c  
Verify that the task was loaded  
Pass.  
Test Conductor: JPR 2002-06-17, Day 168 

5.2.6.16.2 SAIL Task SASMoverSizeLimit (SAIL540)  
Load task sasmoversizelimit_c  
Verify that the task was loaded  
Pass.  
Test Conductor: JPR 2002-06-17, Day 168
Verify that the task was not loaded

Test Engineer
Date

5.2.6.16.2 Test Results

Test Conductor: JPR 2002-06-17, Day 168
Reason/Resolution
The upload header command contains two nibbles indicating the expected size of the upload. These nibbles are now ignored. The SAIL processor will reject any attempt to load too many blocks in a segment. The SAIL processor will reject any attempt to load a task that is too large. This is verified by test case SARR020 which attempts to load a task which is too large and fails. The SAIL Processor no longer rejects uploads on the basis of the upload header command but will reject any actual attempt to load a task that is too large.

5.3 SAIL Acceptance Regression Test 2 (2003-06-05)

Starting the Hardware
On the SUN workstation connected via the ODP, log in as root, and open an xterm in the
`/hirdls/tornado/ipu/app_code/[IPU_VERSION]/build` directory. Where IPU_VERSION is the name of the directory containing the current IPU code build. Now, run the script `kermitA`. Power on the ODP by pushing the large orange button on the ODP lab power supply, then toggling the "b" reset switch on the ODP reset box. The Kermit window should show the ODP SUROM memory scrubbing or counting down to autoboot. Once the memory scrub is finished, run the script `loadimage`. You can hit the spacebar in the Kermit window to skip the memory scrub if desired.

Image successfully loaded:

Starting the IEGSE
Start a xterm on the IEGSE computer. In the xterm, clean and start the IEGSE by typing:
`
iegse_clean;iegse_start`
Start the SAIL display:
`
start_testdisplay $IEGSE_DISPLAYS/sail`

Aliveness Test
Power on the IPU and obtain data in the IEGSE displays.

Record Software Version Numbers

IPU Application code version number      3.6.30
IEGSE version number   4.10

Test Engineer
Date

5.3.1 Test Results

Test Conductor: JPR 2002-06-10

Procedural Redlines: Bulletize instructions for clearer reading in the future. Include version numbers for any associated tools. the loadimage tool is now started by: loadimage ipu a
5.3.2 Test Cases/Tasks

There is a stol proc to control each SAIL task.

Here are the task status meanings.

- Halted = 0
- Ready = 1
- Running = 2
- Suspended = 3
- Wait = 4

This status nibble can be located in the lower nibble of the TASK_STATUS word. The upper nibble is a counter, which counts up to 0xF and then starts over with zero. This counter will change any time a task waits or ends.

An memory dump is the start of each task's data memory. For data read vs. Verify that the output array is correctly loaded.

The contents of these arrays might not be in the exact location, but all is required is to watch for the sequence x subsequences on the diagnostic area for SAIL memory in the science packet. Somewhere in that block will be the sequence of numbers.

5.3.2.1 SARR Test Case

5.3.2.1.1 SAIL Task SARR010

From the CSTOL operator window, enter "start sarr010".

Initialize SAIL by entering 10 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Set memory dump to the start of task's data memory. Use data_read to Verify that the output array is correctly loaded.

The contents of these arrays might not be in the exact location, but all that is required is to watch for the sequence in subsequences on the diagnostic area for SAIL memory in the science packet. Somewhere in that block will be the sequence of numbers.

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

If it is not already set, set memory dump to the beginning of the task data memory with the following command:

```
@hir_tlm_diagfmt 5,1,258,128,0
```

Run data_read

Verify that in the first SAIL task data memory block the data array is loaded with 1's from byte 67 to 257.

Kill/Erase the task by entering 14 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.1.1.1 Test Results

Success
3.5.3.1.2 SAIL Task SARR011

From the CSTOL operator window, enter "start sarr011".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Test Engineer

Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.
Enter 128 when asked to enter a SAIL block.
Run data_read.
Verify that in the first SAIL task data memory block the data array is loaded with 0xFFs ending with one entry on line 256.

Test Engineer

Set memory dump to the end of the task data memory by entering 8 when prompted to enter menu choice.
Enter 255 when asked to enter a SAIL block.
Run data_read.
Verify that in the first SAIL task data memory block the data array is loaded with 0xFFFFs up to line 248.

Test Engineer

Kill/Erase the task by entering 14 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Test Engineer

5.3.2.1.2.1 Test Results: Success

Test Conductor: JPR 2003-06-10

3.5.3.1.3 SAIL Task SARR012

From the CSTOL operator window, enter "start sarr012".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Test Engineer

Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.
Enter 128 when asked to enter a SAIL block.
Run data_read.
Verify that in the first SAIL task data memory block the data array is loaded with 0xFFFFs ending with one entry on line 256.

Test Engineer

Set memory dump to the end of the task data memory by entering 8 when prompted to enter menu choice.
Enter 255 when asked to enter a SAIL block.
Run data_read.
Verify that in the first SAIL task data memory block the data array is loaded with 0xFFFFs up to line 248.

Test Engineer

5.3.2.1.3 Test Results: Success

Test Conductor: JPR 2003-06-10
Test Procedure:

**5.3.2.1.3 Test Results:**
- **Success**
- **Test Conductor:** JPR 2003-06-10

**Task SARR013**
- From the CSTOL operator window, enter "start sarr013".
- Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
- Verify procedure returns to main menu without error.
- Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.
- Enter 128 when asked to enter a SAIL block.
- Run data_read
- Verify that in the first SAIL task data memory block the data array is loaded with numbers counting up starting from 0.
- Set memory dump to the end of the task data memory by entering 8 when prompted to enter menu choice.
- Enter 255 when asked to enter a SAIL block.
- Run data_read
- Verify that in the last SAIL task data memory block the data array is loaded with numbers counting up to 16349 or 0x3FDD.

**5.3.2.1.4.1 Test Results:**
- **Success**
- **Test Conductor:** JPR 2003-06-10

**Task SARR014**
- From the CSTOL operator window, enter "start sarr014".
- Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
- Verify procedure returns to main menu without error.
- Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.
- Enter 128 when asked to enter a SAIL block.
- Run data_read
- Verify that in the first SAIL task data memory block the data array is loaded with 7FEF FFFF FFFF FFFB starting at byte 64 up to 257.

---

Test Engineer
5.3.2.1.5 SAIL Task SARR020

From the CSTOL operator window, enter "start sarr020".

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

5.3.2.1.6 SAIL Task SARR030

From the CSTOL operator window, enter "start sarr030".

Load by entering 1 when prompted to enter menu choice.

This test case should generate an error.

Verify that the task did not load and SCCMD_REJCT incremented with Result Code 0x1FCD (F_SAI_SEG_TOO_BIG).

Enter "n" when prompted to Continue after failure, then enter "go" in the CSTOL window.

Halt SAIL by entering 11 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Initialize SAIL by entering 10 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.1.6.1 Test Results:

Success
5.3.2.2 SCMP Test Case

5.3.2.2.1 SAIL Task SCMP020 (Flight Portion)

From the CSTOL operator window, enter "start scmp020".

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.

Enter 128 when asked to enter a SAIL block.

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x45

Kill/Erase the task by entering 14 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Test Engineer

Date

5.3.2.2.1.1 Test Results

Success

Test Conductor: JPR 2003-06-10

5.3.2.3 SCOMS Test Case

5.3.2.3.1 SAIL Task SCOMS010

This task will attempt to send a command which should be rejected the first time because the command has not been allowed. The operator must then allow the command and the task to then run successfully with the command.

From the CSTOL operator window, enter "start scomso10".

Load, Create, Resume, and Monitor task to suspension by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

SAIL command reject count increments

Test Engineer

Date

5.3.2.3.1.1 Test Results

Success

Test Conductor: JPR 2003-06-10

5.3.2.4 SCOMS Test Case

5.3.2.4.1 AAIL Task SCOMS0000

This task will attempt to send a command which should be rejected the first time because the command has not been allowed. The operator must then allow the command and the task to then run successfully with the command.

From the CSTOL operator window, enter "start scoms0000".

Load, Create, Resume, and Monitor task to suspension by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

SAIL command reject count increments

Test Engineer

Date

5.3.2.4.1.1 Test Results

Success

Test Conductor: JPR 2003-06-10
Send the following command from the CSTOL operator window:

```plaintext
@hir_sai_command 1,41
```
Resume the task by entering 3 when prompted to enter menu choice. This will allow SAIL to send command #41 and resume task 0. The SAIL task should then successfully send the command. Enter the following command in the CSTOL operator window:

```plaintext
write SPU_CH_01_ZERO
```
Verify procedure returns to main menu without error and SPU_CH_01_ZERO = 4660.

---

**5.3.2.3.1 Test Results:**

**Success**

---

**5.3.2.3.2 SAIL Task SCOMS020**

This task will attempt to overflow the IPU sail command queue by sending the HIR_SSH_DOORUNPROT command as fast as possible. This task is run once with control of the sunshield enabled and once with control disabled.

Send the following commands:

```plaintext
@hir_sai_command 1, 46
@hir_sai_control 1, 0
```
Verify that the reject count did not increment.

---

**5.3.2.3.3 Test Results:**

**Success**
Verify that the reject count did not increment.

Test Procedure
- Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
- Verify procedure returns to main menu without error.
- Kill/Erase the task by entering 14 when prompted to enter menu choice.
- Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.3.3 Test Case

5.3.2.3.3.1 Test Results

Success

Test Conductor: JPR 2003-06-10

5.3.2.4 SEXEC Test Case

5.3.2.4.1 SAIL Tasks SEXEC000 - SEXEC016

Use the "sexec" stol proc to load 17 tasks and run 16 of them. Use TASK_STATUS for each of the 16 tasks in the SAIL display to verify that the task run counter increments and the task status is ready, running or waiting or "1", "2" or "4" until each task ran 15 times then all tasks should suspend.

From the CSTOL operator window, enter "start sexec".
- Verify that all 15 tasks ran interleafed (not sequentially) 16 times and suspended, then enter "go" into the CSTOL window.
- Verify that the new task 1 ran and suspended, then enter "go" into the CSTOL window.

5.3.2.4.1.1 Test Results

Success

Test Conductor: JPR 2003-06-10
5.3.2.5 SFERR Test Case

5.3.2.5.1 SAIL Task SFERR010
This test case should generate errors. Task did not run to completion, did not increment the run counter, and killed itself.

From the CSTOL operator window, enter "start sferr010".
Load, Create, and Resume task by entering 12 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Test Engineer
Verify correct error generation

CURRENT ERROR
ERR TYPE = 253
LOCATION = 0x1C

Test Engineer
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Test Engineer

5.3.2.5.1.1 Test Results
Success

5.3.2.5.2 SAIL Task SFERR020
This test case should generate errors. Task should not run to completion, should not increment the run counter, and should kill itself.

From the CSTOL operator window, enter "start sferr020".
Load, Create, and Resume task by entering 12 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Test Engineer
Verify correct error generation

CURRENT ERROR
ERR TYPE = 253
LOCATION = 0xc2

Test Engineer
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Test Engineer

5.3.2.5.2.1 Test Results
Success

5.3.2.5.2.1 Test Results
Success
5.3.2.5.3 SAIL Task SFERR030

This test case should generate errors. Task should not run to completion, should not increment the run counter, and should die.

From the CSTOL operator window, enter "start sferr030".

Load, Create, and Resume task by entering 12 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer
Verify correct error generation

CURRENT ERROR          PREVIOUS ERROR
TASK NUM = 0 TASK NUM = 0
ERR TYPE = 230 ERR TYPE = 253
LOCATION = 0xf0 LOCATION = 0x28

Test Engineer

Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Test Engineer

5.3.2.5.4 SAIL Task SFERR040

This test case should generate errors. Task should not run to completion, should not increment the run counter, and should die.

From the CSTOL operator window, enter "start sferr040".

Load, Create, and Resume task by entering 12 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer
Verify correct error generation

CURRENT ERROR          PREVIOUS ERROR
TASK NUM = 0 TASK NUM = 0
ERR TYPE = 234 ERR TYPE = 253
LOCATION = 0x100 LOCATION = 0x2E

Test Engineer

Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Test Engineer

5.3.2.5.4.1 Test Results:

Success
5.3.2.5.5 SAIL Task SFERR050
This test case should generate errors. Task should not run to completion, should not increment the run counter, and should ERE-MM.

From the CSTOL operator window, enter "start sferr050".
Load, Create, and Resume task by entering 12 when prompted to enter menu choice. Verify procedure returns to main menu without error.

<table>
<thead>
<tr>
<th>CURRENT ERR</th>
<th>PREVIOUS ERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK NUM = 0</td>
<td>TASK NUM = 0</td>
</tr>
<tr>
<td>ERR TYPE = 231</td>
<td>ERR TYPE = 253</td>
</tr>
<tr>
<td>LOCATION = 0xE1</td>
<td>LOCATION = 0x34</td>
</tr>
</tbody>
</table>

Test Engineer
Verify correct error generation

Test Engineer
Erase the task by entering 7 when prompted to enter menu choice. Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.6.1 Test Results
Success

Test Conductor: JPR 2003-06-10

Procedural Redline:
The location of the current error was actually 0xe4. The test procedure will be updated to reflect this. This occurred when the task was recompiled with SAIL v2.5, which shifted the location of the ASM instruction. The procedure was not updated to reflect this.

5.3.2.6 SFUN Test Case

5.3.2.6.1 SAIL Task SFUN010
From the CSTOL operator window, enter "start sfun010".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice. Verify procedure returns to main menu without error.

Test Engineer
Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice. Enter 128 when asked to enter a SAIL block. Run data_read.

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x16

Test Engineer
Kill/Erase the task by entering 14 when prompted to enter menu choice. Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.
5.3.2.6.1 Test Results:

Test Conductor: JPR 2003-06-10

5.3.2.6.2 SAIL Task SRTL010

From the CSTOL operator window, enter “start srtl010”.

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.

Enter 128 when asked to enter a SAIL block.

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x0D.

Kill/Erase the task by entering 14 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

5.3.2.6.3 SAIL Task SRTL020

From the CSTOL operator window, enter “start srtl020”.

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify that the SAIL task is suspended.

Kill/Erase the task by entering 14 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

Clear the errors this task intentionally caused by sending the command:

@hir_ipu_faultclr

from the CSTOL operator window.

5.3.2.6.3.1 Test Results:

Test Conductor: JPR 2003-06-10
5.3.2.7 SIERR010 Test Case

5.3.2.7.1 SAIL Task SIERR010

From the CSTOL operator window, enter "start sierr010".

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer

Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.

Enter 128 when asked to enter a SAIL block.

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x06

Test Engineer

Kill/Erase the task by entering 14 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Test Engineer

5.3.2.7.1.1 Test Results:

Success

Test Conductor: JPR 2003-06-10

5.3.2.8 SMEM Test Case

5.3.2.8.1 SAIL Task SMEM010

From the CSTOL operator window, enter "start smem010".

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer

Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.

Enter 128 when asked to enter a SAIL block.

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x40

Test Engineer

Verify that in the SAIL task data memory block MemoryArray counts down from 0x12345678 for 15 counts.

Test Engineer

Set memory dump to the beginning of the private shared memory by entering 8 when prompted to enter menu choice.

Enter 0 when asked to enter a SAIL block.

Run data_read

Verify that the SAIL shared memory block starts with 0x12345678 and counts down for 15 counts.

Test Engineer

Set memory dump to the beginning of the common shared memory by entering 8 when prompted to enter menu choice.

Test Engineer

5.3.2.8.1.1 Test Results:

Success

Test Conductor: JPR 2003-06-10
5.3.2.8 SNST Test Case

5.3.2.8.1 SAIL Task SNST010
From the CSTOL operator window, enter "start snst010".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.
Enter 128 when asked to enter a SAIL block.
Run data_read
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x3E
Kill/Erase the task by entering 14 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.9 SOP Test Case

5.3.2.10.1 SAIL Task SOP010
From the CSTOL operator window, enter "start sop010".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.
Enter 128 when asked to enter a SAIL block.
Run data_read
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x5E
For memory dump to the beginning of the task data memory by entering 8 when prompted to main menu choice.

Test Engineer
Set memory dump to the beginning of the task data memory by entering 8 when prompted to main menu choice.

Test Engineer
Verify that in the SAIL task data memory block intArray counts from 0xf4 to 0x44.

Test Engineer
Enter 129 when asked to enter a SAIL block.

Test Engineer
Run data_read.

Test Engineer
Verify that in the SAIL task data memory block intArray counts from 0xf4 to 0x44.

Test Engineer
Kill/Erase the task by entering 14 when prompted to enter menu choice.

Test Engineer
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.10.1 Test Results
Date
Test Conductor: JPR 2003-06-10
Success
Test Conductor: JPR 2003-06-10

From the CSTOL operator window, enter "start sop020".

Test Engineer
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Test Engineer
Verify procedure returns to main menu without error.

Test Engineer
Set memory dump to the beginning of the task data memory by entering 8 when prompted to main menu choice.

Test Engineer
Enter 128 when asked to enter a SAIL block.

Test Engineer
Run data_read.

Test Engineer
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x1D.

Test Engineer
Kill/Erase the task by entering 14 when prompted to enter menu choice.

Test Engineer
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.10.2 Test Results
Date
Test Conductor: JPR 2003-06-10
Success
Test Conductor: JPR 2003-06-10

From the CSTOL operator window, enter "start sovr010".

Test Engineer
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Test Engineer
Verify procedure returns to main menu without error.

Test Engineer
Set memory dump to the beginning of the task data memory by entering 8 when prompted to main menu choice.

Test Engineer
Enter 128 when asked to enter a SAIL block.

Test Engineer
Run data_read.

Test Engineer
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x1D.

Test Engineer
Kill/Erase the task by entering 14 when prompted to enter menu choice.

Test Engineer
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.11 Test Case
5.3.2.11.1 SAIL Task SOVR010
From the CSTOL operator window, enter "start sovr010".

Test Engineer
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Test Engineer
Verify procedure returns to main menu without error.

Test Engineer
Set memory dump to the beginning of the task data memory by entering 8 when prompted to main menu choice.

Test Engineer
Enter 128 when asked to enter a SAIL block.

Test Engineer
Run data_read.

Test Engineer
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x1D.

Test Engineer
Kill/Erase the task by entering 14 when prompted to enter menu choice.

Test Engineer
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.11.2.1 Test Results
Date
Test Conductor: JPR 2003-06-10
Success
Test Conductor: JPR 2003-06-10

5.3.2.11.1.1 SAIL Task SOVR010
From the CSTOL operator window, enter "start sovr010".

Test Engineer
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Test Engineer
Verify procedure returns to main menu without error.

Test Engineer
Set memory dump to the beginning of the task data memory by entering 8 when prompted to main menu choice.

Test Engineer
Enter 128 when asked to enter a SAIL block.

Test Engineer
Run data_read.

Test Engineer
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x1D.

Test Engineer
Kill/Erase the task by entering 14 when prompted to enter menu choice.

Test Engineer
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.
Run data_read
Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x12

Test Engineer

Kill/Erase the task by entering 14 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter main menu choice.

Test Engineer

Date

5.3.2.11.1 Test Results
Success
Test Conductor: JPR 2003-06-10

5.3.2.11 SST Test Case

5.3.2.11.1 SAIL Task SST010

From the CSTOL operator window, enter “start sst010”.
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Test Engineer

Date

5.3.2.11.1.1 Test Results
Success
Test Conductor: JPR 2003-06-10

5.3.2.12 SST Test Case

5.3.2.12.1 SAIL Task SDATA010

From the CSTOL operator window, enter “start sdata010”.
Turn on the SAIL test pattern by entering 15 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Test Engineer

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Test Engineer

Date

5.3.2.12.1.1 Test Results
Success
Test Conductor: JPR 2003-06-10

5.3.2.12.1.2 SAIL Task SDATA020

From the CSTOL operator window, enter “start sdata020”.
When the task starts, all the ENG WORDS are set to 0xFFFFFFFF. As each science data type is verified, the nibbles of the ENG WORDS will change to 0, or to 1 if there was an error.

Test Engineer

Leak, Check, Reformat, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Test Engineer

Date

5.3.2.12.1.2.1 Test Results
Success
Test Conductor: JPR 2003-06-10
Test Engineer

Verify in the SAIL display that ENG WORD 0 – 4 equals 0x00000000 and ENG WORD 5 – 7 equals 0xFFFF0000.

Test Engineer

Turn off the SAIL test pattern by entering 16 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer

Kill/Erase the task by entering 14 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

---

5.3.2.1.2.1 Test Results:
Success

Test Conductor: JPR 2003-06-10

---

5.3.2.13.1 STYPE Test Case

5.3.2.13.1.1 SAIL Task STYPE010

From the CSTOL operator window, enter “start stype010”.

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer

Set memory dump to the beginning of the task data memory by entering 8 when prompted to enter menu choice.

Enter 128 when asked to enter a SAIL block.

Run data_read

Verify that in the SAIL task data memory block intArray counts from 0x00 to 0x08.

Test Engineer

Kill/Erase the task by entering 14 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

---

5.3.2.13.1.1 Test Results:
Success

Test Conductor: JPR 2003-06-10

---

5.3.2.14 Test Case WAIT

5.3.2.14.1 SAIL Tasks WAIT010 – WAIT020

From the CSTOL operator window, enter “start wait010”.

Load, Create, and Resume task by entering 12 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer

Suspend task by entering 5 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

---

5.3.2.14 Test Case WAIT

5.3.2.14.1.1 SAIL Task WAIT010 – WAIT020

From the CSTOL operator window, enter “start wait010”.

Resume, Create, and Resume task by entering 12 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer

Suspend task by entering 5 when prompted to enter menu choice.

Verify procedure returns to main menu without error.
Resume task by entering 3 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then enter “X” to exit when prompted to enter menu choice.

Verify that task 0 and task 1 are now suspended, then enter “X” to exit when prompted to enter menu choice.

From the CSTOL operator window, enter “start wait020”.
Load, Create, and Resume task by entering 12 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then enter “X” to exit when prompted to enter menu choice.

From the CSTOL operator window, enter “start wait010”.
Verify that task 1 is suspended.

Kill/Erase the task by entering 14 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

From the CSTOL operator window, enter “start wait020”.
Kill/Erase the task 0 and task 1 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

From the CSTOL operator window, enter “start wait010”.
Load, Create, and Resume task by entering 12 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then enter “X” to exit when prompted to enter menu choice.

From the CSTOL operator window, enter “start stovr010”.
Load, Create, and Resume task by entering 12 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Verify that task 1 is suspended.
Test Engineer
Verify correct error generation
task = 1
Type = 238
Test Engineer
Verify that task 2 is suspended
Test Engineer
Verify that task 1 is running
Test Engineer
Halt SAIL by entering 11 when prompted to enter menu choice.
Restart SAIL by entering 10 when prompted to enter menu choice.
Enter "X" to exit when prompted to enter menu choice.

5.3.2.15.1 Test Results:
Success
Test Conductor: JPR 2003-06-10

5.3.2.16.1 SASM Test Case
5.3.2.16.1.1 SASM Test Case
From the CSTOL operator window, enter "start sasmlimit".
Load task by entering 1 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Halt SAIL by entering 11 when prompted to enter menu choice.
Initialize SAIL by entering 10 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then enter "X" to exit when prompted to enter menu choice.

5.3.2.16.1.1 Test Results:
Success
Test Conductor: JPR 2003-06-10

5.3.2.17 SPEC Test Case
5.3.2.17.1 SPEC Test Case
From the CSTOL operator window, enter "start sw_loc_117_3_1_1_01".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

5.3.2.17.1 Test Results:
Success
Test Conductor: JPR 2003-06-10

5.3.2.17 SPEC Test Case
5.3.2.17.1 SPEC Test Case
From the CSTOL operator window, enter "start sw_loc_117_3_1_1_01".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Test Conductor: JPR 2003-06-10

5.3.2.17 SPEC Test Case
5.3.2.17.1 SPEC Test Case
From the CSTOL operator window, enter "start sw_loc_117_3_1_1_01".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Test Conductor: JPR 2003-06-10
Verify procedure returns to main menu without error.

Test Engineer
Verify in SAIL display that task 0 is halted and the current error is 235.
Test Engineer
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that text diagnostic item is examined post test for final verification.

5.3.2.17.1.1 Test Results:
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17.2 SAIL Task sw_loc_117_3_1_1_02
From the CSTOL operator window, enter "start sw_loc_117_3_1_1_02".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Test Engineer
Verify in SAIL display that task 0 is halted and the current error is 239.
Test Engineer
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that text diagnostic item is examined post test for final verification.

5.3.2.17.2.1 Test Results:
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17.3 SAIL Task sw_loc_117_3_1_2_01
From the CSTOL operator window, enter "start sw_loc_117_3_1_2_01".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Test Engineer
Verify in SAIL display that task 0 is halted and the current error is 235.
Test Engineer
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that text diagnostic item is examined post test for final verification.

5.3.2.17.3.1 Test Results:
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17.4 SAIL Task sw_loc_117_3_1_3_01
From the CSTOL operator window, enter "start sw_loc_117_3_1_3_01".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Test Engineer
Verify in SAIL display that task 0 is halted and the current error is 235.
Test Engineer
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that text diagnostic item is examined post test for final verification.

5.3.2.17.4.1 Test Results:
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10
5.3.2.17.4 SAIL Task sw_loc_117_3_1_2_02

From the CSTOL operator window, enter “start sw_loc_117_3_1_2_02”.
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Verify in SAIL display that task 0 is halted and the current error is 235.
Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

Test Engineer

Date

Ensure that the text diagnostic item is examined post-test for final verification.

5.3.2.17.4.1 Test Results:

Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17.5 SAIL Task sw_loc_117_3_2_1_01

From the CSTOL operator window, enter “start sw_loc_117_3_2_1_01”.
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Verify in SAIL display that task 0 is halted and the current error is 235.
Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

Test Engineer

Date

Ensure that the text diagnostic item is examined post-test for final verification.

5.3.2.17.5.1 Test Results:

Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17.6 SAIL Task sw_loc_117_3_3_1_2_1_01

From the CSTOL operator window, enter “start sw_loc_117_3_3_1_2_1_01”.
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Verify in SAIL display that task 0 is halted and the current error is 235.
Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

Test Engineer

Date

Ensure that the text diagnostic item is examined post-test for final verification.

5.3.2.17.6.1 Test Results:

Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10
Verify in SAIL display that task 0 is halted and the current error is 235.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that the test diagnostics form is examined post-test in final verification.

5.3.2.17.6 Test Results:

Test Successful

Analysis Successful

Test Conductor: JPR 2003-06-10

5.3.2.17.7 SAIL Task sw_loc_117_3_3_1_2_1_02

From the CSTOL operator window, enter "start sw_loc_117_3_3_1_2_1_02".

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Verify in SAIL display that task 0 is halted and the current error is 235.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that the test diagnostics form is examined post-test in final verification.

5.3.2.17.7.1 Test Results:

Test Successful

Analysis Successful

Test Conductor: JPR 2003-06-10

5.3.2.17.8 SAIL Task sw_loc_117_3_3_1_2_2_01

From the CSTOL operator window, enter "start sw_loc_117_3_3_1_2_2_01".

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Verify in SAIL display that task 0 is halted and the current error is 235.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that the test diagnostics form is examined post-test in final verification.

5.3.2.17.8.1 Test Results:

Test Successful

Analysis Successful

Test Conductor: JPR 2003-06-10

5.3.2.17.9 SAIL Task sw_loc_117_3_3_1_2_2_01

From the CSTOL operator window, enter "start sw_loc_117_3_3_1_2_2_01".

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that the test diagnostics form is examined post-test in final verification.

5.3.2.17.9.1 Test Results:

Test Successful

Analysis Successful

Test Conductor: JPR 2003-06-10

5.3.2.17.10 Test Results:

Test Successful
5.3.2.17.9. SAIL Task sw_loc_117_3_3_1_2_3_01

From the CSTOL operator window, enter “start sw_loc_117_3_3_1_2_3_01”. Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer
Verify in SAIL display that task 0 is halted and the current error is 235.

