

---

---

**ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ СИСТЕМЫ**  
**INFORMATION-MEASURING AND CONTROL SYSTEMS**

---

---

УДК 629.783  
DOI: 10.17586/0021-3454-2024-67-1-33-45

**АРХИТЕКТУРА ОТКАЗОУСТОЙЧИВОЙ РАСПРЕДЕЛЕННОЙ  
СИСТЕМЫ УПРАВЛЕНИЯ НАНОСПУТНИКОМ SamSat-ION**

В. Д. МЕЩЕРЯКОВ<sup>1\*</sup>, П. Н. НИКОЛАЕВ<sup>1,2</sup>, С. В. ШАФРАН<sup>1,2</sup>, А. С. ЭСПИНОЗА ВАЛЛЕС<sup>1</sup>

<sup>1</sup>*Самарский национальный исследовательский университет им. акад. С. П. Королева, Самара, Россия*  
*\*meshcheryakov.vd@ssau.ru*

<sup>2</sup>*Институт космических исследований РАН, Москва, Россия*

**Аннотация.** Представлены ключевые особенности архитектуры отказоустойчивой распределенной системы управления наноспутником SamSat-ION. Одна из особенностей архитектуры — использование подхода „ведущий/ведомый“, для чего в каждой бортовой системе установлены равнозначные контроллеры, подключенные к общей шине данных, и каждый контроллер может выполнять роль ведущего на шине и обращаться напрямую к другим системам. Оснащение бортового компьютера двухъядерным контроллером с асимметричной структурой ядер позволяет увеличить производительность — осуществлять сбор и хранение телеметрии на ведомом ядре, а выполнение циклограммы полета — на ведущем ядре. Информационный обмен между системами наноспутника SamSat-ION осуществляется с помощью синхронной шины передачи данных I2C; все бортовые системы могут управляться как с бортового компьютера, так и с приемника по командам из центра управления полетом наноспутников, что повышает отказоустойчивость SamSat-ION.

**Ключевые слова:** наноспутник, отказоустойчивость, архитектура программного обеспечения, многоядерные контроллеры, многоконтроллерные системы

**Благодарности:** исследование выполнено при поддержке Российского научного фонда, грант № 23-72-30002; <https://rscf.ru/project/23-72-30002>.

**Ссылка для цитирования:** Мещеряков В. Д., Николаев П. Н., Шафран С. В., Эспиноза Валлес А. С. Архитектура отказоустойчивой распределенной системы управления наноспутником SamSat-ION // Изв. вузов. Приборостроение. 2024. Т. 67, № 1. С. 33—45. DOI: 10.17586/0021-3454-2024-67-1-33-45.

**ARCHITECTURE OF A FAULT-TOLERANT DISTRIBUTED CONTROL  
SYSTEM FOR THE SamSat-ION NANOSATELLITE**

V. D. Meshcheryakov<sup>1\*</sup>, P. N. Nikolaev<sup>1,2</sup>, S. V. Shafran<sup>1,2</sup>, A. S. Espinoza Valles<sup>1</sup>

<sup>1</sup>*Samara University, Samara, Russia*  
*\*meshcheryakov.vd@ssau.ru*

<sup>2</sup>*Space Research Institute of the RAS, Moscow, Russia*

**Abstract.** The key features of the architecture of a fault-tolerant distributed control system for the SamSat-ION nanosatellite are presented. One of the features of the architecture is the use of a “master/slave” approach, for which each on-board system has equivalent controllers connected to a common data bus, and each controller can act as a master on the bus and access other systems directly. Equipping the on-board computer with a dual-core controller with an asymmetric core structure allows to increase productivity - collect and store telemetry on the slave core, and execute the flight cyclogram on the master core. Information exchange between SamSat-ION nanosatellite systems is carried out using a synchronous I2C data transmission bus; all on-board systems can be controlled both from the on-board computer and from the receiver using commands from the nanosatellite flight control center, which increases the fault tolerance of SamSat-ION.

**Keywords:** nanosatellite, fault tolerance, software architecture, multicore controllers, multicontroller systems

---

© Мещеряков В. Д., Николаев П. Н., Шафран С. В., Эспиноза Валлес А. С., 2024

**Acknowledgments:** the research was supported by the Russian Science Foundation, grant No. 23-72-30002; <https://rscf.ru/project/23-72-30002>.

**For citation:** Meshcheryakov V. D., Nikolaev P. N., Shafran S. V., Espinoza Valles A. S. Architecture of a fault-tolerant distributed control system for the SamSat-ION nanosatellite. *Journal of Instrument Engineering*. 2024. Vol. 67, N 1. P. 33—45 (in Russian). DOI: 10.17586/0021-3454-2024-67-1-33-45.

**Введение.** В настоящее время наблюдается значительный рост числа космических миссий, одной из главных причин этого является использование наноспутников в исследовательских и образовательных задачах. В данном сегменте задач предпочтительно применение наноспутников благодаря использованию коммерчески доступных компонентов, что позволяет ускорить процесс разработки и удешевить миссию [1, 2]. Вместе с тем происходит усложнение космических миссий, что требует более гибких как аппаратных, так и программных архитектурных решений [3—5].

Настоящая статья посвящена разработке архитектуры отказоустойчивой распределенной системы управления научно-образовательным наноспутником SamSat-ION формата CubeSat 3U. SamSat-ION был разработан в Самарском университете и запущен 27 июня 2023 г. с космодрома „Восточный“ на околокруговую солнечно-синхронную орбиту с наклоном  $97,5^\circ$  и высотой 550 км. На рис. 1 показана ориентация наноспутника SamSat-ION в штатном режиме функционирования.

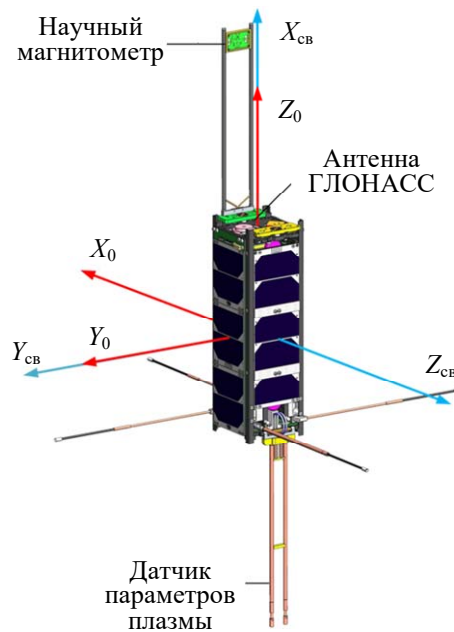


Рис. 1

Основная цель SamSat-ION — исследование верхних слоев ионосферы Земли дистанционным и контактными методами, для чего в качестве полезной нагрузки на наноспутнике установлены датчик параметров плазмы [6] и приемник ГЛОНАСС разработки Самарского университета [7, 8]. На SamSat-ION предусмотрено проведение ряда экспериментов: по спутниковой навигации, по отработке алгоритмов трехосевой ориентации [9—11] и раскрытию трансформируемых конструкций [12]. Задачи, стоящие перед SamSat-ION, требуют отказоустойчивости его бортовых систем и программного обеспечения.

