

**SOFTWARE REQUIREMENTS SPECIFICATION  
FOR THE  
INSTRUMENT A  
DATA PROCESSING UNIT  
FOR THE  
COMPANY X GAMMA RAY DETECTOR EXPLORER**

DPUSRS-01

Rev 2 Chg 2

March 4, 2003

Prepared by

X University

**SOFTWARE REQUIREMENTS SPECIFICATION  
FOR THE  
INSTRUMENT A  
DATA PROCESSING UNIT  
FOR THE  
COMPANY X GAMMA RAY DETECTOR EXPLORER**

DPUSRS-01

Rev 2 Chg 2

March 2003

REVISION NOTICE

Version Identifier	Date of Issue	Summary of Changes
WIP090600	September 6, 2000	Draft version for use in Software Requirements Review conducted on September 6, 2000, at PSU.
Rev 0 Chg 0	October 17, 2000	Initial baseline.
Rev 1 Chg 0	April 12, 2001	<p>The following ECRs are incorporated:</p> <p>eeprm-38 – DPU will have 3MB of EEPROM instead of 4MB.  dci-39 – Addition of DCI hardware windowing.  ssi-40 – Changes to SSI hardware design.  ccm-41 – Some ICUI CSC requirements reallocated to CCM.  ccm-42 – Make the DPU CVT a design issue.  tis-43 – Missing requirement to select alternate 1PPS.  dci-44 – DCI CSC requirement clarifications.  dpu-45 – Updates required for final ICU/DPU ICD.  ccm-46 – Missing req'ment for handling of obs messages.</p> <p>Additional non-ECR driven changes include:</p> <p>Removal of various TBDs and TBRs.  Updates to versions of Referenced Documents.  Corrections and clarifications to the context diagrams.  Updates and corrections to the Data Dictionary.  Design-driven updates to the EEPROM Memory Map  Miscellaneous editorial corrections and nomenclature updates.  Added traceability to SDS column to requirements tables.</p>
Rev 2 Chg 0	September 12, 2001	<p>Updated requirements matrix for traceability to the SDS and to unit and verification tests.</p> <p>Miscellaneous as-built updates to the EEPROM Memory Map.</p> <p>The following ECRs are incorporated:</p> <p>adc-56: Analog channel needed for ICU power bus voltage  bs-58: Clock resets on WDT eliminating warm boot features  tmali-68: Changes to various bootup default values  edac-73: Additional exception data added to EEPROM write  Company X-107: Correction to requirements verification levels</p>
Rev 2 Chg 1	April 9, 2002	<p>Updated requirements matrix for science software. Included traceability of requirements to SVP.</p> <p>Algorithms for some science software components added. Others will be completed after I&amp;T.</p> <p>Updated EEPROM memory map to include Science Software application configuration area.</p>
Rev 2 Chg 2	March 4, 2003	<p>Update document redlines from Build 6 Review #1.</p> <p>5.15.1.1 modified to say “compression software” since we will</p>

		<p>not be implementing a compression task. At the request of the science team, we have designed a lossy compression scheme, so the word “lossless” was stricken.</p> <p>5.15.1.2 modified to say “compression software” since we will not be implementing a compression task.</p> <p>5.15.3.2 same</p> <p>1.15.3.2 same</p> <p>5.15.2.1 No longer relevant in current design. No DCX task.</p> <p>5.15.2.2 No longer relevant in current design. No DCX task. This storage will be re-apportioned to DPA-TMALI and DPA-SCUI.</p> <p>5.15.3.1 We feel it is wiser to throw away new data until the current data will fit. The DPU hardware accommodates this design nicely.</p> <p>5.19.1.6 Moved to Build 7, as it is relevant to tracking software.</p> <p>5.19.3.1 Remove TBR.</p>
--	--	--

This document contains information that is as complete as possible. Where final numerical values or specification references are not available, best estimates are given and noted **TBR** (To Be Reviewed). Items which are not yet defined are noted **TBD** (To Be Determined). The following table summarizes the TBD/TBR items in this revision of the document, and supplements the revision notice above.

Section(s)	Description

---

TABLE OF CONTENTS

1.	Scope.....	1
1.1	System Overview.....	1
1.2	Document Overview.....	1
1.3	Relationship to Other Specifications .....	2
2.	Referenced Documents.....	3
3.	Abbreviations .....	8
4.	Overview.....	10
4.1	System Context.....	10
4.2	Operational Concepts .....	11
4.3	Constraints.....	16
4.4	Goals .....	17
4.5	Software Components.....	18
5.	Detailed Requirements .....	21
5.1	Bootstrap CSC.....	21
5.2	Operating System CSC.....	22
5.3	Built-In Tests CSC .....	23
5.4	Error Detection and Correction CSC.....	24
5.5	MIL-STD-1553B Driver CSC .....	24
5.6	Reserved Section .....	25
5.7	Analog-to-Digital Converter Driver CSC.....	25
5.8	Synchronous Serial Interface Driver CSC .....	25
5.9	Data Capture Interface Driver CSC .....	26
5.10	EEPROM Driver CSC .....	26
5.11	EEPROM Filesystem CSC .....	27
5.12	Command and Control CSC .....	28
5.13	Telescope Module Access Library and Interface CSC .....	29
5.14	SCU Interface CSC.....	30
5.15	Data Compression CSC .....	31
5.16	Time Synchronization CSC .....	32
5.17	Reserved .....	32
5.18	ICU Interface CSC.....	33
5.19	Data Processing Algorithms CSC .....	34
6.	Data Dictionary .....	35

## LIST OF APPENDICES

APPENDIX A – Software Requirements Matrix

APPENDIX B – EEPROM Memory Map

APPENDIX C – Data Processing Algorithms

---

LIST OF FIGURES

Figure 1 - Requirements Flow-Down.....	2
Figure 2. DPU Flight Software Context Diagram – System Interface Perspective .....	10
Figure 3. DPU Flight Software Context Diagram – Hardware Interface Perspective .....	10
Figure 4. DPU Flight Software States.....	11
Figure 5. DPU Data Flow Diagram .....	20
Figure 6. Bootstrap CSC Context Diagram.....	21
Figure 7. Operating System CSC Context Diagram.....	22
Figure 8. Built-In Tests CSC Context Diagram .....	23
Figure 9. Error Detection and Correction CSC Context Diagram.....	24
Figure 10. MIL-STD-1553B Driver Context Diagram .....	24
Figure 11. Analog-to-Digital Converter Driver Context Diagram .....	25
Figure 12. Synchronous Serial Interface CSC Context Diagram .....	25
Figure 13. Data Capture Interface Context Diagram .....	26
Figure 14. EEPROM Interface Driver Context Diagram .....	26
Figure 15. EEPROM Filesystem CSC Context Diagram.....	27
Figure 16 – Command and Control CSC Context Diagram.....	28
Figure 17. Telescope Module Access Library and Interface CSC Context Diagram.....	29
Figure 18. SCU Interface CSC Context Diagram .....	30
Figure 19. Data Compression CSC Context Diagram.....	31
Figure 20. Time Synchronization CSC Context Diagram .....	32
Figure 21. ICU Interface CSC Context Diagram .....	33
Figure 22. Data Processing Algorithms CSC Context Diagram.....	34

LIST OF TABLES

Table 1. DPU FSW States ..... 12

Table 2. DPU Science and Engineering Modes ..... 12

Table 3. DPU DRAM Usage Summary ..... 14

Table 4. Constraints ..... 16

Table 5. DPU Software Goals ..... 17

Table 6. Data Dictionary..... 35



## 1. SCOPE

This document defines the software requirements for the Company X INSTRUMENT A Data Processing Unit (DPU) Flight Software (FSW). This document is a Level 4 specification as defined in document 410.4-SPEC-0004, *Company X Missions Requirements Document*.

### 1.1 System Overview

The Company X observatory is the next in a series of medium-class explorer satellites and is the first-of-its-kind observatory for multi-wavelength transient astronomy. The goal of the Company X mission is to determine the origin of Gamma-Ray Detectors (GRDs) and to exploit data from these detectors to probe the early universe. Company X instrumentation will exploit newly discovered GRD afterglow characteristics to make a comprehensive study of approximately 1000 detectors over its planned three-year mission. Company X will determine the origin of GRDs, reveal how GRD blast waves interact with surroundings, and identify different classes of detectors and associated physical processes. To accomplish these mission goals, Company X employs three semi-autonomous science instruments. Instrument B is a wide-angle x-ray telescope that detects GRDs. On detection, the spacecraft slews in the direction of the GRD, bringing it into the view of two narrow-field telescopes for higher-resolution multi-wavelength observation. The narrow-field telescopes are Instrument C, and INSTRUMENT A.

The INSTRUMENT A makes the Company X observatory a complete multi-wavelength facility. Co-aligned with the other instruments, INSTRUMENT A provides simultaneous ultra violet (UV) and optical coverage over a certain field. The INSTRUMENT A is a powerful complement to the other instruments because of its UV capabilities and the absence of atmospheric extinction, diffraction, and background. Since INSTRUMENT A has photon-counting detectors that retain individual photon positions and timing information, it operates in a mode more similar to typical x-ray telescopes than typical optical telescopes. INSTRUMENT A consists of two separate processing units. The Instrument Control Unit (ICU) controls commanding of the telescope. The Data Processing Unit (DPU) handles data collection, processing, and formatting.

The DPU communicates with the ICU through the Synchronous Serial Interface (SSI), and receives raw photon position and timing data from detector electronics through the serial Data Capture Interface (DCI). Because the amount of raw event data that can be collected exceeds the INSTRUMENT A telemetry allocation, the DPU employs histogramming and lossless data compression to reduce the size of its data products. The DPU formats data as Consultative Committee for Space Data Systems (CCSDS) Source Packets, and forwards telemetry to the Spacecraft Control Unit (SCU) through a MIL-STD-1553 (1553) interface. The DPU maintains a local copy of the spacecraft clock to timestamp the telemetry.

### 1.2 Document Overview

This Software Requirements Specification describes the functional, interface, performance, and error recovery requirements for the DPU FSW. The following sections comprise the remainder of this document:

- Referenced Documents,
- Abbreviations,
- Overview,
- Detailed Requirements,
- Data Dictionary, and
- EEPROM Memory Map.

This specification contains a data dictionary (see Section 6) which defines the high-level data items for the DPU FSW. References to data dictionary data items in this specification are shown in the following font: **DATA\_ITEM**.

The actual requirements for each computer software component (CSC) are maintained in an electronic spreadsheet to facilitate requirements traceability and verification tracking. A copy of the spreadsheet is contained in Appendix A.

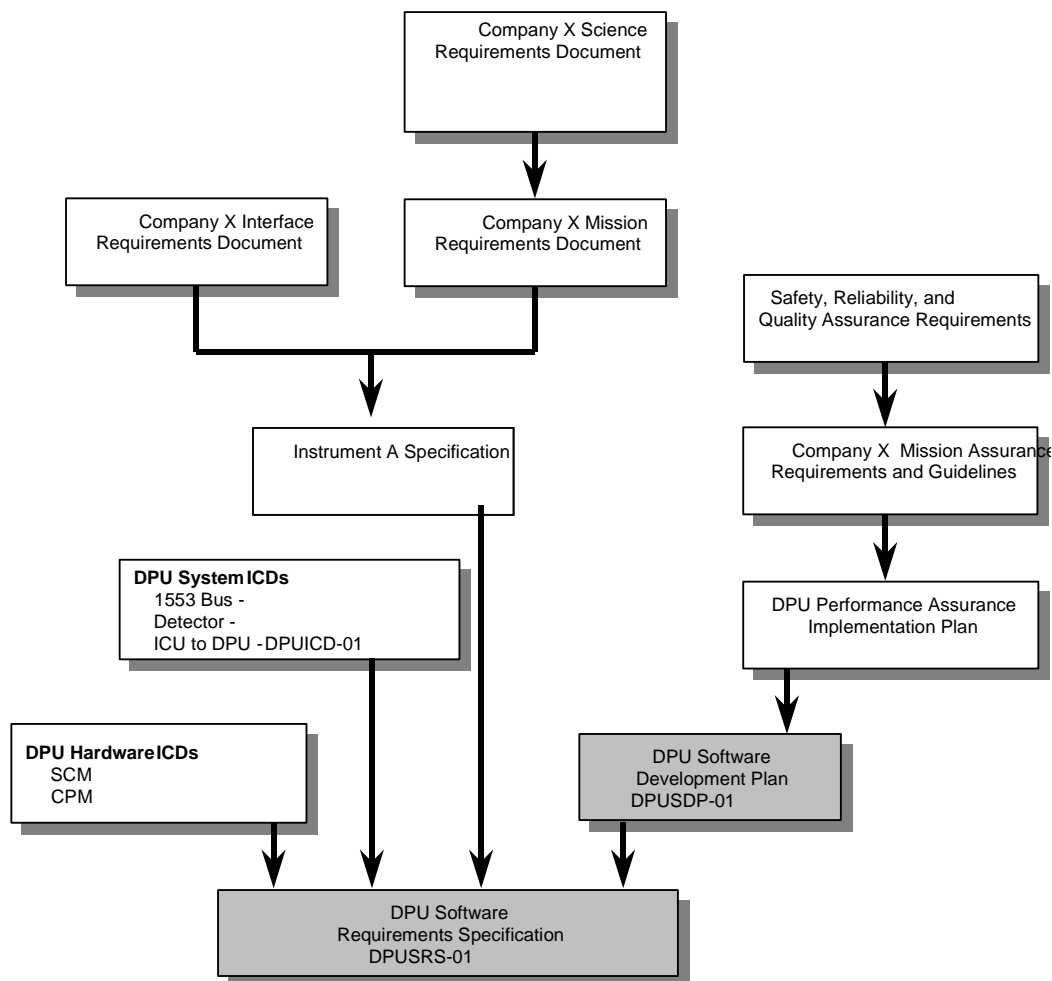
The electronic spreadsheet is configuration-controlled, and the copy attached to this document contains the version of the requirements applicable to the indicated revision of this document.

### 1.3 Relationship to Other Specifications

This specification is established in the Company X DPU Software Development Plan, document DPUSDP-01 and is required by the Standard Software Process (SSP) of the Software Engineering Department (SED) and by document PAIP-00-15-3691, *Company X DPU Performance Assurance Implementation Plan (PAIP)*. This document specifies the requirements for the design and development of the Company X INSTRUMENT A DPU FSW, and also serves as the basis for the DPU Software Verification Procedures test cases.

The DPU operates as a slave to the Company X INSTRUMENT A Instrument Control Unit (ICU). The interface between the DPU and the ICU are specified in document DPUICD-01, *Interface Control Document for the ICU to DPU / DPU to ICU Protocol for INSTRUMENT A*. Requirements for the ICU FSW are specified in document COMPANY X-INSTRUMENT A /SP/001.3, *COMPANY X-INSTRUMENT A ICU On-Board Software Requirements*.

The DPU FSW requirements flow down from upper-level specifications as shown in Figure 1. The applicability of the documents shown in the figure is described in Section 2.



**Figure 1 - Requirements Flow-Down**

## 2. REFERENCED DOCUMENTS

The following documents, of the exact issue shown, were referenced as indicated during the development of this SRS. The applicability statement associated with each document reference indicates *Superceding* if the referenced document supersedes this document in the event of a conflict.

Document ID:	
Originator:	Company X
Issue:	Rev 1 Chg 1 (May 2001)
Title:	Company X Specification for the Company X Communications Module
Applicability:	Specifies the interface to the Company X Communications Module in the DPU from which software requirements in this document were derived.
Document ID:	DPUICD-01
Originator:	Company X
Issue:	Rev 1 Chg 0 (June 2001)
Title:	Interface Control Document for the ICU to DPU/DPU to ICU Protocol for the Instrument A
Applicability:	Specifies the data protocol for the SSI-based interface to the ICU.
Document ID:	DPUSDP-01
Originator:	Company X
Issue:	Rev 1 Chg 0 (November 2000)
Title:	Software Development Plan for the Instrument A Data Processing Unit for the Company X Gamma Ray Detector Explorer.
Applicability:	Establishes and identifies this document, and describes the requirements analysis process used to produce it.
Document ID:	DPUTPUT-01
Originator:	Company X
Issue:	Version 17 (February 14, 2001)
Title:	DPU Throughput Analysis
Applicability:	A spreadsheet used to estimate the DPU CPU utilization and DRAM usage.
Document ID:	10-26977
Originator:	Company X
Issue:	January 2000
Title:	Company X Digital Electronics Module (DEM) Chassis and Data Processing Unit (DPU)
Applicability:	Is the proposal to X University for the system-level flight software activities which are addressed by this SRS.
Document ID:	10-26977A
Originator:	Company X
Issue:	Revision A, May 3, 2000
Title:	Company X Digital Electronics Module (DEM) Chassis and Data Processing Unit (DPU) Software Addendum
Applicability:	Is the proposal to University X for the application framework flight software activities which are addressed by this SRS.

Document ID: 1143  
Originator:  
Issue: 25 AUGUST 2000  
Title: Company X 1553 Bus Protocol Interface Control Document.  
Applicability: Specifies the instrument-generic interface between the remote terminal (RT) Instruments and the Spacecraft from which software requirements in this document are derived. *Superseding.*

Document ID: 1143  
Originator:  
Issue: Rev – (06 JUNE 2001)  
Title: Spacecraft to Payload Telecommand Interface Control Document  
Applicability: Defines the various messages which will be transmitted between the Spacecraft and the various Instruments.

Document ID: 7384  
Originator: Company X  
Issue: Rev 0 Chg 0 (February 1997)  
Title: Bootstrap Monitor Protocol Specification for the Space Station Furnace Facility Control Units.  
Applicability: Specifies the Bootstrap Monitor interface for the SSFF CUs. The SSFF Bootstrap Monitor was reused on X with minimal modifications to the user interface. The X bootstrap will be reused on the DPU with minor adjustments to accommodate hardware address differences. Therefore, the protocol and user interface documented in the referenced specification are relevant.

Document ID: 8089  
Originator: Company X  
Issue: Rev 0 Chg 0 (April 1998)  
Title: Software Design Specification for the Central Instrument Data Processor for X  
Applicability: Specifies the design for the X reuse CSCs which are referenced by this document.

Document ID: 8089  
Originator: Company X  
Issue: Rev 0 Chg 1 (March 1999)  
Title: Software Requirements Specification for the Central Instrument Data Processor for X  
Applicability: Specifies the requirements for the X reuse CSCs which are referenced by this document.

Document ID: 9000-0013  
Originator:  
Issue: June 1999  
Title: Phase A Study Report  
Applicability: Describes the Company X mission science goals, overall observatory design, and outlines the high level component design and integration plans.

Document ID: DOC  
Originator:  
Issue: 5.3.1 (March 1997)  
Title: VxWorks Programmer's Guide  
Applicability: Describes VxWorks conventions from which are derived some error reporting and other conventions in these requirements.

Document ID: 410.4-SPEC  
Originator:  
Issue: Version 1.0 (August 21, 2000)  
Title: Company X Science Requirements Document  
Applicability: Defines the Company X mission and specifies high-level requirements for the Company X observatory, and is the Level 1 specification for Company X. *Superseding.*

Document ID: 410.4-SPEC  
Originator:  
Issue: Version 1.1 (August 19, 2000)  
Title: Company X Mission Requirements Document  
Applicability: Specifies the mission requirements for the Company X observatory, and is the Level 2 specification for Company X. *Superseding.*

Document ID: ICD-0006  
Originator:  
Issue: REVISION 1.03 (April 3, 2002)  
Title: Onboard Operational Messaging Interface Document  
Applicability: Defines the messages to be transmitted by the Instrument B and Figure of Merit (FoM), and which describes the concept of operations for the Company X observatory.

Document ID: POWER-ARCH (for reference in the document only)  
Originator:  
Issue: Version 1.53 (July 22, 1991)  
Title: POWER Processor Architecture  
Applicability: Contains the procedure for accessing the Rios Single Chip (RSC)-VME processor board Real-Time Clock.

Document ID: FAULT-MGMT (for reference in the document only)  
Originator:  
Issue: Version 1.1 (February 3, 1992)  
Title: RSC System: Fault Handling and Storage Management  
Applicability: Contains a description of RSC processing unit fault handling and storage management facilities, from which software requirements in this document are derived.

