NASA/TM—2007-214828

Modeling and Analysis of Space Based Transceivers

Michael S. Moore, Jeremy C. Price, and Ben Abbott
Southwest Research Institute, San Antonio, Texas

John Liebetreu
General Dynamics Advanced Information Systems, Scottsdale, Arizona

Richard C. Reinhart
Glenn Research Center, Cleveland, Ohio

Thomas J. Kacpura
ZIN Technologies, Inc., Brook Park, Ohio

June 2007

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—2007-214828

Modeling and Analysis of Space Based Transceivers

Michael S. Moore, Jeremy C. Price, and Ben Abbott
Southwest Research Institute, San Antonio, Texas

John Liebetreu
General Dynamics Advanced Information Systems, Scottsdale, Arizona

Richard C. Reinhart
Glenn Research Center, Cleveland, Ohio

Thomas J. Kacpura
ZIN Technologies, Inc., Brook Park, Ohio

Prepared for the
29th Annual Software Engineering Workhop (SEW–29 2005)
sponsored by the Institute of Electrical and Electronics Engineers and the National Aeronautics
and Space Administration
Greenbelt, Maryland, April 6–7, 2005

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

June 2007

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

Modeling and Analysis of Space Based Transceivers

Michael S. Moore, Jeremy C. Price, and Ben Abbott
Southwest Research Institute
San Antonio, Texas 78238

John Liebetreu
General Dynamics Advanced Information Systems
Scottsdale, Arizona 85257

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

Thomas J. Kacpura
ZIN Technologies, Inc.
Brook Park, Ohio 44142

Abstract

This paper presents the tool chain, methodology, and initial results of a study to provide a thorough, objective, and quantitative analysis of the design alternatives for space Software Defined Radio (SDR) transceivers. The approach taken was to develop a set of models and tools for describing communications requirements, the algorithm resource requirements, the available hardware, and the alternative software architectures, and generate analysis data necessary to compare alternative designs. The Space Transceiver Analysis Tool (STAT) was developed to help users identify and select representative designs, calculate the analysis data, and perform a comparative analysis of the representative designs. The tool allows the design space to be searched quickly while permitting incremental refinement in regions of higher payoff.

1. Introduction

The National Aeronautics and Space Administration's (NASA) Glenn Research Center (GRC) is investigating the development and suitability of software-based reconfigurable transceivers (RTs) and Software Defined Radios (SDRs) to enable advanced operations and to reduce mission costs for space platforms. By replacing fixed hardware with flexible software, the ability to change the operation of a radio by changing only the software opens a world of possibilities that was not previously feasible. Software defined radio for space is attractive, as in many terrestrial applications, because of the inherent reconfigurability, which enhances capability. Also, there may be cost savings through software reuse and waveform portability.

NASA GRC's Space Telecommunications Radio Systems (STRS) project is a study and modeling effort to explore the SDR technologies, cost, benefits, and risks to establish an open-architecture for reconfigurable transceivers and software defined radios for NASA. The STRS team, including space communication engineers at NASA GRC, General Dynamics, and Southwest Research Institute (SwRI), is evaluating the applicability and tradeoffs concerning the use of SDR technologies for space missions. The approach is to quantitatively assess the costs (size, weight, power, and latency, as well as development costs) of employing SDR technologies in general, and moreover of adopting an open standard software architecture for space SDR transceivers.

The approaches considered were (1) leveraging or adapting the Joint Tactical Radio System (JTRS) Software Communication Architecture (SCA) for use in space or (2) developing a tailored, space-specific architecture for NASA. The study examined the scalability of SDR architectures to meet the range of NASA missions, from small to large mission classes, and data rates ranging from low Kbps data rates to high data rates of 100's or 1000's Mbps.

The STRS project produced a modeling and analysis tool for performing trades to quantify the implication of the software architecture approach on the size, weight, power consumption and modem latency of the mission-specific transceivers. The tool was used to perform trade studies in support of the project goals.

While NASA continues to consider aspects of the SCA combined with its own approach for open architecture software defined radios, tools are helpful to evaluate design trades, saving the cost of implementing each option to assess performance and cost. The development and modeling tools should advance the capability to abstract software, promote design and software reuse, and provide a means to quantify the resources required and the performance of different hardware architectures implementing a given communication architecture. The tool described in this research paper, the Space Transceiver Analysis Tool, is a first step to combine waveform and mission objectives with the mapping of functional allocations to a representative platform to determine resource requirements for the defined waveform algorithms, mapping of those waveforms to the hardware, and assessment of whether mission or waveform objectives are met.

The Space Transceiver Analysis Tool (STAT) provides a level of rigor and automation to the modeling and analysis process. STAT aids the user in identifying and selecting representative designs, calculating the various cost metrics, and performing a comparative analysis of the relative costs of those representative designs. The study applies this tool to examine the impact of design choices such as software architecture, middleware, number and type of hardware components, and channel parameters (e.g., modulation scheme, channel data rate) on costs (e.g., size, weight, power, engineering costs, purchase costs) and transceiver capabilities (e.g., in-flight reconfigurability and re-programmability). The STAT tool allows the design space to be searched quickly while supporting incremental refinement in regions of higher payoff.

NASA GRC's STRS project team, comprised of government and contractor participants, is supporting the Space Communications and Data Services program. The team is investigating the development and suitability of software-based reconfigurable transceivers (RTs) and software defined radios (SDRs) to enable advanced operations and reduce mission costs for space platforms. By replacing fixed hardware with flexible software, the ability to change the operation of a radio by changing only the software (or firmware) opens a world of possibilities that was not previously feasible. Software defined radio for space is attractive, as in many terrestrial applications, because of the inherent reconfigurability which enhances capability. Also, there may be cost savings through software reuse and waveform portability. NASA has recently performed space demonstrations with reconfigurable transceivers, and the commercial and defense sectors have significant interest in this area as well. The advantages of flexibility and improved performance in the digital domain offer significant capabilities compared to legacy radios.

