Computer-Based Control System Safety Requirements

International Space Station Program

Revision B
November 17, 1995

National Aeronautics and Space Administration
International Space Station Program
Johnson Space Center
Houston, Texas
Contract No. NAS15-10000

REVISION AND HISTORY PAGE

REV.	DESCRIPTION	PUB. DATE
-	Initial Release to meet Product Group Statement of Work costing activities (Approved by NASA TPR)	04-04-94
A	Revision A per SSCN 000085 Interim release (Approved by NASA TPR)	09-07-00
B	Revision B per SSCN 000261 Eff. 09-16-96	09-07-00

ERU: /s/Beth Mason 9/7/00

PREFACE

Effective safety for International Space Station (ISS) dictates that effective control for computers, and the associated software and hardware, be established. The requirements specified herein are considered a minimum set of requirements for computer-based control of systems. Changes to this document will be controlled through the ISS formal change process.

The contents of this document are intended to be consistent with the tasks and products to be prepared by Program participants. Computer-Based Control System Safety Requirements shall be implemented on ISS participants and internal activities for flight computer based control systems and facilities for the development and maintenance of the flight software. This document will be under the control of the Space Station Safety and Mission Assurance Integrated Product Team.

/s/ J. Harold Taylor	11/28/95
J. Harold Taylor Manager, ISS Safety And Mission Assurance	Date
/s/ Jack E. Martin	11/28/95
Jack E. Martin Manager, Prime Safety And Mission Assurance	Date

INTERNATIONAL SPACE STATION PROGRAM
COMPUTER-BASED CONTROL SYSTEM
SAFETY REQUIREMENTS
November 17, 1995

CONCURRENCE

PREPARED BY:	/s/ Roger S. Chrostowski	11/27/95
	SIGNATURE	DATE
	Roger S. Chrostowski	Prime Safety
	PRINT NAME	ORGN
CHECKED BY:	/s/ Eric H. Clark	11/27/95
	SIGNATURE	DATE
	Eric Clark	SRP Engineer
	PRINT NAME	ORGN
CONCURRED:	/s/ M. G. Martin	11/27/95
	SIGNATURE	DATE
	Matt G. Martin	Manager,Prime Safety
	PRINT NAME	ORGN
CONCURRED:	/s/ Gregg J. Baumer	11/28/95
	SIGNATURE	DATE
	Gregg J. Baumer	Manager, IP Safety
	PRINT NAME	ORGN
CONCURRED:	/s/ Larry B. McWhorter	11/28/95
	SIGNATURE	DATE
	Larry McWhorter	Manager, Avionics
	PRINT NAME	ORGN
CONCURRED:	/s/ Kevin A. Klein	11/28/95
	SIGNATURE	DATE
	Kevin A. Klein	Manager, ISS SRP
	PRINT NAME	ORGN

INTERNATIONAL SPACE STATION PROGRAM
COMPUTER-BASED CONTROL SYSTEM
SAFETY REQUIREMENTS

LIST OF CHANGES

November 17, 1995

All changes to paragraphs, tables, and figures in this document are shown below:

SSP 50038 Revision B replaces all previous versions.

TABLE OF CONTENTS

PARAGRAPH PAGE

APPENDICES

APPENDIX PAGE

1.0 INTRODUCTION

Computer-based control systems use computer hardware and software as an integral part of the System Safety Program. Computer-based control system safety is the application of engineering and management principles, criteria, and techniques to provide hardware failure and software error tolerance to minimize risks associated with the use of computers to control hazards.

These requirements apply to computer-based flight systems that control flight system capabilities essential to the survival of the crew and the Space Station (this does not include simulation and training devices), and to the computer based control system software used in the prevention of catastrophic and critical hazardous events. This includes all flight software and firmware regardless of the media the software resides on.

Appendix A contains the CBCS design for minimum risk approach.

Appendix B contains abbreviations and acronyms used in this document.

Appendix C provides definitions.

Appendix D provides the implementation of SSP 50038 sections 3 and 4 requirements into the Segment Specification for the United States On-Orbit Segment, SSP 41162, and the lower tiered Prime Item Development Specifications.

2.0 APPLICABLE DOCUMENTS

The following documents are applicable to the extent specified herein:

DOCUMENT NO.	TITLE
SSP 30309	Safety Analysis Requirements

3.0 Computer Based Control System Safety Requirements

The purpose of section 3.1 is to define the requirements for computer based control of hazards. The approaches identified provide requirements which will implement the necessary and sufficient hazard controls. These approaches are based on the type of hazard being controlled and are to be applied on a hazard by hazard basis. Section 3.1 is the top level requirement that is decomposed into the requirements in the subordinate paragraphs. Section 3.1.1 contains general requirements which must be met in all CBCS designs. Section 3.1.2 contains requirements that must be met in the control of functions that must work in order for the ISS to be safe. Section 3.1.3 contains requirements for functions whose inadvertent operation would cause a hazard (i.e. must-not-work functions). Within Section 3.1.3, either the set of requirements in 3.1.3.1 or the set of requirements in 3.1.3.2 must be met in order to control the hazard.

3.1 System Level CBCS Safety Requirements

A CBCS shall provide hazardous function control where the inadvertent activation or deactivation of the function or capability could result in an identified critical or catastrophic hazard. (SSP 41000 3.3.6.3.2)

3.1.1 General CBCS Requirements

This section of the computer-based control system requirements must be applied to all CBCS designs irrespective of function.

3.1.1.1 The CBCS shall safely initialize to a known, safe state. (SSP 41000 3.3.6.3.1 c)

3.1.1.2 The CBCS shall perform an orderly shut down of a function to a known, safe state upon receipt of a termination command or detection of a termination condition. (SSP 41000 3.3.6.3.1 a)

3.1.1.3 A processor shall continue to operate safely during off-nominal power conditions, or contain design features which safe the processor during off-nominal power conditions.

3.1.1.4 Overrides shall require at least two independent actions by the operator.

3.1.1.5 Where execution of commands out of sequence can cause a hazard, the CBCS shall reject commands received out of sequence.

3.1.1.6 A CBCS shall detect and recover from inadvertent memory modification during use.

3.1.1.7 A CBCS shall recover to a known safe state upon detection of an anomaly within the CBCS.

3.1.1.8 The CBCS shall be capable of discriminating between valid and invalid inputs from sources external to the CBCS and remain or]recover to a known safe state in the event of an invalid external input.

3.1.1.9 All flight software shall be traceable to a system or software requirement.

3.1.1.10 All code shall be documented.

3.1.1.11 Integrity checks shall be performed when data or commands are exchanged across transmission or reception lines and devices.

3.1.1.12 The Space Station shall provide privacy for audio communications on the uplink/downlink, and protection for uplinked commands to prevent unauthorized third party control of the on-orbit station. (SSP 41000 3.3.9)

3.1.1.13 The CBCS shall reject hazardous commands which do not meet prerequisite checks for execution. (SSP 41000 3.3.6.3.1 b)

3.1.2 CBCS Must Work Function Requirements

The requirements of this section are applicable to the design of CBCS functions whose inadvertent shutdown would cause a hazard.

3.1.2.1 Fault Tolerant Approach

A computer–based control system shall be designed such that no combination of two failures, or two operator actions, or one of each will cause a catastrophic hazardous event, or no single failure or operator action will cause a critical hazardous event.

3.1.2.1.1 Where loss of a capability could result in a catastrophic hazard the CBCS shall provide two independent and unique command messages to deactivate any function within a failure tolerant capability.

3.1.2.1.2 Where loss of a capability could result in a critical hazard the CBCS shall provide two independent and unique command messages to deactivate the capability.

3.1.2.1.3 At least one independent operator action shall be required for each operator initiated command messages used in the shutdown of a capability or function that could lead to a hazard.

3.1.2.1.4 Where software provides the sole control for safety critical must work assembly functions, another non-identical method for commanding the function shall be provided.

3.1.2.1.5 Alternate or redundant functional paths shall be separate or protected such that any single credible event which causes the loss of one functional path will not result in the loss of the redundant functional path. (SSP 41000 3.2.3.4)

3.1.2.1.6 Respond to loss of function

The purpose of this capability is to respond, on-orbit, to the loss of system functions, which are required for 24 autonomous operations or that may manifest a catastrophic or critical hazard. The on-orbit Space Station shall automatically recover functional performance for those capabilities requiring automatic recovery, identified in Table III, column 3 of SSP 41000. The on-orbit Space Station shall automatically safe in less than the time to catastrophic of critical effect, any hazardous condition or functional operation that may, within 24 hours, manifest a catastrophic or critical hazard. (SSP 41000 3.2.1.1.1.4.)

3.1.3 CBCS Must-Not Work Function Requirements

The requirements of this section are applicable to the design of CBCS functions whose inadvertent operation would cause a hazard.

3.1.3.1 Fault Containment Approach

3.1.3.1.1 The CBCS shall perform prerequisite checks for the safe execution of hazardous commands.

3.1.3.1.2 A unique command message shall be required to enable the removal of inhibits.

3.1.3.1.3 Command messages to change the state of inhibits shall be unique for each inhibit.

3.1.3.1.4 For inhibits used to control hazards the CBCS shall make available to the crew and ground operators the status of monitored inhibits.

3.1.3.1.5 Where hazardous commands can be initiated by a hard-coded failure recovery automated sequence, a separate, functionally independent parameter shall be checked before issuance or execution of each hazardous command.

3.1.3.1.6 Where hazardous commands can be initiated by a hard-coded failure recovery automated sequence, at least one of the functionally independent parameters checked before issuance or execution of a hazardous command shall be operator controllable.

3.1.3.1.7 Each operator initiated command message used to remove an inhibit that controls a hazard shall be initiated by at least one independent operator action.

3.1.3.1.8 A CBCS shall make available to the crew or ground operators the status of software inhibits used to disable the execution of hazardous commands.

