



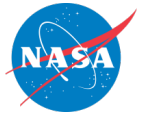
## Table of Contents

Glossary.....	ii
8.0 Small Spacecraft Avionics.....	219
8.1 Introduction.....	219
8.1.1 Chapter Organization and Disclaimers.....	220
8.2 State-of-the-Art: Onboard Computing .....	220
8.2.1 Radiation Mitigation and Tolerance Schemes .....	221
8.2.2 Computing Form Factors .....	222
8.2.3 Processor Architectures.....	223
8.2.4 Memory, Electronic Function Blocks, and Components.....	223
8.2.5 Bus Electrical Interfaces.....	224
8.2.6 Information Processing and AI .....	225
8.2.7 Highly Integrated Onboard Computing Products .....	228
8.3 State-of-the-Art: Flight Software .....	236
8.3.1 Implication of Onboard Computing on FSW .....	236
8.3.2 Frameworks .....	236
8.3.3 Operating Systems .....	239
8.4 On the Horizon: Avionics .....	239
8.5 Summary.....	240
References .....	241



## Glossary

(AI/ML)	Artificial Intelligence/Machine Learning
(ASICs)	Application-specific Integrated Circuits
(CDH)	Command and Data Handling
(CMOS)	Complementary Metal-Oxide-Semiconductor
(ConOps)	Concept of Operations
(COTS)	Commercial-off-the-shelf
(CRAM)	Chalcogenide RAM
(DDD)	Displacement Damage Dose
(DRAM)	Dynamic RAM
(EPS)	Electrical Power System
(FERAM)	Ferro-Electric RAM
(FPGAs)	Field Programmable Gate Arrays
(FSW)	Flight Software
(I/O)	Input & Output
(LEO)	Low-Earth Orbit
(MRAM)	Magnetoresistive RAM
(OBC)	Onboard Computer
(PCM)	Phase Change Memory
(PSA)	Payload and Subsystems Avionics
(Rad-hard)	Radiation-hardened
(SDRs)	Software-defined Radios
(SEEs)	Single-event Effects
(SEL)	Single-event Latch-up
(SEUs)	Single-event Upsets
(SoCs)	System-on-chip
(SRAM)	Static Random-Access Memory
(SSA)	Small Spacecraft Avionics
(SWaP)	Size, Weight, and Power
(SWaP-C)	Size, Weight, Power, and Cost
(TID)	Total Ionizing Dose



## 8.0 Small Spacecraft Avionics

### 8.1 Introduction

Spacecraft avionics encompasses all electronic subsystems, components, instruments, and functional elements that enable a spacecraft to operate and fulfill its mission. This includes the primary flight sub-elements command and data handling (CDH), flight software (FSW), and the payload and subsystems avionics. These elements are configured according to specific mission architectures, operating concepts, development environments, and applicable standards. Collectively, the avionics system forms the foundational infrastructure upon which all spacecraft functions depend—the “brain and nervous system” of the spacecraft. It provides command, control, data management, and communication interfaces to every subsystem in some manner, whether in a direct point-to-point, distributed, integrated, or hybrid computing mode.

At its core, the CDH system is the FSW running on the onboard computer (OBC) responsible for managing spacecraft operations. CDH avionics are responsible for command and telemetry processing, real-time control using sensor inputs, network management, and data storage required to meet mission requirements. The onboard computing hardware within the CDH avionics includes the OBC along with the specialized integrated circuit (IC) chips on the OBC that support critical spacecraft functions e.g., memory devices, interface controllers, or data processing components. FSW consists of programs or algorithms running on the CDH avionics and all software running on the various subsystems and payload(s). FSW provides the instructions, operating environment, and autonomy logic that allow the spacecraft to execute both routine and mission-critical functions, manage subsystem operations, and generate and transmit telemetry.

Fundamentally, the principal avionics functions do not differ greatly between large and small spacecraft since mission objectives directly influence the avionics architecture. In both traditional spacecraft and SmallSats, the avionics system can be either centralized or distributed, single-string or redundant, and modular or monolithic. What matters is the required computational performance, data bandwidth, environmental robustness, and risk tolerance. Traditional large spacecraft are typically a high-size, weight, power, and cost (SWaP-C), flagship system with high-SWaP-C avionics design to reduce risk and address higher reliability requirements. Conversely, SmallSats—especially CubeSats—are low-SWaP-C systems built with more limited margins but greater tolerance for risk due to lower cost. While this approach is straightforward and reduces mission cost, it decreases redundancy that could introduce single points of failure and does not scale well for missions with high-rate payloads or advanced processing needs such as sophisticated autonomy, data handling, or fault-tolerant features.

Advances in electronics and semiconductor engineering, and miniaturized computing have transformed spacecraft avionics primarily for the smaller class of spacecraft that were previously limited by the low SWaP-C envelope, and accordingly low complexity and redundancy. As missions have grown more ambitious, increasing demands for onboard processing, modularity, and reliability have driven a shift toward more capable processors, distributed computing architectures, and flexible FSW frameworks. These developments now enable system-, board-, and chip-level designs for small spacecraft, including CubeSats, that can achieve functional parity with larger spacecraft through distributed architectures, system-level redundancy across nodes, and collaborative data processing.

As missions venture into more dynamic or distant environments, advanced fault-tolerant designs to enhance autonomy and resilience have become major drivers in avionics development. These farther distances imply that increased computational capability onboard will have high sensitivity to radiation. Mitigation strategies—such as watchdog timers, selective power cycling, and



software-based fault isolation—can improve robustness. Modern CDH avionics employ radiation-hardened-by-design (RHBD) techniques, fault tolerant architectures, and error mitigation approaches implemented at the software, hardware, and circuit levels.

Other notable trends include the use of high-speed data interfaces for large payloads (1), onboard AI/ML accelerators for real-time decision-making, and increased focus on cybersecurity to protect command and data links (2). Additional trends include the use of commercial off-the-shelf RFSOCs to interface with radio links and attitude and control systems that require hard real-time constraints (3). Further developments include onboard AI/ML-enabled autonomy and FSW middleware designed for SmallSats constellations, supporting resource management, scheduling and task assignment, and fault tolerance.

### **8.1.1 Chapter Organization and Disclaimers**

Given the distributed and integrated nature of modern spacecraft avionics, this chapter examines onboard computing and FSW separately to clearly outline capabilities. Additionally, technology readiness for avionics can be muddled because all spacecraft avionics designs will differ per mission. Applying NASA's traditional TRL scale to onboard computing and FSW is challenging because avionics hardware and software evolve continuously often through incremental updates, reuse of components, and mission-specific customization rather than discrete, testable prototypes. Therefore, Section 8.2 describes the current state of the art of onboard computing, including components like memory, form factors, interfaces, and information processing, and radiation mitigation. Commercial products are listed in Table 8-4 under Section 8.2.7. The FSW environment is provided in Section 8.3, while Section 8.4 presents what is on the horizon for avionics development.

The information described below is not intended to be exhaustive; rather it provides an overview of current state-of-the-art technologies and their development status for small satellite subsystems. The list of organizations and companies in this chapter is not all-encompassing and does not constitute an endorsement from NASA. There is no intention of prioritizing certain companies or omit others based on their technologies or relationship with NASA. The information is provided for awareness and guidance only. The performance advertised may differ from actual performance because the information has not been independently verified by NASA subject matter experts and relies on information provided directly by manufacturers or from publicly available sources. It should be noted that TRL designations may vary depending on specific payload, mission requirements, reliability considerations, and the environment in which performance was demonstrated. Readers are highly encouraged to consult companies directly for further information regarding performance and TRL of the technology described.

## **8.2 State-of-the-Art: Onboard Computing**

The capability of onboard computing is determined by the OBC hardware, including the IC chips responsible for executing instructions and performing calculations reliably in the harsh environment of space. The OBC hardware—board design and form factor, processor choice, memory architecture, power conditioning, radiation tolerance—forms the foundation of the spacecraft's command and data handling functions. Tertiary IC chips may be integrated into the OBC, including processors for computation, memory components such as RAM and Flash for data storage, and Field-Programmable Gate Arrays (FPGAs) for flexible logic operations. These components enable the OBC to communicate with subsystems, store data, and interface with sensors, however all primary computing functions are executed by the processor within the OBC. Collectively, these capabilities enable real-time control, command execution, telemetry generation, data storage, and subsystem coordination. These functions can be implemented in several ways, and advances in avionics now enable more options for low-SWaP-C SmallSat missions.



The way the OBC is used defines the avionics architecture; specifically, how control, decision-making, and data handling are distributed. This distribution may occur through a single, centralized control authority or through subsystems with their own processors or microcontrollers performing meaningful functions. If the primary OBC is responsible for all commanding—including control loops, subsystem commanding, data handling and routing, timekeeping and scheduling, and fault detection—the avionics architecture is centralized, since a single source is performing all the meaningful functions. Conversely, if the primary OBC communicates with additional support components that can operate without commands from the OBC, the architecture is distributed at the subsystem level. Architectures where information processing is distributed across multiple nodes can almost eliminate the single-point-of-failure issues, as the system does not rely on a single computing element.