Since software changes constitute the radio's reconfigurability, a radio with a clearly defined software architecture is paramount. The architecture of such a software system will allow for flexibility and change as well as promote the sharing and integration of software from a variety of sources. Portability and software reuse will play significant roles in keeping the cost of software radio development under control. A good open architecture software design will enable the separation of software from the hardware. Defining an open architecture for NASA space radios is part of the larger STRS program currently underway to define NASA's future communications system architecture.

Taking advantage of the latest advances in processing hardware is an important advantage of using an open architecture approach. An open architecture has published design rules enabling developers to produce software products compliant with the standard. This allows competition among different vendors developing hardware, software, waveforms, and operating environments independently. It may not, however, eliminate proprietary implementations of specific modules and functions that comply with the standard. An example of an open architecture for terrestrial systems is the JTRS SCA.

Features of a communications architecture include non-proprietary open standards, and application programming interfaces to enable software reuse and portability, independent hardware and software development, and hardware and software functional separation. Also of interest is the concept of a middle layer of software for hardware/software independence for maximum portability. This is a key concept to software defined radios, but a careful balance is needed to keep this aspect from overwhelming the processing requirements of the radio, especially for the space environment.

The space environment offers many challenges in design, development, and verification of software-defined radios. The harsh environmental conditions of space significantly limit hardware choices; for instance, available space processors lag the state of the art by one or two generations. The space environment restricts the extensive use of commercial off-the-shelf (COTS) components and requires radiation hardened equivalents. Spacecraft have fixed resources; and flight systems are constrained with limits on the available size, weight, and power, including the radio. As the processing power of Analog-to-Digital Converters (ADCs), Digital-to-Analog Converters (DACs), Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs) increase, more radio tasks can be moved from hardware to software, increasing the flexibility of the radios. However, the increased power to provide this flexibility needs to be weighed carefully against the power available from the spacecraft. Another challenge is to establish an open architecture to which manufacturers can build, considering the small number of radios required by NASA for the various missions. Most missions have highly customized requirements, and tailoring is required on a mission-by-mission basis. These last two factors reduce the impact of software reusability as a key driver for implementing software defined radios.

In spite of these challenges, there still exist several benefits for NASA to use SDRs in space that outweigh the disadvantages. The radios can be developed later in the project cycle (closer to launch time), and the software can be changed later than the hardware. There exists the capability of re-programming radios in flight (fix problems, evolve radios remotely) since the majority of the space missions disallow direct human intervention to make changes. Reconfigurability may not be required as part of planned mission operations; but if an unplanned event occurs, this flexibility may provide the only means to recover full operations or to take advantage of some extended mission functionality. Since the radios can be programmed for different frequencies or modes, the potential exists to reduce the number of radios per mission, which translates in reduced required size, weight, and power resources from the spacecraft. Several space radio manufacturers have been adding reconfigurability to their product lines as this provides them with the opportunity to leverage the software or firmware from previous efforts for new applications, thereby reducing new-product development costs.

While it is possible to compare the cost of two particular implementations by manually tabulating anticipated performance and expense of the design choices made along the way to describing the particular instances, this type of comparison only shows the two choices made. Many "what-if" questions quickly arise concerning alternate approaches for achieving comparable results. As such, manual tabulation for evaluating the cost and efficacy of regions of the design space was rejected by the study team. Rather, the approach adopted has been to develop a set of models that describe the communications requirements, the processing requirements, the available hardware, and the relevant properties of the alternative software architectures, and then to analyze the design space using objective cost and capability metrics.

2. Space Transceiver Analysis Tool (STAT)

2.1. STAT Overview

The STAT was built as part of the STRS program to evaluate alternative SDR architectures and implementations with quantitative, objective analysis. STAT is designed to allow users to model available software architectures, hardware platforms, and waveform implementation approaches, study the effect on resource utilization, modem latency, size, weight, power, and development or purchase cost of transceiver designs, and to vary the design choices. Design alternatives include functional design selections (waveform algorithms), hardware design (type, number, and interconnection of hardware components), software architecture (operating environment), and allocation (which waveform algorithms will be executed by which hardware components).

The STAT consists of pre-defined databases of hardware components and software infrastructures. The databases can be updated as new technology develops or filled with anticipated technology components to evaluate potential technology advancements. Users build models of different aspects of the transceiver communication system such as 1) the communications requirements for space missions, 2) the wireless communications waveforms that are used for space applications, 3) the computer hardware available for space flight, and 4) the properties of the various software architectures that could be used to implement the transceivers. These aspects of the communication system are then mapped to the transceiver platform hardware to form the basis of the radio system for the particular scenario. The STAT then aids the user by specifying and analyzing the properties of the point design, referred to as "as-built transceivers." An as-built transceiver is defined as a choice of hardware platform, software infrastructure, waveform algorithms, channel parameters, and mapping of waveform components onto hardware components. The design space includes all possible as-built transceivers that could potentially support the communications requirements of a specific mission. The as-built transceiver represents a unique solution within the design space.

The STAT tool calculates resource utilization, size, weight, and power, and modem latency for as-built transceivers. A specific as-built transceiver design is considered *valid* if no resources are over-utilized and the modem latency is within mission specifications. In addition to utilization and latency, the STAT sums up the size, weight, power, non-recurring engineering cost, and purchase cost for the hardware and software components for as-built transceiver designs. The STAT tool helps a designer identify valid as-built transceiver designs and compare and contrast the valid designs. The net effect is that quantitative trade analyses comparing the size, weight, power, latency, and costs of design alternatives can be performed quickly and efficiently.

