Platform-Based Embedded Solution for Small Satellite's Onboard Computing

Olga Mamoutova, Alexander Fedotov, Alexey Filippov, Alexander Antonov
Peter the Great St. Petersburg State Polytechnic University
Saint-Petersburg, the Russian Federation
mamoutova@kspt.icc.spbstu.ru, afedotov@spbstu.ru, {filippov, antonov}@eda-lab.icc.spbstu.ru

Abstract—**Design of command and data handling subsystems for small satellites is in the focus for many in space industry and academia. For today a plenty of small satellites have been designed, manufactured and launched. However, while the application calls for the highly standardized platform, an overview of the projects shows a vast diversity of different hardware solutions on the market. The aim of this paper is to explore the possibilities of platform-based design for the task. A Universal Module for Managing the Information is presented to be a basic node for an onboard network that comprises a command and data handling subsystem of a spacecraft bus.**

I. Introduction

Command and Data Handling subsystem (C&DH) is a common name for a specialized embedded system of a spacecraft. C&DH system of any modern satellite is an onboard network of computers that manage all information received, processed, stored and transmitted by subsystems of the spacecraft, and control the interactions between those subsystems. Small satellites' C&DH inherit those traditional functions.

Since launch cost is significantly less than that of a traditional satellite the overall design of a small satellite is less demanding due to smaller cost of an error. However, while other requirements become loose, timeline for design remains strict due to the launch planning. This problem is widely recognized. For example, an Operationally Responsive Space program aims to have rapidly deployable capabilities for warfighter needs [1].

As a result, the key desired characteristics of a modern spacecraft bus are simplified testing and setup, as well as configurability during assembly and later during the flight. This calls for high levels of modularity and flexibility of units comprising the satellite bus.

One side of the problem is how to connect nodes of the onboard network in a simple, reliable way. A well established Plug-and-Play concept addresses this aspect.

The aim of the presented work is to address a hardware-design side of the problem. Our thorough review of the topic showed the lack of uniformity in how C&DH system is composed. Since its functions are physically distributed among the spacecraft, designers tend to create custom controllers for separate functions. A heterogeneous system of diverse physical devices is used for the payload controllers, central computer, mass memory, and other tasks.

We suggest utilizing the concept of a platform-based design, which is already a de-facto standard design methodology for embedded SoC systems. The key step of the design is to map a specified functionality onto a predesigned architecture. The advantage of this platform-based approach is in a reduced design space and huge reuse of both HW and SW parts of design. A predesigned, tested, verified and customizable platform considerably reduces design times and nonrecurring engineering costs [2].

The idea is to apply the platform concept to the higher, printed circuit board level: to use a unified, parameterized design of a C&DH module to be customized for the current payload and desired onboard network architecture of a particular project. Levels of customization include parameters of module's components, mixture of optional components and interfaces in the module, and HW/SW partitioning.

Universal Module for Managing the Information (UMoMI) has been blueprinted as a hardware platform for a small satellite bus. This paper presents the UMoMI of two types – an end point/host and a router of the onboard network. Following sections contain preliminary information about the topic, a description of the UMoMI platform, examples of UMoMI-based onboard network topologies and a use case for a typical small satellite.

II. Preliminaries

A. Small satellites

A modern satellite is generally called small if its mass is less than 500 kg (1102 lb). The satellite is often used for a small, specific task and can work as a part of a satellite constellation. Small satellites are characterized by a relatively small cost of production and launch and short design cycle due to active utilization of design practices

ISSN 2305-7254

that are not conventional for traditional space industry. This opens big opportunities literally to anybody interested in the space satellite design.

As a result, there is a plenty of architectures from both industry and academia. Industry typically has proprietary solutions in some form of a heritage from bigger satellites' designs. Examples are platforms from Surrey satellite technology (Great Britain), ATK (USA), Boeing (USA), Satrec Initiative (South Korea), etc. Academia produces plenty of designs mainly for smaller satellites in CubeSat form-factor [3], often as a first step into the space satellite design area. Also industry-academia cooperation lately generated several solutions, e.g., flying laptop project [4].

Based on open data from sites eoportal.org [5] and space.skyrocket.de [6], during the first six month of the 2014 year around a couple of dozens of small satellites has been successfully launched. Among those missions were satellites for Earth observation (50%), scientific experiments (25%), telecommunication (20%) and technology demonstration (20%). (Some satellites combine missions, usually technology demonstration being a secondary task.)