**Обзор существующих подходов к обеспечению отказоустойчивости.** При проектировании отказоустойчивой системы разработчики спутников, как правило, формируют дерево ее отказов с учетом предполагаемых типов отказов, которые необходимо устранить [13].

Отказоустойчивость — это способность системы продолжать выполнение намеченных функций при наличии отказа [14]. В широком смысле отказоустойчивость связана с надежностью, успешной работой и отсутствием поломок. Отказоустойчивая система должна обладать способностью обрабатывать сбои в отдельных аппаратных или программных компонентах, сбое электропитания и т.п., соответствуя при этом спецификациям [15].

На борту больших космических аппаратов широко применяется подход к достижению отказоустойчивости за счет аппаратных средств с использованием, например, тройного модульного резервирования [15]. Этот подход связан со значительными накладными расходами [16], что осложняет его применение на наноспутниках. Вследствие массогабаритных и энергетических ограничений разработки далеко не все наноспутники являются отказоустойчивыми [17], порядка 43 % запускаемых в настоящее время наноспутников не используют резервирование [18]. Тем не менее выделяется ряд миссий наноспутников, использующих резервирование [19—22].

Существует подход к обеспечению отказоустойчивости за счет распределения управления на нескольких узлах [23, 24]. Под распределенной системой управления в общем случае понимается любая вычислительная система, включающая несколько удаленно расположенных процессоров, где каждый процессор выполняет определенную функцию в задачах вычисления или управления.

Выделяются три подхода к поддержанию отказоустойчивости в распределенной системе управления. При реализации первого подхода [25—27] для выполнения набора задач используются несколько резервных процессоров. При этом по режиму работы резервирование является ненагруженным (холодный резерв), после отказа основного процессора резервный принимает на себя его функции. При втором подходе [28, 29] используется аппаратное резервирование и репликация программного обеспечения. Данный режим работы контроллеров является нагруженным (горячий резерв), при котором два или более контроллера выполняют одни и те же задачи. В каждом из указанных подходов надежность обеспечивается за счет аппаратного резервирования, которое применяется к каждому отдельному процессору. Следовательно, если один процессор выходит из строя, его избыточный резервный узел берет на себя управление, однако если и этот узел выходит из строя, то другого заменяющего нет. В этих обстоятельствах космический аппарат должен работать с ограниченной функциональностью в безопасном режиме. Третий подход [30—33] направлен на решение вышеуказанных проблем путем внедрения распределенного управления с использованием концепции FDIR (Fault Detection, Isolation and Restoration — локализация отказов, устранение и реконфигурация) и поддержки миграции задач в случае сбоя узла.

FDIR — это концепция, используемая в различных распределенных и взаимосвязанных системах для обеспечения их непрерывной работы при наличии отказов или ошибок в работе космических аппаратов [33—39]. В этих системах часто используется подход „ведущий/ведомый“, когда имеются один ведущий узел и один или несколько ведомых, которые взаимодействуют с ведущим [39]. Ведущий узел отвечает за принятие решений, распределение задач и управление подчиненными, ведомые узлы выполняют определенные задачи и общаются о статусе ведущему.

В работе [33] представлена распределенная архитектура бортового компьютера, использующая подход „ведущий/ведомый“ FDIR. Данная архитектура состоит из главного узла, узла-наблюдателя и рабочих узлов, соединенных друг с другом через сеть SpaceWire (технология передачи данных, широко используемая в современных космических миссиях, включая спутники дистанционного зондирования Земли [40], космические телескопы [41] и марсоходы [42]). Эта схема поддерживает концепцию, согласно которой задача может выполняться на разных целевых объектах.

Обобщенная классификация аппаратного резервирования представлена в работах [43, 44]. По видам аппаратное резервирование подразделяется на структурное, временное, информационное, нагрузочное, функциональное. Структурное резервирование представляет собой минимальное

изменение структуры объекта (при отказе части объекта функционал сохраняется); временное резервирование заключается в проектировании объекта для работы на промежутках времени, превышающих минимально необходимое; информационное резервирование заключается в использовании избыточности информации для устранения ее искажения; функциональное резервирование — это замещение функции одного метода другим, когда достижение результата первым методом становится невозможным при возникновении нештатной ситуации.

Помимо перечисленных подходов, повышение отказоустойчивости наноспутников достигается за счет использования сторожевых таймеров [45, 46] и методов защиты данных. В системах электропитания сторожевые таймеры следят за исправностью источника питания, колебаниями напряжения, перегрузками [47], а в приемниках и передатчиках непрерывно отслеживают состояние связи и выявляют проблемы, связанные с зависанием и отсутствием связи [48]. При обнаружении отказов таймеры запускают процедуры восстановления либо сброса для предотвращения нештатных ситуаций. Также проводится анализ отчетов об ошибках, проверяются важные параметры, такие как энергопотребление и температура, для обнаружения нештатного поведения и при необходимости отправляются соответствующие команды из центра управления полетом [19, 46].

Следует отметить также программные методы повышения отказоустойчивости, которые не распространены на наноспутниках, но широко применяются в других приложениях. Первый часто используемый метод заключается в создании нескольких копий задач [49—51], которые могут выполняться одновременно (активная репликация) или поочередно (пассивная репликация). Другой подход основан на перепланировании задач (если на них уже повлияла ошибка [52, 53]) или замене задач (оценивается вероятность возникновения ошибки и определяется, может ли замена двух задач снизить частоту „мягких“ ошибок [54] — временную неразрушающую ошибку, возникающую в результате временного нарушения электрической или электромагнитной среды [15, 55]). Также выделяется метод использования контрольных точек [31, 56], в котором регулярно сохраняется состояние выполнения, и в случае отказа выполнения задачи осуществляется перезапуск с последней контрольной точки. Подробный обзор методов повышения надежности программными средствами приведен в [57].

В [15] описан ряд показателей отказоустойчивости, например, таких, как частота возникновения отказа — ожидаемое количество отказов в единицу времени, среднее время до отказа — ожидаемое время возникновения первого системного сбоя, среднее время, необходимое для восстановления системы. Используя данные показатели в совокупности с математическим аппаратом цепей Маркова [58], возможно описать эволюцию надежности системы.

**Постановка задачи.** В общем случае для базового функционирования наноспутника необходимо наличие системы электропитания (СЭП), обеспечивающей энергией все системы наноспутника; бортового компьютера (БК), отвечающего за выполнение циклограммы полета, обработку данных и команд; системы связи, состоящей из приемника (Пр) и передатчика (Прд), необходимых для приема команд и передачи телеметрии.

Проведение научных и технологических экспериментов на наноспутнике SamSat-ION требует использования таких систем, как система управления движением (СУД), которая необходима для приведения SamSat-ION в гравитационное положение равновесия (см. рис. 1) для измерений электронной концентрации ионосферы с помощью датчика параметров плазмы; навигационная система (НС), служащая как для координатно-временной привязки научной информации, так и для изучения ионосферы методами радиопросвечивания; система раскрытия трансформируемых конструкций (СР). Система раскрытия трансформируемых конструкций состоит из двух плат: первая управляет отведением замковых плит, удерживающих четыре резонатора антенны и резонатор датчика параметров плазмы; вторая плата — штанги выносного магнитометра. От успешной работы этой системы зависит как установление ра-

диосвязи, так и проведение основной части научных экспериментов, соответственно данная система требует повышенной отказоустойчивости.

