

CS Requirements to Design Traceability

CS Version 2.5.0

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Requirement Text** | **Software Design Element** | **Applicable Software Functions** |
| CS1000 | Upon receipt of a No-Op command, CS shall increment the CS Valid Command Counter and generate an event message. | Operation | CS\_NoopCmd |
| CS1001 | Upon receipt of a Reset command, CS shall reset the following housekeeping variables to a value of zero:   a) Valid Ground Command Counter   b) Ground Command Rejected Counter   c) Non-volatile CRC Miscompare Counter   d) OS Code Segment CRC Miscompare Counter   e) cFE Code Segment CRC Miscompare Counter   f) Application CRC Miscompare Counter   g) Table CRC Miscompare Counter   h) User-defined Memory CRC Miscompare Counter   i) Checksum Pass Counter (number of passes through all of the checksum areas) | Operation | CS\_ResetCmd |
| CS1002 | For all CS commands, if the length contained in the message header is not equal to the expected length, CS shall reject the command and issue an event message. | Operation | CS\_VerifyCmdLength  CS\_NoopCmd  CS\_ResetCmd  CS\_OneShotCmd  CS\_CancelOneShotCmd  CS\_EnableAllCSCmd  CS\_DisableAllCSCmd  CS\_EnableCfeCoreCmd  CS\_DisableCfeCoreCmd  CS\_ReportBaselineCfeCoreCmd  CS\_RecomputeBaselineCfeCoreCmd  CS\_EnableOSCmd  CS\_DisableOSCmd  CS\_ReportBaselineOSCmd  CS\_RecomputeBaselineOSCmd  CS\_EnableEepromCmd  CS\_DisableEepromCmd  CS\_ReportBaselineEntryIDEepromCmd  CS\_RecomputeBaselineEepromCmd  CS\_EnableEntryIDEepromCmd  CS\_DisableEntryIDEepromCmd  CS\_GetEntryIDEepromCmd  CS\_EnableMemoryCmd  CS\_DisableMemoryCmd  CS\_ReportBaselineEntryIDMemoryCmd  CS\_RecomputeBaselineMemoryCmd  CS\_EnableEntryIDMemoryCmd  CS\_DisableEntryIDMemoryCmd  CS\_GetEntryIDMemoryCmd  CS\_EnableTablesCmd  CS\_DisableTablesCmd  CS\_ReportBaselineTablesCmd  CS\_RecomputeBaselineTablesCmd  CS\_EnableNameTablesCmd  CS\_DisableNameTablesCmd  CS\_EnableAppCmd  CS\_DisableAppCmd  CS\_ReportBaselineAppCmd  CS\_RecomputeBaselineAppCmd  CS\_EnableNameAppCmd  CS\_DisableNameAppCmd |
| CS1003 | If CS accepts any command as valid, CS shall execute the command, increment the CS Valid Command Counter and issue an event message | Operation | CS\_NoopCmd  CS\_ResetCmd  CS\_OneShotCmd  CS\_CancelOneShotCmd  CS\_EnableAllCSCmd  CS\_DisableAllCSCmd  CS\_EnableCfeCoreCmd  CS\_DisableCfeCoreCmd  CS\_ReportBaselineCfeCoreCmd  CS\_RecomputeBaselineCfeCoreCmd  CS\_EnableOSCmd  CS\_DisableOSCmd  CS\_ReportBaselineOSCmd  CS\_RecomputeBaselineOSCmd  CS\_EnableEepromCmd  CS\_DisableEepromCmd  CS\_ReportBaselineEntryIDEepromCmd  CS\_RecomputeBaselineEepromCmd  CS\_EnableEntryIDEepromCmd  CS\_DisableEntryIDEepromCmd  CS\_GetEntryIDEepromCmd  CS\_EnableMemoryCmd  CS\_DisableMemoryCmd  CS\_ReportBaselineEntryIDMemoryCmd  CS\_RecomputeBaselineMemoryCmd  CS\_EnableEntryIDMemoryCmd  CS\_DisableEntryIDMemoryCmd  CS\_GetEntryIDMemoryCmd  CS\_EnableTablesCmd  CS\_DisableTablesCmd  CS\_ReportBaselineTablesCmd  CS\_RecomputeBaselineTablesCmd  CS\_EnableNameTablesCmd  CS\_DisableNameTablesCmd  CS\_EnableAppCmd  CS\_DisableAppCmd  CS\_ReportBaselineAppCmd  CS\_RecomputeBaselineAppCmd  CS\_EnableNameAppCmd  CS\_DisableNameAppCmd |
| CS1004 | If CS rejects any command, CS shall abort the command execution, increment the CS Command Rejected Counter and issue an error event message | Operation | CS\_ProcessCmd  CS\_VerifyCmdLength  CS\_NoopCmd  CS\_ResetCmd  CS\_OneShotCmd  CS\_CancelOneShotCmd  CS\_EnableAllCSCmd  CS\_DisableAllCSCmd  CS\_EnableCfeCoreCmd  CS\_DisableCfeCoreCmd  CS\_ReportBaselineCfeCoreCmd  CS\_RecomputeBaselineCfeCoreCmd  CS\_EnableOSCmd  CS\_DisableOSCmd  CS\_ReportBaselineOSCmd  CS\_RecomputeBaselineOSCmd  CS\_EnableEepromCmd  CS\_DisableEepromCmd  CS\_ReportBaselineEntryIDEepromCmd  CS\_RecomputeBaselineEepromCmd  CS\_EnableEntryIDEepromCmd  CS\_DisableEntryIDEepromCmd  CS\_GetEntryIDEepromCmd  CS\_EnableMemoryCmd  CS\_DisableMemoryCmd  CS\_ReportBaselineEntryIDMemoryCmd  CS\_RecomputeBaselineMemoryCmd  CS\_EnableEntryIDMemoryCmd  CS\_DisableEntryIDMemoryCmd  CS\_GetEntryIDMemoryCmd  CS\_EnableTablesCmd  CS\_DisableTablesCmd  CS\_ReportBaselineTablesCmd  CS\_RecomputeBaselineTablesCmd  CS\_EnableNameTablesCmd  CS\_DisableNameTablesCmd  CS\_EnableAppCmd  CS\_DisableAppCmd  CS\_ReportBaselineAppCmd  CS\_RecomputeBaselineAppCmd  CS\_EnableNameAppCmd  CS\_DisableNameAppCmd |