The models required to perform the analysis include models of the mission communications requirements, the wireless communications waveforms, the available computer hardware, and the software architectures. These models represent different candidate implementations of the transceivers. The analysis will be performed on an "as-built" transceiver model, which is a specific instantiation of a transceiver including computing hardware, waveform algorithm blocks, software infrastructure components, and allocation of waveform algorithm blocks to hardware components. The analysis will take into account the architecture-specific waveform implications of the component to hardware allocation (e.g., adapter software objects that must exist when SCA components are implemented in an FPGA). The trades between cost and architectural approach are analyzed by comparing the as-built costs of designs utilizing different software infrastructure models or implementations.

2.2. Mission Models

The function of mission models is to capture the communication-related requirements of a mission, which include specifications of the channels that must be supported during the mission and during which parts of the mission those channels will be active. Each mission is made up of one or more *stages*. A stage identifies a time period of a mission during which the communications channels are fixed. Communications channels are reconfigured between stages, not during stages. For example, the first stage of a mission might be *launch*, when a vehicle is in ascent. When that launch vehicle attains a low-earth orbit, the communications systems may be reconfigured to support a different set of channels. This stage might be called *low earth orbit*. A mission stage model has one or more *communicators*, which are the entities that are communicating (e.g., launch vehicle, ground station, satellite, etc.). A mission stage model also has one or more *channels*, each which specifies a *sender* and *receiver* communicator and the parameters with which they are communicating, such as waveform, channel frequency, and data rate. The channels are modeled as uni-directional, so a full-duplex communication channel includes two channel models, one in each direction.

Each mission model stage implies that the communicators have transceivers capable of supporting simultaneous operation of all of the waveforms that the communicator will be using during that stage. The sequence of stages in a mission model implies both simultaneous and non-simultaneous waveform requirements on the transceivers of the various communicators in the mission. During a stage, the channels are assumed to operate simultaneously; and between stages, channels are removed and added. For each mission stage, the set of simultaneously supported waveforms is called a *transceiver configuration*. The set of transceiver configurations implied by a mission model is the *waveform set* for that mission.

2.3. Waveform Models

Figures 1 and 2 illustrate waveform models that specify the algorithms used to implement a communications channel endpoint. The models assume a generic top-level signal flow for transmitters and another for receivers. The flow required depends upon which endpoint of the uni-directional channel the waveform is implementing. The generic signal flow diagrams for the transmitter and receiver, referred to as the *notional signal flow* diagrams, are shown below.

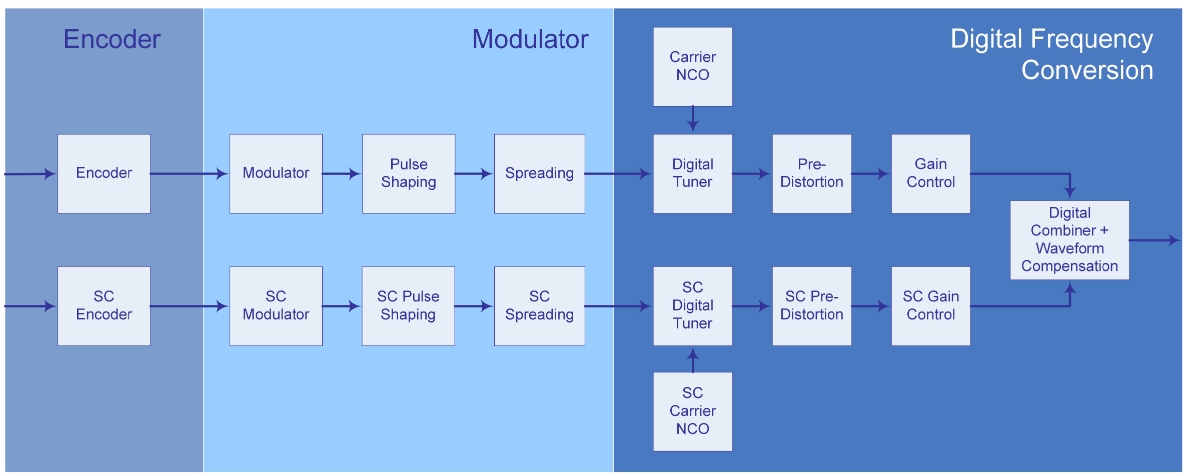


Figure 1.—Transmitter notional signal flow.