3.1.3.1.9 The CBCS shall make available to crew or ground operators the data, necessary and sufficient, for the performance of manual system safing for identified hazards.

3.1.3.1.10 A processor shall not independently control multiple inhibits to a hazard.

3.1.3.2 The Control Path Separation (CPS) Approach

3.1.3.2.1 For inhibits used to control catastrophic or critical hazards the CBCS shall make available to the crew or ground operators the status of monitored inhibits.

3.1.3.2.2 A computer–based control system shall have a separate control path (SCP) for each inhibit used to control a hazard.

3.1.3.2.3 Command messages to change the state of an inhibit shall be unique.

3.1.3.2.4 Each SCP initiated by an hard coded automated failure recovery sequence, shall include a check of at least one parameter functionally independent of the parameters checked by other SCPs initiated by the same sequence.

3.1.3.2.5 At least one functionally independent parameter checked by a SCP initiated by a hard coded automated failure recovery sequence shall be operator controllable.

3.1.3.2.6 Each operator initiated command message used to remove an inhibit that controls a hazard shall be initiated by at least one independent operator action.

3.1.3.2.7 For the control of a hazardous function, a computer–based control system shall use SCPs with different functionality for each inhibit used to control the hazard.

3.1.3.2.8 A CBCS shall make available to the crew or ground operators the status of software inhibits used to disable the execution of hazardous commands.

3.1.3.2.9 Capability: Monitor system status

The purpose of this capability is to acquire performance, configuration and status data from the on-orbit Space Station. The acquired data is assessed to determine station, failure, hazard or out-of-sequence events which require operator or automated action.

The on-orbit Space Station shall generate and collect data relating to the operational performance, configuration, status, failures and hazards of all on-orbit Space Station capabilities listed in Table III, column 1. The on-orbit Space Station shall automatically assess the collected data to detect failures of those capabilities requiring automatic assessment, identified in Table III, column 2, and to detect hazards that may exhibit a time to catastrophic or critical effect of less than 24 hours. (SSP 41000, 3.2.1.1.1.7)

4.0 Verification Requirements

4.1 RESERVED

4.2 RESERVED

4.3.0 Computer Based Control System Safety Requirements

4.3.1 System Level CBCS Safety Requirements

An analysis of lower level hazardous function control shall be performed per SSP 30309 and by separate system engineering analysis to identify the functions or capabilities where inadvertent activation or deactivation can result in a critical or catastrophic hazard. The analysis shall also identify those functions or capabilities which utilize a CBCS to control a hazard. The verification shall be considered successful when the analysis shows that the functions and capabilities identified contain, at a minimum, the required CBCS hazard controls.

4.3.1.1 General CBCS Requirements

4.3.1.1.1 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to show that during initialization, the CBCS hardware remains in a safe state, provides no spurious output signals, and completes to a known safe state. This verification shall be considered successful when the analysis shows that the CBCS hardware resides in the identified safe state upon completion of initialization, and that no spurious signals were generated throughout the initialization process (i.e., from power application or commanded initialization through initialization completion).

4.3.1.1.2 An analysis shall be performed per SSP 30309, and through separate system engineering analyses to identify CBCS functions and their termination commands or conditions. Analysis of lower level verifications shall identify the state the function enters upon termination and show that the defined state is safe. This verification shall be considered successful when the analysis shows that the identified safe state is the state into which each CBCS enters upon receipt of the termination command or condition.

4.3.1.1.3 Testing shall demonstrate that a computer–based control system continues to operate nominally in the presence of off–nominal power input conditions such that the computer–based control system does not allow erroneous or spurious commanding of hazardous function. If the computer–based control system does not operate nominally under off–nominal power conditions, then an analysis of the design features shall demonstrate that the computer–based control system is safed during these conditions.

4.3.1.1.4 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify commands which allow an operator to override prerequisite checking. The analysis shall identify the number of operator actions necessary to initiate each command to perform an override. This analysis verification shall be considered successful when the analysis shows that at least two separate operator actions are necessary to initiate the command(s) to perform an override.

4.3.1.1.5 An analysis of lower level verifications shall be performed per SSP 30309 and separate engineering analysis to identify those safety critical commands that must be executed in the correct sequence to prevent a hazard. The analysis shall show that out of sequence commands will be rejected. This verification shall be considered successful when it has been shown that the CBCS rejects out of sequence commands for safety critical commands that must be executed in the correct sequence.

4.3.1.1.6 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify the memory areas within a CBCS which are used to store code and adaptation data. Analysis shall show that the CBCS can detect and recover from inadvertent memory modification of stored code and adaptation data. This verification shall be considered successful when the analysis shows that the CBCS can detect and recover from inadvertent memory modification of stored code and adaptation data.

4.3.1.1.7 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify detectable CBCS anomalies. The analysis shall show that detected anomalies are recovered to a known safe state. This verification shall be considered satisfied when the analysis has shown that detected anomalies are recovered to a known safe state.

4.3.1.1.8 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify the valid safety related inputs to the CBCS. The analysis shall show that, in the presence of invalid external inputs, the CBCS will either remain in or recover to a known safe state. This verification shall be considered satisfied when the analysis has shown that the CBCS either remains in or recovers to a known safe state in the presence of invalid external inputs.

4.3.1.1.9 Verification of this requirement shall be by inspection of the software development standards used by the software developer. This verification shall be considered successful when the inspection shows that the software development standards require traceability of flight software to system or software requirements.

4.3.1.1.10 Verification of this requirement shall be by inspection of the software development standards used by the software developer. This verification shall be considered successful when the inspection shows that the software development standards require the documentation of all code.

4.3.1.1.11 An analysis shall be performed per SSP 30309 and separate system engineering analyses to identify the checks on data or commands which must be performed to ensure the quality of data or command transmission between processors. The analysis shall show that invalid inputs which are the result of poor transmission or reception lines or devices are rejected. This verification shall be considered successful when the analysis has shown that invalid inputs which are the result of poor transmission or reception lines or devices are rejected.

4.3.1.1.12 Verification of this requirements shall be by analysis of the applicable segment qualification results. The qualification shall be considered successful when the applicable segment level test, demonstration, analysis or inspection requirement are shown to be satisfied.

4.3.1.1.13 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to characterize the initialization process for CBCS components. This analysis shall show that during initialization, the CBCS hardware produces no spurious output signals, and completes to a known safe state. This verification shall be considered successful when the analysis shows that the CBCS hardware resides in the identified safe state upon completion of initialization, and that no spurious signals were generated throughout the initialization process (i.e., from power application or commanded initialization through initialization completion).

4.3.1.2 CBCS Must Work Function Requirements

4.3.1.2.1 Fault Tolerant Approach

An integrated analysis of the verifications required in this section for the fault tolerance approach shall be performed per SSP 30309 and separate system engineering analyses to show that no combination of one or two failures or operator actions or one of each will cause either a catastrophic or critical hazardous event. The verification shall be considered successful when the analysis shows that compliance with requirements of this section, as shown through successful verifications, has been accomplished and that no combination of two failures, two operator actions, or one of each will cause a catastrophic hazardous event, or no single failure or operator action will cause a critical hazardous event.

4.3.1.2.1.1 An analysis shall be performed per SSP 30309 and by separate system engineering analysis to identify the redundant functions within a capability whose loss could result in a catastrophic hazard, the hardware which supports the redundant functions, the command string which controls the function and the commands that could result in deactivation of the function. An analysis shall be conducted to show that no single command message can result in the deactivation of a function within a failure tolerant capability. Verification shall be considered successful when it has been shown that tow independent and unique commands are required to deactivate a redundant function in a failure tolerant capability.

4.3.1.2.1.2 An analysis of lower level verifications shall be performed per SSP 30309 and separate system engineering analyses to identify capabilities whose loss could cause a critical hazard. The analysis shall identify that two unique and independent command messages are required to command the shutdown of the capability. This verification shall be considered successful when the analysis shows that at a minimum, two unique and independent command messages are required to command the shutdown of the capability.

4.3.1.2.1.3 Analysis of lower level verification shall be performed per SSP 30309 and separate systems engineering analyses to identify the process and algorithms used to translate operator actions into command messages and that subsequently release the command message. Analysis shall be conducted to show that every operator initiated command message has at least one corresponding independent operator action. Verification shall be considered successful when it has been shown that an operator initiated command message can only be initiated by at least one independent operator action.

4.3.1.2.1.4 Analysis of lower level verifications shall be performed per SSP 30309 and separate engineering analysis to identify where software provides the sole control for safety critical must work assembly functions. An additional analysis showing that a command function exists for the initiation of assembly must-work functions that is non-identical to the primary command function. This verification shall be considered successful when the analyses show that at least one command function exists for initiating the assembly must work function which is not identical to the primary command function.

4.3.1.2.1.5 The separation of redundant paths requirement shall be verified by an inspection of engineering drawings to ensure spacing requirements developed by engineering analysis of single credible events have been implemented in the design. The separation of redundant paths verification shall be successful when drawing inspection demonstrates that the design meets the separation requirements for all single credible events.

4.3.1.2.1.6 Respond to loss of function

Recovery, from loss of functions listed in Table III which are required for 24 hour autonomous operation, shall be verified by an integrated failure recovery analysis. The analysis shall evaluate each function listed in Table III which is required for 24 hour autonomous isolation and recovery using data from Reliability Block Diagrams (RBDs), Failure Modes Effects and Criticality Analysis (FMECA), Integrated Program Command List (IPCL), schematics and software detailed design documents. The analysis will also be supported by testing conducted at the Space Station Verification and Training Facility (SSVTF). The SSVTF testing shall simulate automatically for each function listed in Table III which is required for 24 hour autonomous operation. The requirement will be considered satisfied when the analysis, supported by test data, shows that after failures that result in loss of functions identified in Table III which are required for 24 hour autonomous operation, the on-orbit Space Station automatically; (1) isolates the failure to the recovery level, (2) recovers functional operations, and (3) confirms that the function has been restored.