| CS1005 | CS shall use the <MISSION\_DEFINED> CRC algorithm to compute the CRCs for any segment | Operation | CS\_ComputeEepromMemory  CS\_ComputeTables  CS\_ComputeApp  CS\_OneShotChildTask |
| CS2001 | The Checksum App shall calculate CRCs for each Table-Defined Non-volatile segment and compare them against the corresponding baseline Non-volatile segment CRCs if:  a) Checksumming (as a whole) is Enabled  b) Non-volatile segment checksumming is Enabled  c) Checksumming for the individual Non-volatile segment is Enabled | Operation | CS\_ComputeEepromMemory  CS\_BackgroundEeprom |
| CS2001.1 | If the Non-volatile segment CRC is not equal to the corresponding baseline CRC, CS shall increment the Non-volatile CRC Miscompare Counter and send an event message | Operation | CS\_BackgroundEeprom |
| CS2001.2 | If the table-defined segment is invalid, CS shall send an error event message | Operation | CS\_BackgroundEeprom |
| CS2002 | Upon receipt of a Enable Non-volatile Checksumming command, CS shall enable non-volatile checksumming | Operation | CS\_EnableEepromCmd |
| CS2003 | Upon receipt of a Disable Non-volatile Checksumming command, CS shall disable non-volatile checksumming | Operation | CS\_DisableEepromCmd |
| CS2004 | Upon receipt of a Enable Non-volatile Segment command, CS shall enable checksumming of the command-specified non-volatile segment | Operation | CS\_EnableEntryIDEepromCmd |
| CS2005 | Upon receipt of a Disable Non-volatile Segment command, CS shall disable checksumming of the command-specified non-volatile segment | Operation | CS\_DisableEntryIDEepromCmd |
| CS2006 | Upon receipt of a Recompute Non-volatile Checksum Segment command, CS shall:  a) Recompute the baseline checksum for the command-specified non-volatile segment  b) Set the Recompute In Progress Flag to TRUE | Operation | CS\_RecomputeBaselineEepromCmd  CS\_RecomputeEepromMemoryChildTask |
| CS2006.1 | If CS is already processing a Recompute CRC command or a One Shot CRC command, CS shall reject the command. | Operation | CS\_RecomputeBaselineEepromCmd |
| CS2006.2 | Once the baseline CRC is computed, CS shall:  a) Generate an informational event message containing the baseline CRC  b) Set the Recompute In Progress Flag to FALSE | Operation | CS\_RecomputeBaselineEepromCmd  CS\_RecomputeEepromMemoryChildTask  CFE\_EVS\_SendEvent |
| CS2007 | Upon receipt of a Report Non-volatile Checksum Segment command, CS shall send an event message containing the baseline checksum for the command-specified non-volatile segment | Operation | CS\_ReportBaselineEntryIDEepromCmd  CFE\_EVS\_SendEvent |
| CS2008 | Upon receipt of a Get Non-volatile Checksum Segment command, CS shall send an event message containing the segment number for the command-specified non-volatile address | Operation | CS\_GetEntryIDEepromCmd  CFE\_EVS\_SendEvent |
| CS2009 | If a command-specified segment is invalid (for any of the non-volatile memory commands where segment is a command argument), CS shall reject the command and send an event message | Operation | CS\_GetEntryIDEepromCmd  CS\_DisableEntryIDEepromCmd  CS\_EnableEntryIDEepromCmd  CS\_ReportBaselineEntryIDEepromCmd |
| CS2010 | CS shall provide the ability to dump the baseline CRCs and status for the non-volatile memory segments via a dump-only table | Operation |  |