Для решения научных и технологических задач, стоящих перед наноспутником SamSat-ION, необходимо функционирование основных бортовых систем в течение всего срока активного существования. Поэтому архитектура распределенной системы управления наноспутником SamSat-ION должна быть устойчивой к отказам.

**Архитектура отказоустойчивой распределенной системы управления наноспутником SamSat-ION.** Отказоустойчивость распределенной системы управления наноспутником SamSat-ION достигается как аппаратными, так и программными средствами.

К аппаратным средствам повышения отказоустойчивости наноспутника SamSat-ION относится дублирование микроконтроллера (МК) на платах системы раскрытия, являющейся одной из критически важных для успеха миссии. На рис. 2 приведена временная диаграмма этапа раскрытия. Алгоритм раскрытия трансформируемых конструкций выполняется сначала на первом МК платы с двумя замками, затем с 5-минутной задержкой повторяется на втором МК, после чего с 5-минутным шагом обрабатывают МК на плате с одним замком.

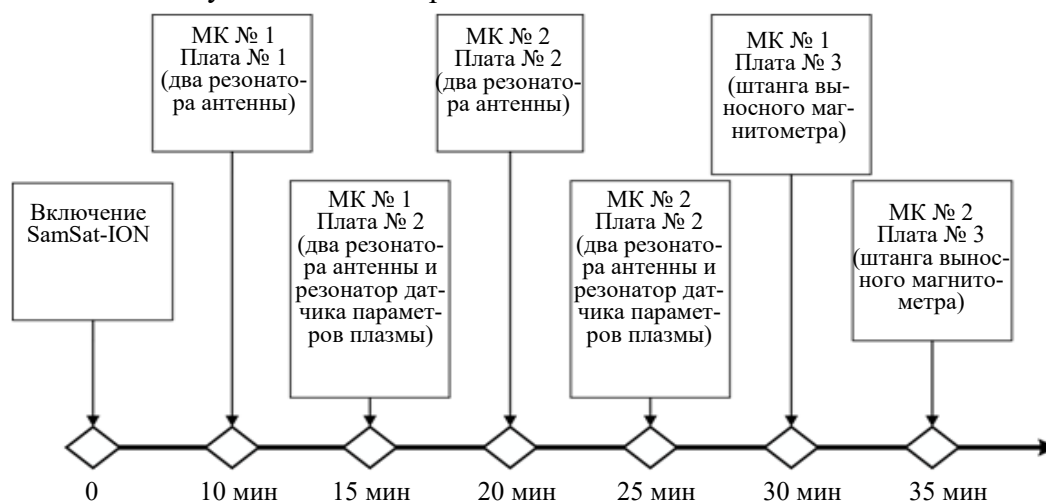


Рис. 2

БК наноспутника SamSat-ION оснащен двумя МК — двухъядерным с асимметричной архитектурой ядер LPC4357 (ведущее ядро Cortex-M4F и ведомое ядро Cortex-M0) и энергоэффективным ATxmega128U4. Выбранная конфигурация позволяет разделять задачи между МК в штатном режиме функционирования: так, менее производительный МК ATxmega1284U используется для взаимодействия с бортовыми системами, тогда как двухъядерный МК LPC4357 отвечает за выполнение циклограммы полета.

Помимо этого, на наноспутнике SamSat-ION реализован подход „ведущий/ведомый“ — в каждой бортовой системе используются равнозначные МК, подключенные к общей шине. Каждый из МК может выполнять роль ведущего на шине и обращаться напрямую к другим системам. Данное решение позволяет реализовать управление SamSat-ION не только через БК, но и через другие системы.

Так, штатное управление из центра управления полетами наноспутников реализуется двумя способами. В рамках первого способа команды становятся в очередь на Приемник, при этом МК LPC4357 на БК является ведущим и производит их последовательное чтение и исполнение; в рамках второго способа Приемник является ведущим на шине и производит прямое обращение к подсистеме, после чего отправляет результат исполнения команды через Передатчик в центр управления полетом.

Структура взаимодействия бортовых систем SamSat-ION, представленная на рис. 3, схематично разделена на три вертикальные колонки. В первой колонке расположены блоки наземного программного обеспечения (ПО) — операционного и для проведения тестирования,

во второй колонке — ведущие системы (Пр и два МК АТхmega128U4 и NXP LPC4357 на БК), которые могут отправлять команды на подсистемы, расположенные в третьей колонке, и получать с них данные. В третьей колонке расположены МК системы раскрытия выносных элементов спутника СР1—4 (по два на каждом торце наноспутника), СУД, Прд, СЭП и вычислитель навигационного приемника (Нав) ZYNQ.

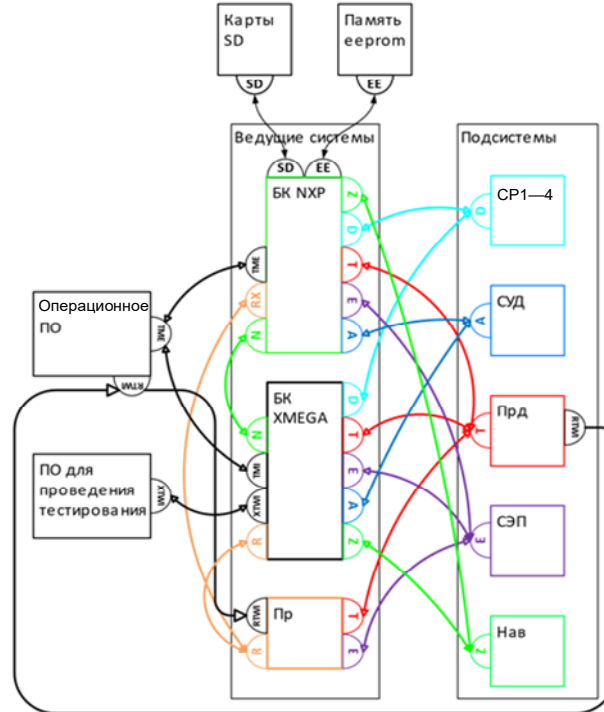


Рис. 3

Блок „Операционное ПО“ отвечает за радиосвязь со спутником и поддерживает два транспортных протокола: RTWI (Radio Two Wire Interface) и TME (TeleMEtry). RTWI — виртуальный двухпроводной интерфейс, работающий по радиоканалу, который обеспечивает доступ к следующим подсистемам: 1) Передатчику, СЭП и БК АТхmega128U4 напрямую через Приемник; 2) вычислителю навигационного приемника ZYNQ, СЭП, СУД, БК NXP LPC4357, СР1—4 через Приемник и БК АТхmega128U4. TME — протокол типа запрос-ответ, обеспечивающий взаимодействие с МК БК спутника LPC4357 через набор телеметрических команд управления подсистемами, а также доступ к записям научной и телеметрической информации на microSDHC-картах.

