View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by NASA Technical Reports Server

NASA/TM—2008-215283

Hardware Architecture Study for NASA's Space Software Defined Radios

Richard C. Reinhart and Maximilian C. Scardelletti
Glenn Research Center, Cleveland, Ohio

Dale J. Mortensen and Thomas J. Kacpura
ZIN Technologies, Cleveland, Ohio

Monty Andro
Glenn Research Center, Cleveland, Ohio

Carl Smith and John Liebetreu
General Dynamics-C4S, Scottsdale, Arizona

July 2008

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to *help@sti.nasa.gov*

- Fax your question to the NASA STI Help Desk at 301–621–0134

- Telephone the NASA STI Help Desk at 301–621–0390

- Write to:
  NASA Center for AeroSpace Information (CASI)
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TM—2008-215283

Hardware Architecture Study for NASA's Space Software Defined Radios

Richard C. Reinhart and Maximilian C. Scardelletti
Glenn Research Center, Cleveland, Ohio

Dale J. Mortensen and Thomas J. Kacpura
ZIN Technologies, Cleveland, Ohio

Monty Andro
Glenn Research Center, Cleveland, Ohio

Carl Smith and John Liebetreu
General Dynamics-C4S, Scottsdale, Arizona

Prepared for the
Wireless and Microwave Technology Conference (WAMI)
sponsored by the IEEE, MTTS, and the Electron Devices Society
Clearwater, Florida, December 4–5, 2006

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

July 2008

Acknowledgments

The authors wish to acknowledge the contributions of the Jet Propulsion Laboratory and SDR Forum, Space Applications Work Group member companies contributing to the hardware architecture definition.

This report contains preliminary findings, subject to revision as analysis proceeds.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076–1320

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

Available electronically at http://gltrs.grc.nasa.gov

Hardware Architecture Study for NASA's Space Software Defined Radios

Richard C. Reinhart and Maximilian C. Scardelletti
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Dale J. Mortensen and Thomas J. Kacpura
ZIN Technologies
Cleveland, Ohio 44135

Monty Andro
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Carl Smith and John Liebetreu
General Dynamics-C4S
Scottsdale, Arizona 85257

Abstract

This study defines a hardware architecture approach for software defined radios to enable commonality among NASA space missions. The architecture accommodates a range of reconfigurable processing technologies including general purpose processors, digital signal processors, field programmable gate arrays (FPGAs), and application- specific integrated circuits (ASICs) in addition to flexible and tunable radio frequency (RF) front-ends to satisfy varying mission requirements.

The hardware architecture consists of modules, radio functions, and interfaces. The modules are a logical division of common radio functions that comprise a typical communication radio. This paper describes the architecture details, module definitions and the typical functions on each module as well as the module interfaces. Trade-offs between component-based, custom architecture and a functional-based, open architecture are described. The architecture does not specify the internal physical implementation within each module, nor does the architecture mandate the standards or ratings of the hardware used to construct the radios.

Introduction

A software defined radio (SDR) is a collection of hardware and software technologies that enable reconfigurable systems for communication networks. Software defined radios are programmable systems with partitioned software and hardware modules controlled by managing software that conform to defined interfaces to allow design reuse, software portability, and provide scalability across hardware platforms. The advantages of flexible and adaptable operation in the digital and RF domains offer significant capabilities and performance compared to legacy radios with fixed, dedicated functionality.

A software defined radio has the potential to save missions cost since these multi-mode, multi-band, multi-functional radio systems can be dynamically changed using software upgrades. In other words, the same piece of "hardware" can be modified to perform different functions at different times through reprogrammable software modules, reducing the total number of radios required for a given mission.

Software defined radios based on a *standard* architecture provides NASA with a consistent and common framework to develop, space qualify, operate and maintain these complex reconfigurable and reprogrammable radio systems for space applications. A standard *open* architecture has published interfaces enabling different vendors to provide radios that meet the interface standard, providing commonality among different implementations, and enabling interoperability between providers of different hardware and software. Open interfaces allow for flexible component replacement and technology insertion. Commonality among radios, design reuse, and software reuse potentially reduce NASA mission risk. The Space Telecommunications Radio System (STRS) Architecture (ref. 1) is one approach under consideration to establish a standard, open architecture for NASA's SDR developments.

As NASA looks to the future of space exploration, the growth of reconfigurable electronics and software defined radios provide an opportunity to change the way space missions develop and operate space transceivers for communications and navigation.