Because for a smaller spacecraft C&DH can usually be simplified to a single custom computer board, in our work we focus on small satellites with mass in range from 100 to 500 kg, which demand C&DH network.

B. Spacecraft bus

Spacecraft bus (or satellite bus) is a mandatory concept for any modern satellite. It is a unified model, providing standard payload and launch vehicle interfaces for a series of satellites. The spacecraft bus gives advantages for a series production: non-recurring engineering cost is distributed among the batch, reliability is improved due to tests over multiple design iterations, and time to market for satellites in a batch is significantly reduced.

A spacecraft bus is a framework of a satellite, which combines following subsystems:

- Power supply,
- Thermal control,
- Guidance and navigation (for small satellites – optional or simplified),
- Mechanical (for small satellites – usually optional),
- Communications,
- Telemetry, tracking and command,
- and Command and data handling.

Modern satellites tend to have those typical subsystems in a distributed form, where parts of different subsystems are merged. Accordingly, a C&DH subsystem is distributed as well and demands complex communications over the onboard network. The overview of C&DH subsystem's communications is presented in Fig. 1.

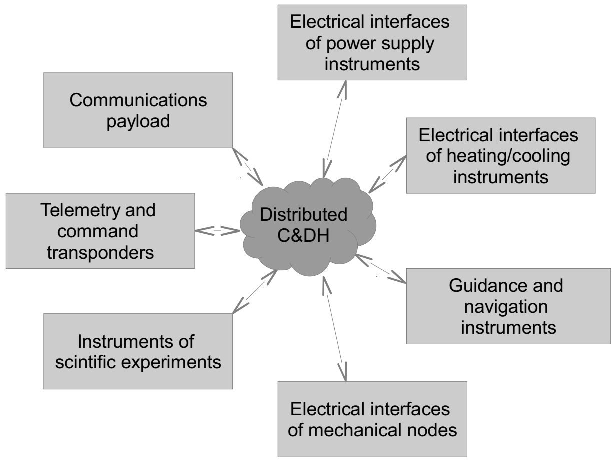

Fig. 1. C&DH environment

The current trend in design for small satellites is utilization of the Plug-and-Play (PnP) concept similar to one in conventional design of desktop computers. Corresponding standards were developed and successfully applied in real projects – Space Plug-and-Play Architecture (SPA) [7] and SpaceWire Plug-and-Play [8]. Those standards define protocols for smooth communication between devices and integration of a new device in the SpaceWire-based [9] C&DH network.

The presented concept of a hardware C&DH platform takes into account current trends in small satellites' design and focuses on the structural representation of the design. It abstracts from the physical view, but considers mapping of C&DH functionality to the structure.

III. UMoMI PLATFORM

We suggest a single parameterized platform for design of C&DH subsystem's components – UMoMI. The design methodology then focuses on the effective mapping of target mission functionality onto the network of unified modules. The resulting mapping thus defines the way of modules' customization: what optional physical components are used and what hardware and software functions are implemented.

The requirements for the platform are:

- Several degrees of programmability, including remote reprogrammability during the flight,
- Standard onboard network interface,
- Support of various onboard network configurations,
- Interfaces to various types of the payload and subsystems' specific devices,
- Several degrees of reliability implementation.

As the core technologies we selected: SpaceWire (SpW) as a de-facto standard media for spacecraft onboard networks and Field Programmable Gate Array (FPGA) as a main resource for reprogrammability.

The basic design of UMoMI is a computational node of a distributed C&DH network. While the goal is to have a single basic design of UMoMI, the analysis of C&DH subsystem's architectural demands showed that in order to cover all levels of complexity, system design requires additional type of device – the router UMoMI-R.

The rest of this section gives the overall look at the UMoMI platform. Technical details of implementation are intentionally omitted, as we present the platform as an example of a design methodology. For a particular design team those details will certainly differ due to technologies availability, experience of the team, heritage solutions, etc.

A. Basic UMoMI module

The main task of the UMoMI is to provide an interface to the sources of information from the payload or the satellite's subsystems, and to process and transmit the information to the desired destination – either directly connected to the UMoMI, or over the onboard network. This way the UMoMI should be capable to manage any C&DH function: front end data processing, buffering, data reduction, encoding/decoding, and digital signal processing (DSP) for science/payload telemetry, housekeeping telemetry, command and control, and satellite's service data.