Document ID: EMAIL (for reference in this document only)  
Originator:  
Issue: November 24, 1997  
Title: Email Company X - "FYI: RAD6000 Diagnostic Mode"  
Applicability: Contains a description of the behavior of the EDAC capability of the RAD6000 DRAM, from which software requirements in this document are derived.

Document ID: RSC-WB (for reference in this document only)  
Originator:  
Issue: September 10, 1996  
Title: RSC VME Engineering Workbook (Breadboard/EM/Flight FPGA-Based Configuration)  
Applicability: Contains design details of the RAD6000 CPU Module from which software requirements in this document are derived.

Document ID: MIL-STD-1553B  
Originator: Department of Defense, Washington DC  
Issue: September 21, 1978, with Notices 1 & 2  
Title: MIL-STD-1553B  
Applicability: Describes the MIL-STD-1553B standard referenced by software requirements within this document.

---

Document ID: PAIP-00-15-3691  
Originator: Company X  
Issue: Revision 0 (December 2000)  
Title: Company X DPU Performance Assurance Implementation Plan  
Applicability: Provides performance assurance guidelines for the Company X DPU project, as derived from the Company X Mission Assurance Requirements and Company X Mission Assurance Guidelines. *Superseding.*

Document ID: SED-SSP (for reference in this document only)  
Originator: Company X  
Issue: April 2000  
Title: Software Engineering Department Standard Software Process  
Applicability: Specifies the standard processes and procedures for software development in the SED.

Document ID: COMPANY X-INSTRUMENT A-001-RD (also COMPANY X-INSTRUMENT A-002)  
Originator:  
Issue: Rev 0 Chg 0 (July 2000)  
Title: Company X Instrument Specification for the Instrument A  
Applicability: Specifies the science requirements for the INSTRUMENT A, and is the Level 3 specification for the DPU. *Superseding.*

Document ID: SWT-INSTRUMENT A/SP/001.3  
Originator:  
Issue: Issue 3 (December 2000)  
Title: COMPANY X-INSTRUMENT A ICU On-Board Software Requirements  
Applicability: Specifies the software requirements for the ICU.

Document ID: SUMMIT (for reference in this document only)  
Originator:  
Issue: 1994  
Title: Summit LX/DX 1553 Product Handbook  
Applicability: Describes the MIL-STD-1553B controller interface from which software requirements in this document are derived.

Document ID: XMM  
Originator:  
Issue: 8  
Title: XMM-OM Electrical Interfaces Specification  
Applicability: Specifies the hardware interface and timing of the SSI interface on the XMM-OM mission., which was referenced during the development of these software requirements.

Document ID: XMM  
Originator:  
Issue: 5  
Title: XMM-OM User Manual: On-board Software  
Applicability: Describes software previously used with similar low-level interfaces on the XMM-OM project. This information was referenced during the development of the performance/functional requirements for the Data Capture Interface.

Document ID: XMM  
Originator:  
Issue: 4  
Title: XMM-OM User Manual: ICU-DPU Protocol Definitions  
Applicability: Specifies the protocol used over the SSI interface on the XMM-OM mission, which was referenced during the development of these software requirements.

---

Document ID: XMM  
Originator:  
Issue: 0004.03 (1991)  
Title: DPU Processing for XMM/OM – Tracking and Compression Algorithm  
Applicability: Describes the data compression and science algorithms used on the XMM-OM mission, which are being reused in the DPU FSW design.

### **3. ABBREVIATIONS**

μsecs	Microseconds
1553	MIL-STD-1553B
1PPS	One Pulse Per Second
ADIO	Analog/Discrete I/O
Aka	Also Known As
API	Application Programming Interface
BIT	Built-In Test
C&DH	Command and Data Handling System
CCD	Charge Coupled Device
CCM	Command and Control Module CSC
CCSDS	Consultative Committee for Space Data Systems
CIDP	Central Instrument Data Processor
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
DCI	Data Capture Interface
DCX	Data Compression CSC
DPA	Data Processing Algorithm CSC
DPU	Data Processing Unit
DRAM	Dynamic Random Access Memory
EDAC	Error Detection And Correction
EEFS	EEPROM File System
EEPROM	Electrically Erasable Programmable Read-Only Memory
FC	Finding Chart
FSW	Flight Software
GRD	Gamma Ray Detector
GSE	Ground Support Equipment
GSFC	Goddard Space Flight Center
GSW	Ground Software
HK	Housekeeping
I/O	Input/Output
ICD	Interface Control Document
ICU	Instrument Control Unit
ICUI	ICU Interface CSC
IMAGE	Imager for Magnetopause-to-Aurora Global Exploration
ITOS	Integrated Test and Operations System
Kb	Kilo-bits
KB	Kilo-bytes
MB	Mega-bytes
MET	Mission Elapsed Time
MOC	Mission Operations Center
MS-DOS	Microsoft Disk Operating System



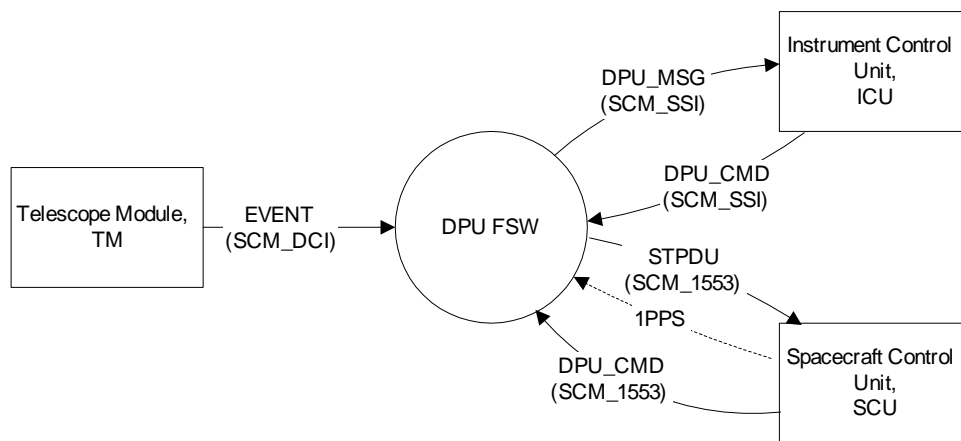
msec	Millisecond
PROM	Programmable Read-Only Memory
RSC	Rios Single Chip
RT	Remote Terminal; (1553 term for a science instrument on the 1553 bus)
CompanyXCM	Company X Communications/Memory Module
SCU	Spacecraft Control Unit
SCUI	SCU Interface CSC
SDAT	Science Data Analysis Terminal
SDP	Software Development Plan
SED	Software Engineering Department
SRS	Software Requirements Specification
SSFF	Space Station Furnace Facility
SSI	Synchronous Serial Interface
SSP	Standard Software Process
SVP	Software Verification Procedures
Company X	Company X
TBD	To Be Determined
TBR	To Be Reviewed
TDRSS	Tracking and Data Relay Satellite System
TMALI	Telescope Module Access Library Interface CSC
XMM-OM	X-Ray Multi-Mirror-Optical Monitor

## 4. OVERVIEW

This section provides an overview of the DPU FSW including a system context, operational concepts, goals and constraints, and a list of the computer software components (CSCs) which comprise the DPU FSW. Detailed functional, performance, error recovery, and interface requirements for the identified CSCs are provided in Section 5, Detailed Requirements.

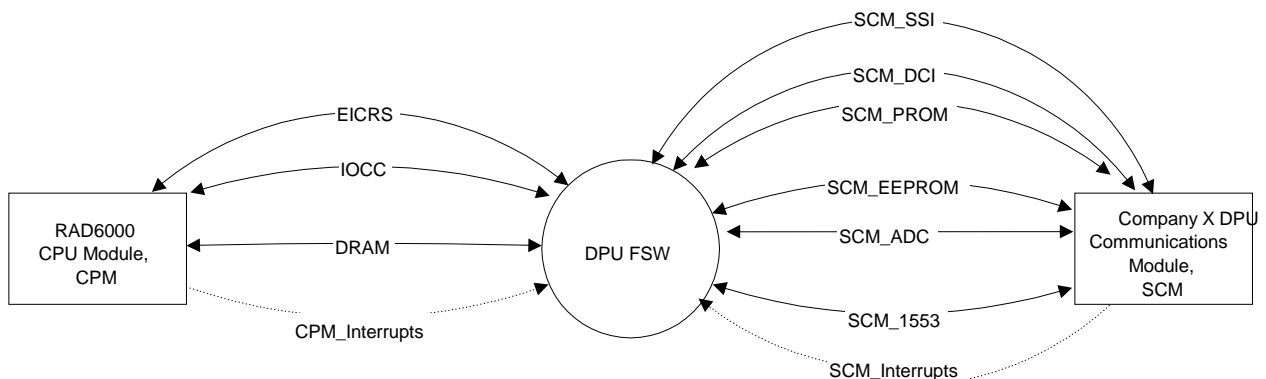
### 4.1 System Context

From a system (observatory) perspective, the DPU FSW interfaces with the SCU via the MIL-STD-1553B bus and the one pulse per second (1PPS) interface, with the ICU via the SSI, and with the detector electronics via the DCI. The interfaces of the DPU FSW in this context are illustrated in the following figure.



**Figure 2. DPU Flight Software Context Diagram – System Interface Perspective**

From a hardware interface perspective, the DPU FSW runs on the RAD6000 processor on the RAD6000 CPU Module (CPM). The Company X Communications Module (SCM) contains the EEPROM, and the System Interfaces. The System Interfaces Include the SSI, the DCI, and the MIL-STD-1553B Bus. The interfaces of the DPU FSW in this context are illustrated in the following figure.



**Figure 3. DPU Flight Software Context Diagram – Hardware Interface Perspective**

## 4.2 Operational Concepts

The following sections describe the operational concepts of the DPU FSW, including its functions, interfaces, performance characteristics, error detection, reporting and recovery mechanisms, and ground systems concepts.

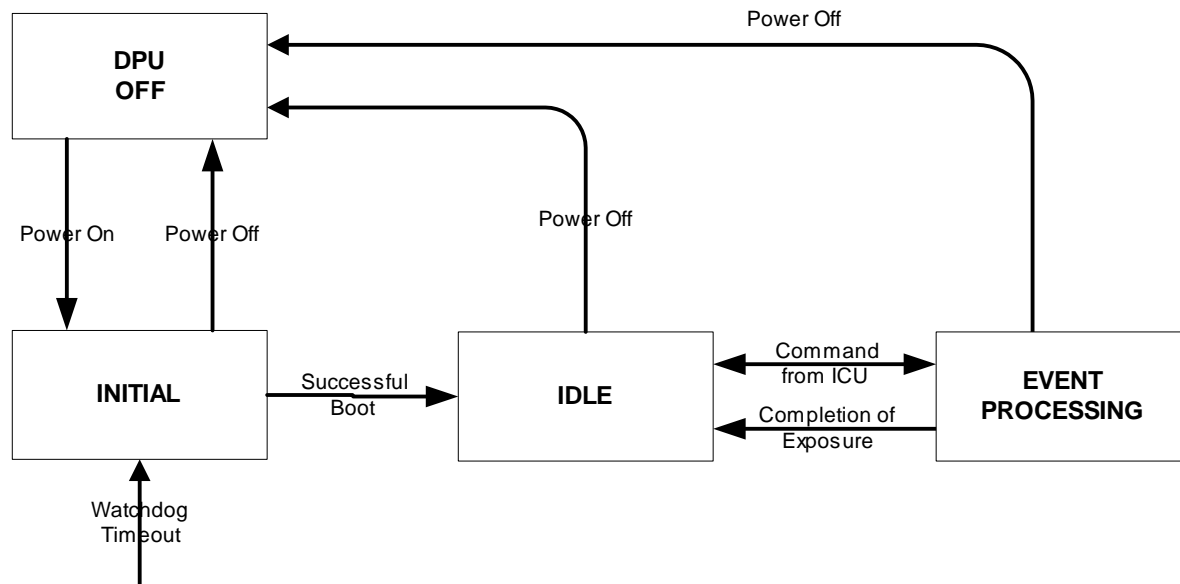
### 4.2.1 Functions

The DPU FSW is primarily a data processing slave for the INSTRUMENT A Instrument under control of the ICU. The DPU FSW has the following primary functions:

- Receive science data in the form of detector (photon) events from the telescope, process the events in accordance with the commanded mode, optionally compress the data, and relay the resulting data product(s) to the SCU in the form of CCSDS Source Packets.
- Receive commands from the ICU which establish the current event processing mode, exposure time, and related parameters.
- Derive channel boundaries and transmit them to the ICU, on command from the ICU.
- Calculate observatory drift by tracking bright stars in a reference image and use to drift-correct detector events.
- Provide a DPU heartbeat and communicate state of health information to the ICU.
- Transmit detailed housekeeping data to the SCU in the form of CCSDS Source Packets.
- Receive a time message from the SCU and synchronize the DPU local copy of the spacecraft clock using the one pulse-per-second (1PPS) signal.

#### 4.2.1.1 Software States

The DPU FSW has three states: Initial, Idle, and Event Processing. The following diagram illustrates these states and their transitions.



**Figure 4. DPU Flight Software States**

The following table describes the DPU states and, for each state, indicates whether the FSW is commandable, whether the FSW produces any telemetry, and whether or not detector events are processed.

Table A. DPU FSW States					
State	Entered On	Description	Command	Telemetry	Events
Initial	Cold Boot, or Warm Boot, or Command	Bootstrap software executes, performs self-test, and then boots the application FSW from EEPROM. The DPU will remain in the Initial State if the DPU FSW fails to boot after a programmable number of attempts.	No	No	Discarded
Idle	After Initial, or Exposure Completion, or Command	Normal housekeeping is produced. Data compression and transmission of residual science/engineering telemetry from prior exposures may be in progress.	Yes	Yes	Discarded
Event Processing	Command	DPU FSW is receiving and processing detector events and producing housekeeping and science/engineering telemetry.	Yes	Yes	Processed

#### 4.2.1.2 Science and Engineering Modes

The DPU FSW has multiple science and engineering modes. These modes determine how the DPU FSW processes detector events, and what science and engineering data packets the DPU FSW produces. Each mode is entered on ICU command, and is exited on command or when the exposure time commanded by the ICU has expired. The DPU FSW may be in only one mode at a given time. These modes are listed and described in the following table.

Table B. DPU Science and Engineering Modes		
Mode and Description	Science Phases	Data Product(s)
Event (Ev) Mode The purpose of Event Mode is to gather early photon events as soon as possible following GRD detection. This mode may also be used to produce timing information of astrophysical objects.	Transient	<b>PROD_EVENT</b> <b>PROD_ACS_MSG</b> (if commanded)
	Early GRD Exposures and some pre-planned exposures	<b>PROD_EVENT</b>
Image (Im) Mode The purpose of Image Mode is to produce drift-corrected images of the target of interest during GRD and pre-planned observations. Histogrammed images are the primary science product. For new GRD observations, a Finding Chart is produced, whose purpose is to rapidly produce a data set allowing ground-based observatories to locate the GRD.  Detector events produced in the commanded detector mode are received, drift-corrected, and histogrammed over a commanded integration time. A commanded software window is applied to the image and the reduced image is transmitted to the ground.	Pre-planned observations and later stages of GRD observation phases	<b>PROD_IMAGE</b> <b>PROD_FINDING_CHART</b> (if commanded) <b>PROD_TRACKING_REC</b> (if tracking is enabled)

**Table B. DPU Science and Engineering Modes**

<b>Mode and Description</b>	<b>Science Phases</b>	<b>Data Product(s)</b>
<p>Image / Event (IE) Science Mode</p> <p>The purpose of Image/Event Mode is to provide both drift-corrected, histogrammed images while simultaneously providing high time resolution data.</p> <p>Image/Event Mode is a mode that combines the Image and Event modes.</p>	Early GRD Exposures and some pre-planned exposures	<p><b>PROD_IMAGE</b></p> <p><b>PROD_EVENT</b></p> <p><b>PROD_FINDING_CHART</b> (if commanded)</p>
<p>Full-Frame (FF) Engineering Mode</p> <p>The purpose of Full-Frame Engineering Mode is to monitor the health of the detector to locate hot spots and dead pixels.</p> <p>Detector events produced in Full Frame High Resolution detector format are received and histogrammed for a commanded integration time. The entire image is transmitted to the ground. This mode can use high resolution (all pixels) or low resolution imaging (2x2 , 4x4 pixel binning).</p>	Testing and Calibration	<b>PROD_IMAGE</b>
<p>Raw Event List (RE) Engineering Mode</p> <p>The purpose of Raw Event List Engineering Mode is to diagnose problems with the detector.</p> <p>Detector events produced in a given detector mode are collected over a commanded integration time and transmitted to the ground as a list of raw events.</p>	Testing and Calibration	<b>PROD_EVENT</b>
<p>Channel Boundary (CB) Engineering Mode</p> <p>The purpose of Channel Boundary Mode is to derive the optimum channel boundary settings from a flat field.</p> <p>Detector events produced in the M,N detector mode are collected and histogrammed for a commanded integration time. The "pseudo image" is transmitted to the ground. Channel boundaries are derived from the image and transmitted to the ICU and the ground.</p>	Testing and Calibration	<p><b>PROD_CHAN_BOUND</b></p> <p><b>PROD_MN_DATA</b></p>
<p>Centroiding Confirmation (CC) Engineering Mode</p> <p>The purpose of Centroiding Confirmation Engineering Mode is to determine the validity of derived channel boundaries.</p> <p>Detector events produced in the Full Frame High-Resolution detector format are collected and histogrammed for a commanded integration time. The full field is divided into sub-images, each of centroided by pixels. These sub-images are then modulo binned to produce a set of pixels pseudo-images. The images are column (y axis) ordered, as are the pixels within them. These pseudo images are transmitted to the ground.</p>	Testing and Calibration	<b>PROD_CENTROID_CONF</b>
<p>Intensifier Characteristics (IC) Engineering Mode</p> <p>The purpose of Intensifier Characteristics Engineering Mode is to assess detector health and performance.</p> <p>Detector events produced in Full Frame High Resolution detector format are collected and histogrammed for a commanded integration time.</p>	Testing and Calibration	<b>PROD_INTENSE_CHAR</b>

#### 4.2.2 Interfaces and Performance

The DPU FSW must receive and process detector events via the DCI at a maximum rate of a certain number of events per second. Theoretical event flow is estimated at X events per second for most filters, with Y events per second possible for the white filter. Each event is A bits; however, the events are transferred from the DCI to the

processor DRAM via the VME bus in D32 mode that results in B-byte events. Therefore, the maximum data rate over the DCI is so many bytes per second. When conducting an exposure, the telescope scans the CCD surface for photon arrivals every so many milliseconds. The detector electronics calculates each photon arrival to a location within a pixel area within the CCD pixel in which the event occurred, resulting in a total image size of so many pixels. Detector events are histogrammed to a depth of B bits. This results in a memory usage of a certain amount of megabytes (MB) for one full-frame image.

The INSTRUMENT A is expected to observe an average of one GRD per day. When a GRD is detected, the DPU will stop any pre-planned observation and discard remaining unprocessed detector events (and, if commanded, completed data products awaiting compression or transmission to the SCU) in preparation for observation of the GRD. During a GRD observation, the DPU will produce multiple data products as described in Table B, resulting in an estimated amount of data to the SCU. This value includes packetization overhead and is based on the baseline GRD observation timeline and analysis presented in DPUTPUT-01, *DPU Throughput Analysis*.

The INSTRUMENT A interface with the SCU provides for a daily average of C kbps per second of science data, with a peak rate of D kilobits per second, which can be sustained so long as the total amount of data transmitted does not exceed the average daily rate.

Therefore, the INSTRUMENT A can transmit a maximum of so many MB of data per day. Subtracting the amount of data produced by a single GRD, the INSTRUMENT A has E MB of data bandwidth remaining to conduct pre-planned observations.

The following table summarizes DPU DRAM usage, based on the analysis presented in DPUTPUT-01.