Architecture Description

A software defined radio architecture "is: …a comprehensive, consistent set of functions, components, and design rules according to which radio communication systems may be organized, designed, constructed, deployed, operated and evolved over time. A useful architecture partitions functions and components such that a) functions are assigned to components clearly and b) physical interfaces among components correspond to logical interfaces among functions" (ref. 2).

Current implementations for space-deployed SDR hardware are typically represented in terms of the device components which comprise the SDR. Figure 1 shows a high-level, component-based, custom hardware architecture. The custom approach emphasizes typical hardware transceiver elements such as ASICs, FPGAs, specific memory elements (e.g., RAM, EEPROM), physical interfaces (e.g., RS-232, Spacewire), and RF signal conversion and filtering components (e.g., band pass filter, specific analog-to-digital converter). These are generally represented using available technology.

The diagram illustrates custom connections between radio processing devices and between the processing elements and the RF front end. These custom radio interface connections are often proprietary to the radio developer.

The custom approach often provides an efficient solution to meet a particular mission's requirements. Design, development, testing, and space qualification are specific and unique for that implementation and radio design. However, modification of the design and new software development is often required to accommodate new parts due to parts obsolescence and to allow for technology insertion to meet new mission requirements.

The proprietary interfaces limit NASA's ability to extend and reuse the software investments in software radio developments from one mission to another. Reuse of the custom implementation is limited to missions with a similar set of requirements and requires NASA to use the same provider. However, hardware reuse of a specific vendor's radio increases costs due to loss of competition.

Figure 2 shows the current STRS open hardware architecture. The open hardware architecture emphasizes the radio functions and key interfaces. The radio functions are distributed among different modules, to organize different platform services and waveform functions within the radio. *Modules* are a logical division of functionality to maintain common interface descriptions, terminology and documentation among SDR developments. A *waveform* comprises the end to end functionality (e.g., modulation, coding, frequency conversion, filtering) and bidirectional transformations applied to information content that is transmitted over the air.

The three major modules of the architecture are shown, illustrating the command and control, signal processing, and analog portion of a radio.

The General-Purpose Processing Module (GPM) provides the basic software execution processes based on general purpose processors. The GPM is a required module supporting the operating environment responsible for waveform instantiation and execution, radio services, and hardware abstraction. The Signal Processing Module (SPM) and RF Module (RFM) conduct signal processing and RF front end functions, respectively. Other modules not explicitly shown include security, networking, and optical as required by the transceiver. The radio developer has the flexibility to combine these modules and their functionality as necessary during the radio design process to meet the specific mission requirements.

Several key external interfaces of the architecture include Host Telemetry, Tracking and Command (TT&C), Ground Test, Data, Clock, and Antenna. The Host/TT&C interface represents the low-latency, low-rate interface for the spacecraft (or other host) to communicate with the radio. This type of information includes health, status, and performance parameters of the radio and the link in use. In addition, telemetry often includes radiometric tracking and navigation data. Information found on this interface includes configuration parameters, configuration data files, new software data files, and operational commands.

The Ground Test Interface is exclusively used for ground-based integration and testing functions. It typically provides low level access to internal parameters that are unavailable to the Spacecraft TT&C Interface.

The Data Interface is the primary interface for information that is transmitted or received by the radio. This interface is separate from the TT&C interface because it typically has a different set of transfer parameters (e.g., protocol, speeds, data volumes) than the TT&C information. This interface is also characterized by medium to high latency and high data rates.

The Clock Interface is used for receiving the frequency reference required to support navigation and tracking. This type of input frequency reference is essential to the operation of the radio and provides references to the SPM and RFM.

The Antenna Interface is used for connecting the electromagnetic signal (input or output) to the radiating element or elements of the spacecraft. It also includes the necessary capability for switching among the elements as required.

Some internal interfaces must be defined in an open manner to support the overall goals of the architecture. Internal interfaces include the system bus between the GPM and SPM, various control lines between the GPM and RFM, ground test interface to each module from the GPM, and frequency reference from the RFM to the other modules.

The System Bus provides the primary interconnect between the GPM's microprocessor and the GPM's memory elements, interconnect to the external interfaces, and Telemetry, Tracking, and Command and Ground Test Interfaces. The System Bus is the primary interface between the GPM and the SPM. The System Bus provides the interface to re-program and reconfigure elements of the SDR. It supports the read/write access to the SPM elements, as well as reloading of configuration files to the FPGAs.