Current trends in SmallSat onboard computing generally mirror those of larger-scale systems, emphasizing greater computational power and efficiency. Centralized avionics designs remain common due to their simplicity, lower cost, and straightforward integration using a single-board OBC. However, more complex SmallSat and CubeSat missions now have feasible solutions to employ higher-performance components or distributed avionics architectures. Factors to consider when selecting onboard computing solutions include the computing form factors (how the OBC is formed), information processing enhancements, radiation mitigation and tolerance schemes, memory, and bus electrical interfaces.

Cost and availability are likely primary factors in selecting a CDH subsystem design from a given manufacturer, although many groups develop custom CDH platforms. The ability to spread nonrecurring engineering costs over multiple missions and reduce software development through reuse are both desirable in a competitive market. Heritage designs work well for customers looking to select components with proven reliability for their mission provided key components are still in production.

### **8.2.1 Radiation Mitigation and Tolerance Schemes**

While on-orbit, space radiation is a major factor to consider when designing the avionics system, and the effects of radiation on electronic devices can vary broadly (4). Radiation effects are often categorized into long-term cumulative effects and transient single-event effects (SEEs). Long-term effects are measured by total ionizing dose (TID) and displacement damage dose (DDD). TID, measured in krad, is the ionizing radiation absorbed by the device material over time, causing parametric or functional degradation of the device. DDD is the nonionizing damage caused by particle collisions with the device structure over time. SEEs occur when a single radiation particle strike deposits enough charge to cause an effect. SEEs can be destructive or nondestructive. Single-event upsets (SEUs) are nondestructive SEEs that can affect the logic state of a memory cell. Single-event latch-ups (SELs) are destructive SEEs that manifest as parasitic structures in complementary metal-oxide-semiconductor (CMOS) logic or bipolar transistor structures, potentially causing a high-current state.

Traveling into planetary space or longer LEO missions compel developers to consider reliability requirements and incorporate radiation-mitigation strategies into their respective spacecraft designs. For missions in harsher environments, spaceflight heritage and demonstrated radiation performance become primary differentiators when selecting electronics, especially compared to designs that rely heavily on commercial-off-the-shelf (COTS) parts. In contrast, experimental, high-risk missions typically focus on low-cost, easy-to-develop systems that take advantage of open-source software and hardware to provide an easy entry into space systems development, especially for hobbyists or those who lack specific spacecraft expertise. CubeSats are often either composed of only COTS components or a hybrid combination of COTS, rad-hard, and radiation-tolerant components. A default approach in SmallSat avionics has traditionally been COTS



components first (e.g., processor and memory), combined with rad-hardened supporting electronics such as ECC, watchdog timers, scrubbing, and redundancy to maximize the benefits of both technologies. These designs may also incorporate radiation-mitigation techniques to further enhance overall system reliability.

COTS components typically offer superior performance, energy efficiency, and affordability compared to their rad-hard alternatives; however, COTS devices tend to be highly susceptible to radiation. The miniaturization of electronics, sensors, and instruments has enabled spacecraft manufacturers to use more space-qualified parts. Advantages of COTS components have enabled low-cost CDH development, while also allowing developers to leverage start-of-the-art technologies in their designs.

An emerging trend is the adoption of radiation-hardened by design (RHBD) techniques within commercial semiconductor processes. In RHBD approaches, companies are modifying the transistors, gates, and physical layouts inside the silicon so the chip is inherently more resistant to radiation. These methods include specialized circuit topologies, hardened standard-cell libraries, and radiation-aware layout practices that provide protection against SEUs, SET-induced logic glitches, functional upsets, and latch-up. High-performance processors increasingly use selective hardening for the most vulnerable blocks, while applying architectural redundancy only where necessary to avoid excessive power or area overhead.

### **8.2.2 Computing Form Factors**

The underlying platform for onboard computing systems is a printed circuit board (PCB) with chip(s) soldered on, which mechanically connect and communicate depending on the PCB layout. Modular form factors use standardized physical and electrical interfaces that allow components to interoperate or be swapped. Conversely, monolithic architectures often require significant non-recurring engineering because components are custom-designed and tightly coupled.

Standard CubeSat CDH designs are PCBs that can be stacked vertically using board-to-board connectors, often without a backplane and sometimes without a formal chassis (e.g., PC/104, CubeSat card stacks). This form factor is compact and lightweight, but limited in bandwidth compared to modern formats (5). The backplane-based plug-in card (PIC) systems use cards that slide into slots and connect with a common backplane (e.g., CompactPCI, SpaceVPX). This setup supports high-bandwidth, provides rigid mechanical alignment, and allows for hot swaps in some standards, although a chassis is required (6). Some vendors have made modifications to stackable interface connectors to address reliability and throughput concerns. Many vendors use mezzanine stackable "daughter" boards to enhance modularity between subsystem elements and payloads that must adhere to form factor constraints. Mezzanine modules are small add-on cards that are mounted onto a host board (e.g., PMC, XMC). This modular package allows users to select from a variety of computational processors. NASA's R5 Realizing Rapid, Reduced-cost high-Risk Research project moved away from rigid PC/104 stacking to adopt flexible, modular frameworks that accommodate late-stage payload changes (7)(8).

Cable-connected topologies connect boards or modules through dedicated wiring harnesses rather than through stacked connectors or a backplane. In this approach, modules are linked using interfaces such as SpaceWire, Ethernet, CAN, or flexible flat-cable (FFC/flat-flex) connections. These cabling-based networks allow subsystems to be physically separated within the spacecraft while still supporting high-speed data transfer and deterministic communication paths. They also provide design flexibility, as each module can be independently located, thermally isolated, or shielded as needed.

It is also common for modern spacecraft to use a hybrid of architectures rather than committing to a single topology. For example, a central OBC might use an internal backplane for power and



housekeeping functions, while payload processors or interface modules connect via SpaceWire or Ethernet harnesses. This mixed approach enables designers to balance mass, integration simplicity, fault isolation, and performance, allowing each subsystem to be optimized for its specific role.

### **8.2.3 Processor Architectures**

The choice of processor architecture is key in the spacecraft CDH layout, and mission complexity is often reflected directly in the processing hardware selected. While avionics technology continues to evolve rapidly—especially for SmallSat platforms—there is no single “best” processor type for all missions. Instead, the diversity of mission needs, data-handling requirements, autonomy levels, and power constraints leads engineers to select among several processor classes, each offering distinct advantages.

Traditional radiation-hardened processors have long formed the backbone of spacecraft computing, valued for their reliability, radiation tolerance, and predictable real-time behavior. These devices, including rad-hard microcontrollers (MCU) and radiation-tolerant ARM-based MCUs, typically provide modest processing capability, but extremely high robustness. They are well suited for deterministic tasks such as spacecraft housekeeping, sensor interfacing, and closed-loop control. Their architectural simplicity and flight heritage make them a dependable choice for missions prioritizing longevity, safety, and stable operation over raw compute throughput.

Modern high-performance processors, including advanced microprocessors and radiation-tolerant multicore devices, offer significantly greater compute capacity, throughput, and energy efficiency than earlier designs. These processors can support demanding onboard computing, including high-rate data handling, image processing, compression, and real-time decision-making. Their ability to run more capable operating systems and support complex software stacks makes them increasingly attractive for missions requiring autonomy, rapid onboard data reduction, or support for advanced payloads. As a result, these processors can easily satisfy the processing needs of most current spacecraft subsystems and are expected to remain viable for future spacecraft bus architectures.

One of the most significant architectural shifts in recent years is the emergence of System-on-Chip (SoC) platforms. Engineers can now integrate multiple computing elements—including CPUs, hardware accelerators, memory controllers, and programmable logic—onto a single piece of silicon. SoCs dramatically reduce size, mass, and power requirements compared to earlier discrete avionics designs, while providing substantially greater computational capability. Their high level of integration also simplifies board design and can reduce interconnect complexity, making them well suited for next-generation spacecraft that require both compact hardware and extensive onboard processing.

### **8.2.4 Memory, Electronic Function Blocks, and Components**

The range of onboard memory for small spacecraft varies widely—typically from hundreds of KB to several GB—and scales upward with technological improvements. For CDH functions, onboard memory requires high reliability. A diverse suite of memory technologies are available for specific traits, including: volatile memory types for real-time processing, buffers, and temporary data handling (e.g., Static Random-Access Memory (SRAM) and Dynamic RAM (DRAM)); and non-volatile storage options for telemetry logs, subsystem data, and temporary payload storage (e.g., Magnetoresistive RAM (MRAM), Ferro-Electric RAM (FERAM), and Phase Change Memory (PCM)). SRAM remains a common choice due to its cost-effectiveness, maturity, and availability; typical space-grade SRAM is available in densities from 4 Mb to 32 Mb (0.5–4 MB), with some QSPI and NOR flash reaching higher.