Table C. DPU DRAM Usage Summary			
Process	Usage	Amount	Notes
N/A	Program & Misc Data Space	A MB	Memory for the VxWorks™-based flight software and space for miscellaneous data structures.
Data Capture	Event Queue	A MB	Accommodates getting E seconds behind during calculation of Parameterized Finding Chart, with full F events/second input rate. The amount allocated to this queue is configurable at bootup.
Data Processing Event Mode	Event Buffer	B MB	Accommodates full-frame events with full F events/second input rate, with an Event packet produced after a maximum of seconds of event accumulation.
Data Processing Tracking	Reference Frame	B MB	The frame used as the star zero-drift position reference for the tracking process. Assumes full-frame image, and each pixel is D bits deep.
Data Processing Tracking	Current Frame (Ping & Pong)	C MB	The frame in which newly arriving events are histogrammed, prior to shift-and-add.
Data Processing Image Mode	Accumulation Frame	B MB	The frame containing the drift-corrected image currently being accumulated.
Data Processing Data Product Generation	Product Frame	B MB	The frame containing a completed image exposure which is being rasterized into CCSDS packets.

Table C. DPU DRAM Usage Summary			
Process	Usage	Amount	Notes
Data Compression	Data Compression Input Queue	B MB	The size of the queue is based on the maximum amount of data expected in the compression queue during a detector observation. The amount allocated to this queue is configurable at bootup.
Packetization Spacecraft Transfer	Spacecraft Transmit Queue	B MB	The size of the queue is based on the maximum amount of data expected in the spacecraft transfer queue during a detector observation. The amount allocated to this queue is configurable at bootup.
	<b>Total Memory Required</b>	<b>D MB</b>	
	<b>Total Memory Available</b>	<b>128 MB</b>	
	<b>Total Margin</b>	<b>??%</b>	

The nominal command route is from the ICU via the SSI interface. State of health data is transmitted to the ICU over the same interface. The number of messages between the ICU and the DPU will be much lower than the interface bandwidth and so the SSI interface throughput is not a design driver.

### 4.2.3 Error Detection, Reporting and Recovery

The following sections summarize the error handling, reporting and recovery mechanisms of the DPU. The DPU is not responsible for the health and safety of the INSTRUMENT A instrument.

#### 4.2.3.1 Software Configuration Integrity

The DPU maintains primary and alternate FSW configurations in EEPROM. The primary FSW configuration in EEPROM is hardware strapped to be read-only. The primary FSW always provides MIL-STD-1553B communications with the SCU provided there is no hardware failure. The DPU bootstrap software autonomously switches to the alternate FSW configuration in the event the primary configuration fails to boot.

Problems with the DPU FSW that are identified on-orbit can be corrected by patch or by a complete software reload. The DPU FSW contains an EEPROM-resident filesystem on which an object file containing a software patch can be loaded. This object file can then be dynamically loaded and linked into the active FSW. Alternatively, a complete software build can be loaded to the alternate FSW location in EEPROM and the DPU commanded to boot the alternate configuration rather than the primary.

#### 4.2.3.2 Memory Error Detection and Correction

The DPU DRAM, PROM, and EEPROM are designed with hardware EDAC. A software scrubber task walks DRAM to trip single-bit errors (SBEs) before they become uncorrectable multiple-bit errors (MBEs). The DPU FSW maintains a count of SBEs and MBEs that occur in DRAM and EEPROM, and the location of the last SBE and MBE. This table is reported to the ground as a part of regular housekeeping telemetry. In the event of an uncorrectable DRAM error, the DPU FSW will record the contents of the memory error table to EEPROM and wait for a watchdog reset to occur at which time the memory is re-initialized. EEPROM memory errors must be detected by ground systems and corrected by uploading the correct value.

#### 4.2.3.3 Error Reporting

Errors are reported in DPU housekeeping telemetry. Errors and events which result in the DPU failing to perform an action commanded by the ICU, or which affect the health and safety of the Instrument, are reported to the ICU. If an unrecoverable error occurs (such as a program exception) the DPU will record the current exception vector and memory error table to EEPROM and force a reboot via hardware watchdog timer.

#### 4.2.3.4 Keep Alive Messaging

The DPU periodically transmits a heartbeat message to the ICU. The ICU interprets the absence of the heartbeat messages as a failure in the DPU. Refer to documents DPUICD-01 and INSTRUMENT A for detail on the ICU's response to the absence of heartbeat messages from the DPU.

No heartbeat message is transmitted to the SCU by the DPU. A heartbeat message from the ICU to the SCU serves to indicate "aliveness" of the INSTRUMENT A Instrument to the Spacecraft.

#### 4.2.4 Ground Systems

Ground systems are required for the following purposes:

- Low-level driver integration and testing,
- DPU FSW integration and verification testing, and
- Operational display of downlinked data.

Low-level driver integration and testing is accomplished using a DPU-resident test application that exercises the hardware interfaces via actual flight software drivers. A application running on the Ground Support Equipment (GSE) communicates with the test application running on the DPU through an RS-232 port (unused on flight). The GSE-resident application commands the DPU-resident application to output or receive data on a particular hardware interface. The GSE has a direct connection to each hardware interface, and stimulates or measures the interface in accordance with the command sent to the DPU. Because the RS-232 port is not used on flight, each interface can be tested without interfering with the command-and-response communication between the DPU and GSE resident applications.

FSW integration and verification is accomplished by using simulators. A Detector Simulator supplies playback detector events to the DPU. An ICU Simulator provides a means to transmit ICU-to-DPU commands to the DPU and to receive and record DPU data alerts. A Spacecraft Simulator provides a means to send ground commands to the DPU and to forward telemetry packets to the ground system.

To simplify integration, testing will employ the same command and telemetry system that will be used at the Mission Operations Center (MOC). Conceptually, this means test scripts, the command and telemetry database, housekeeping displays, and science data visualization software could be reused for mission operations. The Integrated Test and Operations System (ITOS) provides a display interface for housekeeping telemetry. The Science Data Analysis Terminal (SDAT) provides a display interface for science telemetry. The SDAT system receives data from the ITOS system via TCP/IP connection or file transfer.

### 4.3 Constraints

Certain constraints are imposed upon the specification and design of the DPU FSW, and are derived from upper-level specifications and known system design constraints. These constraints are listed in Table D, along with the implications of the constraint.

Table D. Constraints			
#	Constraint	Implication(s)	Source
C1	The TDRSS downlink bandwidth is limited to A kbps.	The INSTRUMENT A transmit a Finding Chart (FC) down the TDRSS link within an end-to-end time of B seconds from initial detector detection. C seconds of the B seconds are consumed in slewing, various processing, and transmission to the S/C. The size of the FC image is expected to be D MB, which is larger than TDRSS can accommodate. The DPU FSW will have to create a very small parameterized FC for transmission via TDRSS in order to meet the requirement, and send the FC image down the regular science link.	410.4-SPEC-0004



<b>Table D. Constraints</b>			
<b>#</b>	<b>Constraint</b>	<b>Implication(s)</b>	<b>Source</b>
<b>C2</b>	Malindi ground contacts are limited to a maximum of fifteen E-minute ground contacts per day, with an uplink rate of D kbps and a downlink rate of B mbps. This is based on a F-minute orbit period.	The design of the DPU FSW must avoid any time-consuming setup or configuration procedures as part of its nominal operations. This may also have implications for the way in which large software loads/dumps are structured and/or performed.	Company X Phase A Report, section 3.6.6.2
<b>C3</b>	The interface with the S/C provides that real-time HK packets be limited to F bytes or less. In addition, all Instrument HK will be placed into the last F bytes of the regular S/C RT HK frame.	The DPU FSW design must structure its HK packets such that the F-byte constraint is not violated. In addition, the HK rate should be optimized to help ensure that a DPU HK packet can reasonably appear in the S/C frame at an acceptable rate. It is not clear whether this also has implication to memory dumps. The DPU FSW may have to provide for small dump packets if going down the RT link, and larger ones if going to the SSR.	Document 1143 section 4.8.3.
<b>C4</b>	The S/C does not reassemble segmented packets and the ITOS ground system does not currently capable of reassembling segmented packets.	Any packet which must be recognized and processed by ITOS should not be segmented. Based on a meeting with John Doe on some date, ITOS will be upgraded to reassemble packets, but it is not yet known when this capability will be implemented. This may also have impact to memory dump packets.	Document 1143 section 4.8. Meeting with John Doe on Some Date.
<b>C5</b>	The ITOS ground system will not be capable of decompressing packets.	Any packet which must be recognized and processed by ITOS (HK telemetry and the TDRSS-borne Parameterized Finding Chart) cannot be compressed by the DPU FSW.	Meeting with John Doe on Some Date.

## 4.4 Goals

Table E presents goals that serve to guide the specification, design, and development of the software. These goals should contribute to the simplicity (S), reliability (RI), maintainability (M), reusability (Ru), and testability (T) of the system.

<b>Table E. DPU Software Goals</b>						
<b>Goal</b>		<b>S</b>	<b>RI</b>	<b>M</b>	<b>Ru</b>	<b>T</b>
<b>G1</b>	Maintain simple, consistent data flow interfaces between the DPU and its external interfaces.	X				X
<b>G2</b>	Produce a design which requires as little a-priori knowledge of the internal operations of the ICU and the SCU as possible.	X		X		X
<b>G3</b>	Produce modular, project-generic designs and code to maximize reusability on other system components and on future projects. This should be done in such a way as to minimize modifications required as a result of project or component-specific design, coding, comments, or naming conventions.			X	X	
<b>G4</b>	Produce a design which provides for upgrade and maintenance			X		
<b>G5</b>	Produce a design which includes the necessary mechanisms and flexibilities to support ground I&T and to provide for off-nominal configurations in flight.		X			X
<b>G6</b>	Produce a design which maximizes reuse of software components from the X, XX, and XXX projects.				X	
<b>G7</b>	Produce error-free code.		X			

Table E. DPU Software Goals						
Goal		S	RI	M	Ru	T
<b>G8</b>	Minimize the amount of re-work necessary at each level of integration.		X		X	X
<b>G9</b>	Produce a design which is reasonably fault-tolerant.		X			

## 4.5 Software Components

The Software Development Plan for the DPU lists and identifies the computer software configuration items (CSCIs) for the DPU FSW. The following sections describe these CSCIs and their components.

### 4.5.1 System Software

This section lists and describes the computer software components (CSCs) of the DPU System Software CSCI, identified as DPUFSW-01. The DPU System Software consists of a bootstrap, a real-time operating system, built-in test software, and a set of interface device drivers. This list represents the list of CSCs which are evident at the requirements phase; additional CSCs may be identified at design time.

The Bootstrap CSC, identified DPU-BOOT, is a PROM-resident program which performs a basic hardware built-in-test (BIT), loads the DPU FSW from EEPROM, and provides a simple RS-232-based monitor useful during development for examining memory and for downloading programs.

The Built-In Tests CSC, identified DPU-BIT, provides a set of functions to perform and record the results of memory and interface tests on the hardware modules included in the DPU.

The Operating System, identified DPU-RTOS, provides a real-time, multi-tasking environment. The DPU-OPER-SYS is a COTS product, identified as VxWorks 5.3, kernel version WIND 2.5, from Wind River Systems, Alameda CA. The Operating System CSC is supplemented with a library of miscellaneous project-specific system utilities.

The Error Detection and Correction CSC, identified DPU-EDAC, provides a set of functions to facilitate the tracking, handling, and recording of memory errors.

The EEPROM Interface Driver, identified DPU-EEPRM, provides an application interface to the EEPROM on the Company X Communication/Memory Module (SCM).

The EEPROM File System CSC, identified DPU-EEFS, provides a file system which is media-compatible with MS-DOS. The file system facilitates dynamic loading of application programs using the VxWorks loader.

The Analog-To-Digital Converter Driver, identified DPU-ADC, provides an application interface to the analog-to-digital converter hardware on the SCM.

The MIL-STD-1553B Driver, identified DPU-1553, provides an application interface to the MIL-STD-1553B data bus hardware on the SCM.

The Synchronous Serial Interface Driver, identified DPU-SSI, provides an application interface to the SSI interface hardware on the SCM.

The Data Capture Interface Driver, identified DPU-DCI, provides an application interface to the DCI interface hardware on the SCM.

### 4.5.2 Application Software

This section lists and describes the computer software components (CSCs) of the DPU Application Software CSCI identified as DPUFSW-02. The DPU Application Software consists of command and control software, algorithmic modules, and interface software. The interface software manages communication with the detector, the ICU, and the Spacecraft Computer. This list represents the list of CSCs which are evident at the requirements phase; additional CSCs may be identified at design time.

---

The Command and Control CSC, identified DPU-CCM, is an application program that initializes the flight software, establishes and maintains the current system state, implements the command dispatch queue, collects housekeeping telemetry, monitors the running tasks, and is responsible for overall error handling.

The Instrument Control Unit Interface CSC, identified DPU-ICUI, is an application program that manages communications with the ICU over the SSI interface at the application data protocol level.

The Telescope Module Access Library and Interface CSC, identified DPU-TMALI, is an application program that handles the transfer of raw events from the DCI interface via the DCI driver, and makes these events available to the Data Processing Algorithm CSC.

The Data Processing Algorithm CSC, identified DPU-DPA, is an application program that receives and processes INSTRUMENT A detector events and produces science and engineering data products.

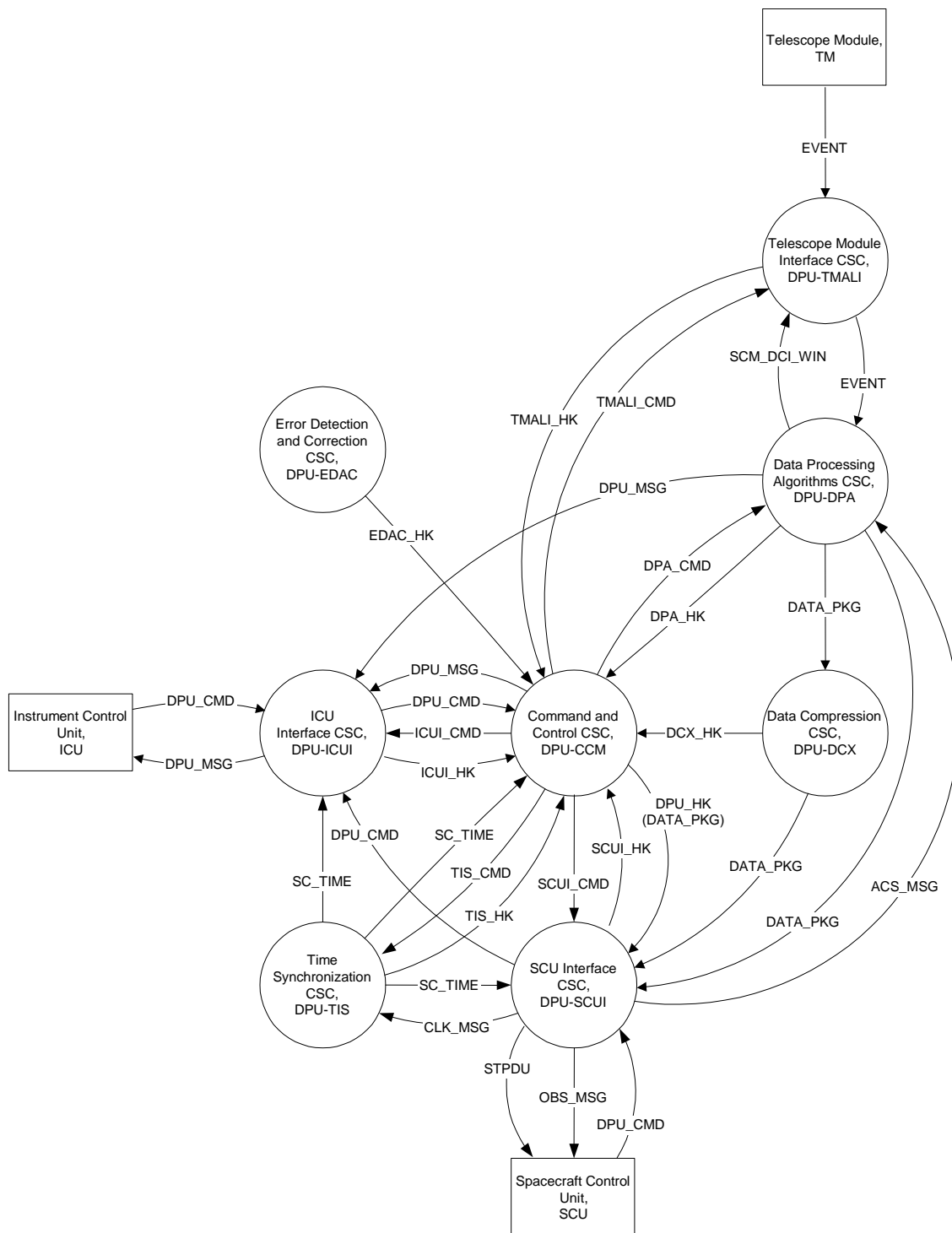
The Data Compression CSC, identified DPU-DCX, is an application program that compresses the science and engineering data products created by the Data Processing Algorithm CSC.

The Spacecraft Control Unit Interface CSC, identified DPU-SCUI, is an application program which reformats queued data as CCSDS Source Packets for transmission to the SCU, and which manages communications with the SCU over the 1553 interface at the application data protocol level.

The Time Synchronization CSC, identified as DPU-TIS, is an application program which maintains time synchronization with the spacecraft clock, and which provides access to the DPU clock via an API.

The relationship among the application software CSCs is illustrated in the following high-level Data Flow Diagram (DFD). The terminators in the diagram represent observatory components rather than device drivers to show the context of the application software within the satellite itself. The following control and data flows are not shown in order to avoid cluttering the diagram:

- CSCs which have associated operating system tasks report a task heartbeat to the CCM CSC, which are not shown.
- Each application CSC has an **ERRNO** flow to the CCM CSC, which are not shown.
- The CCM CSC has direct interfaces to several system software layer CSCs, which are not shown.



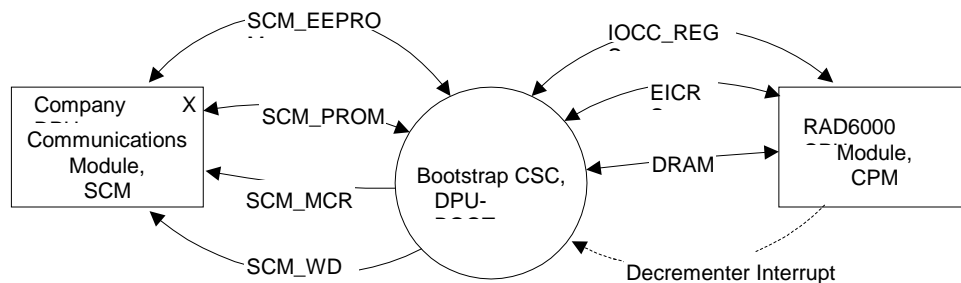
**Figure 5. DPU Data Flow Diagram**

## 5. DETAILED REQUIREMENTS

The following sections provide the detailed software requirements for the DPU FSW, broken down by CSC. Each section contains a context diagram for the CSC and any notes relevant to the CSC. The actual requirements are enumerated in an electronic spreadsheet to facilitate requirements traceability and verification tracking. A copy of this spreadsheet is contained in Appendix A. The electronic spreadsheet is configuration-controlled, and the copy attached to this document contains the version of the requirements applicable to the indicated revision of this document.

### 5.1 Bootstrap CSC

A context diagram for the Bootstrap CSC is shown in the following figure.