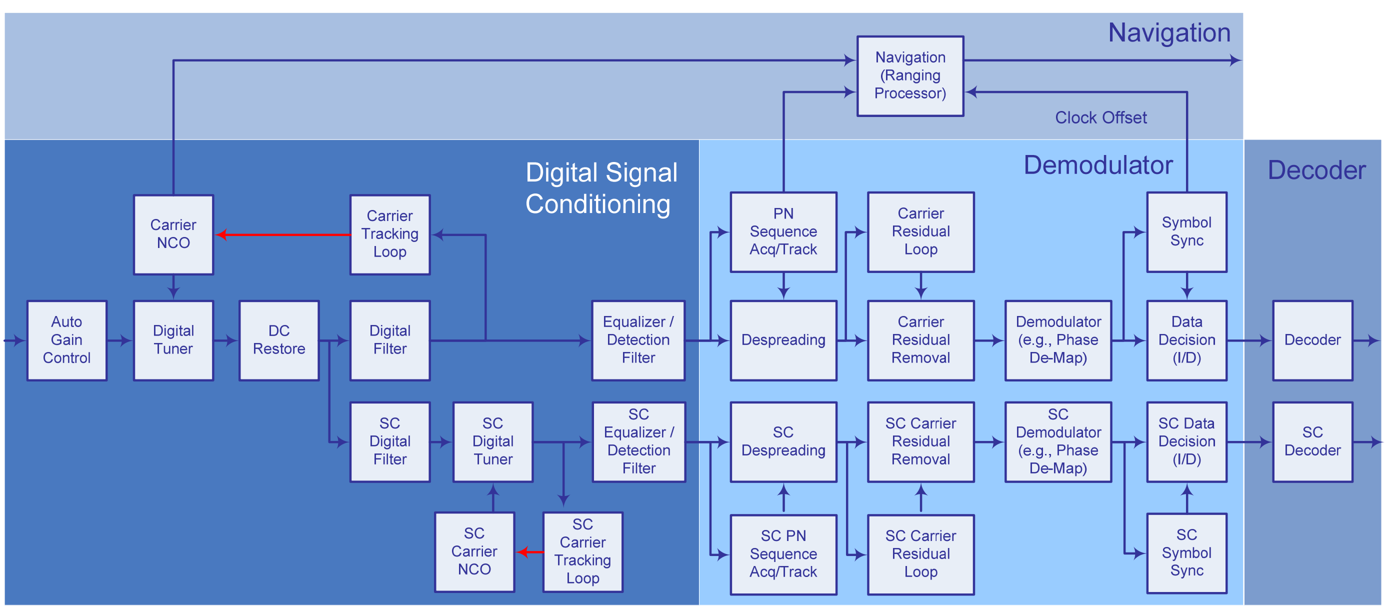


Figure 2.—Receiver notional signal flow.

These signal flow diagrams are considered generic and flexible enough to describe any of the NASA waveforms from a functional perspective. Each box in these signal flow diagrams is referred to as a *waveform algorithm* block. These diagrams are implementation independent, and are used as placeholders for the algorithmic selections that together form the functional specification of a waveform (e.g., a transmitter waveform using RS coding, BPSK modulation, no sub-carrier, and complex multiply digital up-conversion). A transmitter or receiver waveform may not use all of the waveform algorithm blocks. In that case, the waveform algorithm blocks can be set to *passthrough*, which is equivalent to replacing that block with a wire. Note that the waveform model does not specify channel specifics (data rate, bandwidth, carrier frequency) or implementation specifics (allocation to implementation hardware components, and physical transport for connections and infrastructure properties). Thus, these notional signal flow diagrams with selections made for each algorithm block represent a platform independent functional model of a waveform.

Each algorithm block modeled in the notional signal flow is modeled in an algorithm library (i.e., database), and has several parameters that describe the development cost and computation requirements. These include the estimates of the development cost (number of lines of code) and specifications of the computational requirements [Millions Instructions Per Second (MIPS), number of gates, memory] for each algorithm.

The computational requirements of an algorithm will sometimes vary based on the rates at which that algorithm must be executed. For example, the MIPS requirements for a software implementation will depend upon the data rate of the input connection, and the clockrate requirements of a firmware implementation will vary with the clock domain of the block. In general, the resource requirement models can be functions of the channel parameters (data rate, bandwidth, intermediate frequency), the sample rate at which the algorithm is executing (the input link rate for software components or clock domain for firmware components). For that reason, the resource requirements can be specified as either fixed numerical values or as functions of a list of context sensitive parameters. This flexibility is accomplished via the MATLAB (The Mathworks, Inc.) functional concept described in a later section. The modeler can define a functional relationship between channel and block parameters and the resources an algorithm will require when implemented on each type of hardware component.

The model includes a field that specifies the relative confidence that the modeler places on the resource estimates, called the *confidence*, as well as a field to describe the origin of the estimate (e.g., engineering estimate, MATLAB simulation, or benchmark on a particular platform). This provides the ability to gauge the fidelity of a model and to gradually evolve the model over time. For instance, the first estimates of the MIPS requirements of an algorithm often employ engineering rules of thumb. Such estimates provide a rough order of magnitude estimate of the actual requirements, so should be given a low confidence number such as 0.5. When simulations of the algorithm are performed at a later time, the MIPS requirements can be revised, and the confidence number increased to something like 0.75. Then when benchmarks are performed on a concrete implementation, the MIPS requirements can be finalized and the confidence number increased to 1.0, meaning that it is not an estimate but a measurement.

2.4. Hardware Models

The available computational resources are modeled in a library of *hardware component* models. Each hardware component model has fields that quantify computational resources such as MIPS, gates, and memory, and other attributes such as the clock speed, size, weight, and power consumption. These fields can contain constant values, or MATLAB functionals (see the later section on integration with MATLAB). *Hardware Assembly* models are sets of hardware components with their interconnections.

2.5. Infrastructure Models

A core intent of the STAT is to support analysis of the implications of the various architectural approaches. Perhaps the most crucial type of specification is the *infrastructure* model. This model is composed of one or more *infrastructure components*, which each have resource requirements such as processing and communication overheads, dynamic memory size, static memory size, and configuration time. Examples of infrastructure components include a real-time operating system, middleware layers (e.g., Common Object Request Broker Architecture (CORBA) and component frameworks (e.g., SCA core framework requires components to adhere to the SCA component model). The key contents of the infrastructure component models are equations describing the communications latency for local and remote packet transfers and the maximum communications throughput.