Test Engineer
Ensure that the text diagnostic item is examined post-test for final verification.

5.3.2.17.9.1 Test Results
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17.10. SAIL Task sw_loc_117_3_3_1_2_4_01

From the CSTOL operator window, enter “start sw_loc_117_3_3_1_2_4_01”. Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer
Verify in SAIL display that task 0 is halted and the current error is 235.

Test Engineer
Ensure that the text diagnostic item is examined post-test for final verification.

5.3.2.17.10.1 Test Results
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17.11. SAIL Task sw_loc_117_3_3_1_2_5_01

From the CSTOL operator window, enter “start sw_loc_117_3_3_1_2_5_01”. Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer
Verify in SAIL display that task 0 is halted and the current error is 235.

Test Engineer
Ensure that the text diagnostic item is examined post-test for final verification.

5.3.2.17.11.1 Test Results
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10
5.3.2.17.1 Test Results

Test Successful

Analysis Successful

Test Conductor: JPR 2003-06-10

5.3.2.17.12 SAIL Task sw_loc_117_3_3_1_3_01

From the CSTOL operator window, enter "start sw_loc_117_3_3_1_3_01".

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer

Verify in SAIL display that task 0 is halted and the current error is 235.

Test Engineer

Erase the task by entering 7 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Test Engineer

Date

Ensure that the text diagnostic item is examined post-test for final verification.

5.3.2.17.12.1 Test Results

Test Successful

Analysis Successful

Test Conductor: JPR 2003-06-10

5.3.2.17.13 SAIL Task sw_loc_117_3_3_1_4_01

From the CSTOL operator window, enter "start sw_loc_117_3_3_1_4_01".

Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.

Verify procedure returns to main menu without error.

Test Engineer

Verify in SAIL display that task 0 is halted and the current error is 235.

Test Engineer

Erase the task by entering 7 when prompted to enter menu choice.

Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Test Engineer

Date

Ensure that the text diagnostic item is examined post-test for final verification.

5.3.2.17.13.1 Test Results

Test Successful

Analysis Successful

Test Conductor: JPR 2003-06-10
5.3.2.17.14 SAIL Task sw_loc_117_3_3_2_1_01
From the CSTOL operator window, enter "start sw_loc_117_3_3_2_1_01". Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify task by entering "X" when prompted to enter menu choice.
Verify in SAIL display that task 0 is halted and the current error is 235.
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.17.14.1 Test Results
Test Conductor: JPR 2003-06-10

5.3.2.17.15 SAIL Task sw_loc_117_3_3_2_2_01
From the CSTOL operator window, enter "start sw_loc_117_3_3_2_2_01". Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify task by entering "X" when prompted to enter menu choice.
Verify in SAIL display that task 0 is halted and the current error is 235.
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

5.3.2.17.15.1 Test Results
Test Conductor: JPR 2003-06-10

5.3.2.17.16 SAIL Task sw_loc_117_3_4_1_1_01
From the CSTOL operator window, enter "start sw_loc_117_3_4_1_1_01". Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify task by entering "X" when prompted to enter menu choice.
Verify in SAIL display that task 0 is halted and the current error is 235.
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.
5.3.2.17 Task sw_loc_117_3_4_1_2_01
From the CSTOL operator window, enter “start sw_loc_117_3_4_1_2_01.”
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Verify in SAIL display that task 0 is halted and the current error is 235.
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

5.3.2.17.1 Test Results
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17 Task sw_loc_117_3_5_2_01
From the CSTOL operator window, enter “start sw_loc_117_3_5_2_01.”
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.
Verify in SAIL display that task 0 is halted and the current error is 235.
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering “X” when prompted to enter menu choice.

5.3.2.17.1.1 Test Results
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17 Task sw_loc_117_6_01
From the CSTOL operator window, enter “start sw_loc_117_6_01.”
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Verify in SAIL display that task 0 is suspended.
Resume the task by entering 3 when prompted to enter menu choice.

Verify in SAIL display that task 0 is halted and the current error is 235.
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that the test diagnostics item is examined post-test for final verification.

5.3.1.17.19.1 Test Results
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.2.17.20 SAIL Task sw_loc_117_6_02
From the CSTOL operator window, enter "start sw_loc_117_6_02".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Verify in SAIL display that task 0 is halted and the current error is 235.
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.

Ensure that the test diagnostics item is examined post-test for final verification.

5.3.1.17.20.1 Test Results
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10

5.3.1.17.21 SAIL Task sw_loc_117_6_03
From the CSTOL operator window, enter "start sw_loc_117_6_03".
Load, Create, Resume, and Monitor task to completion by entering 13 when prompted to enter menu choice.
Verify procedure returns to main menu without error.

Verify in SAIL display that task 0 is halted and the current error is 235.
Erase the task by entering 7 when prompted to enter menu choice.
Verify procedure returns to main menu without error, then exit by entering "X" when prompted to enter menu choice.
Ensure that the test diagnostic item is examined post test for final verification.

5.3.2.17.21.1 Test Results
Test Successful
Analysis Successful
Test Conductor: JPR 2003-06-10
6.1 Code Inspection Session 1 (2002-06-22)

6.1.1 IR_SAI_1000 Dynamic Memory Inspection (SAIL180)
In order for the SAIL language to support dynamic memory allocation, the language would need to be specifically designed with this support in mind.

The SAIL language was not designed to support dynamic memory allocation.

6.1.2 IR_SAI_1010 Pointer Access Prohibited Inspection (SAIL270)
In order for the SAIL language to support pointer access, the language would need to be specifically designed with this support in mind.

The SAIL language was not designed to support pointer access, prohibiting such access by not providing it.

6.1.3 IR_SAI_1020 Non-Optimizing Compiler Inspection (SAIL270)
In order for the SAIL compiler to support optimization, the compiler would need to be specifically designed with this support in mind.

The SAIL compiler was not designed to perform optimization.

6.1.4 IR_SAI_1030 Memory Management Inspection (SAIL560, SAIL570)
The SAIL executive is not capable of allocating memory at any time other than task creation. This is ensured because the functions that assign memory to a task can only be called in response to the HIR_SAI_CREATE command, indicating a task creation.

The design of the SAIL system prevents a task from accessing memory at any time other than at task creation. There are no pointers, and all memory assignments for local variables are handled by the assembler and the interpreter. There is no way to write a task that can access memory outside of its allocated task memory.

6.1.5 IR_SAI_1040 Task Starting (SAIL600)
The functions to start a SAIL task are accessible only via the command HIR_SAI_RESUME. This command cannot be issued by a SAIL task, and is allowed only from the ground.

6.1.6 IR_SAI_1050 Task and Data (SAIL610)
The SAIL system provides access to shared memory. The contents of this shared memory are accessible to all tasks, and can be changed via ground uploads while any task is running. Any mutable data tables required by a task may be loaded into the shared memory. A task is assigned its own section of memory, and this section is accessible only by the task.

7 VERIFICATION CROSS-REFERENCE MATRIX

The traceability of the SAIL requirements from the parent specifications is provided in this table which is taken from the SAIL Requirements Document (SW-HIR-147).

See "Table 1-1 Verification Events" in this document for more detail on the entries listed in the Verification Event column.
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Test Method</th>
<th>Test Case</th>
<th>Software Acceptance Test</th>
<th>Software Acceptance Regression Test</th>
<th>Request Off</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1 SAIL controls mechanisms</td>
<td>Executive T SCOMS SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-05-02)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.2 SAIL writes to housekeeping data stream</td>
<td>Executive T SST SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-05-02)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.3 SAIL writes to diagnostic ports</td>
<td>Executive T ALL SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-05-02)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.4 SAIL reads all data buffers</td>
<td>Executive T SST SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-05-02)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.5 SAIL writes to diagnostic ports</td>
<td>Executive T ALL SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-05-02)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1.6 SAIL accesses spacecraft real-time info</td>
<td>Executive T SST SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-05-02)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Source</td>
<td>Status</td>
<td>Start Date</td>
<td>End Date</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>--------</td>
<td>--------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>4.1.7</td>
<td>Mnemonic access to all input to IPS Compiler</td>
<td>SST SQE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test 1 (2002-06)</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Simple language Compiler Demonstration</td>
<td>N/A SQE</td>
<td>Accepted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.2.2</td>
<td>Extensive comments Compiler</td>
<td>SCMP SQE</td>
<td>Accepted</td>
<td>2002-06-17</td>
<td></td>
</tr>
<tr>
<td>4.2.3</td>
<td>SAIL compiler and assembler language</td>
<td>DELETED</td>
<td>Not Compliant</td>
<td>D016</td>
<td></td>
</tr>
<tr>
<td>4.2.4</td>
<td>Non-optimizing compiler</td>
<td>IR_SAI_1020 SQE</td>
<td>Accepted</td>
<td>2002-06-22</td>
<td></td>
</tr>
<tr>
<td>4.2.5</td>
<td>Telemetry mnemonics in re-definable table Compiler</td>
<td>SST SQE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test 1 (2002-06)</td>
</tr>
<tr>
<td>4.2.6</td>
<td>All variables declared before use Compiler</td>
<td>SCERR SQE</td>
<td>Accepted</td>
<td>2002-06-17</td>
<td></td>
</tr>
<tr>
<td>4.2.7</td>
<td>No dynamic memory allocation Compiler</td>
<td>T</td>
<td>Accepted</td>
<td>2002-06-17</td>
<td></td>
</tr>
<tr>
<td>4.2.8</td>
<td>No multiple assignments Compiler</td>
<td>SCERR</td>
<td>Accepted</td>
<td>2002-06-17</td>
<td></td>
</tr>
<tr>
<td>4.2.9.1</td>
<td>Byte data type Compiler, Interpreter</td>
<td>STYPE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test 1 (2002-06)</td>
</tr>
<tr>
<td>4.2.9.2</td>
<td>Unsigned data type Compiler, Interpreter</td>
<td>STYPE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test 1 (2002-06)</td>
</tr>
<tr>
<td>4.2.9.3</td>
<td>Integer data type Compiler, Interpreter</td>
<td>STYPE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test 1 (2002-06)</td>
</tr>
<tr>
<td>4.2.9.4</td>
<td>Integer arithmetic and products Compiler, Interpreter</td>
<td>T</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test 1 (2002-06)</td>
</tr>
<tr>
<td>4.2.9.5</td>
<td>Integer arithmetic and products Compiler, Interpreter</td>
<td>T</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test 1 (2002-06)</td>
</tr>
<tr>
<td>4.2.9.6</td>
<td>Integer arithmetic and products Compiler, Interpreter</td>
<td>T</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test 1 (2002-06)</td>
</tr>
<tr>
<td>Code/Title</td>
<td>Test ID</td>
<td>Category</td>
<td>Description</td>
<td>Status</td>
<td>Notes</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>----------</td>
<td>-------------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>4.2.9.4 Float data type</td>
<td>Compiler, Interpreter</td>
<td>T STYPE</td>
<td>SQE Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.9.5 Logical data type</td>
<td>Compiler, Interpreter</td>
<td>T STYPE</td>
<td>SQE Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.9.6 Array type</td>
<td>Compiler, Interpreter</td>
<td>T SARR</td>
<td>SQE Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.9.7 Telemetry Access Methods</td>
<td>Compiler, Executive, Interpreter</td>
<td>T SST</td>
<td>SQE Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.10.1 Single IF…ELSE structure</td>
<td>Compiler</td>
<td>T SCMP</td>
<td>SQE Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.10.2 Single loop structure</td>
<td>Compiler</td>
<td>T SCMP</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.10.3 Jump to label</td>
<td>Compiler</td>
<td>T SCMP</td>
<td>SQE Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.10.4 Subprograms implemented</td>
<td>Compiler</td>
<td>T SFUN</td>
<td>SQE Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Subsection</td>
<td>Description</td>
<td>Status</td>
<td>Date</td>
<td>Test Run</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>-------------</td>
<td>--------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>4.2.10.4</td>
<td>Subprogram parameter conformance</td>
<td>Compiler T SFUN SQE</td>
<td>Accepted</td>
<td>2002-06-17</td>
<td>Software Acceptance Regression Test (2002-04-10 - 2002-04-12)</td>
</tr>
<tr>
<td>4.2.11.1</td>
<td>Assignment operator</td>
<td>Interpreter T SOP SQE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test (2002-04-10 - 2002-04-12)</td>
</tr>
<tr>
<td>4.2.11.2</td>
<td>Arithmetic operators</td>
<td>Interpreter, Compiler T SOP SQE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test (2002-04-10 - 2002-04-12)</td>
</tr>
<tr>
<td>4.2.11.3</td>
<td>Bit manipulation</td>
<td>Interpreter, Compiler T SOP SQE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test (2002-04-10 - 2002-04-12)</td>
</tr>
<tr>
<td>4.2.11.4</td>
<td>Logical combination</td>
<td>Interpreter, Compiler T SOP SQE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test (2002-04-10 - 2002-04-12)</td>
</tr>
<tr>
<td>4.2.11.5</td>
<td>Relational operators</td>
<td>Interpreter, Compiler T SOP SQE</td>
<td>Accepted</td>
<td>2002-04-10 - 2002-04-12</td>
<td>Software Acceptance Regression Test (2002-04-10 - 2002-04-12)</td>
</tr>
<tr>
<td>4.2.12</td>
<td>Math functions</td>
<td>Interpreter T SFUN SQE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1.1.1</td>
<td>Subroutine execution</td>
<td>Compiler</td>
<td>1.5</td>
<td>100</td>
<td>Accepted</td>
</tr>
<tr>
<td>3.1.1.1</td>
<td>Logical combination</td>
<td>Compiler</td>
<td>2</td>
<td>100</td>
<td>Accepted</td>
</tr>
<tr>
<td>4.2.11.5</td>
<td>Relational operators</td>
<td>Interpreter, Compiler</td>
<td>T SOP SQE</td>
<td>Accepted</td>
<td>Software Acceptance Regression Test (2002-04-10 - 2002-04-12)</td>
</tr>
<tr>
<td>Requirement</td>
<td>Vendor</td>
<td>Test</td>
<td>Acceptance</td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------</td>
<td>------</td>
<td>------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>4.2.13.1 Defined operator precedence</td>
<td>PostgreSQL</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Regression Test 1 (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.13.2 Defined expression evaluation Compiler T SOP SQE</td>
<td>IBM</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.13.3 Brackets to override operator precedence</td>
<td>Oracle</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.13.4 Evaluation of mixed type expressions Compiler T STYPE SQE</td>
<td>Microsoft</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.13.5 Treat underflow as zero Interpreter T SOVR SQE</td>
<td>Oracle</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.13.6 Flag overflow Interpreter T SOVR SQE</td>
<td>IBM</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.14 SAIL High Level Specification Document Demonstration</td>
<td>Oracle</td>
<td>N/A</td>
<td>Accepted</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>4.2.15 SAIL task ascertain millisecond left in MaF Interpreter T SST SQE</td>
<td>IBM</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.16 SAIL task ascertain current time Interpreter T SST SQE</td>
<td>Oracle</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.17 SAIL task ascertain elapsed time Interpreter T SST SQE</td>
<td>IBM</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.18 SAIL task ascertain millisecond left in date Interpreter T SST SQE</td>
<td>Oracle</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.2.19 SAIL task ascertain current date Interpreter T SST SQE</td>
<td>IBM</td>
<td>1</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.3.1</td>
<td>Up to 16 tasks</td>
<td>Executive</td>
<td>T</td>
<td>SEXEC</td>
<td>SQE</td>
</tr>
<tr>
<td>4.3.2</td>
<td>16 tasks/clock period</td>
<td>Executive</td>
<td>T</td>
<td>SEXEC</td>
<td>SQE</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Task over-run</td>
<td>Executive</td>
<td>T</td>
<td>STOVR</td>
<td>SQE</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Inter-task communication</td>
<td>Executive, Compiler</td>
<td>T</td>
<td>SMEM</td>
<td>SQE</td>
</tr>
<tr>
<td>4.4.1</td>
<td>SAIL code &lt; 65536 bytes</td>
<td>Assembler Executive</td>
<td>T</td>
<td>SASM</td>
<td>SQE</td>
</tr>
<tr>
<td>4.4.2</td>
<td>SAIL variables &lt; 65536 bytes</td>
<td>Compiler, Executive</td>
<td>T</td>
<td>SARR</td>
<td>SQE</td>
</tr>
<tr>
<td>4.4.3</td>
<td>Memory allocation</td>
<td>Executive, Compiler</td>
<td>T</td>
<td>IR_SAI_1030</td>
<td>SQE</td>
</tr>
<tr>
<td>4.4.4</td>
<td>Memory protection</td>
<td>Executive, Inspection</td>
<td>T</td>
<td>SIERR</td>
<td>IR_SAI_1030</td>
</tr>
<tr>
<td>4.4.5</td>
<td>Spare task storage</td>
<td>Executive</td>
<td>T</td>
<td>SEXEC</td>
<td>SQE</td>
</tr>
<tr>
<td>Articulation</td>
<td>Test Cases</td>
<td>Test Case Name</td>
<td>Test Case ID</td>
<td>Test Case Description</td>
<td>Test Case Input</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>----------------</td>
<td>--------------</td>
<td>-----------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>4.4.6</td>
<td></td>
<td>SAIL reads IPU memory Executive T SMEM SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.5.1</td>
<td></td>
<td>SAIL task info dump IPU T TC_IPU_TL M_250 from TR-LOC-955A</td>
<td>Accepted</td>
<td>All Flight Test Cases SQE</td>
<td></td>
</tr>
<tr>
<td>4.5.2</td>
<td></td>
<td>Starting and stopping tasks from ground Executive, IPU CSCI T ALL SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.5.3a</td>
<td></td>
<td>Only start a task from ground Executive Inspection IR_SAI_1040 SQE</td>
<td>Accepted</td>
<td>Code Inspection Session 1 (2002-06-17);</td>
<td></td>
</tr>
<tr>
<td>4.5.3b</td>
<td></td>
<td>Suspended or Killed from ground Executive T WAIT SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.5.3c</td>
<td></td>
<td>Suspending a task from SAIL Executive T WAIT SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.5.3d</td>
<td></td>
<td>Killing a task from SAIL Executive T SFERR SQE</td>
<td>Accepted</td>
<td>Software Acceptance Test (2002-04-10 - 2002-04-12), Software Acceptance Regression Test 1 (2002-06-17)</td>
<td></td>
</tr>
<tr>
<td>4.5.5</td>
<td></td>
<td>Complete tasks and data tables Executive Test, Inspection WAIT SMEM IR_SAI_1050 SQE</td>
<td>Accepted</td>
<td>Code Inspection Session 1 (2002-06-17)</td>
<td></td>
</tr>
</tbody>
</table>
8 LIST OF ACRONYMS

ASCII  American Standard Code for Information Interchange
ATC  Advanced Technology Center
CSC  Computer Software Component
CSCI  Computer Software Configuration Item
ESD  ElectroStatic Discharge
FTP  File Transfer Protocol
HIRDLS  High Resolution Dynamics Limb Sounder
I&T  Integration and Test
IEGSE  Instrument Electrical Ground Support Equipment
IPU  Instrument Processor Unit
MaF  Major Frame
PC  Personal Computer
PSA  Processor Sub Assembly
RCS  Revision Control System
SAIL  Science Algorithm Implementation Language
STOL  System Test and Operation Language
TEU  Telescope Electronics Unit
VCRM  Verification Cross Reference Matrix

9 Appendix

9.1 Software Regression Test 1 (2002-06-17)

9.1.1 SAIL_SWACC_2002_06_17.txt

F:\SAIL\SAIL_CSCI\SAIL_v2.4\SAILtasks\SAIL_SWACC>CALL SC /R /V /E SARR010

SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp$

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\n
The file SARR010.sai does not exist.

Attempting to retrieve from RCS ...
The file SARR011.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\SARR011.sai
Working file: SARR011.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------

--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------

The file SARR012.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\SARR012.sai
Working file: SARR012.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------

--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------

86
The file SARR013.sai does not exist.

Attempting to retrieve from RCS ...

RCS file: RCS\SARR013.sai
Working file: SARR013.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

============================================================================
----------------- First Pass Errors/Warnings -----------------
----------------- Second Pass Errors/Warnings -----------------
----------------- Assembler Pass Errors/Warnings ----------------

The file SARR014.sai does not exist.

Attempting to retrieve from RCS ...

RCS file: RCS\SARR014.sai
Working file: SARR014.sai
head: 1.1
branch:
locks: strict
The file SARR020.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS/SARR020.sai
Working file: SARR020.sai
head: 1.1
branch: 
locks: strict
access list: 
symbolic names: kv
keyword substitution: kv
total revisions: 1

The file SARR030.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS/SARR030.sai
Working file: SARR030.sai
head: 1.1
branch: 
locks: strict
access list: 
symbolic names: kv
keyword substitution: kv
total revisions: 1
The file SARR030.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS\SARR030.sai
Working file: SARR030.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================
--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------

SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///////// COMPILING TASK SCERR010 /////////

The file SCERR010.sai does not exist. Attempting to retrieve from RCS....

RCS file: RCS\SCERR010.sai
Working file: SCERR010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================
--------------- First Pass Errors/Warnings -----------
Error: An invalid preprocessor directive was found.
Error occured in file SCERR010.sai, line 16, column 26
Error: This include directive was improperly formatted.
Error occured in file SCERR010.sai, line 17, column 26

//\\\\\\\\ COMPILING TASK SCERR010 //\\\\\\\\

Error: An invalid preprocessor directive was found.
Error: This include directive was improperly formatted.
Error: A bad special character was encountered.
Error occured in file SCERR010.sai, line 18, column 2
Error: Only source files with a "*.sai" extension can be processed.
Error occured in file xxxxxxxxx.h
Error: An error occurred while trying to include a file.
Error occured in file SCERR010.sai, line 21, column 1
Error: The file unexpectedly ended.
Error occured in file SCERR010.sai, line 21, column 2
Error: The sort of the file was incorrect, but the comment was not terminated.
Error occured in file SCERR010.sai, line 21, column 2
Error: An invalid floating-point number was encountered.
Error occured in file SCERR010.sai, line 23, column 24
Error: This floating-point number has no sign for its exponent.
Error occured in file SCERR010.sai, line 23, column 24
Error: This floating-point number has no sign for its exponent.
Error occured in file SCERR010.sai, line 23, column 24
Error: The exponent for this floating-point number is too large.
Error occured in file SCERR010.sai, line 24, column 39
Error: This floating-point number has no sign for its exponent.
Error occured in file SCERR010.sai, line 24, column 39
Error: An exponent was expected for this floating-point number.
Error occured in file SCERR010.sai, line 24, column 39
Error: The exponent for this floating-point number is too large.
Error occured in file SCERR010.sai, line 25, column 42
Error: This floating-point number has no sign for its exponent.
Error occured in file SCERR010.sai, line 25, column 42
Error: An exponent was expected for this floating-point number.
Error occured in file SCERR010.sai, line 25, column 42
Error: The exponent for this floating-point number is too large.
Error occured in file SCERR010.sai, line 26, column 46
Error: This floating-point number is too long.
Error occured in file SCERR010.sai, line 27, column 55
Error: This number has created overflow.
Error occured in file SCERR010.sai, line 28, column 31
Error: This floating-point number has created overflow.
Error occured in file SCERR010.sai, line 29, column 31
Fatal Error: A symbol has violated the symbol size limitation.
Fatal Error occured in file SCERR010.sai, line 30, column 69
Fatal Error(s) encountered during first pass, exiting
SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

////////////// COMPILING TASK SCERR040 //////////////

The file SCERR040.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\SCERR040.sai
Working file: SCERR040.sai
head: 1.2
branch:
Error: The type of the reference parameter badref must match the exact type of the corresponding reference argument.

Error: You cannot pass Immediates or Expressions by reference.

Error: In an expression, the token [] can't be placed after the value intarray.

Error: While passing baddimension to functionnormalarray, the dimension(s) of array did not match the dimension(s) of array as defined in the prototype.

Warning: The function functionwithreturn has a non-void return type, but it does not return a value.
Warning: Line: 111 Column: 5 File: SCERR040.sai

Warning: The return value from function functionwithreturn was ignored.
Warning: Line: 117 Column: 20 File: SCERR040.sai

Error: An incorrect number of parameters were passed to functionparameter.

Error: A goto cannot jump to the label unplaced because it has not been placed.
Error: Line: 135 Column: 2 File: SCERR040.sai

Error: The label alreadyplaced has already been placed.
Error: Line: 143 Column: 2 File: SCERR040.sai

Error: The identifier undefined cannot be used until it is defined.

Error: The identifier name duplicate has already been used.

Error: A non-zero, non-negative constant expression result was expected to dimension an array, but the token -1 was found instead.
Error: Line: 164 Column: 18 File: SCERR040.sai

Error: This expression is too complex.

Error: This expression is missing parentheses.
Error: Line: 180 Column: 20 File: SCERR040.sai

Error: Only functions may be called. An attempt was made to call the nonfunction tokernormalfunction.
Error: You cannot use the non-RTL function functioncall inside error while inside the body of the error function.

Error: The function functioncall has a void return type, but it returns a value.

Error: The function functioncall cannot be defined until it has been prototyped.

Error: The function functioncall has an empty return type, but it returns a value.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock01 is in an inner block relative to the jumping goto statement.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock02 is in an inner block relative to the jumping goto statement.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock03 is in an inner block relative to the jumping goto statement.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock04 is in an inner block relative to the jumping goto statement.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock05 is in an inner block relative to the jumping goto statement.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock06 is in an inner block relative to the jumping goto statement.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock07 is in an inner block relative to the jumping goto statement.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock08 is in an inner block relative to the jumping goto statement.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock09 is in an inner block relative to the jumping goto statement.

Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock10 is in an inner block relative to the jumping goto statement.

Error: The function functioncall must have a non-void return type to be used in an expression.
Error: The function `whoami` must be called.

Error: In an expression, the token `=` can't be placed in assignment.

Error: You must define a main function.

Error: The function main has been prototyped, but not defined.

Error: The file `SCERR050.sai` does not exist.

Attempting to retrieve from RCS ...

RCS file: RCS/SCERR050.sai
Working file: SCERR050.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================
The file SCERR060.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS/SCERR060.sai
Working file: SCERR060.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
total revisions: 1

--------------- First Pass Errors/Warnings ------------
Error: The function functionrecursion cannot call itself, directly or indirectly.
Error: Line: 37 Column: 6 File: SCERR060.sai

--------------- Second Pass Errors/Warnings ----------

SAIL Compile Utility, Version: $Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory: f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file SCGOTO010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS/SCGOTO010.sai
Working file: SCGOTO010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
total revisions: 1

--------------- First Pass Errors/Warnings ------------

--------------- Second Pass Errors/Warnings ----------

94
--------------- Assembler Pass Errors/Warnings ------------

SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

COMPILING TASK SCMP010

The file SCMP010.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\SCMP010.sai
Working file: SCMP010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================

--------------- First Pass Errors/Warnings ------------

SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

COMPILING TASK SCMP020

The file SCMP020.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\SCMP020.sai
Working file: SCMP020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================
The file SCOMS010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS010.sai
Working file: SCOMS010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS020.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS020.sai
Working file: SCOMS020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS010.sai
Working file: SCOMS010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS020.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS020.sai
Working file: SCOMS020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS010.sai
Working file: SCOMS010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS020.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS020.sai
Working file: SCOMS020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS010.sai
Working file: SCOMS010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS020.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS020.sai
Working file: SCOMS020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS010.sai
Working file: SCOMS010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS020.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS020.sai
Working file: SCOMS020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS010.sai
Working file: SCOMS010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS020.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS020.sai
Working file: SCOMS020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS010.sai
Working file: SCOMS010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS020.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS020.sai
Working file: SCOMS020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS010.sai
Working file: SCOMS010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS020.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS020.sai
Working file: SCOMS020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
============================================================================= 

The file SCOMS010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SCOMS010.sai
Working file: SCOMS010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
The file SDATA010.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\SDATA010.sai
Working file: SDATA010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
The file SEXEC000 does not exist.
Attempting to retrieve from RCS...