Analysis: Safing, for hazardous functional operation or out-of-tolerance conditions, shall be verified by an integrated failure safing analysis. The analysis shall evaluate identified hazardous conditions that, due to functional operation or out-of-tolerance condition, may manifest a catastrophic or critical hazard within 24 hours. The analysis will use data from Hazard Analysis Reports, FMECA, IPCL, schematics and software detailed design documents. The analysis will also be supported by testing conducted at the SSVTF. The SSVTF testing shall simulate hazardous functional operation or out-of-tolerance conditions that require automatic safing in less time than the time to catastrophic or critical effect to show that the system will isolate and safe automatically for each identified hazardous condition that may manifest a catastrophic or critical hazard within 24 hours.

The requirement will be considered satisfied when the analysis, supported by test data, shows that for functional operation or out-of-tolerance conditions that may manifest a catastrophic or critical hazard within 24 hours, the on-orbit Space Station automatically; (1) isolates to the safing level, (2) safe the hazardous condition, and (3) confirms that the hazardous condition has been safed.

4.3.1.3 CBCS Must-Not Work Function Requirements

4.3.1.3.1 Fault Containment Approach

4.3.1.3.1.1 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify the prerequisite checks that must be met for the safe execution of hazardous commands. This verification shall be considered successful when the analyses show that prerequisite checks are provided and satisfied prior to execution of hazardous commands.

4.3.1.3.1.2 An analysis of lower level verifications shall be performed per SSP 30309 and separate systems engineering analysis to identify all system inhibits, the commands that enable the removal of system inhibits and show that the identified commands are unique. This verification shall be considered successful when it is shown that all inhibit removal commands can be enabled and disabled, and that the enabling commands are unique.

4.3.1.3.1.3 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify safety inhibits and identify the command messages necessary to remove each inhibit. The analysis shall show that no single command message can remove more than one inhibit or can place an inhibit in more than one state (i.e., will not toggle an inhibit). This verification shall be considered successful when it is shown that a single command message does not remove more than one inhibit or place a single inhibit in more than one state.

4.3.1.3.1.4 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify all monitored inhibits. The analysis shall identify the parameters available for monitoring the status of these inhibits. This verification shall be considered successful when the analysis shows that the status of each monitored inhibit is available to the operator.

4.3.1.3.1.5 Analysis of lower level verifications shall be performed per SSP 30309 and separate engineering analysis to identify each hard coded failure recovery automated sequence which can initiate hazardous commands and identify which hazardous commands can be initiated by the sequence and identify the functionally independent parameter which must be checked before the execution of each hazardous command. The analysis shall also show that the parameter checked before execution of one hazardous command is functionally independent from the parameters checked for the other hazardous commands initiated by a hard coded automated sequence. The analysis shall show that the functionally independent parameters being checked are those parameters for the initiation or prevention of the automated sequence. This verification shall be considered successful when the analysis shows that the parameters checked are functionally independent and that the parameters are checked before the execution of each hazardous command.

4.3.1.3.1.6 Analysis of lower level verifications shall be performed per SSP 30309 and separate engineering analysis to identify each hard coded failure recovery automated sequence which can initiate hazardous commands and identify which hazardous commands can be initiated by the sequence and identify the functionally independent parameter which must be checked before the execution of each hazardous command. The analysis shall show that one of the parameters checked is operator controllable. This verification shall be considered successful when the analysis shows that one of the functionally independent parameters checked before execution of the hazardous commands is operator controllable.

4.3.1.3.1.7 Analysis of lower level verification shall be performed per SSP 30309 and separate systems engineering analyses to identify the process and algorithms used to translate operator actions into command messages and that subsequently release the command message. Analysis shall be conducted to show that every operator initiated command message that removes an inhibit has at least one corresponding independent operator action. Verification shall be considered successful when it has been shown that an operator initiated command message that removes an inhibit can only be initiated by at least one independent operator action.

4.3.1.3.1.8 An analysis of lower level verifications shall be performed per SSP 30309 and separate engineering analysis to identify software inhibits used to disable the execution of hazardous commands. The analysis shall show that inhibit status is available to the operator. This verification shall be considered successful when the analysis shows that the status of software inhibits is available to the operator.

4.3.1.3.1.9 An analysis shall be performed per SSP 30309 to identify system hazards which may require manual safing. An analysis of lower level verifications and separate engineering analysis shall identify the data required by the operator to identify the hazard and to perform manual system safing. This verification shall be considered successful when the analysis has shown that the data identified can be made available to the operator.

4.3.1.3.1.10 An analysis of lower level verifications shall be performed per SSP 30309 and separate engineering analysis to identify the processor or processors that control those inhibits to a hazard. This verification shall be considered successful when the analysis shows that all inhibits to a hazard are not controlled by the same processor or shows that the processor does not independently control more than one of the system hazard controls.

4.3.1.3.2 The Control Path Separation (CPS) Approach

4.3.1.3.2.1 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify all monitored inhibits. The analysis shall identify the parameters available for monitoring the status of these inhibits. This verification shall be considered successful when the analysis shows that the status of each monitored inhibit is available to the operator.

4.3.1.3.2.2 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify the control path used to control each inhibit and the functionality of the control path. The analysis shall show that for a single hazard, each inhibit controlled by a single CBCS processor is controlled by a separate control path within that processor and that each control path provides different functionality such that common cause failures are prevented. This verification shall be considered successful when the analysis shows that each inhibit for a given hazard is controlled by a control path which provides functionality which is different from the control path(s) controlling the other inhibits to the hazard within the CBCS processor and that one control path cannot cause the execution of another control path or control an inhibit belonging to another control path for the hazard.

4.3.1.3.2.3 An analysis shall be performed per SSP 30309 and by separate system engineering analysis to identify safety inhibits and identify the command messages necessary to remove each inhibit. The analysis shall show that no single command message can remove more than one inhibit or can place an inhibit in more than one state (i.e., will not toggle an inhibit). This verification shall be considered successful when it is shown that a single command message does not remove more than one inhibit or place a single inhibit in more than one state.

4.3.1.3.2.4 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify each hard coded automated sequence which initiates hazardous commands, the hazardous commands which are initiated, and the parameters which are checked before the initiation or execution of each of these commands. The analysis shall show that at least one of the parameters checked for one hazardous command is functionally independent from the parameters checked for the other hazardous commands initiated by the same hard coded automated sequence. This verification shall be considered successful when the analysis shows that the parameters checked are functionally independent and that the parameters are checked before the initiation or execution of each hazardous command.

4.3.1.3.2.5 Analysis of lower level verifications shall be performed per SSP 30309 and separate engineering analysis to identify each hard coded automated failure recovery sequence which can initiate SCPs and identify which SCPs can be initiated by the sequence. The analysis shall show that .one of the parameters checked is operator controllable. This verification shall be considered successful when the analysis shows that one of the functionally independent parameters checked by the SCPs is operator controllable

4.3.1.3.2.6 I. An analysis shall be performed per SSP 30309 and by separate system engineering analysis to identify operator initiated commands which could cause a hazard or perform an override. The analysis shall identify the number of separate operator actions required to release each command message. This verification shall be considered successful when the analysis shows that each identified command message can only be released by at least one separate operator action.

4.3.1.3.2.7 An analysis of lower level verifications shall be performed per SSP 30309 and by separate system engineering analysis to identify the control path used to control each inhibit and the functionality of the control path. The analysis shall show that for a single hazard, each inhibit controlled by a single CBCS processor is controlled by a separate control path within that processor and that each control path provides different functionality such that common cause failures are prevented. This verification shall be considered successful when the analysis shows that each inhibit for a given hazard is controlled by a control path which provides functionality which is different from the control path(s) controlling the other inhibits to the hazard within the CBCS processor and that one control path cannot cause the execution of another control path or control an inhibit belonging to another control path for the hazard.

4.3.1.3.2.8 An analysis of lower level verifications shall be performed per SSP 30309 and separate engineering analysis to identify software inhibits used to disable the execution of hazardous commands. The analysis shall show that inhibit status is available to the operator. This verification shall be considered successful when the analysis shows that the status of software inhibits is available to the operator.

4.3.1.3.2.9 Monitor system status shall be verified by an integrated BIT Effectivity analysis. The analysis evaluate each capability in Table III, using data from Hazard Analysis Reports, RBDA, FMEA, schematics drawings, and software detailed design documents. Test results from the verification facilities shall be used to support this analysis. The verification facilities testing shall simulate a subset of the capabilities to show that the System: (1) generates and collects performance, configuration, status, failure, and hazard data for the capabilities listed in Table III, column 1, and (2) automatically assesses performance, configuration, status, failure, and hazard data for the capabilities identified in Table III, column 2.

The requirement will be considered satisfied when the analysis, supported by test data, shows that the system: (1) for capabilities listed in Table III, column 1, generates and collects performance, configuration, status, failure, and hazard data for use by the ISS, and (2) for capabilities identified in Table III, column 2, automatically assesses performance, configuration, status, failure, and hazard data.

APPENDIX A CBCS DESIGN FOR MINIMUM RISK

A.1 CBCS DESIGN FOR MINIMUM RISK

It is acceptable for a unique set of computer based control system requirements to be used for the control of hazards, provided these requirements are reviewed by the SRP and found acceptable. The SRP will use appendix A as a tool to assess the acceptability of these requirements. Each item in appendix A shall be addressed with either compliance, or an explanation why an item does not apply. The following is a listing of the top level items:

a) Separation of Commands/Functions/Files/Ports

b) Interrupts

c) Shutdown/Recovery/Safing

d) Preventing/Precluding/Disallowing Actions

e) Memory/Storage/Data Transfer

f) Verification/Validation Checks

g) Logic Structure/Unique Codes/Interlocks

h) Monitoring/Detection

I) Reasonableness Checks

j) Initialization/Timing/Sequencing/Status Checking

k) Operator Responses/Limitations

l) Operator Notification

m) General/Miscellaneous

The SRP will use this unique set of requirements to assess compliance for the applicable hazards. The element integrator must determine which set of safety requirements will apply to a particular hazard.