Блок „ПО для проведения тестирования“ отвечает за наземную отладку бортовых систем спутника и использует проводной протокол XTWI (Xmega Two Wire Interface), который является внешней надстройкой над протоколом UART.

Для повышения отказоустойчивости наноспутника SamSat-ION программными методами используются сторожевые таймеры на СЭП и Приемнике. Так, в случае программного сбоя на контроллере СЭП или отсутствия обращения от ведущих систем в течение 12 ч производится перезагрузка питания SamSat-ION. В случае отсутствия связи в течение 24 ч Приемник передает команду СЭП на перезагрузку борта.

Для уменьшения возможных ошибок на стадии разработки в программном обеспечении SamSat-ION используются унифицированные файлы конфигурации, каждый из которых является программным интерфейсом, содержащим список команд, формат запросов и ответов, необходимых для взаимодействия с бортовой системой. Протоколы RTWI и XTWI получают доступ к системам через эти интерфейсы (обозначены буквами R, N, Z, D, T, E, A на рис. 3).

В архитектуре программного обеспечения БК наноспутника SamSat-ION воплощен модульный подход, который представлен на рис. 4.

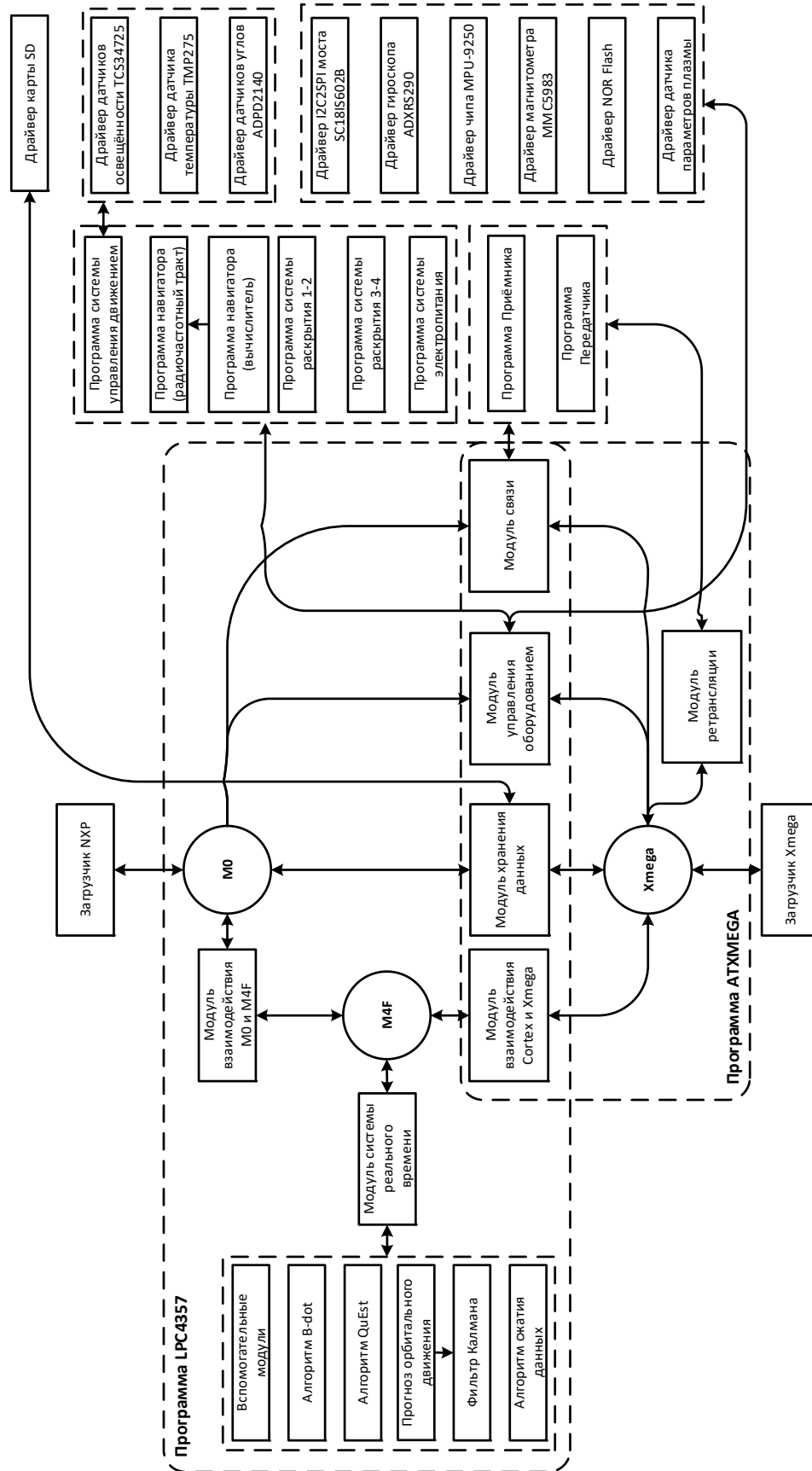


Рис. 4



На МК LPC4357 энергоэффективное ведомое ядро Cortex-M0 осуществляет взаимодействие с другими подсистемами спутника, опрос датчиков и запись полученных данных на носитель информации microSDHC. Работа модулей хранения данных, управления оборудованием и связи не требует операций с плавающей точкой, что позволило использовать их на ведомом ядре Cortex-M0 МК LPC4357 и ATxmega128U4. Это позволяет динамически перенести задачи с Cortex-M0 на ATxmega128U4 в случае прекращения работы МК LPC4357. Ведущее ядро Cortex-M4F отвечает за выполнение алгоритмов, требующих вычислений с плавающей точкой, таких как алгоритм стабилизации движения наноспутника и определения его ориентации, а также ресурсозатратного алгоритма сжатия данных.

Поскольку ядро Cortex-M0 МК LPC4357 и МК ATxmega128U4 способны выполнять одинаковый базовый функционал, то их работу возможно описать в качестве цепи Маркова двухкомпонентной параллельной системы. Эволюция надежности ( $R(t)$ ) SamSat-ION схематично представлена на рис. 5 по сравнению с использованием только одного МК (здесь  $\lambda$  — частота возникновения отказа для МК ATxmega128U4). При параллельном подключении улучшаются такие показатели, как среднее время между отказами и среднее время восстановления системы.

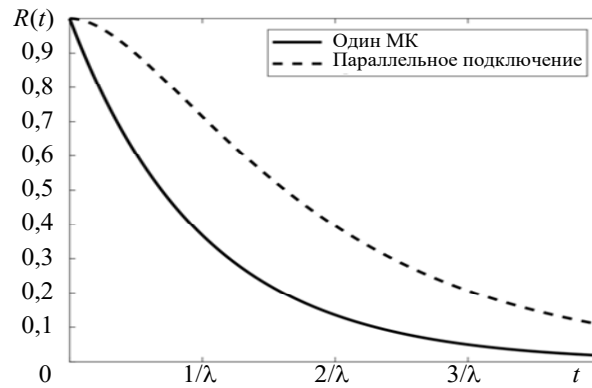


Рис. 5

Многоядерная архитектура МК обеспечивает безопасное перепрограммирование бортового компьютера. Передача программы осуществляется по радиоканалу с сохранением на карту памяти, что позволяет передать ее за несколько сеансов связи, производя при этом проверку целостности с помощью алгоритмов контрольных сумм. После проверки по команде из центра управления полетами начинается перепрограммирование.