| CS3000 | Checksum shall calculate CRC for the OS code segment and compare them against the corresponding baseline OS code segment CRC if:  a) Checksumming (as a whole) is Enabled  b) OS segment checksumming is Enabled | Operation | CS\_BackgroundOS  CS\_ComputeEepromMemory |
| CS3000.1 | If the OS code segment CRC is not equal to the baseline OS code segment CRC, CS shall increment the OS Code Segment CRC Miscompare Counter and send an event message | Operation | CS\_BackgroundOS  CFE\_EVS\_SendEvent |
| CS3002 | Upon receipt of a Enable OS code segment command, CS shall enable checksumming of the OS Code segment | Operation | CS\_EnableOSCmd |
| CS3003 | Upon receipt of a Disable OS code segment command, CS shall Disable checksumming of the OS Code segment | Operation | CS\_DisableOSCmd |
| CS3004 | Upon receipt of a Recompute OS code segment CRC command, CS shall:   a) Recompute the baseline CRC for the OS code segment   b) Set the Recompute In Progress Flag to TRUE | Operation | CS\_RecomputeBaselineOSCmd  CS\_RecomputeEepromMemoryChildTask |
| CS3004.1 | Once the baseline CRC is computed, CS shall:   a) Generate an event message containing the baseline CRC   b) Set the Recompute In Progress Flag to FALSE | Operation | CS\_RecomputeBaselineOSCmd  CS\_RecomputeEepromMemoryChildTask  CFE\_EVS\_SendEvent |
| CS3004.2 | If CS is already processing a Recompute CRC command or a One Shot CRC command, CS shall reject the command. | Operation | CS\_RecomputeBaselineOSCmd |
| CS3005 | Upon receipt of a Report OS code segment CRC command, CS shall send an event message containing the baseline OS code segment CRC. | Operation | CS\_ReportBaselineOSCmd  CFE\_EVS\_SendEvent |
| CS3006 | Checksum shall calculate CRC for the cFE code segment and compare them against the corresponding baseline cFE code segment CRC if:   a) Checksumming (as a whole) is Enabled   b) cFE segment checksumming is Enabled | Operation | CS\_BackgroundCfeCore  CS\_ComputeEepromMemory |
| CS3006.1 | If the cFE code segment CRC is not equal to the baseline cFE code segment CRC, CS shall increment the cFE Code Segment CRC Miscompare Counter and send an event message | Operation | CS\_BackgroundCfeCore  CFE\_EVS\_SendEvent |
| CS3007 | Upon receipt of a Enable cFE code segment command, CS shall enable checksumming of the cFE Code segment | Operation | CS\_EnableCfeCoreCmd |
| CS3008 | Upon receipt of a Disable cFE code segment command, CS shall Disable checksumming of the cFE Code segment | Operation | CS\_DisableCfeCoreCmd |
| CS3009 | Upon receipt of a Recompute cFE Code Segment CRC command, CS shall:  a) Recompute the baseline CRC for the cFE Code Segment  b) Set the Recompute In Progress Flag to TRUE | Operation | CS\_RecomputeBaselineCfeCoreCmd  CS\_RecomputeEepromMemoryChildTask |
| CS3009.1 | Once the baseline CRC is computed, CS shall:  a) Generate an event message containing the baseline CRC  b) Set the Recompute In Progress Flag to FALSE | Operation | CS\_RecomputeBaselineCfeCoreCmd  CS\_RecomputeEepromMemoryChildTask  CFE\_EVS\_SendEvent |
| CS3009.2 | If CS is already processing a Recompute CRC command or a One Shot CRC command, CS shall reject the command. | Operation | CS\_RecomputeBaselineCfeCoreCmd |
| CS3010 | Upon receipt of a Report cFE code segment CRC command, CS shall send an event message containing the baseline cFE code segment CRC. | Operation | CS\_ReportBaselineCfeCoreCmd  CFE\_EVS\_SendEvent |
| CS4000 | Checksum shall calculate CRCs for each Table-Defined Application's code segment and compare them against the corresponding Application's baseline code segment CRC if:   a) Checksumming (as a whole) is Enabled   b) App code segment checksumming is Enabled   c) Checksumming of the individual Application Code Segment is Enabled | Operation | CS\_BackgroundApp  CS\_ComputeApp |