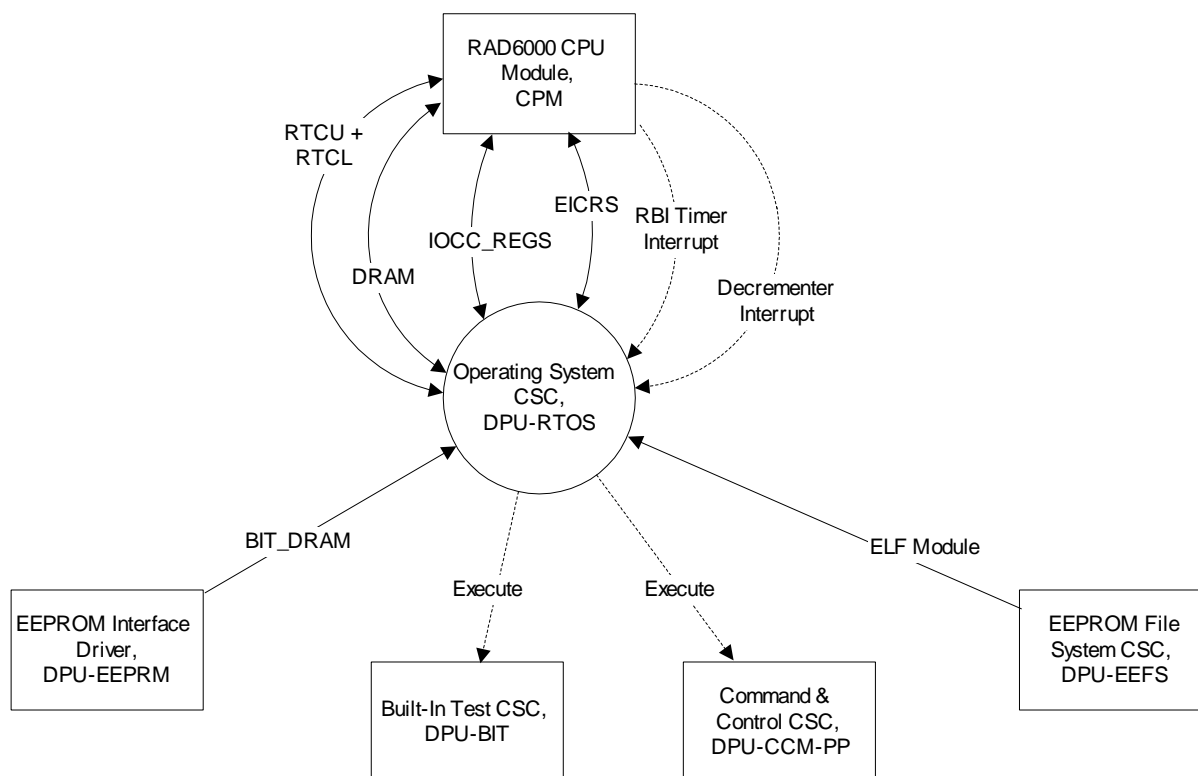


**Figure 6. Bootstrap CSC Context Diagram**

The Bootstrap CSC is a Level 1 reuse component from the X program. The X Bootstrap CSC requirements are therefore derived from document 8089-CIDPSRS-01. Because of the criticality of this CSC, and to facilitate verification, the entire list of requirements for the Bootstrap CSC are contained in Appendix A. Changes to the requirements are highlighted, and are limited to hardware interface differences between the X and the Company X DPU. In addition, references to X acronyms have been replaced with the equivalent Company X acronym in otherwise unmodified requirements.

## 5.2 Operating System CSC

A context diagram for the Operating System CSC is shown in the following figure. The diagram depicts hardware interaction only; software task interaction is not described.

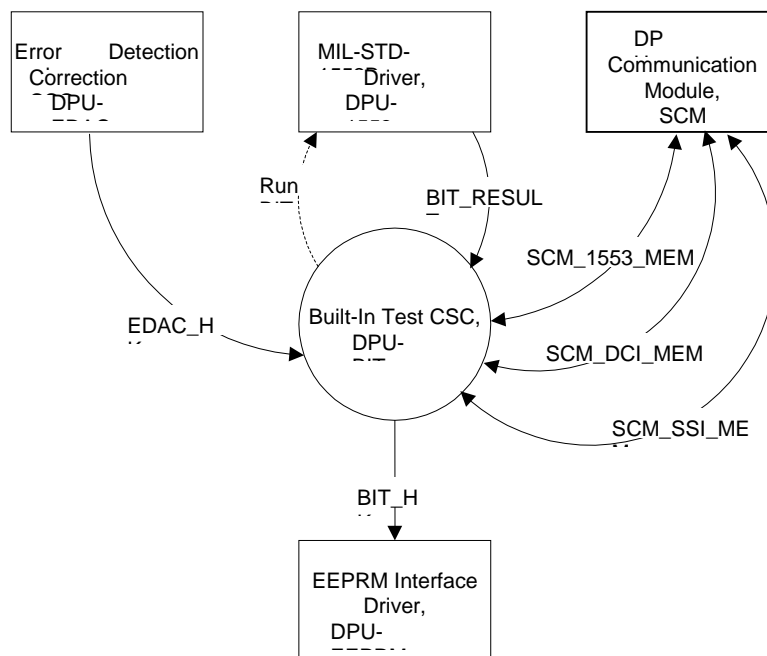


**Figure 7. Operating System CSC Context Diagram**

The Operating System is planned to be a COTS component called VxWorks™ with heritage from many space programs. The operating system libraries are complemented with some project-specific functions which also have heritage on the X and/or XX programs. The requirements for the Operating System CSC include high-level functional requirements and customization requirements only, and are listed in Appendix A.

### 5.3 Built-In Tests CSC

A context diagram for the Built-In Tests (BIT) CSC is shown in the following figure.

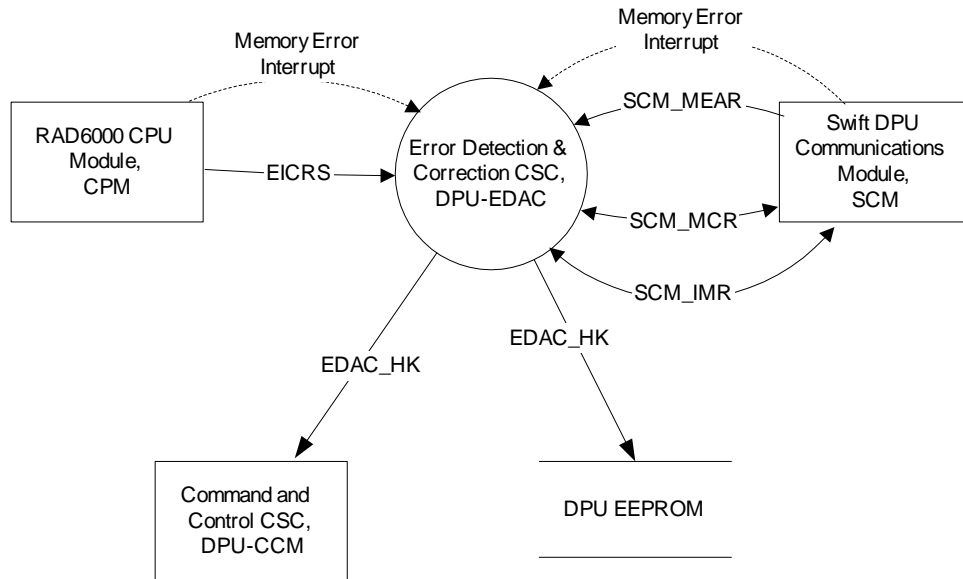


**Figure 8. Built-In Tests CSC Context Diagram**

The BIT CSC is a reuse component from the XX and X programs. However, while the data and program structure of the BIT remains the same, the actual tests performed by the BIT are project-specific. Therefore the BIT requirements (rather than changes only) are listed in Appendix A.

## 5.4 Error Detection and Correction CSC

A context diagram for the Error Detection and Correction (EDAC) CSC is shown in the following figure.

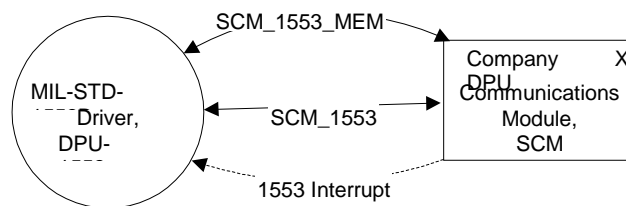


**Figure 9. Error Detection and Correction CSC Context Diagram**

The EDAC CSC is a reuse component from the X program. While the EDAC CSC for the DPU will be virtually identical to the X EDAC CSC, some of the requirements specified for the X EDAC CSC were unclear and therefore the complete requirements for the EDAC CSC are listed in Appendix A.

## 5.5 MIL-STD-1553B Driver CSC

A context diagram for the MIL-STD-1553B (1553) Driver CSC is shown in the following figure.



**Figure 10. MIL-STD-1553B Driver Context Diagram**

The 1553 Driver is a heritage software component from the XX and X programs. The requirements for the 1553 driver are described in documents 7384-SRS-01 and 8089. Changed requirements are listed in Appendix A, and are limited to hardware interface differences between the X and the Company X DPU.

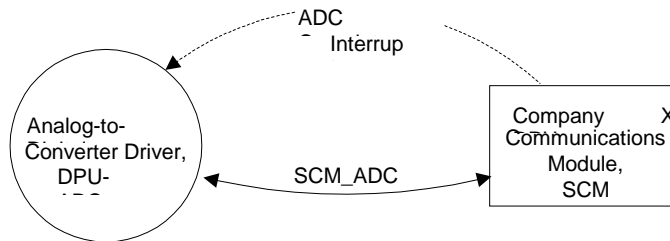


## 5.6 Reserved Section

A number of CSCs are being reused from the X FSW, whose requirements are described in document 8089. In order to maintain requirement number synchronization with the X SRS, this section is inserted as a reserved section.

## 5.7 Analog-to-Digital Converter Driver CSC

A context diagram for the Analog-to-Digital Converter (ADC) Driver CSC is shown in the following figure.

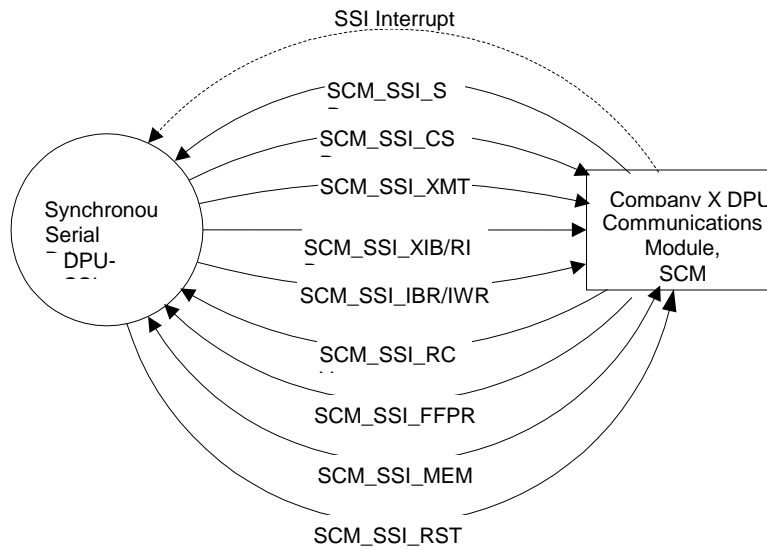


**Figure 11. Analog-to-Digital Converter Driver Context Diagram**

The ADC Driver is a custom device driver for the Company X INSTRUMENT A DPU, with some heritage software from the X program. The requirements for the ADC Driver are listed in Appendix A.

## 5.8 Synchronous Serial Interface Driver CSC

A context diagram for the Synchronous Serial Interface (SSI) Driver CSC is shown in the following figure.

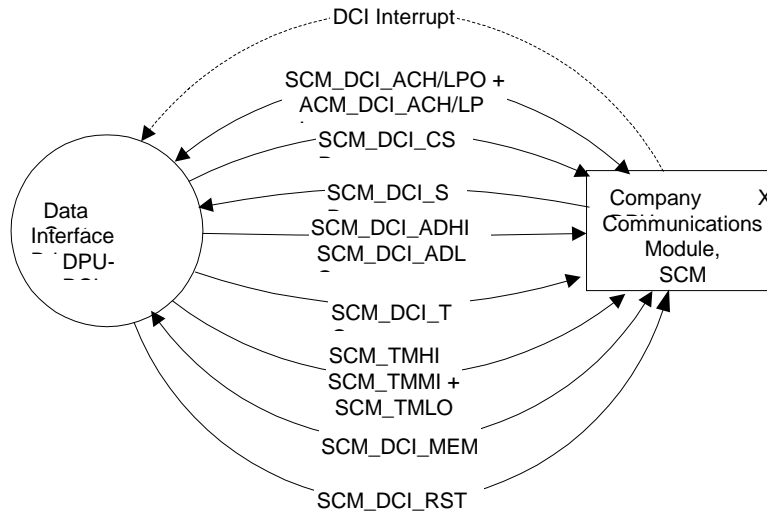


**Figure 12. Synchronous Serial Interface CSC Context Diagram**

The SSI Driver CSC is a custom device driver for the Company X INSTRUMENT A DPU. The requirements for the SSI Driver CSC are listed in Appendix A.

## 5.9 Data Capture Interface Driver CSC

A context diagram for the Data Capture Interface (DCI) Driver CSC is shown in the following figure.

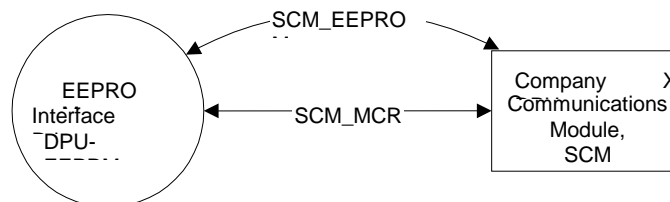


**Figure 13. Data Capture Interface Context Diagram**

The DCI Driver CSC is a custom device driver for the Company X INSTRUMENT A DPU. The requirements for the DCI Driver CSC are listed in Appendix A.

## 5.10 EEPROM Driver CSC

A context diagram for the EEPROM Driver CSC is shown in the following figure.

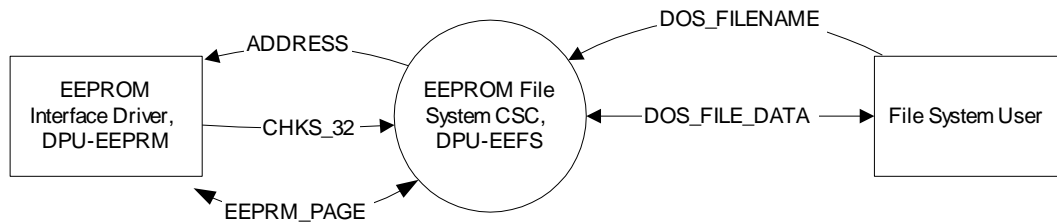


**Figure 14. EEPROM Interface Driver Context Diagram**

The EEPROM Driver is a heritage software component from the X program. The requirements for the EEPROM Driver are described in document 8089. Changed requirements are listed in Appendix A, and are limited to hardware interface differences between the X and the Company X DPU.

## 5.11 EEPROM Filesystem CSC

A context diagram for the EEPROM Filesystem (EEFS) CSC is shown in the following figure.

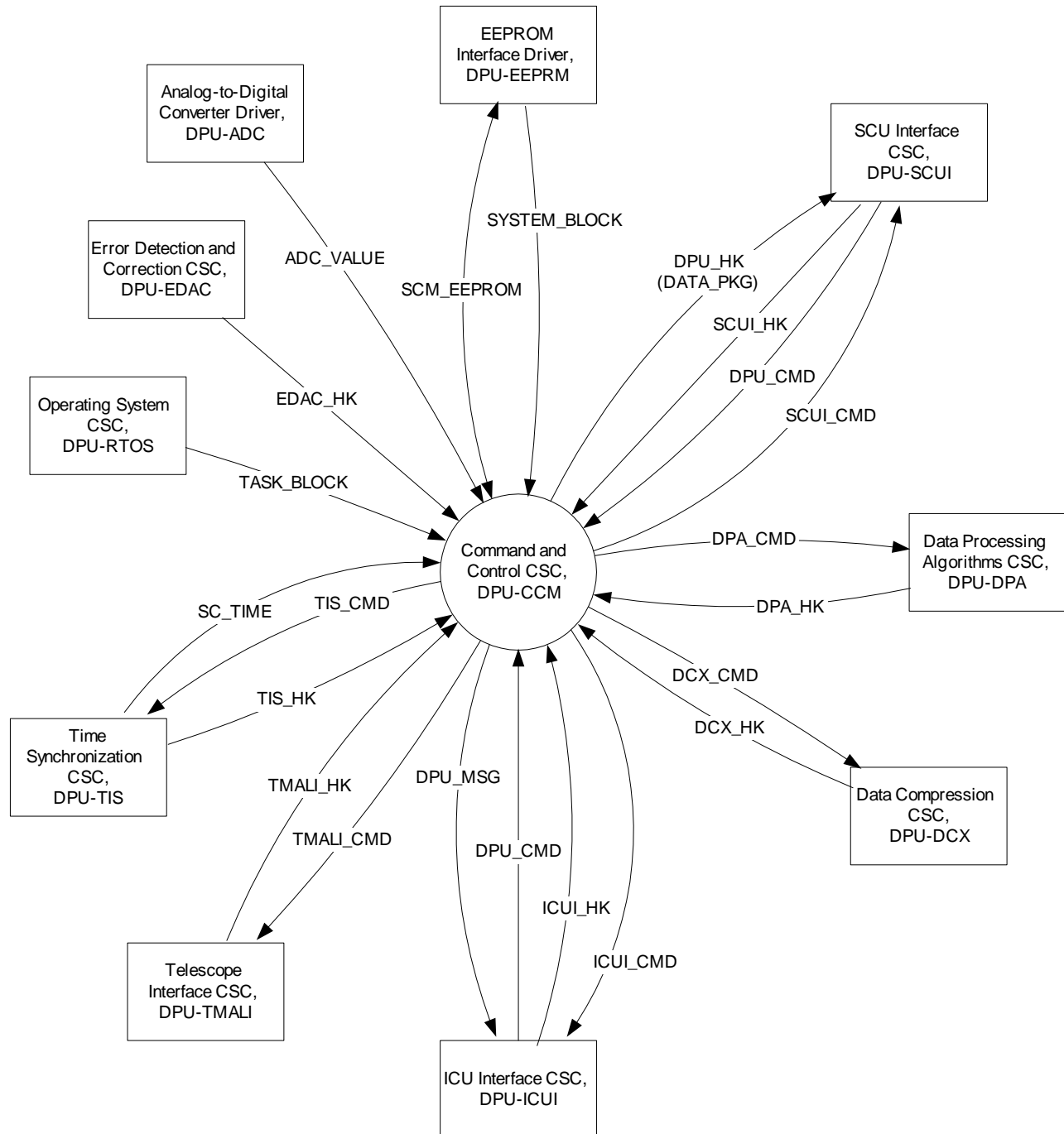


**Figure 15. EEPROM Filesystem CSC Context Diagram**

The EEFS CSC is a heritage software component from the X program. The requirements for the EEFS CSC are described in document 8089. There are no requirements changes applicable to the EEFS CSC.

## 5.12 Command and Control CSC

A context diagram for the DPU-CCM is shown in the following figure.



**Figure 16 – Command and Control CSC Context Diagram**

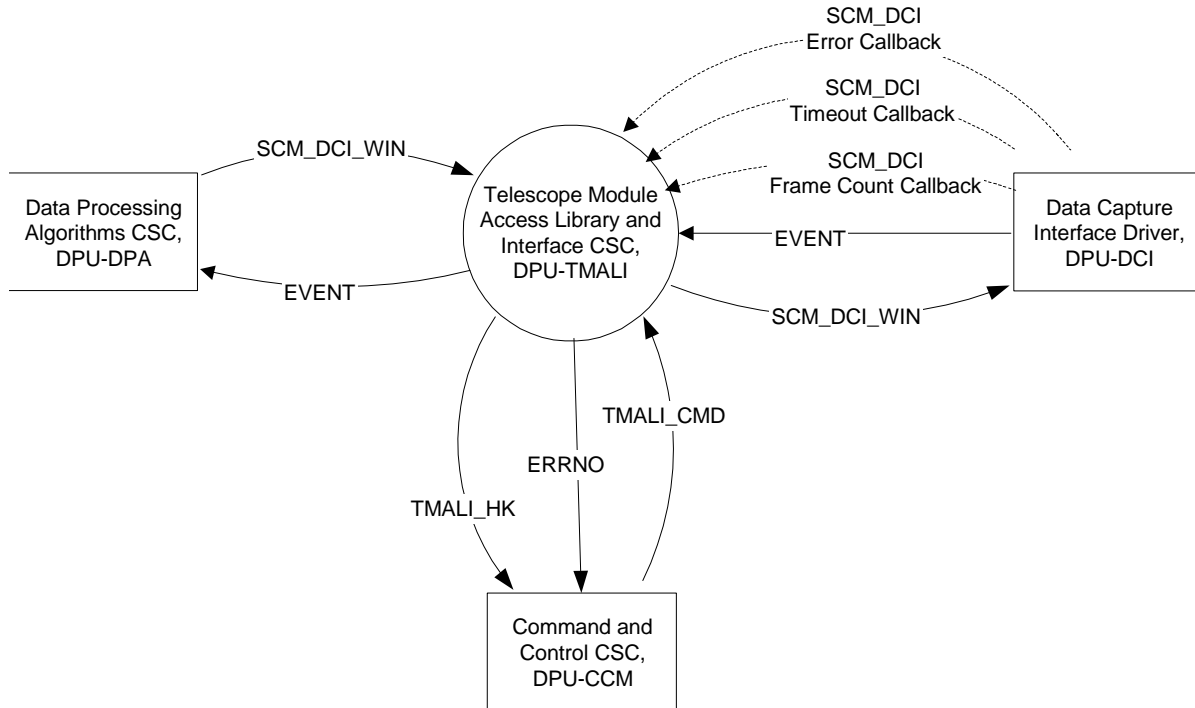
The following control and data flows are not shown in order to avoid cluttering the diagram:

- CSCs which have associated operating system tasks report a task heartbeat to the CCM CSC.
- Each application CSC has an **ERRNO** flow to the CCM CSC.