Component frameworks imply an implementation style for components and include estimates of the amount of framework-specific code that should be required to implement the interfaces between the infrastructure and the waveform functional code. For instance, suppose that a given *algorithm block*, say a Binary Phase Shift Keying (BPSK) modulator, can be implemented in 256 lines of code, not including any framework specific interface code. This would be likely implemented by a single function that takes in binary data and produces waveform signal samples as an output. In order for this modulator to live within an SCA environment, it would be packed in a CORBA component that provides all of the SCA-specific interfaces. This extra code is implied by the combination of the BPSK algorithm block with an SCA infrastructure. The amount of framework-specific code that is implied by a particular framework is modeled in the infrastructure model lines of code field.

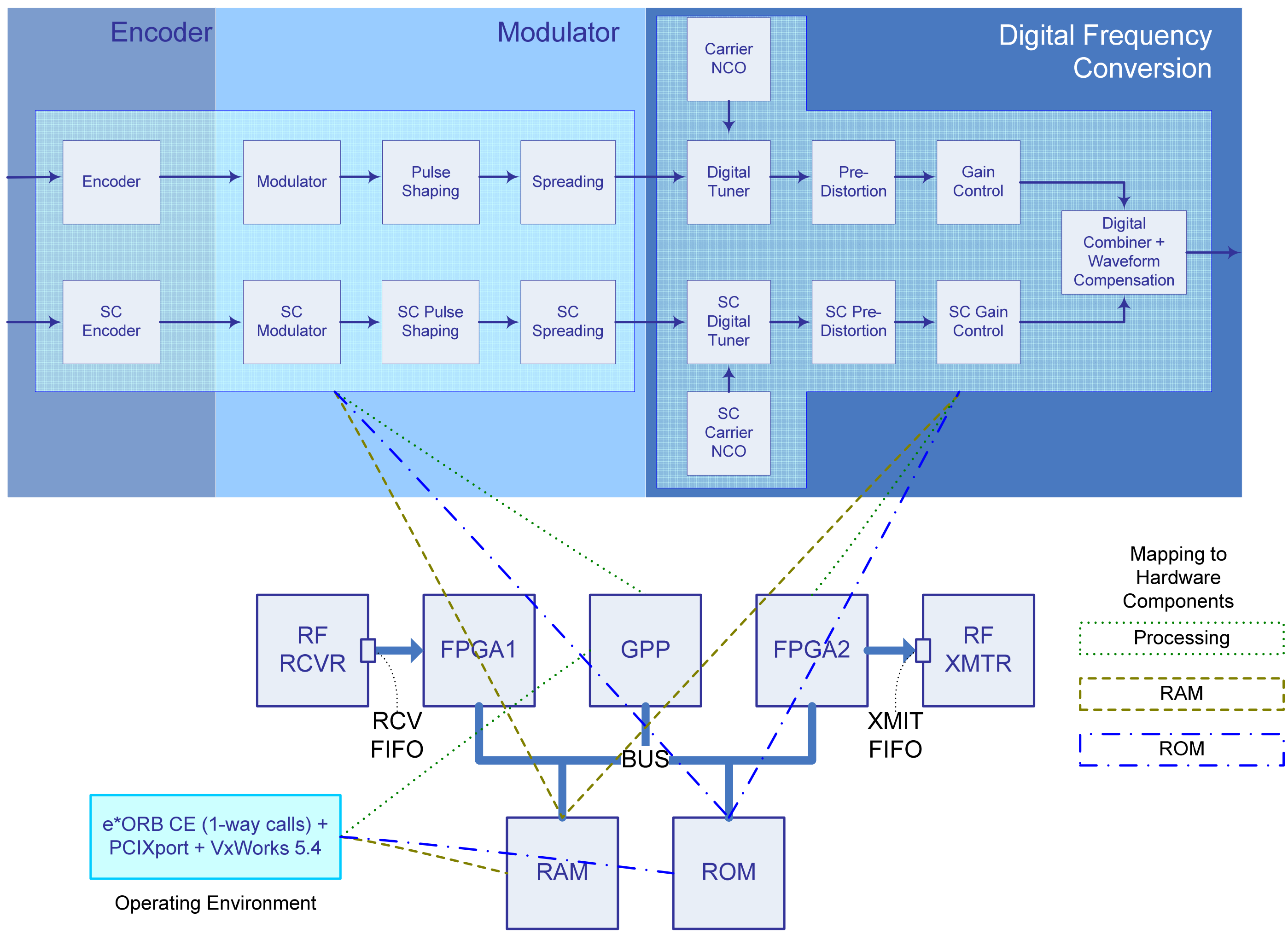


Figure 3.—Waveform algorithm block to HW component allocation.

2.6. As-Built Transceiver Models

To define an implementation against the mission requirements, an implementation platform (a set of hardware assemblies and an infrastructure) is selected, and software components are allocated to hardware components. Figure 3 shows the allocation of an example waveform notional signal flow to hardware.

The result of the waveform-to-hardware allocation process is an *as-built* transceiver design. An as-built transceiver is the combination of a waveform set model, a set of hardware assembly models, a software infrastructure model, and a mapping of the waveform components to the hardware components. This combination specifies a complete design and answers how and by which hardware component each algorithm block will be implemented, which physical transport and infrastructure services will carry the intra-waveform data, and what type of software infrastructure will be running on the processing nodes. Figure 4 shows the as-built transceiver signal flow that results from the allocation in figure 3. Note that the allocation of the modulation blocks to an FPGA has resulted in the addition of a software component on the General Purpose Processor (GPP) that acts as an “adapter” to handle interfaces to the FPGA. The dark blue rectangles to the left of each component running on the GPP are the architecture-specific (e.g., SCA) interfaces for each of those components. The processing and memory overheads induced by these component interfaces and adapters are rolled into the processing and memory “taxes” specified in the infrastructure component model.