| CS4000.1 | If the Application's code segment CRC is not equal to the corresponding Application's baseline code segment CRC, CS shall increment the Application Code Segment CRC Miscompare Counter and send an event message. | Operation | CS\_BackgroundApp  CFE\_EVS\_SendEvent |
| CS4000.2 | If the table-defined Application code segment is invalid, CS shall send an event message and skip that Application code segment. | Operation | CS\_BackgroundApp  CS\_ComputeApp  CFE\_EVS\_SendEvent |
| CS4001 | Upon receipt of a Enable Application checksumming command, CS shall enable checksumming of all Application Code segments. | Operation | CS\_EnableAppCmd |
| CS4002 | Upon receipt of a Disable Application checksumming command, CS shall Disable checksumming of all Application Code segments. | Operation | CS\_DisableAppCmd |
| CS4003 | Upon receipt of a Enable Application code segment command, CS shall enable checksumming of the command-specified Application. | Operation | CS\_EnableNameAppCmd |
| CS4004 | Upon receipt of a Disable Application code segment command, CS shall Disable checksumming of the command-specified Application. | Operation | CS\_DisableNameAppCmd |
| CS4005 | Upon receipt of a Recompute Application Code Segment CRC command, CS shall:   a) Recompute the baseline CRC for the Application   b) Set the Recompute In Progress Flag to TRUE | Operation | CS\_RecomputeBaselineAppCmd  CS\_RecomputeAppChildTask |
| CS4005.1 | Once the baseline CRC is computed, CS shall:   a) Generate an event message containing the baseline CRC   b) Set the Recompute In Progress Flag to FALSE | Operation | CS\_RecomputeBaselineAppCmd  CFE\_EVS\_SendEvent |
| CS4005.2 | If CS is already processing a Recompute CRC command or a One Shot CRC command, CS shall reject the command. | Operation | CS\_RecomputeBaselineAppCmd |
| CS4006 | Upon receipt of a Report Application code segment CRC command, CS shall send an event message containing the baseline Application code segment CRC | Operation | CS\_ReportBaselineAppCmd  CFE\_EVS\_SendEvent |
| CS4007 | If the command-specified Application is invalid (for any Application Code Segment command where the Application is a command argument, CS shall reject the command and send an event message | Operation | CS\_EnableNameAppCmd  CS\_DisableNameAppCmd  CS\_GetAppResTblEntryByName |
| CS4008 | CS shall provide the ability to dump the baseline CRCs and status for the Application code segment memory segments via a dump-only table | Operation |  |
| CS5000 | Checksum shall calculate CRCs for each Table-Defined Table and compare them against the corresponding Table's baseline CRC if:   a) Checksumming (as a whole) is Enabled   b) Table checksumming is Enabled   c) Checksumming of the Individual Table is Enabled | Operation | CS\_BackgroundTables  CS\_ComputeTables |
| CS5000.1 | If the Table's CRC is not equal to the corresponding Table's baseline CRC and the table has not been modified (thru a table load), CS shall increment the Table CRC Miscompare Counter and send an event message. | Operation | CS\_BackgroundTables  CFE\_EVS\_SendEvent |
| CS5000.2 | If the Table's CRC is not equal to the corresponding Table's baseline CRC and the table has been modified (thru a table load), CS shall recompute the table baseline CRC. | Operation | CS\_ComputeTables |
| CS5000.3 | If the table-defined Table is invalid, CS shall send an event message and skip that Table. | Operation | CS\_ComputeTables |
| CS5001 | Upon receipt of a Enable Table Checksumming command, CS shall enable checksumming of all Tables. | Operation | CS\_EnableTablesCmd |
| CS5002 | Upon receipt of a Disable Table Checksumming command, CS shall Disable checksumming of all Tables. | Operation | CS\_DisableTablesCmd |
| CS5003 | Upon receipt of a Enable Table Name command, CS shall enable checksumming of the command-specified Table. | Operation | CS\_EnableNameTablesCmd |