**Обсуждение результатов.** На наноспутнике SamSat-ION реализован подход „ведущий/ведомый“ с динамическим переносом задач, благодаря которому отказоустойчивость была повышена без использования контроллеров нагруженного и ненагруженного резерва работы. Это позволило избежать массогабаритных и энергетических накладных расходов за счет введения дополнительных бортовых компьютеров. Однако использование распределенной системы потребовало более сложной архитектуры программного обеспечения.

Бортовые системы и бортовое программное обеспечение наноспутника SamSat-ION прошли летную квалификацию в ходе запуска с космодрома „Восточный“ 27 июня 2023 г. Управление наноспутником проводилось через приемопередатчик и контроллер ATxmega128U4.

На рис. 6 (на основе полетных данных) приведен пример реализации процедуры принудительного снижения энергопотребления при функционировании SamSat-ION. Так, с 9 августа 2023 г. по 12 августа 2023 г. наблюдалось снижение заряда аккумуляторных батарей (С) с 95 до 65 % (рис. 6, а; на рис. 6, б показан график тока канала питания 3,3 В). Было принято решение отключить лишних потребителей и 12 августа 2023 г. были отключены МК LPC4357, а также МК системы Раск и СУД. МК ATxmega128U4 выполнялись задачи по управлению — сбор телеметрической информации, ведение архива и об-



работка телеметрических команд. После отключения генерация энергии превысила потребление и SamSat-ION начал заряжаться. По достижении 95 % МК и системы были снова включены. Задачи по ведению архива телеметрической информации были возвращены на ядро Cortex-M0 МК LPC4357. Данный пример демонстрирует возможность гибкой подстройки в управлении наноспутником SamSat-ION и эквивалентен ситуации отказа одного из контроллеров БК.

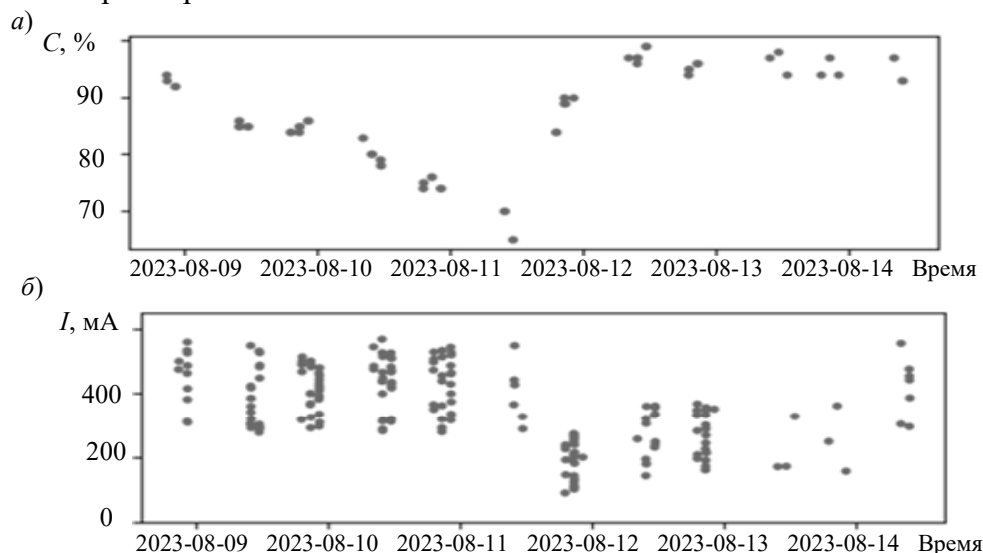


Рис. 6

**Заключение.** Принятые аппаратные и программные решения по наноспутнику SamSat-ION позволяют гибко подстраивать систему управления в случае возникновения отказов. Данный опыт может быть полезен разработчикам наноспутников.

#### СПИСОК ЛИТЕРАТУРЫ

1. CubeSat Handbook. From Mission Design to Operations / Ed: C. Cappelletti, S. Battistini, B. K. Malphrus. Academic Press, 2021. P. 199—219.
2. Lumbwe L. T. Development of an onboard computer (OBC) for a CubeSat / Cape Peninsula University of Technology. 2013.
3. Roberto C., Gianluca G., Gionata B., Christian C., Riccardo M. GPU@SAT: A General-Purpose Programmable Accelerator for on Board Data Processing and Satellite Autonomy // Studies in Computational Intelligence. 2022. Vol. 1088. P. 35—48. DOI: 10.1007/978-3-031-25755-1\_3.
4. Shiyu W., Shengbing Z., Jihe W., Xiaoping H. Towards Energy Efficient Architecture for Spaceborne Neural Networks Computation // Algorithms and Architectures for Parallel Processing: 20th Intern. Conf., ICA3PP 2020, NY, USA, Oct. 2—4, 2020, Proc., Part II. P. 575—586. DOI: 10.1007/s12567-020-00321-9.
5. Saurabh M. R., Abhishek G., Shubham S., Kushagra A., Dhananjay M., Tanuj K. Development of On Board Computer for a Nanosatellite // 68th Intern. Astronautical Congress (IAC), Adelaide, Australia. 2017.
6. Галка А. Г., Костров А. В., Мальшев М. С. Резонансный метод измерения концентрации ионосферной плазмы на микроспутниках // ЖТФ. 2023. Т. 93, № 1. С. 81.
7. Kramlikh A. V., Lomaka I. A., Shafran S. V. Estimation Method for Nanosatellite Orbital Parameters in Case of Abnormal Operation of Navigation Equipment // 27th Saint Petersburg Intern. Conf. on Integrated Navigation Systems, ICINS 2020: Proc. 2020.
8. Nikolaev P. N., Kudryavtsev I. A., Shafran S. V. Requirements for nanosatellite-mounted GNSS-based instrument measuring ionospheric total electron content // IOP Conf. Ser.: Materials Science and Engineering. 2020. Vol. 984, № 1. P. 012022.
9. Leonov A. I., Nikolaev P. N. Providing Nanosatellite Triaxial Gravitational Orientation Using Magnetic Actuators // IOP Conf. Ser.: Materials Science and Engineering. 2022. Vol. 1215, N 1. P. 012005.