2.7. MATLAB Integration

As was mentioned in previous sections, to provide the ability to model the functional relationship between context sensitive parameters such as the input sample rate and the resource requirements of an algorithm, it was necessary to provide the ability to model what is referred to as *functionals*. The STAT models have special types of entries that are called *MATLAB functionals. These* are MATLAB scripts that are

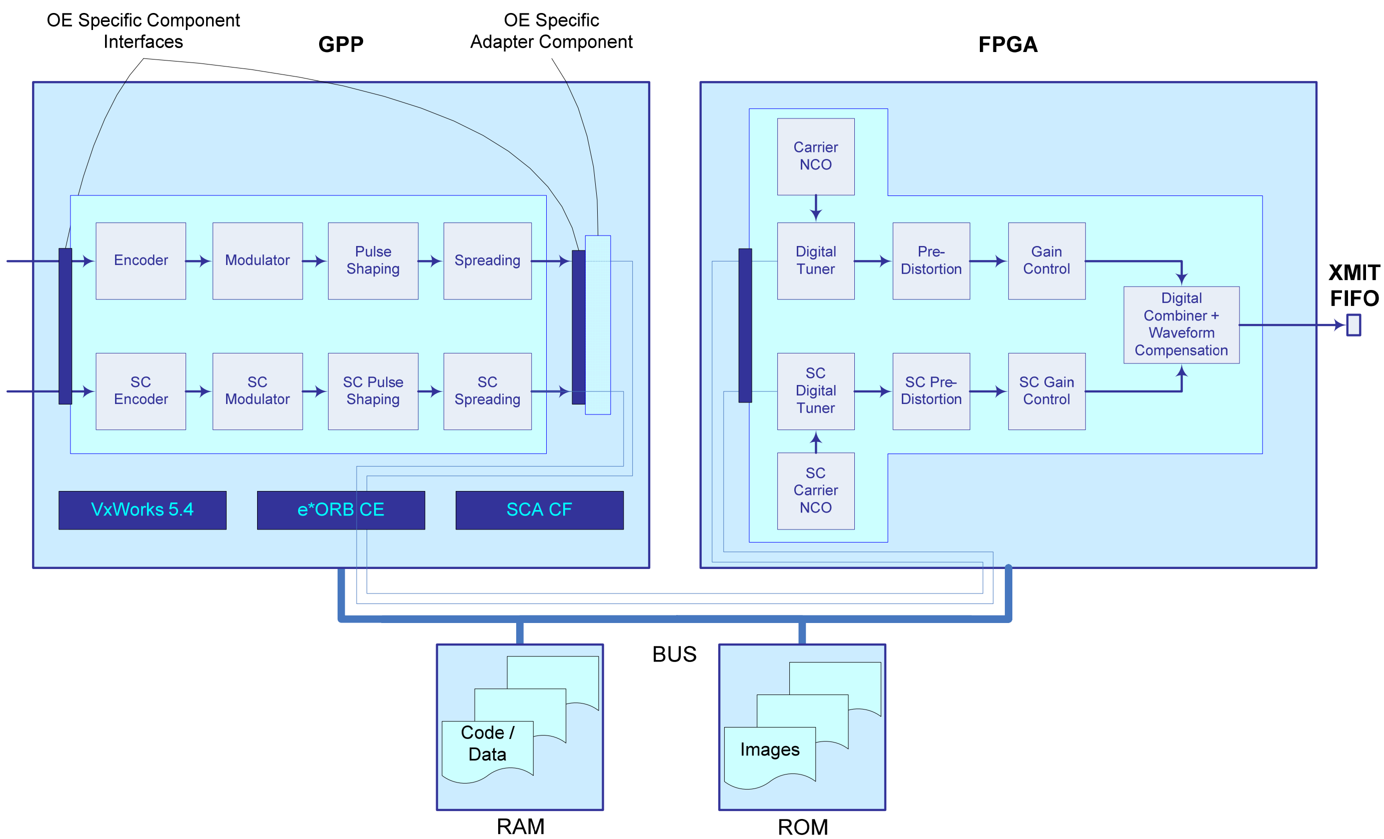
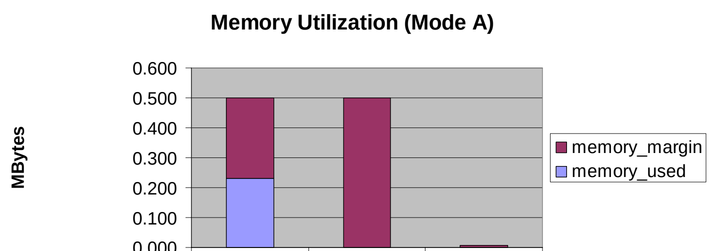


Figure 4.—"As-built" transceiver signal flow with architecture specific implications.



	fixed:ROM_1	fixed:RAM_1	fixed:GPP_1
memory_margin	0.268	0.500	0.008
memory_used	0.232	0.001	

Figure 5.—An example utilization chart.

embedded into the comment field of the cells of the Excel (Microsoft Corporation) spreadsheets that hold the models. These functionals take context sensitive parameters as inputs, which are selected from a pre-defined list of *keywords* defined in the STAT *dictionary*.

2.8. STAT Analysis Products

The overall goal of the STAT is to automatically generate analysis data that quantifies the resource utilization, size, weight, and power (SWaP), and modem latency for the as-built transceiver. This section describes the calculations that are performed for each type of analysis.