Semiconductor manufacturers describe a memory chip's organization to indicate how the chip is structured internally. This typically includes its total storage capacity and the width of its data bus (such as 8-bit, 16-bit, or 32-bit). These details determine how many identical memory chips must be combined to match the processor's data-bus width. For a processor to access memory at full speed, the combined width of all memory chips must equal the CPU's bus width. If it does not, the system may need extra cycles to read or write data or require a memory controller to bridge the mismatch. Package can refer to the physical dimensions of the device, but it can also describe how the memory capacity is organized internally—such as how many bits the chip presents on its data bus.

Flash memory excels in delivering high-density storage, but has limited write cycles and slower operations. In contrast, ferroelectric RAM offers smaller capacity, yet boasts rapid write speed, low power, and ultra-high endurance—ideal for frequent data logging or sustaining mission-critical parameters. Sampler memory (SRAM/DRAM) sits in between, supporting runtime responsiveness rather than long-term data retention. Numerous manufacturers offer space-qualified, radiation-hardened memory and components certified to ESCC/QML and military standards. A chart comparing the various memory types and their performance is shown in Table 8-1.

<b>Feature</b>	<b>SRAM</b>	<b>DRAM</b>	<b>Flash</b>	<b>MRAM</b>	<b>FERAM</b>	<b>PCM</b>
<b>Non-volatile</b>	No	No	Yes	Yes	Yes	Yes
<b>Operating Voltage</b>	3.5 – 5 V	1.35-3.3 V	3.3 & 5 V	3.3 V	3.3 V	3.3 V
<b>Data Retention (70°C)</b>	N/A	N/A	>100 years	10 years	10 years	10 years
<b>Endurance (Erase/Write cycles)</b>	Unlimited	Unlimited	1E5	1E13	1E13	1E13
<b>Access Time</b>	10-25 ns	25 ns	50 ns	300 ns	300 ns	100 ns
<b>Power</b>	500 mW	300 mW	30 mW	900 mW	270 mW	Unk

### 8.2.5 Bus Electrical Interfaces

CubeSat class spacecraft continue to use interfaces that are common in the microcontroller or embedded systems world. Highly integrated systems will typically provide several interfaces to accommodate a wide range of users, and to ease the task of interfacing with peripheral devices and other controllers. FPGAs are commonly used for these interfaces because of their flexibility and ability to change interfaces as needed. Some of the most common bus electrical interfaces are listed below, along with applicable interface standards.

<b>Electrical Interfaces</b>	<b>Interface Standards</b>
Serial Communication Interfaces (SCI)	RS-232, RS-422, RS-485 etc.
Synchronous Serial Communication Interface	I2C, SPI, SSC and Enhanced Synchronous Serial Interface (ESSI)
Networks	Ethernet, LonWorks, etc.
Fieldbuses	CAN Bus, LIN-Bus, PROFIBUS, etc.
Timers	PLL(s), Capture/Compare and Time Processing Units



Discrete IO	General Purpose Input/Output (GPIO)
A/D, D/A	Analog to Digital/Digital to Analog (ADC/DAC)
Debugging	JTAG, ISP, ICSP, BDM Port, BITP, etc.
SpaceWire/SpaceFibre	a standard for high-speed serial links and networks
High-speed data	RapidIO, XAUI, SerDes and MGT protocols are common in routing large quantities of mission data in the gigabit per second speeds

### 8.2.6 Information Processing and AI

Advances in on-orbit spacecraft information processing have garnered significant attention in recent years, particularly for Earth observation missions. A spacecraft's ability to collect, process, store, and transmit data efficiently within tight constraints is the core of information processing. While on-orbit, a spacecraft (large or small) collects significant volumes of raw data, stores it temporarily, and later transmits it to Earth for post-processing and analysis. Ideally, a spacecraft would process and compute the data in real time and downlink only distilled, useful information rather than vast amounts of unfiltered raw data. Advances in onboard electronics now enable computation on the “edge” of the spacecraft—or within the CDH avionics—where incoming data can be filtered, analyzed, and interpreted directly on orbit.

This shift from basic data logging to autonomous decision-making is driving the adoption of edge computing, Machine Learning (ML), and Artificial Intelligence (AI). Edge computing focuses specifically on where and how data is processed—moving high-value computation as close as possible to the source of the data. ML refers to the techniques that allow models to identify patterns or make predictions based on onboard sensor and payload data. AI encompasses higher-level logic and decision-making processes that manage how information is interpreted, prioritized, and acted upon within the spacecraft.

Spacecraft autonomy can be considered part of management, direction, and control for all spacecraft subsystems and functions. These subsystems may also have a degree of autonomy depending on the complexity of its local “smart subsystems” processor. NASA examines autonomous systems as a cross-domain capability that enables the spacecraft to operate in a dynamic environment independent of external control, including situational and self-awareness, reasoning and acting, collaboration and interaction, and engineering and integrity. SmallSat designers have particular interest in autonomous systems applications, and studies like Whitney & Melville's 2025 work on reinforcement learning for attitude control demonstrate that even 3U CubeSats are being equipped with trusted, autonomous onboard decision-making systems (9). Another example of autonomous spacecraft operations is the ASTREA (Agentic System for Thermal Regulation and Embedded Adaptation) AI system, which uses large language model (LLM)-guided control for in-flight thermal management, demonstrating semantically aware and adaptive decision-making. In 2025, ASTREA was integrated onboard the International Space Station (ISS), and was able to understand current limitations of agentic LLM-based systems in real flight environments (10).

#### *AI Space Applications*

As a field of research, AI is more than any single technique, such as a neural-network or LLM. Instead, AI encompasses numerous approaches for planning, reasoning, search, reinforcement learning, optimization, robotics, and more. Around 2015, researchers began running AI workloads on graphics processing units (GPUs), leading to massive performance gains and paved the way for modern GPUs and neural processing units (NPU). To achieve the robust autonomous capabilities on-orbit, researchers can employ a heterogeneous architecture—combining multiple



processors like CPUs, GPUs, NPUs, or FPGAs, with increased emphasis on memory bandwidth, interconnect efficiency, and power-optimized system design (11).

These advancements have enabled onboard neural networks for autonomous station-keeping, orbit planning, and near real-time payload processing. In July 2023, D-Orbit's ION SCV004 CubeSat successfully ran the RaVAEn (Relational and Analogical Visual rEasoNing) VAE neural network onboard, demonstrating image compression and on-chip few-shot training (12).

**Table 8-3: Examples of AI Space Applications**

<b>AI Space Application</b>	<b>Description</b>	<b>Example(s)</b>
Spacecraft image analysis	AI-driven computer vision greatly accelerates the interpretation of satellite imagery.	The Advanced Intelligent Monitoring System (AIMS) rapidly analyzes imagery data collected by GOES-R spacecraft to provide actionable information that has improved both operational efficiency and mission resilience (13).
Autonomous Navigation & Robotics	Spacecraft can navigate and make decisions w/o human command.	NASA's Starling mission demonstrating inter-satellite formation flying.
Space Debris Tracking & Collision Avoidance	Machine learning can improve the tracking and predictive modeling of objects in orbit, helping identify high-risk conjunctions.	LeoLabs and Neuraspace similarly use AI to sift sensor data and predict close approaches, issuing automated "conjunction" warnings (14)(15).
Space weather forecasting	Predicts solar storms or other events that could harm spacecraft or power grids.	NOAA has several AI-based systems that leverage various data sources to generate weather forecasts comparable to those produced by traditional weather prediction systems. DAGGER (Deep Learning Geomagnetic Perturbation) that analyzes satellite measurements of the solar wind to predict the impact of solar storms on Earth with 30 minutes of advance warning (17).
Mission Planning and Optimization	Allows the spacecraft to perform automated, resource-efficient sequences of activities—such as maneuvering, data collection, and constellation inter-communication-- in various approaches with mathematical algorithms.	By the Jet Propulsion Laboratory, the ASPEN Mission Planner is an AI-assisted tool that helps streamline space mission planning and scheduling, optimizing mission efficiency; and CLASP (Coverage Planning & Scheduling) is for resource allocation and scheduling, ensuring mission activities are executed seamlessly.
Communications and Data Transmission	Cognitive communications research aims to mitigate the increasing communication complexity for mission users by increasing the autonomy of links, networks, and service scheduling.	Cognitive Communications project at NASA GRC aims to develop decentralized space networks with AI agents optimizing communication link throughput, data routing, and system-wide asset management to mitigate this challenge (16).
Satellite Health Monitoring & Predictive Maintenance	Implementing AI, machine learning, and advanced telemetry analysis systems to detect, diagnose, and mitigate potential spacecraft failures before they impact missions.	



### 8.2.7 Highly Integrated Onboard Computing Products

The demand for on-board computing capabilities, autonomous systems, and machine learning implementation has driven the commercial sector to emphasize high-performance products. Commercial-grade, highly integrated computing products—such as advanced System-on-Chip (SoC) modules and rugged compute boards—are transforming mission- and edge-critical environments. Modern data processing units that support higher-level capabilities such as autonomy, onboard information processing, and edge-computing algorithms are now widely implemented across SmallSat missions.