10. *Kramlikh A. V., Lomaka I. A., Nikolaev P. N.* Damping control system design for SamSat nanosatellite platform // IOP Conf. Ser.: Materials Science and Engineering. 2020. Vol. 862.
11. *Kramlikh A., Nikolaev P., Rylko D. V.* Implementation Features of Attitude Determination Algorithm for the SamSat-ION Nanosatellite // 29th Saint Petersburg Intern. Conf. on Integrated Navigation Systems, ICINS 2022: Proc. 2022.
12. *Lomaka I. A., Elisov N. A., Boltov E. A.* et al. A novel design of CubeSat deployment system for transformable structures // *Acta Astronautica*. 2022. Vol. 197. P. 179—190.
13. *Thambidurai P., You-keun P.* Interactive consistency with multiple failure modes // Proc. of the 7th Symp. on Reliable Distributed Systems. 1988. P. 93—100. DOI: 10.1109/reldis.1988.25784.
14. *Johnson B.* Fault-Tolerant Microprocessor-Based Systems // *IEEE Micro*. 1984. Vol. 4, N 6. P. 6—21. DOI: 10.1109/MM.1984.291277
15. *Dubrova E.* Fault-Tolerant Design. N. Y.: Springer, 2013. DOI: 10.1007/978-1-4614-2113-9.
16. *Velazco R., McMorro D., Estela J.* Radiation Effects on Integrated Circuits and Systems for Space Applications. Springer Intern. Publ., 2019. DOI: 10.1007/978-3-030-04660-6.
17. *Erlank A. O., Bridges C. P.* Satellite stem cells: The benefits & overheads of reliable, multicellular architectures // 2017 IEEE Aerospace Conf., 2017. P. 1—12. DOI: 10.1109/AERO.2017.7943732.
18. *Erlank A., Bridges C.* Reliability analysis of multicellular system architectures for low-cost satellites // *Acta Astronautica*. 2018. Vol. 147. P. 183—194. DOI: 10.1016/j.actaastro.2018.04.006.
19. *Laizans K.* et al. Design of the fault tolerant command and data handling subsystem for estcube-1 // Proc. of the Estonian Academy of Sciences. 2014. P. 222—231. DOI: 10.3176/proc.2014.2S.03.
20. *Praks J.* et al. Aalto-1, multi-payload CubeSat: Design, integration and launch // *Acta Astronautica*. 2021. Vol. 187. P. 370—383. DOI:10.1016/j.actaastro.2020.11.042.
21. *Samsuzzaman Md.* et al. BIRDS-1 CubeSat Constellation Using Compact UHF Patch Antenna // *IEEE Access*. 2018. Vol. 6. P. 54282—54294. DOI:10.1109/ACCESS.2018.2871209.
22. *Duarte R. O.* et al. Development of an Autonomous Redundant Attitude Determination System for Cubesats // *J. Aerosp. Techn. Manag.* 2020. N 12. DOI: 10.5028/jatm.v12.1166.
23. *Bostan V., Martiniuc A., Secieru N., Vărzaru V., Melnic V., Ilco V.* The evaluation of the on-board computer architecture for TUMnanoSAT series of nanosatellites for carrying out missions // 12th Intern. Conf. on Electronics, Communications and Computing, Moldova. 2022.
24. *Fuchs C. M., Chou P., Wen X.* A Fault-Tolerant MPSoC For CubeSats // 32nd IEEE Intern. Symp. on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT). 2019.
25. *Lala J. H., Harper R. E., Jaskowiak K. R.* et al. Advanced information processing system (AIPS)-based fault tolerant avionics architecture for launch vehicles // Proc. IEEE/AIAA/NASA 9th Digital Avionics Systems Conf. 1990. P. 125—132.
26. *Rennels D. A.* Architectures for fault-tolerant spacecraft computers // Proc. IEEE. 1978. Vol. 66, P. 1255—1268.
27. *Chau S. N., Alkalai L., Tai A. T.* et al. Design of a fault-tolerant COTS-based bus architecture // *IEEE Trans. Reliab.* 48. P. 351—359.
28. *Kopetz H., Damm A., Koza C.* et al. Distributed fault-tolerant real-time systems: the Mars approach // *IEEE Micro* 9. 1989. P. 25—40.
29. *Powell D.* Distributed fault tolerance: lessons from Delta-4 // *IEEE Micro* 14. 1994. P. 36—47.
30. *Fayyaz M., Vladimirova T.* Fault-tolerant distributed approach to satellite on-board computer design // Proc. IEEE Aerospace Conf. 2014. P. 1—12.
31. *Fayyaz M., Vladimirova T., Caujolle J. M.* Adaptive middleware design for satellite fault-tolerant distributed computing // Proc. IEEE NASA/ESA Adaptive Hardware and Systems (AHS) Conf. 2012. P. 23—30.
32. *Vladimirova T., Fayyaz M.* Wireless fault-tolerant distributed architecture for satellite platform computing // *Convergence and Hybrid Information Technology*. Springer, 2012. P. 428—436.
33. *Ludtke D., Westerdorff, Stohlmann K.* et al. OBC-NG: towards a reconfigurable on-board computing architecture for spacecraft // *Aerospace Conf.* 2014. P. 1—13.