| CS5004 | Upon receipt of a Disable Table Name command, CS shall Disable checksumming of the command-specified Table. | Operation | CS\_DisableNameTablesCmd |
| CS5005 | Upon receipt of a Recompute Table CRC Command, CS shall:   a) Recompute the baseline CRC for the command-specified table   b) Set the Recompute In Progress Flag to TRUE | Operation | CS\_RecomputeBaselineTablesCmd  CS\_RecomputeTablesChildTask |
| CS5005.1 | Once the baseline CRC is computed, CS shall:   a) Generate an event message containing the baseline CRC   b) Set the Recompute In Progress Flag to FALSE | Operation | CS\_RecomputeTablesChildTask  CFE\_EVS\_SendEvent |
| CS5005.2 | If CS is already processing a Recompute CRC command or a One Shot CRC command, CS shall reject the command. | Operation | CS\_RecomputeBaselineTablesCmd |
| CS5006 | Upon receipt of a Report Table CRC command, CS shall send an event message containing the baseline Table CRC for the command-specified table. | Operation | CS\_ReportBaselineTablesCmd  CFE\_EVS\_SendEvent |
| CS5007 | If the command-specified Table in invalid (for any CS Table command where a table name is a command argument ), CS shall reject the command and send an event message | Operation | CS\_EnableNameTablesCmd  CS\_DisableNameTablesCmd |
| CS5008 | CS shall provide the ability to dump the baseline CRCs and status for the tables via a dump-only table. | Operation | CS\_GetAppResTblEntryByName  CS\_ReportBaselineTablesCmd  CS\_RecomputeBaselineTablesCmd  CS\_DisableNameTablesCmd  CS\_EnableNameTablesCmd |
| CS6000 | Checksum shall calculate CRCs for each Table-Defined User-Defined Memory and compare them against the corresponding baseline CRC if  a) Checksumming (as a whole) is Enabled  b) User-Defined Memory checksumming is Enabled  c) Checksumming of the Individual Memory segments is Enabled | Operation | CS\_BackgroundMemory  CS\_ComputeEepromMemory |
| CS6000.1 | If the User-Defined Memory's CRC is not equal to the corresponding baseline CRC, CS shall increment the User-Defined Memory CRC Miscompare Counter and send an event message. | Operation | CS\_BackgroundMemory  CFE\_EVS\_SendEvent |
| CS6000.2 | If the table-defined Memory is invalid, CS shall send an event message. | Operation | CS\_ComputeEepromMemory  CFE\_EVS\_SendEvent |
| CS6001 | Upon receipt of a Enable User-Defined Memory Checksumming command, CS shall enable checksumming of all User-Defined Memory. | Operation | CS\_EnableMemoryCmd |
| CS6002 | Upon receipt of a Disable User-Defined Memory Checksumming command, CS shall disable checksumming of all User-Defined Memory. | Operation | CS\_DisableMemoryCmd |
| CS6003 | Upon receipt of a Enable User-Defined Memory Item command, CS shall enable checksumming of the command-specified Memory. | Operation | CS\_EnableEntryIDMemoryCmd |
| CS6004 | Upon receipt of a Disable User-Defined Memory Item command, CS shall Disable checksumming of the command-specified Memory. | Operation | CS\_DisableEntryIDMemoryCmd |
| CS6005 | Upon receipt of a Recompute User-Defined Memory CRC command, CS shall:   a) Recompute the baseline CRC for the command-specified User-Defined Memory.   b) Set the Recompute In Progress Flag to TRUE | Operation | CS\_RecomputeBaselineMemoryCmd  CS\_RecomputeEepromMemoryChildTask |
| CS6005.1 | Once the baseline CRC is computed, CS shall:   a) Generate an event message containing the baseline CRC   b) Set the Recompute In Progress Flag to FALSE | Operation | CS\_RecomputeEepromMemoryChildTask  CFE\_EVS\_SendEvent |
| CS6005.2 | If CS is already processing a Recompute CRC command or a One Shot CRC command, CS shall reject the command. | Operation | CS\_RecomputeBaselineMemoryCmd |
| CS6006 | Upon receipt of a Report User-Defined Memory CRC command, CS shall send an event message containing the baseline CRC for the command-specified User-Defined Memory. | Operation | CS\_ReportBaselineEntryIDMemoryCmd  CFE\_EVS\_SendEvent |
