CAMS 10188
Process Escape PR LCA 4168 - GOAL Booster APU/Hydraulics program BAT04 did not execute per requirements

1st Problem Description:

PR LCA 4168 was initiated as a result of a problem that occurred during the afternoon run of the STS117/S0044 executed on April 18th 2007. The reported problem indicated that data did not appear to be updating on the CRT that was executing program BAT04 and sublevel program BAT03 once the countdown clock passed T-31 seconds. Normal operation is that the operator expects to see all of the vehicle data update as the Booster APUs start to run. This did not happen and data was not updated on the CRT during the final 30 seconds of the simulated countdown. Analysis revealed timing issues in the software due to logic timing in the software set that had been dramatically changed. This anomaly occurred because available execution time for existing console application programs was reduced due to the new program BAT06 executing in a previously empty concurrency. In this instance, the update loop did not encounter the line of code that determines if it is the correct time to enter the Rapid Update Mode between the T-30 to T-25 second time-period. When in Rapid Update Mode, BAT03 is to update all the Booster APU parameters that are displayed during the critical point when the Booster APUs are started.

At the time the software should have entered Rapid Update Mode, the software was in a sublevel program (BAT03) and did not return to BAT04 until T-28 seconds. At that time, the APUs were starting and measurement interrupts were being processed for the isolation valve states changing (measurement interrupts have a higher priority than the timer interrupts). When the timer interrupt was finally processed, the countdown clock was inside T-25 seconds. Therefore, when the software reached the statement checking to see if the countdown clock was between T-30 and T-25 seconds, since the clock value was already greater than these values, the software branched around the Rapid Update Logic section of code.

Root Cause:

- This is an inherent design problem in the BAT04 software introduced through changes made back in 1987. At that time, the potential existed that a series of measurement interrupts with higher priority than timer or operator interrupts could occupy the software during the T-30 sec to T-25 sec window, such that when the TIMER interrupt was actually fielded, the countdown clock could have progressed past the T-25 sec point. This design flaw was realized when the BAT06 software was developed to monitor the Aft Skirt Heated GN2 Purge Display (ESR K89459) and began executing in a previously non-used concurrency. This provided less execution time for BAT04 to process interrupts, for the concurrencies allotted time slice, from system software. The design flaw was further exasperated when ESR K89433 was implemented to reduce the RDA/IC for BAT04, as this caused more of the concurrency allotted time to be spent in sublevel program BAT03 such that BAT04 did not immediately field any measurement or TIMER interrupts that arrived while BAT03 was executing.

- Testing of the software change was inadequate at the Unit, Integrated and Formal test level. In reviewing test plans (both integrated and acceptance) for BAT06, neither had test steps where BAT06 and BAT04 were running concurrently in a launch configuration scenario. Thus no test runs were done with the BAT06 program that would reflect the new fully loaded console configuration. Had the launch configuration scenarios been included in integrated and acceptance testing, this might have revealed the code timing problems. Engineers performing software test are not putting the software in to configurations that will exist on station in various operations.

  Note: Three previous LPS Sw Process Escapes have pointed root cause towards inadequate Sw testing activities.

  - This specific problem was fixed by changing the when interrupt time on the countdown time from T-30 seconds to T-50 seconds and:
    1. Moving the inhibits of the 32 vehicle measurements to T-50 seconds
    2. Setting a new when interrupt time of T-50 Seconds
    3. Activate interrupt processing
  - The software will now inhibit processing on the isolation valves before they begin to change state and send interrupts to the program. This will allow more than sufficient time to recognize when the software needs to enter the Rapid Update Mode.

2nd Problem Description:

Second problem reported was that BAT04 did not control GG Bed Temperatures within the range of 205 to 230. At the end of OPS Transition, the Gas Bed Generator Temperatures on the Left SRB did not control the activation and deactivation of the heaters on the GG Bed. After reviewing data (ANACTC, GOAL trace, list changes and SPCATs) and the GOAL code, it was discovered the GG Bed temperature GOAL Exception limits after OPS transition ended did begin to control the heaters, just not right away. This is due to the widening of the offending side of the limit range by 5 degrees when an interrupt on the temperatures is processed. The software had just processed an interrupt and had moved the temperature range up to 248 degrees. The temperature at the time was 244 degrees, so the software did not receive an interrupt until the temperature exceeded 248 degrees. Verified by data retrievals, the software did turn off the heater once that threshold had been reached. The console operator performed several actions including going to manual control of the heaters and back to automatic mode. Also, the operator repeatedly cursored the auto heater mode several times in succession. This had no effect since the software was in auto mode and cursoring has no effect without going to manual mode first.

Root Cause:

- This appears to be a lack of understanding by the new System Engineers on how the software was designed to execute during an OPS Transition event. It appears only the ASWT Lead in the Shuttle Engineering group (no longer with the company) was aware of this design feature.

  - This issue was corrected by adding code logic in BAT04 to reset the GG Bed temperature's GOAL limits to 205 and 230 when PASS OPS Transition has been completed. This will move the control limits back to where the Operators expect them. If the temperatures are out of these limits, the heaters will be commanded to the correct state immediately, not after the temperatures have moved up another five degrees.

3rd Problem Description:

Third problem reported was that BAT04 did not respond to a PFPK6 command to terminate itself after the simulation was over. BAT04 did not terminate in response to this PFPK6 because higher level interrupts on the GG Bed temperatures were being processed over and over again. This was due to temperatures now above the upper limit for the transducers (500 degrees) and the exception limit in the FEP was being incremented in 5-degree intervals. These interrupts had a higher priority in the processing than the PFPK interrupt, thus, the PFPK interrupt was not responded to.

Root Cause:

- The software design allowed for measurement interrupts to continue to be activated and received for measurements that were outside of the expected limits. This, in essence, created an infinite loop in which the software was unable to process any operator requested action because the measurement interrupts have a higher priority and were being sent to the program from the FEP immediately after FEP Interrupt Processing was activated.

  - This problem was addressed by adding code logic to the T-30 second when interrupt target area to inhibit all program level interrupts and FEP interrupt checks on the GG Bed temperatures, since GLS turns off the commands prior to APU start (between T-31 and T-28 seconds).

Recommendations/Corrective Action Plan:

- Stronger emphasis must be placed on testing use cases. Software test configurations should represent the operational environment /scenarios in which they will be used as closely as possible in order to correctly verify the full nature of the software changes being implemented, as referenced in:
  - IDS-SEPG-058 IDS Organizational Software Process Section:
    2.5.2 Establish the Verification/Validation Environment
  - IDS-SEPG-113 Testing Practice Sections:

3.8.2.3.C Integrated Testing
3.8.2.5.B Acceptance Testing
4.4.2.4.B.5 Acceptance Testing Tasks

- IDS-SEPG-062 AppSw Project Plan Appendix:
  B.2 Testing Expectations
  B.3 Testing Environments
  B.4.5 Test Procedures
  B.5 Test Performance

- Software Verification Procedures should be more thoroughly examined to make sure these procedures fully test out all software modifications in the operational environment/scenarios in which they will be used.

- Inclusion of other potentially affected software systems / sets is of major concern in all software implementations that have the potential to impact another system.