The interface between the GPM and the RFM is primarily a control/status interface. It is important to have a hardware-based confirmation and limit-check on the software control of

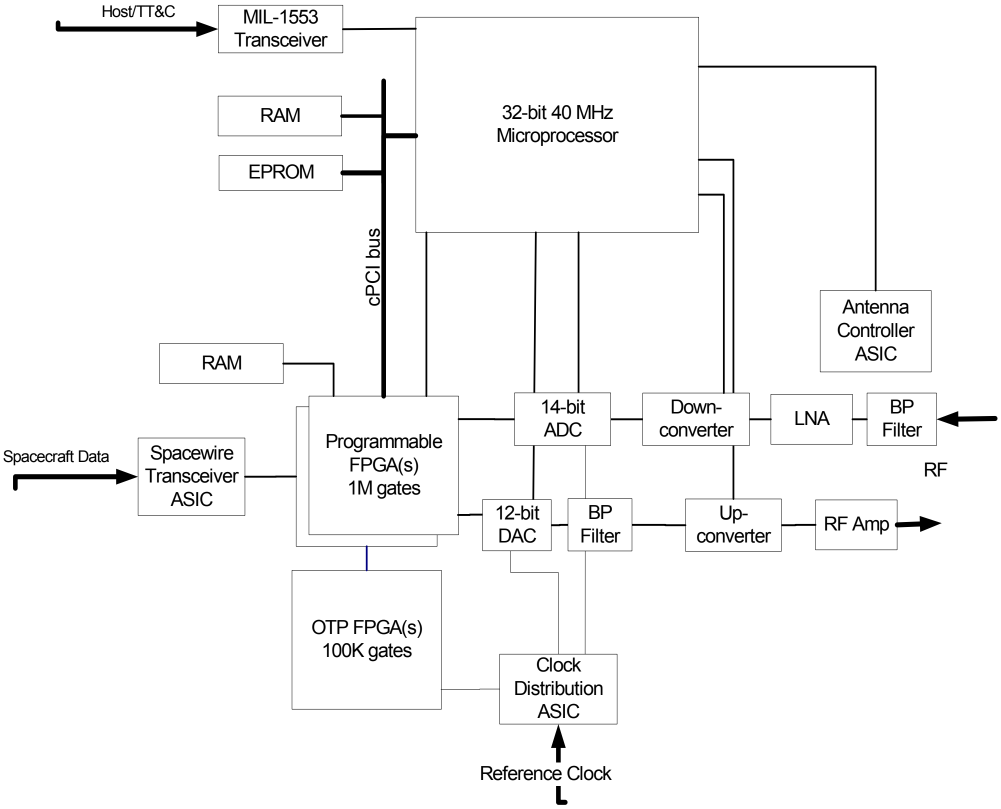

Figure 1.—Custom implementation hardware architecture.

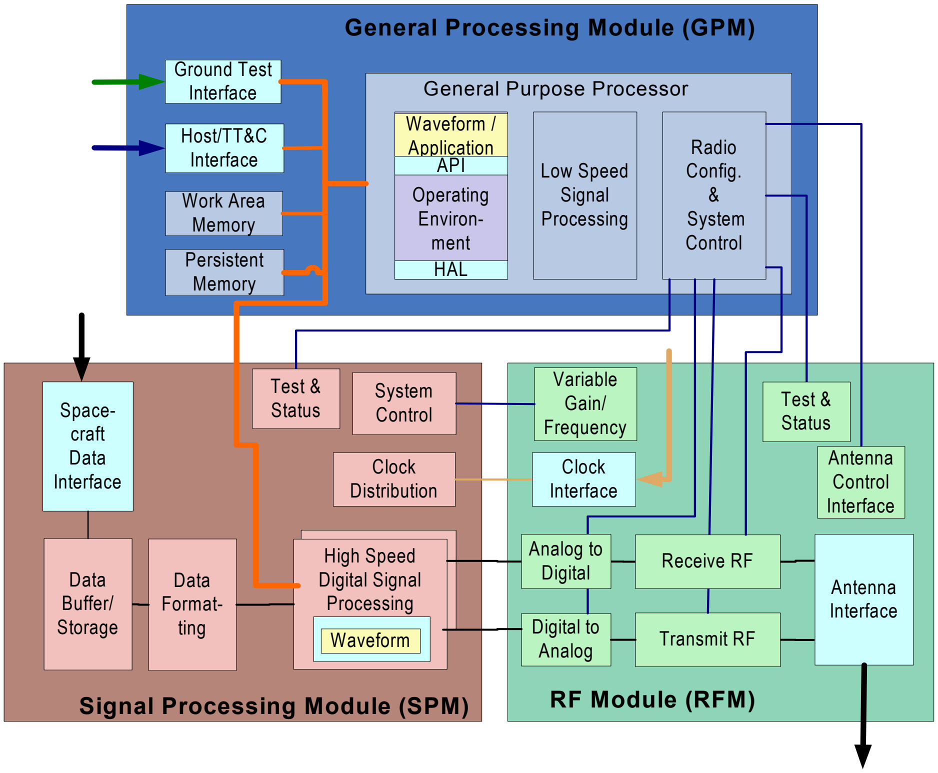

Figure 2.—Open hardware architecture.