The requirements for the DPU-CCM CSC are listed in Appendix A.

### 5.13 Telescope Module Access Library and Interface CSC

A context diagram for the DPU-TMALI is shown in the following figure.

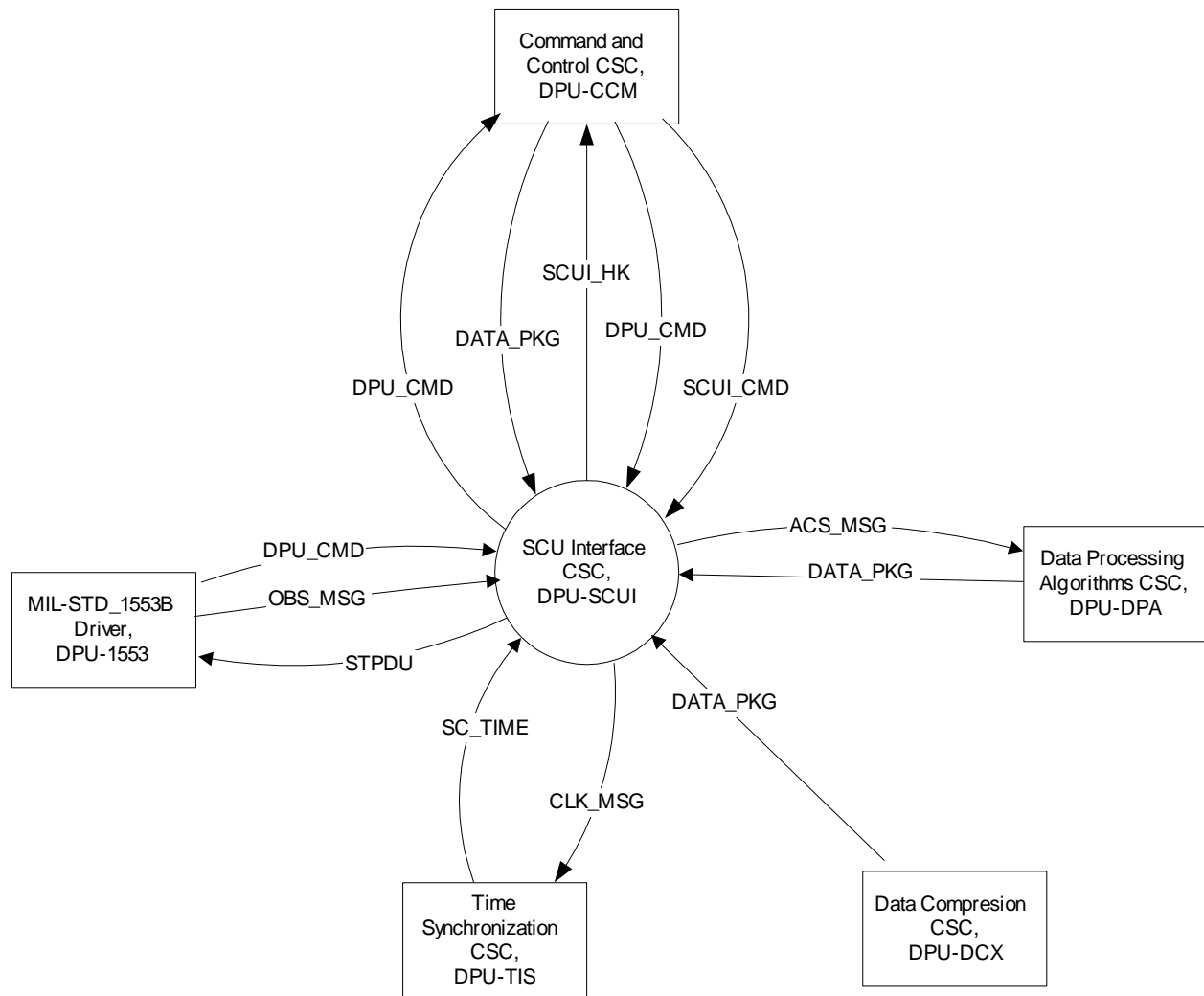


**Figure 17. Telescope Module Access Library and Interface CSC Context Diagram**

The requirements for the DPU-TMALI CSC are listed in Appendix A.

## 5.14 SCU Interface CSC

A context diagram for the DPU-SCUI is shown in the following figure.

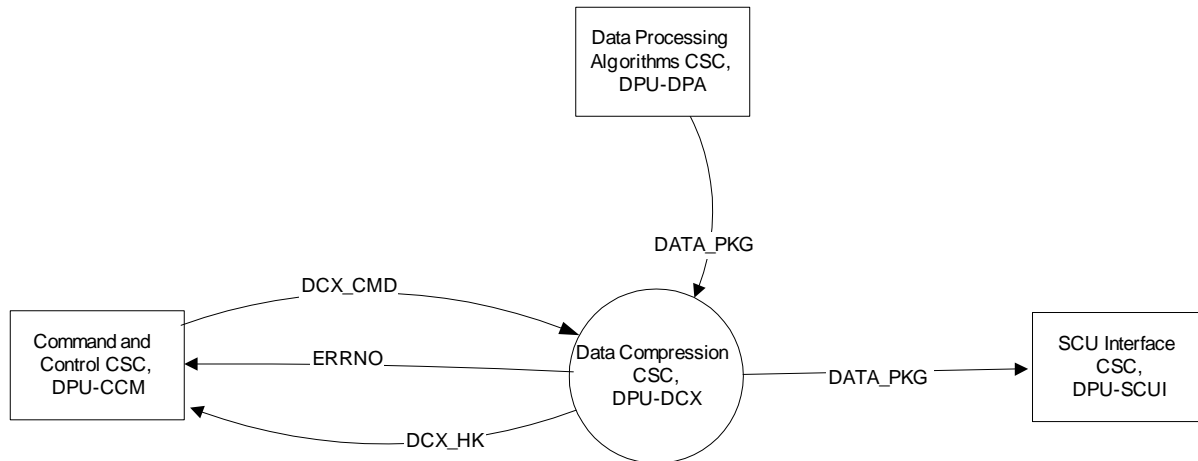


**Figure 18. SCU Interface CSC Context Diagram**

The requirements for the DPU-SCUI CSC are listed in Appendix A.

## 5.15 Data Compression CSC

A context diagram for the DPU-DCX is shown in the following figure.

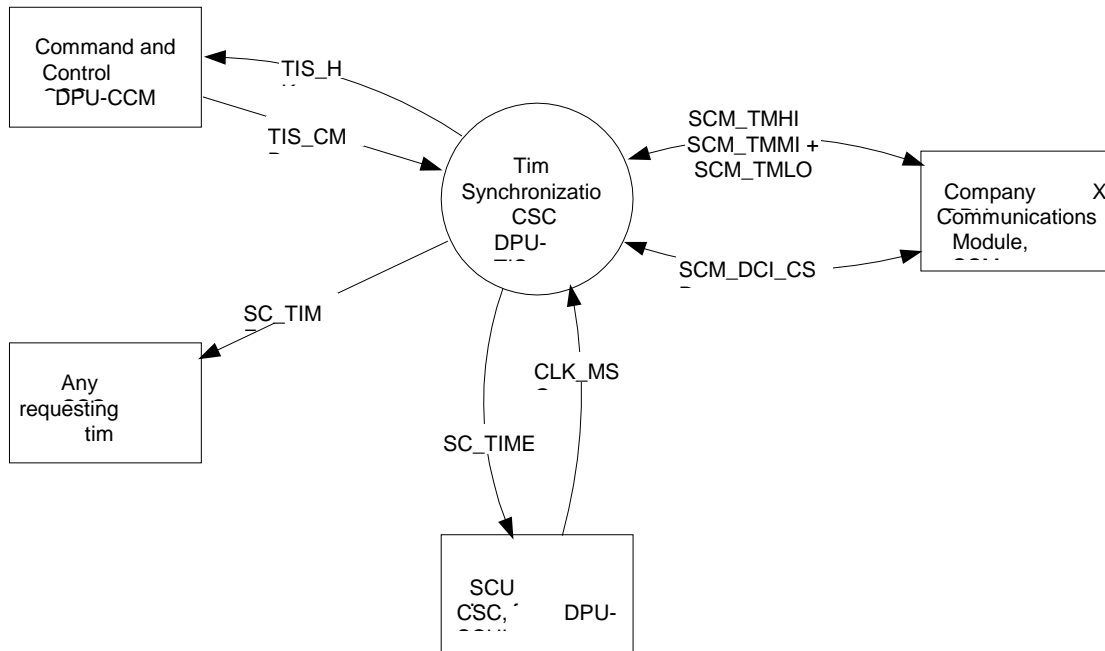


**Figure 19. Data Compression CSC Context Diagram**

The requirements for the DPU-DCX CSC are listed in Appendix A.

## 5.16 Time Synchronization CSC

A context diagram for the DPU-TIS is shown in the following figure.



**Figure 20. Time Synchronization CSC Context Diagram**

The requirements for the DPU-TIS CSC are listed in Appendix A.

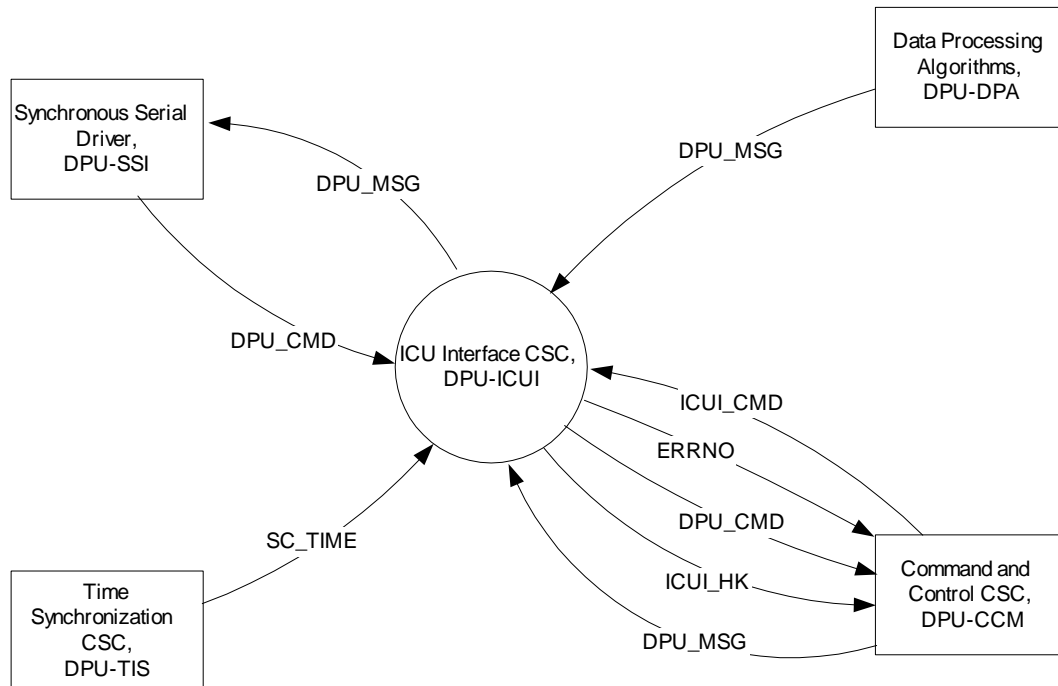
## 5.17 Reserved

A number of CSCs are being reused from the X FSW, whose requirements are described in document 8089. In order to maintain requirement number synchronization with the X SRS, this section is inserted as a reserved section.



## 5.18 ICU Interface CSC

A context diagram for the DPU-ICUI is shown in the following figure.

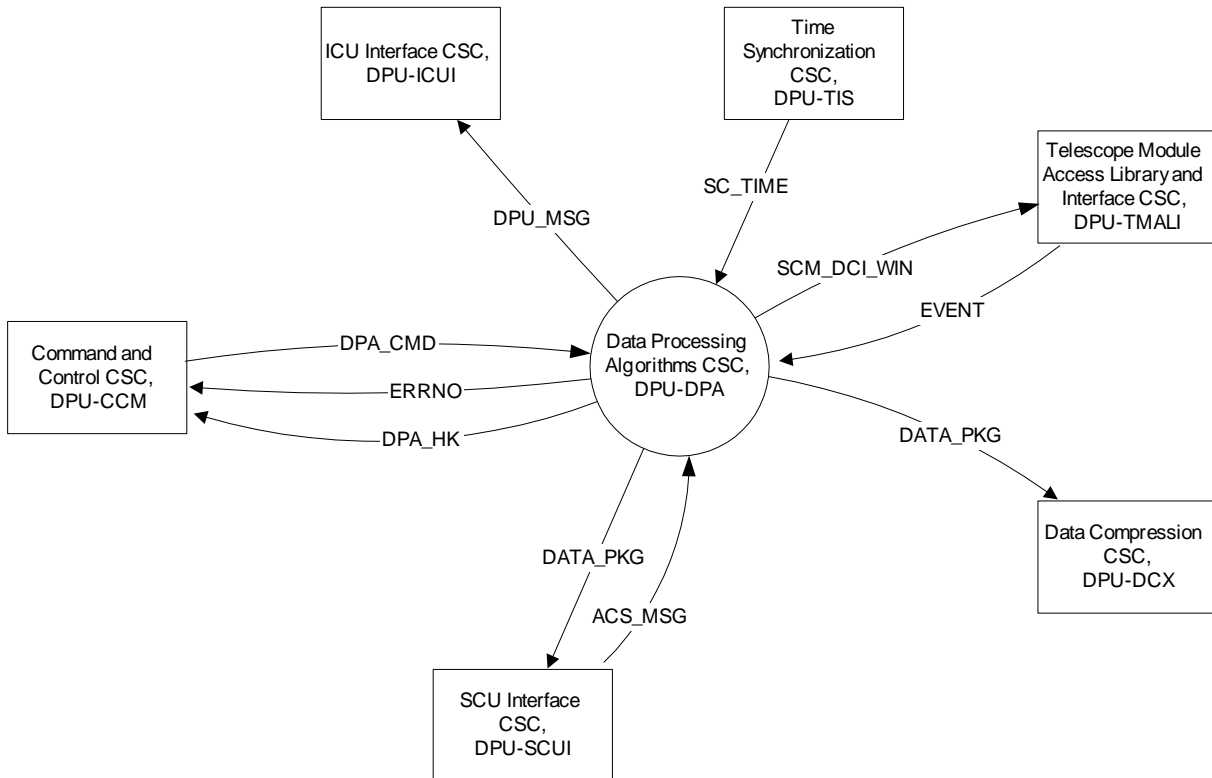


**Figure 21. ICU Interface CSC Context Diagram**

The requirements for the DPU-ICUI CSC are listed in Appendix A.

## 5.19 Data Processing Algorithms CSC

A context diagram for the DPU-DPA is shown in the following figure.



**Figure 22. Data Processing Algorithms CSC Context Diagram**

The requirements for the DPU-DPA CSC are listed in Appendix A.

## 6. DATA DICTIONARY

This section contains the data dictionary for the DPU Flight Software. In this dictionary, data elements are described either as types or as composites. Composite Data Elements are constructed from more elementary components. Most of the elements defined in this table will become data types or variables in the design, and so an effort is made to define the data items to a reasonable degree of accuracy and to maintain naming consistency with the eventual design. However in the context of this requirements document, the emphasis is on general data flow rather than on precise definitions, and so some changes in name and definition is anticipated at design time.

Table F. Data Dictionary		
Name	Attributes	Description
ACS_MSG	Composite: SC_TIME + Right Ascension + Declination + Roll	Attitude Control System message transmitted from the spacecraft to each instrument at 5Hz.  This definition includes only those components required within the context of its use here. For a complete definition of this message, refer to the applicable ICD.
ADC_VALUE	Type: UINT16	Value returned by the analog-to-digital converter. For Company X, will either be a thermistor reading or a voltage measurement.
ADDRESS	Type: Fundamental, 32-bit, char *	Address
APID	Type: UINT16 Least significant 11 bits denote the APID	Application ID
BC_INDEX	Type: UINT32 Location: scmBase32 + 0x6F0400 Value: 0x0: Boot Alternate 0x1-0xFFFFFFFF: Boot Primary	Boot Configuration Index
BIT_1553_INT	Type: BIT_RESULT Location: scmBase32 + 0x6F1084	MIL-STD-1553B Internal BIT results
BIT_1553_RAM	Type: BIT_RESULT Location: scmBase32 + 0x6F1080	MIL-STD-1553B Device RAM BIT results
BIT_CPU_BRANCH	Type: BIT_RESULT Location: scmBase32 + 0x6F1040	Branch Processor BIT result
BIT_CPU_FLTPT	Type: BIT_RESULT Location: scmBase32 + 0x6F1048	Floating Point Processor BIT result
BIT_CPU_FXPT	Type: BIT_RESULT Location: scmBase32 + 0x6F1044	Fixed Point Processor BIT result
BIT_CPU_INT	Type: BIT_RESULT Location: scmBase32 + 0x6F104C	Interrupt Processor BIT results
BIT_CPU_TIMER	Type: BIT_RESULT Location: scmBase32 + 0x6F1054	Timer Processor BIT results

Table F. Data Dictionary

Name	Attributes	Description
BIT_DATA	Composite: BIT_CPU_BRANCH + BIT_CPU_FXPT + BIT_CPU_FLTPT + BIT_CPU_INT + BIT_CPU_TIMER + BIT_EDAC_SBE + BIT_EDAC_MBE + BIT_PROM_CHKS + BIT_1553_RAM + BIT_1553_INT + BIT_DRAM  Location: scmBase32 + 0x6F1000 through scmBase32 + 0x6F11FC	Results of built-in test (stages 1 and 2).
BIT_DRAM	Composite: 16{BIT_RESULT}16  Location: scmBase32 + 0x6F1100 through scmBase32 + 0x6F113C	Results of DRAM built-in test.
BIT_EDAC_MBE	Type: BIT_RESULT Location: scmBase32 + 0x6F105C	Multiple-bit error EDAC BIT result
BIT_EDAC_SBE	Type: BIT_RESULT Location: scmBase32 + 0x6F1058	Single-bit error EDAC BIT result
BIT_HK	Composite: BIT_CPU_BRANCH + BIT_CPU_FXPT + BIT_CPU_FLTPT + BIT_CPU_INT + BIT_CPU_TIMER + BIT_EDAC_SBE + BIT_EDAC_MBE + BIT_PROM_CHKS + BIT_1553_RAM + BIT_1553_INT + BIT_DRAM	Built-In Tests Housekeeping Data
BIT_PROM_CHKS	Type: BIT_RESULT Location: scmBase32 + 0x6F1060	PROM checksum BIT result
BIT_RESULT	Type: UINT32  Value: 0: PASS -1: FAIL	Built-In Test Result
BOOT_CNT	Type: UINT32  Location: scmBase32 + 0x6F0404  Range: 0x00000000 – 0xFFFFFFFF	Count of the number of times the DPU has booted