34. *Eickhoff J.* Onboard Computers, Onboard Software and Satellite Operations. Springer-Berlin-Heidelberg, 2012.
35. *Jalilian S.* et al. Fault detection, isolation and recovery (FDIR) in satellite onboard software // Program Mühendisliyyinin Aktual Elmi-Praktiki Problemləri I Respublika Konfransinin Materialları. 2017.
36. *Nannipieri P.* et al. A serial high-speed satellite communication CODEC: Design and implementation of a SpaceFibre interface // Acta Astronautica. 2020. Vol. 169. P. 206—215.
37. *Rathsman P.* et al. SMART-1: Development and lessons learnt // Acta Astronautica. 2005. Vol. 57, N 2—8. P. 455—468.
38. *Durou O.* et al. Hierarchical fault detection, isolation and recovery applied to cof and atv avionics // Acta Astronautica. 2002. Vol. 50, N 9. P. 547—556.
39. *Olive X.* FDI(R) for satellites: How to deal with high availability and robustness in the space domain? // Intern. Journal of Applied Mathematics and Computer Science. 2012. Vol. 22, N 1. P. 99—107.
40. *Girard A., Provost A., Nodet J., Desmet P., Cossard P.* Overview of Implementing SpaceWire in Observation Satellites from Thales Alenia Space // Proc. of the Intern. SpaceWire Conf., St. Petersburg, Russia. 2010.
41. *Rakow G. P., Schmirr R., Dailey C. L., Shakoorzadeh K.* Reliable Transport Over SpaceWire for James Webb Space Telescope // 2003 IEEE Aerospace Conf. Proc. 2003.
42. *Josset J. L.* et al. The Close-Up Imager Onboard the ESA ExoMars Rover: Objectives, Description, Operations, and Science Validation Activities // Astrobiology. 2017. Vol. 17, N 6—7. P. 595—611.
43. *Макаров А. П., Соловьев В. А.* Анализ автоматизированных методов расчета показателей надежности радиоэлектронной аппаратуры // НиКа: Тр. Междунар. симп. „Надежность и качество“. 2018.
44. *Жаднов В. В., Лазарев Д. В.* Структурно-резервированные технические системы и их показатели надежности // НиКа: Тр. Междунар. симп. „Надежность и качество“. 2005.
45. PW-SAT 2 preliminary requirements review: On-board computer / Warsaw University of Technology. 2014.
46. *Chen L. W., Huang T. C., Juang J. C.* Implementation of the fault tolerance module in PHOENIX CubeSat // Presentation at 10th IAA Symp. on Small Satellites for Earth Observation. 2015.
47. *Selčan D., Kirbiš G., Kramberger I.* Nanosatellites in LEO and beyond: Advanced Radiation protection techniques for COTS-based spacecraft // Acta Astronautica. 2017. Vol. 131. P. 131—144.
48. *Yang M.* et al. Space Flight Validation of Design and Engineering of the ZDPS-1A Pico-satellite // Chinese Journal of Aeronautics. 2012. Vol. 25, N 5. P. 725—738.
49. *Ghosh S., Melhem R., Mosse D.* Fault-tolerance through scheduling of aperiodic tasks in hard real-time multiprocessor systems // IEEE Trans. Parallel Distrib. Syst. 1997. Vol. 8. P. 272—284. DOI: 10.1109/71.584093.
50. *Wang S.* et al. A reliability-aware task scheduling algorithm based on replication on heterogeneous computing systems // Journal of Grid Computing. 2017. Vol. 15 P. 23—39. DOI: 10.1007/s10723-016-9386-7.
51. *Haque M. A., Aydin H., Zhu D.* On reliability management of energy-aware realtime systems through task replication // IEEE Trans. Parallel Distrib. Syst. 2017. Vol. 28, N 3. P. 813—825. DOI: 10.1109/TPDS.2016.2600595.
52. *Mei J.* et al. Fault-tolerant dynamic rescheduling for heterogeneous computing systems // Journal of Grid Computing. 2015. Vol. 13, N 4. P. 507—525. DOI: 10.1007/s10723-015-9331-1.
53. *Hasan M., Goraya M. S.* A framework for priority based task execution in the distributed computing environment // Intern. Conf. on Signal Processing, Computing and Control (ISPCC), 2015. P. 155—158. DOI: 10.1109/ISPCC.2015.7375016.
54. *Naithani A., Eyerhan S., Eeckhout L.* Optimizing soft error reliability through scheduling on heterogeneous multicore processors // IEEE Trans. Comput. 2018. Vol. 67, N 6. P. 830—846. DOI: 10.1109/TC.2017.2779480.
55. *Baumann R.C.* Radiation-induced soft errors in advanced semiconductor technologies // IEEE Trans. on Device and Materials Reliability. 2005. Vol. 5, N. 3. P. 305—316.
56. *Singh M.* Performance analysis of checkpoint based efficient failure-aware scheduling algorithm // Intern. Conf. on Computing, Communication and Automation (ICCCA), 2017. P. 859—863. DOI: 10.1109/CCTA.2017.8229916.
57. *Goloubeva O., Rebaudengo M., Reorda M. S., Violante M.* Software-Implemented Hardware Fault Tolerance. Springer, 2006. DOI: 10.1007/0-387-32937-4.

58. Norris J. R. Markov Chains. N. Y.: Cambridge Univ. Press, 1998.

#### Сведения об авторах

- Вадим Дмитриевич Мещеряков** — Самарский национальный исследовательский университет им. акад. С. П. Королева, межвузовская кафедра космических исследований; инженер-программист; E-mail: meshcheryakov.vd@ssau.ru
- Петр Николаевич Николаев** — канд. физ.-мат. наук; Самарский национальный исследовательский университет им. акад. С. П. Королева, межвузовская кафедра космических исследований; Институт космических исследований РАН; ст. научный сотрудник; E-mail: nikolaev.pn@ssau.ru
- Степан Вячеславович Шафран** — Самарский национальный исследовательский университет им. акад. С. П. Королева, межвузовская кафедра космических исследований; Институт космических исследований РАН; мл. научный сотрудник; E-mail: shafran.sv@ssau.ru
- Ангело Сальватор Эспиноза Валлес** — Самарский национальный исследовательский университет им. акад. С. П. Королева, межвузовская кафедра космических исследований; инженер-программист; E-mail: espinoza\_valles.a@ssau.ru

Поступила в редакцию 11.09.2023; одобрена после рецензирования 25.10.2023; принята к публикации 14.11.2023.

#### REFERENCES

- Cappelletti C., Battistini S., Malphrus B.K., eds., *CubeSat Handbook. From Mission Design to Operations*, 2021, pp. 10–17, 199–219.
- Lumbwe L.T. *Development of an Onboard Computer (OBC) for a CubeSat*, Cape Peninsula University of Technology, 2013, pp. 1–3, 14–18.
- Roberto C., Gianluca G., Gionata B., Christian C., Riccardo M. *Studies in Computational Intelligence*, 2022, vol. 1088, pp. 35–48, DOI: 10.1007/978-3-031-25755-1\_3.
- Shiyu W., Shengbing Z., Jihe W., Xiaoping H. *Algorithms and Architectures for Parallel Processing 20th International Conference, ICA3PP 2020, NY, USA, October 2–4, 2020, Pt. II*, pp. 575–586, DOI: 10.1007/s12567-020-00321-9.
- Saurabh M.R., Abhishek G., Shubham S., Kushagra A., Dhananjay M., Tanuj K. *68th Intern. Astronautical Congress (IAC)*, Adelaide, Australia, 2017.
- Galka A.G., Kostrov A.V., Malyshev M.S. *Technical Physics*, 2023, no. 1(93), pp. 81, DOI: 10.21883/TP.2023.01.55442.192-22.
- Kramlikh A.V., Lomaka I.A., Shafran S.V. *27th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2020*, Proceedings, Institute of Electrical and Electronics Engineers Inc., 2020.
- Nikolaev P.N., Kudryavtsev I.A., Shafran S.V. *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2020, no. 1(984), pp. 012022.
- Leonov A.I., Nikolaev P.N. *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2022, no. 1(1215), pp. 012005.
- Kramlikh A.V., Lomaka I.A., Nikolaev P.N. *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 862.
- Kramlikh A., Nikolaev P., Rylko D.V. *29th Saint Petersburg International Conference on Integrated Navigation Systems, ICINS 2022*, 2022.
- Lomaka I.A., Elisov N.A., Boltov E.A. et al. *Acta Astronautica*, 2022, vol. 197, pp. 179–190.
- Thambidurai P., You-keun P. *Proc. of 7th Symposium on Reliable Distributed systems*, 1988, pp. 93–100, DOI:10.1109/reldis.1988.25784.
- Johnson B. *IEEE Micro*, 1984, no. 6(4), pp. 6–21, DOI: 10.1109/MM.1984.291277.
- Dubrova E. *Fault-Tolerant Design*, NY, Springer, 2013, DOI:10.1007/978-1-4614-2113-9.
- Velazco R., McMorrow D., Estela J. *Radiation Effects on Integrated Circuits and Systems for Space Applications*, Springer International Publishing, 2019, DOI: 10.1007/978-3-030-04660-6.
- Erlank A.O., Bridges C.P. *2017 IEEE Aerospace Conference*, 2017, pp. 1–12, DOI:10.1109/AERO.2017.7943732.
- Erlank A., Bridges C. *Acta Astronautica*, 2018, vol. 147, pp. 183–194, DOI: 10.1016/j.actaastro.2018.04.006.
- Laizans K. et al. *Proc. of the Estonian Academy of Sciences*, 2014, pp. 222–231, DOI:10.3176/proc.2014.2S.03.
- Praks J. et al. *Acta Astronautica*, 2021, vol. 187, pp. 370–383, DOI:10.1016/j.actaastro.2020.11.042.
- Samsuzzaman Md. et al. *IEEE Access.*, 2018, vol. 6, pp. 54282–54294, DOI:10.1109/ACCESS.2018.2871209.
- Duarte R.O. et al. *J. Aerosp. Technol. Manag.*, 2020, no. 12, DOI:10.5028/jatm.v12.1166.
- Bostan V., Martiniuc A., Secieru N., Vărzaru V., Melnic V., Ilco V. *The 12th International Conference on Electronics, Communications and Computing*, Moldova, 2022.
- Fuchs C.M., Chou P., Wen X. *32nd IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2019.
- Lala J.H., Harper R.E., Jaskowiak K.R. et al. *Proc. IEEE/AIAA/NASA 9th Digital Avionics Systems Conf.*, 1990, pp. 125–132.
- Rennels D.A. *Proc. IEEE*, 1978, vol. 66, pp. 1255–1268.