The utilization analysis creates a table and associated bar charts. The table contains a row for each combination of mission stage and hardware component. Each table row specifies the number of waveform components that have been mapped to that component for that mission stage, the amount of each resource that is utilized, and the margin for each resource. Bar charts are automatically built that plot the usage and margins for each resource. The resources are grouped in the charts by type (e.g., all memory resources are displayed in the same chart). For example, the chart in figure 5 shows one bar chart that is automatically generated by the STAT utilization analysis engine for a platform that containing three memory banks: a Read-Only Memory (ROM), a Random

Access Memory (RAM), and a bank of memory on a processor.

Table 1 and table 2 list the resource types and related units that are used in STAT utilization analysis.

TABLE 1.—RESOURCE UTILIZATION ANALYSIS PARAMETERS

Resource type	Unit
Generic operations	Million generic ops/sec (MOPS)
FPGA/ASIC gates	Kilo logic equivalents (kLEs)
Throughput	Megabytes/sec
Clock rate	MHz
Clock count	Number of clocks
Memory	Megabytes

TABLE 2.—SWAP ANALYSIS PARAMETERS

Attribute	Unit	Aggregation
Footprint	cm^2	None
Volume	cc (cm^3)	Summed
Weight	Grams	Summed
Power	Milli-Watts	Summed
NRE	Dollars	Summed
Cost	Dollars	Summed
Availability	Year	Maximum

The timing analysis creates a table that contains a row for each channel and stage. The table contains a calculated latency, a latency requirement (taken from the mission requirements model), and a resultant latency margin.

The automated utilization, SWaP, and latency analysis provide the mechanisms necessary to support efficient trade analysis and to allow the designer to quickly build and analyze alternative as-built transceiver designs for a mission. Trades can be performed to vary the channel parameters, the number and type of hardware components, the software infrastructure, and the mapping of waveform components to hardware components. By choosing representative examples for each infrastructure type, the impact of the various architectural approaches on total cost can be examined.

3. Example STAT Analysis

After exploring existing missions with moderately complex communications scenarios that were considered to be good candidates for SDR implementation, the STRS project team chose the Landsat 7 mission as a representative low earth orbit system. Choosing an existing system to model and analyze is advantageous as the analysis results can be compared to a known baseline (the parameters of an existing system) providing some level of validation. This section presents a trade study that uses the STAT to analyze three alternative implementations of the Landsat 7 mission.

3.1. LANDSAT Mission Description

The Landsat 7 satellite was launched on April 15, 1999. The spacecraft, depicted in figure 6, is an earth observing satellite designed for a sun synchronous 705 km, 16-day repeating orbit. The focus of the mission is to provide earth observation through a single nadir-pointing instrument, the Enhanced Thematic Mapper Plus (ETM+), a high-bandwidth imaging instrument.

The Landsat 7 satellite stores sensor data on the Solid State Recorder (SSR). The SSR holds 42 min (or approximately 100 images) of instrument data and 29 hours of housekeeping telemetry. The X-Band (8 GHz) frequency band is used for high-speed instrument data downlink, while S-Band is used for commanding and telemetry operations. The satellite is capable of transmitting an X-band downlink to ground stations located on the horizon of the earth, as viewed by the satellite. This transmission capability enables the satellite to downlink separate 150 Mbps data streams to as many as three ground stations simultaneously. Each data stream contains sensor image data, sensor calibration data, and Payload Correction Data.

The data rate at the input of each of the four X-band transmitters is 150 Mbps, consisting of I and Q channels of 75 Mbps each. The data is packetized using Consultative Committee for Space Data Systems (CCSDS) standards and carriers at three frequencies: High (8342.5 MHz), Low (8082.5 MHz), and Mid (8212.5 MHz). The transmitter output power is 3.5 W per transmitter.

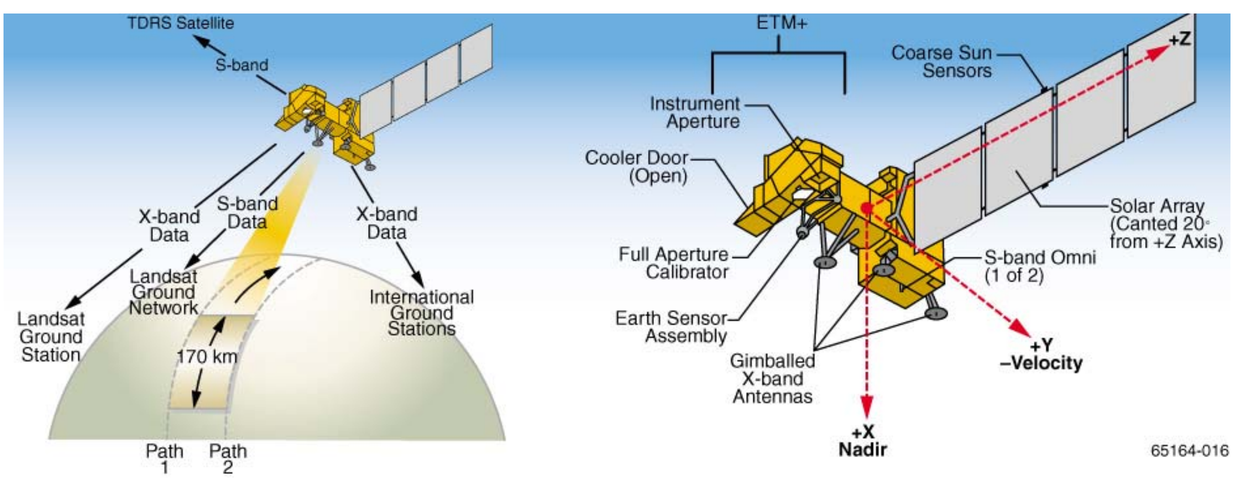


Figure 6.—LANDSAT mission scenario.