RCS file: RCS/SEXEC000.sai
Working file: SEXEC000.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ------------
--------------- Assembler Pass Errors/Warnings -------

--------------- First Pass Errors/Warnings ------------

--------------- Second Pass Errors/Warnings ------------

--------------- Assembler Pass Errors/Warnings -------

The file SEXEC001 does not exist.
Attempting to retrieve from RCS...

RCS file: RCS/SEXEC001.sai
Working file: SEXEC001.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

--------------- First Pass Errors/Warnings ------------

--------------- Second Pass Errors/Warnings ------------

--------------- Assembler Pass Errors/Warnings -------

98
The file SEXEC002.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SEXEC002.sai
Working file: SEXEC002.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

The file SEXEC003.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SEXEC003.sai
Working file: SEXEC003.sai
head: 1.1
The file SEXEC004.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SEXEC004.sai
Working file: SEXEC004.sai
Head: 1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

The file SEXEC004.sai does not exist.

SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)
The file SEXEC005.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS/SEXEC005.sai
Working file: SEXEC005.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

First Pass Errors/Warnings
Second Pass Errors/Warnings
Assembler Errors/Warnings

SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file SEXEC006.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS/SEXEC006.sai
Working file: SEXEC006.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

First Pass Errors/Warnings
Second Pass Errors/Warnings
Assembler Errors/Warnings

101
The file SEXEC007.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS/SEXEC007.sai
Working file: SEXEC007.sai
head: 1.1
branch:
locks: strict
access list:

The file SEXEC008.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS/SEXEC008.sai
Working file: SEXEC008.sai
head: 1.1
branch:
locks: strict
access list:
The file SEXEC009.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS/SEXEC009.sai
Working file: SEXEC009.sai
head: 1.1
branch: 
locks: strict
access list:
symbolic names: 
keyword substitution: kv
total revisions: 1

The file SEXEC010.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS/SEXEC010.sai
Working file: SEXEC010.sai
head: 1.1
branch: 
locks: strict
access list:
symbolic names: 
keyword substitution: kv
total revisions: 1
The file SEXEC010.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\SEXEC010.sai
Working file: SEXEC010.sai
head: 1.1
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv
total revisions: 1

-------------- First Pass Errors/Warnings ------------
-------------- Second Pass Errors/Warnings -----------
-------------- Assembler Pass Errors/Warnings ------

The file SEXEC011.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\SEXEC011.sai
Working file: SEXEC011.sai
head: 1.1
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv
total revisions: 1

-------------- First Pass Errors/Warnings ------------
-------------- Second Pass Errors/Warnings -----------
-------------- Assembler Pass Errors/Warnings ------

/\\/\\/\\/\\/ COMPILING TASK SEXEC011 /\\/\\/\\/\\/

The file SEXEC011.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\SEXEC011.sai
Working file: SEXEC011.sai
head: 1.1
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv
total revisions: 1

-------------- First Pass Errors/Warnings ------------
-------------- Second Pass Errors/Warnings -----------
-------------- Assembler Pass Errors/Warnings ------
The file SEXEC012.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SEXEC012.sai
Working file: SEXEC012.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------

The file SEXEC013.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SEXEC013.sai
Working file: SEXEC013.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
The file SEXEC014.sai does not exist. Attempting to retrieve from RCS ...

The file SEXEC015.sai does not exist. Attempting to retrieve from RCS ...

188
The file SEXEC016.sai does not exist.
Attempting to retrieve from RCS...

---

\--- First Pass Errors/Warnings ---
\--- Second Pass Errors/Warnings ----
\--- Assembler Pass Errors/Warnings ---

---

The file SEXEC016.sai does not exist.
Attempting to retrieve from RCS...

---

\--- First Pass Errors/Warnings ---
\--- Second Pass Errors/Warnings ----
\--- Assembler Pass Errors/Warnings ---

---

The file SEXEC016.sai does not exist.
Attempting to retrieve from RCS...

---

\--- First Pass Errors/Warnings ---
\--- Second Pass Errors/Warnings ----
\--- Assembler Pass Errors/Warnings ---

---
REM COMPILING TASK SFERR010

The file SFERR010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SFERR010.sai
Working file: SFERR010.sai
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2

=============================================================================  
--------------- First Pass Errors/Warnings  -----------

--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
The file SFERR020.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS/SFERR020.sai
Working file: SFERR020.sai
head: 1.2
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv
total revisions: 2

============================================================================= 

The file SFERR030.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS/SFERR030.sai
The file SFERR040.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SFERR040.sai
Working file: SFERR040.sai
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2

--------------- Assembler Pass Errors/Warnings -------
SAIL Preprocessor/Compiler/Assembler Root Directory: f:\SAIL\SAIL_CSCI\SAIL_v2.4
The file SFERR050.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SFERR050.sai
Working file: SFERR050.sai
head: 1.2
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv
total revisions: 2

--------------- First Pass Errors/Warnings --------------
--------------- Second Pass Errors/Warnings --------------

SAIL Compile Utility, Version: $Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file SFUN010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SFUN010.sai
Working file: SFUN010.sai
head: 1.1
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv
total revisions: 1

--------------- First Pass Errors/Warnings --------------
--------------- Second Pass Errors/Warnings --------------
--------------- Assembler Pass Errors/Warnings -----------
SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///////// COMPILING TASK SIERR010 /////////

The file SIERR010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SIERR010.sai
Working file: SIERR010.sai
head: 1.2
branch:
locks: strict
access list:
symbolic names:

=============================================================================
--------------- First Pass Errors/Warnings -----------------
--------------- Second Pass Errors/Warnings ----------------
--------------- Assembler Pass Errors/Warnings ---------------

SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///////// COMPILING TASK SMEM010 /////////

The file SMEM010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SMEM010.sai
Working file: SMEM010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
The file SNST010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\SNST010.sai
Working file: SNST010.sai
(head: 1.1)

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)
The file SNMPERR.sai does not exist. Attempting to retrieve from RCS ...

RCS file: RCS/SNMPERR.sai
Working file: SNMPERR.sai
head: 1
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv

Total revisions: 1

=============================================================================

--------------- First Pass Errors/Warnings ----------------

--------------- Second Pass Errors/Warnings ----------------

Error: Calling depth may exceed 32. While calling function function32 this limit was exceeded.
Error: Line: 444 Column: 10 File: SNMPERR.sai

--------------- Assembler Pass Errors/Warnings ------------

SAIL Compile Utility, Version: $Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory: f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

//\//\//\//\//\// COMPILING TASK SOP010 //\//\//\//\//\//

The file SOP010.sai does not exist. Attempting to retrieve from RCS ...

RCS file: RCS/SOP010.sai
Working file: SOP010.sai
head: 1
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv

Total revisions: 1

=============================================================================

--------------- First Pass Errors/Warnings ----------------

--------------- Second Pass Errors/Warnings ----------------
The file SOP020.sai does not exist. Attempting to retrieve from RCS ...

RCS file: RCS/SOP020
Working file: SOP020
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

The file SOVR010.sai does not exist. Attempting to retrieve from RCS ...

RCS file: RCS/SOVR010
Working file: SOVR010
head: 1.2
branch:
locks: strict
access list:
The file SRTL010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS/SRTL010.sai
Working file: SRTL010.sai
head: 1.1
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv
total revisions: 1
The file SST010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS\SST010.sai
Working file: SST010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================  
--------------- First Pass Errors/Warnings  -----------
--------------- Second Pass Errors/Warnings -----------
--------------- Assembler Pass Errors/Warnings -------

SAIL Compile Utility, Version:
$Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.4
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

//\\\\\\\\\ COMPILING TASK STYPE010 //\\\\\\\\\

The file STYPE010.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS\STYPE010.sai
Working file: STYPE010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================  
--------------- First Pass Errors/Warnings  -----------
--------------- Second Pass Errors/Warnings -----------
--------------- Assembler Pass Errors/Warnings -------


117
The file STOVR010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\STOVR010.sai
Working file: STOVR010.sai
head: 1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------

The file WAIT010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\WAIT010.sai
Working file: WAIT010.sai
head: 1
The file WAIT020.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\WAIT020.sai
Working file: WAIT020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

RCS file: RCS\SASMoverSizeLimit.a
Working file: SASMoverSizeLimit.a
@@@---------- Assembler Pass Errors/Warnings -------
@@@- Assembler errors
@@@- Code size exceeds 64K at line 65539
@@@- F:\SAIL\SAIL_CSCI\SAIL_v2.4\SAILtasks\SAIL_SWACC\SASMoverSizeLimit.a
@@@- SAIL Compile Utility, Version: $Id: sc.bat 1.4 2002/06/07 07:38:42 jpriley Exp $
@@@- SAIL Preprocessor/Compiler/Assembler Root Directory: 
@@@- (Note: To change Root directory, modify the SAIL_ROOT user environment variable) 

\<<<<<<<<< COMPILING TASK SASMsizeLimit >>>>>>>>>>

\<< file: RCS\SASMsizeLimit.a
\<< Working file: SASMsizeLimit.a
\<< head: 1.1
\<< branch:
\<< locks: strict
\<< access list:
\<< symbolic names:
\<< keyword substitution: kv
\<< total revisions: 1

@---------- Assembler Pass Errors/Warnings -------
F:\SAIL\SAIL_CSCI\SAIL_v2.4\SAILtasks\SAIL_SWACC\SASMsizeLimit.a

\<<<<<<<<< SCERR010 errors >>>>>>>>>>

\<<<<<<<<< SCERR010 errors >>>>>>>>>>

Error occurred in file SCERR010.sai, line 16, column 26
Error occurred in file SCERR010.sai, line 17, column 26
Error occurred in file SCERR010.sai, line 18, column 2
Error occurred in file SCERR010.sai, line 20, column 1
Error: The file unexpectedly ended.
Error occurred in file scerr020.sai, line 12, column 2

Error: The end of the file was reached, but the comment was not terminated.
Error occurred in file scerr020.sai, line 12, column 2

Error: An invalid floating point number was encountered.
Error occurred in file scerr020.sai, line 12, column 2

Error: This floating point number has no sign for its exponent.
Error occurred in file scerr020.sai, line 12, column 2

Error: The exponent for this floating point number is too large.
Error occurred in file scerr020.sai, line 12, column 2

Error: This exponent has no sign for its exponent.
Error occurred in file scerr020.sai, line 12, column 2

Error: This floating point number has created overflow.
Error occurred in file scerr020.sai, line 12, column 2

Fatal Error: A symbol has violated the symbol size limitation.
Fatal Error occurred in file scerr020.sai, line 12, column 2

------------ SCERR040 errors --------------

Error: The type of the reference parameter badref must match the exact type of
the reference argument.

Error: You cannot pass Immediate or Expressions by reference.

Error: While passing baddimension to function normalarray, the dimension(s) of
baddimension did not match the dimension(s) of array as defined in the prototype
for function normalarray.

Error: The function functionwithreturn has a non-void return type, but it does
not appear to return a value.
Warning:  Line: 111 Column: 5 File: SCERR040.sai

Warning: The function functionwithreturn has a non-void return type, but it does
not appear to return a value.
Warning:  Line: 117 Column: 20 File: SCERR040.sai

Warning: The function functionwithreturn has a non-void return type, but it does
not appear to return a value.

Error: An incorrect number of parameters were passed to function parameter.

Error: A goto cannot jump to the label unplaced because it has not been placed.
Error:  Line: 135 Column: 2 File: SCERR040.sai

Error: The label alreadyplaced has already been placed.
Error:  Line: 143 Column: 2 File: SCERR040.sai
Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock02 is in an inner block relative to the jumping goto statement.
Error: Line: 229 Column: 5 File: SCERR040.sai
Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock03 is in an inner block relative to the jumping goto statement.
Error: Line: 236 Column: 3 File: SCERR040.sai
Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock04 is in an inner block relative to the jumping goto statement.
Error: Line: 237 Column: 3 File: SCERR040.sai
Error: The function functionwithoutreturn must have a non-void return type to be used in an expression.
Error: Line: 275 Column: 34 File: SCERR040.sai
Error: The function whoami must be called.
Error: In an expression, the token = can't be placed in assignment.
Error: Line: 289 Column: 10 File: SCERR040.sai
Error: You must define a main function.
Error: Line: 290 Column: 1 File: SCERR040.sai
Error: The function main has been prototyped, but not defined.
Error: Line: 290 Column: 1 File: SCERR040.sai
------- SASMoverSizeLimit errors ---------
@Assembler errors
Code size exceeds 64K at line 65539
9.1.3 Sar102 2002 06 17 presentation.txt
# Converted from raw 2002/06/17-17:14-25 to 2002/06/17-16:30-05
1/100/13 9/13/23 13/13/23 31/13/23 31/13/23 31/13/23
9.1.4 Sail Task Code

9.1.4.1 SARR Test Case

9.1.4.1.1 SAIL TASK SARR010.sai

// SARR010 sal test that SAIL can handle arrays of any type up to the maximum size
// $Id: SARR010.sai 1.1 2002/05/07 07:28:31 jpriley Exp $

/* $Log: SARR010.sai $ 
Revision 1.1  2002/05/07 07:28:31  jpriley 
Initial revision */

//

void main(void)
{
  int notUsedHere;
  int intVal = 1;
  int taskID;
  logical logicalVal = TRUE;
  logical logicalArray[16350];
  while(TRUE)
  {
    for(intVal=0;intVal<16350;intVal = intVal + 1)
    {
      logicalArray[intVal] = logicalVal;
      if(intVal%100 == 0)
      {
        put_int(intVal,0);
        put_ascii(13);
        put_ascii(10);
9.1.4.1.2 SAIL TASK SARR011.sai

// SARR011.sai test that SAIL can handle arrays of any type up to the
// maximum size (Compiles, assembles, loads and runs, checks hypokrayg 0 - 10249
// file SARR011.sai 1.1 SRL99/2002/07/18/13 [Jerry]

/* $Id: SARR011.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
*/

void main(void)
{
  int intVal = 1;
  int taskID;
  byte byteVal = 255;
  // logical
  byte byteArray[16350];
  while(TRUE)
  {
    for(intVal=0;intVal<16350;intVal = intVal + 1)
    {
      byteArray[intVal] = byteVal;
      if(intVal%100 == 0)
      {
        put_int(intVal,0);
        put_ascii(13);
        put_ascii(10);
        wait_frame(0);
      }
    }
    taskID = whoami();
  }
}
9.1.4.1.3 SAIL TASK SARR012.sai

// SARR012 test
// SAIL can handle arrays of any type up to the maximum size
// Compile, assemble, load and run, check unsignedArray 0 - 16349
// $Id: SARR012.sai 1.1 2002/05/07 07:28:31 jpriley $
/* $Log: SARR012.sai $ */
/* Revision 1.1  2002/05/07 07:28:31 jpriley */

void main(void)
{
  int intVal = 1;
  int taskID;
  unsigned unsignedVal = 65535;
  unsigned unsignedArray[16350];
  while(TRUE)
  {
    for(intVal=0;intVal<16350;intVal = intVal + 1)
    {
      unsignedArray[intVal] = unsignedVal;
      if(intVal%100 == 0)
      {
        put_int(intVal,0);
        put_ascii(13);
        put_ascii(10);
        wait_frame(0);
      }
    }
    taskID = whoami();
    put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
    suspend(taskID);
  }
}

/* $Log: SARR012.sai $ */
/* Revision 1.1  2002/05/07 07:28:31 jpriley */
void main(void)
{
  int intVal = 1;
  int taskID;

  int intArray[16350];

  while(TRUE)
  {
    for(intVal=0;intVal<16350;intVal = intVal + 1)
    {
      intArray[intVal] = intVal;
      if(intVal%100 == 0)
      {
        put_int(intVal,0);
        put_ascii(13);
        put_ascii(10);
        wait_frame(0);
      }
    }

    taskID = whoami();
    put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
    suspend(taskID);
  }
}
void main(void)
{
    int intVal = 1;
    int taskID;
    float floatVal = 1.797693134862315E+308;
    // logical
    float floatArray[8175];
    while(TRUE)
    {
        for(intVal=0;intVal<8175;intVal = intVal + 1)
        {
            floatArray[intVal] = floatVal;
            if(intVal%100 == 0)
            {
                put_int(intVal,0);
                put_ascii(13);
                put_ascii(10);
                wait_frame(0);
            }
        }
        taskID = whoami();
        put_ascii(taskID + 48);
        put_ascii(13);
        put_ascii(10);
        suspend(taskID);
    }
}

9.1.4.1.6 SAIL TASK SARR020.sai
// SARR020.sai test that SAIL can handle arrays of any type up to the
// expected size
// SARR020.sai tests the SAIL Array Handler.  This file generates a large array to
// test the SAIL Array Handler.  This file is large
// $Id: SARR020.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
void main(void)
{
  int intVal = 1;
  int taskID;
  int passIntVal = 0x12345678;
  // integer
  int intArray[16350 + 10];

  while(TRUE)
  {
    for(intVal=0;intVal<16350;intVal = intVal + 1)
    {
      intArray[intVal] = intVal;

      if(intVal%100 == 0)
      {
        put_int(intVal,0);
        put_ascii(13);
        put_ascii(10);
        wait_frame(0);
      }
    }
    taskID = whoami();
    put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
    suspend(taskID);
  }
}

9.1.4.1.7 SAIL TASK SARR030.sai

// SARR030.sai test - can SAIL handle arrays in a variety of conditions
// compile, assemble, load and run, check intArray in SAIL memory
// $Id: SARR030.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
/* $Log: SARR030.sai $ */
/* Revision 1.1  2002/05/07 07:28:31  jpriley */
//
```c
int intFunction(int index); 
int function1(int intArray[3][10], int index); 
int function2(int intArray[3][10], int index); 

void main(void) 
{ 
    int intVal = 1; 
    int taskID; 
    int passedIntVal = 1; 
    // logical 
    intArray[3][10]; 
    while(TRUE) 
    { 
        intArray[0][1] = function1(intArray,0); // index = 0 -> array[0][1] = 1; 
        intArray[0][3] = function1(intArray,2); // index = 2 -> array[0][3] = 3; 
        intArray[0][5] = function1(intArray,4); // index = 4 -> array[0][5] = 3; 
        intArray[0][7] = function1(intArray,6); // index = 6 -> array[0][7] = 3; 
        intArray[0][9] = function1(intArray,8); // index = 8 -> array[0][9] = 3; 
        intArray[1][1] = intFunction(intArray[1][0] + 1); 
        intArray[1][3] = intArray[1][intArray[0][2]] + 1; 
        intArray[1][5] = intArray[1][intArray[0][2] + 2] + 1; 
        intArray[1][7] = intArray[1][intArray[0][4 + 2]] +1; 
        intArray[1][9] = intArray[1][7 + 1] + 1; 
        intArray[2][intFunction(intArray[0][1] - 1)] = 20; 
        intArray[2][intArray[0][1]] = 21; 
        intArray[2][intArray[0][1] + 1] = 22; 
        intArray[2][intArray[0][2 + 1]] = 23; 
        intArray[2][intArray[0][2]] = 24; 
        taskID = whoami(); 
        put_ascii(taskID + 48); 
        put_ascii(13); 
        put_ascii(10); 
        suspend(taskID); 
    } 

    int intFunction(int index) 
    { 
        return index; 
    } 

    int function1(int intArray[3][10], int index) 
    { 
        intArray[0][index] = function2(intArray,index); // array[0][0] = 0, 
        return intArray[0][index] + 1; 
    } 

    int function2(int intArray[3][10], int index) 
    { 
    } 
```
9.1.4.2 SCERR Test Case

9.1.4.2.1 SAIL TASK SCERR010

// SCERR010 test case to test compiler pass #1 handles error correctly
// Tested OK 05/30/01 TJS
// $Id: SCERR010.sai 1.1 2002/05/07 07:28:31 jpriley Exp $

#include <stdio.h>

float badFloat = 1234A.e-22; // NUMBER_BAD_FLOAT
float badFloatSignExpected = 1234.5e22; // NUMBER_BAD_FLOAT_SIGN_EXPECTED
float badFloatExponentExpected = 1234.5e-; // NUMBER_BAD_FLOAT_EXPONENT_EXPECTED
float badFloatExponentTooLarge = 1234.5e-2222; // NUMBER_BAD_FLOAT_EXPONENT_TOO_LARGE
float badFloatTooLong = 12345678901234567890123456e-22; // NUMBER_BAD_FLOAT_TOO_LONG
int intOverflow = 3000000000; // NUMBER_OVERFLOW
float floatOverflow = 3.0e+308; // NUMBER_FLOAT_OVERFLOW
int xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx; // SYMBOL_TOO_LONG
9.1.4.2.3 SAIL TASK SCERR030.sai

// SCERR030.sai included in SCERR010
// $Id: SCERR030.sai 1.1 2002/05/07 07:28:31 jpriley Exp $

/* $Log: SCERR030.sai $ */
/* Revision 1.1  2002/05/07 07:28:31  jpriley */
/* Initial revision */
/* */
/* */

9.1.4.4 SAIL TASK SCERR040.sai

// SAIL compiler pass #2 handles error correctly
// Tested OK 06/01/01
// $Id: SCERR040.sai 1.2 2002/06/16 12:17:42 jpriley Exp $

/* $Log: SCERR040.sai $ */
/* Revision 1.2  2002/06/16 12:17:42  jpriley */
/* Removed obsolete error function prototype. */
/* Revision 1.1  2002/05/07 07:28:31  jpriley */
/* Initial revision */
/* */
/* */

// error file for pass #2 should include each of the errors listed
// 9 times
// E_RefArgTypeMismatch
// E_BadRefParameter
// E_ExpressionSyntax
// E_DimensionMismatch
// E_MustReturnValue
// E_ReturnIgnored
// E_ParameterMismatch
// E_GotoUnplacedLabel
// E_LabelAlreadyPlaced
// E_Undefined
// E_Duplicate
// E_Expected
// E_ExpressionTooComplex
// E_ExpressionParentheses
// E_CallOfNonfunction
// E_Disabled
// E_NoPrototype
// E_CantReturnValue
// E_GotoInnerBlock
// E_ExpressionFunction
// E_MustCallFunction
// E_ExpressionSyntax
// E_NoMain
// E_NoDefinition
void functionNormal(void);
void functionNormalArray(int array[4][5]);
void functionCallWithDimensionMismatch(void);
int functionWithReturn(void);
void functionReturnIgnored(void);
void functionWithoutReturn(void);
void functionShouldNotReturn(void);
void functionExpressionFunction(void);
void functionRefParameter(int& badRef);
void functionBadRefParameter(void);
void functionParameter(int parameterMismatch);
void functionParameterMismatch(void);
int functionCallInsideError(void);
void functionGotoInnerBlock(void);
void functionGotoUnplacedLabel(void);
void functionLabelAlreadyPlaced(void);
void functionUndefined(void);
void functionDuplicate(void);
void functionBadArray(void);
void functionExpressionSyntax(void);
void functionExpressionTooComplex(void);
void functionExpressionParentheses(void);
void functionCallOfNonfunction(void);
void functionMustCallFunction(void);
void functionRefArgTypeMismatch(void);
void functionMultipleAssign(void);

/*******************************************************************************
*/
void functionNormal(void) {
}
/*******************************************************************************
*/
void functionRefParameter(int& badRef) {
int i;
}

/****************************************************************************/
void functionRefParameter(int& badRef) {
}

/*******************************************************************************
*/
void functionRefParameterMismatch(void) {
int i;
}

/*******************************************************************************
*/
void functionExpressionSyntax(void) {
int assignTo;
int intArray[5];
assignTo = intArray[4][1]; // E_ExpressionSyntax
}

/*******************************************************************************
*/
void functionGotoInnerBlock(void) {
}

/*******************************************************************************
*/
void functionGotoUnplacedLabel(void) {
}

/*******************************************************************************
*/
void functionLabelAlreadyPlaced(void) {
}

/*******************************************************************************
*/
void functionUndefined(void) {
}

/*******************************************************************************
*/
void functionDuplicate(void) {
}

/*******************************************************************************
*/
void functionBadArray(void) {
}

/*******************************************************************************
*/
void functionExpressionSyntax(void) {
int assignTo;
int intArray[5];
assignTo = intArray[4][1]; // E_ExpressionSyntax
}

*******************************************************************************
void functionNormalArray(int array[4][5])
{
}

void functionCallWithDemensionMismatch(void)
{
    int badDimension[4][6];
    functionNormalArray(badDimension); // E_DimensionMismatch
}

int functionWithReturn(void)
{
} // E_MustReturnValue

void functionReturnIgnored(void)
{
    functionWithReturn(); // E_ReturnIgnored
}

void functionParameter(int parameterMismatch)
{
}

void functionParameterMismatch(void)
{
    functionParameter(); // E_ParameterMismatch
}

void functionGotoUnplacedLabel(void)
{
    label unplaced;
    goto unplaced; // E_GotoUnplacedLabel
}

void functionLabelAlreadyPlaced(void)
{
    label alreadyPlaced;
    alreadyPlaced:
    alreadyPlaced: // E_LabelAlreadyPlaced
}

void functionUndefined(void)
{
    int i;
    i = undefined; // E_Undefined
}
void functionDuplicate(void)
{
    int duplicate;
    int duplicate; // E_Duplicate
    duplicate = 1;
}

void functionBadArray(void)
{
    int badArray[-1]; // E_Expected
}

void functionExpressionTooComplex(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z;
    a = (b+c)+(d+e)+(f+g)+(h+i)+(j+k)+(l+m)+(n+o)+(p+q)+(r+s)+(t+u)+(v+w)+z; // E_ExpressionTooComplex
}

void functionExpressionParentheses(void)
{
    int assignTo;
    int assign;
    assignTo = (assign; // E_ExpressionParentheses
}

void functionCallOfNonfunction(void)
{
    int x,y,z;
    int callOfNonFunction;
    x = y*callOfNonFunction() + z; // E_CallOfNonfunction
}

int functionCallInsideError(void)
{
    return 1;
}

void error(void)
{
    int intVal;
    intVal = functionCallInsideError(); // E_Disabled
}

void functionWithoutAPrototype(void) // E_NoPrototype

void functionShouldNotReturn(void)
{
    return 1; // E_CantReturnValue
}

void functionGotoInnerBlock(void)
{
    label InnerBlock01;
    label InnerBlock02;
    label InnerBlock03;
    label InnerBlock04;
    label InnerBlock05;
    label InnerBlock06;
    label InnerBlock07;
    label InnerBlock08;
    label InnerBlock09;
    label InnerBlock10;
    int i,j;
    goto InnerBlock01; // error #6 E_GotoInnerBlock
    for(i=0;i<10;i=i+1)
    {
        i = i + 1;
        InnerBlock01:
        InnerBlock06:
        InnerBlock08:
        goto InnerBlock02; // error #7 E_GotoInnerBlock
        goto InnerBlock03; // error #8 E_GotoInnerBlock
        for(j=0;j<10;j=j+1)
        {
            i = i + 1;
            InnerBlock03:
            InnerBlock05:
            InnerBlock07:
            InnerBlock09:
            goto InnerBlock04; // error #9 E_GotoInnerBlock
        }
        goto InnerBlock05; // error #1 E_GotoInnerBlock
    }
    goto InnerBlock06; // error #2 E_GotoInnerBlock
    goto InnerBlock07; // error #3 E_GotoInnerBlock
   for(i=0;i<10;i=i+1)
   {
    i = i + 1;
    InnerBlock02:
    InnerBlock04:
    goto InnerBlock08; // error #4 E_GotoInnerBlock
    goto InnerBlock09; // error #5 E_GotoInnerBlock
   }
}


/*
 * void functionWithOutReturn(void)
 */
void functionExpressionFunction(void)
{
    int assignTo;
    int assignFrom = 10;
    assignTo = functionWithOutReturn(); // E_ExpressionFunction
}
/*
 * void functionMustCallFunction(void)
 */
void functionMultipleAssign(void)
{
    int x,y;
    x = y = 4; // E_ExpressionSyntax
}
// E_NoMain
// E_NoDefinition
/**************************** ERROR FILE
****************************8
Error: The type of the reference parameter badref must match the exact type of
the corresponding reference argument.
Error: You cannot pass Immediates or Expressions by reference.
Error: In an expression, the token [ can't be placed after the value intarray.
Error: While passing baddimension to functionnormalarray, the dimensions of
baddimension did not match the dimensions of array as defined in the prototype
for functionnormalarray.
Error:  Line: 97 Column: 22 File: SCERR040.sai
Error: The function functionwithreturn has a non-void return type, but it does
not return a value.
Error:  Line: 101 Column: 5 File: SCERR040.sai
Warning: The return value from function functionwithreturn was ignored.
Error: An incorrect number of parameters were passed to functionparameter.
Error:  Line: 118 Column: 20 File: SCERR040.sai
Error & prior errors jump to the label explained because it has not been placed.
Error: The label alreadyplaced has already been placed.
Error: Line: 125 Column: 2 File: SCERR040.sai

Error: The identifier undefined cannot be used until it is defined.
Error: Line: 133 Column: 2 File: SCERR040.sai

Error: The identifier name duplicate has already been used.
Error: Line: 140 Column: 6 File: SCERR040.sai

Error: A non-zero, non-negative constant expression result was expected to
dimension an array, but the token -1 was found instead.