27. Chau S.N., Alkalai L., Tai A.T. et al. *IEEE Trans.*, 1999, Reliab. 48, pp. 351–359.
28. Kopetz H., Damm A., Koza C. et al. *IEEE Micro* 9, 1989, pp. 25–40.
29. Powell D. *IEEE Micro* 14, 1994, pp. 36–47.
30. Fayyaz M., Vladimirova T. *Proc. IEEE Aerospace Conf.*, 2014, pp. 1–12.
31. Fayyaz M., Vladimirova T., Caujolle J.M. *Proc. IEEE NASA/ESA Adaptive Hardware and Systems (AHS) Conf.*, 2012, pp. 23–30.
32. Vladimirova T., Fayyaz M. *Convergence and Hybrid Information Technology*, Springer, 2012, pp. 428–436.
33. Ludtke D., Westerdorff., Stohmann K. et al. *Aerospace Conference*, 2014, pp. 1–13.
34. Eickhoff J. *Onboard Computers, Onboard Software and Satellite Operations*, Springer Berlin Heidelberg, 2012.
35. Jalilian S. et al. *Proqram mühendisliyinin aktual elmi-praktiki problemləri I respublika konfransının materialları*, 2017.
36. Nannipieri P. et al. *Acta Astronautica*, 2020, vol. 169, pp. 206–215.
37. Rathsmann P. et al. *Acta Astronautica*, 2005, no. 2–8(57), pp. 455–468.
38. Durou O. et al. *Acta Astronautica*, 2002, no. 9(50), pp. 547–556.
39. Olive X. *International Journal of Applied Mathematics and Computer Science*, 2012, no. 1(22), pp. 99–107.
40. Girard A., Provost A., Nodet J., Desmet P., Cossard P. *Proc. of the International SpaceWire Conference*, St. Petersburg, Russia, 2010.
41. Rakow G.P., Schmirr R., Dailey C.L., Shakoorzadeh K. *2003 IEEE Aerospace Conference Proceedings*, 2003.
42. Josset J.L. et al. *Astrobiology*, 2017, no. 6–7(17), pp. 595–611.
43. Makarov A.P., Solov'yev V.A. *Trudy Mezhdunarodnogo simpoziuma "Nadezhnost' i kachestvo"* (Proceedings of the International Symposium "Reliability and Quality"). 2018, vol. 1. (in Russ.)
44. Zhadnov V.V., Lazarev D.V. *Trudy Mezhdunarodnogo simpoziuma "Nadezhnost' i kachestvo"* (Proceedings of the International Symposium "Reliability and Quality"), 2005. (in Russ.)
45. *Warsaw University of Technology PW-SAT 2 Preliminary Requirements Review: On-board computer*, 2014.
46. Chen L.W., Huang T.C., Juang J.C. *Presentation at 10th IAA Symposium on Small Satellites for Earth Observation*, 2015.
47. Selčan D., Kirbiš G., Kramberger I. *Acta Astronautica*, 2017, vol. 131, pp. 131–144.
48. Yang M. et al. *Chinese Journal of Aeronautics*, 2012, no. 5(25), pp. 725–738.
49. Ghosh S., Melhem R., Mosse D. *IEEE Trans. Parallel Distrib. Syst.*, 1997, vol. 8, pp. 272–284, DOI: 10.1109/71.584093.
50. Wang S. et al. *J. Grid Comput.*, 2017, vol. 15, pp. 23–39, DOI: 10.1007/s10723-016-9386-7.
51. Haque M.A., Aydin H., Zhu D. *IEEE Trans. Parallel Distrib. Syst.*, 2017, no. 3(28), pp. 813–825, DOI: 10.1109/TPDS.2016.2600595.
52. Mei J. et al. *J. Grid Comput.*, 2015, no. 4(13), pp. 507–525, DOI: 10.1007/s10723-015-9331-1
53. Hasan M., Goraya M.S. *2015 International Conference on Signal Processing, Computing and Control (ISPCC)*, 2015, pp. 155–158, DOI: 10.1109/ISPCC.2015.7375016.
54. Naithani A., Eyerhan S., Eeckhout L. *IEEE Trans. Comput.*, 2018, no. 6(67), pp. 830–846, DOI: 10.1109/TC.2017.2779480.
55. Baumann R.C. *IEEE Transactions on Device and Materials Reliability*, 2005, no. 3(5), pp. 305–316.
56. Singh M. *International Conference on Computing, Communication and Automation (ICCCA)*, 2017, pp. 859–863, DOI: 10.1109/CICA.2017.8229916.
57. Goloubeva O., Rebaudengo M., Reorda M.S., Violante M. *Software-Implemented Hardware Fault Tolerance*, Springer, 2006, DOI: 10.1007/0-387-32937-4.
58. Norris J.R. *Markov Chains*, NY, Cambridge University Press, 1998.

**Data on authors**

- |  |   |
|--|---|
| <b>Vadim D. Meshcheryakov</b>          | — Samara University, Inter-University Department of Space Research; Software Engineer; E-mail: meshcheryakov.vd@ssau.ru                                     |
| <b>Petr N. Nikolaev</b>                | — PhD; Samara University, Inter-University Department of Space Research; Space Research Institute of the RAS; Senior Researcher; E-mail: nikolaev.pn@ssau.r |
| <b>Stepan V. Shafran</b>               | — Samara University, Inter-University Department of Space Research; Space Research Institute of the RAS; Junior Researcher; E-mail: shafran.sv@ssau.ru      |
| <b>Angelo Salvador Espinoza Valles</b> | — Samara University, Inter-University Department of Space Research; Software Engineer; E-mail: espinoza_valles.a@ssau.ru                                    |

Received 11.09.2023; approved after reviewing 25.10.2023; accepted for publication 14.11.2023.