A.2 CBCS DESIGN FOR MINIMUM RISK CHECKLIST

- **Separation of Commands/Functions/Files/Ports**

  Provides for using separate authorization and separate control functions to initiate a critical function

  Provides for requiring separate arm" and fire" commands for critical capabilities.

  Precludes using input/output ports for both critical and non-critical functions

  Provides for sufficient difference in addresses for critical input/output ports versus non-critical ports that a single address bit failure does not allow access to critical functions or ports

  Provides for having files that are unique and have a single purpose

  Provides for consistent inter-CSCI interfaces

- **Interrupts**

  Provides for defining specific interrupt priorities and responses

  Provides for software system management of interrupt control so as not to compromise safety-critical operations

  Provides for a fail safe recovery from inadvertent instruction jumps

- **Shutdown/Recovery/Safing**

  Shutdown provisions are included in software upon detection of unsafe conditions

  Provides for the system reverting to a known predictable safe state upon detection of an anomaly

  Provides for software safing of safety-critical hardware items

  Provides for an orderly system shutdown as the result of a command shutdown, power interruptions, or other failures

  Requires that the software be capable of discriminating between valid and invalid external interrupts and shall recover to a safe state in the event of an erroneous external interrupt

Provides for entry into a safe state in the event of erroneous entry into a critical routine

Protects against out-of-sequence transmission of safety-critical function messages by detecting any deviation from the normal sequence of transmission. When this condition is detected, the software terminates all transmissions, recycles to a known safe state, and displays the existing status so the operator can take compensatory action

Provides for initializing all unused memory locations to a pattern, that if executed as an instruction, will cause the system to revert to a known safe state

Provides for identifying safing scenarios for safety-critical hardware and including them into the decision logic

Provides for the capability of reversing or terminating authorization functions

Provides for preventing inadvertent generation of critical commands

Provides for disallowing coexistence of potentially hazardous routines

- **<u>Preventing/Precluding/Disallowing Actions</u>**

Provides for preventing bypass of safety devices during test

Following computer memory loading, automatic control is prevented until all data is loaded and verified

Precludes inadvertent operation of data entry control to critical routines

Provides for precluding a change in state if data synchronization is lost

Provides for prevention of hardware failure or power interruption from causing a memory change

Provides for prevention of memory alteration or degradation over time during use

Provides for program protection against unauthorized changes

Provides for not allowing the safety-critical time limits in decision logic to be changed by the console operator

Provides for preventing inadvertent entry into a critical routine

Provides for not allowing a hazardous sequence to be initiated by a single keyboard entry

Prohibits transmission of any critical command found to be in error and notifies the operator of the error

Provides that controlling or monitoring of catastrophic actions be incapable of bypassing operator control of safety-critical functions

Provides for disallowing use of work around procedures when reverting to a safe configuration after the detection of an anomaly

Provides for not using a _stop" or _halt" instruction or causing a CPU _wait" state. The CPU is always executing, whether idling with nothing to do or actively processing

Provides for detection and termination of commands requesting actions beyond the performance capability of the system

Provides for disallowing performance of a potentially hazardous routine concurrently with a maintenance action

- **<u>Memory/Storage/Data Transfer</u>**

Provides for self-test capability to assure memory integrity

Provides for prevention of a hardware failure or power interruption from causing memory alteration

Provides for prevention of memory alteration or degradation over time during use

Provides for limiting control access to storage devices/ memory

Provides for protecting the accessibility of memory regions dedicated to critical functions

Provides for having safety-critical operational software instructions resident only in nonvolatile read-only memory

Provides for not using scratch files for storing or transferring safety-critical information between computers

Provides that remote transfer of data cannot be accomplished until verification of data to be transferred is accomplished and authorization to transfer the data has been provided by the operator(s)

- **<u>Verification/Validation Checks</u>**

When a test specifies for the removal of safety interlocks, the software provides for verification of reinstatement of these safety interlocks at the completion of the testing

Provides for verification and validation of status flags

Requires that critical data communicated from one CPU to another be verified prior to operational use

Provides for software validation of critical commands

Provides for verification of the existence of prerequisite conditions prior to command issuance in accordance with predefined operational requirements

Provides for verification of the results of safety-critical algorithms prior to use

Provides for verification of safety-critical parameters or variables before an output is allowed

Decisioning verifies the sequence and logic of all safety-critical command messages and rejects commands when sequence or logic is incorrect

Provides that remote transfer of data cannot be accomplished until verification of data to be transferred is accomplished and authorization to transfer the data has been provided by the operator(s)

Provides that all operator actions that set up safety-critical signals are verified by software based on control device positions

Provides for control of analog functions having feedback mechanisms that provide positive indications of the function having occurred

Provides for verification and validation of the prompt for the initialization of a hazardous operation or sequence of hazardous operations

Provides for verification of accomplishment of each step of a hazardous operation, or sequence of hazardous operations, by setting of a dedicated status flag prior to proceeding to and initiating the next step in the operation or series of operations

Provides for verification/validation of all critical commands prior to transmission

- **Logic Structure/Unique Codes/Interlocks**

Provides for identification of flags to be unique and single purpose

Provides for using unique arming codes to control critical safety devices

Provides for inclusion of system interlocks

Provides for using a minimum of two separate independent commands to initiate a safety-critical function

Provides for the majority of safety-critical decisions and algorithms to be contained within a single (or few) software development module(s)

Provides for single CPU control to be incapable of satisfying all of the requirements for initiation of a process if the process can result in major system loss, system damage, or loss of human life

Requires that decision logic using data which obtain values from end-item hardware and software not be based on values of all _ones" or all _zeroes"

Requires that decision logic using data which obtain values from end-item hardware and software use specific binary patterns to reduce the likelihood of malfunctioning end-item hardware/software satisfying the decision logic

Provides for having safety-critical modules with only one entry and one exit point

Provides for having files that are unique and have a single purpose

Provides for not having operational program loads contain unused executable code

- **<u>Monitoring/Detection</u>**

Provides for inclusion of monitoring of safety devices

Provides for detection of inadvertent computer character outputs

Provides for detection of errors during computer memory loading to terminal loading process

Provides for detection of unauthorized operation of data entry control

Provides for identification of safety-critical functions requiring continuous monitoring

Provides for detection of improper processing that could degrade safety

Provides for detection of a fault having the potential of degrading safety

Provides for detecting a predefined safety-critical anomaly and informing the operator what action was taken

Requires that the software be capable of discriminating between valid and invalid external interrupts and shall recover to a safe state in the event of an erroneous external interrupt

Provides for detection of improper sequence requests by the operator

Provides for detection of inadvertent transfer to safety-critical routines

Provides for detection and termination of commands requesting actions beyond the performance capability of the system

- **Reasonableness Checks**

Provides for software system reasonableness checks of all safety-critical inputs

Provides for performing parity or other checks, requiring two decisions, before providing an output

- **Initialization/Timing/Sequencing/Status Checking**

Provides for a status check of critical system elements prior to executing a potentially hazardous sequence

Provides the proper configuration of inhibits, interlocks, safing logic, and exception limits at initialization

Provides for issuance of good guidance signal subsequent to satisfaction of performance of flight safety checks

Provides for timing sufficiency of commands relative to response to detect unsafe conditions

Provides for software initialization to a known safe state

Provides for performing a status check of safety-critical elements prior to executing a potentially hazardous sequence

Provides that all critical timing relative to hazardous operations processing is automated

Provides for employing time limits for operations impacting system safety and having these time limits included in decision logic

Protects against out-of-sequence transmission of safety-critical function messages by detecting any deviation from the normal sequence of transmission. When this condition is detected, the software terminates all transmissions, recycles to a known safe state, and displays the existing status so the operator can take compensatory action

Provides for initializing all unused memory locations to a pattern, that if executed as an instruction, will cause the system to revert to a known safe state

Applies use of software timing coincident with hardware timing to prevent initiation of safety-critical functions

Provides for verification and validation of the prompt for the initialization of a hazardous operation or sequence of hazardous operations

Provides for verification of accomplishment of each step of a hazardous operation, or sequence of hazardous operations, by setting of a dedicated status flag prior to proceeding to and initiating the next step in the operation or series of operations

- **<u>Operator Responses/Limitations</u>**

Requires an operator response for initiation of any potentially hazardous sequence

Provides for not allowing the safety-critical time limits in decision logic to be changed by the console operator

Provides for concise definition of operator interactions with the software

Provides for operator cancellation of current processing in a safe manner

Requires that an operator cancellation of current processing be verified by an additional operator response

Provides that controlling or monitoring of catastrophic functions be incapable of bypassing operator control of safety-critical functions

Requires that the system responds to predefined safety- critical anomalous conditions by notifying the operator of the condition and identifying the action taken

Provides that upon safing the system, the resulting system configuration or status be provided to the operator and await definition of subsequent software activity

Provides that remote transfer of data cannot be accomplished until verification of data to be transferred is accomplished and authorization to transfer the data has been provided by the operator(s)

Provides that operator control of safety-critical functions is maintained under all circumstances

Provides that all manual actions that set up safety-critical signals are verified by software based on control device positions

- **Operator Notification**

Requires that an override of a safety interlock be identified to the test conductor by a display on the test conductor's panel

Provides for generation of critical status to operator

Provides to operator identification of overrides to safety interlocks

Provides for software indication if unauthorized action has taken place

Provides for the system informing the operator of the anomaly detected

Provides system configuration status to operator upon safing of safety-critical hardware items

Provides for positive reporting of changes of safety-critical states, e.g., absence of an armed indication does not constitute a safe condition

Provides for detecting a predefined safety-critical anomaly and informing the operator what action was taken

Provides for the software system to display safety-critical timing data to the operator

Provides for the software systems to indicate to the operator the currently active operation(s) and function(s)

Provides for identification to the operator that a safing function execution has occurred; provides the reason for the execution with a description of the safing action taken

Provides for notification of improper keyboard entries by the operator

Prohibits transmission of any critical command found to be in error and notifies the operator of the error

Provides that upon safing the system, the resulting system configuration or status be provided to the operator and await definition of subsequent software activity

- **General/Miscellaneous**

Provides for precluding dependence on administrative procedures

Provides for using information control concept for deriving the authorization code for the activation of the authorization device

Provides that the software contains only features or capabilities required by the system, and that it does not contain additional capabilities, e.g., testing, troubleshooting, etc.