Overview of the UMoMI's microarchitecture and an example of mapping of the C&DH functions are presented in Fig. 2. Optional components are outlined with the dotted lines. Some necessary non-C&DH components are also shown: thermal control, synchronization and power supply, which provide corresponding resources for all of the components.

The core of the module is a Service FPGA, which performs basic interface and control functions: input/output communication (I/O, optional), debug and setup support, onboard network architecture support, and configuration. Regarding onboard network support, it can also implement some control functions outside of the SpaceWire protocol stack, such as support of Plug-and-Play architecture and telecommand decoding (DC). A standard JTAG interface is used as a configuration interface for the test and setup stages of design and is not supposed to be used during mission flight. Fig. 2 encapsulates a configuration controller and a configuration memory in the Configuration block.

The second FPGA is optional and can duplicate functions of the Service FPGA or perform additional computational tasks in the form of programmable soft processor or hardware implementation of DSP algorithms. Both FPGA's SpaceWire interfaces implement dual-modular redundancy, which can be reconfigured for a double-throughput network connection.

The microcontroller (MCU) is also optional and can be programmed to perform computational or control tasks. For the sake of performance an additional high speed interface is used to connect MCU and FPGA. The Memory array is optional as well and can be used to support functions of MCU and FPGAs, or as an all-purpose data buffer.

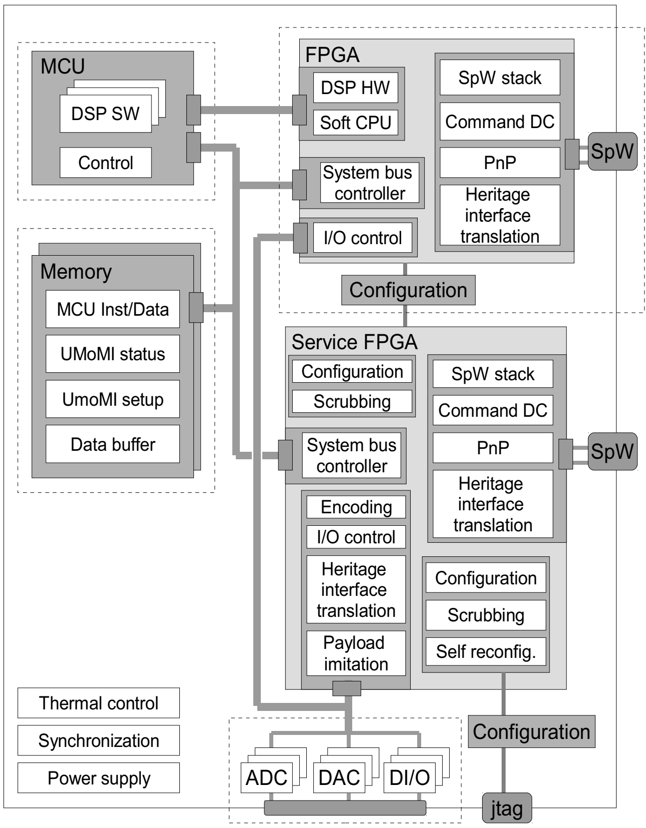

Fig. 2. Basic design of UMoMI

Depending on the selection of the components the UMoMI module can serve as an end-point data processor with high performance, as a control host of the onboard network, as a smart network controller for the payload, or can combine those roles. Another useful role is a heritage interface translator for the integration of a non-SpaceWire device into the SpaceWire network. In this case the heritage device is connected to the I/O interface of UMoMI and UMoMI translates device's protocol stack to the SpaceWire protocols and back.

Presented structure provides enough flexibility to map an arbitrary functionality with regards to performance and reliability requirements. For example, the array of analog and digital input/output components has structural redundancy that can be utilized either to have triple-modularity with voter implemented in the Service FPGA, or to have a high-throughput/high-precision data channel, when used in parallel.

The UMoMI platform provides several levels of reconfiguration:

- Plug-and-Play on the onboard network level during assembly,
- Automatic or commanded in-flight remapping of functions between UMoMIs: powering on and off redundant units, switching high-performance mode for similar units to high-redundancy or vice versa.
- Commanded on-orbit FPGA reconfiguration in order to adjust the functionality for changing mission's needs,
- Commanded on-orbit flight software update (also for changing mission's needs).