Error: This expression is too complex.

Error: This expression is missing parentheses.
Error: Line: 170 Column: 20 File: SCERR040.sai

Error: Only functions may be called. An attempt was made to call the nonfunction
token callofnonfunction.

Error: The function functionwithoutaprototype cannot be defined until it has
been prototyped.

Error: You cannot use the non-RTL function functioncallinsideerror while inside
the body of the error function.

Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock05 is in an inner block relative to the jumping goto
statement.
Error: Line: 239 Column: 2 File: SCERR040.sai

Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock06 is in an inner block relative to the jumping goto
statement.
Error: Line: 242 Column: 5 File: SCERR040.sai

Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock07 is in an inner block relative to the jumping goto
statement.

Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock08 is in an inner block relative to the jumping goto
statement.
Error: Line: 250 Column: 3 File: SCERR040.sai

Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock09 is in an inner block relative to the jumping goto
statement.
Error: Line: 251 Column: 3 File: SCERR040.sai
Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock01 is in an inner block relative to the jumping goto statement.
Error: Line: 217 Column: 5 File: SCERR040.sai
Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock02 is in an inner block relative to the jumping goto statement.
Error: Line: 226 Column: 3 File: SCERR040.sai
Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock03 is in an inner block relative to the jumping goto statement.
Error: Line: 227 Column: 3 File: SCERR040.sai
Error: A goto cannot jump from an outer code block into an inner code block. The label innerblock04 is in an inner block relative to the jumping goto statement.
Error: The function functionwithoutreturn must have a non-void return type to be used in an expression.
Error: Line: 265 Column: 34 File: SCERR040.sai
Error: The function whoami must be called.
Error: Line: 272 Column: 12 File: SCERR040.sai
Error: In an expression, the token = can't be placed in assignment.
Error: Line: 279 Column: 10 File: SCERR040.sai
Error: You must define a main function.
Error: Line: 280 Column: 1 File: SCERR040.sai
Error: The function main has been prototyped, but not defined.
Error: Line: 280 Column: 1 File: SCERR040.sai
*/

9.1.4.2.5 SAIL TASK SCERR050.sai

void functionNormal(void);
void functionRecursion(void);
int devideByZero = 1/0; // E_DivideByZero

*/

// error file for pass #2 should include each of the errors listed above
// only one of these errors can occur at a time
void main(void)
{
    functionNormal();
}

void functionNormal(void)
{
    int i;
    float floatMismatch;
    functionRecursion();
    functionDevideByZero();
}

// this function must be called for the compiler to flag the error
void functionDevideByZero(void)
{
    // functionDevideByZero(); // E_Recursion
}

/********************************** ERROR FILE **********************************
Fatal Error: A constant expression attempted to divide by zero. The numerator
was: 1.

9.1.4.2.6 SAIL TASK SCERR060.sai
// SCERR050.sai test that sail compiler pass #2 handles error correctly
// Tested OK 06/01/02 TJS
// $Id: SCERR060.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
/* $Log: SCERR060.sai $ */
/* Revision 1.1  2002/05/07 07:28:31  jpriley */
/* Initial revision */

// error file for pass #2 should include each of the errors listed here
// E_Recursion
// only one of these errors can occur at a time
void functionNormal(void)
void functionRecursion(void)
void functionDevideByZero(void)
int devideByZero = 1/0; // E_DivideByZero
void main(void)
{
    functionNormal();
}

void functionNormal(void)
{
    int i;
    float floatMismatch;
}

Synopsis:
functionRecursion();

// this function must be called for the compiler to flag the error
void functionRecursion(void)
{
    functionRecursion(); // E_Recursion
}

/************************************* ERROR FILE *****************************/
Error: The function functionrecursion cannot call itself, directly or
indirectly.
Error: Line 29 Column 6 File: SCERR060.sai
*

9.1.4.3 SCMP Test Case

9.1.4.3.1 SAIL TASK SCMP010.sai

// SCMP010.sai tests that pass1 removes both type of comments

main()
{
    /* first type of comment
    can be on multiple lines
    */
    // second type of comment is on a hole single line
    // in each file, // in only part of it
    // check the SCMP010.p1 output file to verify that the comments, and only the
    // comments have bin removed
    // correct output is shown below
    /*
    84 103 0 6 1  main
    85 21 0 6 5  (n
    86 22 0 6 6  )
    87 49 0 7 1  {
    88 2 0 14 4  int
    89 104 0 14 8  intval
    90 47 0 14 14  ;
    91 50 0 17 1  }
    */
9.1.4.3 SAIL TASK SCMP020.sai

// SCMP020.c test each of the condition statements
// SCMP020 test 1.2.26/02/2002 07:18:31 Gordon Esp 6
// Simple SCMP020 test 0
// Revision 1.1.26/02/2002 07:18:31 Gordon Esp 6

void main(void)
{
  int index, secondIndex;
  int intVal = 1;
  int intArray[0x200];
  label label_1;
  int taskID;

  while(TRUE)
  {
    // simple if-else
    if(intVal == 1)
      intArray[intVal] = intVal;
    else
      intArray[intVal] = -intVal;

    intVal = intVal + 1;  // 2
    if(intVal != 2)
      intArray[intVal] = -intVal;
    else
      intArray[intVal] = intVal;

    // short-circuit conditional processing
    intVal = intVal + 1;  // 3
    if(intVal == 0 && 10/(intVal - intVal) < 10)
      intArray[intVal] = -intVal;
    else
      intArray[intVal] = intVal;

    intVal = intVal + 1;  // 4
    if(intVal == intVal || 10/(intVal - intVal) < 10)
      intArray[intVal] = intVal;
    else
      intArray[intVal] = -intVal;

    ...
  }
}
for loop
for(intVal = 5; intVal < 10; intVal = intVal + 1) // 5-A
    intArray[intVal] = intVal;
if(intVal != 10) // just a check
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;

// while loop
while(intVal < 15)
{
    intArray[intVal] = intVal;
    intVal = intVal + 1;  // C-F
}
if(intVal != 15) // just a check
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
// break in a while
while(intVal < 20)
{
    if(intVal > 17)
        break;
    intArray[intVal] = intVal;
    intVal = intVal + 1;  // 11-12
}
if(intVal != 18) // just a check
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
// continue in a while
while(intVal < 20)
{
    intVal = intVal + 1;  // 13-14
    intArray[intVal] = intVal;
    continue;
    intArray[intVal] = -intVal; // never executed;
}
// break in a for
for(intVal = intVal; intVal < 25; intVal = intVal + 1) // 15-17
{
    if(intVal > 22)
        break;
    intArray[intVal] = intVal;
}
if(intVal != 23) // just a check
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
// continue in a for
for(intVal = intVal; intVal < 30; intVal = intVal + 1) // 19-1F
{
    intArray[intVal] = intVal;
    continue;
    intArray[intVal] = -intVal; // never executed;
}
// goto
label_1:
    intArray[intVal] = intVal;
    intVal = intVal + 1;   // 0x20-0x25
    if(intVal < 0x25)
        goto label_1;
    intArray[intVal] = intVal;
    // continue in while inside for inside while
    while(TRUE)
    {
        intVal = intVal + 1; // 0x26
        intArray[intVal] = intVal;
        for(index = intVal; index < 0x28; index = index + 1)
        {
            intVal = intVal + 1; // 0x27 + 0x32
            if(intVal > 0x27 && intVal < 0x31)
                intArray[intVal] = -intVal; // while continue went to up one lable
            else
                intArray[intVal] = intVal;
        }
        while(TRUE)
        {
            intVal = intVal + 1; // 0x28-0x30 + 0x33
            intArray[intVal] = intVal;
            if(intVal < 0x30)
                continue;
            break;
        }
    }
    intVal = intVal + 1; // 0x31 + 0x34
    if(intVal < 0x31)
        intArray[intVal] = -intVal; // while continue falled
    else
        intArray[intVal] = intVal;
    continue;
    intVal = intVal + 1;
    intArray[intVal] = -intVal;
}
break;
intVal = intVal + 1; // 0x35
if(intVal == 0x35)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
// continue in for inside for inside while
for(index = 0; index < 1; index = index + 1)
{
intVal = intVal + 1; // 0x36
intArray[intVal] = intVal;
for(secondIndex = intVal; secondIndex < 0x38; secondIndex =
secondIndex + 1)
{
    intVal = intVal + 1; // 0x37 + 0x41
    if(intVal > 0x37 && intVal < 0x41)
        intArray[intVal] = -intVal;
    else
        intArray[intVal] = intVal;
while(TRUE)
{
    intVal = intVal + 1; // 0x38 - 0x39 + 0x42
    intArray[intVal] = intVal;
    if(intVal < 0x3F)
        continue;
    break;
}
intVal = intVal + 1; // 0x40 + 0x43
    if(intVal < 0x40)
        intArray[intVal] = -intVal;
    else
        intArray[intVal] = intVal;
    continue;
intVal = intVal + 1;
    intArray[intVal] = -intVal;
    }
    if(intVal == 0x44)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    taskID = whoami();
    put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
suspend(taskID);
}
Send command to allow SAIL to send a hir_spu_timing command and then Restart task
Check intArray 0x00 - 0x07
Check the command output file
Check SAIL display command counters

Also checks the following communication variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Offset</th>
<th>Size</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>_q_size</td>
<td>int</td>
<td>24</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>_cmd_received</td>
<td>int</td>
<td>26</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>_cmd_rejected</td>
<td>int</td>
<td>27</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>_cmd_result</td>
<td>int</td>
<td>28</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>_cmd_number</td>
<td>int</td>
<td>29</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>_executed_cmd_pktseq</td>
<td>int</td>
<td>31</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>_qued_cmd_pktseq</td>
<td>int</td>
<td>32</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

```c
void main(void) {

  int intArray[0x10];
  int intIndex;
  int q_size;
  int cmd_received;
  int cmd_rejected;
  int cmd_result;
  int executed_cmd_pktseq;
  int qued_cmd_pktseq;
  int taskID;

  while(TRUE) {
    // The operator must now send the command to not allow SAIL to send a
    // hir_spu_timing command
    intIndex = 0;
    intArray[intIndex] = -1;
    q_size = _q_size;
    cmd_received = _cmd_received;
    cmd_rejected = _cmd_rejected;
    command(hir_spu_timing, 0, 3, 0x1234); // this command should be
    while(_q_size != q_size);           // wait for the command to be
    if(_cmd_rejected == cmd_rejected + 1)
      intArray[intIndex] = intIndex; // 0x00
    intIndex = intIndex + 1; // 0x01
    taskID = whoami();
    put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
    suspend(taskID);
  }
}
```
The operator must now send the command to allow SAIL to send a
hir_spu_timing command
and then Resume the task

// The operator must now send the command to allow SAIL to send a
// hir_spu_timing command
// and then Resume the task
put_ascii(88);
executed_cmd_pktseq = _executed_cmd_pktseq;
qued_cmd_pktseq = _qued_cmd_pktseq;
q_size = _q_size;
cmd_received = _cmd_received;
cmd_rejected = _cmd_rejected;

 command(hir_spu_timing,0,3,0x1234); // this command should be accepted

if(_q_size == q_size + 1)
  intArray[intIndex] = intIndex; // 0x01
  intIndex = intIndex + 1; // 0x02

if(_qued_cmd_pktseq == qued_cmd_pktseq + 1)
  intArray[intIndex] = intIndex; // 0x02
  intIndex = intIndex + 1; // 0x03

while(_q_size != q_size);               // wait for the command to be taken
  intArray[intIndex] = intIndex; // 0x03
  intIndex = intIndex + 1; // 0x04

if(_cmd_received == cmd_received + 1)
  intArray[intIndex] = intIndex; // 0x04
  intIndex = intIndex + 1; // 0x05

if(_cmd_rejected == cmd_rejected)
  {
    put_int(2,FORMAT_DEC);
    put_ascii(41);
    put_ascii(32);
    put_float(cmd_rejected,FORMAT_DEC);
    put_ascii(61);
    cmd_rejected = _cmd_rejected;
    put_float(cmd_rejected,FORMAT_DEC);
    put_ascii(13);
    put_ascii(10);
  }
else
  {
    put_int(0,FORMAT_DEC);
    put_ascii(0);
    put_ascii(0);
    put_ascii(0);
    put_ascii(0);
    put_ascii(0);
    put_ascii(0);
  }

  intArray[intIndex] = intIndex; // 0x05
  intIndex = intIndex + 1; // 0x06

if(_cmd_number == 39) // hir_spu_timing
  intArray[intIndex] = intIndex; // 0x06
  intIndex = intIndex + 1; // 0x07

if(_executed_cmd_pktseq == executed_cmd_pktseq + 1)
  intArray[intIndex] = intIndex; // 0x07
  intIndex = intIndex + 1; // 0x08

put_ascii(taskID + 48);
put_ascii(13);
put_ascii(10);
suspend(taskID);
}
9.1.4.4.2 SAIL TASK SCOMS020.sai

// SCOMS020.sai Tests the ability of the IPU to handle maximum commands rate. If IPU SAIL Cmd Q is full, the F_SAI_CMD_Q_FULL error should be set. No further commands can be sent until the Q has free room.

/* $Id: SCOMS020.sai 1.1 2002/05/07 07:28:31 jpriley Exp $ */

void error(void);
#define F_SAI_CMD_Q_FULL 0x1Fe4
//

int intIndex;

int main(void)
{
  int intArray[0x200];
  int taskID;
  int counter;

  intIndex = 0;
  for(counter = 0; counter < 0x200; counter++)
    intArray[counter] = 0xffff;

  taskID = whoami();
  while(TRUE)
  {
    for(counter = 0; counter < 32; counter++)
    {
      command(hir_ssh_doorunprot);
      intArray[intIndex] = _q_size;
      intIndex = intIndex + 1;

      if(_cmd_result == F_SAI_CMD_Q_FULL)
        intArray[intIndex] = 1;
    }

    if(_q_size >= 64)
    {
      intArray[0] = _cmd_result;
      suspend(taskID);
    }
    else
      wait_frame(0);
  }
}

9.1.4.5 SST Test Case

9.1.4.5.1 SAIL TASK SDATA010.sai
// SDATA010.sal Test each of the following communication variables

#define MAX_MATH_ERROR 0.0000000000001
#define CR_COUNTS 64
#define CHANALS 6
#define DETECTORS 21

int intArray[0x1180];

void main(void)
{
    int cr_count;
    float hirdls_time_stamp = 0;
    int spu_qhss_state = 0;
    int spu_phase_state = 0;
    float radiance, radRead;
    int tss_state = 0;
    int elev1prim = 0x10000 + 0x200;
    int elev1sec = 0x20000 + 0x200;
    int elev2prim = 0x10000 + 0x400;
    int elev2sec = 0x20000 + 0x400;
    int azimprim = 0x10000;
    int azimsec = 0x20000;
    int gyro = 0xAAAA;
    int value;
    int intVal;
    int chopperRotation; // 0 - 36
    int cc; // 0 - 5
    int channal;
    int temp;
}

// Before running this test program load "matrixFile" to set the
// filter factors
// Compile, assemble, load, run.
// Check intArray counts from 0x00 to 0x??

// $Id: SDATA010.sai 1.1 2002/05/07 07:28:31 jpriley Exp $

// $Log: SDATA010.sai $
// Revision 1.1  2002/05/07 07:28:31  jpriley
// Initial revision
//
float ftemp;
int taskID;

wait_frame(0);

while(TRUE)
{
    intVal = 0;

    //********** _cr_count **********//
    OK
    // check that mod 64 of the _cr_count counts from 0 to 63
    for(chopperRotation=0;chopperRotation<64;chopperRotation =
        chopperRotation + 1)
    {
        intVal = intVal + 1;
        if(_cr_count[chopperRotation]%64 == chopperRotation)
        {
            intArray[intVal] = intVal;
        }
        else
        {
            intArray[intVal] = -intVal;
            put_int(1,FORMAT_HEX);
            put_ascii(41);
            put_ascii(32);
            put_int(intVal,FORMAT_HEX);
            put_ascii(32);
            put_int(chopperRotation,FORMAT_DEC);
            put_ascii(61);
            temp = _cr_count[chopperRotation]%64;
            put_int(temp,FORMAT_DEC);
            put_ascii(13);
            put_ascii(10);
        }
    }
    wait_frame(0);
    intVal = 0x40;

    //********** _hirdls_time_stamp **********//
    OK
    // check that the _hirdls_time_stamp increases for each chopper
    hirdls_time_stamp = _hirdls_time_stamp[0]; // store the time of the
    first Copper Rotation
    for(chopperRotation=1;chopperRotation<64;chopperRotation =
        chopperRotation + 1)
    {
        intVal = intVal + 1;
        if(_hirdls_time_stamp[chopperRotation] > hirdls_time_stamp) //
            compare this to the previous Copper Rotation
        {
            intArray[intVal] = intVal;
        }
        else
        {
            intArray[intVal] = -intVal;
            put_int(chopperRotation,FORMAT_DEC);
            put_ascii(41);
            put_ascii(32);
            put_int(intVal,FORMAT_HEX);
            put_ascii(32);
            put_float(hirdls_time_stamp,FORMAT_SCI_FLOAT);
            put_ascii(61);
        }
    }
    wait_frame(0);
}
tr_loc-937c

2003-06-12

152

ftemp = _hirdls_time_stamp[chopperRotation];
put_float(ftemp,FORMAT_SCI_FLOAT);
put_ascii(13);
put_ascii(10);
}

hirdls_time_stamp = _hirdls_time_stamp[chopperRotation]; // store
this time for next check

intArray[intVal] = intVal; //just to keep it even

wait_frame(0);

intVal = 0x80;

//********** _spu_qhss_state **********//

for(chopperRotation=0;chopperRotation<64;chopperRotation =
chopperRotation + 1)
{
    intVal = intVal + 1;
    spu_qhss_state = 0; // input by test program
    if(_spu_qhss_state[chopperRotation] == spu_qhss_state)
    intArray[intVal] = intVal;
    else
    {
        intArray[intVal] = -intVal;
        put_int(3,FORMAT_HEX);
        put_ascii(41);
        put_ascii(32);
        put_int(intVal,FORMAT_HEX);
        put_ascii(32);
        put_int(spu_qhss_state,FORMAT_HEX);
        put_ascii(61);
        temp = _spu_qhss_state[chopperRotation];
        put_int(temp,FORMAT_HEX);
        put_ascii(13);
        put_ascii(10);
    }
}

wait_frame(0);

intVal = 0xC0;

//********** _spu_phase_state **********//

for(chopperRotation=0;chopperRotation<64;chopperRotation =
chopperRotation + 1)
{
    intVal = intVal + 1;
    spu_phase_state = 0; // ??;
    if(_spu_phase_state[chopperRotation] == spu_phase_state)
    intArray[intVal] = intVal;
    else
    {
        intArray[intVal] = -intVal;
        put_int(4,FORMAT_HEX);
        put_ascii(41);
        put_ascii(32);
        put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(spu_phase_state,FORMAT_HEX);
put_ascii(61);
temp = _spu_phase_state[chopperRotation];
put_int(temp,FORMAT_HEX);
put_ascii(13);
put_ascii(10);
}
}
wait_frame(0);
intVal = 0x100;

//********** _radiance **********//
cc = 2; // let's just try this one channel
for(chopperRotation=0;chopperRotation<64;chopperRotation =
chopperRotation + 1)
{
for(channal=0;channal<21;channal = channal + 1)
{
if(channal%8 == 0)
wait_frame(0);

radiance = ((chopperRotation/8)<<13) + ((channal + 1)<<8) +
(((chopperRotation)%8)<<4) + 2*cc;
radRead = (_radiance[chopperRotation*6 + cc][channal];
intVal = intVal + 1;
if(radRead >= radiance - MAX_MATH_ERROR
&& radRead <= radiance + MAX_MATH_ERROR)
intArray[intVal] = intVal;
else
{
intArray[intVal] = -intVal;
put_int(5,FORMAT_HEX);
put_ascii(41);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_float(radiance,FORMAT_SCI_FLOAT);
put_ascii(61);
ftemp = radRead;
put_float(ftemp,FORMAT_SCI_FLOAT);
put_ascii(13);
put_ascii(10);
}
}
wait_frame(0);
intVal = 0x640;

//********** _tss_state **********//
for(chopperRotation=0;chopperRotation<64;chopperRotation =
chopperRotation + 1)
{
intVal = intVal + 1;

//************_time_map ************/
for(chopperRotation=0;chopperRotation<64;chopperRotation =
chopperRotation + 1)
intVal = intVal + 1;
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HEX);
put_ascii(32);
put_int(intVal,FORMAT_HE
tss_state = 0; // The TSS state must be a constant zero

if (_tss_state[chopperRotation] == tss_state)
  intArray[intVal] = intVal;
else
{
  //intArray[intVal] = -intVal;
  intArray[intVal] = (_tss_state[chopperRotation] << 16) & 0xFFFF0000 | (tss_state & 0x0000FFFF);

  put_int(6, FORMAT_HEX);
  put_ascii(41);
  put_ascii(32);
  put_int(intVal, FORMAT_HEX);
  put_ascii(32);
  put_int(tss_state, FORMAT_HEX);
  put_ascii(61);
  temp = _tss_state[chopperRotation];
  put_int(temp, FORMAT_HEX);
  put_ascii(13);
  put_ascii(10);
}

wait_frame(0);

intVal = 0x680;

//********** _elev1prim **********/

for (chopperRotation = 0; chopperRotation < 64; chopperRotation = chopperRotation + 1)
{
  if (chopperRotation % 4 == 0)
    wait_frame(0);

  for (cc = 0; cc < 6; cc = cc + 1)
  {
    intVal = intVal + 1;

    value = elev1prim + ((chopperRotation / 8) << 12) + (((chopperRotation) % 8) << 4) + cc;

    if (_elev1prim[chopperRotation][cc] == value)
      intArray[intVal] = intVal;
    else
    {
      intArray[intVal] = -intVal;
      put_int(7, FORMAT_HEX);
      put_ascii(41);
      put_ascii(32);
      put_int(intVal, FORMAT_HEX);
      put_ascii(32);
      put_int(value, FORMAT_HEX);
      put_ascii(61);
      temp = _elev1prim[chopperRotation][cc];
      put_int(temp, FORMAT_HEX);
      put_ascii(13);
      put_ascii(10);
    }
  }
}

wait_frame(0);

intVal = 0x680;
wait_frame(0);
intVal = 0x800;

//********** _elev1sec **********//
for(chopperRotation=0;chopperRotation<64;chopperRotation =
chopperRotation + 1)
{
    if(chopperRotation%4 == 0)
        wait_frame(0);

    for(cc=0;cc<6;cc = cc + 1)
    {
        intVal = intVal + 1;

        value = elev1sec + ((chopperRotation/8)<<12) +
        (((chopperRotation)%8)<<4) + cc;

        if(_elev1sec[chopperRotation][cc] == value)
            intArray[intVal] = intVal;
        else
        {
            intArray[intVal] = -intVal;
            put_int(8,FORMAT_HEX);
            put_ascii(41);
            put_ascii(32);
            put_int(intVal,FORMAT_HEX);
            put_ascii(32);
            put_int(value,FORMAT_HEX);
            put_ascii(61);
            temp = _elev1sec[chopperRotation][cc];
            put_int(temp,FORMAT_HEX);
            put_ascii(13);
            put_ascii(10);
        }
    }
}

wait_frame(0);
intVal = 0x980;

//********** _elev2prim **********//
for(chopperRotation=0;chopperRotation<64;chopperRotation =
chopperRotation + 1)
{
    if(chopperRotation%4 == 0)
        wait_frame(0);

    for(cc=0;cc<6;cc = cc + 1)
    {
        intVal = intVal + 1;

        value = elev2prim + ((chopperRotation/8)<<12) +
        (((chopperRotation)%8)<<4) + cc;

        if(_elev2prim[chopperRotation][cc] == value)
            intArray[intVal] = intVal;
        else
        {
            intArray[intVal] = -intVal;
            put_int(9,FORMAT_HEX);
            put_ascii(41);
            put_ascii(32);
            put_int(intVal,FORMAT_HEX);
            put_ascii(32);
            put_int(value,FORMAT_HEX);
            put_ascii(61);
            temp = _elev2prim[chopperRotation][cc];
            put_int(temp,FORMAT_HEX);
            put_ascii(13);
            put_ascii(10);
        }
    }
}
put_ascii(32);
put_int(value,FORMAT_HEX);
put_ascii(61);
temp = _elev2prim[chopperRotation][cc];
put_int(temp,FORMAT_HEX);
put_ascii(13);
put_ascii(10);
}
}
}
wait_frame(0);
intVal = 0xB00;

//********** _elev2sec **********//
for(chopperRotation=0;chopperRotation<64;chopperRotation =
chopperRotation + 1)
{
    if(chopperRotation%4 == 0)
        wait_frame(0);
    for(cc=0;cc<6;cc = cc + 1)
    {
        intVal = intVal + 1;
        value = elev2sec + ((chopperRotation/8)<<12) +
        (((chopperRotation)%8)<<4) + cc;
        if(_elev2sec[chopperRotation][cc] == value)
            intArray[intVal] = intVal;
        else
        {
            intArray[intVal] = -intVal;
            put_int(10,FORMAT_HEX);
            put_ascii(41);
            put_ascii(32);
            put_int(intVal,FORMAT_HEX);
            put_ascii(32);
            put_int(value,FORMAT_HEX);
            put_ascii(61);
            temp = _elev2sec[chopperRotation][cc];
            put_int(temp,FORMAT_HEX);
            put_ascii(13);
            put_ascii(10);
        }
    }
}
wait_frame(0);
intVal = 0xC80;

//********** _azimprim **********//
for(chopperRotation=0;chopperRotation<64;chopperRotation =
chopperRotation + 1)
{
    if(chopperRotation%4 == 0)
        wait_frame(0);
    for(cc=0;cc<6;cc = cc + 1)
```c
intVal = intVal + 1;
value = azimprim + ((chopperRotation/8)<<12) + (((chopperRotation)%8)<<4) + cc;
if(_azimprim[chopperRotation][cc] == value)
    intArray[intVal] = intVal;
else
{
    intArray[intVal] = -intVal;
    put_int(11,FORMAT_HEX);
    put_ascii(41);
    put_ascii(32);
    put_int(intVal,FORMAT_HEX);
    put_ascii(32);
    put_int(value,FORMAT_HEX);
    put_ascii(61);
    temp = _azimprim[chopperRotation][cc];
    put_int(temp,FORMAT_HEX);
    put_ascii(13);
    put_ascii(10);
}
}
wait_frame(0);
intVal = 0xE00;

//********** _azimsec **********//
for(chopperRotation=0;chopperRotation<64;chopperRotation = chopperRotation + 1)
{
for(cc=0;cc<6;cc = cc + 1)
{
intVal = intVal + 1;
value = azimsec + ((chopperRotation/8)<<12) + (((chopperRotation)%8)<<4) + cc;
if(_azimsec[chopperRotation][cc] == value)
    intArray[intVal] = intVal;
else
{
    intArray[intVal] = -intVal;
    put_int(11,FORMAT_HEX);
    put_ascii(41);
    put_ascii(32);
    put_int(intVal,FORMAT_HEX);
    put_ascii(32);
    put_int(value,FORMAT_HEX);
    put_ascii(61);
    temp = _azimsec[chopperRotation][cc];
    put_int(temp,FORMAT_HEX);
    put_ascii(13);
    put_ascii(10);
}
}
wait_frame(0);
```
wait_frame(0);
intVal = 0xF80;