any RFM elements. The system control element of the GPM provides this functionality, thus keeping the GPM RFM control bus within operational limits.

The internal Test and Status Interfaces provide specific control and status signals from different modules or functions to the external Ground Test Interface. These interfaces are used during development and testing to validate the operation of the various radio functions.

Finally, the data paths are the various streams of bits, symbols, and RF waves connecting the major blocks of the primary data-path. For any particular implementation, the data path or bit streams are defined by the particular waveform implemented in the functional blocks. The interface between the RFM and SPM, however, should be well-defined and should have characteristics suitable for that level of conversion between the analog and digital domains.

This open architecture abstracts functionality away from specific hardware devices through the hardware and software interfaces enabling greater reuse of a design and minimizing the impact associated with parts obsolescence. Since the software is abstracted from the hardware, there is a greater likelihood of reusing the software in future developments.

The open architecture approach allows NASA to reuse its investment in software radio developments, yet:

1. Maintain company proprietary approaches and designs behind the common interfaces,
2. Reuse the architecture specification among different developments and different vendors, and
3. Preserve commonality among designs, development, testing, and space qualification processes.

Table 1 summarizes the trades between a custom architecture and an open architecture SDR approach. While the custom approach efficiently meets mission requirements, the approach is limited to today's technologies. The open architecture provides more flexibility to NASA, yet maintains proprietary implementations of the respective developers.

TABLE 1.—HARDWARE ARCHITECTURE TRADES

Hardware Architecture Trade Summary	
Custom	Open
Cost/power efficient for specific mission requirements	Published interfaces
Applicable to today's technology	Reduces impact of parts obsolescence
Unique design, test, space qualification	Functions abstracted from hardware
Proprietary to specific developer	Retains proprietary radio aspects

To achieve the benefit of hardware and software reuse, an STRS repository is envisioned where appropriate transceiver interfaces; documentation and software artifacts submitted by developers are reused by subsequent developers.

Hardware modules that can be used again for other missions can potentially reduce cost, system integration, and risk. Using previously developed hardware reduces risk and cost since the hardware has space flight heritage, has demonstrated reliability and space qualification procedures are known.

Often, the motivation to change hardware is either new requirements or parts obsolescence. Defining standard interfaces between modules allows developers to insert new hardware while still reusing other aspects of the radio. The ability to insert new hardware allows multiple vendors to contribute. Software interfaces help mitigate parts obsolescence by reusing software with the new parts, thus saving development time and cost.

Radio Development Process

During the radio development process, one can apply the open hardware architecture from both a function-based view and a more component-based view as the process evolves, as illustrated in figure 3. In both instances, the architecture benefits emerge where software and hardware modules, documentation, and interfaces can be common among successive developments.

The radio development process begins with an assessment of mission requirements applicable to the communications system (e.g., communications and navigation radio). During this requirements phase, the team determines radio and then ultimately waveform functionality (e.g., modulation type, coding, filtering, frequency conversion) required for the mission. The hardware architecture depicted in figure 2 illustrates how this functionality can be divided among the various standard modules for consistent terminology and use of common interfaces regardless of developer. As the process in figure 3 illustrates, the mission designers may access the STRS repository to reuse functionality based in software to reduce time and cost during the design and development phase.

As the transition from functional description to hardware design and implementation begins, the module representation along with published interfaces aids in reuse, test and verification. At this stage, designers conduct a mapping of waveform functions to specific signal processing and RF devices. The common architecture provides waveform independence from the platform developer through the standard interfaces. The interfaces defined by the standard provide an open and published interface, yet protects the intellectual property of the different developers. Using this approach, NASA (or in many cases the prime contractor) could integrate the best hardware modules from different developers into a single product based on the common interfaces.