Provides for positive control of system safety-critical functions at all times

Provides for safety-critical subroutines and subprograms to include Come From" checks to verify that they are being called from a valid calling program

APPENDIX B ABBREVIATIONS AND ACRONYMS

B.1 ABBREVIATIONS AND ACRONYMS

APM	Attached Pressurized Module
ASA	Auxiliary Solar Array
CBCS	Computer Based Control System
CBCSSTT	Computer Based Control System Safety Task Team
CPS	Control Path Separation
DDCU	Direct Current to Direct Current Converter Unit
DISU	Dual Input Switching Unit
ESA	European Space Agency
EVA	Extravehicular Activity
FDIR	Failure Detection, Isolation, and Recovery
GN&C	Guidance, Navigation, and Control
HAB	Habitation Element
IP	International Partner
IPT	Integrated Product Team
ISS	International Space Station
ITS	Integrated Truss Segment
JEM	Japanese Experiment Module
MDM	Multiplexer Mating Adapter
MSS	Mobile Servicing System
NSTS	National Space Transportation System
ORGN	Organization
ORU	Orbital Replacement Unit
PDMA	Pressurized Docking Module Adapter
PG	Product Group
PMA	Pressurized Mating Adapter
POST	Power On Self Test
RACU	Russian American Converter Unit
RBD	Reliability Block Diagram
RPCM	Remote Power Control Mechanism

S&MA	Safety and Mission Assurance
SMC	Station Management and Control
SRP	Safety Review Panel
SSWG	Software Safety Working Group
UPS	Uninterruptible Power Supply
US	United States
USGS	United States Ground Segment

APPENDIX C DEFINITIONS

C.1 DEFINITIONS

Anomaly - A state or condition which is not expected. It may be hazardous but it is the result of a transient hardware or coding error.

Catastrophic Hazard - Any condition which may cause a disabling or fatal personnel injury, or cause loss of one of the following: the Orbiter, ISS or major ground facility. Loss of ISS: Loss of the ISS is to be limited to those conditions resulting from failures or damages to elements in the critical path of the ISS that render the ISS unusable for further operations, even with contingency repair or replacement of hardware, or which render the ISS in a condition which prevents further rendezvous and docking operations with ISS launch elements.

Command Message - The structure of bits within a CBCS that represent an operator or system initiated command.

Computer Based Control System (CBCS) - A control system which utilizes computer hardware, software and or firmware which accepts input information, and processes that information to provide outputs to perform a defined task.

Computer Hardware - Devices capable of accepting and storing computer data, executing a systematic sequence of operations on computer data, or producing control outputs. Such devices can perform substantial interpretation, computation, communication, control, or other logical functions.

Computer Program - A combination of computer instructions and data definitions that enable computer hardware to perform computational or control functions.

Control Path - The logical sequence of flow of a control or command message from the source to the implementing effector or function. A control path may cross the boundaries of two or more computers. Portions of multiple control paths may exist in a single computer.

Credible Failure - A condition that has a potential of occurring based on actual failure modes in similar systems.

Critical Hazard - Any condition which may cause a non-disabling personnel injury, severe occupational illness; loss of a ISS element, on-orbit life sustaining function or emergency system; or involves damage to the Orbiter or a major ground facility. For safety failure tolerance considerations, critical hazards include loss of ISS elements that are not in the critical path for station survival or damage to an element in the critical path which can be restored through contingency repair.

Database - A collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via a database management system.

Design for Minimum Risk - Design for minimum risk are areas where hazards are controlled by specification requirements rather than failure tolerance. Examples are structures, pressure vessels, pressurized lines and fittings, functional pyrotechnic devices, material compatibility, flammability, etc.

Error Handling - An implementation (system or compiler feature) mechanism or design technique by which software faults are detected, isolated and recovered to allow for correct runtime program execution.

Firmware - The combination of a hardware device and computer instructions and/or computer data that reside as read-only software on the hardware device.

Hazard - The presence of a potential risk situation caused by an unsafe act or condition. A condition or changing set of circumstances that presents a potential for adverse or harmful consequences; or the inherent characteristics of any activity, condition or circumstance which can produce adverse or harmful consequences.

Hazardous Command - A command that can create an unsafe or hazardous condition which potentially endangers the crew or station safety. It is a command whose execution can lead to an identified hazard or a command whose execution can lead to a reduction in the control of a hazard.

Hazard Controls - Design or operational features used to reduce the likelihood of occurrence of a hazardous effect. Hazard controls are implemented in the following order of precedence:

A. Elimination of hazards by removal of hazardous sources and operations by appropriate design measures.
B. Prevention of hazards through the use of safety devices or features
C. Control of hazards through the use of warning devices.
D. Special procedures and/or emergency devices.
E. Minimization of hazards through a maintainability program and adherence of adequate maintenance and repair schedule(s).

Independent Command - Two or more commands are independent if no single credible failure, event or environment can eliminate more than one command from performing its intended function.

Independent Inhibit - Two or more inhibits are independent if no single credible failure, event or environment can eliminate more than one inhibit.

Independent Parameter - A parameter is independent from an automated function if a) neither the fault(s) that could initiate the automated function nor b) the operation of the automated function itself can change the value or status of the parameter.

Independent Safing Action - Safing actions are independent if no single fault can prevent one or more of the safing actions from transitioning the system to a safe state.

Inhibit - A design feature that provides a physical interruption between an energy source and a function (e.g., a relay or transistor between a battery and a pyrotechnic initiator, a latch valve between a propellant tank and a thruster, etc.). Note: Software inhibits are not counted in meeting safety requirements for multiple inhibits.

Interlock - A design feature that ensures that any conditions prerequisite for a given function or event are met before the function or event can proceed.

Near Real Time Monitoring - Notification of changes in inhibit or safety status on a periodic basis (nominally once per orbit) (From: NSTS 1700.7B, Paragraph 201.1c(1).

Operator Error - An inadvertent action by flight crew or ground operator that could eliminate, disable, or defeat an inhibit, redundant system, containment feature, or other design features that is provided to control a hazard.

Override - The forced bypassing of prerequisite checks on the operator-commanded execution of a function. Execution of any command (whether designated as a “hazardous command” or not) as an override is considered to be a hazardous operation requiring strict procedural controls and operator safing.

Prerequisite Checks - The validation by the CBCS that coded states or conditions necessary for the execution of a command has been met.

Real Time Monitoring - Notification of changes in inhibit or safety status to the crew (From: NSTS 1799.7B, Paragraph 201.1c(2)

Reflown Hardware - Payloads or elements of payloads which are made up of hardware items that have already physically flown on the Orbiter and are being manifested for reflight.

Risk - Exposure to the chance of injury or loss. Risk is a function of the possible frequency of occurrence of an undesirable event, of the potential severity of the resulting consequences, and of the uncertainties associated with the frequency and severity.

Safe - A general term denoting an acceptable level of risk, relative freedom from and low probability of: personal injury; fatality; loss or damage to vehicles, equipment or facilities; or loss or excessive degradation of the function of critical equipment.

Safety Critical - A condition, event, operation, process, function, equipment or system (including software and firmware) with potential for personnel injury or loss, or with potential for loss or damage to vehicles, equipment or facilities, loss or excessive degradation of the function of critical equipment, or which is necessary to control a hazard.

Safety Critical Software - Software which:

A. Exercises direct command and control over the condition or state of hardware components or software functions and, if not performed or if performed out of sequence or incorrectly, could result in control function loss or error which could cause a hazard

B. Monitors the condition or state of hardware components and, if monitoring is not performed or is performed incorrectly, could provide data which results in erroneous operator or companion system decisions which could cause a hazard.

C. Exercises direct command and control over the condition or state of hardware components or software functions and, if not performed or if performed out of sequence or incorrectly in conjunction with human error or hardware failure, could cause a hazard.

Separate Control Path (SCP) - A control path which provides functional independence to a command used to control an inhibit to an identified critical or catastrophic hazard. Functional independence exists when no other control path exists which can remove a hazard's inhibit belonging to this SCP. SCPs controlling different inhibits for the same hazard may co-exist within the same processor.

Software - Computer programs and computer databases. As used in a CBCS, "software" refers to all flight software regardless of the media on which the software resides, including software that resides on hardware devices (i.e., firmware).

Software Controllable Inhibit - A system-level "hardware" inhibit whose state is controllable by software commands.

Software Error - The difference between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition.

Software Fault - An incorrect step, process or data definition in a computer system.

Software Inhibit - A software or firmware feature that prevents a specific software event from occurring or a specific software function from being available.

Time to Criticality - The time between the occurrence of a failure, event or condition and the subsequent occurrence of a hazard or other undesired outcome. Time to criticality will be established by engineering or operational analysis.