//********** _gyro **********//
for(chopperRotation=0; chopperRotation<64; chopperRotation =
    chopperRotation + 1)
{
    if(chopperRotation%4 == 0)
        wait_frame(0);
    for(cc=0; cc<4; cc = cc + 1)
    {
        intVal = intVal + 1;
        value = ((chopperRotation/8)<<12) +
            (((chopperRotation)%8)<<4) + cc;
        if(_gyro[chopperRotation][cc] == value)
            intArray[intVal] = intVal;
        else
        {
            intArray[intVal] = -intVal;
            put_int(12, FORMAT_HEX);
            put_ascii(41);
            put_ascii(32);
            put_int(intVal, FORMAT_HEX);
            put_ascii(32);
            put_int(value, FORMAT_HEX);
            put_ascii(61);
            temp = _gyro[chopperRotation][cc];
            put_int(temp, FORMAT_HEX);
            put_ascii(13);
            put_ascii(10);
        }
    }
}

intVal = 0x1180;
taskID = whoami();
put_ascii(taskID + 48);
put_ascii(13);
put_ascii(10);
suspend(taskID);

9.1.4.5.2 SAIL TASK SST010.sai

// SST010.sal Test each of the following communication variables
/_command
len 0 0 0 33  // fill until 0 - fac2

158
#include <stdio.h>

int packet_count, sci_packet_count, error_last_time, error_location, error_tag, error_type, mif, fshared, shared, maf, current_cr, hirdls_time, spacecraft_time;

// Fill this array with 0x23

// Fill this array with 0x24

// Fill this array with 0x25

// Fill this array with 0x26

int main(void)
{
    int intVal;
    int index;
    int intArray[0x400];
    float floatArray[8];
    int taskID;
    float floatVal = 1.2345E-67;

    int packet_count;
    int mif;
    int maf;
    int current_cr;
    float hirdls_time;
    float spacecraft_time;
    unsigned time_code[4];

    while(TRUE)
    {
        intVal = 0;

        //********** _command **********//
        for(index=0;index<0x20;index = index + 1)
        {
            _command[index] = intVal;
            intVal = intVal + 1; // 0x19
        }

        for(index=0x20;index>0;index = index - 1)
        {
            intVal = intVal - 1;
            intArray[intVal] = _command[index - 1];
        }
    }

    return(0);
}
intVal = 0x20; // 0x20;

//********** _eng_packet_count **********//
packet_count = _eng_packet_count;
wait_frame(6);
if(_eng_packet_count > packet_count)
  intArray[intVal] = intVal;

intVal = 0x21;

//********** _sci_packet_count **********//
packet_count = _sci_packet_count;
wait_frame(6);
if(_sci_packet_count > packet_count)
  intArray[intVal] = intVal;

intVal = 0x22;

//********** _error_XXXXX **********//
_error_last_time = intVal;
intVal = intVal + 1; // 0x23
_error_location = intVal;
intVal = intVal + 1; // 0x24
_error_tag = intVal;
intVal = intVal + 1; // 0x25
_error_type = intVal;
intArray[intVal] = _error_type;
intVal = intVal - 1;
intArray[intVal] = _error_tag;
intVal = intVal - 1;
intArray[intVal] = _error_location;
intVal = intVal - 1;
intArray[intVal] = _error_last_time;
intVal = 0x26; // 0x26;

// ********** _mif is read only **********//
mif = _mif;
wait_frame(1);
if((_mif >= mif + 8) && (_mif <= mif + 16))
  intArray[intVal] = intVal;

intVal = 0x27; // 0x27

// _fshared and _shared are at the end of this file

//********** _maf is read only **********//
maf = _maf;
wait_frame(0);
if(_maf == maf + 1)
  intArray[intVal] = intVal;

// _cr_count is read only @@@@@
// _hirdls_time_stamp is read only @@@@@
// _spu_qhss_state is read only @@@@@
// _spu_phase_state is read only @@@@@
// _radiance is read only @@@@@
// _tss_state is read only @@@@@
// _elev1prim is read only @@@@@

}
```c
for(index=0;index<0x31E;index = index + 1)
{
    _hk_array[index] = intVal;
    intArray[intVal] = _hk_array[index];
    intVal = intVal + 1; // 0x346
    if(intVal%100 == 0)
    {
        put_int(intVal,0);
        put_ascii(13);
        put_ascii(10);
        wait_frame(0);
    }
}
intVal = intVal + 1;
```

```c
#define _hirdls_time _hirdls_time
#define _current_cr _current_cr;

if((_current_cr >= current_cr + 64) && (_current_cr <= current_cr + 128))
    intArray[intVal] = intVal; // 0x28
intVal = 0x29; // 0x29
```

```c
#define _cmd_received _cmd_received
#define _cmd_rejected _cmd_rejected
#define _cmd_result _cmd_result
#define _cmd_number _cmd_number
#define _executed_cmd_pktseq _executed_cmd_pktseq
#define _qued_cmd_pktseq _qued_cmd_pktseq
```

```c
#define _hirdls_time _hirdls_time
#define _current_cr _current_cr;

if(_hirdls_time > hirdls_time)
    intArray[intVal] = intVal; // 0x29
else
{
    put_int(1,FORMAT_DEC);
    put_ascii(41);
    put_ascii(32);
```
intVal = 0x2A; // 0x2A

//********** _spacecraft_time is read only **********//
spacecraft_time = _spacecraft_time;
wait_frame(1);
if(_spacecraft_time > spacecraft_time)
intArray[intVal] = intVal; // 0x2A
else{
    put_int(2,FORMAT_DEC);
    put_ascii(41);
    put_ascii(32);
    put_float(spacecraft_time,FORMAT_SCI_FLOAT);
    put_ascii(61);
    spacecraft_time = _spacecraft_time;
    put_float(spacecraft_time,FORMAT_SCI_FLOAT);
    put_ascii(13);
    put_ascii(10);
}

intVal = 0x2B; // 0x2B

// _time_code is read only not checked
taskID = whoami();

//********** _shared **********//
for(index=0;index<0x10;index = index + 1)
    _shared[(taskID*16) + index] = intVal;
intVal = intVal + 1; // 0x3B
for(index=0x10;index>0;index = index - 1)
    intVal = intVal - 1;
    intArray[intVal] = _shared[(taskID*16) + index - 1];

//suspend(taskID);

//********** _fshared **********//
for(index=0;index<0x8;index = index + 1)
    _fshared[(taskID*8) + index] = floatVal;
    floatArray[index] = _fshared[(taskID*8) + index];

put_ascii(taskID + 48);
put_ascii(13);
9.1.4.6 SEXEC Test Case

9.1.4.6.1 SAIL TASK SEXEC000.sai

// SEXEC000.sai test that SAIL can handle 16 tasks
// SAIL configuration with the setup of tasks or waiting
// and the number of runs increase up to 16
// $Id: SEXEC000.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
/*
* $Log: SEXEC000.sai $
* Revision 1.1  2002/05/07 07:28:31  jpriley
* Initial revision
*
*/

void main(void) {
    int intVal = 1;
    int taskID;
    int taskVal;
    int runCount = 0;
    taskVal = 0;
    while (TRUE) {
        taskID = whoami();
        taskVal = 0;
        put_int(taskVal, FORMAT_DEC);
        put_ascii(44);
        put_int(taskID, FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if (runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}
9.1.4.6.2 SAIL TASK SEXEC001.sai

// SEXEC001.sai test that SAIL can handle 16 tasks
// Compile, assemble, load, run and as task 1
// SAIL display should show the tasks as ready or waiting
// and the number of runs increase up to 15

void main(void)
{
    int intVal = 1;
    int taskID;
    int taskVal;
    int runCount = 0;

    taskVal = 1;
    while(TRUE)
    {
        taskID = whoami();
        taskVal = 1;
        put_int(taskVal, FORMAT_DEC);
        put_ascii(44);
        put_int(taskID, FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if(runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}

9.1.4.6.3 SAIL TASK SEXEC002.sai

// SEXEC002.sai test that SAIL can handle 16 tasks
// Compile, assemble, load, run and as task 2
// SAIL display should show the tasks as ready or waiting
// and the number of runs increase up to 15

void main(void)
{
    int intVal = 1;
    int taskID;
    int taskVal;
    int runCount = 0;

    taskVal = 1;
    while(TRUE)
    {
        taskID = whoami();
        taskVal = 1;
        put_int(taskVal, FORMAT_DEC);
        put_ascii(44);
        put_int(taskID, FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if(runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}
```c
#include <stdio.h>
#include <stdlib.h>

void main(void) {
    int intVal = 3;
    int taskID;
    int taskVal;
    int runCount = 0;
    taskVal = 3;
    while(TRUE) {
        taskID = whoami();
        taskVal = 3;
        printf(taskVal, 3);
        put_ascii(44);
        put_int(taskID, FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if(runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}

9.1.4.6.4 SAIL TASK SEXEC003.sai

//  SEXEC003.sal test that SAIL can handle 16 tasks
//  Compile, assemble, load, run and as task 3
//  SAIL display should show the tasks as ready or waiting
//  and the number of runs increase up to 15
//  $Id: SEXEC003.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
// $Log: SEXEC003.sai $
// Revision 1.1  2002/05/07 07:28:31  jpriley
// Initial revision
```

```c
#include <stdio.h>
#include <stdlib.h>

void main(void) {
    int intVal = 3;
    int taskID;
    int taskVal;
    int runCount = 0;
    taskVal = 3;
    while(TRUE) {
        taskID = whoami();
        taskVal = 3;
        printf(taskVal, 3);
        put_ascii(44);
        put_int(taskID, FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if(runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}

9.1.4.6.4 SAIL TASK SEXEC003.sai

//  SEXEC003.sal test that SAIL can handle 16 tasks
//  Compile, assemble, load, run and as task 3
//  SAIL display should show the tasks as ready or waiting
//  and the number of runs increase up to 15
//  $Id: SEXEC003.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
// $Log: SEXEC003.sai $
// Revision 1.1  2002/05/07 07:28:31  jpriley
// Initial revision
```
void main(void)
{
    int intVal = 4;
    int taskID;
    int taskVal;
    int runCount = 0;
    taskVal = 4;
    while(TRUE) {
        taskID = whoami();
        taskVal = 4;
        put_int(taskVal,FORMAT_DEC);
        put_ascii(44);
        put_int(taskID,FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if(runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}
9.1.4.6 SAIL TASK SEXEC005.sai
// SEXEC005.sal test that SAIL can handle 16 tasks
// Compile, assemble, load, run and as task 5
// SAIL display should show the tasks as ready or waiting
// and the number of runs increase up to 15
// File: SEXEC005.sai 1.1 2002/05/07 07:28:31 jpriley

void main(void)
{
    int intVal = 5;
    int taskID;
    int taskVal;
    int runCount = 0;
    taskVal = 5;
    while(TRUE)
    {
        taskID = whoami();
        taskVal = 5;
        put_int(taskVal,FORMAT_DEC);
        put_ascii(44);
        put_int(taskID,FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if(runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}

9.1.4.7 SAIL TASK SEXEC006.sai
// SEXEC006.sal test that SAIL can handle 16 tasks
// Compile, assemble, load, run and as task 6
// SAIL display should show the tasks as ready or waiting

void main(void)
{
  int taskID;
  int taskVal;
  int runCount = 0;
  taskVal = 6;
  while(TRUE)
  {
    taskID = whoami();
    taskVal = 6;
    put_int(taskVal,FORMAT_DEC);
    put_ascii(44);
    put_int(taskID,FORMAT_DEC);
    put_ascii(13);
    put_ascii(10);
    if(runCount < 16)
      wait_frame(0);
    else
      suspend(taskID);
    runCount = runCount + 1;
  }
}

9.1.4.6.8 SAIL TASK SEXEC007.sai

void main(void)

9.1.4.6.8 SAIL TASK SEXEC007.sai

int intVal = 7;
int taskID;
int taskVal;
int runCount = 0;

while(TRUE)
{
    taskID = whoami();
    taskVal = 7;
    put_int(taskVal, FORMAT_DEC);
    put_ascii(44);
    put_int(taskID, FORMAT_DEC);
    put_ascii(13);
    put_ascii(10);
    if(runCount < 16)
    {
        wait_frame(0);
    }
    else
    {
        suspend(taskID);
        runCount = runCount + 1;
    }
}

9.1.4.6.9 SAIL TASK SEXEC008.sai

// SEXEC008.sal test that SAIL can handle 16 tasks
// Compile, assemble, load, run and as task 8
// SAIL display should show the tasks as ready or waiting
// and the number of runs increase up to 15

void main(void)
{
    int intVal = 8;
    int taskID;
    int taskVal;
    int runCount = 0;
    taskVal = 8;
    while(TRUE)
    {
        taskID = whoami();
        taskVal = 8;
        put_int(taskVal, FORMAT_DEC);
    }
}

9.1.4.6.10 SAIL TASK SEXEC009

// SEXEC009.sal test that SAIL can handle 16 tasks
// compile, assemble, load, run and as task 9
// SAIL display should show the tasks as ready or waiting
// file SAIL009.sal 1.1 2002/05/07 07:38:31 jpriley Sal

/*
 * $Log: SEXEC009.sai $
 * Revision 1.1  2002/05/07 07:28:31  jpriley
 * Initial revision
 */

//

void main(void)
{
    int intVal = 9;
    int taskID;
    int taskVal;
    int runCount = 0;
    taskVal = 9;
    while(TRUE)
    {
        taskID = whoami();
        taskVal = 9;
        put_int(taskVal,FORMAT_DEC);
        put_ascii(44);
        put_int(taskID,FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if(runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}
9.1.4.6.11 SAIL TASK SEXEC010.sai

// SEXEC010.sai tests that SAIL can handle 16 tasks
// Compiles, assembles, load, run and as task 10
// SAIL display should show the tasks as ready or waiting
// Also, SEXEC010 displays the number of runs on the screen

/* $Id: SEXEC010.sai 1.1 2002/05/07 07:28:31 jpriley $ */

void main(void)
{
    int intVal = 10;
    int taskID;
    int taskVal;
    int runCount = 0;

    taskID = whoami();
    taskVal = 10;
    put_int(taskVal,FORMAT_DEC);
    put_ascii(44);
    put_int(taskID,FORMAT_DEC);
    put_ascii(13);
    put_ascii(10);
    if(runCount < 16)
        wait_frame(0);
    else
        suspend(taskID);
    runCount = runCount + 1;
}

9.1.4.6.12 SAIL TASK SEXEC011.sai

// SEXEC011.sai tests that SAIL can handle 16 tasks
// Compiles, assembles, load, run and as task 11
// SAIL display should show the tasks as ready or waiting
// Also, SEXEC011 displays the number of runs on the screen

/* $Id: SEXEC011.sai 1.1 2002/05/07 07:28:31 jpriley $ */

void main(void)
{
    int intVal = 10;
    int taskID;
    int taskVal;
    int runCount = 0;

    taskID = whoami();
    taskVal = 10;
    put_int(taskVal,FORMAT_DEC);
    put_ascii(44);
    put_int(taskID,FORMAT_DEC);
    put_ascii(13);
    put_ascii(10);
    if(runCount < 16)
        wait_frame(0);
    else
        suspend(taskID);
    runCount = runCount + 1;
}
void main(void)
{
    int taskVal = 11;
    int taskID;
    int runCount = 0;

    taskID = whoami();
    taskVal = 11;
    put_int(taskVal, FORMAT_DEC);
    put_ascii(44);
    put_int(taskID, FORMAT_DEC);
    put_ascii(13);
    put_ascii(10);
    if (runCount < 16)
        wait_frame(0);
    else
        suspend(taskID);
    runCount = runCount + 1;
}

9.1.4.6.13 SAIL TASK SEXEC012.sai

void main(void)
{
    int intVal = 12;
    int taskID;
    int taskVal;
    int runCount = 0;

    taskID = whoami();
    taskVal = 11;
    put_int(taskVal, FORMAT_DEC);
    put_ascii(44);
    put_int(taskID, FORMAT_DEC);
    put_ascii(13);
    put_ascii(10);
    if (runCount < 16)
        wait_frame(0);
    else
        suspend(taskID);
    runCount = runCount + 1;
}
```c
int main(void) {
    int taskID;
    int taskVal;
    int runCount = 0;
    taskVal = 13;
    while(TRUE) {
        taskID = whoami();
        taskVal = 13;
        put_int(taskVal, FORMAT_DEC);
        put_ascii(44);
        put_int(taskID, FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if (runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}
```

9.1.4.6.14 SAIL TASK SEXEC013.sai

// SEXEC013.sal test that SAIL can handle 16 tasks
// SAIL will run 16 tasks, 16 at a time
// If SAIL stops, it should give the tasks an error or warning
// and the number of runs will increase up to 16
// File: SEXEC013.sal 1.1 2002/04/17 07:31:41 Getting 1

/*
 * Revision 1.1  2002/04/17 07:31:41  Getting 1
 * Initial revision
 */

```
9.1.4.16 SAIL TASK SEXEC016.sai

// SEXEC016.sai test that SAIL can handle 16 tasks
// Compile, assemble, load, run and as task 14
// SAIL display should show the tasks as ready or waiting
// and the number of runs increase up to 15

* $Id: SEXEC016.sai 1.1 2002/05/07 07:28:31 jpriley Exp *
* $Log: SEXEC016.sai *
* Revision 1.1  2002/05/07 07:28:31  jpriley
* Initial revision
* *
*
*
*
#endif

void main(void)
{
    int taskID;
    int taskVal;
    int runCount = 0;
    while(TRUE)
    {
        taskID = whoami();
        taskVal = 14;
        put_int(taskVal,FORMAT_DEC);
        put_ascii(44);
        put_int(taskID,FORMAT_DEC);
        put_ascii(10);
        put_ascii(13);
        if(runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}

9.1.4.16 SAIL TASK SEXEC016.sai

174
void main(void)
{
  int intVal = 15;
  int taskID;
  int taskVal;
  int runCount = 0;
  taskVal = 15;
  while(TRUE)
  {
    taskID = whoami();
    taskVal = 15;
    put_int(taskVal,FORMAT_DEC);
    put_ascii(44);
    put_int(taskID,FORMAT_DEC);
    put_ascii(13);
    put_ascii(10);
    if(runCount < 16)
      wait_frame(0);
    else
      suspend(taskID);
    runCount = runCount + 1;
  }
}

9.1.4.6.17 SAIL TASK SEXEC016.sal
// SEXEC016.sal test that SAIL can handle a reserve tasks
// Compile, assemble and load into memory locations 20 for code and
// 144 for data.
// With 16 task loaded and running kill task 1 and creat a new task
//              SAIL display should show the task as ready or waiting
//              and the number of runs increase up to 15
// $Id: SEXEC016.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
void main(void)
{
    int intVal = 1;
    int taskID;
    int taskVal;
    int runCount = 0;

    taskVal = 1;
    while(TRUE)
    {
        taskID = whoami();
        taskVal = 1;
        put_int(taskVal, FORMAT_DEC);
        put_ascii(44);
        put_int(16, FORMAT_DEC);
        put_ascii(13);
        put_ascii(10);
        if(runCount < 16)
            wait_frame(0);
        else
            suspend(taskID);
        runCount = runCount + 1;
    }
}

9.1.4.6.18 SAIL TASK SEXEC020.sai

// SEXEC020.sal load sail shared memory and Engineering data
// Compiler, assembler, and all tasks 1 - 15, run all 16 tasks
// Check SAIL parameters in diagnostic class 0 from 0 to 0xFF
// Check all SAIL shared memory + address and engineering + 8
//
// $Id: SEXEC020.sai 1.1 2002/05/07 07:30:51 jpriley Exp $
// $Log: SEXEC020.sai $
// Revision 1.1  2002/05/07 07:30:51  jpriley
// Initial revision

void main(void)
{
    int intVal = 1;
    int taskID;
    int taskVal;
    int index;
    taskVal = 0;
}
9.1.4.7 SFERR Test Case

9.1.4.7.1 SAIL TASK SFERR010.sai

// SFERR010.c runs a task which delivers two SFERRs
// and kills the task

/* $Id: SFERR010.sai 1.2 2002/06/16 12:17:42 jpriley Exp $ */
/* $Log: SFERR010.sai $ */
/* Revision 1.2 2002/06/16 12:17:42 jpriley */
/* Removed obsolete error function prototypes. */
/* Revision 1.1 2002/05/07 07:28:31 jpriley */

int intVal;
int errorType;
int ERR_Dual_Errors = -19;
int ERR_No_Resume = -20;
int ERR_Call_During_Error = -21;
int ERR_Ret_During_Error = -22;
int ERR_Error_Kill = -23;

void function(void);

void main(void)
{
    intVal = 1;
    int taskID;
    int errorVal;
    while(TRUE)
    {
        ...
        ...
    }
}
errorType = ERR_Dual_Errors;         // "Multiple Errors" OK
errorVal = 1/0;
errorType = ERR_No_Resume;           // "Can Not Resume" OK
// errorVal = 1/0;
errorType = ERR_Ret_During_Error;    // "Return in (error)" OK
// errorVal = 1/0;
errorType = ERR_Error_Kill;          // "Task Error Killed" OK
// errorVal = 1/0;
// the function call must be added to the assembly code
errorType = ERR_Call_During_Error;   // "Call in (error)" OK
// errorVal = 1/0;
// check error output from sail
suspend(taskID);
put_ascii(taskID + 48);
put_ascii(13);
put_ascii(10);
}
}

void function(void)
{

}

void error(void)
{
int errorVal;
if(_error_last_time == _error_location)
    error_kill();
if(_error_type == ERR_Divide_By_0)
{
    if(errorType == ERR_Dual_Errors)
        error_retry();
    if(errorType == ERR_No_Resume)
        errorVal = 1/0;
    if(errorType == ERR_Call_During_Error)
        // function(); // must be added in assembler "CALL function"
    if(errorType == ERR_Ret_During_Error)
        return;
    if(errorType == ERR_Error_Kill)
        error_kill(); // kill the task, error is repeating
    if(errorType == ERR_Call_During_Error)
        // kill the task, error is repeating
    initial = initial + 1;
    error_ignore();
}

9.1.4.7.2 SAIL TASK SFERR020.sai
int intVal;
int errorType;
int ERR_Dual_Errors = -19;
int ERR_No_Resume = -20;
int ERR_Call_During_Error = -21;
int ERR_Ret_During_Error = -22;
int ERR_Error_Kill = -23;

void function(void);

void main(void)
{
    intVal = 1;
    int taskID;
    int errorVal;
    
    while(TRUE)
    {
        // generate error to be handled by the error function
        // only one of the error can be generated at a time
        errorType = ERR_Dual_Errors;         // "Multiple Errors" OK
        // errorVal = 1/0;
        
        errorType = ERR_No_Resume;           // "Can Not Resume" ?
        errorVal = 1/0;
        
        errorType = ERR_Ret_During_Error;    // "Return in (error)" OK
        // errorVal = 1/0;
        errorType = ERR_Error_Kill;          // "Task Error Killed" OK
        // errorVal = 1/0;
        
        errorType = ERR_Call_During_Error;   // "Call in (error)" OK
        //errorVal = 1/0;
        
        // check error output from SAIL
        taskID = whoami();
        put_ascii(taskID + 48);
        put_ascii(13);
        put_ascii(10);
        suspend(taskID);
    }
}

void function(void)
```c
void error(void)
{
    int errorVal;
    if (_error_last_time == _error_location)
        error_kill();
    if (_error_type == ERR_Divide_By_0)
    {
        if(errorType == ERR_Dual_Errors)
            error_kill();
        if(errorType == ERR_No_Resume)
            errorVal = 1/0;
        if(errorType == ERR_Call_During_Error)
            // function(); // must be added in assembler "CALL function"
        if(errorType == ERR_Ret_During_Error)
            return;
        if(errorType == ERR_Error_Kill)
            error_kill(); // kill the task, error is repeating
    }
    error_ignore();
}