Table F. Data Dictionary

Name	Attributes	Description
CCM_CMD	Composite: No Op command   Reboot DPU command   Command echo enable/disable command   Set housekeeping rate command   Resent startup packet command   Memory poke command   Memory load command   Memory dump command   Memory checksum command	CCM CSC Housekeeping Values
CCM_HK	Composite: Number of commands received + Number of commands executed + Last command executed + Last command rejected + Errnos + Command echo + Temperatures + Voltages + BC_INDEX + BOOT_CNT + RETRY_CNT + TASK_BLOCKS + TIS_SYNC_ENABLE status + DCX_SYNC_ENABLE status	CCM CSC Housekeeping Values
CHKS_16	Type: UINT16 Value: Modulo 65536 addition of each byte of the address range.	CCSDS Telecommand or Telemetry Packet Checksum, 16-bit
CHKS_32	Type: UINT32 Value: Sum of several UINT32 ignoring carry	PROM or EEPROM checksum, 32-bit
CLK_MSG	Composite: See SC_TIME	Clock message transmitted from the spacecraft to each instrument at 1Hz.  This definition includes only those components required within the context of its use here. For a complete definition of this message, refer to document 1143-EI-S19121.
COLD_MEM_SIZE	Type: MEM_SIZE Location: scmBase32 + 0x6F0010 Value: 0x08000000	Number of bytes of memory to test/clear following a cold boot
COLD_SKIP_BIT	Type: SKIP_BIT Location: scmBase32 + 0x6F0004 Value: 0x0	Indicates whether or not to perform Stage 1 Built-In Tests on cold boot
CPU_SPEED	Type: UINT32 Location: scmBase32 + 0x6F0034 Range: 1-4 Value: 1: 2.5 MHz 2: 5 MHz 3: 10 MHz 4: 20 MHz (default)	Processor Clock Speed

Table F. Data Dictionary

Name	Attributes	Description
DATA_PKG	Composite: APID + DATA_PKG_CTRL + <Data length> UINT16 + 1{UINT16}65536	Data and necessary ancillary information which will be converted into a CCSDS telemetry packet.
DATA_PKG_CTRL	Type: UINT8 Bits: 0x1: Data is compressed 0x2: Data is high priority 0x4: Data can be segmented 0x8: Compute checksum	Control bits to control the creation and routing of the CCSDS telemetry packet.
DCX_BUFFER_SIZE	Type: UINT32 Location: (see EEPROM memory map) Range: 0x100000 (1MB) – 0x6400000 (100MB)	Size of the data compression input queue
DCX_CMD	Type: Composite Purge compression queue command	Commands affecting the DCX CSC.
DCX_ENABLE	Type: Boolean	Indicates whether data compression is enabled or not.
DCX_HK	Composite: Bytes currently waiting on DCX queue + Bytes compressed + Average compression ratio	Data Compression Housekeeping All counting values reset when HK is reported.
DOS_FILE_DATA	Composite: 0{UINT8}64K	DOS File Data
DOS_FILENAME	Composite: [A-Z   a-z] + 0{[A-Z   a-z   0-9]}7 + 0{.}1 + 0{A-Z   a-z   0-9}3	DOS Filename
DPA_CMD	Composite: Mode command (see MODE_INFO)   XRT Position command   Stop Mode command   Abort Mode command   Enable/Disable Compression command	Commands affecting the DPA CSC.
DPA_HK	Composite: Number of events received + Number of Image Mode events + Number of Event Mode events + Number of bad events + Number of events lost + Number of Mode commands received + Current mode (see MODE_INFO) + Compression status (see DCX_ENABLE)	Data Processing Algorithms Housekeeping All counting values reset when HK is reported.
DPU_CMD	Composite: CCM_CMD   DCX_CMD   DPA_CMD   ICUI_CMD   SCUI_CMD   TMALI_CMD	A command to the DPU originating from the spacecraft, the ICU or the ground (via the spacecraft and/or the ICU).  Commands arrive at the DPU in the form of CCSDS telecommand packets. Refer to document 1143-EI-S19121 for the format of Company X telecommand packets.

Table F. Data Dictionary

Name	Attributes	Description
DPU_CNFG_PARMS	Composite: DPU_CTRL_STATUS + SCU_POLL_RATE + SCU_BUFFER_RATE + <lpq>SCU_BUFFER_SIZE + <hpq>SCU_BUFFER_SIZE + TMALI_PP_LIMIT + TMALI_TIMEOUT + TMALI_BUFFER_SIZE + DCX_BUFFER_SIZE + EDAC_MODULUS + EDAC_DELAY + DPA_PARM[6]	DPU Configuration Parameters  Contains DPU startup and dynamic configuration parameters. A copy of the DPU_CNFG_PARMS is contained in EEPROM for bootup defaults, and in DRAM for dynamic configuration..
DPU_CTRL_STATUS	Type: Fundamental, UINT32  Bits:  0x00000008: TIS_SYNC_ENABLE 0x00000010: DCX_ENABLE 0x00000100: TIS_ALTERNATE_1PPS  Values:  0x0: Disabled 0x1: Enabled	Status Indicator for various control options.  Data type used in the DPU_CNFG_PARMS table to indicate whether a particular capability is enabled or disabled. Data type is 32 bits since this data is recorded in EEPROM.
DPU_HK	Composite:  BIT_HK + CCM_HK + DCX_HK + DPA_HK + EDAC_HK + ICUI_HK + SCUI_HK + TIS_HK + TMALI_HK	DPU Housekeeping
DPU_HK_START	Composite:  BIT_HK + MET at Last Boot + SYSTEM_BLOCK + CPU_SPEED (queried) + SYSTEM_CONFIG_AREA Cksum (calculated)+ BC0 Checksum (calculated) + BC0 Checksum (calculated) + FSW CSC Versions	DPU Startup Packet
DPU_MSG	Composite:  Heartbeat Message   Mode Ready Message   Mode Complete Message   Channel Boundaries Message   DPU Boot Complete Message   Upload Start Message   Upload End Message	Message from the DPU to the ICU.  Refer to document DPUICD-01 for additional detail.
DRAM	Composite: {UINT32}  Locations: 0x0 through 0x08000000	DRAM on the RAD6000.

Table F. Data Dictionary

Name	Attributes	Description
EDAC_HK	Composite: EdacRscSbeCnt + EdacRscSbeAdrs + EdacRscSbePrev + EdacRscMbeCnt + EdacRscMbeAdrs + EdacRscMbePrev + EdacScmSbeCnt + EdacScmSbeAdrs + EdacScmSbePrev + EdacScmMbeCnt + EdacScmMbeAdrs + EdacScmMbePrev	Error Detection and Correction Data
EdacRscMbeAdrs	Type: ADDRESS	RSC Multiple-Bit Error Last Occurrence
EdacRscMbeCnt	Type: UINT32	RSC Multiple-Bit Error Count
EdacRscMbePrev	Type: ADDRESS	RSC Multiple-Bit Error Next-to-last Occurrence
EdacRscSbeAdrs	Type: ADDRESS	RSC Single-Bit Error Last Occurrence
EdacRscSbeCnt	Type: UINT32	RSC Single-Bit Error Count
EdacRscSbePrev	Type: ADDRESS	RSC Single-Bit Error Next-to-last Occurrence
EdacScmMbeAdrs	Type: ADDRESS	SCM Multiple-Bit Error Last Occurrence
EdacScmMbeCnt	Type: UINT32	SCM Multiple-Bit Error Count
EdacScmMbePrev	Type: ADDRESS	SCM Multiple-Bit Error Next-to-last Occurrence
EdacScmSbeAdrs	Type: ADDRESS	SCM Single-Bit Error Last Occurrence
EdacScmSbeCnt	Type: UINT32	SCM Single-Bit Error Count
EdacScmSbePrev	Type: ADDRESS	SCM Single-Bit Error Next-to-last Occurrence
EEPRM_PAGE	Composite: 128{UINT32}128	EEPROM Page
EICR_EIM0	Type: UINT16 Location: eicrBase32 + 0x000000 Bits: 31: RBI Timer Interrupt 30: Reserved 29: UART Interrupts 28-24: Reserved 23: VME IRQ 7 22: VME IRQ 6 21: VME IRQ 5 20: VME IRQ 4 19: VME IRQ 3 18: Reserved 17: VME IRQ 2 16: VME IRQ 1 15-0: Reserved	External Interrupt Mask Register
EICR_MEAR	Type: UINT16 Location: eicrBase32 + 0x00001C Bits: 31-24: Syndrome 23-0: Bits 3-26 or Real Address	Machine Check Error Address Register



Table F. Data Dictionary

Name	Attributes	Description
EICR_MESR	Type: UINT16 Location: eicrBase32 + 0x000018 Bits: 31: Error occurred in Diagnostic Mode 30: Error occurred on a processor load or store 29: Reserved 28: Address Exception 27: Attempted store into a Read-Only Segment 26: Uncorrectable ECC Error 25-0: Reserved	Machine Check Error Status Register
EICR_SBAR	Type: UINT16 Location: eicrBase32 + 0x00002C Bits: 31-24: Syndrome 23-0: Bits 3-26 or Real Address	Single-Bit Error Address Register
EICR_SBSR	Type: UINT16 Location: eicrBase32 + 0x000028 Bits: 31: Single-Bit ECC Error 30-0: Reserved	Single-Bit Error Status Register
EicrBase32	Type: ADDRESS Value: 0xD0000000	Base address for External Interrupt Control Registers on the RAD6000 CPU Module
EICRS	Composite: EICR_MEAR + EICR_MESR + EICR_SBAR + EICR_SBSR + EICR_EIM0	External Interrupt Control Registers
ENET_HOST_IP	Type: UINT32 Location: scmBase32 + 0x6F01F8	32-Bit Integer representation of Host IP address Each element of the quad-address is encoded in each of the four bytes of this value.
ENET_MAC	Composite: 2{UINT32}2 Location: scmBase32 + 0x6F0014 through scmBase32 + 0x6F0018	Ethernet Media Access Control Address Only first six bytes are valid.
ENET_TARG_IP	Type: UINT32 Location: scmBase32 + 0x6F01F4	32-Bit Integer representation of DPU IP address Each element of the quad-address is encoded in each of the four bytes of this value.
ERRNO	Type: UINT32 Bits: 31-16: Module Number 15-8: Error Number 7-0: Error Supplemental Data	Error Number
EVENT	Composite: EVENT_TYPE + EVENT_DATA	Detector event.

**Table F. Data Dictionary**

<b>Name</b>	<b>Attributes</b>	<b>Description</b>
EVENT_DATA	Type: UINT24 Detector Event: If event is a detector event, refer to document XMM-OM/MSSL/0008.05 for a list of possible event types, and to document 036911400 for alternate bit arrangement for Science Mode 3. Timestamp: If event is a timestamp, then the complete timestamp will be provided in two consecutive events – refer to document 036911400.	Bits within the least-significant 24-bits of an EVENT which contains a 24-bit event, or a partial timestamp.
EVENT_ERR	Type: UINT32 Range: 0x00000000 – 0xFFFFFFFF	Number of bad events
EVENT_NUM	Type: UINT32 Range: 0x00000000 – 0xFFFFFFFF	Number of events
EVENT_TYPE	Type: UINT8 Bits: 0x80 – Type (0=timestamp, 1=event) 0x40 – Event error (more than 24 bits) 0x20 – Event error (less than 24 bits) 0x10 – Parity error 0x08 – Unused 0x04 – Unused 0x02 – Unused 0x01 – If timestamp, 0=upper timestamp, 1=lower timestamp	Bits within the most-significant nibble of an EVENT which determine how to interpret the EVENT_DATA. Refer to document 11400.
EXP_TIME	Type: UINT32 Range: 0x00000000 – 0xFFFFFFFF	The actual exposure time over which the DPU-DPA has processed events. This may differ from the commanded exposure length in MODE_INFO
FC_STAR	Type: Composite UINT16 X_POS + UINT16 Y_POS + UINT16 Intensity	Star description
FC_STAR_NUM	Type: UINT8	Number of stars
FILE_SYSTEM_BLOCK	Composite: {UINT32} Location: scmBase32 + 0x500000 through scmBase32 + 0x7EFFFF	File System Block
ICUI_CMD	Composite: <None at this time>	Commands affecting the ICUI CSC.
ICUI_HK	Composite: Commands received + Commands rejected + Messages sent + SCM_SSI_CSR	ICU Interface Housekeeping

Table F. Data Dictionary

Name	Attributes	Description
IMAGE_ROW	Composite: UINT16 Target ID + UINT32 Sequence ID + UINT32 Exposure Length + UINT16 X_IMG_POS + UINT16 Y_IMG_POS + UINT16 X_IMG_WIDTH + UINT16 Y_IMG_HEIGHT + UINT16 Scan Line Number in image+ UINT16 #n bytes in Scan Line+ Scan Line (n 16-bit pixels)	Header + horizontal scan line
IMAGE_DATA	Composite : # of IMAGE_ROW's IMAGE_ROW+ IMAGE_ROW+ IMAGE_ROW+ ...	Image in PROD_IMAGE Image is broken up into horizontal scan lines
INT16	Type: Fundamental, 16-bit signed integer	Short Integer
INT32	Type: Fundamental, 32-bit signed integer	Integer
IOCC_EOI_IRQ1	Type: UINT32 Location: ioccBase32 + 0x41008C Bits: 31-0: Reserved	End of Interrupt Register (IRQ1)
IOCC_EOI_IRQ2	Type: UINT32 Location: ioccBase32 + 0x42008C Bits: 31-0: Reserved	End of Interrupt Register (IRQ2)
IOCC_EOI_IRQ3	Type: UINT32 Location: ioccBase32 + 0x43008C Bits: 31-0: Reserved	End of Interrupt Register (IRQ3)
IOCC_EOI_IRQ4	Type: UINT32 Location: ioccBase32 + 0x44008C Bits: 31-0: Reserved	End of Interrupt Register (IRQ4)
IOCC_EOI_IRQ5	Type: UINT32 Location: ioccBase32 + 0x45008C Bits: 31-0: Reserved	End of Interrupt Register (IRQ5)
IOCC_EOI_IRQ6	Type: UINT32 Location: ioccBase32 + 0x46008C Bits: 31-0: Reserved	End of Interrupt Register (IRQ6)
IOCC_EOI_IRQ7	Type: UINT32 Location: ioccBase32 + 0x47008C Bits: 31-0: Reserved	End of Interrupt Register (IRQ7)

Table F. Data Dictionary

Name	Attributes	Description
IOCC_EOI_REGS	Composite: IOCC_EOI_SYSFAIL + IOCC_EOI_IRQ1 + IOCC_EOI_IRQ2 + IOCC_EOI_IRQ3 + IOCC_EOI_IRQ4 + IOCC_EOI_IRQ5 + IOCC_EOI_IRQ6 + IOCC_EOI_IRQ7	End of Interrupt Registers
IOCC_EOI_SYSFAIL	Type: UINT32 Location: ioccBase32 + 0x40008C Bits: 31-0: Reserved	End of Interrupt Register (SYSFAIL)
IOCC_RBI_CFG	Type: UINT32 Location: ioccBase32 + 0x400010 Bits: 31: Master Enable 30: Reserved 29-28: Turbo clock 27-26: VME AML 25-24: VME LIM 23: Reserved 22-20: TCW Table Size 19: Reserved 18: Bus Hold/ 3P DMA K bit 17: PIO/3P Select 16: Clock Control 15-0: Reserved	RBI Configuration Register
IOCC_RBI_TIMER	Type: UINT32 Location: ioccBase32 + 0x480004 Bits: 31-16: Interrupt Interval (two's complement of interval) 15-0: Reserved Units: 2.17 $\mu$ secs	RBI Time Control Register, Real Time Incrementer (Interval Timer)
IOCC_REGS	Composite: IOCC_IRQ_REGS + IOCC_RBI_TIMER + IOCC_INT_REG + IOCC_EOI_REGS	Input/Output Channel Controller Registers
IOCC_SYSFAIL	Type: UINT32 Location: ioccBase32 + 0x400004 Bits: 31-0: Reserved	VME SYSFAIL Interrupt Register, Store Generate, Load Acknowledge
ioccBase32	Type: ADDRESS Value: 0xE0000000	Base address for the Input/Output Channel Controller on the RAD6000 CPU Module

Table F. Data Dictionary

Name	Attributes	Description
LAST_BOOT_IVEC	Composite: Last Exception Task ID + Last Exception Vector Number + Last Exception Stack Pointer + Last Exception Vector Offset + Last Exception Errno + Last Exception Data Access Register + Last Exception Data Storage Interrupt Reg. + Last Exception Floating Point Status Register + Last Exception External Interrupt Mask Reg 0 + Last Exception External Interrupt Mask Reg 1 + Last Exception General Purpose Regs [32] + Last Exception Machine State Register + Last Exception Link Register + Last Exception Count Register + Last Exception Program Counter + Last Exception Condition Register + Last Exception Fixed Point Exception Register + Last Exception MQ Register	State of interrupt vector at the point of the most recent warm boot (if caused by exception).
LOCATION_BLOCK	Composite: Beginning Address in EEPROM + Ending Address in EEPROM + Copy Address in DRAM + Execute Address in DRAM + Checksum	Data used by the Bootstrap to copy and execute a boot configuration.  Refer to the EEPROM Memory Map.
MEM_SIZE	Type: UINT32 Range: 0x00800000-0x08000000  Value (samples): 0x00800000: 8 MB 0x01000000: 16 MB 0x02000000: 32 MB 0x04000000: 64 MB 0x08000000: 128 MB	Size in bytes of DRAM to clear/test
MODE_INFO	Composite: UINT8 Mode ID + UINT8 Detector Electronics Output Format + UINT8 Target Type + UINT16 Target ID + UINT32 Sequence ID + UINT32 Exposure Length + UINT16 X_IMG_POS + UINT16 Y_IMG_POS + UINT16 X_IMG_WIDTH + UINT16 Y_IMG_HEIGHT + UINT16 X_EVENT_POS + UINT16 Y_EVENT_POS + UINT16 X_EVENT_WIDTH + UINT16 Y_EVENT_HEIGHT + UINT16 Tracking Frame Time + UINT16 Criterion Mask + UINT8 Number of Guide Stars	Mode command parameters

Table F. Data Dictionary

Name	Attributes	Description
OBS_MSG	Composite: ACS_MSG + CLK_MSG	Messages transmitted to the DPU which originate in the spacecraft or in another instrument.  This does not represent the complete list of messages which may be transmitted; rather only those that the DPU will utilize. Refer to the applicable Company X program-level ICD (which did not exist at the time this document was created) for a complete list of these messages and their formats.
PROD_ACS_MSG	Composite: <# Msgs> UINT32 + <Start> SC_TIME + <End> SC_TIME + 1{ACS_MSG}N (max messages in pkt)	ACS Message Data Product
PROD_CENTROID_CONF	Type: Composite MODE_INFO + EXP_TIME + PSUEDO_IMAGE_NUM + PSUEDO_IMAGE	Centroiding Confirmation Engineering mode data product
PROD_CHAN_BOUND	Type: Composite MODE_INFO + EXP_TIME + X_BOUND + Y_BOUND	Channel Boundaries Engineering Mode Data Product
PROD_EVENT	Composite: MODE_INFO + EXP_TIME + EVENT_NUM + EVENT_ERR + 1{EVENT}N	DPU-DPA Event Mode Data Product
PROD_FINDING_CHART	Type: Composite MODE_INFO + EXP_TIME + FC_STAR_NUM + FC_STAR [FC_STAR_NUM]	Parameterized finding chart
PROD_IMAGE	Type: Composite: MODE_INFO + EXP_TIME + EVENT_NUM + EVENT_ERR + CCD_FRAME_NUM + TRK_FRAME_NUM + IMAGE_DATA	DPU-DPA Image Mode Data Product
PROD_INTENSE_CHAR	Type: Composite MODE_INFO + EXP_TIME + INTENSITY_VECTOR	Intensifier Characteristics Engineering mode data product
PROD_MN_DATA	Type: Composite MODE_INFO + EXP_TIME + EVENT_NUM + MN_DATA	M, N Data Product