APPENDIX D USOS AND PRIME ITEM DEVELOPMENT SPECIFICATIONS IMPLEMENTATION

D.1

The implementation of sections 3 and 4 of this document have been implemented in the USOS Specification (SSP 41162) as follows:

3.3.6.3 Computer based control of hazardous functions

The following requirements apply where a Computer Based Control System (CBCS) is used to operate or control an on-orbit function with critical or catastrophic hazardous potential:

3.3.6.3.1 General

Where a CBCS is used to operate or control an on-orbit function with critical or catastrophic hazardous potential, the CBCS:

a) shall perform an orderly transition to a known, safe state upon receipt of a termination command or detection of a termination condition.
(SSP 50038B - 3.1.1.2)

b) shall be designed such that override commands require at least two independent actions by the operator. (SSP 50038B - 3.1.1.4)

c) shall reject hazardous commands which do not meet prerequisite checks for execution. (SSP 50038B - 3.1.1.13)

d) shall safely initialize to a known, safe state. (SSP 50038B - 3.1.1.1)

e) shall require at least one separate operator action to initiate each command message. (SSP 50038B - 3.1.1.4, 3.1.2.1.3, 3.1.3.1.7, 3.1.3.2.6)

3.3.6.3.2 Function deactivation (must work function)

Where the inadvertent deactivation of a function controlled by a CBCS could result in a critical or catastrophic hazard, the CBCS shall require two independent and unique command messages to deactivate a leg of redundancy in that function. (SSP 50038B - 3.1.2.1.1, 3.1.2.1.2)

3.3.6.3.3 Function activation (must not work function)

Where the inadvertent activation of a function controlled by a CBCS could result in an identified critical or catastrophic hazard, the CBCS:

a) shall make available to the operators the status of the monitored inhibit(s).
(SSP 50038B - 3.1.3.1.4, 3.1.3.2.1)

b) shall provide a unique command message to remove an inhibit.
(SSP 50038B - 3.1.3.1.3, 3.1.3.2.3)

c) shall make available to the operator the status of any software inhibits used to disable the execution of the command. (SSP 50038B - 3.1.3.1.8, 3.1.3.2.8)

d) shall check a separate, functionally independent parameter before issuance or execution of each hazardous command when the hazardous command is being initiated by a hard-coded automated sequence.
(SSP 50038B - 3.1.3.1.5, 3.1.3.2.4)

3.3.6.3.4 Function activation (must not work function) - multiple hazard controls

Where the inadvertent activation of a function controlled by a CBCS could result in an identified critical or catastrophic hazard and the CBCS provides two or more controls to the hazard, the CBCS shall provide a separate control path (SCP) with different functionality for each inhibit. (SSP 50038B - 3.1.3.2.2, 3.1.3.2.7)

VERIFICATION

4.3.3.6.3 Computer based control of hazardous functions

4.3.3.6.3.1 General

a) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify CBCS functions and their termination commands or conditions. Analysis of lower level verifications shall identify the state the function enters upon termination and show that the defined state is safe. This verification shall be considered successful when the analysis shows that the identified safe state is the state into which each CBCS enters upon receipt of the termination command or condition.

b) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify commands which allow an operator to override prerequisite checking. The analysis shall identify the number of operator actions necessary to initiate each command to perform an override. This verification shall be considered successful when the analysis shows that at least two separate operator actions are necessary to initiate the command(s) to perform an override.

c) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify the prerequisite checks that must be met for the execution of hazardous commands. An analysis shall identify the conditions which are checked prior to execution of these commands. This verification shall be considered successful when the analyses show that the hazardous commands will be rejected when any of the identified prerequisite checks for execution have not been satisfied.

d) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to characterize the initialization process for CBCS components. This analysis shall show that during initialization, the CBCS hardware produces no spurious output signals, and completes to a known safe state. This verification shall be considered successful when the analysis shows that the CBCS hardware resides in the identified safe state upon completion of initialization, and that no spurious signals were generated throughout the initialization process (i.e., from power application or commanded initialization through initialization completion).

e) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify operator initiated commands which could cause a critical or catastrophic hazard or could perform an override. The analysis shall identify the number of separate operator actions required to release each command. This verification shall be considered successful when the analysis shows that each identified command message can only be released by at least one separate operator action.

4.3.3.6.3.2 Function deactivation (must work function)

An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify functions whose loss could cause a critical or catastrophic hazard. The analysis shall identify the command messages necessary to deactivate legs of redundancy in the identified functions. This verification shall be considered successful when the analysis shows that, at a minimum, two unique and independent command messages are necessary to command the deactivation of each leg of redundancy in the identified functions.

4.3.3.6.3.3 Function activation (must not work function)

a) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify all monitored inhibits. The analysis shall identify the parameters available for monitoring the status of these inhibits. This verification shall be considered successful when the analysis shows that the status of each monitored inhibit is available to the operator.

b) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify safety inhibits and identify the command messages necessary to remove each inhibit. The analysis shall show that no single command message can remove more than one inhibit or toggle the state of an inhibit. This verification shall be considered successful when it is shown that a single command message does not remove more than one inhibit or place a single inhibit in more than one state.

c) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify software inhibits used to disable the execution of hazardous commands. The analysis shall show that software inhibit status is available to the operator. This verification shall be considered successful when the analysis shows that the status of each identified software inhibit is available to the operator.

d) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify each hard coded automated sequence which initiates hazardous commands, the hazardous commands which are initiated, and the parameters which are checked before the initiation or execution of each of these commands. The analysis shall show that at least one of the parameters checked for one hazardous command is functionally independent from the parameters checked for the other hazardous commands initiated by the same hard coded automated sequence. This verification shall be considered successful when the analysis shows that the parameters checked are functionally independent and that the parameters are checked either within the CBCS component or within the control path between the CBCS component and the inhibit before the initiation or execution of each hazardous command.

4.3.3.6.3.4 Function activation (must not work function) - multiple hazard controls

An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify the control path used to control each inhibit and the functionality of the control path. The analysis shall show that for a single hazard, each inhibit controlled by a single CBCS processor is controlled by a separate control path within that processor and that each control path provides different functionality such that common cause failures are prevented. This verification shall be considered successful when the analysis shows that each inhibit for a given hazard is controlled by a control path which provides functionality which is different from the control path(s) controlling the other inhibits to the hazard within the CBCS processor and that one control path cannot cause the execution of another control path or control an inhibit belonging to another control path for the hazard.

D.2

The implementation of sections 3 and 4 of this document have been implemented in the US Prime Item Development Specifications as follows:

3.3.6.3 Computer based control of hazardous functions

The following requirements apply where a Computer-Based Control System (CBCS) is used to operate or control an on-orbit function with critical or catastrophic hazardous potential:

3.3.6.3.1 General

Where a CBCS is used to operate or control an on-orbit function with critical or catastrophic hazardous potential, the CBCS components of the << End-Item >> :

a) shall perform an orderly transition to a known, safe state, upon receipt of a termination command or detection of a termination condition.
(SSP 50038B - 3.1.1.2)

b) shall require at least two separate command messages to perform an override.
(SSP 50038B - 3.1.1.4)

c) shall detect and recover from inadvertent memory modification during use.
(SSP 50038B - 3.1.1.6)

d) shall discriminate between valid and invalid inputs from sources external to each CBCS component and remain in or recover to a known safe state in the event of an invalid external input. (SSP 50038B - 3.1.1.8)

e) shall reject hazardous commands which do not meet prerequisite checks for execution. (SSP 50038B - 3.1.1.13)

f) shall provide for safe recovery from interrupts and exception conditions (e.g., debug exceptions, non-maskable interrupts, breakpoint, overflow, bounds check, invalid op code, co-processor not available, co-processor error, divide error, segment or gate not present, stack fault, general protection failure, page fault) within the CBCS component. (SSP 50038B - 3.1.1.7)

g) shall safely initialize to a known, safe state. (SSP 50038B - 3.1.1.1)

h) shall continue to operate safely during off-nominal power conditions, or contain design features which safe the processor during off-nominal power conditions. (SSP 50038B - 3.1.1.3)

i) shall require at least one separate operator action to initiate each command message. (SSP 50038B - 3.1.1.4, 3.1.2.1.3, 3.1.3.1.7, 3.1.3.2.6)

3.3.6.3.2 Function deactivation (must work function)

Where the inadvertent deactivation of a function controlled by a CBCS component in the << End-Item >> could result in a critical or catastrophic hazard, the CBCS component shall require two independent and unique command messages to deactivate a leg of redundancy in that function.
(SSP 50038B - 3.1.2.1.1, 3.1.2.1.2)

3.3.6.3.3 Function activation (must not work function)

Where the inadvertent activation of a function controlled by a CBCS component in the << End-Item >> could result in an identified critical or catastrophic hazard, the CBCS component:

a) shall make available to the operators the status of the monitored inhibit(s).
(SSP 50038B - 3.1.3.1.4, 3.1.3.2.1)

b) shall provide a unique command message to remove an inhibit.
(SSP 50038B - 3.1.3.1.3, 3.1.3.2.3)

c) shall make available to the operator the status of any software inhibits used to disable the execution of the command. (SSP 50038B - 3.1.3.1.8, 3.1.3.2.8)

d) shall check a separate, functionally independent parameter before issuance or execution of each hazardous command when the hazardous command is being initiated by a hard-coded automated sequence.
(SSP 50038B - 3.1.3.1.5, 3.1.3.2.4)

3.3.6.3.4 Function activation (must not work function) - multiple hazard controls

Where the inadvertent activation of a function controlled by a CBCS component in the << End-Item >> could result in an identified critical or catastrophic hazard and the CBCS component provides two or more controls to the hazard, the CBCS component shall provide a separate control path (SCP) with different functionality for each inhibit. (SSP 50038B - 3.1.3.2.2, 3.1.3.2.7)

VERIFICATION

4.3.3.6.3 Computer based control of hazardous functions

4.3.3.6.3.1 General

a) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify CBCS functions and their termination commands or conditions. Analysis of lower level verifications shall identify the state the function enters upon termination and show that the defined state is safe. This verification shall be considered successful when the analysis shows that the identified safe state is the state into which each CBCS enters upon receipt of the termination command or condition.

b) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify the functions providing for an operator override of prerequisite checking. The analysis shall identify the number of command messages necessary to perform an override. This verification shall be considered successful when the analysis shows that at least two separate command messages are necessary to perform an override.

c) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify the memory areas within a CBCS which are used to store code and adaptation data. The analysis shall identify the CBCS functions and processes which detect and recover from inadvertent memory modification of stored code and adaptation data. This verification shall be considered successful when the analysis shows that the CBCS can detect and recover from inadvertent memory modification of stored code and adaptation data.

d) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify the valid safety related inputs to each CBCS component. The analysis shall also identify the CBCS component response to invalid external inputs. This verification shall be considered successful when the analysis has shown that the CBCS component either remains in or recovers to a known safe state in the presence of invalid external inputs.

e) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify the prerequisite checks that must be met for the execution of hazardous commands. An analysis shall identify the conditions which are checked prior to execution of these commands. This verification shall be considered successful when the analyses show that the hazardous commands will be rejected when any of the identified prerequisite checks for execution have not been satisfied.

f) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify the CBCS component's unique interrupts and exceptions. This analysis shall identify how each is handled within each CBCS component. This verification shall be considered successful when the analysis shows that no spurious signals are generated by the CBCS component as a result of an interrupt or an exception.

g) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to characterize the initialization process for CBCS components. This analysis shall show that during initialization, the CBCS hardware produces no spurious output signals, and completes to a known safe state. This verification shall be considered successful when the analysis shows that the CBCS hardware resides in the identified safe state upon completion of initialization, and that no spurious signals were generated throughout the initialization process (i.e., from power application or commanded initialization through initialization completion).

h) An analysis of lower level verification shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify the CBCS reaction to off-nominal power conditions. This analysis shall show that the CBCS components do not output spurious signals during off-nominal power conditions. This verification shall be considered successful when the analysis shows that the CBCS continues to operate nominally or safes in the presence of off–nominal power input conditions.

i) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify operator initiated commands which could cause a critical or catastrophic hazard or could perform an override. The analysis shall identify the number of separate operator actions required to release each command. This verification shall be considered successful when the analysis shows that each identified command message can only be released by at least one separate operator action.

4.3.3.6.3.2 Function deactivation (must work function)

An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify functions whose loss could cause a critical or catastrophic hazard. The analysis shall identify the command messages necessary to deactivate legs of redundancy in the identified functions. This verification shall be considered successful when the analysis shows that, at a minimum, two unique and independent command messages are necessary to command the deactivation of each leg of redundancy in the identified functions.

4.3.3.6.3.3 Function activation (must not work function)

a) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify all monitored inhibits. The analysis shall identify the parameters available for monitoring the status of these inhibits. This verification shall be considered successful when the analysis shows that the status of each monitored inhibit is available to the operator.

b) An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify safety inhibits and identify the command messages necessary to remove each inhibit. The analysis shall show that no single command message can remove more than one inhibit or toggle the state of an inhibit. This verification shall be considered successful when it is shown that a single command message does not remove more than one inhibit or place a single inhibit in more than one state.

c) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify software inhibits used to disable the execution of hazardous commands. The analysis shall show that software inhibit status is available to the operator. This verification shall be considered successful when the analysis shows that the status of each identified software inhibit is available to the operator.

d) An analysis shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify each hard coded automated sequence which initiates hazardous commands, the hazardous commands which are initiated, and the parameters which are checked before the initiation or execution of each of these commands. The analysis shall show that at least one of the parameters checked for one hazardous command is functionally independent from the parameters checked for the other hazardous commands initiated by the same hard coded automated sequence. This verification shall be considered successful when the analysis shows that the parameters checked are functionally independent and that the parameters are checked either within the CBCS component or within the control path between the CBCS component and the inhibit before the initiation or execution of each hazardous command.

4.3.3.6.3.4 Function activation (must not work function) - multiple hazard controls

An analysis of lower level verifications shall be performed in accordance with SSP 30309 and through separate engineering analyses to identify the control path used to control each inhibit and the functionality of the control path. The analysis shall show that for a single hazard, each inhibit controlled by a single CBCS processor is controlled by a separate control path within that processor and that each control path provides different functionality such that common cause failures are prevented. This verification shall be considered successful when the analysis shows that each inhibit for a given hazard is controlled by a control path which provides functionality which is different from the control path(s) controlling the other inhibits to the hazard within the CBCS processor and that one control path cannot cause the execution of another control path or control an inhibit belonging to another control path for the hazard.

D.3

The following table provides the specific implementation detail for the US PIDSs:

PIDS Paragraph Number	Requirement	SSP 50038B Para #	Node 1 MDM App	SMC	GN&C App	PMCU App	ITS Z1	USL	HAB-A	Node 1	Node 2	Airlock	PV Module P6	Truss Element P5	PV Module P4	ITS P3	ITS P1	ITS S0	ITS S1	ITS S3	PV Module S4	Truss Element S5	PV Module S6	ULC	Resupply/Return Containment	PMA	PMA-1	Cupola	Mobile Transporter	EVA Aids	Cargo Handling Interface Assembly (ORU	Refrigerator/Freezer Rack Assy	Auxillary Power Converter Unit	Active Rack Isolation System
3.3.6.3.1	Where a CBCS is used to operate or control an on-orbit function with critical or catastrophic hazardous potential, the CBCS components of the << End-Item >> :		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.1.a	shall perform an orderly transition to a known, safe state, upon receipt of a termination command or detection of a termination condition	3.1.1.2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.1.b	shall require at least two unique command messages to perform an override	3.1.1.4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.1.c	shall detect and recover from inadvertent memory modification during use	3.1.1.6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.1.d	shall discriminate between valid and invalid inputs from sources external to each CBCS components and remain in or recover to a known safe state in the event of an invalid external input	3.1.1.8	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.1.e	shall reject hazardous commands which do not meet prerequisite checks for execution	3.1.1.13	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.1.f	shall provide for safe recovery from interrupts and exception conditions (e.g., debug exceptions, non-maskable interrupts, breakpoint, overflow, bounds check, invalid op code, co-processor not available, co-processor error, divide error,	3.1.1.7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.1.f	segment or gate not present, stack fault, general protection failure, page fault) within the CBCS component.		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.1.g	shall safely initialize to a known, safe state.	3.1.1.1	-	X	-	-	-	X	X	X	-	X	X	-	X	-	-	-	-	-	X	-	X	X	-	-	-	-	X	X	X	X	X	X
3.3.6.3.1.h	shall continue to operate safely during off-nominal power conditions, or contain design features which safe the processor during off-nominal power conditions.	3.1.1.3	-	X	-	-	-	X	X	X	-	X	X	-	X	-	-	-	-	-	X	-	X	X	-	-	-	-	X	X	X	X	X	X
3.3.6.3.1.i	shall require at least one separate operator action to initiate each command message.	3.1.1.4 3.1.2.1.3 3.1.3.1.7 3.1.3.2.6	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3.3.6.3.2	Where the deactivation of a function controlled by a CBCS in the << End-Item >> could result in a critical or catastrophic hazard, the CBCS component shall require two independent and unique cmd messages to deactivate a leg of redundancy in that function	3.1.2.1.1 3.1.2.1.2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.3	Where the activation of a function controlled by a CBCS component in the << End-Item >> could result in a critical or catastrophic hazard, the CBCS component :		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.3.a	shall make available to the operators the status of the monitored inhibit(s)	3.1.3.1.4 3.1.3.2.1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.3.b	shall provide a unique command message to remove an inhibit	3.1.3.1.3 3.1.3.2.3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.3.c	shall make available to the operator the status of any software inhibits used to disable the execution of the command	3.1.3.1.8 3.1.3.2.8	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.3.d	shall check a separate, functionally independent parameter before issuance or execution of each hazardous command when the hazardous command is being initiated by a hard-coded automated sequence	3.1.3.1.5 3.1.3.2.4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.4	Where the inadvertent activation of a function controlled by a CBCS component in the << End-Item >> could result in an identified critical or catastrophic hazard and the CBCS component provides two or more controls to the hazard, the CBCS component		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3.3.6.3.4	shall provide a separate control path (SCP) with different functionality for each inhibit.	3.1.3.2.2 3.1.3.2.7	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