9.1.4.7.3 SAIL TASK SFERR030.sai
// SFERR030.c test SAIL user defined error function
// compiled, assembled, test and run, generates a SE_ERR_Divide_By_0, a SE_ERR_Call_During_Error and kills the task
// $Id: SFERR030.sai 1.2 2002/06/16 12:17:42 jpriley Exp $
/* $Log: SFERR030.sai $ */
/* Revision 1.2  2002/06/16 12:17:42  jpriley */
/* Removed obsolete error function prototype. */
/* Revision 1.1  2002/05/07 07:28:31  jpriley */
/* Initial revision */
/
int intVal;
int errorType;
int ERR_Dual_Errors = -19;
int ERR_No_Resume = -20;
int ERR_Call_During_Error = -21;
int ERR_Ret_During_Error = -22;
int ERR_Error_Kill = -23;
void function(void);
```

void main(void)
{
  intVal = 1;
  int taskID;
  int errorVal;
  while(TRUE)
  {
    // generate error to be handled by the error function
    // only one of the error can be generated at a time
    errorType = ERR_Dual_Errors; // "Multiple Errors" OK
    errorVal = 1/0;
    errorType = ERR_No_Resume; // "Can Not Resume" OK
    errorVal = 1/0;
    errorType = ERR_Ret_During_Error; // "Return in (error)" OK
    errorVal = 1/0;
    errorType = ERR_Error_Kill; // "Task Error Killed" OK
    errorVal = 1/0;
    // check error output from SAIL, // must be added in assembler "CALL function"
    taskID = whoami();
    put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
    suspend(taskID);
    // check intArray
  }
}

void function(void)
{
}

void error(void)
{
  int errorVal;
  if(_error_last_time == _error_location)
    error_kill();
  if(_error_type == ERR_Divide_By_0)
    {
      if(errorType == ERR_Dual_Errors)
        error_retry();
      if(errorType == ERR_No_Resume)
        errorVal = 1/0;
      if(errorType == ERR_Call_During_Error)
        return;
      if(errorType == ERR_Ret_During_Error)
        return;
      if(errorType == ERR_Error_Kill)
        error_kill(); // kill the task, error is repeating
    }
}
intVal = intVal + 1;
error_ignore();

9.1.4.7.4 SAIL TASK SFERR040.sai

// SFERR040.c test SAIL user defined error function
// Compile, assemble, load and run, generates a SE_ERR_Divide_By_0, a
// ERR_Error_Kill and kills the task
// $Id: SFERR040.sai 1.2 2002/06/16 12:17:42 jpriley Exp $
/*
* $Log: SFERR040.sai $
* Revision 1.2  2002/06/16 12:17:42  jpriley
* Removed obsolete error function prototype.
* 
* Revision 1.1  2002/05/07 07:28:31  jpriley
* Initial revision
* $ */

int intVal;
int errorType;

int ERR_Dual_Errors = -19;
int ERR_No_Resume = -20;
int ERR_Call_During_Error = -21;
int ERR_Ret_During_Error = -22;
int ERR_Error_Kill = -23;

void function(void);

void main(void)
{
    intVal = 1;
    int taskID;
    int errorVal;
    while(TRUE)
    {
        // generate error to be handled by the error function
        // ensure that the macro ERR_No_Resume (or generation of same) does not
        // get included twice
        // errorVal = 1/0;
        errorType = ERR_Dual_Errors;         // "Multiple Errors" OK
        // errorVal = 1/0;
        errorType = ERR_No_Resume;           // "Can Not Resume" OK
        // errorVal = 1/0;
        errorType = ERR_Ret_During_Error;    // "Return in (error)
        // errorVal = 1/0;
        errorType = ERR_Error_Kill;          // "Task Error Killed" OK
        // errorVal = 1/0;
        errorType = ERR_Call_During_Error;   // "Call in (error)
        // check error output from SAIL
    }
}

9.1.4.7.5 SAIL TASK SFERR050.sai

// SFERR050 task (SAIL) user defined error function
// SFERR050.c test SAIL user defined error function
// Compile, assemble, load and run, generates a SE_ERR_Divide_By_0, a
// SE_ERR_Call_During_Error and SE_ERR_Error_Kill
// $Id: SFERR050.sai 1.2 2002/06/16 12:17:42 jpriley Exp $
// $Log: SFERR050.sai $
// Revision 1.2  2002/06/16 12:17:42  jpriley
// Removed obsolete error function prototype.
// Revision 1.1  2002/05/07 07:28:31  jpriley
// Initial revision

// SFERR050.sai

// SFERR050 task (SAIL) user defined error function
// SFERR050.c test SAIL user defined error function
// Compile, assemble, load and run, generates a SE_ERR_Divide_By_0, a
// SE_ERR_Call_During_Error and SE_ERR_Error_Kill
// $Id: SFERR050.sai 1.2 2002/06/16 12:17:42 jpriley Exp $
// $Log: SFERR050.sai $
// Revision 1.2  2002/06/16 12:17:42  jpriley
// Removed obsolete error function prototype.
// Revision 1.1  2002/05/07 07:28:31  jpriley
// Initial revision

//
int intVal;
int errorType;
int ERR_Dual_Errors = -19;
int ERR_No_Resume = -20;
int ERR_Call_During_Error = -21;
int ERR_Ret_During_Error = -22;
int ERR_Error_Kill = -23;

void function(void);

void main(void)
{
    intVal = 1;
    int taskID;
    int errorVal;

    while(TRUE)
    {
        // generate error to be handled by the error function
        // only one of the error can be generated at a time
        errorType = ERR_Dual_Errors;         // "Multiple Errors" OK
        // errorVal = 1/0;
        errorType = ERR_No_Resume;           // "Can Not Resume" ?
        //errorVal = 1/0;
        errorType = ERR_Ret_During_Error;    // "Return in (error)" OK
        //errorVal = 1/0;
        errorType = ERR_Error_Kill;          // "Task Error Killed" OK
        // errorVal = 1/0;
        // the function call must be added to the assemble code
        errorType = ERR_Call_During_Error;   // "Call in (error)" OK
        errorVal = 1/0;

        // check error output from SAIL
        taskID = whoami();
        put_ascii(taskID + 48);
        put_ascii(13);
        put_ascii(10);
        suspend(taskID);
    }
}

void function(void)
{
}

void error(void)
{
    int errorVal;
    if(_error_last_time == _error_location)
        error_kill();
    if(_error_type == ERR_Divide_By_0)
    {
        if(errorType == ERR_Dual_Errors)
            error_retry();
    }
    // errorVal = 1/0;
    // put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
}
if(errorType == ERR_No_Resume)
    errorVal = 1/0;
if(errorType == ERR_Call_During_Error)
    // function(); // must be added in assembler "CALL function"
if(errorType == ERR_Ret_During_Error)
    return;
if(errorType == ERR_Error_Kill)
    error_kill(); // kill the task, error is repeating

intVal = intVal + 1;
error_ignore();

9.1.4.8 SFUN Test Case

9.1.4.8.1 SAIL TASK SFUN010.sai

// SFUN010.c test the running of a function including passing perimeters and
getting a return value

// SFUN010.sai 1.1 2002/05/07 07:28:31 jpriley

foo int intVal = 1;
intArray[100];

// functions passing and retruning each type
logical functionLogical(logical logicalVal);
void functionVoid(void);
byte functionByte(byte byteVal);
unsigned functionUnsigned(unsigned unsignedVal);
int functionInt(int intValue);
float functionFloat(float floatVal);
void functionArray(int arrayVal[4][3]);

// functions passing by reference for each type
void functionLogicalReff(logical &logicalVal);
void functionByteReff(byte &byteVal);
void functionUnsignedReff(unsigned &unsignedVal);
void functionIntReff(int &intValue);
void functionFloatReff(float &floatVal);

// function with multiple peramiters
void functionMultiple(logical logicalVal, byte byteVal, unsigned unsignedVal, int passIntVal,
                      void (int arrayVal[4][3]);

foo
// Uses the passed value and ref in different ways
void functionUsing(int passIntVal, int &reffIntVal, int arrayVal[4][3]);

void main(void)
{
    logical logicalVal = TRUE;
    byte byteVal;
    unsigned unsignedVal;
    int passIntVal;
    float floatVal = 0.123;
    int taskID;
    int arrayVal[4][3];
    intVal = 1;
    while(TRUE)
    {
        intArray[intVal] = intVal;
        intVal = intVal + 1; // 0x02
        intArray[intVal] = -intVal;
        if(functionLogical(logicalVal) == FALSE) // 0x03
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x04
        intArray[intVal] = -intVal;
        if(functionLogical(FALSE) == TRUE) // 0x05
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x06
        functionVoid(); // 0x07
        byteVal = intVal - 1;
        intArray[intVal] = functionByte(byteVal);
        intVal = intVal + 1; // 0x08
        unsignedVal = intVal - 1;
        intArray[intVal] = functionUnsigned(unsignedVal);
        intVal = intVal + 1; // 0x09
        passIntVal = intVal - 1;
        intArray[intVal] = functionInt(passIntVal);
        // Test that passIntVal did not change
        intVal = intVal + 1; // 0x0A
        if(passIntVal == intVal - 2)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x0B
        floatVal = floatVal + (intVal - 1);
        intArray[intVal] = functionFloat(floatVal);
        intVal = intVal + 1; // 0x0C
        arrayVal[2][1] = intVal - 1;
        functionArray(arrayVal);
    }
}
intArray[intVal] = arrayVal[1][2];  // pass by reference
intVal = intVal + 1;                 // 0x0D
logicalVal = TRUE;
functionLogicalReff(logicalVal);
if(logicalVal == FALSE)
  intArray[intVal] = intVal;
else
  intArray[intVal] = -intVal;
intVal = intVal + 1;                 // 0x0E
byteVal = intVal - 1;
functionByteReff(byteVal);
intArray[intVal] = byteVal;
intVal = intVal + 1;                 // 0x0F
unsignedVal = intVal - 1;
functionUnsignedReff(unsignedVal);
intArray[intVal] = unsignedVal;
intVal = intVal + 1;                 // 0x10
passIntVal = intVal - 1;
functionIntReff(passIntVal);
intArray[intVal] = passIntVal;
intVal = intVal + 1;                 // 0x11
floatVal = 0.123 + (intVal - 1);
functionFloatReff(floatVal);
intArray[intVal] = floatVal;
// function with multiple parameters
intVal = intVal + 1;                 // 0x12
functionMultiple(logicalVal, byteVal, unsignedVal, passIntVal, floatVal, arrayVal);
arrayVal[1][1] = 0;
arrayVal[2][2] = 0;
intVal = intVal + 1;                 // 0x13
passIntVal = intVal;
functionUsing(passIntVal, passIntVal, arrayVal);
if(arrayVal[1][1] == 0x14)
  intArray[intVal] = intVal;
else
  intArray[intVal] = -intVal;
intVal = intVal + 1;                 // 0x16
if(arrayVal[2][2] == 0x15)
  intArray[intVal] = intVal;
else
  intArray[intVal] = -intVal;
taskID = whoami();
put_ascii(taskID + 48);
put_ascii(13);
put_ascii(10);
suspend(taskID);
TR-LOC-982C

Logical function

```
Logical(logicalVal)
{
    intArray[intVal] = intVal;
    intVal = intVal + 1;
    if(logicalVal == TRUE)
        return FALSE;
    return TRUE;
}
```

Void function

```
Void(void)
{
    intArray[intVal] = intVal;
    intVal = intVal + 1;
}
```

Byte function

```
Byte(byte byteVal)
{
    return(byteVal + 1);
}
```

Unsigned function

```
Unsigned(unsigned unsignedVal)
{
    return(unsignedVal + 1);
}
```

Int function

```
Int(int intValue)
{
    intValue = intValue + 1;
    return intValue;
}
```

Float function

```
Float(float floatVal)
{
    return(floatVal + 1);
}
```

Void function

```
Array(int arrayVal[4][3])
{
    arrayVal[1][2] = arrayVal[2][1] + 1;
}
```

Void function

```
LogicalReff(logical &logicalVal)
{
    logicalVal = FALSE;
}
```

Void function

```
ByteReff(byte &byteVal)
{
    byteVal = byteVal + 1;
}
```

Void function

```
UnsignedReff(unsigned &unsignedVal)
{
    unsignedVal = unsignedVal + 1;
}
```

188
void functionIntReff(int &intValue)
{
    intValue = intValue + 1;
}

void functionFloatReff(float &floatVal)
{
    floatVal = floatVal + 1;
}

// function with multiple parameters
void functionMultiple(logical logicalVal,
byte byteVal,
unsigned unsignedVal,
int passIntVal,
float floatVal,
int arrayVal[4][3])
{
    intArray[0x20] = logicalVal + byteVal + unsignedVal + passIntVal +
arrayVal[1][2];
    intArray[0x21] = logicalVal;
    intArray[0x22] = byteVal;
    intArray[0x23] = unsignedVal;
    intArray[0x24] = passIntVal;
    intArray[0x25] = arrayVal[1][2];

    if(logicalVal + byteVal + unsignedVal + passIntVal +
arrayVal[1][2] == intVal*4 - (2 + 3 + 4 + 6))
    // 18*4 = 72
    intArray[intVal] = intVal;
    else
    intArray[intVal] = -intVal;
}

// uses the passed value and self in different ways
void functionUsing(int passIntVal, int &reffIntVal, int arrayVal[4][3])
{
    intVal = passIntVal; // 0x13
    intArray[passIntVal] = intVal;
    intVal = intVal + 1; // 0x14
    arrayVal[passIntVal - 0x12][passIntVal - 0x12] = intVal; // 1x1
    intArray[reffIntVal + 1] = intVal;
    intVal = intVal + 1; // 0x15
    arrayVal[reffIntVal - 0x11][reffIntVal - 0x11] = intVal; // 2x2
}

9.1.4.8.2 SAIL TASK SRTL010.sai

// SRTL010.sai test run time library functions
// Compile, assemble, load with purpose = 0, run and check intArray in memory saved - call

199
```c
#define MAX_MATH_ERROR 0.0000000000001

int intVal;
intArray[0x20];

int main(void)
{
    float z, x;
    byte b = 0x31;
    while(TRUE)
    {
        intVal = 1;
        z = atan2(1.0,1.0);
        x = PI/4.0;
        if(z >= x - MAX_MATH_ERROR && z <= x + MAX_MATH_ERROR)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x02
        a = abs(-2);
        if(a == 2)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x03
        z = fabs(-1.234);
        if(z >= 1.234 - MAX_MATH_ERROR && z <= 1.234 + MAX_MATH_ERROR)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x04
        z = sin(PI/2.0);
        if(z >= 1.0 - MAX_MATH_ERROR && z <= 1.0 + MAX_MATH_ERROR)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        put_int(1, FORMAT_DEC);
        put_ascii(41);
        put_ascii(32);
        put_float(z, FORMAT_SCI_FLOAT);
        put_ascii(61);
        put_float(x, FORMAT_SCI_FLOAT);
        put_ascii(13);
    }
}```
```c
put_ascii(10);
intVal = intVal + 1; // 0x05
z = cos(PI);
if(z >= -1.0 - MAX_MATH_ERROR && z <= -1.0 + MAX_MATH_ERROR)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x06
z = exp(1.0);
if(z >= x - MAX_MATH_ERROR && z <= x + MAX_MATH_ERROR)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x07
z = ln(1.6487212707001);
if(z >= 0.5 - MAX_MATH_ERROR && z <= 0.5 + MAX_MATH_ERROR)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x08
z = sqrt(2.0);
x = 1.414213562373;
if(z >= x - MAX_MATH_ERROR && z <= x + MAX_MATH_ERROR)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x09
z = asin(0.7071067811865);
x = PI/4.0;
if(z >= x - MAX_MATH_ERROR && z <= x + MAX_MATH_ERROR)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0A
z = acos(0.7071067811865);
x = PI/4.0;
if(z >= x - MAX_MATH_ERROR && z <= x + MAX_MATH_ERROR)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
```

else
    intArray[intVal] = -intVal;

    nop();
    intVal = intVal + 1; // 0x0C

    z = real(1);
    if(z >= 1.0 - MAX_MATH_ERROR && z <= 1.0 + MAX_MATH_ERROR)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;

    // send(); handled in SCOM010.sai
    intVal = intVal + 1; // 0x0D

    a = whoami();
    b = purpose(a);
    if(b == 0)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;

    put_ascii(a + 48);
    put_ascii(13);
    put_ascii(10);
    suspend(a);

9.1.4.9 SIERR Test Case

9.1.4.9.1 SAIL TASK SIERR010.sai

// SIERR010.sai test that sail interpreter handles error correctly
// Compile, assemble, load, run and check intArray in memory 0x00 - 0x06
// This task generates errors

*/

*/

*/
void main(void)
{
    intVal = 1;
    int taskID;
    int passIntVal = 2147483647;
    float floatVal = 1.797693134862315E+308;
    while(TRUE)
    {
        error_type = ERR_Domain;    // Domain Error -2,254
        floatVal = ln(-1);
        //intVal = intVal + 1; // 0x02
        error_type = ERR_Divide_By_0;    // Divide by 0 -3,253
        passIntVal = passIntVal/0;
        //intVal = intVal + 1; // 0x03
        error_type = ERR_Bad_Index;    // Bad Index -5,251
        intArray[0x11] = 1;
        //intVal = intVal + 1; // 0x04
        error_type = ERR_Mem_Read;   // Mem Read Error -7,249
        intArray[9] = read_mem(0x02000001);
        //intVal = intVal + 1; // 0x05
        error_type = ERR_Mem_Write;   // Mem Write Error -8,248
        _shared[0x18] = 1;
        //intVal = intVal + 1; // 0x06
        error_type = ERR_Bad_RTL;    // Bad RTL Call -6,250
        bad_RTL();
        //intVal = intVal + 1; // 0x07
        // this gives a compiler error
        error_type = ERR_comm_write;   // Write to a read only
        // communication variable
        _maf = 1;
        //intVal = intVal + 1; // 0x08
        // this will cause a "Over Time"
        error_type = ERR_Over_Time;   // Over Time -18,238
        // any code following this task
        // should generate a "Not Run"
        taskID = whoami();
        put_ascii(taskID + 48);
        put_ascii(13);
        put_ascii(10);
        suspend(taskID);
    }
}

void error(void)
{
    int local_error_type;
}
local_error_type = error_type;
put_int(local_error_type, FORMAT_DEC);
if (_error_last_time == _error_location)
    error_kill();
if (_error_type == error_type)
    intArray[intVal] = intVal;
else
    intArray[intVal] = _error_type;
    intVal = intVal + 1;
error_ignore();
}

9.1.4.10 SMEM Test Case

9.1.4.10.1 SAIL TASK SMEM010.sai

// SMEM010.sai is a sail test program
//  it write 0x12345678 - index to 0x10 memory location;
//  it write 0x12345678 - index to 0x10 shared memory location;
//  IP pointer commented out must know location and input it before
//  the compile
//  it write 0x12345678 - index to one IP memory location;
//  Check SAIL memory above
//  Check intArray[0x01 - 0x40]

#define myvariable 0x12345678
void main(void)
{
    int MemoryArray[0x10];
    int index;
    int taskID;
    int location;
    int intVal;
    int intArray[100];
    while(TRUE)
    {
        intVal = 0;
        taskID = whoami();
        /**********************************************************/
        // write info
        // this task's write area
        for (index = 0; index<0x10; index = index + 1)
        {
            _shared[(taskID*16) + index] = myvariable - index;
        }
        /**********************************************************/
        /**********************************************************/
        // write info
        // this task's write area
        for (index = 0; index<0x10; index = index + 1)
        {
            _shared[(taskID*16) + index] = myvariable - index;
        }
    }
}
for (index = 0; index<0x10; index = index + 1)
{
    _shared[(16*16) + index] = myvariable - index;
}

for (index = 0; index<0x10; index = index + 1)
{
    _command[index] = myvariable - index;
}

for (index = 0; index<0x10; index = index + 1)
{
    MemoryArray[index] = myvariable - index;
}

/*/ location = 0x51a488;
active_base(location,myvariable);
*/

/***********************************************************/
// read info;
// this task's write area
for (index = 0; index<0x10; index = index + 1)
{
    intVal = intVal + 1; // 0x01 - 0x10
    if(_shared[(taskID*16) + index] == myvariable - index)
    {
        intArray[intVal] = intVal;
    }
    else
    {
        intArray[intVal] = -intVal;
    }
}

// any task write area
for (index = 0; index<0x10; index = index + 1)
{
    intVal = intVal + 1; // 0x11 - 0x20
    if(_shared[(16*16) + index] == myvariable - index)
    {
        intArray[intVal] = intVal;
    }
    else
    {
        intArray[intVal] = -intVal;
    }
}

// any task write area
for (index = 0; index<0x10; index = index + 1)
{
    intVal = intVal + 1; // 0x21 - 0x30
    if(_command[index] == myvariable - index)
    {
        intArray[intVal] = intVal;
    }
    else
    {
        intArray[intVal] = -intVal;
    }
}

// any task write area
for (index = 0; index<0x10; index = index + 1)
{
    intVal = intVal + 1; // 0x31 - 0x40
    if(MemoryArray[index] == myvariable - index)
    {
        intArray[intVal] = intVal;
    }
    else
    {
        intArray[intVal] = -intVal;
    }
}

}
9.1.4.11 SNST Test Case

9.1.4.11.1 SAIL TASK SNST010.sai

// SNST010.sai test for correct handling of the maximum nesting depth
//             Compile, assemble, load, run and check the intArray 0x01 - 0x3E
// $Id: SNST010.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
/*
* $Log: SNST010.sai $
* Revision 1.1  2002/05/07 07:28:31  jpriley
* Initial revision
* $ */

//
//
// int intMainValue = 1;
// int intArray[0x100];
// int function01(int &intVal);
// int function02(int &intVal);
// int function03(int &intVal);
// int function04(int &intVal);
// int function05(int &intVal);
// int function06(int &intVal);
// int function07(int &intVal);
// int function08(int &intVal);
// int function09(int &intVal);
// int function10(int &intVal);
// int function11(int &intVal);
// int function12(int &intVal);
// int function13(int &intVal);
// int function14(int &intVal);
// int function15(int &intVal);
// int function16(int &intVal);
```c
void main(void)
{
  int taskID;
  int temp;

  while(TRUE)
  {
    intMainValue = 1;
    temp = function01(intMainValue);
    intArray[intMainValue] = temp;
    taskID = whoami();
    put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
    suspend(taskID);
  }
}

int function01(int &intVal)
{
  int temp;
  intArray[intVal] = intVal;
  intVal = intVal + 1; // 0x02
  temp = function02(intVal);
  intArray[intVal] = temp;
  intVal = intVal + 1; // 0x3E
  return intVal;
}

int function02(int &intVal)
{
  int temp;
  intArray[intVal] = intVal;
  intVal = intVal + 1; // 0x03
  temp = function03(intVal);
  intArray[intVal] = temp;
  intVal = intVal + 1; // 0x3D
  return intVal;
}
```

```c
int function03(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x04
    temp = function04(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x3C
    return intVal;
}

int function04(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x05
    temp = function05(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x3B
    return intVal;
}

int function05(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x06
    temp = function06(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x3A
    return intVal;
}

int function06(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x07
    temp = function07(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x39
    return intVal;
}

int function07(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x08
    temp = function08(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x38
    return intVal;
}

int function08(int &intVal)
{
    // function8 code...
}
```
```c
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x09

temp = function09(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x37
return intVal;
}

int function09(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0A

temp = function10(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x36
return intVal;
}

int function10(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0B

temp = function11(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x35
return intVal;
}

int function11(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0C

temp = function12(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x34
return intVal;
}

int function12(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0D

temp = function13(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x33
return intVal;
}

int function13(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0E

temp = function09(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x32
return intVal;
}
```

int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0E

temp = function14(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x32
return intVal;
}

int function14(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0F

temp = function15(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x31
return intVal;
}

int function15(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x10

temp = function16(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x30
return intVal;
}

int function16(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x11

temp = function17(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x2F
return intVal;
}

int function17(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x12

temp = function18(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x2E
return intVal;
}

int function18(int &intVal)
{
int temp;
```c
int function19(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x14
    temp = function20(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x2C
    return intVal;
}

int function20(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x15
    temp = function21(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x2B
    return intVal;
}

int function21(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x16
    temp = function22(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x09
    return intVal;
}

int function22(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x17
    temp = function23(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x77 (New)
    return intVal;
}

int function23(int &intVal)
{
    int temp;
```
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x18
temp = function24(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x28
return intVal;
}

int function24(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x19
temp = function25(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x27
return intVal;
}

int function25(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x1A
temp = function26(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x26
return intVal;
}

int function26(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x1B
temp = function27(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x25
return intVal;
}

int function27(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x1C
temp = function28(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x24
return intVal;
}

int function28(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x1D
...
int function29(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x1E
    temp = function30(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x22
    return intVal;
}

int function30(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x1F
    temp = function31(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x21
    return intVal;
}

int function31(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x20
    //temp = function32(intVal);
    //intArray[intVal] = temp;
    //intVal = intVal + 1;
    return intVal;
}

int function32(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1;
    return intVal;
}
/*
* Revision 1.1  2002/05/07 07:28:31  jpriley
* Initial revision
*---------------------------------------------------------------
* Error: Calling depth may not exceed 32. While calling function function32 this
* limit was exceeded.
* Error: Line: 436 Column: 10 File: SNSTERR.sai
*/

int intMainValue = 1;
int intArray[0x100];
int function01(int &intVal);
int function02(int &intVal);
int function03(int &intVal);
int function04(int &intVal);
int function05(int &intVal);
int function06(int &intVal);
int function07(int &intVal);
int function08(int &intVal);
int function09(int &intVal);
int function10(int &intVal);
int function11(int &intVal);
int function12(int &intVal);
int function13(int &intVal);
int function14(int &intVal);
int function15(int &intVal);
int function16(int &intVal);
int function17(int &intVal);
int function18(int &intVal);
int function19(int &intVal);
int function20(int &intVal);
int function21(int &intVal);
int function22(int &intVal);
int function23(int &intVal);
int function24(int &intVal);
int function25(int &intVal);
int function26(int &intVal);
int function27(int &intVal);
int function28(int &intVal);
int function29(int &intVal);
int function30(int &intVal);
int function31(int &intVal);
int function32(int &intVal);

void main(void)
```c
int taskID;
int temp;
while(TRUE)
{
    intMainValue = 1;
    temp = function01(intMainValue);
    intArray[intMainValue] = temp;
    taskID = whoami();
    put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
    suspend(taskID);
    // This should generate a "Call Overrun" error
}

int function01(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x02
    temp = function02(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x3E
    return intVal;
}

int function02(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x03
    temp = function03(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x3D
    return intVal;
}

int function03(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x04
    temp = function04(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x3C
    return intVal;
}

int function04(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    return intVal;
}
```

```c
int function05(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x06
    temp = function06(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x3A
    return intVal;
}

int function06(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x07
    temp = function07(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x39
    return intVal;
}

int function07(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x08
    temp = function08(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x38
    return intVal;
}

int function08(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x09
    temp = function09(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x37
    return intVal;
}

int function09(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x0A
    temp = function09(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x37
    return intVal;
}
```

The code snippet contains a series of function definitions, each calling the next one, with modifications to the `intVal` parameter at each step. This creates a chain of function calls, with each function updating the `intVal` parameter before returning it. The `intArray` serves as a container for storing the updated `intVal` values.
temp = function10(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x36
return intVal;
}

int function10(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0B
temp = function11(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x35
return intVal;
}

int function11(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0C
temp = function12(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x34
return intVal;
}

int function12(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0D
temp = function13(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x33
return intVal;
}

int function13(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0E
temp = function14(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x32
return intVal;
}

int function14(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x0F
intArray[intVal] = temp;
intVal = intVal + 1; // 0x31
return intVal;
}

int function15(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x10
temp = function16(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x30
return intVal;
}

int function16(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x11
temp = function17(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x2F
return intVal;
}

int function17(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x12
temp = function18(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x2E
return intVal;
}

int function18(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x13
temp = function19(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x2D
return intVal;
}

int function19(int &intVal)
{
int temp;
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x14
temp = function20(intVal);
intArray[intVal] = temp;
intVal = intVal + 1; // 0x2C
return intVal;
}
```c
int function20(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x15
    temp = function21(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x2B
    return intVal;
}

int function21(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x16
    temp = function22(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x2A
    return intVal;
}

int function22(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x17
    temp = function23(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x28
    return intVal;
}

int function23(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x18
    temp = function24(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x27
    return intVal;
}

int function24(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x19
    temp = function25(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1; // 0x27
    return intVal;
}
```
int function25(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;  // 0x1A
    intVal = intVal + 1;  // 0x26
    temp = function26(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1;  // 0x26
    return intVal;
}

int function26(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;  // 0x1B
    intVal = intVal + 1;  // 0x25
    temp = function27(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1;  // 0x25
    return intVal;
}

int function27(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;  // 0x1C
    intVal = intVal + 1;  // 0x24
    temp = function28(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1;  // 0x24
    return intVal;
}

int function28(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;  // 0x1D
    intVal = intVal + 1;  // 0x23
    temp = function29(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1;  // 0x23
    return intVal;
}

int function29(int &intVal)
{
    int temp;
    intArray[intVal] = intVal;  // 0x1E
    intVal = intVal + 1;  // 0x22
    temp = function30(intVal);
    intArray[intVal] = temp;
    intVal = intVal + 1;  // 0x22
    return intVal;
}
int function30(int &intVal)
{
 int temp;
 intArray[intVal] = intVal;
 intVal = intVal + 1; // 0x1F
 temp = function31(intVal);
 intArray[intVal] = temp;
 intVal = intVal + 1; // 0x21
 return intVal;
}

int function31(int &intVal)
{
 int temp;
 intArray[intVal] = intVal;
 intVal = intVal + 1; // 0x20
 temp = function32(intVal); // This call should generate an "Fail"
 intArray[intVal] = temp;
 return intVal;
}

int function32(int &intVal)
{
 int temp;
 intArray[intVal] = intVal;
 intVal = intVal + 1;
 return intVal;
}

9.1.4.12 SOP Test Case

9.1.4.12.1 SAIL TASK SOP010.sai

// SOP010.sai test verifies that SAIL handles all types of operators
// + - * / %
// -
// << >> ~ & ^ |