Applying the Hardware Architecture

Figure 4 illustrates a working example of deploying a reconfigurable transmit waveform on an SDR platform while adhering to the hardware architecture. The example waveform represents typical low-rate command/telemetry waveform; quadrature phase shift keying (QPSK) modulation, ½ rate

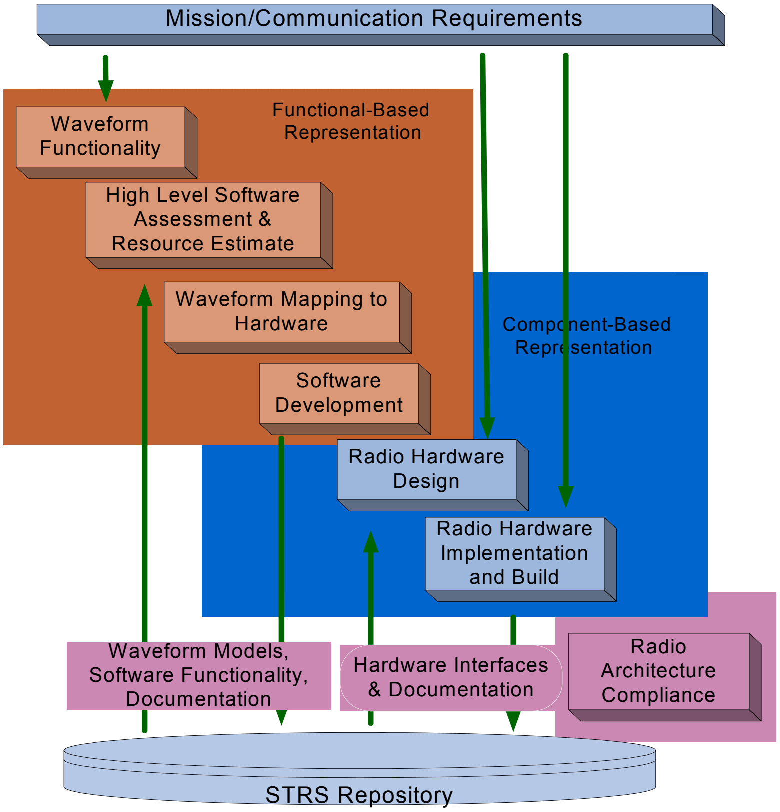
Mission/Communication Requirements
Functional-Based Representation
Waveform Functionality
High Level Software Assessment & Resource Estimate
Waveform Mapping to Hardware
Software Development
Component-Based Representation
Radio Hardware Design
Radio Hardware Implementation and Build
Waveform Models, Software Functionality, Documentation
Hardware Interfaces & Documentation
Radio Architecture Compliance
STRS Repository

Figure 3.—Radio development process

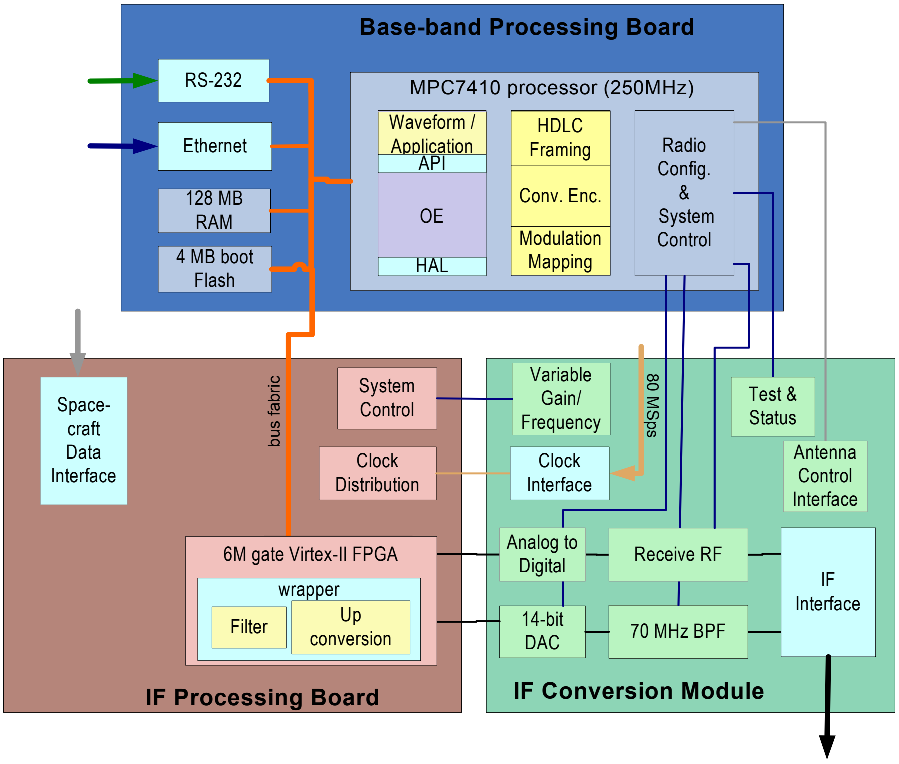

Figure 4.—Transmit waveform applied to hardware architecture.