B. UMoMI-R module

SpaceWire links support only point-to-point communication. A simple onboard network (for example, with a ring topology) can be formed with basic UMoMI modules that implement duplicated SpaceWire interface. If the number of network devices rises, those types of network will not suffice, and star or tree topologies will be more suitable.

The UMoMI-R module is designed to provide control functions of the router for such networks. The UMoMI-R has several SpaceWire interfaces and all its control functionality is implemented in a Service FPGA, similar to one in the basic UMoMI. The particular reasonable number of SpaceWire ports in UMoMI-R starts from five.

We suggest extending the router with mass memory functionality in order to shorten the path of data in the satellite. Then the Service FPGA also controls transactions with array of memory devices, e.g., SSD. Overview of the UMoMI-R microarchitecture is presented in Fig. 3.

IV. UMOMI-BASED ONBOARD NETWORK CONFIGURATIONS

Since the UMoMI platform is designed for the network-based C&DH, it is necessary to describe the typical scenarios of the network applications for UMoMI. This section covers those cases and also gives examples of possible UMoMI configurations.

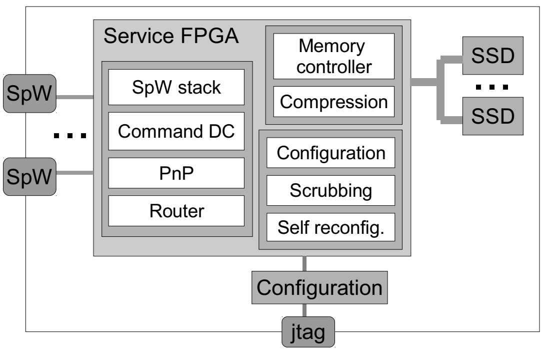

Fig. 3. Design of UMOMI-R

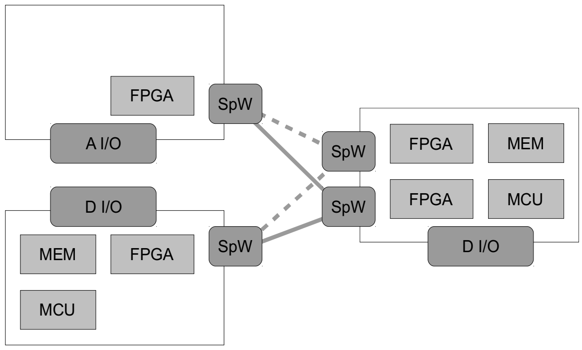

Fig. 4. UMoMI-based network with point-to-point links

A. Single-module satellite

Although the aim of the platform is embedded networks application, it can be applied for single-computer satellites as well, albeit with a certain unnecessary redundancy.

In this case a full set of UMoMI's components will be able to cover all the necessary C&DH functions. At the same time, both SpaceWire interfaces (at least the main one) will be probably left unused. Moreover, a single-module satellite designs often call for high levels of integration and therefore a custom solution will suit this task better.

B. Point-to-point communication

As mentioned earlier, if the onboard network is small, only point-to-point links without routing can be used. Fig. 4 presents an example of the network, where three modules are almost fully connected.

Both modules on the left are end nodes of the network. The modules perform communications with other satellite's subsystems via analog (top) and digital (bottom) interfaces, being either the sources, or the destinations in the data paths. The bottom module has an additional memory and MCU for advanced digital data processing.

Accordingly, the module on the right is a host. That module can implement a dual modular structural redundancy, when the same interface and C&DH functions are implemented in both FPGAs and there is a hand-shaking voting via the system bus. Another mapping of the functionality can be done so that the significant portion of C&DH is distributed between those two FPGAs, thus increasing the overall performance of the system.

C. Ring topology

There are multiple successful implementations of ring architectures for spacecraft buses, the IPDR from Sierra Nevada Corporation Space Systems [10] being one of them. Fig. 4 presents the double-ring network configuration with the same functionality as in a previous example.