// + - * / %
// -
// << >> ~ & ^ |

/ / - - / 
/ / vs vs - vs 

211
```c
#include <stdio.h>

#define MAX_MATH_ERROR 0.0000000000001

int function(int input);

logical funcTrue(logical input);

unsigned funcUnsigned(unsigned input);

void main(void)
{
    int intArray[0x200];
    int intVal = 1;
    int resault;
    int array[10];
    unsigned unsignedArray[10];
    int taskID;
    unsigned unsignedOne, unsignedTwo;

    while(TRUE)
    {
        // operators
        if(10 + 10 == 20)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x02

        if(10 - 10 == 0)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x03

        if(10*10 == 100)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x04

        if(10/10 == 1)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x05

        if(11%10 == 1)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x06

        if(5 - 10 == -5)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x07

        if(0x10 << 4 == 0x100)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x08
    }
}
```

intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x09

unsignedOne = 0x1234; // 0b1010101010101010;
unsignedTwo = 0xEDCB; // 0b101010101010101;
if(~unsignedTwo == unsignedOne)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0A

if((0x7FFF0000 & 0x7F00FF00) == 0x7F000000)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0B

if(0x7FFF0000 ^ 0x7F00FF00 == 0xFFFF00)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0C

if(0x7FFF0000 | 0x7F00FF00 == 0x7FFFFF00)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0D

if(10 == 10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0E

if(10 != 11)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0F

if(10 < 11)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x10

if(11 > 10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x11

if(10 <= 10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x12

if(10 <= 11)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x13

if(10 >= 10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x14
if(11 >= 10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;

intVal = intVal + 1; // 0x15
if(!FALSE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;

intVal = intVal + 1; // 0x16
if(TRUE && TRUE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;

intVal = intVal + 1; // 0x17
if(TRUE && FALSE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;

intVal = intVal + 1; // 0x18
if(FALSE && TRUE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;

intVal = intVal + 1; // 0x19
if(FALSE && FALSE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;

intVal = intVal + 1; // 0x1A
if(TRUE || TRUE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;

intVal = intVal + 1; // 0x1B
if(TRUE || FALSE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;

intVal = intVal + 1; // 0x1C
if(FALSE || TRUE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;

intVal = intVal + 1; // 0x1D
if(FALSE || FALSE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;

if(3*(2 + 1) == 9)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x20

// precedence
// 6th priority befor 7th priority
if(FALSE && FALSE == FALSE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x21

if(TRUE || TRUE == FALSE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x22

if(FALSE && FALSE != TRUE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x23

if(TRUE || TRUE != TRUE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x24

if(FALSE && FALSE < TRUE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x25

if(TRUE || TRUE < TRUE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x26

if(FALSE && TRUE > FALSE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x27

if(TRUE || TRUE > TRUE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x28

if(FALSE && FALSE <= TRUE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x29

if(TRUE || TRUE <= FALSE)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x2A

if(FALSE && TRUE >= FALSE) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x2B
if(FALSE >= TRUE || TRUE)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x2C
// 5th priority befor 6th priority
if(0x10 == 0x1 << 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x2D
if(0x10 != 0x10 << 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x2E
if(0x10 < 0x10 << 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x2F
if(0x10 > 0x1 << 4) // S/B FALSE
intArray[intVal] = -intVal;
else
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x30
if(0x10 <= 0x1 << 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x31
if(0x1 >= 0x1 << 4) // S/B FALSE
intArray[intVal] = -intVal;
else
intArray[intVal] = intVal;
intVal = intVal + 1; // 0x32
if(0x10 == 0x100 >> 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x33
if(0x10 != 0x10 >> 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x34
if(0x1 < 0x100 >> 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x35
if(0x10 > 0x10 >> 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x36
if(0x10 <= 0x100 >> 4)
intArray[intVal] = intVal;
```c
else
    intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x37
    if (0x1 >= 0x10 >> 4)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x38
    if (0x10 == 0x11 & 0x10)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x39
    if (0x10 != 0x10 & 0x1)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x3A
    if (0x10 < 0x11 & 0x1) // S/B FALSE
        intArray[intVal] = -intVal;
    else
        intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x3B
    if (0x11 > 0x11 & 0x1)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x3C
    if (0x11 <= 0x11 & 0x1) // S/B FALSE
        intArray[intVal] = -intVal;
    else
        intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x3D
    if (0x10 >= 0x11 & 0x1)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x3E
    if (0x11 == 0x1 ^ 0x1) // S/B FALSE
        intArray[intVal] = -intVal;
    else
        intArray[intVal] = intVal;
    intVal = intVal + 1; // 0x3F
    if (0 != 0x11 ^ 0x1)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x40
    if (0x10 < 0x110 ^ 0x1)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x41
    if (0x11 > 0x1 ^ 0x1)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x42
    if (0x10 <= 0x11 ^ 0x1)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;
    intVal = intVal + 1; // 0x43
```

```c
intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x43
if(0x11 >= 0x11 ^ 0x1)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x44
if(0x10 == 0x10 | 0x1) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x45
if(0x11 != 0x10 | 0x1) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x46
if(0x11 < 0x10 | 0x1) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x47
if(0x11 > 0x11 | 0x1) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x48
if(0x11 <= 0x1 | 0x1) // S/B FALSE
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x49
if(0x10 >= 0x10 | 0x1)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x4A
// 4th priority befor 5th priority
if(0x1 << 3 + 1 == 0x10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x4B
if(0x10 >> 3 + 1 == 0x1)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x4C
if(0x10 & 0xF + 1 == 0x10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x4D
if(0x10 ^ 0xF + 1 == 0)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
```
intVal = intVal + 1; // 0x4E
if(0x10 | 0xF + 1 == 0x10)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x4F
if(0x1 << 5 - 1 == 0x10)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x50
if(0x10 >> 5 - 1 == 0x1)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x51
if(0x10 & 0x11 - 1 == 0x10)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x52
if(0xF ^ 0x10 - 1 == 0)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x53
if(0xF | 0x10 - 1 == 0xF)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x54
// 3rd priority before 4th priority
if(1 + 2*3 == 7)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x55
if(4 - 2*3 == -2)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x56
if(1 + 12/4 == 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x57
if(4 - 12/4 == 1)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x58
if(1 + 11%4 == 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
else  
    unsignedArray[1] = unsignedOne;  
    if(~unsignedArray[1] == unsignedOne)  
        intArray[intVal] = intVal;  
    else  
        intArray[intVal] = -intVal;  
    intVal = intVal + 1;  // 0x60

}
int function(int input)
{
    return input*4;
}

logical funcTrue(logical input)
{
    return TRUE;
}

unsigned funcUnsigned(unsigned input)
{
    return 0x1234;
}

9.1.4.12.2 SAIL TASK SOP020.sai

// SOP020.sal test verifies than SAIL handles all types of operators
// + - * / %
// -
// << >> ~ & ^ |
// == != < > <= >=
// ! && ||
// [ ] ()
// =

int function(int input);
logical funcTrue(logical input);
void main(void)
{
    int intArray[0x100];
    int intVal = 0;
    int resault;
    int array[10];
    int taskID;

    while(TRUE)
    {
        // operators
        if(-10 + 10 == 0)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x01
if(-10 - 10 == -20)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x02
if(-10*10 == -100)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x03
if(-10/10 == -1)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x04
if(-11%10 == -1)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x05
if(-5 - 10 == -15)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x06
if(-10 == -10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x07
if(-10 != -11)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x08
if(-10 > -11)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x09
if(-11 < -10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0A
if(-10 >= -10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0B
if(-10 >= -11)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0C
if(-10 <= -10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0D
if(-11 <= -10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0E
if(-10 + -10 == -20)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x0F
if(-10 - -10 == 0)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x10
if(-10*-10 == 100)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x11
if(-10/-10 == 1)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x12
if(-11%-10 == 1)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x13
if(-5 - -10 == 5)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x14
if(-10 == 10)
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x15
if(-10 != -10)
    intArray[intVal] = -intVal;
else
    intArray[intVal] = intVal;
intVal = intVal + 1; // 0x16
if(10 > -11)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x17
if(-11 < 10)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x18
if(!~-1 == 1)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1;// 0x19

if(~-!1 == -1)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1;// 0x1A

if(4 - 2 + 2 == 4)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1;// 0x1B

if(11*5/7%3 == 1)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1;// 0x1C

if(0xFFFFFFFF << 16 | 0xEE & 0x00FF00FF ^ 0x0055AA00 >> 1 ==
0x00555577)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1;// 0x1D

if(FALSE && FALSE || TRUE == TRUE)
intArray[intVal] = intVal;
else
intArray[intVal] = -intVal;
intVal = intVal + 1;// 0x1E

for(int i = 0; i < 10; i++)
for(int j = 0; j < 10; j++)
for(int k = 0; k < 10; k++)

if(function(input)
{
return input*4;
}

logical funcTrue(logical input)
{
return TRUE;
}
9.1.4.1.3 SAIL TASK SOVR010.sai

// SOVR010.c test SAIL user defined error function, overflows and underflows
// Compile, assemble, load, run and check intArray in memory 0x00 - 0x12
// Check errors in SAIL display

#define SAILUSER 0.2.0003.07 07/01/91 Testing Req 8

// $Id: SOVR010.sai 1.2 2002/06/16 12:17:42 jpriley Exp $
/*
* $Log: SOVR010.sai $
* Revision 1.2  2002/06/16 12:17:42  jpriley
* Removed obsolete error function prototype.
* Revision 1.1  2002/05/07 07:28:31  jpriley
* Initial revision
*/

int intArray[0x20];
int errorArray[0x20];
int intVal;
void main(void)
{
    intVal = 1;
taskID;
    float floatVal;
    logical logicalVal;
    byte byteVal;
    unsigned unsignedVal;
    int passIntVal;
    while(TRUE)
    {
        // logical
        logicalVal = 2;
        if(logicalVal == 1)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x02
        logicalVal = FALSE;
        logicalVal = logicalVal - 1;
        if(logicalVal == 1)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        // byte
        intVal = intVal + 1; // 0x03
        byteVal = 255;
        byteVal = byteVal + 2;
        if(byteVal == 1)
            intArray[intVal] = intVal;
    }
}
else
    if (byteVal == 254)
        intArray[intVal] = intVal;
    else
        intArray[intVal] = -intVal;

// unsigned
intVal = intVal + 1; // 0x05
unsignedVal = 65535;
unsignedVal = unsignedVal + 2;
if (unsignedVal == 1)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;

// int
intVal = intVal + 1; // 0x07
passIntVal = 2147483647;
_error_tag = 100;
passIntVal = passIntVal + 1; // overflow 0x08
passIntVal = -2147483648;
_error_tag = 200;
passIntVal = passIntVal - 1; // overflow 0x09
_error_tag = 0;
passIntVal = 0x123ABC; // hex input
if (passIntVal == 1194684)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;

// float
intVal = intVal + 1; // 0x0B
floatVal = 1.797693134862315E+308;
_error_tag = 300;
floatVal = floatVal + 0.000000000000002E+308; // overflow 0x0C and 0x0D
floatVal = -1.797693134862315E+308;
_error_tag = 400;
floatVal = floatVal - 0.000000000000002E+308; // overflow 0x0E and 0x0F
_error_tag = 0;
floatVal = 9.8813129168249e-324;
if(floatVal != 0)
  intArray[intVal] = intVal;
else
  intArray[intVal] = -intVal;

intVal = intVal + 1; // 0x10
floatVal = floatVal/4.0; // underflow set to zero
if(floatVal == 0)
  intArray[intVal] = intVal;
else
  intArray[intVal] = -intVal;

intVal = intVal + 1; // 0x11
floatVal = -9.8813129168249e-324;
if(floatVal != 0)
  intArray[intVal] = intVal;
else
  intArray[intVal] = -intVal;

intVal = intVal + 1; // 0x12
floatVal = floatVal/4.0; // underflow set to zero
if(floatVal == 0)
  intArray[intVal] = intVal;
else
  intArray[intVal] = -intVal;

intVal = intVal + 1;

void error(void)
{
  errorArray[intVal] = _error_type;
  if(_error_last_time == _error_location)
    error_kill(); // suspend the task error is repeting
  if(_error_type == ERR_OVERFLOW || _error_type == ERR_INF_NAN)
    intArray[intVal] = intVal;
  else
    intArray[intVal] = -intVal;
  intVal = intVal + 1;
  error_ignore();
}

9.1.4.14 STOVR Test Case

9.1.4.14 SAIL TASK STOVR010.sai

// STOVR010.sai test - this program runs till task overrun
void main(void)
{
    int intVal = 1;
    int taskID;
    while(TRUE)
    {
        for(intVal=0;TRUE;intVal = intVal + 1)
        {
            if(intVal%100 == 0)
            {
                intVal = 0;
                put_int(intVal,0);
                put_ascii(13);
                put_ascii(10);
            }
        }
        taskID = whoami();
        put_ascii(taskID + 48);
        put_ascii(13);
        put_ascii(10);
        suspend(taskID);
    }
}

9.1.4.15 STYPE Test Case

9.1.4.15.1 SAIL TASK STYPE010.sai
// STYPE010.c Test the sail types including all conversions
// Compile, assemble, load, run and check the intArray 0x00 - 0x08
// $Id: STYPE010.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
void main(void)
{
    int intArray[0x10];
    int intVal;
    int taskID;
    logical logicalVal1 = TRUE;
    logical logicalVal2;
    byte byteVal1 = 5;
    byte byteVal2;
    unsigned unsignedVal1 = 5;
    unsigned unsignedVal2;
    int passIntVal1 = 5;
    int passIntVal2;
    float floatVal = 5.123;

    while(TRUE)
    {
        // convert down
        intVal = 1;  // 0x01
        logicalVal2 = floatVal;
        if(logicalVal1 == logicalVal2)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x02
        byteVal2 = floatVal;
        if(byteVal1 == byteVal2)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x03
        unsignedVal2 = floatVal;
        if(unsignedVal1 == unsignedVal2)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x04
        passIntVal2 = floatVal;
        if(passIntVal1 == passIntVal2)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;

        // convert up
        intVal = intVal + 1; // 0x05
        floatVal = logicalVal1;
        if(floatVal == 1.0)
            intArray[intVal] = intVal;
        else
            intArray[intVal] = -intVal;
        intVal = intVal + 1; // 0x06
        floatVal = byteVal1;
        if(floatVal == 5.0)
```c
intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x07
floatVal = unsignedVal1;
if(floatVal == 5.0)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
intVal = intVal + 1; // 0x08
floatVal = passIntVal1;
if(floatVal == 5.0)
    intArray[intVal] = intVal;
else
    intArray[intVal] = -intVal;
taskID = whoami();
put_ascii(taskID + 48);
put_ascii(13);
put_ascii(10);
suspend(taskID);
}
}
```