A variety of vendors develop highly integrated, modular, onboard computing systems for small spacecraft and CubeSats. These CDH packages combine processors and/or FPGAs with various memory banks and a variety of standard interfaces for other subsystems onboard.

System developers are gravitating toward ready-to-use hardware and software development platforms that can provide seamless migration to higher performance architectures. As with non-space applications, there is a reluctance to change controller architectures due to the cost of retraining and code migration. Following the lead of microprocessor and FPGA vendors, CubeSat avionics vendors are now providing simplified tool sets and basic, cost-effective evaluation boards.

FPGAs and software-defined architectures also give designers a level of flexibility to upload modifications and adapt to new requirements and interfaces. Emerging CubeSat avionics are increasingly integrating FPGA-based SDRs, enabling software-defined communications adaptable to multiple protocols.

Table 8-4 provides an overview of the current state-of-the-art for some of these components. It should be noted that while some products have achieved TRL 9 by virtue of a space-based demonstration, what is relevant in one application may not be relevant to another, and different space environments and/or reliability considerations may result in lower TRL assessments. Some larger, more sophisticated computing systems have significantly more processing capability than what is traditionally used in SmallSat onboard computing; however, the increase in processing power may be a useful tradeoff if payload processing and CDH functions can be combined. Overall throughput should be analyzed to ensure proper functionality under the most stressful operating conditions.



Table 8-4: Sample of Highly Integrated Onboard Computing Systems

Manufacturer Headquarters	Product	Processor/ SoC/ FPGA	Radiation Hardness Assurance (RHA)	Board Dimension (cm)	Power Consumption (W)	Orbits Flown
AAC Clyde Space <sup>Sweden</sup>	Kryten-M3	Smart Fusion 2 SoC including an ARM Cortex-M3 processor delivering 62.5 DMIPS	TID 20 krad (system level)	9.59x9.02x0.55	0.4	LEO
	Sirius OBC & TCM	32-bit LEON3FT (IEEE-1754 SPARC v8) fault-tolerant processor	TID 20 krad (system level)	9.59x9.02x1.72	1.3	LEO, lunar lander
Alén Space <sup>Spain</sup>	DARA OBC	ARM 32-bit Cortex-M7 with FPU	N/A	8.93x9.33x1.2	Max: 0.280	LEO, 2024
Aitech Systems Inc. <sup>USA</sup>	SP0-S	Power PC 1020, Alcatel RTAX	100 krad TID, SEL/LET 40 Mev-cm <sup>2</sup> /mg (without enclosure)	3U cPCI: 10.06x16x20.3	12	LEO, MEO, GEO, Lunar, Deep Space*
	SP1-S	64-bit Arm® Cortex®-A72 cores @ 1.8 GHz	100 krad TID (Target)	3U Space VPX: 10.06 x16x20.3	<25	TBD
	S-A1760	Pascal™ Architecture GPU w/256 CUDA® cores NVIDIA Denver 2 Dual-Core ARM® CPU + Cortex® A57 Quad-Core ARM® CPU	1.05 krad TID, < 1 type 2 SEFI update in 158 days (without enclosure)	12.7 x12.7x5.2	Configurable ≤ 5 Idle, 8-10 under typical CUDA load, <20 when System on Module is fully utilized	LEO*
	S-C8780	Intel® Pentium-D or Xeon-D, discrete 2D GPU, Xilinx UltraScale+ FPGA w/Dual-core ARM® CPU	TBD	3U Open VPX: 10.06 x16x20.3	Configurable 25-55	TBD
	S-C8500	Intel® Tiger Lake UP3 SoC, 4 cores/8 threads, integrated GPGPU with 96 Execution Units, Xilinx UltraScale+ FPGA w/Dual-Core ARM® CPU is Optional	1 krad TID, SEL/LET TBD (without enclosure)	3U Open VPX: 10.06 x16x20.3	CPU Configurable from 12-28, Total system 22-42 configs.	TBD
Aerospacelab <sup>Belgium</sup>	OBC	Xilinx Zynq-7045 ARM dual-core Cortex A9 running at 667MHz Kintex-7 FPGA with 350k Programmable Logic Cells, 218,600 Look-up tables (LUT) and 437,200 flip-flops (FF) Linux and FreeRTOS compatible	Radiation hardened supervisor MCU Radiation hardened watchdog Heavy ions (62.5 MeV) No destructive event	187x84.1x116.6	<4	LEO



Table 8-4: Sample of Highly Integrated Onboard Computing Systems

Manufacturer Headquarters	Product	Processor/ SoC/ FPGA	Radiation Hardness Assurance (RHA)	Board Dimension (cm)	Power Consumption (W)	Orbits Flown
Andes Aerospace Chile	Andes OBC	2x ARM 32-bit Cortex-M7 with FPU @ up to 550MHz	Estimated TID: 30krad (not tested) OBC+MRAM+NAND full cold redundancy	10x10x1.4	Depends on configuration. Active: ~0.1	LEO
Argotec Italy	OBC FERMI	Dual-Core LEON3FT SPARC V8 Processor with fault-tolerant memory controller +Microchip RTG4	Rad-hard	10.24x10x4.49	5 (depending on load)	Deep Space, Lunar Orbit, LEO
Berlin Space Technologies GmbH Germany	CDH-110	STM32F103	TID 30kRad (PCB level)	17.6x9.4x6.9	< 2	LEO
Bradford Space Netherlands	Stellar OBC + PDHS	Dual Polarfire SoC	30 krad (Si) + functional SEE resilience through SW EDAC and internal cold redundancy	13.3x11.5	< 25W Full Load / < 3 Typical / < 1 Idle	TBD
C3S Electronics Development LLC Hungary	OBC	32-bit ARM Cortex-M7	Continuous integrity check on the program memory, multi-level watchdog system, MRAM storage	9.2x8.95 x1.36 w/ daughterboard, 9.2x8.95x1.23 w/o	0.46 (measured in test mode, using eMMC as mass storage)	LEO
CesiumAstro USA	SBC-1461	1.4 GHz ARM Cortex	Rad Tolerant	5x8.4x1.3	8 typ.	LEO
	RDP-23FV	Xilinx Versal	Rad Tolerant to Rad Hard	16x10x2.54	50	LEO
Emxys Spain	ODALISS OBC	Dual 32-bit STM32 processor unit, ARM, Cortex-M7 + Cortex-M4, at 480MHz	Rad Tolerant TID of 20Krad	8.7x8.7x1	0.65	LEO
	ODALISS DIPP	Xilinx Zynq-7000 SoC Dual 667MHz 32-bit ARM Cortex-A9	Rad Tolerant TID of 20Krad	8.7x8.7x1	1.65	LEO
EnduroSat Bulgaria	OBC I	ARM Cortex M7	Tested at 40 krad TID	8.9x9.4x2.3 (with integrated GNSS)	1.5 Peak 0.2 Idle	400-600 km SSO, equatorial
	OBC II	ARM Cortex M7	40 krad TID (to be tested)	8.9 x 9.4 x 2.3 (with integrated GNSS)	6 W Peak < 1 W Idle	LEO



Table 8-4: Sample of Highly Integrated Onboard Computing Systems

Manufacturer Headquarters	Product	Processor/ SoC/ FPGA	Radiation Hardness Assurance (RHA)	Board Dimension (cm)	Power Consumption (W)	Orbits Flown
EnduroSat <sup>Bulgaria</sup>	EnduroSat GPC	NVIDIA Jetson Orin	40 krad TID (to be tested)	22x13.5x5	130 W Peak < 15 W Idle	LEO
GomSpace <sup>Denmark</sup>	NanoMind HP MK3	Xilinx Zynq 7030/7045	>20 krad	9.5x9.5x3.15	Mission dependent	LEO
Ibeos <sup>USA</sup>	EDGE-1100 (3U SpaceVPX)	AMD Ryzen SOC	TID: 30 krad SEE: >37 MeV	3U SpaceVPX; 16x10x2.5(pitch)	6-35	LEO & GEO
	EDGE-1100 Genie	AMD Ryzen SOC	TID: 30 krad SEE: >37 MeV	14.8x10.8x 3.4 – Stand-alone box	8-35	LEO & GEO
	Core-250	High Performance Space Computer Processor (HPSC), 8 x RISC-V cores, 3U SpaceVPX	--	16x10x2.5(Pitch)	<25	LEO, GEO
Innoflight <sup>USA</sup>	CFC-400XS	AMD-Xilinx Defense Grade UltraScale+ MPSoC	TID: 30 krad	8.2x8.2x2.7	6-25	LEO & GEO
	CFC-400XP	AMD-Xilinx Defense Grade UltraScale+ MPSoC	TID: 30 krad	17.2x10x2.5	4-30	LEO & MEO
	CFC-510P	AMD Ryzen GPGPU	TID: 30 krad	17.2x10x2.5	12-40	LEO
	CFC-600P	AMD-Xilinx Versal AI Edge	TID: 30 krad	17.2x10x2.5	10-70	LEO & GEO
KP Labs <sup>Poland</sup>	Antelope OBC and DPU	OBC – RM57 Herkules microcontroller (Dual 300 MHz ARM Cortex-R5F with FPU in lock-step) DPU – AMD Xilinx Zynq UltraScale+ MPSoC (ZU2EG, ZU3EG, ZU4EG, ZU5EG), Quad ARM Cortex-A53 CPU, Dual ARM Cortex-R5 in lock-step	COTS with SEE mitigation	Motherboard – 1x1x1 Daughterboard - 7x4x2	From less than 0.5 (DPU is off) to 6 (DPU processes the data)	LEO
	Leopard DPU	AMD Xilinx Zynq UltraScale+ MPSoC (ZU6EG, ZU9EG, ZU15EG); Quad ARM Cortex-A53 CPU; Dual ARM Cortex-R5 in lock-step	COTS with SEE mitigation	Non-redundant: 9x9.5x5 Redundant: 9x9.5x7.8	7 static; up to 10 for deep learning inference	LEO
	Lion DPU	AMD Xilinx Kintex Ultrascale FPGA (KU035, KU060, KU095)	COTS with SEE mitigation	16x10x5	<15 for KU035 version	TBD
Laboratory for Atmospheric and Space Physics <sup>USA</sup>	LASP Generic Small-sat FPGA Board	Kintex-7	25 krad	8.763x8.763	1	LEO