Table F. Data Dictionary

Name	Attributes	Description
PROD_TRACKING_SWITCH	Type: Composite SC_TIME + TRK_FRAME_NO	Generated within Image mode when the DPU-DPA switches between tracking frame buffers
PROD_TRACKING_REC	Type: Composite MODE_INFO + EXP_TIME + TRK_FRAME_NO + TRK_DELTA_X + TRK_DELTA_Y + TRK_DELTA_THETA	Product resulting from the Shift and Add calculation within the DPU-DPA
RETRY_CNT	Type: UINT32 Location: scmBase32 + 0x6F0408	Incremented during a warm boot.
RTCL	Type: UINT32 Range: 0x0 – 0xFFFF8 Units: 16.9 nanoseconds	Real Time Clock Low register on the RAD6000. The lower 7 bits of RTCL are unimplemented so actually increments at 2.17 microseconds.
RTCU	Type: UINT32 Range: 0x0 – 0xFFFFFFFF Units: 16.9 seconds	Real Time Clock Upper register on the RAD6000.
SC_TIME	Composite: <S/C Clock> SC_TIME_COARSE + <S/C Clock> SC_TIME_FINE + <UTC Delta> SC_TIME_COARSE + <UTC Delta> SC_TIME_FINE	Spacecraft Mission Elapsed Time and UTC Delta.
SC_TIME_COARSE	Type: UINT32 Units: 1 second	Spacecraft Coarse Time
SC_TIME_FINE	Type: UINT16 Range: 0 – 49999 Units: 20 us	Spacecraft Fine Time
SCM_1553	Composite: SCM_1553_BIT + SCM_1553_BLK + SCM_1553_CTRL + SCM_1553_CWD + SCM_1553_ILL + SCM_1553_ILR + SCM_1553_IMR + SCM_1553_INIT + SCM_1553_IPR + SCM_1553_RTBITS + SCM_1553_STS + SCM_1553_TMR	SCM 1553 hardware interface components sans the shared memory.
SCM_1553_BIT	Type: UINT16 Location: scmBase16 + 0x00400C Bits: 15: DMA Fail 14: Wrap Fail 13: Terminal Address Parity Fail 12: BIT Fail 11: Channel A Fail 10: Channel B Fail 9-0: User-Defined Bits	Summit Interrupt BIT Word Register

Table F. Data Dictionary

Name	Attributes	Description
SCM_1553_BLK	Type: UINT16 Location: scmBase16 + 0x004010 Bits: 15-0: Remote Terminal Descriptor Address Bits	Summit Remote Terminal Descriptor Pointer Register
SCM_1553_CTRL	Type: UINT16 Location: scmBase16 + 0x004000 Bits: 15: Start Execution 14: Start BIT 13: Start Software Reset 12: Channel A Enable 11: Channel B Enable 10: External Timer Clock Enable 9-7: Reserved 6: Buffer Mode Enable 5: Reserved 4: Broadcast Enable 3: Dynamic Bus Control Acceptance 2: Ping-Pong Enable 1: Interrupt Log Enable 0: Transmit Last Status Word	Summit Control Register
SCM_1553_CWD	Type: UINT16 Location: scmBase16 + 0x004004 Bits: 15-0: Current Command Bits	Summit Current Command Register
SCM_1553_ILL	Type: UINT16 Location: scmBase16 + 0x004020 Bits: Refer to UTMC-SUMMIT	Summit Illegalization Register
SCM_1553_ILR	Type: UINT16 Location: scmBase16 + 0x00400A Bits: 15-0: Interrupt Log List Pointer Bits	Summit Interrupt Log List Pointer Register
SCM_1553_IMR	Type: UINT16 Location: scmBase16 + 0x004006 Bits: 15: DMA Fail Interrupt 14: Wrap Fail Interrupt 13: Terminal Address Parity Fail Interrupt 12: BIT Fail Interrupt 11: Message Error Interrupt 10: Subaddress Accessed Interrupt 9: Broadcast Command Received Interrupt 8: Index Equal Zero Interrupt 7: Illegal Command Interrupt 6-0: Reserved	Summit Interrupt Mask Register
SCM_1553_INIT	Type: UINT16 Location: scmBase16 + 0x004014 Bits: Refer to UTMC-SUMMIT	Summit Initialization Block Register



Table F. Data Dictionary

Name	Attributes	Description
SCM_1553_IPR	Type: UINT16 Location: scmBase16 + 0x004008 Bits: 15: DMA Fail Interrupt 14: Wrap Fail Interrupt 13: Terminal Address Parity Fail Interrupt 12: BIT Fail Interrupt 11: Message Error Interrupt 10: Subaddress Accessed Interrupt 9: Broadcast Command Received Interrupt 8: Index Equal Zero Interrupt 7: Illegal Command Interrupt 6-0: Reserved	Summit Interrupt Pending Register
SCM_1553_MEM	Type: {UINT16} Location: scmBase16 + 0x30000 through scmBase16 + 0x7FFFF	1553 Interface Shared Memory
SCM_1553_RTBITS	Type: UINT16 Location: scmBase16 + 0x004012 Bits: 15: Immediate Clear Function 14-10: Reserved 9: Instrumentation Bit 8: Service Request Bit 7-4: Reserved 3: Busy 2: Subsystem Flag Bit 1: Reserved 0: Terminal Flag	Summit Status Word Bits Register
SCM_1553_STS	Type: UINT16 Location: scmBase16 + 0x004002 Bits: 15: Terminal Address Bit 4 14: Terminal Address Bit 3 13: Terminal Address Bit 2 12: Terminal Address Bit 1 11: Terminal Address Bit 0 10: Terminal Address Parity 9: Mode Select 1 8: Mode Select 0 7: Military Standard A or B 6: LOCK Pin 5: AUTOEN Pin 4: SSYSF Pin 3: Summit MCM-C Executing 2: Terminal Parity Fail 1: READY Pin 0: TERACTION	Summit Operational Status Register
SCM_1553_TMR	Type: UINT16 Location: scmBase16 + 0x00400E Bits: 15-0: Time-tag Counter Bits	Summit Time-Tag Register

Table F. Data Dictionary

Name	Attributes	Description
SCM_ADC	Composite: SCM_ADC_MUX + SCM_ADC_DAT	Analog to Digital Converter hardware interface registers.
SCM_ADC_DAT	Type: UINT16 Location: scmBase16 + 0x00C00A Bits: 15-12: Reserved 11-0: A/D result	Analog to Digital Converter Interface start-conversion and data readout register. A write to the register with any value starts the A/D conversion. A read from the register returns the result of the last A/D conversion.
SCM_ADC_MUX	Type: UINT16 Location: scmBase16 + 0x00C008 Bits: 15: A/D Data Ready 14-10: Reserved 9-8: Mux bank selector 7-4: Thermistor mux selector 3-0: Voltage mux selector	Analog to Digital Converter Interface setup register. Refer to document 036911400 for additional detail.
SCM_DCI	Composite: SCM_DCI_ACHPI + SCM_DCI_ACHPO + SCM_DCI_ACLPI + SCM_DCI_ACLPO + SCM_DCI_FRMLIM + SCM_DCI_CSR + SCM_DCI_RSTD + SCM_DCI_SR + SCM_DCI_TMHI + SCM_DCI_TMLO + SCM_DCI_TMMI + SCM_DCI_TO + SCM_DCI_WIN	SCM data capture hardware interface components sans the shared memory.
SCM_DCI_ACHPI	Type: UINT16 Location: scmBase16 + 0x00A00E Bits: 15-4: Reserved 3-0: Address Counter High Bits	SCM DCI Address Counter High Register for Ping Buffer
SCM_DCI_ACHPO	Type: UINT16 Location: scmBase16 + 0x00A012 Bits: 15-4: Reserved 3-0: Address Counter High Bits	SCM DCI Address Counter High Register for Pong Buffer
SCM_DCI_ACLPI	Type: UINT16 Location: scmBase16 + 0x00A010 Bits: 15-0: Address Counter Low Bits	SCM DCI Address Counter Low Register for Ping Buffer
SCM_DCI_ACLPO	Type: UINT16 Location: scmBase16 + 0x00A014 Bits: 15-0: Address Counter Low Bits	SCM DCI Address Counter Low Register for Pong Buffer

Table F. Data Dictionary

Name	Attributes	Description
SCM_DCI_CSR	Type: UINT16 Location: scmBase16 + 0x00A000 Bits: 15: RESET_ARM 14: INTADR 13: INTTO 12: INTERR 11: Reserved 10: WIN_EN 9: ONE_PPS 8: BIT_ARR 7: TIMEJAM_EN 6: INTADR_EN 5: INTTO_EN 4: INTERR_EN 3: PING_RD 2: ACQ_EN 1: FORCESWAP 0: ARMSWAP	SCM DCI Control/Status Register
SCM_DCI_FRMLIM	Type: UINT16 Location: scmBase16 + 0x00A00C Bits: 15-0: Frame Limit Value – Integer Number of Frames	SCM DCI Frame Limit Register
SCM_DCI_MEM	Type: {UINT32} Location: scmBase32 + 0x800000 through scmBase32 + 0x9FFFFFFF	Data Capture Interface Shared Memory
SCM_DCI_RSTD	Type: UINT16 Location: scmBase16 + 0x00A016 Bits: 15-0: Reserved	SCM DCI Reset Register A write to this register, regardless of contents, resets the DCI controller if the RESET_ARM bit is set in SCM_DCI_CSR.
SCM_DCI_SR	Type: UINT16 Location: scmBase16 + 0x00A002 Bits: (Same as SCM_DCI_CSR)	SCM DCI Status Register
SCM_DCI_TO	Type: UINT16 Location: scmBase16 + 0x00A00C Bits: 15-0: Timeout Value in milliseconds	SCM DCI Timeout Register
SCM_DCI_WIN	Composite: SCM_DCI_WINX + SCM_DCI_WINY	SCM DCI event filtering window parameters
SCM_DCI_WINX	Type: UINT16 Location: scmBase16 + 0x00A018 Bits: 15-8: Maximum X range 7-0: Minimum X range	SCM DCI event filtering window parameter for the X axis.

Table F. Data Dictionary

Name	Attributes	Description
SCM_DCI_WINY	Type: UINT16 Location: scmBase16 + 0x00A01A Bits: 15-8: Maximum Y range 7-0: Minimum Y range	SCM DCI event filtering window parameter for the Y axis.
SCM_EEPROM	Type: {UINT32} Locations: scmBase32 + 0x400000 through scmBase32 + 0x6FFFFC	EEPROM on the SCM. Refer to Appendix B for memory map.
SCM_ICR	Type: UINT16 Locations: IRQ1: scmBase16 + 0x080016 Ethernet IRQ2: scmBase16 + 0x080014 Reserved IRQ3: scmBase16 + 0x080012 Reserved IRQ4: scmBase16 + 0x080010 Reserved IRQ5: scmBase16 + 0x08000E MIL-STD-1553 IRQ6: scmBase16 + 0x08000C Reserved IRQ7: scmBase16 + 0x08000A Memory Error Bits: 15-0: Reserved	SCM Interrupt Clear Registers Write clears indicated SCM-generated IRQ
SCM_IMR	Type: UINT16 Locations: scmBase16 + 0x080004 Bits: 15-8: Reserved 7: Memory Error Interrupt 6: Reserved 5: MIL-STD-1553B Interrupt 4-0: Reserved	SCM Interrupt Mask Register
SCM_IPR	Type: UINT16 Location: scmBase16 + 0x080006 Bits: 15-8: Reserved 7: Memory Error Interrupt 6: Reserved 5: MIL-STD-1553B Interrupt 4-0: Reserved	SCM Interrupt Pending Register
SCM_IVR	Type: UINT16 Locations: IVR1: scmBase16 + 0x080016 Ethernet IVR2: scmBase16 + 0x080018 Reserved IVR3: scmBase16 + 0x08001A Reserved IVR4: scmBase16 + 0x08001C Reserved IVR5: scmBase16 + 0x080020 MIL-STD-1553 IVR6: scmBase16 + 0x080022 Reserved IVR7: scmBase16 + 0x080024 Memory Error Bits: 15-8: Reserved 7-0: Status/ID	SCM Interrupt Vector Registers

Table F. Data Dictionary

Name	Attributes	Description
SCM_MCR	Type: UINT16 Location: scmBase16 + 0x080002 Bits: 15-8: Reserved 7: EEPROM Power Disable 6: EEPROM Write-Protect Enable 5: EDAC Enable 4: EDAC Test Mode Enable 3-0: Reserved	Memory Control Register
SCM_MEAR	Type: UINT16 Location: scmBase16 + 0x080000 Value: Address of last SBE or MBE in PROM or EEPROM	Memory Error Address Latch Register
SCM_MEM_1553_J1	Composite: {UINT16} Location: scmBase16 + 0x030000 through scmBase16 + 0x03FFFC	MIL-STD-155B Shared Memory
SCM_PROM	Composite: {UINT32} Location: scmBase32 + 0x0A0000 through scmBase32 + 0x0A7FFC	Boot PROM on the SCM
SCM_PROM_CHKS	Type: CHKS_32 Location: scmBase32 + 0x0A7FFC Value: (See CHKS_32)	Preprogrammed PROM Checksum
SCM_PROM_MBE	Type: UINT32 Location: scmBase32 + 0x0A7FF4 Value: 0xC0000000 (uncorrected)	Preprogrammed PROM Multiple Bit Error Location
SCM_PROM_SBE	Type: UINT32 Location: scmBase32 + 0x0A7FF8 Value: 0x00000000 (corrected) 0x80000000 (uncorrected)	Preprogrammed PROM Single-Bit Error Location
SCM_SSI	Composite: SCM_SSI_CSR + SCM_SSI_IBR + SCM_SSI_RCV + SCM_SSI_RSTS + SCM_SSI_SR + SCM_SSI_XMT	SCM synchronous serial hardware interface components sans the shared memory.

Table F. Data Dictionary

Name	Attributes	Description
SCM_SSI_CSR	Type: UINT16 Location: scmBase16 + 0x00C000 Bits: 15: RESETARM 14: INTRF 13: INTRB 12: INTERR 11: INTXF 10: INTXB 9: RCV EMPT 8: XMT FULL 7: Reserved 6: INTRF EN 5: INTRB EN 4: INTERR EN 3: INTXF EN 2: INTXB EN 1: Reserved 0: RCV EN	SCM SSI Control/Status Register
SCM_SSI_IBR	Type: UINT16 Location: scmBase16 + 0x00C006 Bits: 15-8: Reserved 7-0: Inter-Word Gap Units: 50 us	SCM SSI Inter-Block Gap Register The Inter-Word Gap is specified in multiples of X us.
SCM_SSI_RCV	Type: UINT16 Location: scmBase16 + 0x00C004 Bits: 15-0: RCV Word	SCM SSI Receive Register
SCM_SSI_RSTS	Type: UINT16 Location: scmBase16 + 0x00C010 Bits: 15-0: Reserved	SCM SSI Reset Register A write to this register, regardless of contents, resets the SSI controller if the RESET_ARM bit is set in SCM_SSI_CSR.
SCM_SSI_SR	Type: UINT16 Location: scmBase16 + 0x00C002 Bits: (See SCM_SSI_CSR)	SCM SSI Status Register
SCM_SSI_XMT	Type: UINT16 Location: scmBase16 + 0x00C00C Bits: 15-0: XMT Word	SCM SSI Transmit Register
SCM_TMHI	Type: UINT16 Location: scmBase16 + 0x00A004 Bits: 15-0: Timer Bits	SCM Timer Register High
SCM_TMLO	Type: UINT16 Location: scmBase16 + 0x00A006 Bits: 15-0: Timer Bits	SCM Timer Register Low

**Table F. Data Dictionary**

<b>Name</b>	<b>Attributes</b>	<b>Description</b>
SCM_TMSS	Type: UINT16 Location: scmBase16 + 0x00A008 Bits: 15-0: Timer Bits	SCM Timer Register Middle
SCM_WDR	Type: UINT16 Location: scmBase16 + 0x00C00E Bits: 15-0: Reserved	SCM Watchdog Strobe Register A write to this register, regardless of contents, strobes the watchdog timer on the Power Supply Module (PSM).
ScmBase16	Type: ADDRESS Value: 0xCF000000	Base address of SCM (D16 access) (Base address of SCM is 0xFF000000 VME.)
ScmBase32	Type: ADDRESS Value: 0xBF000000	Base address of SCM (D32 access) (Base address of SCM is 0xFF000000 VME.)
SCU_BUFFER_RATE	Type: UINT32 Range: Range of UINT32 Nominal: 69120 Units: bits/second	Rate at which data should be buffered to the spacecraft. Maximum rate is dictated by spacecraft polling rate. A value greater than the current poll rate effectively disables rate buffering.
SCU_BUFFER_SIZE	Type: UINT32 Location: see EEPROM Memory Map Range: 0x100000 (1MB) – 0x6400000 (100MB)	Size of the packetization/spacecraft transmit queues
SCU_POLL_RATE	Type: UINT32 Range: X Nominal: X Hz Units: Hz	Specifies to the DPU the rate at which the spacecraft is polling the DPU for STPDUs. Affects the internal task timeout.
SCUI_CMD	Composite: Set SCU_BUFFER_RATE command   Set SCU_POLL_RATE command   Purge science queue command	Commands affecting the SCU CSC.
SCUI_HK	Composite: Number of commands received + Bytes waiting on queue + Bytes transmitted + STPDUs transmitted	SCU Interface Housekeeping
SKIP_BIT	Type: UINT32 Value: 0x00000000-0x736B697F: Perform BIT 0x736B6970: Skip Stage 1 BIT 0x736B6971-0xFFFFFFFF: Perform BIT	Skip Stage 1 Built-In Tests
STPDU	Composite: Transfer Request Counter + 0{CCSDS_Telemetry Packet}N + Zero Fill	Company X Telemetry Protocol Data Unit used to transfer one or more CCSDS telemetry packets over the 1553 bus. Refer to document 1143-EI-S19121 for additional detail.

Table F. Data Dictionary

Name	Attributes	Description
SYSTEM_BLOCK	Composite: SYSTEM_CONFIG_AREA + SYSTEM_VOLATILE_AREA + BIT_DATA  Location: scmBase32 + 0x6F0000 through scmBase32 + 0x6FFFFFFF	System Configuration Block  Refer to the EEPROM Memory Map contained in Appendix B for additional detail.
SYSTEM_CONFIG_AREA	Composite: COLD_SKIP_BIT + COLD_MEM_SIZE + ENET_MAC + CPU_SPEED + WARM_SKIP_BIT + WARM_MEM_SIZE + BC1_START_ADDR + BC1_END_ADDR + BC1_ENTRY_ADDR + BC1_CHKS_32 + ENET_IP + ENET_HOST_IP + SC_CHKS_32  Location: scmBase32 + 0x6F0000 through scmBase32 + 0x6F01FF	System Configuration Area
SYSTEM_VOLATILE_AREA	Composite: BC_INDEX + BOOT_CNT + DPU_MODE + CQ_START_ADDR  Location: scmBase32 + 0x6F0400 through scmBase32 + 0x6F05FF	System Volatile Area
TASK_BLOCK	Composite: TASK_ENTRY + TASK_ID + TASK_PRIORITY + TASK_STATUS + TASK_PC + TASK_SP + TASK_ERRNO	Task Control Block Information
TASK_ENTRY	Composite: 8{UINT8}8	Task Entry Point Symbol, First 8 characters
TASK_ERRNO	Type: UINT32	Task Error Number
TASK_ID	Type: ADDRESS	Task Identifier, Pointer to Task Control Block
TASK_PC	Type: ADDRESS	Task Program Counter
TASK_PRIORITY	Type: UINT8	Task Priority
TASK_SP	Type: ADDRESS	Task Stack Pointer
TASK_STATUS	Type: UINT8	Task Status
TIS_ALTERNATE_1PPS	Type: Boolean	If asserted, the TIS CSC will reference the alternate (backup) 1PPS signal, else it will reference the primary 1PPS signal.