| CS6007 | If the command-specified User-Defined Memory is invalid (for any of the User-Defined memory commands where the memory ID is a command argument), CS shall reject the command and send an event message | Operation | CS\_EnableEntryIDMemoryCmd  CS\_DisableEntryIDMemoryCmd  CS\_GetEntryIDMemoryCmd  CS\_ReportBaselineEntryIDMemoryCmd |
| CS6008 | CS shall provide the ability to dump the baseline CRCs and status for all the User-Defined Memory via a dump-only table. | Operation | CS\_EnableEntryIDMemoryCmd  CS\_DisableEntryIDMemoryCmd  CS\_GetEntryIDMemoryCmd |
| CS6009 | Upon receipt of a Get User-Defined Memory Entry ID command, CS shall send an informational event message containing the User-Defined Memory Table Entry ID for the command-specified Memory Address. | Operation | CS\_GetEntryIDMemoryCmd  CFE\_EVS\_SendEvent |
| CS6009.1 | If the command-specified Memory Address cannot be found within the User-Defined Memory Table, CS shall send an informational event message. | Operation | CS\_GetEntryIDMemoryCmd  CFE\_EVS\_SendEvent |
| CS7000 | The CS software shall limit the amount of bytes processed during each of its execution cycles to a maximum of <PLATFORM\_DEFINED> bytes | Operation | CS\_ComputeTables  CS\_ComputeApp  CS\_ComputeEepromMemory  OS\_TaskDelay |
| CS8000 | Upon receipt of a Enable Checksum command, CS shall start calculating CRCs and compare them against the baseline CRCs. | Operation | CS\_EnableAllCSCmd |
| CS8001 | Upon receipt of a Disable Checksum command, CS shall stop calculating CRCs and comparing them against the baseline CRCs. | Operation | CS\_DisableAllCSCmd |
| CS8002 | Upon receipt of a One Shot command, CS shall:   a) Calculate the CRC starting at the command-specified address for the command-specified bytes at the command-specified rate (Max Bytes Per Cycle).   b) Set the One Shot In Progress Flag to TRUE | Operation | CS\_OneShotCmd  CS\_OneShotChildTask |
| CS8002.1 | Once the CRC is computed CS shall:  a) Issue an event message containing the CRC  b) Set the One Shot In Progress Flag to FALSE | Operation | CS\_OneShotChildTask  CFE\_EVS\_SendEvent |
| CS8002.2 | If CS is already processing a One Shot CRC command or a Recompute CRC command, CS shall reject the command. | Operation | CS\_OneShotCmd |
| CS8002.3 | If the command-specified rate is zero, CS shall calculate the CRC at the <PLATFORM\_DEFINED> rate (Max Bytes Per Cycle). | Operation | CS\_OneShotCmd |
| CS8003 | Upon receipt of a Cancel One Shot command, CS shall stop the current One Shot calculation. | Operation | CS\_CancelOneShotCmd  CFE\_ES\_DeleteChildTask |
| CS9000 | CS shall generate a housekeeping message containing the following:  a) Valid Ground Command Counter  b) Ground Command Rejected Counter  c) Overall CRC enable/disable status  d) Total Non-volatile Baseline CRC  e) OS code segment Baseline CRC  f) cFE code segment Baseline CRC  g) Non-volatile CRC Miscompare Counter  h) OS Code Segment CRC Miscompare Counter  i) cFE Code Segment CRC Miscompare Counter  j) Application CRC Miscompare Counter  k) Table CRC Miscompare Counter  l) User-Defined Memory CRC Miscompare Counter  m) Last One Shot Address  n) Last One Shot Size  o) Last One Shot Checksum  p) Checksum Pass Counter (number of passes thru all of the checksum areas)  q) Current Checksum Region (Non-volatile, OS code segment, cFE code segment etc)  r) Non-volatile CRC enable/disable status  s) OS Code Segment CRC enable/disable status  t) cFE Code Segment CRC enable/disable status  u) Application CRC enable/disable status  v) Table CRC enable/disable status  w) User-Defined Memory CRC enable/disable status  x) Last One Shot Rate  y) Recompute In Progress Flag  z) One Shot In Progress Flag | Operation | CS\_HousekeepingCmd  CFE\_SB\_TransmitMsg |