Table 8-4: Sample of Highly Integrated Onboard Computing Systems

Manufacturer Headquarters	Product	Processor/ SoC/ FPGA	Radiation Hardness Assurance (RHA)	Board Dimension (cm)	Power Consumption (W)	Orbits Flown
Micro Aerospace Solutions <sup>USA</sup>	MAS-SBC-107	Arm® Cortex®-M7	Total Ionizing Dose of 30 krad (Si)	9x9.6	<30	LEO
Nara Space Technology <sup>Busan</sup>	NSTOBC	Atmel ARM 9 based Microprocessor Unit	<24 krad	9x9x1.6	0.429 (Idle)	LEO
NOVI LLC <sup>USA</sup>	Gen 2 OBC	AMD/Xilinx Versal (AI Edge Series): Dual-core ARM® Cortex®-A72, Dual-core ARM® Cortex®-R5F, Programmable Logic (FPGA), AI Engines. Vorago ARM MCU: Low-power ARM® Cortex®-M4, Embedded rad-hard FRAM.	Radiation-hardened Vorago ARM MCU, SEL immune Xilinx processor, Optional radiation-hardened memory, Built in SEU/SEFI mitigation, ECC protected memory interfaces, Redundant storage for boot images.	9.3x9.3x2.5	<0.75W (Low Power Mode) 5-50W (High Performance Modes, Application Dependent)	LEO
Novo Space <sup>USA</sup>	SBC002AV	Zynq Ultrascale+	TID: 30 krad; Automotive parts with Rad-tolerant in high criticality parts; Rich telemetry with local event response;	16x10	application dependent active mode max: 15	TBD
	SBC003AV	SmartFusion2	Board sectorization and power control; ECC in Volatile memories;	16x10	application dependent active mode max: 8	TBD
	SBC004AF	Versal ACAP	CRC / Reed-Solomon / Interleaving in Non-Volatile memories; FPGA and fabric use selective TRM on critical functionality;	16x10	Under development	TBD
	SBC005AF	Polarfire	Scrubbing on soft FPGAs	16x10	application dependent active mode max: 8	TBD



Table 8-4: Sample of Highly Integrated Onboard Computing Systems

Manufacturer Headquarters	Product	Processor/ SoC/ FPGA	Radiation Hardness Assurance (RHA)	Board Dimension (cm)	Power Consumption (W)	Orbits Flown
Novo Space USA	GPU001AF	Jetson TX2i	TID: 30 krad	8.7x5	active mode: 9-15	TBD
	GPU002AF	Jetson Orin Nano	TID: 30 krad	8.7x5	active mode: 6-11	TBD
Pumpkin Space Systems USA	PPM A1	MCU+DSP	N/A	90x96	0.05	LEO
	PPM A2	MCU+DSP	N/A	90x96	0.05	LEO
	PPM A3	MCU+DSP	N/A	90x96	0.05	LEO
	PPM B1	MCU+DSP	N/A	90x96	0.05	LEO
	PPM D1	MCU	N/A	90x96	0.05	LEO
	PPM D2	MCU+DSP	N/A	90x96	0.05	LEO
	PPM E1	MCU	N/A	90x96	0.05	LEO
	PPM B1	MCU+DSP	N/A	90x96	0.05	LEO
	PPM D1	MCU	N/A	90x96	0.05	LEO
	MBM2 w/BBB	MCU	~5 krad	90x96	2	LEO
Ramon Space USA	NuPod	Space Grade Xilinx Versal ACAP, Cortex A72 4x / 16x Core ARM	TID: 30 krad SEL: 43 MeV·cm <sup>2</sup> /mg	28.7x21.5x7.5 Mounted enclosure	20-90	TBD
Resilient Computing USA	RadPC-SBC-001	RISC-V 32-Bit (AMD Artix 7)	COTS with SEE mitigation	10x10	1.5	LEO, Lunar
SatRev S.A. Poland	OBC	ARM Cortex-A72 v8, quad-core 1.5 GHz, 64-bit	N/A	9.95x9.6x2.15	2.8 W typical, 5 W peak	LEO
SENYT Argentina	Q-OBC	STM32	COTS w/watchdog latch-up protection	9.6x9.0x1.8 w/ enclosure	< 0.2	TBD
Sierra Space USA	UMCD	Microchip RTSX72	100kRad TID (Si)	11.4x11.4x4.4	7	LEO, GEO
SkyLabs Slovenia	NANOhpc-obc	4x RISC-V 64-bit / PolarFire / SoC	COTS w/ SEE mitigation	9.5x9.1x1.3	3	LEO
	NANOhpm-obc	32-bit RISC-V core / PolarFire	COTS w/ SEE mitigation	9.5x9.1x1.3	<5	LEO, MEO
	NANOhpc-AI	4x RISC-V 64-bit; 2x Hailo-8 AI accelerator	COTS w/ SEE mitigation	9.5x9.1x2.8	8	TBD
	NANOhpc-DPU	4x RISC-V 64-bit; Zynq™ UltraScale+™, MPSoC ZU7EV	COTS w/ SEE mitigation	9.5x9.1x2.8	11	TBD
	NANOobc-2	PicoSkyFT / IGLOO 2	COTS w/ SEE mitigation	9.5x9.1x1.1	<1 Peak, <0.9 Idle	LEO, MEO



Table 8-4: Sample of Highly Integrated Onboard Computing Systems

Manufacturer Headquarters	Product	Processor/ SoC/ FPGA	Radiation Hardness Assurance (RHA)	Board Dimension (cm)	Power Consumption (W)	Orbits Flown
Space Dynamics Laboratory USA	SDL-25	UT700 LEON3-FT + RT3P FPGA	10 krad	13x9	2-6	LEO, GEO
	LEON 4	GR740 LEON4 + RTG4	50 krad	SpaceVPX 6U	12	LEO, GEO
Space Inventor Denmark	OBC-P4	4x ARM Cortex-M7	TBD	9.3x9.3x0.875	-	LEO
	Z7000	Xilinx Zynq 7030	TBD	-	-	LEO, GEO
Spacemanic Czech Republic	DeepThought	SAMV71	-	6.7x4.2x0.7	0.06	LEO
	Eddie	MSP430	-	6.7x4.2x0.7	0.33	LEO
SPiN USA USA	MA61C CubeSat	GR712RC dual-core 32-bit LEON3 fault-tolerant, SPARC V8 processor	Processor is 300 krad	9.599x9.027	1-1.2	LEO
	MA61C smallsat	GR712RC dual-core 32-bit LEON3 fault-tolerant SPARC V8 processor	Processor is 300 krad	10.5x10.5	1-1.2	LEO
	MA61C cPCI serial space	GR712RC dual-core 32-bit LEON3 fault-tolerant SPARC V8 processor	Processor is 300 krad	16x10	1-1.2	LEO
Spiral Blue Australia	Space Edge One	Nvidia Jetson Xavier NX	N/A	10x10	7	LEO
Southwest Research Institute (SwRI) USA	Centaur SBC	GR712RC dual core LEON3-FT CPU, Xilinx Ultrascale or Microchip RT-ProAsic or RTG4 FPGA	TID: Up to 60 krad (Si), more with shielding SEL: Immune to >67eV/mg/cm2 SEU: < 1 error per 24-hour period	Available in 3U/6U cPCI form factors	Low power < 4 Operation	LEO
	HP-SBC	PowerPC based MPC8548e CPU, Microsemi RT-ProASIC FPGA, Optional: Microsemi RTG4 FPGA, Optional: Xilinx Ultrascale FPGA	TID: Up to 60 krad (Si), more with shielding SEL: Immune to >67eV/mg/cm2 SEU: < 1 error per 24-hour period	Available in 3U/6U cPCI form factors	12-20 depending on clock frequency and FPGA pairing	LEO and GEO
TelePIX Co., Ltd. South Korea	TetraPLEX OBP	1. Microchip PolarFire® SoC (MPFS250T) - 64-bit RV64GC Quad Application processing cores (667 MHz) - 64-bit RV64IMAC monitor processor core (667 MHz) 2. NVIDIA® Jetson ORIN NX 16GB System-on-Module	N/A	10x10x10	36 (max)	LEO (SSO)