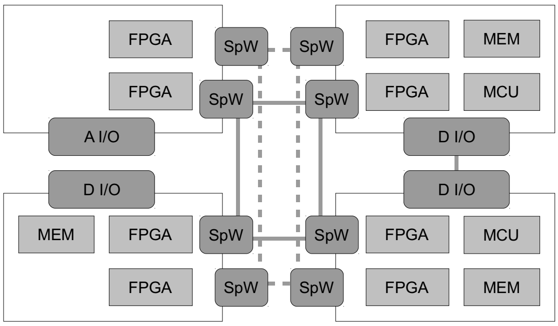

Fig. 5. UMoMI-based network with ring topology

However, opposed to the previous example, all four modules have both SpaceWire ports in order to provide a double-ring connectivity. In regards to onboard network communications, all the four modules are identical in functionality as they are obligated to transmit packages between adjacent nodes. One of them can be assigned to be a host during assembly of the spacecraft, or the Plug-and-Play implementation may provide a feature of automatic host selection and configuration.

Regarding performance/reliability tradeoff, two UMoMIs on the right have similar configuration and can be configured to work either in parallel, or as redundant modules.

D. Start/tree topology with router

This topology should be used for an onboard network, consisting of a big number of nodes. In this case UMoMI-R handles packet transfer between the modules. Fig. 6 depicts an example of a network with duplicated routers.

V. EARTH OBSERVATION SATELLITE

Every satellite bus is usually designed for a particular mission type. However, C&DH tasks are typical for the widest range of space applications. Therefore, we presume that a universal platform can be used as a basic unit for the C&DH subsystem.

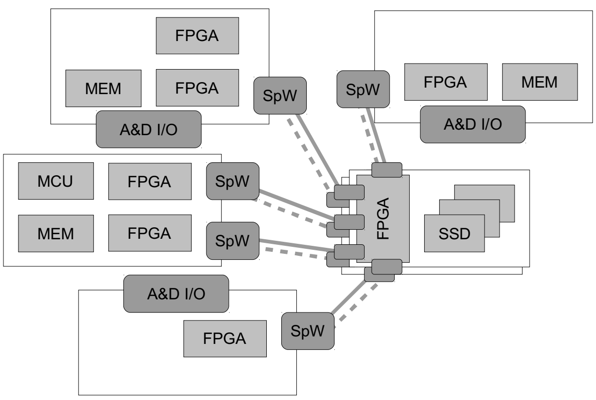

Fig. 6. UMoMI-based network with star topology

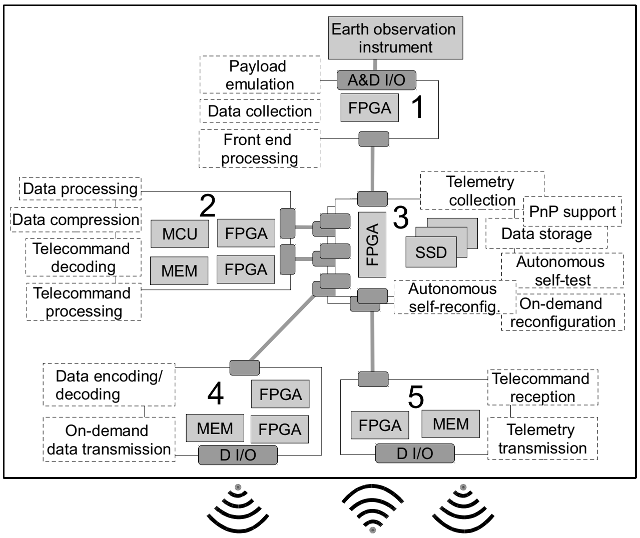

Fig. 7. Earth observation functionality mapped onto UMoMI platform

To explore that idea, this section gives a more detailed presentation of the platform, used for a small communication satellite. The mission task of the satellite is to conduct an Earth observation and transmit collected data on demand to the ground user terminal. There are following C&DH modes for the satellite:

- Payload emulation during assembly, integration and test of the satellite,
- Plug-and-Play support during assembly, integration and test of the satellite,
- In-flight continuous payload's data collection, processing, compression and storing,
- On demand transmission of collected data to the mobile ground station,
- Telemetry collection and transmission to the ground station – either continuous, or on demand,
- Telecommand reception, decoding and processing,
- In-flight on-demand reconfiguration,
- Autonomous self-test and self-reconfiguration.

Fig. 7 presents one of the possible C&DH functions' mapping on UMoMI modules. The modules are numbered from 1 to 5 for the reference.

The 1-st UMoMI is used to interface (or emulate) a payload – an Earth observation instrument (camera, radiometers, etc.). This UMoMI, or if necessary, an auxiliary one, can collect telemetry data from available sensors and send control signals to actuators. This UMoMI performs pre-processing and stores portions of data to be sent to the 2-nd UMoMI, where the most computational-heavy tasks are performed. The 2-nd UMoMI also executes telecommands and coordinates all satellite's subsystems.