Table F. Data Dictionary

Name	Attributes	Description
TIS_CMD	Composite: Select 1PPS command   Enable/Disable Time Sync command   Set Time command	Commands affecting the TIS CSC.
TIS_HK	Composite: 1PPS Selection + Jam Enable Setting	TIS CSC Housekeeping
TIS_SYNC_ENABLE	Type: Boolean	If asserted, the TIS CSC will synchronize the DPU clock with the time provided in the CLK_MSG; else the CLK_MSG is ignored.
TMALI_BUFFER_SIZE	Type: UINT32 Location: see EEPROM memory map Range: X	Size of the DCI input event queue
TMALI_CMD	Composite: Set Frame Depth command   Set Data Timeout command	Commands affecting the TMALI CSC.
TMALI_HK	Composite: TMALI Buffer Start Address + SCM_DCI_CSR + SCM_DCI_ADHI/LO + SCM_DCI_TO + SCM_DCI_WIN	TMALI Housekeeping
TMALI_PP_LIMIT	Type: UINT32 Range: X Default: X Units: # of Frames	Specifies the limit in frames at which the DCI hardware will switch ping/pong buffers.  The limit X is arrived at by dividing the size of the buffer by the maximum number of events per frame .
TMALI_TIMEOUT	Type: UINT32 Range: X Default: X Units: milliseconds	Specifies the number of milliseconds for the DCI hardware event timeout.  The limit of X is driven by the size of the DCI hardware register which will contain this value at execution time. The minimum is driven by the frame time which is approximately 11 milliseconds, and so the default is set to twice the frame time.
TRK_FRAME_NUM	Type: UINT16 Range: X	Number of tracking frames
UINT16	Type: Fundamental, 16-bit unsigned integer	Unsigned Short Integer
UINT32	Type: Fundamental, 32-bit unsigned integer	Unsigned Integer
UINT8	Type: Fundamental, 8- bit unsigned char	Unsigned Character, Byte

# **APPENDIX A**

## **DETAILED SOFTWARE REQUIREMENTS**

Note: If reviewing this document electronically, the detailed software requirements are contained in a separate Microsoft® Excel spreadsheet file, *dpusrs-01.xls*.

### Key to Requirements Spreadsheet

The software development plan (document DPUSDP-01) provides general information regarding the content of these requirements matrices. The following paragraphs provided some additional information to assist in the interpretation of the columns and data contained in the attached requirements matrices.

#### ***Verification Level/Method/Strategy***

Each software requirement has an associated Verification Level, Method, and Strategy. The DPU SRS denotes the *highest* level of verification required for each software requirement; that is, a requirement may be verified at a higher level than that listed, but cannot be verified at a lower level exclusive of the level listed, without an approved change request. These levels are as follows:

- Software Unit Level (Ut),
- Software Verification Level (Sv)
- Instrument Verification Level (Iv).

The *Company X Specification for the Instrument A*, document number COMPANY X-INSTRUMENT A-002, describes the levels of verification shown in the following table. A mapping between the levels defined in the INSTRUMENT A DPU SRS is also provided in the table below.

INSTRUMENT A Verification Levels	Corresponding DPU FSW Verification Levels
M – Module Level	Ut – Unit Level Sv – Software Verification Level
I – Instrument Level O – Observatory Level	Iv – Instrument Level

The Test Method is one of the following:

- **Test** – the requirement is verified by a test
- **Inspection** – the requirement is verified by inspection of the design or code
- **Demonstration** – the requirement is verified by demonstrating the requirement is implemented in the system
- **Similarity** – the requirement is verified as a result of the same software being previously verified in another environment or context, and the change in context or environment cannot impact the verification.

Finally, note that the Test Strategy column provides a place to record a *suggested* (not required) method of verifying the subject requirement.

#### ***Spreadsheet Color Coding Legend***

As described, a Microsoft® Excel spreadsheet is used to author and maintain the list of software requirements. During the development, implementation, and verification of these requirements, this spreadsheet is used as a tool to help track progress. As a result, each requirements spreadsheet has a color-coded legend at the top, and the color-coding of rows in the matrix denotes the status of the various requirements, such as whether or not the requirement has been verified, and whether or not it needs a waiver. Note that when printed in black and white, the color-coded legend in the matrices are merely shaded. In a final copy of the matrices, none of the rows are color-coded, but the color-coded legend is left in the spreadsheet pages for convenience.

## **APPENDIX B**

# **EEPROM MEMORY MAPS**

The following table details the locations of the various components of the Electrically-Eraseable Programmable Read-Only Memory, **SCM\_EEPROM**.

SCM_EEPROM Memory Map	
Location	Contents
400000 through 400010	LOCATION_BLOCK for primary boot configuration (BC0)
400014 through 47FFFF	Storage for primary boot configuration (BC0)
480000 through 48FFFF	Storage for alternate boot configuration (BC1)
500000 through 6EFFFF	DOS file system block
6F0000 through 6FFFFFFF	SYSTEM_BLOCK

A description of a **LOCATION\_BLOCK** for a boot configuration is shown in the following table. There are two **LOCATION\_BLOCKS** in the EEPROM; the “Base” shown in the Location field in this table indicates an offset from the beginning of the **LOCATION\_BLOCK**.

LOCATION_BLOCK for Boot Configuration	
Location	Contents
Base + 000000	Beginning ADDRESS in EEPROM of boot configuration.
Base + 000004	Ending ADDRESS in EEPROM of boot configuration.
Base + 000008	ADDRESS in DRAM at which to copy the boot configuration for execution: Compressed: 0x500000 (VxWorks 5.2), 0x400000 (VxWorks 5.3) Uncompressed: 0x140000
Base + 00000C	ADDRESS in DRAM at which to begin execution of the boot configuration. Compressed: 0x500000 (VxWorks 5.2), 0x400000 (VxWorks 5.3) Uncompressed: 0x140000
Base + 000010	Checksum ( <b>CHKS_32</b> ) of the boot configuration.

The following table details the locations of the various components of the **SYSTEM\_BLOCK**.

SYSTEM_BLOCK Memory Map	
Location	Contents
6F0000 through 6F01FF	<b>SYSTEM_CONFIG_AREA</b>
6F0200 through 6F03FF	Not Used
6F0400 through 6F05FF	<b>SYSTEM_VOLATILE_AREA</b>
6F0600 through 6F0FFF	Not Used
6F1000 through 6F11FF	<b>BIT_DATA</b>
6F1200 through 6FFFFFFF	Science Software Application Configuration Area

The following table details the locations of the components of the **SYSTEM\_CONFIG\_AREA**.

SYSTEM_CONFIG_AREA Memory Map	
Location	Contents
6F0000	Reserved
6F0004	COLD_SKIP_BIT
6F0008 through 6F000C	Reserved
6F0010	COLD_MEM_SIZE
6F0014 through 6F0018	ENET_MAC (Ground Use Only)
6F001C through 6F0030	Reserved
6F0034	CPU_SPEED
6F0038	Reserved
6F003C	Reserved
6F0040 through 6F0050	LOCATION_BLOCK for alternate boot configuration (BC1)
6F0054 through 6F0054	Reserved
6F0058 through 6F0098	DPU_CNFG_PARMS
6F009C through 6F00F0	Reserved
6F01F4	ENET_TARG_IP (Ground Use Only)
6F01F8	ENET_HOST_IP (Ground Use Only)
6F01FC	SYSTEM_BLOCK checksum (CHKS_32)

The following table details the locations of the components of the **DPU\_CNFG\_PARMS**. Refer to the Data Dictionary for additional definitions.

DPU_CNFG_PARMS Memory Map	
Location	Contents
6F0058	DPU_CTRL_STATUS
6F005C	SCU_POLL_RATE
6F0060	SCU_BUFFER_RATE
6F0064	SCU_BUFFER_SIZE (Low Priority Queue)
6F0068	SCU_BUFFER_SIZE (High Priority Queue)
6F006C	TMALI_PP_LIMIT
6F0070	TMALI_TIMEOUT
6F0074	TMALI_BUFFER_SIZE
6F0078	DCX_BUFFER_SIZE
6F007C	EDAC_MODULUS
6F0080	EDAC_DELAY
6F0084 – 6F0098	DPA_PARM [1..6]

The following table details the locations of the components of the **SYSTEM\_VOLATILE\_AREA**.

SYSTEM_VOLATILE_AREA Memory Map	
Location	Contents
6F0400	BC_INDEX
6F0404	BOOT_CNT
6F0408	RETRY_CNT
6F040C	Reserved
6F0410	Reserved
6F0414	Exception Stack Frame Task ID
6F0418	Exception Stack Frame Vector Number
6F041C	Exception Stack Frame Stack Pointer
6F0420	Exception Stack Frame Vector Offset
6F0424	Exception Stack Frame Errno
6F0428	Exception Stack Frame Data Access Register
6F042C	Exception Stack Frame Data Storage Interrupt Status Register
6F0430	Exception Stack Frame Floating Point Control and Status Register
6F0434	Exception Stack Frame External Interrupt Mask Register 0
6F0438	Exception Stack Frame External Interrupt Mask Register 1
6F043C – 6F04B8	Exception Stack Frame General Purpose Registers [32]
6F04BC	Exception Stack Frame Machine State Register
6F04C0	Exception Stack Frame Link Register
6F04C4	Exception Stack Frame Count Register
6F04C8	Exception Stack Frame Program Counter
6F04CC	Exception Stack Frame Condition Register
6F04D0	Exception Stack Frame Fixed Point Exception Register
6F04D4	Exception Stack Frame MQ Register
6F04D8	Number of DRAM SBEs
6F04DC - 6F04E0	Last Two DRAM SBE Error Locations
6F04E4	Number of DRAM MBEs
6F04E8 - 6F04EC	Last Two DRAM MBE Error Locations
6F04F0	Number of CMM SBEs
6F04F4 - 6F04F8	Last Two CMM SBE Error Locations
6F04FC	Number of CMM MBEs
6F0500 - 6F0504	Last Two CMM MBE Error Locations
6F0508	Number of Double Words Scrubbed
6F050C - 6F05FF	Reserved

The following table details the locations of the components of the **BIT\_DATA**.

<b>BIT_DATA Memory Map</b>	
Value: (one result/word [4 bytes]) Failure: FFFFFFFF Success: 00000000	
Location	Contents
6F1000 through 6F103C	Reserved
6F1040	BIT_CPU_BRANCH
6F1044	BIT_CPU_FXPT
6F1048	BIT_CPU_FLTPT
6F104C	BIT_CPU_INT
6F1050	Reserved (BIT_CPU_STORAGE value is written and always pass)
6F1054	BIT_CPU_TIMER
6F1058	BIT_EDAC_SBE
6F105C	BIT_EDAC_MBE
6F1060	BIT_PROM_CHKS
6F1064 through 6F1078	Reserved
6F107C	BIT_SUMMARY 0x0008 – CPU Test Summary Bit 0x0010 - DCI Pong Buffer (DCI 1) 0x0020 – DCI Ping Buffer (DCI 0) 0x0040 – CMM EDAC Test Summary Bit 0x0080 – PROM Checksum Test Summary Bit 0x0200 – 1553 Test Summary Bit 0x4000 – DRAM Test Summary Bit
6F1080 through 6F1084	Reserved
6F1088	BIT_1553_RAM
6F108C	BIT_1553_INT
6F1090 through 6F10A8	Reserved
6F10AC through 6F10B0	Reserved
6F10B4 through 6F10FC	Reserved
6F1100 through 6F113C	BIT_DRAM
6F1140 through 6F11FC	Reserved



The following table details the locations of the components of the **BIT\_DRAM**.

<b>BIT_DRAM Memory Map</b>	
Value: Packed (1 bit for each 256Kb block) Failure: 1 Success: 0	
<b>Location</b>	<b>Contents</b>
6F1100	0MB-8MB Result
6F1104	8MB-16MB Result
6F1108	16MB-24MB Result
6F110C	24MB-32MB Result
6F1110	32MB-40MB Result
6F1114	40MB-48MB Result
6F1118	48MB-56MB Result
6F111C	56MB-64MB Result
6F1120	64MB-72MB Result
6F1124	72MB-80MB Result
6F1128	80MB-88MB Result
6F112C	88MB-96MB Result
6F1130	96MB-104MB Result
6F1134	104MB-112MB Result
6F1138	112MB-120MB Result
6F113C	120MB-128MB Result

## **APPENDIX C**

# **DATA PROCESSING ALGORITHMS**

The descriptions contained in this Appendix summarize the required algorithms. The algorithms are described in more detail in document XMM-OM, *DPU Processing for XMM/OM – Tracking and Compression Algorithm*.

## EVENT PROCESSING ALGORITHM

The goal of the Event Processing Algorithm is to balance rapid processing of events with responsiveness to commands. It is desirable to do this in such a way that data is dropped in scientifically manageable units. To accomplish these goals, the event processor loop must check for commands on CCD frame boundaries. Each time a timestamp is found in the data stream, the event processor loop must check for a “message waiting” flag, which is set whenever an “interesting” message arrives (not all messages delivered to the DPU are of interest to the event processor). If an interesting message is discovered, then the software breaks out of the event processing loop to service the message; otherwise, it continues processing events until another timestamp is detected.

Note that events are processed in blocks as they arrive, not as a continuous stream. The software determines the number of events waiting in the input queue, loops over those events as described above, checks for a new message, and loops back to determine the number of events waiting to be processed.

Event mode events are always filtered in software as specified by the MODE and XRT\_POSITION commands. Image mode events are never filtered as they arrive, as it takes more time to filter events than to save them in an image. The image window specification is used solely to restrict the portion of the accumulated image that is telemetered.

## SOURCE DETECTION ALGORITHM

The source detection algorithm (and software) is heritage. See XMM-OM documentation for a full description of the source detection software. The algorithm was modified slightly for use with INSTRUMENT A to use a statistics-based background estimation.

## FINDING CHART ALGORITHM

The Finding Chart algorithm produces a list of “Source” data structures sorted by the brightness of the source. Each structure contains a 5x5 pixel image centered on the brightest pixel of the source (pixel “a” in the figure below), and the detector coordinates of the base of that 5x5 image: (starXbase,starYbase). For convenience below let’s label the 5x5 image like this:

-	s	t	u	-
o	p	e	q	r
m	c	a	d	n
i	j	b	k	l
-	f	g	h	-

Where the lower left corner is the base of the image, [starXbase,starYbase].

The Finding Chart telemetry design must balance the competing interests of speed, maximum number of sources that can be sent, and size of the region around each source. A three-tier scheme is used as shown below. Three different blocks of finding chart data are produced and telemetered as a single packet. The idea is to send as much of the high and medium priority data as possible. Then fill whatever space is left with low priority data.

The ordering of fields within packets was chosen to improve the compressibility of the packets when compression code is available in Build 6. In build 5 the packets would not be compressed.

#### HIGH PRIORITY DATA BLOCK:

This data block provides the ground with source positions accurate to  $\sim 0.5''$  (by specifying the location of the brightest pixel) and rough source photometry (by supplying the brightest pixel). We expect that ground software could, if necessary or desired, estimate (via a PSF model) the source flux using only the central pixel value.

#### Tertiary Header:

Exposure Descriptor  
Exposure ID  
RA (pointing sampled at start of exposure)  
Dec (pointing sampled at start of exposure)  
Roll  
Filter ID  
  
; Image window specification  
  
X0  
Y0  
Xmax  
Ymax  
num\_sources

In Build 5 the widths of these fields will be fixed and a multiple of 8 bits; in Build 6 they will be fixed but sized to accommodate the range of the field values.

#### Source Locations:

starYbase(0) starXbase(0) {location of first source}  
...  
starYbase(n-1) starXbase(n-1) {location of last source}

In Build 5 these locations will be the position of the 5x5 source neighborhood in the 2048x2048 image, stored in 4-byte fields. In Build 6 these locations will be converted to the 1-D position within the FC window, stored in fixed-width fields sized to accommodate the number of pixels in the FC window. Obviously the central pixel's coordinates are (starXbase+2, starYbase+2).

#### Central Pixel Values:

a(0)  
...  
a(n-1)

In Build 5 these pixel values will be 2-bytes. In Build 6 they will be de-correlated by taking first differences (effective since the sources are sorted by brightness) and then coded.

#### MEDIUM PRIORITY DATA BLOCK:

This block provides the ground with the immediate neighborhood of each source, allowing more accurate positions & photometry to be estimated. This packet is truncated if the total FC telemetry exceeds 2000 bytes.

Exposure ID  
Immediate Neighborhoods:  
bcde(0)  
...  
bcde(n-1)

In Build 5 these pixel values will be 2-bytes. In Build 6 they will be de-correlated by taking first differences (effective since the neighbor pixels in a given source are likely to be similar, and because the sources are sorted by brightness) and then coded.

#### LOW PRIORITY DATA BLOCK

This block provides the ground with the more distant neighborhood of each source, allowing more accurate positions & photometry and possibly allowing the ground to determine if a source is extended. This packet is truncated if the total FC telemetry exceeds 2000 bytes.

Exposure ID  
Distant Neighborhoods:  
fghijklmnopqrstu(0)  
...  
fghijklmnopqrstu(n-1)

In Build 5 these pixel values will be 2-bytes. In Build 6 they will be de-correlated by taking first differences (effective since the neighbor pixels in a given source are likely to be similar, and because the sources are sorted by brightness) and then coded.

#### BUILD 5 SOURCE CAPACITIES:

In Build 5 where the field sizes are fixed each source consumes the following number of bytes:

Bytes in Packet		
High	Medium	Low
66	8	32

The source data that can be sent using ~2000 bytes (not counting packet overhead) depends on how many sources there are:

Number of Sources	high	med	low
1- 43	all	all	all
44-142	all	all	some
143-333	all	some	none

The software uses an application configuration parameter in EEPROM to control the maximum number of high priority sources that are sent. A value of 190 produces a finding chart packet that contains a High Priority Data Block with 190 stars and a Low Priority Data Block with 90 stars and no Low Priority data.

## CHANNEL BOUNDARIES COMPUTATION ALGORITHM

The Channel Boundaries Computation Algorithm calculates a set of channel boundaries (9 numbers) used by the Blue Processing Electronics to compute the location of each photon event to a resolution of 1/8 CCD pixel. To understand how channel boundaries are computed, an understanding of how the BPE uses the table is helpful.

A “photon event” is by definition a local maximum in the CCD pixel array. The pixel labeled “b” in Table C-1 is a local maximum. The BPE must be capable of computing the location of up to 200,000 photon events per second, but does not have the processing power to compute mathematical centroids at that rate. The BPE uses a table lookup algorithm to compute centroids.

	a	
d	b	e
	c	

**Table C-1 Photon Event Island**

For each photon event, the BPE computes two ordered pairs,  $[M_x, N_x]$  and  $[M_y, N_y]$ , where,

$$\begin{aligned} M_y &= c - a, \\ N_y &= 4b - 2a - 2c \end{aligned}$$

and,

$$\begin{aligned} M_x &= d - e, \\ N_x &= 4b - 2d - 2e \end{aligned}$$

The quantities  $M_x/N_x$  and  $M_y/N_y$  are crude estimates of the centroid in the x and y directions, which the BPE looks up in a table containing all possible values of M/N.

The BPE determines the sub-pixel location of an event by mapping the quantity  $M/N$  to a set of 9 channel boundaries that are computed by the DPU and delivered to the ICU.

In Channel Boundary Computation Mode, the BPE processes each event to the point of computing the ordered pairs  $[M_x, N_x]$  and  $[M_y, N_y]$ . It then sends this data to DPU, where the event processor software bins the data into two  $M/N$  images (two  $256 \times 256$  32-bit pixel arrays – one array for x data and one array for y data). After the exposure is complete, the software computes the quantity  $M/N$  for each pixel and histograms the result. Channel boundaries are computed from the resulting histogram by using a histogram equalization technique. That is to say that boundaries are chosen such that the number of photons in each of the eight output bins contains the same number of photons.

Channel boundaries are scaled by 1000, converted to integers, sent to the ICU, and finally loaded into the BPE. Centroid Confirmation mode is used to check the results of the channel boundary computation.

## **CENTROID CONFIRMATION ALGORITHM**

Channel boundaries are verified by binning photon events into a  $64 \times 64$  pixel image in the following way. Use the upper three bits of each event location  $[x, y]$  to compute a tile number. Use the lower three bits of each event location to compute a sub-pixel location (within the tile). The output image is telemetered to the ground for analysis.

## **CHOOSE GUIDE STARS ALGORITHM**

This algorithm will scan the bright stars in the reference frame, choose the guide stars and set up the window/memory allocation. The algorithm will be based on XMM-OM software.

## **DRIFT CORRECTION ALGORITHM**

The algorithm will be based on XMM-OM software.

## **DATA COMPRESSION ALGORITHMS**

See Science Data Compression Algorithms.