Table 8-4: Sample of Highly Integrated Onboard Computing Systems

Manufacturer Headquarters	Product	Processor/ SoC/ FPGA	Radiation Hardness Assurance (RHA)	Board Dimension (cm)	Power Consumption (W)	Orbits Flown
		- 64-bit Cortex A78AE Armv8.2 (Eight-core) with Ampere GPU (1024 NVIDIA® CUDA® cores 32 Tensor cores)				
Trident Space Electronic Systems <sup>USA</sup>	VDRT	Versal VC1902	30 krad / 50MeV LU	10x14.6x2.54	<60	LEO/ MEO
	UDRT	MPSoC ZU19	30 krad / 40MeV LU	10x14.6x2.54	<50	LEO
UARX Space <sup>Spain</sup>	OBC	SmartFusion2 FPGA w/Cortex M3 SoC	Mission specific RHA to be performed	10x16	5	LEO
Unibap <sup>Sweden</sup>	iX10-100A	AMD Ryzen V1000 (CPU and GPU), Intel Myriad-X (VPU)	Radiation Tolerant COTS with SEE mitigation	10x10x6	25-40	LEO
	iX5-106	AMD Stepe Eagle (CPU and GPU), Intel Myriad-X (VPU) Microchip SmartFusion2 (FPGA)	Radiation Tolerant COTS with SEE mitigation	10x10x5	15-25	TBD
Xiphos <sup>Quebec</sup>	Q7S	AMD-Xilinx Zynq-7020 Dual-core ARM Cortex-A9	25 krad	7.8x4.3x0.9	2	LEO
	Q8S	AMD-Xilinx Zynq Ultrascale+ MPSOC Quad-core ARM Cortex-A53	30 krad	8x8x1.12	>5	LEO
	Q8Js	AMD-Xilinx Zynq Ultrascale+ MPSOC Quad-core ARM Cortex-A53	30 krad	8x8x1.12	>5	LEO
ZAITRA <sup>Czech Republic</sup>	SKAIDOCK	AMD-Xilinx Zynq Ultrascale+ MPSOC, Quad-core ARM Cortex-A53	30 krad	9.01x9.4x2.91	5-20	LEO

\*Orbits flown are on larger spacecraft



### 8.3 State-of-the-Art: Flight Software

Complexity in FSW can come from the architecture (e.g., the topology of the subsystems, automation/autonomy, and fault management), as well as the operational concept. Since the verification effort scales with software complexity, this process—if not properly regarded—can drive up the cost and schedule for a program or mission. Designs that enhance reliability and fault tolerance come at a cost of complexity, which further increases development overhead, and changing requirements will compound these costs. It is recommended to engage software teams early in the planning process. Increased processor performance, proliferation of rad-tolerant components, and higher mission autonomy are driving demands on FSW, requiring earlier, more integrated planning, richer tool chains, and deeper verification processes (19).

While the majority of FSW advances are proprietary within private companies, there are many considerations when choosing a development environment and/or operating system for a space mission such as memory and processing requirements, cost and schedule, software heritage and maturity, and subsystem-specific availability (20). The OBC operating system can range from lightweight RTOS (FreeRTOS, Zephyr, RTEMS) to full Linux stacks, depending on resource availability, timing requirements, and FSW needs. Onboard computing and FSW have a symbiotic relationship, since all spacecraft operations must be performed in a reliable and predictable fashion throughout the lifetime of the mission. The needs of each mission can vary greatly, but basic deterministic and reliable processing is a fundamental requirement. Adding additional “nice to have” features can balloon the overall complexity of the FSW, reduce the efficacy of testing, and increase risk to the mission as a whole.

It is essential that mission-critical FSW be as simple as possible. Survey options and plan early in any FSW effort. Wherever possible, early development and testing should be performed. Efforts to add additional features should be evaluated critically from a default posture of sticking to the baselined plan. With good planning and careful execution, a favorable outcome can be achieved. It is becoming more common to update software after the hardware is delivered (or even launched), and there are many software frameworks such as cFS that have features to enable software updates after deployment.

#### 8.3.1 Implication of Onboard Computing on FSW

The processor and memory available on the CDH can impose significant limitations on the FSW, particularly when considering whether the selected processor can support the required operating system and its board support package. For smaller or simpler missions, or to reduce electronic complexity, designers often use lower-capacity embedded processors—sometimes in distributed processing architectures—which may also rely on dedicated onboard memory. These processors may have limited threading capabilities (for example, single-threaded execution), which are simpler to program and test but can further restrict FSW design choices. In contrast, modern integrated space avionics increasingly adopt heterogeneous and mixed-criticality architectures. These advanced configurations support more capable operating systems, enable multithreaded software execution, and facilitate complex redundancy strategies such as multiple modular redundant systems at the cost of complexity. Collectively, these trends allow for more robust radiation tolerance, higher system reliability, and expanded operational flexibility for critical small spacecraft missions.

#### 8.3.2 Frameworks

Software frameworks impose boundaries and constraints that shape how the system is designed. A framework constrains the system-level behaviors (messaging, scheduling, component interactions) through its own internal structure—including coding patterns, file structure, message formats, lifecycle methods, state machine boundaries, or timing structures. This also enables



infrastructure services such as command handling, telemetry routing, time synchronization, and others that are either too expensive or error-prone to be done ad hoc.

While frameworks provide a lot of structure, mission-specific algorithms, guidance/navigation logic, and autonomous decision-making procedures are still required. Factors to consider when selecting a framework include the required timing cadence, how to partition responsibilities across components, how to handle concurrency, and how to balance real-time constraints. As an example, NASA's Core Flight System (cFS) is geared toward homogeneous, single-node avionics architectures and is not optimized for extremely tight computational environments (21). F Prime (F') is more rigid than cFS, with much of the architecture baked-in at compile time, although it can be extended to distributed systems with additional effort (22)(23). GERICOS has less flight heritage, but offers greater architectural freedom and supports heterogeneous, distributed architectures (24).

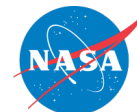
The challenges of flight software are frequently discovered during integration with late-schedule implementations. The more complicated the FSW is, the more risk is introduced into the mission. To manage this complexity, teams should implement early FSW prototyping, version control, and continuous testing, especially when integrating custom or COTS hardware (25), and consider using mature frameworks to reduce the amount of new code to be verified.

Component-based design makes a framework modular by breaking software into small, self-contained units with well-defined interfaces. Each component can be developed, tested, reused, or replaced independently, which reduces coupling and improves maintainability. The CCSDS (Consultative Committee for Space Data Systems) is an important consideration when selecting a framework because its standards support modular and interoperable system design. By defining consistent interfaces for data exchange, commanding, and communication, CCSDS enables reusable parts to integrate cleanly, regardless of the underlying hardware or software implementation. Other helpful considerations are framework services such as OS system support, which describes how well a software framework integrates with—and depends on—the operating system's services and constraints across target platforms. A supporting test suite is a packaged set of tests that comes with the framework to verify that the framework is correctly built, correctly installed, and behaving as expected on supported platforms. There is a trade-off in evaluating frameworks by their date of last release: older software can be stable and battle-tested or outdated and unmaintained, while freshly released software could be responsive to user reports but lacking maturity and testing. The reader is recommended to verify the latest versions of all listed frameworks in Table 8-5.