coded waveform. Other functions include internal data generator, high-level data-link control (HDLC) framing, bandpass filtering, and digital upconversion.

The waveform functions are deployed across the General Purpose Processor Module, the Signal Processing Module, and the RF Module.

The low rate application data enters the radio through the Ethernet interface of the GPM. The data interface of the SPM is not used. The radio is controlled through an RS-232 serial interface for control, and reconfiguration. The GPM handles the low rate functions of the waveform such as HDLC framing, convolutional encoding, and modulation. The data is sent from the GPM to the SPM for filtering, and digital upconversion. The RFM handles the digital to analog conversion, bandpass filtering, signal conditioning (e.g., amplification and IF output). In this case the highest output frequency is 70 MHz, thus a second upconverter is necessary on the RFM to reach a higher RF.

The operating environment (OE) abstracts the low speed signal processing waveform functions from the underlying processor according to the rules of the architecture. In the example, several functions of the transmit telemetry waveform were deployed to the GPM to exercise more functionality of the architecture abstraction. To comply with the STRS Architecture, the developers must provide the radio services described in the STRS Standard and publish the FPGA wrapper interface used in the example. This allows the developer to maintain the proprietary character of their intellectual property associated with the algorithm portion of the filtering and upconversion. This approach exposes the interfaces used in the FPGA for subsequent developers, but protects the investment made by the original developer. The developer must also provide a description of the physical hardware interfaces used in the implementation and a mapping of the control interfaces to each of the modules.

Conclusion

NASA is considering a standard for software defined radios as they begin to make their way into NASA missions. Commonality among different developments could include software and hardware interfaces, test points, command protocols, models, and documentation. There is a role for both functional-based, open architecture and device-based representation in the radio development process. Representing the architecture as functions serves as an aid in early mission and radio functionality definition and development. The device oriented representation better applies during radio design and development. Developers have agreed that standardization applied to software radio hardware will aid in the development, testing, and verification processes. However, discussions continue on the level of standardization and exactly which interfaces to apply the standard. Multiple agencies and technology and standards bodies have joined together to achieve a successful architecture.

References

1. Reinhart, Richard C., et al.: Space Telecommunications Radio System (STRS) Architecture Description. NASA/TP—2008-214821, 2008.
2. J. Mitola, *Software Radio Architecture*, John Wiley & Sons, New York, 2000.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 01-07-2008	**2. REPORT TYPE** Technical Memorandum	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE
Hardware Architecture Study for NASA's Space Software Defined Radios

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Reinhart, Richard, E.; Scardelletti, Maximilian, C.; Mortensen, Dale, J.; Kacpura, Thomas, J.; Andro, Monty; Smith, Carl; Liebetreu, John

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
WBS 439432.04.07.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field
Cleveland, Ohio 44135-3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E-16550

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORS ACRONYM(S)
NASA

11. SPONSORING/MONITORING REPORT NUMBER
NASA/TM-2008-215283

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category: 17
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301-621-0390

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This study defines a hardware architecture approach for software defined radios to enable commonality among NASA space missions. The architecture accommodates a range of reconfigurable processing technologies including general purpose processors, digital signal processors, field programmable gate arrays (FPGAs), and application-specific integrated circuits (ASICs) in addition to flexible and tunable radio frequency (RF) front-ends to satisfy varying mission requirements. The hardware architecture consists of modules, radio functions, and and interfaces. The modules are a logical division of common radio functions that comprise a typical communication radio. This paper describes the architecture details, module definitions, and the typical functions on each module as well as the module interfaces. Trade-offs between component-based, custom architecture and a functional-based, open architecture are described. The architecture does not specify the internal physical implementation within each module, nor does the architecture mandate the standards or ratings of the hardware used to construct the radios.

15. SUBJECT TERMS
Software defined radio

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** STI Help Desk (email:help@sti.nasa.gov)
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	12	**19b. TELEPHONE NUMBER** ***(include area code)*** 301-621-0390