**9.1.4.16 WAIT Test Case**

**9.1.4.16.1 \SAIL TASK WAIT010.sai**

// WAIT010.sal test SAIL program runs and waits over and over
//             Compile, assemble, load as task #0 and run, kill with command,
//             suspend with command, and from \WAIT020
// $Id: WAIT010.sai 1.1 2002/05/07 07:28:31 jpriley Exp $
/*
* $Log: WAIT010.sai $
* Revision 1.1  2002/05/07 07:28:31  jpriley
* Initial revision
* */
//
void main(void)
{
    int intVal = 1;
    int taskID;
    logical logicalVal = TRUE;
    while(TRUE)
    {
        for(intVal=0;TRUE;intVal = intVal + 1)
        {
            if(intVal%10 == 0)
            {
                intVal = 0;
```
```c
void main(void)
{
  int taskID;
  while(TRUE)
  {
    suspend(0);
    taskID = whoami();
    put_ascii(taskID + 48);
    put_ascii(13);
    put_ascii(10);
    suspend(taskID);
  }
}
```

9.1.4 DETAIL TASK WAIT020

// WAIT020.sal test  
// SAIL program suspends task 0. 
// Compile, assemble, load 
// WAIT020.sal as a program, just as any task 0. Compile, assemble, load 
// WAIT010.sal as a program, just as any task 0.
// File: WAIT020.sal  
// __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __ __.__
SAIL Regression Test 2 (2003-06-10)

9.2.1 SAIL_SWACC_2003_06_05.txt

F:\SAIL\SAIL_CSCI\SAIL_v2.5\SAILtasks\SAIL_SWACC>CALL SC /R /V /E /CLD 4 128 sarr010

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

COMPILING TASK sarr010

The file sarr010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sarr010.sai
Working file: sarr010.sai
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2
=============================================================================
RCS file: f:\SAIL\SAIL_CSCI\SAIL_v2.5\bin\RUNTIME_FILES\RCS\cmd.lib
Working file: cmd.lib
head: 1.3
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 3
=============================================================================
RCS file: f:\SAIL\SAIL_CSCI\SAIL_v2.5\bin\RUNTIME_FILES\RCS\comm.lib
Working file: comm.lib
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2
=============================================================================
RCS file: f:\SAIL\SAIL_CSCI\SAIL_v2.5\bin\RUNTIME_FILES\RCS\sailrtl.lib
Working file: sailrtl.lib
head: 1.4
branch:
TR-LOC-957C

locks: strict
access list:
symbolic names:
keyword substitution: kv

=============================================================================

RCS file: \F\SAIL\SAIL_CSCI\SAIL_v2.5\bin\RUNTIME_FILES\RCS\constant.sai
Working file: constant.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

=============================================================================

--------------- First Pass Errors/Warnings ------------

--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sarr010_code.cld ...
Creating sarr010_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file sarr011.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sarr011.sai
Working file: sarr011.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

total revisions: 1

--------------- Creating .cld files ------------------

Creating sarr011_code.cld ...
Creating sarr011_data.cld ...

--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sarr011_code.cld ...
Creating sarr011_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file sarr011.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sarr011.sai
Working file: sarr011.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

total revisions: 1

233
Creating sarr011_code.cld ...

Creating sarr011_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

COMPILING TASK sarr012

The file sarr012.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sarr012.sai
Working file: sarr012.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

The file sarr012.sai does not exist.

BOX file: MEMORY.SAI
RCS file: MEMORY.SAI
User file: MEMORY.SAI
$view: 1
Note:
monthly update
keyword substitution: In (code version)
Creating sarr012_data.cld...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///////\ COMPILING TASK sarr014 ///////\n
The file sarr014.dso does not exist.

The file sarr014.dso does not exist.

The file sarr014.dso does not exist.

The file sarr014.dso does not exist.

The file sarr014.dso does not exist.

Creating sarr013_code.cld...

Creating sarr013_data.cld...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///////\ COMPILING TASK sarr014 ///////\n
The file sarr014.dso does not exist.

The file sarr014.dso does not exist.
Attempting to retrieve from RCS ... 

RCS file: RCS\sarr014.sai  
Working file: sarr014.sai 
head: 1.2  
branch:  
locks: strict  
access list:  
symbolic names:  
keyword substitution: kv  
total revisions: 2 
============================================================================= 

--------------- First Pass Errors/Warnings -----------

--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sarr014_code.cld ...
Creating sarr014_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///COMPILING TASK sarr020 ///

The file sarr020.sai does not exist.
Attempting to retrieve from RCS ... 

RCS file: RCS\sarr020.sai  
Working file: sarr020.sai 
head: 1.1  
branch:  
locks: strict  
access list:  
symbolic names:  
keyword substitution: kv  
total revisions: 1 
============================================================================= 

--------------- First Pass Errors/Warnings -----------

236
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------

Creating sarr020_code.cld ...
Creating sarr020_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

////////
COMPILING TASK sarr030 ////////
The file sarr030.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sarr030.sai
Working file: sarr030.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================
--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------

Creating sarr030_code.cld ...
Creating sarr030_data.cld ...

SAIL Compile Utility, Version:
The file scerr010.sai does not exist.

Attempting to retrieve from RCS...

RCS file: RCS/scerr010.sai
Working file: scerr010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

============================================================================= 
--------------- First Pass Errors/Warnings  ------------
Error: An invalid preprocessor directive was found.
Error occurred in file scerr010.sai, line 19, column 26
Error: This include directive was improperly formatted.
Error occurred in file scerr010.sai, line 20, column 26
Error: A bad special character was encountered.
Error occurred in file scerr010.sai, line 21, column 2
Error: Only source files with a "sai" extension can be processed.
Error occurred in file xxxxxxxxx.h
Error: An error occurred while trying to include a file.
Error occurred in file scerr020.sai, line 15, column 2
Error: The end of the file was reached, but the comment was not terminated.
Error occurred in file scerr030.sai, line 15, column 2
Error: An invalid floating point number was encountered.
Error occurred in file scerr010.sai, line 26, column 24
Error: This floating point number has no sign for its exponent.
Error occurred in file scerr010.sai, line 27, column 39
Error: An exponent was expected for this floating point number.
Error occurred in file scerr010.sai, line 28, column 42
Error: The exponent for this floating point number is too large.
Error occurred in file scerr010.sai, line 29, column 46

236
Error: This floating point number is too long.
Error: This floating point number has created overflow.
Error: This floating point number has created overflow.
Error: This floating point number has created overflow.
Fatal Error: A symbol has violated the symbol size limitation.
SAIL Compile Utility, Version:
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)
\//////// COMPILING TASK scerr040 \////////
The file scerr040.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\scerr040.sai
Working file: scerr040.sai
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2
=============================================================================--------------- First Pass Errors/Warnings -----------
Error: The type of the reference parameter badref must match the exact type of the corresponding reference argument.
Error: Line: 91 Column: 23 File: scerr040.sai
Error: You cannot pass Immediates or Expressions by reference.
Error: Line: 96 Column: 23 File: scerr040.sai
Error: In an expression, the token [ can't be placed after the value intarray.
Error: Line: 104 Column: 24 File: scerr040.sai
Error: While passing baddimension to functionnormalarray, the dimension(s) of baddimension did not match the dimension(s) of array as defined in the prototype for functionnormalarray.
Error: Line: 115 Column: 22 File: scerr040.sai

---------------------------------------------------------------
------------------------------ Second Pass Errors/Warnings ------------------------------
Warning: The function functionwithreturn has a non-void return type, but it does not appear to return a value.

Warning: Line: 119 Column: 5 File: scerr040.sai
Warning: The return value from function functionwithreturn was ignored.

Error: An incorrect number of parameters were passed to functionparameter1.

Error: Line: 141 Column: 24 File: scerr040.sai
Error: An incorrect number of parameters were passed to functionparameter2.

Error: Line: 142 Column: 32 File: scerr040.sai

Error: A goto cannot jump to the label already_been_placed because it has not been placed.

Error: Line: 194 Column: 2 File: scerr040.sai

Error: The identifier undefined cannot be used until it is defined.

Error: Line: 179 Column: 18 File: scerr040.sai
Warning: This expression is too complex.

Warning: This expression is missing parentheses.

Warning: Only functions may be called. An attempt was made to call the non-function token callofnonfunction.

Warning: You cannot use the non-RTL function functioncallinsideerror while inside the body of the error function.

Warning: The function functionwithoutaprototype cannot be defined until it has been prototyped.

Warning: The function functionshouldnotreturn has a void return type, but it returns a value.

Warning: A goto cannot jump from an outer code block into an inner code block.
The label already_been_placed is in an inner block relative to the jumping goto statement.

Error: Line: 264 Column: 3 File: scerr040.sai
Warning: A goto cannot jump from an outer code block into an inner code block.
The label innerblock05 is in an inner block relative to the jumping goto statement.

Error: Line: 267 Column: 5 File: scerr040.sai
Warning: A goto cannot jump from an outer code block into an inner code block.
The label innerblock06 is in an inner block relative to the jumping goto statement.

Error: Line: 268 Column: 2 File: scerr040.sai
Warning: A goto cannot jump from an outer code block into an inner code block.
The label innerblock07 is in an inner block relative to the jumping goto statement.

Error: Line: 268 Column: 2 File: scerr040.sai
Warning: A goto cannot jump from an outer code block into an inner code block.
The label innerblock08 is in an inner block relative to the jumping goto statement.

Error: Line: 268 Column: 2 File: scerr040.sai
Warning: A goto cannot jump from an outer code block into an inner code block.
The label innerblock09 is in an inner block relative to the jumping goto statement.

Error: Line: 268 Column: 2 File: scerr040.sai
Warning: A goto cannot jump from an outer code block into an inner code block.
The label innerblock10 is in an inner block relative to the jumping goto statement.
The file scmp010.doe does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS/scmp010.doe
Working file: scmp010.doe
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================  
--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings ------

The file scmp020.doe does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS/scmp020.doe
Working file: scmp020.doe
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================  
--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings ------
Creating scmp020_code.cld ...

Creating scmp020_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:/SAIL/SAIL_CSCI/SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

...... COMPILING TASK scoms010 ......
The file scoms010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS/scoms010.sai
Working file: scoms010.sai
head: 1.2
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv
total revisions: 2
=============================================================================
--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
Creating scoms010_code.cld ...
Creating scoms010_data.cld ...
SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:/SAIL/SAIL_CSCI/SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

248
The file scoms020.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\scoms020.sai
Working file: scoms020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

total revisions: 1
=============================================================================
--------------- First Pass Errors/Warnings ---------------
--------------- Second Pass Errors/Warnings -------------
--------------- Assembler Pass Errors/Warnings ----------
--------------- Creating .cld files ------------------
Creating scoms020_code.cld ...
Creating scoms020_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file sdata010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sdata010.sai
Working file: sdata010.sai
head: 1.4
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
Creating sdata010_code.cld ...

Creating sdata010_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

//COMPILING TASK sexec000 //

The file sexec000.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sexec000.sai
Working file: sexec000.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

Creating sexec000_code.cld ...

--------- First Pass Errors/Warnings ---------
--------- Second Pass Errors/Warnings ---------
--------- Assembler Pass Errors/Warnings ---------
--------- Creating .cld files ---------

Creating sexec000_code.cld ...

--------- First Pass Errors/Warnings ---------
--------- Second Pass Errors/Warnings ---------
--------- Assembler Pass Errors/Warnings ---------
--------- Creating .cld files ---------
Creating sexec000_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

////////
COMPILING TASK sexec001 ////////

The file sexec001.sai does not exist...

RCS file: RCS\sexec001.sai
Working file: sexec001.sai
head: 1.1

access: read

symbolic names:

keyword substitution: kv

--- total revisions: 1

=============================================================================
--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
Creating sexec001_code.cld 
Creating sexec001_data.cld 

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

////////
COMPILING TASK sexec002 ////////

Creating sexec002_data.cld ...
The file sexec002.sai does not exist.

Attempting to retrieve from RCS ...

RCS file: RCS\sexec002.sai
Working file: sexec002.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================  
--------------- First Pass Errors/Warnings ---------------

--------------- Second Pass Errors/Warnings ------------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sexec002_code.cld ...

Creating sexec002_data.cld ...

SAIL Compile Utility, Version: $Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory: f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--------------- COMPILING TASK sexec003 ---------------

The file sexec003.sai does not exist.

Attempting to retrieve from RCS ...

RCS file: RCS\sexec003.sai
Working file: sexec003.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================  
--------------- First Pass Errors/Warnings ---------------
Creating sexec003_code.cld ...
Creating sexec003_data.cld ...

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file sexec004.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sexec004.sai
Working file: sexec004.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================
The file sexec005.sai does not exist.
Attemping to retrieve from RCS ...

RCS file: RCS/sexec005.sai
Working file: sexec005.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
============================================================================= 
--------------- First Pass Errors/Warnings  -----------
--------------- Second Pass Errors/Warnings -----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
	Creating sexec005_code.cld ...
	Creating sexec005_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--------------- Creating All file ---------------------

Creating sexec006.scl ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)
Creating sexec006_code.cld ...

Creating sexec006_data.cld ...

The file sexec007.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sexec007.sai
Working file: sexec007.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

The file sexec007.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sexec007.sai
Working file: sexec007.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
--------------- Assembler Pass Errors/Warnings ------------

--------------- Creating .cld files -----------------

Creating sexec007_code.cld ...
Creating sexec007_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

//\COMPILING TASK sexec008 //\nThe file sexec008.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sexec008.sai
Working file: sexec008.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

total revisions: 1
=============================================================================  

--------------- First Pass Errors/Warnings -----------------

--------------- Second Pass Errors/Warnings -----------------

--------------- Assembler Pass Errors/Warnings ------------

--------------- Creating .cld files -----------------

Creating sexec008_code.cld ...
Creating sexec008_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:

252
The file sexec009.sai does not exist.

Attemping to retrieve from RCS...

RCS file: RCS\sexec009.sai
Working file: sexec009.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

------------------------------- First Pass Errors/Warnings -----------------------------

------------------------------- Second Pass Errors/Warnings -----------------------------

------------------------------- Assembler Pass Errors/Warnings -----------------------------

------------------------------- Creating .cld files --------------------------------------

Creating sexec009_code.cld ...
Creating sexec009_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file sexec010.sai does not exist.

Attemping to retrieve from RCS...

RCS file: RCS\sexec010.sai
Working file: sexec010.sai
head: 1.1
branch:

251
The file sexec011.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sexec011.sai
Working file: sexec011.sai
head: 1.1
branch: locks: strict
access list: symbolic names:
keyword substitution: kv
total revisions: 1
Creating sexec011_code.cld ...

Creating sexec011_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

---

The file sexec012.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sexec012.sai
Working file: sexec012.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================

--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
Creating sexec012_code.cld ...
Creating sexec012_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)
COMPILING TASK sexec013

The file sexec013.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS\sexec013.sai
Working file: sexec013.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

--------------- First Pass Errors/Warnings ------------

--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files -----------------

Creating sexec013_code.cld ...
Creating sexec013_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

COMPILING TASK sexec014

The file sexec014.sai does not exist. Attempting to retrieve from RCS...

RCS file: RCS\sexec014.sai
Working file: sexec014.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
Creating sexec014_code.cld ...

Creating sexec014_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

COMPILING TASK sexec015

The file sexec015.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sexec015.sai
Working file: sexec015.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

The file sexec015.cai does not exist.

The file sexec015_code.cld does not exist.

The file sexec015_data.cld does not exist.
TR-LOC-957C

2003-06-12

Creating sexec015_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

/\/\/\/\/\/\/\/\ COMPILING TASK sexec016 /\/\/\/\/\/\/\/\

The file sexec016.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sexec016.sai
Working file: sexec016.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================
--------------- First Pass Errors/Warnings -----------

--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files -----------------Creating sexec016_code.cld ...

Creating sexec016_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

/\/\/\/\/\/\/\/\ COMPILING TASK sexec020 /\/\/\/\/\/\/\/\

258


The file sexec020.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\sexec020.sai
Working file: sexec020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================}
--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
Creating sexec020_code.cld ...
Creating sexec020_data.cld ...
Creating sexec020_class.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file sferr010.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\sferr010.sai
Working file: sferr010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================}
--------------- First Pass Errors/Warnings ------------
Creating sferr010_code.cld ...

Creating sferr010_data.cld ...

SAIL Compile Utility, Version:

$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT environment variable)

////////////////////////////////////////////////////////////
// COMPILING TASK sferr020 ///////////////////////////////////////////////////////////

The file sferr020.sai does not exist.

Attempting to retrieve from RCS ...

RCS file: RCS\sferr020.sai
Working file: sferr020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

total revisions: 1

=============================================================================  

////////////////////////////////////////////////////////////
// FIRST PASS ERRORS/WARNINGS \\
//ALSError Bounds Error/warning

The file sferr020.sai does not exist

Attempting to retrieve from RCS ...

RCS file: RCS\sferr020.sai
Working file: sferr020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

total revisions: 1

=============================================================================  

////////////////////////////////////////////////////////////
// SECOND PASS ERRORS/WARNINGS \\
//ALSError Bounds Error/warning

The file sferr020.sai does not exist

Attempting to retrieve from RCS ...

RCS file: RCS\sferr020.sai
Working file: sferr020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

total revisions: 1

=============================================================================  

////////////////////////////////////////////////////////////
// ASSEMBLER PASS ERRORS/WARNINGS \\
//ALSError Bounds Error/warning

The file sferr020.sai does not exist

Attempting to retrieve from RCS ...

RCS file: RCS\sferr020.sai
Working file: sferr020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

total revisions: 1

=============================================================================  

////////////////////////////////////////////////////////////
// CREATE .CLD FILES \\

Creating sferr020_code.cld ...

Creating sferr020_data.cld ...
The file sferr030.sai does not exist.

Attempting to retrieve from RCS ...

RCS file: RCS/sferr030.sai
Working file: sferr030.sai

RCS client: admnt 5.2
compress None
symbolic names:
keyword substitution: kv

=============================================================================
--------------- First Pass Errors/Warnings  -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
Creating sferr030_code.cld ...
Creating sferr030_data.cld ...

The file sferr040.sai does not exist.

Attempting to retrieve from RCS ...

RCS file: RCS/sferr040.sai
Working file: sferr040.sai

RCS client: admnt 5.2
compress None
symbolic names:
keyword substitution: kv

=============================================================================
RCS file: RCS\sferr040.sai
Working file: sferr040.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

============================================================================= 
--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sferr040_code.cld ...
Creating sferr040_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--------------- Creating Task sferr050 -----------------

The file sferr050.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\sferr050.sai
Working file: sferr050.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

============================================================================= 
--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sferr050_code.cld ...
Creating sferr050_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--------------- Creating Task sferr050 -----------------
Creating .cld files

Creating sferr050_code.cld ...

Creating sferr050_data.cld ...

SAIL Compile Utility, Version:

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT environment variable)

The file sfun010.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\sfun010.sai
Working file: sfun010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

============================================================================= 

--------------- First Pass Errors/Warnings -----------

--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sfun010_code.cld ...

Creating sfun010_data.cld ...

SAIL Compile Utility, Version:

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT environment variable)
COMPILING TASK sierr010

The file sierr010.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\sierr010.sai
Working file: sierr010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
Creating sierr010_code.cld...
Creating sierr010_data.cld...

//SAIL Compile Utility, Version: #Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp //SAIL Preprocessor/Compiler/Assembler Root Directory: \f:\SAIL\SAIL_CSCI\SAIL_v2.5 \Note: To change Root directory, modify the $SAIL_ROOT user environment variable

--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
Creating smem010_code.cld...
Creating smem010_data.cld...

//SAIL Compile Utility, Version: #Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp //SAIL Preprocessor/Compiler/Assembler Root Directory: \f:\SAIL\SAIL_CSCI\SAIL_v2.5 \Note: To change Root directory, modify the $SAIL_ROOT user environment variable

COMPILING TASK smem010

The file smem010.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\smem010.sai
Working file: smem010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
Creating smem010_code.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

2003/05/01 17:57:56 TASK smem010

The file snst010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\snst010.sai
Working file: snst010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

Creating smem010_data.cld ...

Creating smem010_code.cld ...
Creating snst010.data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///\///\///\///\///\///COMPILING TASK snsterr ///\///\///\///\///\///\///\///\///\///\///\///

The file snsterr.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\snsterr.sai
Working file: snsterr.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================
--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
Error: Calling depth may not exceed 32. While calling function function32 this limit was exceeded.
Error: Line: 447 Column: 10 File: snsterr.sai
--------------- Assembler Pass Errors/Warnings ------

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///\///\///\///\///\///COMPILING TASK sop010 ///\///\///\///\///\///\///\///\///\///\///\///
The file sop010.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\sop010.sai
Working file: sop010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================
--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
Error: Calling depth may not exceed 32. While calling function function32 this limit was exceeded.
Error: Line: 447 Column: 10 File: sop010.sai
--------------- Assembler Pass Errors/Warnings ------

260
TR-LOC-870C 2003-06-12

--------------- Assembler Pass Errors/Warnings ------------

Creating sop020_code.cld ...

Creating sop020_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Notice: To change Root directory, modify the SAIL_ROOT environment variable)

///\COMPILING TASK sovr010 ///\[/[///

The file sovr010.sai cannot be opened.
âCurrently opening it from ICE.

RCS file: RCS\sovr010.sai
Working file: sovr010.sai

head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

============================================================================= 

--------------- First Pass Errors/Warnings ------------

--------------- Second Pass Errors/Warnings ------------

--------------- Assembler Pass Errors/Warnings ------------

Creating sop020_code.cld ...

Creating sop020_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
The file srtl010.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\srtl010.sai
Working file: srtl010.sai
head: 1.1
branch:

=============================================================================
--------------- First Pass Errors/Warnings --------------
--------------- Second Pass Errors/Warnings --------------
--------------- Assembler Pass Errors/Warnings -----------
--------------- Creating .cld files ---------------------

    Creating srtl010_code.cld ...
    Creating srtl010_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file srtl020.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\srtl020.sai
Working file: srtl020.sai
head: 1.1
branch:
SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

////////\ COMPILING TASK sst010 ////////\ 

The file sst010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sst010.sai
Working file: sst010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------

Creating srtl020_code.cld ...
Creating srtl020_data.cld ...

Creating srtl020_cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
Creating sst010_code.cld ...

Creating sst010_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file stype010.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\stype010.sai
Working file: stype010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================
The file stovr010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\stovr010.sai
Working file: stovr010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------

Creating stovr010_code.cld ...
Creating stovr010_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file wait010.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\wait010.sai
Working file: wait010.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
Creating wait010_code.cld ...

Creating wait010_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

////////
\ COMPILING TASK wait020 \////////

The file wait020.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\wait020.sai
Working file: wait020.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

total revisions: 1

=============================================================================
Creating wait030_code.cld ...

Creating wait030_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///////\ COMPILING TASK wait030 ///////\

The file wait030.sai does not exist.

RCS file: RCS\wait030.sai
Working file: wait030.sai
head: 1.1

branch:

locks: strict

access list:

symbolic names:

keyword substitution: kv

total revisions: 1

=============================================================================
--------------- First Pass Errors/Warnings  -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------

Creating wait030_code.cld ...

Creating wait030_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///////\ COMPILING TASK sasmoversizelimit ///////\

274
Code size exceeds 64K at line 65539

F:\SAIL\SAIL_CSCI\SAIL_v2.5\SAILtasks\SAIL_SWACC\sasmoversizelimit.a

Creating sasmoversizelimit_code.cld...
The file sw_loc_117_3_1_1_01.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_3_1_1_01.sai
Working file: sw_loc_117_3_1_1_01.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================
RCS file: f:\SAIL\SAIL_CSCI\SAIL_v2.5\bin\RUNTIME_FILES\RCS\cmd.lib
Working file: cmd.lib
head: 1.3
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 3
=============================================================================
RCS file: f:\SAIL\SAIL_CSCI\SAIL_v2.5\bin\RUNTIME_FILES\RCS\comm.lib
Working file: comm.lib
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2
=============================================================================
RCS file: f:\SAIL\SAIL_CSCI\SAIL_v2.5\bin\RUNTIME_FILES\RCS\sailrtl.lib
Working file: sailrtl.lib
head: 1.4
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 4
=============================================================================
The file sw_loc_117_3_1_1_02.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_3_1_1_02.sai
Working file: sw_loc_117_3_1_1_02.sai
head: 1.1
branch: 
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================
Creating .cld files

--------------- Creating .cld files ------------------

Creating sw_loc_117_3_1_1_02_code.cld ...

Creating sw_loc_117_3_1_1_02_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT environment variable)

--------------- First Pass Errors/Warnings -------------

--------------- Second Pass Errors/Warnings -----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sw_loc_117_3_1_2_01_code.cld ...

Creating sw_loc_117_3_1_2_01_data.cld ...

The file sw_loc_117_3_1_2_01.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_3_1_2_01.sai
Working file: sw_loc_117_3_1_2_01.sai
head: 1.1
lock: main
access list: main
symbolic names: ...
keyword substitution: kv

=============================================================================

--------------- Second Pass Errors/Warnings -----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sw_loc_117_3_1_2_01_code.cld ...

Creating sw_loc_117_3_1_2_01_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT environment variable)
The file sw_loc_117_3_1_2_02.sai does not exist.
Attempting to retrieve from RCS...
RCS file: RCS\sw_loc_117_3_1_2_02.sai
Working file: sw_loc_117_3_1_2_02.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
Creating sw_loc_117_3_1_2_02_code.cld...
Creating sw_loc_117_3_1_2_02_data.cld...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file sw_loc_117_3_2_1_01.sai does not exist.
Attempting to retrieve from RCS...
RCS file: RCS\sw_loc_117_3_2_1_01.sai
Working file: sw_loc_117_3_2_1_01.sai
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------
Creating sw_loc_117_3_2_1_01_code.cld...
Creating sw_loc_117_3_2_1_01_data.cld...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)
Creating sw_loc_117_3_3_1_2_1_01_code.cld ...

Creating sw_loc_117_3_3_1_2_1_01_data.cld ...

SAIL Compile Utility, Version:
SAIL is the registered trademark of Peter S. von Rader
SAIL Preprocessor/Compiler/Assembler Root Directory: f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

The file sw_loc_117_3_3_1_2_1_01.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_3_3_1_2_1_01.sai
Working file: sw_loc_117_3_3_1_2_1_01.sai
head: 1.1
branch: 
locks: strict
access list: 
symbolic names: 
keyword substitution: kv

The file sw_loc_117_3_3_1_2_1_01.sai does not exist.

Creating sw_loc_117_3_3_1_2_1_01_code.cld ...

Creating sw_loc_117_3_3_1_2_1_01_data.cld ...

281
Creating sw_loc_117_3_3_1_2_1_02_code.cld ...

Creating sw_loc_117_3_3_1_2_1_02_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

----SAIL------\ COMPILING TASK sw_loc_117_3_3_1_2_2_01 /\----SAIL------

The file \sw_loc_117_3_3_1_2_2_01\ does not exist.

Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_3_3_1_2_2_01
Working file: \sw_loc_117_3_3_1_2_2_01\ (head 1.3)
Status: New

access list:
symbolic names:
keyword substitution: kv

total revisions: 1

=============================================================================
--------------- First Pass Errors/Warnings -----------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files ------------------

Creating sw_loc_117_3_3_1_2_2_01_code.cld ...

Creating sw_loc_117_3_3_1_2_2_01_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

----SAIL------\ COMPILING TASK sw_loc_117_3_3_1_2_3_01 /\----SAIL------
The file `sw_loc_117_3_3_1_2_2_01.sai` does not exist.

Attempting to retrieve from RCS...

RCS file: `RCS\sw_loc_117_3_3_1_2_2_01.sai`

Working file: `sw_loc_117_3_3_1_2_2_01.sai`

head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

-----------------------------------------------------------------------------

--------------- First Pass Errors/Warnings ---------------

--------------- Second Pass Errors/Warnings --------------

--------------- Assembler Pass Errors/Warnings ------------

--------------- Creating .cld files ------------------

Creating `sw_loc_117_3_3_1_2_2_01_code.cld` ...

Creating `sw_loc_117_3_3_1_2_2_01_data.cld` ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

///////// COMPILING TASK sw_loc_117_3_3_1_2_3_01 /////////

The file `sw_loc_117_3_3_1_2_3_01.sai` does not exist.

Attempting to retrieve from RCS...

RCS file: `RCS\sw_loc_117_3_3_1_2_3_01.sai`

Working file: `sw_loc_117_3_3_1_2_3_01.sai`

head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

-----------------------------------------------------------------------------

--------------- First Pass Errors/Warnings ---------------

--------------- Second Pass Errors/Warnings --------------

--------------- Assembler Pass Errors/Warnings ------------
Creating .cld files...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor/Compiler/Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--------------- First Pass Errors/Warnings ----------

The file sw_loc_117_3_3_1_2_4_01.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\sw_loc_117_3_3_1_2_4_01.sai
Working file: sw_loc_117_3_3_1_2_4_01.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================

--------------- Second Pass Errors/Warnings -------------

--------------- Assembler Pass Errors/Warnings ---------

Creating .cld files...

Creating sw_code_117_3_3_1_2_4_01.cld...
Creating sw_data_117_3_3_1_2_4_01.cld...
Creating sw_code_117_3_3_1_2_4_01.cld...
Creating sw_data_117_3_3_1_2_4_01.cld...

--------------- Creating .cld files ------------------

Creating sw_code_117_3_3_1_2_3_01.cld...
Creating sw_data_117_3_3_1_2_3_01.cld...
The file \texttt{sw\_loc\_117\_3\_3\_1\_2\_5\_01.sai} does not exist.

Attempting to retrieve from RCS ...

RCS file: \texttt{RCS\sw\_loc\_117\_3\_3\_1\_2\_5\_01.sai}

Working file: \texttt{sw\_loc\_117\_3\_3\_1\_2\_5\_01.sai}

head: 1.1  
branch: 
locks: strict 
access list: 
symbolic names: 

--------------- First Pass Errors/Warnings ------------
--------------- Second Pass Errors/Warnings ----------
--------------- Assembler Pass Errors/Warnings -------
--------------- Creating .cld files -------------------

SAIL Compile Utility, Version:  
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $  
SAIL Preprocessor/Compiler/Assembler Root Directory:  
f:\SAIL\SAIL\CSCI\SAIL\v2.5  
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)
RCS file: RCS\sw_loc_117_3_3_1_3_01.sai
Working file: sw_loc_117_3_3_1_3_01.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=============================================================================  
--------------- First Pass Errors/Warnings -----------  
--------------- Second Pass Errors/Warnings ----------  
--------------- Assembler Pass Errors/Warnings -------  
--------------- Creating .cld files ------------------  
Creating sw_loc_117_3_3_1_3_01_code.cld ...
Creating sw_loc_117_3_3_1_3_01_data.cld ...
SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)
////////\ COMPILING TASK sw_loc_117_3_3_1_4_01 ////////\  
The file sw_loc_117_3_3_1_4_01.sai does not exist.
Attempting to retrieve from RCS ...
RCS file: RCS\sw_loc_117_3_3_1_4_01.sai
Working file: sw_loc_117_3_3_1_4_01.sai
head: 1.2
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 2
=============================================================================  
--------------- First Pass Errors/Warnings -----------  
--------------- Second Pass Errors/Warnings ----------  
--------------- Assembler Pass Errors/Warnings -------  
--------------- Creating .cld files ------------------  

285
Creating files ...

SAIL Compile Utility, Version:
SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5 (Note: To change root directory, modify the SAIL_ROOT user environment variable)

------ First Pass Errors/Warnings ------

The file sw_loc_117_3_3_2_1_01.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_3_3_2_1_01.sai
Working file: sw_loc_117_3_3_2_1_01.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1
=================================================================

------ Second Pass Errors/Warnings ------

------ Assembler Pass Errors/Warnings ------
The file sw_loc_117_3_3_2_2_01.sai does not exist.
Attempting to retrieve from RCS ... 

RCS file: RCS\sw_loc_117_3_3_2_2_01.sai
Working file: sw_loc_117_3_3_2_2_01.sai
head: 1.1
branch:

--- SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$

--- SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--- Assembler Preprocessor/Warnings: -------

--- Second Pass Errors/Warnings: --------

--- First Pass Errors/Warnings: ---------

--- Assembler File Errors/Warnings: ------

--- Creating .cld files ------------------
Creating sw_loc_117_3_3_2_2_01_code.cld ...
Creating sw_loc_117_3_3_2_2_01_data.cld ...

The file sw_loc_117_3_3_2_2_01.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_3_3_2_2_01.sai
Working file: sw_loc_117_3_3_2_2_01.sai
head: 1.1
branch:

--- SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$

--- SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--- Assembler Preprocessor/Warnings: -------

--- Second Pass Errors/Warnings: --------

--- First Pass Errors/Warnings: ---------

--- Assembler File Errors/Warnings: ------

--- Creating .cld files ------------------
Creating sw_loc_117_3_4_1_1_01_code.cld ...
Creating sw_loc_117_3_4_1_1_01_data.cld ...

--- Assembler Preprocessor/Warnings: -------

--- Second Pass Errors/Warnings: --------

--- First Pass Errors/Warnings: ---------

--- Assembler File Errors/Warnings: ------

--- Creating .cld files ------------------
Creating sw_loc_117_3_4_1_1_01_code.cld ...
Creating sw_loc_117_3_4_1_1_01_data.cld ...

--- SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$

--- SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--- Assembler Preprocessor/Warnings: -------

--- Second Pass Errors/Warnings: --------

--- First Pass Errors/Warnings: ---------

--- Assembler File Errors/Warnings: ------

--- Creating .cld files ------------------
Creating sw_loc_117_3_4_1_1_01_code.cld ...
Creating sw_loc_117_3_4_1_1_01_data.cld ...

--- SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$

--- SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--- Assembler Preprocessor/Warnings: -------

--- Second Pass Errors/Warnings: --------

--- First Pass Errors/Warnings: ---------

--- Assembler File Errors/Warnings: ------

--- Creating .cld files ------------------
Creating sw_loc_117_3_4_1_1_01_code.cld ...
Creating sw_loc_117_3_4_1_1_01_data.cld ...

--- SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$

--- SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

--- Assembler Preprocessor/Warnings: -------

--- Second Pass Errors/Warnings: --------

--- First Pass Errors/Warnings: ---------

--- Assembler File Errors/Warnings: ------

--- Creating .cld files ------------------
Creating sw_loc_117_3_4_1_1_01_code.cld ...
Creating sw_loc_117_3_4_1_1_01_data.cld ...

--- SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp$

--- SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)
Creating sw_loc_117_3_4_1_1_01_code.cld ...

Creating sw_loc_117_3_4_1_1_01_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

//\///\///\/// COMPILING TASK sw_loc_117_3_4_1_2_01 //\///\///\///
The file sw_loc_117_3_4_1_2_01.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_3_4_1_2_01.sai
Working file: sw_loc_117_3_4_1_2_01.sai
head: 1.1
branch: 
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

=============================================================================
--------------- Creating .cld files ------------------

Creating sw_loc_117_3_4_1_2_01_code.cld ...

Creating sw_loc_117_3_4_1_2_01_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

/////////\ COMPILING TASK sw_loc_117_3_5_2_01 /////////\

The file sw_loc_117_3_5_2_01.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_3_5_2_01.sai
Working file: sw_loc_117_3_5_2_01.sai
head: 1.2

 keyword substitution: kv
 total revisions: 2

=============================================================================
--------------- First Pass Errors/Warnings  -----------

--------------- Second Pass Errors/Warnings ----------

--------------- Assembler Pass Errors/Warnings -------

--------------- Creating .cld files ------------------

Creating sw_loc_117_3_5_2_01_code.cld ...

Creating sw_loc_117_3_5_2_01_data.cld ...

SAIL Compile Utility, Version:
$Id: sc.bat 1.8 2003/05/01 17:57:56 jpriley Exp $

SAIL Preprocessor\Compiler\Assembler Root Directory:
f:\SAIL\SAIL_CSCI\SAIL_v2.5

(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

289
The file sw_loc_117_6_01.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\sw_loc_117_6_01.sai
Working file: sw_loc_117_6_01.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv

=============================================================================  
--------------- First Pass Errors/Warnings  -----------  
--------------- Second Pass Errors/Warnings ----------  
--------------- Assembler Pass Errors/Warnings -------  
--------------- Creating .cld files ------------------  

Creating sw_loc_117_6_01_code.cld ...  
Creating sw_loc_117_6_01_data.cld ...

---

The file sw_loc_117_6_02.sai does not exist.
Attempting to retrieve from RCS...

RCS file: RCS\sw_loc_117_6_02.sai
Working file: sw_loc_117_6_02.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
Creating sw_loc_117_6_02_code.cld ...

Creating sw_loc_117_6_02_data.cld ...

SAIL Compile Utility, Version:
SAIL Preprocessor/Compiler/Assembler Root Directory: f:\SAIL\SAIL_CSCI\SAIL_v2.5
(Note: To change Root directory, modify the SAIL_ROOT user environment variable)

//COMPILING TASK sw_loc_117_6_03

The file sw_loc_117_6_03.sai does not exist.
Attempting to retrieve from RCS ...

RCS file: RCS\sw_loc_117_6_03.sai
Working file: sw_loc_117_6_03.sai
head: 1.1
branch:
locks: strict
access list:
symbolic names:
keyword substitution: kv
total revisions: 1

//END OF COMPILING TASK sw_loc_117_6_03

Creating sw_loc_117_6_03_code.cld ...
9.2.2 SAIL_SWACC_2003_06_05.err

Error: An invalid preprocessor directive was found.
Error occurred in file scerr010.sai, line 19, column 26
Error: This include directive was improperly formatted.
Error occurred in file scerr010.sai, line 20, column 26
Error: A bad special character was encountered.
Error occurred in file scerr010.sai, line 21, column 3
Error: This source files lacks a file extension can be processed.
Error occurred in file scerr010.sai, line 22, column 3
Error: An error occurred while trying to include a file.
Error occurred in file scerr010.sai, line 23, column 2
Error: The file arrangement is correct.
Error occurred in file scerr010.sai, line 24, column 2
Error: This file is not present, but the comment was not terminated.
Error occurred in file scerr010.sai, line 25, column 2
Error: An invalid floating point number was encountered.
Error occurred in file scerr010.sai, line 26, column 2
Error: This floating point number has no sign for its exponent.
Error occurred in file scerr010.sai, line 27, column 2
Error: An exponent was expected for the floating point number.
Error occurred in file scerr010.sai, line 28, column 2
Error: The exponent for this floating point number is too large.
Error occurred in file scerr010.sai, line 29, column 2
Error: This floating point number in too long.
Error occurred in file scerr010.sai, line 30, column 2
Error: This number has created an overflow.
Error occurred in file scerr010.sai, line 31, column 2
Error: This floating point number has created an overflow.
Error occurred in file scerr010.sai, line 32, column 2
Panel Error & symbol has redefined the symbol name incase.
Error occurred in file scerr010.sai, line 33, column 2
Error: This reference parameter failed must match the same type of the reference argument.
Error: Line 95 Column 20 File: scerr040.sai
Error: This reference must be between or otherwise by reference.
Error: Line 96 Column 20 File: scerr040.sai

Error: An expression, the token `)' must be placed after the valuelinmary.
Error: While passing baddimension to functionnormalarray, the dimension(s) of baddimension did not match the dimension(s) of array as defined in the prototype for functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: A goto cannot jump to the label unplaced because it has not been placed.

Error: The label alreadyplaced has already been placed.

Error: The identifier undefined cannot be used until it is defined.

Error: The identifier alreadydefined cannot be used until it is defined.

Error: The label alreadydefined has already been used.

Error: The identifier alreadydefined has already been used.

Error: The identifier alreadydefined has already been used.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.

Error: An incorrect number of parameters were passed to functionnormalarray.
Error: Line: 264 Column: 3 File: scerr040.sai
Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock06 is in an inner block relative to the jumping goto statement.

Error: Line: 267 Column: 5 File: scerr040.sai
Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock07 is in an inner block relative to the jumping goto statement.

Error: Line: 268 Column: 2 File: scerr040.sai
Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock08 is in an inner block relative to the jumping goto statement.

Error: Line: 275 Column: 3 File: scerr040.sai
Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock09 is in an inner block relative to the jumping goto statement.

Error: Line: 276 Column: 3 File: scerr040.sai
Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock01 is in an inner block relative to the jumping goto statement.

Error: Line: 244 Column: 5 File: scerr040.sai
Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock02 is in an inner block relative to the jumping goto statement.

Error: Line: 251 Column: 3 File: scerr040.sai
Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock03 is in an inner block relative to the jumping goto statement.

Error: Line: 252 Column: 3 File: scerr040.sai
Error: A goto cannot jump from an outer code block into an inner code block.
The label innerblock04 is in an inner block relative to the jumping goto statement.

Error: Line: 261 Column: 4 File: scerr040.sai
Error: The function functionwithoutreturn must have a non-void return type to be used in an expression.

Error: Line: 290 Column: 34 File: scerr040.sai
Error: The function whoami must be called.

Error: Line: 297 Column: 12 File: scerr040.sai
Error: In an expression, the token '=' can't be placed in assignment.

Error: Line: 304 Column: 10 File: scerr040.sai
Error: Check that all curly braces match, because an EOF in the middle of a function was unexpectedly found.

Error: Line: 304 Column: 4 File: scerr040.sai
Error: You must define a main function.

Error: Line: 304 Column: 4 File: scerr040.sai
Error: The function main has been prototyped, but not defined.
--- SCERR050 errors  
Fatal Error: A constant expression attempted to divide by zero. The numerator was: 1.
--- SCERR060 errors  
Error: The function functionrecursion cannot call itself, directly or indirectly.
Error: Line: 40 Column: 6 File: scerr060.sai
--- SNSTERR errors  
Error: Calling depth may not exceed 32. While calling function function32 this limit was exceeded.
Error: Line: 447 Column: 10 File: snsterr.sai
--- SASMoverSizeLimit errors  
@Assembler errors
Code size exceeds 64K at line 65539

9.2.3 SAIL SPEC Test Case Data Output File
Generated from data 2003/156-19:52:00 to 2003/156-20:23:00
ValidateDumpAddr item = 5,Enabled_P1 = 1,Length_P2 = 258,Address_P3 = 0x80,Cyclic_P4 = 0
ValidateDumpAddr item = 14,Enabled_P1 = 1,Length_P2 = 3,Address_P3 = 0x0,Cyclic_P4 = 0