Table 8-5: FSW comparison table

Framework	Primary purpose/ Scope	Core Language	Open-Source?	Modularity	CCSDS	Released Quarter
cFS (core Flight System) (26)	Reusable flight software framework for spacecraft of all sizes (CubeSat to flagship).	C	GitHub cfs.gsfc.nasa.gov	Yes	Yes	Q1 2026
F' (F Prime)	Used to model embedded systems framework.	C++	github.com/nasa/fprime	Yes	No	Q3 2025
NanoSat Mission Operations Framework	Supports missions by facilitating the development, distribution, and deployment of Apps on satellite missions.	Java	nanosat-mo-framework.github.io	Yes	Yes	Q2 2025
Space ROS/ ROS 2 (Space Robot Operating System)	Reusable software framework based on ROS2 for robotics and autonomous space systems.	C++ 14	github.com/david-dorf/spaceros_gz_demos	Yes	With Bridge	Q1 2026
PyCubed	Open-source, radiation-tested CubeSat avionics platform.	Python	github.com/pycubed	Yes	Yes	Q1 2023
pysimCoder	Rapid Prototyping application, that can be used to generate realtime code for different targets.	Linux	www.robots5.com/pysimcoder-installation-tutorial github.com/robertobucher/pysimCoder	Yes	Not native	Q3 2022
Matrix Laboratory	Proprietary programming and numeric computing platform.	Matlab	www.mathworks.com/discovery/what-is-matlab.html	Yes	Yes	Q3 2025
GENERIC Onboard Software (GERICOS)	Offers a set of generic, reusable and customizable software components for the rapid development of payload flight software.	C++	No	Yes	No	Unk
Realtime Object-Oriented Distributed Operating System (RODOS)	a real-time embedded operating system.	C, C++	github.com/SpaceTeam/rodos	Yes	Yes	Unk



### 8.3.3 Operating Systems

Operating systems, when used, sit between flight software and the OBC to manage computer hardware, software resources, and provide common services for computer programs. They are primarily selected based on software compatibility requirements and real-time responsiveness. The industry has largely moved toward real-time operating systems (RTOS) or lightweight Linux distributions to manage multitasking and hardware resource allocation. Examples of commonly used operating systems are listed in Table 8-6.

<b>Name</b>	<b>Description</b>	<b>Real-time?</b>	<b>Implementation Language</b>	<b>Open-Source?</b>
VxWorks	Deterministic, hard real-time operating system.	Yes	C, C++	No
RTEMS	Hard real-time OS designed for embedded and space systems.	Yes	C	Yes, RTEMS License
FreeRTOS	Lightweight real-time kernel for microcontrollers.	Yes	C	Yes, MIT License
Linux	Free, open-source operating system kernel.	Not standard	C	Yes, GNU GPL
Zephyr OS	Real-time OS for embedded systems.	Yes	C	Yes, Apache 2.0 License

## 8.4 On the Horizon: Avionics

Emerging onboard computing trends reside in the improvement of computing capacity of embedded systems (via edge computing) and the implementation of AI/ML algorithms to enable autonomous operations. Many CDH systems will continue to follow trends set for embedded systems. Short-duration LEO missions will continue to take advantage of advances made by industry leaders who provide embedded systems, technologies, and components. In keeping with the low-cost, rapid development theme of CubeSat-based missions, many COTS solutions are available and have flight heritage.

An area of growing interest for expanding into interplanetary space is rendezvous, proximity, and docking operations (RPOD), and the technologies that support these capabilities. Several projects across the space community are actively developing autonomous docking capabilities using AI-based planning. For instance, a new system called Autonomous Rendezvous Transformer (ART) uses a Transformer neural network to generate optimal, safe RPOD trajectories autonomously (31); a Carleton University research team is developing AI-guided satellite docking using computer vision and autonomy to manage relative motion and control (32); and DARPA's AI Next initiative enables advanced autonomy, including docking capabilities.

Cybersecurity is also becoming a driving factor in FSW. This is mostly handled through the encryption of commands, but as space becomes more crowded, more sophisticated cybersecurity schemes may be required of SmallSats. SmallSats can also be a testbed for demonstrating new hardware and software for cybersecurity.

To significantly improve the capabilities of space qualified computing technology, NASA's High-Performance Spaceflight Computing project is developing a radiation-tolerant, multi-GFLOPS CPU with extremely low power consumption. It's designed to dramatically enhance onboard computational power for spacecraft and aviation systems. This next generation,



fault-tolerant, cache-coherent multicore SoCs with onboard Ethernet are designed for  $\geq 100\times$  performance over current spaceflight processors (33).

Neuromorphic computing has emerged as a promising solution to the amplified challenges of data throughput, energy efficiency, and memory bandwidth as a result of edge computing and real-time AI applications. To close the gap between terrestrial neuromorphic computing capabilities and embedded space computing, researchers have begun to focus heavily on radiation-tolerant neuromorphic architectures, which aim to preserve energy-efficient, low-latency processing even in harsh radiation environments. Recent work extends neuromorphic principles—such as spiking neural networks, event-driven computation, and in-memory processing—to aerospace tasks, highlighting their potential to significantly increase onboard compute efficiency for autonomous navigation, hazard detection, and real-time decision-making. This growing research direction can be viewed as a neuromorphic extension of traditional space computing, leveraging biologically inspired processing paradigms to deliver higher performance within the severe power, volume, and reliability constraints of spacecraft systems (34).

Advancements reflecting Industry 4.0 trends continue in spacecraft avionics, predominantly including autonomy and onboard intelligence. AI/ML & machine vision are integrated through onboard accelerators and neural inference engines (35)(36)(37)(12). Digital Twins are being applied such as digital-twin-driven ADCS design for CubeSats fitted with robotic manipulators (36). Finally, software-defined-everything continues advancing avionics flexibility alongside FPGA stacks, reconfigurable radios, and modular SBC designs.

Containerization for FSW is the idea of packaging flight software components inside standardized, isolated runtime environments—like Docker or OCI containers used on Earth—so the software becomes more portable, modular, testable, and maintainable across different embedded flight processors. While containerization is commonplace terrestrially, adapting it for spaceborne embedded computing introduces unique constraints and opportunities.

Some modern tools, technologies, and approaches that should be considered when developing and deploying next-generation small spacecraft avionic systems are:

- Robotics and automation
- Model-based systems engineering
- Cloud computing
- Augmented reality/ virtual reality / mixed reality
- Advanced manufacturing

## 8.5 Summary

Space applications now require considerable autonomy, precision, and robustness, and are refining technologies for such operations as on-orbit servicing, relative and absolute navigation, inter-satellite communication, and formation flying. An exciting trend is that small spacecraft missions are becoming more complex as these platforms are now being used for lunar and deep space science and exploration missions. Small spacecraft technology is expanding to meet the needs of increasing small spacecraft mission complexity. This has accelerated over the past few years to achieve the next-gen goals of using small spacecraft to collect important science in deep space, and mitigate risk for larger, more complex mission-critical situations. In parallel, spacecraft electronics have matured with higher performance and reliability, and with miniaturized components that meet the growing needs of these now very capable spacecraft.

Constellation networks and swarms, synchronized formations, and other multi-satellite cluster formations have opened new opportunities for avionics systems. NASA's Starling swarm of four CubeSats have advanced several technologies that enable a spacecraft to function in a



synchronized manner without ground support (39). The Starling CubeSats also communicate together about their relative position without external commands. These autonomous operations do impose new constraints on the avionics system—for both single satellites and multi-satellite configurations—as the overall mission performance is dependent on all the platform elements acting in a co-dependent fashion.

Beyond raw processing capability, edge computing has emerged as a key trend, particularly for Earth observation missions, enabling onboard data management and reducing reliance on ground-based processing. Multiple vendors offer components that can be readily integrated into space-rated systems. Processing needs for traditional CDH functions are relatively stable, though SmallSats flying increasingly data-heavy payloads (i.e., imaging systems) have significantly raised the demand for onboard data-intensive processing. High-performance processors used to be added as a separate payload processing element to the CDH, but modern distributed avionics architectures can integrate high-performance, FPGA-based SDRs into the primary CDH platform.

The Small Spacecraft Avionics chapter has been updated with a broader, interrelated framework, in which CDH, FSW, and smart payloads are not just independent space platform subsystems but are part of an integrated avionics ecosystem which includes all electronic elements of a space platform, now primarily digitally based or managed. Also, avionics should not be considered as an isolated spaceflight technology component, but rather as a core digital engineering technology emphasis area capable of integrating products, processes, and technologies from other disciplines. To continue to be relevant and efficient, avionics communities must remain cognizant and receptive of the continuously evolving nature of the digital-based Industry 4.0 technology revolution seen in other related or associated vertical disciplines and solutions.

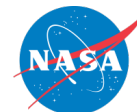
For feedback solicitation, please email: [arc-sst-soa@mail.nasa.gov](mailto:arc-sst-soa@mail.nasa.gov). Please include a valid business email for further contact.