All connections in the onboard network are managed by the 3-rd UMoMI-R. It also stores processed and compressed data in a mass memory array between on-demand transmissions to the user terminal. Moreover, it is convenient to implement Plug-and-Play host, diagnostic and reconfiguration functions in the UMoMI-R, since it has direct connections to all other UMoMI modules. The 3-rd UMoMI-R is duplicated to have a maximum SpaceWire bandwidth.

The 4-th UMoMI interfaces with a communication payload, which sends results of monitoring to the user terminal. The 5-th UMoMI interfaces with a service communication system, which sends telemetry and receives telecommands from a ground station.

VI. Discussion

Previous section shows an example of C&DH design from scratch, where all devices like payload equipment, sensors, actuators and transceivers have no network interface and therefore have to be connected via network adapters – UMoMI modules. This design can fully use flexibility of a platform: arbitrary network topology, arbitrary Plug-and-Play implementation and distributed data processing. The design can be easily extended for a wide network of sensors and actuators by using UMoMI modules for a front-end processing. Those modules will have minimal configuration: digital and analog interfaces and Service FPGA.

However, a more realistic design scenario will demand either integration of a new C&DH module into an existing network, or creating a network using mixture of new modules, existing SpaceWire-compliant modules (including routers) and proprietary modules. In this case UMoMI platform can be used for design of all new modules and adapters. In particular, UMoMI-R module can be used for design of a mass memory device and a router.

Another important application of the platform, not covered in the previous sections, is for design of testing equipment, which can be used during assembly, integration and test of the satellite.

VII. Conclusion

The main goal of any design methodology is to provide a required result while minimizing design time and cost. The same is true for a small satellite technology. So current trends in design of Command and Data Handling embedded networks for small satellites are towards standard architectures, network interfaces and protocols.

This paper explores another aspect of design – possibilities of a unified hardware platform for an onboard network of a small satellite. Presented platform targets modern SpaceWire-based Plug-and-Play architectures. The platform has two types of Universal Modules for Managing the Information (basic computational UMoMI and network router UMoMI-R). The modules provide multiple levels of configurability from the onboard network topology to the on-orbit flight software adjustment. An example of a typical Earth observation satellite shows, that UMoMI platform can be adopted to perform an entire spectrum of C&DH functions, related to payload data and service information.

The future work on the UMoMI project lays in further refinement of the blueprinted concept, continued with application of the platform for design of a series of small communication satellites.

Acknowledgment

This work has been being supported by scientific-educational center “Embedded Microelectronic System” of SPbPU.

References

[1] M. Martin and J. Lyke, “Modular Open Network ARCHitecture (MONARCH): Transitioning plug-and-play to aerospace”, *IEEE Aerospace Conference*, 2013, pp. 1-10.

[2] A. Sangiovanni-Vincentelli, “Quo Vadis, SLD? Reasoning About the Trends and Challenges of System Level Design”, *Proceedings of the IEEE*, Vol. 95, Iss. 3, 2007, pp. 467-506.

[3] *Nano/Microsatellite Market Assessment*. SpaceWorks, 2014.

[4] J. Eickhoff, *A Combined Data and Power Management Infrastructure*. Berlin: Springer-Verlag, 2013.

[5] Earth observation portal, Satellite Missions Database, Web: https://directory.eoportal.org/web/eoportal/satellite-missions.

[6] Gunter’s Space Page, Orbital Launches of 2014, Web: http://space.skyrocket.de/doc_chr/lau2014.htm.

[7] *Space Plug-and-Play Architecture Standards Development Guidebook*. American Institute of Aeronautics and Astronautics, 2013.

[8] P. Mendham, A.F. Florit, and S. Parkes, “SpaceWire Plug-and-Play: A roadmap”, *Spacewire conference*, 2008, pp. 95-103.

[9] J. Sheynin, N. Solokhina, and Y. Petrikovitch, “SpaceWire Technology for Parallel Systems and Airborne Distributed Complexes”, *Electronika:NTB*, No. 5, 2006, pp. 64-75. (In Russian).

[10] H. Curtis and T. Mosher, “Leveraging the First ORS Mission into ORBCOMM and the Implications for Future ORS Missions”, *7th Responsive Space Conference*, Los Angeles, 2009, RS7-2009-3001.