D.4 Computer-Based Control System Requirements Trace

SSP 50038B	PIDS	Segment (USOS)	System	Comments
3.1			3.3.6.3.2	
3.1.1.1	3.3.6.3.1 g	3.3.6.3.1 d	3.3.6.3.1 c	
3.1.1.2	3.3.6.3.1 a	3.3.6.3.1 a	3.3.6.3.1 a	
3.1.1.3	3.3.6.3.1 h	3.2.1.1.1.15	3.2.1.1.1.4	
3.1.1.3	3.3.6.3.1 h	3.2.1.1.1.16	3.2.1.1.1.4	
3.1.1.4	3.3.6.3.1 b	3.3.6.3.1 b	3.3.6.1.1	
3.1.1.4			3.3.6.1.2	
3.1.1.5		3.3.6.3.1 c	3.3.6.3.1 b	SRS level decomp of PIDS 3.3.6.3.1 e
3.1.1.6	3.3.6.3.1 c	3.2.1.1.1.15	3.2.1.1.1.4	
3.1.1.6	3.3.6.3.1 c	3.2.1.1.1.16	3.2.1.1.1.4	
3.1.1.7	3.3.6.3.1 f	3.2.1.1.1.15	3.2.1.1.1.4	
3.1.1.7	3.3.6.3.1 f	3.2.1.1.1.16	3.2.1.1.1.4	
3.1.1.8	3.3.6.3.1 d	3.2.1.1.1.15	3.2.1.1.1.4	
3.1.1.8	3.3.6.3.1 d	3.2.1.1.1.16	3.2.1.1.1.4	
3.1.1.9				S/W standard (SSPS)
3.1.1.10				S/W standard (SSPS)
3.1.1.11		3.2.1.1.1.15	3.2.1.1.1.4	SRS level decomp of PIDS 3.3.6.3.1 d
3.1.1.11		3.2.1.1.1.16	3.2.1.1.1.4	SRS level decomp of PIDS 3.3.6.3.1 d
3.1.1.12		3.3.9	3.3.9	duplicate of SYS 3.3.9
3.1.1.13	3.3.6.3.1 e	3.3.6.3.1 c	3.3.6.3.1 b	
3.1.2.1		3.3.6.1.1	3.3.6.1.1	duplicate of SYS 3.3.6.1.1 and 3.3.6.1.2
3.1.2.1		3.3.6.1.2	3.3.6.1.2	duplicate of SYS 3.3.6.1.1 and 3.3.6.1.2
3.1.2.1.1	3.3.6.3.2	3.3.6.3.2	3.3.6.3.2	
3.1.2.1.2	3.3.6.3.2	3.3.6.3.2	3.3.6.3.2	
3.1.2.1.3	3.3.6.3.1 i	3.3.6.3.1 e	3.3.6.3.2	
3.1.2.1.4	Not Allocated			new requirement
3.1.2.1.5		3.2.3.3	3.2.3.4	duplicate of SYS 3.2.3.4
3.1.2.1.6		3.2.1.1.1.15	3.2.1.1.1.4	duplicate of SYS 3.2.1.1.1.4
3.1.2.1.6		3.2.1.1.1.16	3.2.1.1.1.4	duplicate of SYS 3.2.1.1.1.4
3.1.3.1.1		3.3.6.3.1 c	3.3.6.3.1 b	SRS level decomp of PIDS 3.3.6.3.1 e
3.1.3.1.2		3.3.6.2	3.3.6.2	decomp of SYS 3.3.6.2
3.1.3.1.3	3.3.6.3.3 b	3.3.6.3.3 b	3.3.6.3.2	
3.1.3.1.4	3.3.6.3.3 a	3.3.6.3.3 a	3.3.6.3.2	
3.1.3.1.5	3.3.6.3.3 d	3.3.6.3.3 d	3.3.6.3.2	
3.1.3.1.6		3.2.4.3.4	3.2.4.3.4	intent satisfied by SYS 3.2.4.3.4
3.1.3.1.7	3.3.6.3.1 i	3.3.6.3.1 e	3.3.6.3.2	
3.1.3.1.8	3.3.6.3.3 c	3.3.6.3.3 c	3.3.6.3.2	
3.1.3.1.9		3.2.4.3.1	3.2.4.3.1	intent satisfied by SYS 3.2.4.3.1
3.1.3.1.10		3.3.6.3.4	3.3.6.3.2	intent satisfied by PIDS 3.3.6.3.4
3.1.3.2.1	3.3.6.3.3 a	3.3.6.3.3 a	3.3.6.3.2	
3.1.3.2.2	3.3.6.3.4	3.3.6.3.4	3.3.6.3.2	
3.1.3.2.3	3.3.6.3.3 b	3.3.6.3.3 b	3.3.6.3.2	
3.1.3.2.4	3.3.6.3.3 d	3.3.6.3.3 d	3.3.6.3.2	
3.1.3.2.5		3.2.4.3.4	3.2.4.3.4	intent satisfied by SYS 3.2.4.3.4
3.1.3.2.6	3.3.6.3.1 i	3.3.6.3.1 e	3.3.6.3.2	
3.1.3.2.7	3.3.6.3.4	3.3.6.3.4	3.3.6.3.2	
3.1.3.2.8	3.3.6.3.3 c	3.3.6.3.3 c	3.3.6.3.2	
3.1.3.2.9		3.2.1.1.1.22	3.2.1.1.1.7	duplicate of SYS 3.2.1.1.1.7
3.1.3.2.9		3.2.1.1.1.23	3.2.1.1.1.7	duplicate of SYS 3.2.1.1.1.7
3.1.3.2.9		3.2.1.1.1.24	3.2.1.1.1.7	duplicate of SYS 3.2.1.1.1.7

D.5 Computer-Based Control System Requirements Trace

System	Segment (USOS)	PIDS	SSP 50038 B	Comments
3.2.1.1.1.4	3.2.1.1.1.15		3.1.1.11	SRS level decomp of PIDS 3.3.6.3.1. d
3.2.1.1.1.4	3.2.1.1.1.15		3.1.2.1.6	duplicate of SYS 3.2.1.1.1.4
3.2.1.1.1.4	3.2.1.1.1.15	3.3.6.3.1 c	3.1.1.6	
3.2.1.1.1.4	3.2.1.1.1.15	3.3.6.3.1 d	3.1.1.8	
3.2.1.1.1.4	3.2.1.1.1.15	3.3.6.3.1 f	3.1.1.7	
3.2.1.1.1.4	3.2.1.1.1.15	3.3.6.3.1 h	3.1.1.3	
3.2.1.1.1.4	3.2.1.1.1.16		3.1.1.11	SRS level decomp of PIDS 3.3.6.3.1. d
3.2.1.1.1.4	3.2.1.1.1.16		3.1.2.1.6	duplicate of SYS 3.2.1.1.1.4
3.2.1.1.1.4	3.2.1.1.1.16	3.3.6.3.1 c	3.1.1.6	
3.2.1.1.1.4	3.2.1.1.1.16	3.3.6.3.1 d	3.1.1.8	
3.2.1.1.1.4	3.2.1.1.1.16	3.3.6.3.1 f	3.1.1.7	
3.2.1.1.1.4	3.2.1.1.1.16	3.3.6.3.1 h	3.1.1.3	
3.2.1.1.1.7	3.2.1.1.1.22		3.1.3.2.9	duplicate of SYS 3.2.1.1.1.7
3.2.1.1.1.7	3.2.1.1.1.23		3.1.3.2.9	duplicate of SYS 3.2.1.1.1.7
3.2.1.1.1.7	3.2.1.1.1.24		3.1.3.2.9	duplicate of SYS 3.2.1.1.1.7
3.2.3.4	3.2.3.3		3.1.2.1.5	duplicate of SYS 3.2.3.4
3.2.4.3.1	3.2.4.3.1		3.1.3.1.9	intent satisfied by SYS 3.2.4.3.1
3.2.4.3.4	3.2.4.3.4		3.1.3.1.6	intent satisfied by SYS 3.2.4.3.4
3.2.4.3.4	3.2.4.3.4		3.1.3.2.5	intent satisfied by SYS 3.2.4.3.4
3.3.6.1.1	3.3.6.1.1		3.1.2.1	duplicate of SYS 3.3.6.1.1 and 3.3.6.1.2
3.3.6.1.1	3.3.6.3.1 b	3.3.6.3.1 b	3.1.1.4	
3.3.6.1.1	3.3.6.3.1 e	3.3.6.3.1 i	3.1.1.4	
3.3.6.1.2	3.3.6.1.2		3.1.2.1	duplicate of SYS 3.3.6.1.1 and 3.3.6.1.2
3.3.6.2	3.3.6.2		3.1.3.1.2	decomp of SYS 3.3.6.2
3.3.6.3.1 a	3.3.6.3.1 a	3.3.6.3.1 a	3.1.1.2	
3.3.6.3.1 b	3.3.6.3.1 c		3.1.1.5	SRS level decomp of PIDS 3.3.6.3.1. e
3.3.6.3.1 b	3.3.6.3.1 c		3.1.3.1.1	SRS level decomp of PIDS 3.3.6.3.1. e
3.3.6.3.1 b	3.3.6.3.1 c	3.3.6.3.1 e	3.1.1.13	
3.3.6.3.1 c	3.3.6.3.1 d	3.3.6.3.1 g	3.1.1.1	
3.3.6.3.2			3.1	
3.3.6.3.2	3.3.6.3.1 e	3.3.6.3.1 i	3.1.2.1.3	
3.3.6.3.2	3.3.6.3.1 e	3.3.6.3.1 i	3.1.3.1.7	
3.3.6.3.2	3.3.6.3.1 e	3.3.6.3.1 i	3.1.3.2.6	
3.3.6.3.2	3.3.6.3.2	3.3.6.3.2	3.1.2.1.1	
3.3.6.3.2	3.3.6.3.2	3.3.6.3.2	3.1.2.1.2	
3.3.6.3.2	3.3.6.3.3 a	3.3.6.3.3 a	3.1.3.1.4	
3.3.6.3.2	3.3.6.3.3 a	3.3.6.3.3 a	3.1.3.2.1	
3.3.6.3.2	3.3.6.3.3 b	3.3.6.3.3 b	3.1.3.1.3	
3.3.6.3.2	3.3.6.3.3 b	3.3.6.3.3 b	3.1.3.2.3	
3.3.6.3.2	3.3.6.3.3 c	3.3.6.3.3 c	3.1.3.1.8	
3.3.6.3.2	3.3.6.3.3 c	3.3.6.3.3 c	3.1.3.2.8	
3.3.6.3.2	3.3.6.3.3 d	3.3.6.3.3 d	3.1.3.1.5	
3.3.6.3.2	3.3.6.3.3 d	3.3.6.3.3 d	3.1.3.2.4	
3.3.6.3.2	3.3.6.3.4		3.1.3.1.10	intent satisfied by PIDS 3.3.6.3.4
3.3.6.3.2	3.3.6.3.4	3.3.6.3.4	3.1.3.2.2	
3.3.6.3.2	3.3.6.3.4	3.3.6.3.4	3.1.3.2.7	
3.3.9	3.3.9		3.1.1.12	duplicate of SYS 3.3.9
			3.1.1.9	S/W standard (SSPS)
			3.1.1.10	S/W standard (SSPS)
		Not Allocated	3.1.2.1.4	new requirement