## References

- (1) European Space Agency, "SpaceFibre," [Online] Accessed: February 7, 2026, Available at: [https://www.esa.int/Enabling\\_Support/Space\\_Engineering\\_Technology/Onboard\\_Data\\_Processing/SpaceFibre](https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Onboard_Data_Processing/SpaceFibre)
- (2) Cybersecurity and Infrastructure Security Agency, "Recommendations to Space System Operators for Improving Cybersecurity," April 2024.
- (3) P. Fiedler, E. Meenzen, et al. "Porting a Real-Time Operating System – A Student Endeavour," 2024 Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V., <https://doi.org/10.25967/630347>.
- (4) National Academies of Sciences, Engineering, and Medicine: "Testing at the Speed of Light: The State of U.S. Electronic Parts Space Radiation Testing Infrastructure." Washington, DC: The National Academies Press. 2018.
- (5) RTD Embedded Technologies, Inc. "What is PC104? Rugged, Compact, Stackable and Modular," [Online] Accessed: February 7, 2026, Available at: <https://www.rtd.com/PC104/>
- (6) Curtiss-Wright. "What is VPX," [Online] Accessed: February 7, 2026, Available at: <https://defense-solutions.curtisswright.com/capabilities/open-architectures/other/vpx>
- (7) K. Knesek, M. Alexander and J. Wisbiski, "Avionics Design Architecture for Low-Cost CubeSat Missions and Lessons Learned from R5-S2 and R5-S4," 2025 IEEE Aerospace Conference, Big Sky, MT, USA, 2025, pp. 1-15, doi: 10.1109/AERO63441.2025.11068544.
- (8) H. Zhang, J. Zhou, and G. Liu, "A Building Block Assembly Based Modular Microsatellite Scheme," 2023, Proceedings of 2022 International Conference on Autonomous Unmanned Systems (ICAUS 2022), ICAUS 2022. Lecture Notes in Electrical Engineering, vol 1010. Springer, Singapore. [https://doi.org/10.1007/978-981-99-0479-2\\_12](https://doi.org/10.1007/978-981-99-0479-2_12)



- (9) C. Whitney and J. Melville, "Toward Trusted Onboard Artificial Intelligence (AI): Advancing Small Satellite Operations using Reinforcement Learning," 2025, 39th Annual Small Satellite Conference, Salt Lake City, UT. <https://arxiv.org/pdf/2507.22198>
- (10) A.D. Mousist, "ASTREA: Introducing Agentic Intelligence for Orbital Thermal Autonomy," October 11, 2025. <https://arxiv.org/pdf/2509.13380>
- (11) "The Edge AI Technology Report 2026: How Intelligence at the Edge Is Reshaping Electronics Design," findchips.com Blog, March 11, 2026, [Online] Accessed: March 19, 2026, Available at: <https://blog.findchips.com/the-edge-ai-technology-report-2026-how-intelligence-at-the-edge-is-reshaping-electronics-design/>
- (12) V. Růžička, G. Mateo-Garcia, et al. "Fast model inference and training on-board of Satellites," Computer Science: Artificial Intelligence, July 17, 2023, <https://arxiv.org/abs/2307.08700>
- (13) Z. Li, "Predictive 'AIMS' AI Tool Has Been Aiding Weather Satellite Operations for Years," July 25, 2024, ASRC Federal, [Online] Accessed: February 7, 2026, Available at: <https://www.asrcfederal.com/predictive-aims-ai-tool-has-been-aiding-weather-satellite-operations-for-years/>
- (14) LeoLabs, "Real-time conjunction alerts for safety of flight," [Online] Accessed: February 7, 2026, Available at: <https://leolabs.space/conjunction-alerts/>
- (15) Neuraspace, "Satellite Collision Avoidance with AI," [Online] Accessed: February 7, 2026, Available at: <https://content.neuraspace.com/product-launch>
- (16) National Aeronautics and Space Administration, "Cognitive Communications," August 5, 2025, [Online] Accessed: February 7, 2026, Available at: <https://www.nasa.gov/glenn/glenn-expertise-space-exploration/scan/cognitive-communications/>
- (17) V. Thomas, "NASA-enabled AI Predictions May Give Time to Prepare for Solar Storms," March 30, 2023, [Online] Accessed: February 7, 2026, Available at: <https://www.nasa.gov/science-research/heliophysics/nasa-enabled-ai-predictions-may-give-time-to-prepare-for-solar-storms/>
- (18) [https://spacenews.com/the-stealth-strategy-pays-off-uarx-space-emerges-as-europes-high-reliability-powerhouse/?utm\\_source=linkedin&utm\\_medium=jetpack\\_social](https://spacenews.com/the-stealth-strategy-pays-off-uarx-space-emerges-as-europes-high-reliability-powerhouse/?utm_source=linkedin&utm_medium=jetpack_social)
- (19) A. Prajapati, "Evolution of core Flight System (cFS)," NASA Technical Reports Server (NTRS), [Online] Accessed: February 7, 2026, Available at: <https://ntrs.nasa.gov/api/citations/20240004389/downloads/cFSFSW2024v10.pdf>
- (20) Build a CubeSat, "Flight Software," CubeSat Resources, [Online] Accessed: February 7, 2026, Available at: <https://cubesat-resources.space/development/flight-software/>
- (21) J. Furgala, S. Jero, A. Lin, and R. Skowyra, "Porting NASA's core Flight System to the Formally Verified seL4 Microkernel," Workshop on Security of Space and Satellite Systems (SpaceSec), 23 February 2026, San Diego, CA.
- (22) Jet Propulsion Laboratory, "F Prime," [Online] Accessed: February 7, 2026, Available at: <https://fprime.jpl.nasa.gov/>
- (23) R. L. Bocchino, J. W. Levison and M. D. Starch, "FPP: A Modeling Language for F Prime," 2022 IEEE Aerospace Conference (AERO), Big Sky, MT, USA, 2022, pp. 1-15, doi: 10.1109/AERO53065.2022.9843754.
- (24) P. Plasson, G. Brusq, F. Singhoff, H.N. Tran, S. Rubini, and P. Dissaux, "PLATO N-DPU ON-BOARD SOFTWARE: AN IDEAL CANDIDATE FOR MULTICORE SCHEDULING ANALYSIS," February 2022.
- (25) J.M. Bellardo, "Software: The Overlooked Glue that Holds CubeSats Together," June 26, 2020, [Online] Accessed: February 7, 2026, Available at: [https://www.nasa.gov/wp-content/uploads/2019/05/cubesat\\_software.pdf](https://www.nasa.gov/wp-content/uploads/2019/05/cubesat_software.pdf)



- (26) National Aeronautics and Space Administration, "The core Flight System," 2026, [Online] Accessed: February 7, 2026, Available at: <https://etd.gsfc.nasa.gov/capabilities/core-flight-system/>
- (27) Jet Propulsion Laboratory, "Introduction To F'," [Online] Accessed: February 7, 2026, Available at: <https://fprime.jpl.nasa.gov/latest/docs/user-manual/overview/01-full-intro/#the-origins-of-f>
- (28) "NanoSat MO Framework," <https://github.com/esa/nanosat-mo-framework/tree/master>
- (29) "Space ROS," [Online] Accessed: February 7, 2026, Available at: <https://space.ros.org/>
- (30) Observatoire de Paris, "PLATO N-DPU Flight Software," January 15, 2024, [Online] Accessed: February 7, 2026, Available at: <https://lesia.obspm.fr/PLATO-N-DPU-Flight-Software.html>
- (31) Stanford University, "Project: Autonomous Rendezvous Transformer (ART)," Stanford Space Rendezvous Laboratory, [Online] Accessed: February 7, 2026, Available at: <https://slab.stanford.edu/projects/art>
- (32) J. Careless, "Carleton University Team Demonstrates Autonomous Satellite Docking in Lab," spaceq.com, February 8, 2024, [Online] Accessed: February 7, 2026, Available at: <https://spaceq.ca/carleton-university-team-demonstrates-autonomous-satellite-docking-in-lab/>
- (33) W. Powell, "High Performance Spaceflight Computing (HPSC) for Lunar and Planetary Missions," Planetary Science Conference, March 12, 2025.
- (34) Y. Wu, R. Long, Y. Zhu, Q. Jiang, Y. Zhan, et al. "Radiation-tolerant hafnium-based ferroelectric memcapacitors for neuromorphic computing," Materials Today Communications, Volume 48, 2025, 113567, ISSN 2352-4928, doi.org/10.1016/j.mtcomm.2025.113567.
- (35) F. Ortiz, V. Monzon Baeza, L.M. Garces-Socarras, et al. "Onboard Processing in Satellite Communications Using AI Accelerators. Aerospace. 2023; 10(2):101. doi.org/10.3390/aerospace10020101
- (36) Australian Space Agency, "Aussie SpIRIT completes first phase of its mission," September 23, 2025, [Online] Accessed: February 7, 2026, Available at: <https://www.space.gov.au/news-and-media/aussie-nanosatellite-spirit-completes-first-phase-of-its-mission-september-2025>
- (37) European Space Agency, "Φsat-2 begins science phase for AI Earth images," July 15, 2025, [Online] Accessed: February 7, 2026, Available at: [https://www.esa.int/Applications/Observing\\_the\\_Earth/Phsat-2/Phsat-2\\_begins\\_science\\_phase\\_for\\_AI\\_Earth\\_images](https://www.esa.int/Applications/Observing_the_Earth/Phsat-2/Phsat-2_begins_science_phase_for_AI_Earth_images)
- (38) B. Koszek and Roman Sanchez, "Design of an ADCS enabled by the development of a digital twin of CubeSat equipped with robotic manipulator," Smallsat Europe 27th - 28th May 2025.
- (39) National Aeronautics and Space Administration, "What is Starling?" March 19, 2026, [Online] Accessed: March 19, 2026, Available at: <https://www.nasa.gov/smallspacecraft/what-is-starling>