| CS9001 | Upon any Initialization of the CS Application (cFE Power On, cFE Processor Reset or CS Application Reset), CS shall initialize the following data to Zero:  a) Valid Ground Command Counter  b) Ground Command Rejected Counter  c) Non-volatile CRC Miscompare Counter  d) OS Code Segment CRC Miscompare Counter  e) cFE Code Segment CRC Miscompare Counter  f) Application CRC Miscompare Counter  g) Table CRC Miscompare Counter  h) User-Defined Memory CRC Miscompare Counter  i) Recompute In Progress Flag  j) One Shot In Progress Flag | Initialization | CS\_AppInit |
| CS9002 | Upon a cFE Power On Reset, if the segment's <PLATFORM\_DEFINED> Power-On Initialization state is set to Enabled, CS shall compute baseline CRCs for the following regions:  a) OS code segment  b) cFE code segment | Initialization | CS\_AppInit  CS\_BackgroundCfeCore  CS\_BackgroundOS  CS\_InitSegments |
| CS9003 | Upon a cFE Power On Reset, if the Non-Volatile <PLATFORM\_DEFINED> Power-On Initialization state is set to Enabled, CS shall compute baseline CRCs for Non-volatile segments based on the corresponding table definition for up to <PLATFORM\_DEFINED> segments. | Initialization | CS\_AppInit  CS\_BackgroundEeprom |
| CS9003.1 | If the address range for any of the Non-volatile segments is Invalid, CS shall send an event message and disable Non-volatile Checksumming | Initialization | CS\_AppInit  CS\_ComputeEepromMemory |
| CS9003.2 | CS shall send an event message and disable Non-volatile Checksumming, if the state is not one of the following:  a) enabled  b) disabled  c) empty | Initialization | CS\_AppInit |
| CS9004 | Upon a cFE Power On Reset, if the Non-Volatile <PLATFORM\_DEFINED> Power-On Initialization state is set to Enabled, CS shall compute the baseline CRC for the total of all of non-volatile segments. | Initialization | CS\_AppInit |
| CS9005 | Upon a cFE Power On Reset, if the Application <PLATFORM\_DEFINED> Power-On Initialization state is set to Enabled, CS shall compute baseline CRCs for the Application code segments region based on the corresponding table definition for up to a <PLATFORM\_DEFINED> Applications | Initialization | CS\_AppInit  CS\_BackgroundApp |
| CS9005.1 | CS shall send an event message and disable Application code segment Checksumming, if the state is not one of the following:  a) enabled  b) disabled  c) empty | Initialization | CS\_AppInit |
| CS9006 | Upon a cFE Power On Reset, if the Tables <PLATFORM\_DEFINED> Power-On Initialization state is set to Enabled, CS shall compute baseline CRCs for the tables specified in the corresponding table definition for up to <PLATFORM\_DEFINED> tables | Initialization | CS\_AppInit  CS\_BackgroundTables |
| CS9006.1 | CS shall send an event message and disable Table Checksumming, if the state is not one of the following:  a) enabled  b) disabled  c) empty | Initialization | CS\_AppInit |
| CS9007 | Upon a cFE Power On Reset, if the User-Defined Memory <PLATFORM\_DEFINED> Power-On Initialization state is set to Enabled, CS shall compute baseline CRCs for the User-Defined memory region based on the corresponding table definition for up to <PLATFORM\_DEFINED> memory segments. | Initialization | CS\_AppInit |
| CS9007.1 | If the address range for any of the User-Defined Memory is Invalid, CS shall send an event message and disable User-Defined Memory Checksumming |  | CS\_AppInit  CS\_BackgroundMemory |
| CS9007.2 | CS shall send an event message and disable Checksumming of the User-Defined Memory, if the state is not one of the following:   a) enabled  b) disabled  c) empty | Initialization | CS\_AppInit |
| CS9008 | Upon a cFE Processor Reset or CS Application Reset, if the <PLATFORM\_DEFINED> PRESERVE\_STATES\_ON\_PROCESSOR\_RESET Flag is set to True, CS shall preserve the following:  a) OS Code Segment Checksumming State  b) cFE Code Segment Checksumming State  c) Non-volatile Checksumming State  d) Application Code Segment Checksumming State  e) Table Checksumming State  f) User-Defined Memory Checksumming State | Initialization | CS\_AppInit  CS\_CreateRestoreStatesFromCDS |
| CS9009 | Upon a cFE Processor Reset or CS Application Reset, if the <PLATFORM\_DEFINED> PRESERVE\_STATES\_ON\_PROCESSOR\_RESET Flag is set to False, CS shall perform initialization in accordance with a Power On reset. | Initialization | CS\_AppInit |
| CS9010 | Upon a cFE Processor Reset or CS Application Reset, if the <PLATFORM\_DEFINED> PRESERVE\_STATES\_ON\_PROCESSOR\_RESET Flag is set to True and the segment's state is set to Enabled, CS shall compute baseline CRCs for the following regions:  a) OS code segment  b) cFE code segment | Initialization | CS\_AppInit  CS\_BackgroundCfeCore  CS\_BackgroundOS  CS\_InitSegments |
| CS9011 | Upon a Processor Reset or CS Application Reset, if the <PLATFORM\_DEFINED> PRESERVE\_STATES\_ON\_PROCESSOR\_RESET Flag is set to True and the Non-volatile Checksumming State is Enabled, CS shall compute baseline CRCs for Non-volatile segments based on the corresponding table definition for up to <PLATFORM\_DEFINED> segments. | Initialization | CS\_AppInit  CS\_BackgroundEeprom |
| CS9011.1 | If the address range for any of the Non-volatile segments is Invalid, CS shall send an event message and disable Non-volatile Checksumming | Initialization | CS\_AppInit  CS\_ComputeEepromMemory |
| CS9011.2 | CS shall send an event message and disable Non-volatile Checksumming, if the state is not one of the following:  a) enabled  b) disabled  c) empty | Initialization | CS\_AppInit |
| CS9012 | Upon a cFE Processor Reset or CS Application Reset, if the <PLATFORM\_DEFINED> PRESERVE\_STATES\_ON\_PROCESSOR\_RESET Flag is set to True and the Non-volatile Checksumming State is Enabled, CS shall compute the baseline CRC for the total of all of non-volatile segments | Initialization | CS\_AppInit |
| CS9013 | Upon a cFE Processor Reset or CS Application Reset, if the <PLATFORM\_DEFINED> PRESERVE\_STATES\_ON\_PROCESSOR\_RESET Flag is set to True and the Application Code Segment Checksumming State is Enabled, CS shall compute baseline CRCs for the Application code segments region based on the corresponding table definition for up to <PLATFORM\_DEFINED> Applications | Initialization | CS\_AppInit  CS\_BackgroundApp |
| CS9013.1 | CS shall send an event message and disable Application code segment Checksumming, if the state is not one of the following:  a) enabled  b) disabled  c) empty | Initialization | CS\_AppInit |
| CS9014 | Upon a cFE Processor Reset or CS Application Reset, if the <PLATFORM\_DEFINED> PRESERVE\_STATES\_ON\_PROCESSOR\_RESET Flag is set to True and the Table Checksumming State is Enabled, CS shall compute baseline CRCs for the tables specified in the corresponding table definition for up to <PLATFORM\_DEFINED> tables | Initialization | CS\_AppInit  CS\_BackgroundTables |
| CS9014.1 | CS shall send an event message and disable Table Checksumming, if the state is not one of the following:   a) enabled  b) disabled  c) empty | Initialization | CS\_AppInit |
| CS9015 | Upon a cFE Processor Reset or CS Application Reset, if the <PLATFORM\_DEFINED> PRESERVE\_STATES\_ON\_PROCESSOR\_RESET Flag is set to True and the User-Defined Memory Checksumming State is Enabled, CS shall compute baseline CRCs for the User-Defined memory region based on the corresponding table definition for up to <PLATFORM\_DEFINED> memory segments. | Initialization | CS\_AppInit  CS\_BackgroundMemory |
| CS9015.1 | If the address range for any of the User-Defined Memory is Invalid, CS shall send an event message and disable User-Defined Memory Checksumming | Initialization | CS\_AppInit |
| CS9015.2 | CS shall send an event message and disable Checksumming of the User-Defined Memory, if the state is not one of the following:  a) enabled  b) disabled  c) empty | Initialization | CS\_AppInit |