Of particular interest are the Telemetry Data Formatter (TDF), Payload Correction Data Formatter (PDF), and Baseband Switching Unit (BSU). The TDF encapsulates telemetry data into a specified format (CCSDS packets), and the PDF conducts a similar operation for image correction data. However, the ETM+ payload performs its own CCSDS packet formatting (see fig. 6). Finally, the BSU provides payload data to the X-band transmitters and TDF-formatted data to the S-band transmitter, either in real-time or from the SSR. It is noteworthy that all data formatting and coding are done externally, rather than in the radios. Also of note is the mass and power consumption of the TDF, which weighs 36 lb and consumes 9 W.

The radio frequency (RF) communications system provides S-band (narrowband) telemetry for housekeeping data and tracking ability. The S-band communications are conducted through two omni S-band antennas located on opposite sides of the spacecraft (nadir and zenith pointing). The zenith antenna is used for Tracking Data and Relay Satellite (TDRS) communications; the nadir antenna is used for Landsat Ground Network (LGN) communications. Narrowband data is captured by the SSR from the TDF. The SSR accepts two input rates of 1.216 or 4.864 Kbps and plays back stored telemetry data at 256 Kbps to the S-Band transponder.

3.2. LANDSAT SDR Trade Analysis

STAT was used to model the Landsat communications requirements and to perform an analysis to compare the size, weight, and power of implementations on three competing platforms. The first platform considered utilizes fuse-based FPGAs and is thus not reconfigurable. The second platform is a reconfigurable platform that does not support SCA but has a minimal, custom reconfiguration infrastructure. The third platform is a reconfigurable SCA compliant platform.

A sub-section of the Landsat 7 mission model table is shown in table 3. In order for the table to fit on the page, many of the details in the model are not shown.

TABLE 3.—PARTIAL LANDSAT 7 MISSION MODEL

Channel name	Waveform algorithm	Data rate (Kbps)
S-band ground downlink	S-band BPSK transmitter	256
S-band ground uplink	S-band BPSK receiver	2
X-band ground downlink1	X-band QPSK transmitter	153600
X-band ground downlink2	X-band QPSK transmitter	153600
X-band ground downlink3	X-band QPSK transmitter	153600

The following analysis only considers the stage of the Landsat 7 mission during which the X-Band Ground Downlink channels are active, as well as the S-Band Ground Up and Downlink channels. This is clearly the situation where the resource requirements will be the highest, as the X-band downlink channels are high data-rate waveforms. The transmitter and receiver waveform algorithms that must be simultaneously implemented by the Landsat Satellite include three X-Band QPSK Transmitters, an S-Band BPSK Transmitter, and an S-Band BPSK Receiver. Figures 7 to 9 show the Landsat waveform data-flows.

The waveforms were mapped to each of the three platforms: 1) a fixed platform with of fuse-based FPGAs, minimal RAM, and general-purpose processor; 2) a baseline reconfigurable platform with reconfigurable Xilinx FPGAs; and 3) an SCA compatible platform with the necessary additional processing and memory resources.

The STAT analysis results show that 1) the platforms can support the Mode A waveforms from a utilization standpoint, 2) the weight and power consumption for all designs are dominated by the RF components, not the digital components, and 3) reconfigurability has a non-zero cost. Referring to table 4, the baseline reconfigurable platform is predicted to

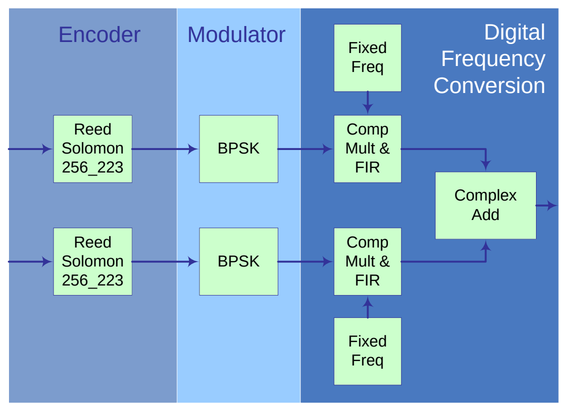


Figure 7.—S-band BPSK transmitter waveform.

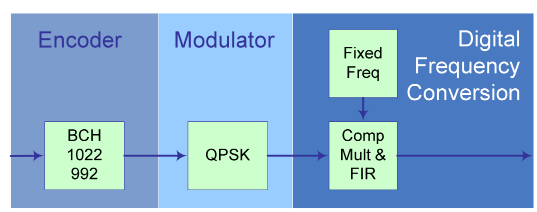


Figure 8.—X-band QPSK transmitter waveform

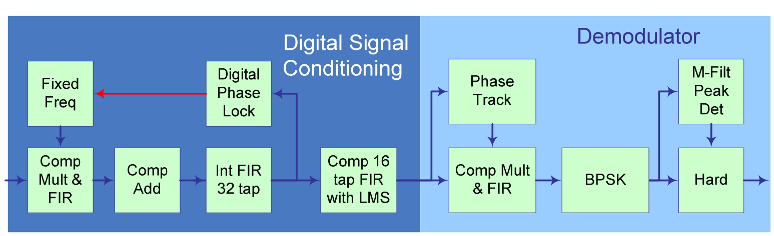


Figure 9.—S-band BPSK receiver waveform.

consume 10.4 percent more power than the fixed platform, with only a small increase in weight. That additional power is consumed by the reconfigurable FPGAs, and the size and weight are due to an additional board that is required to hold the larger, hotter FPGAs. The SCA platform is predicted to have significantly higher power consumption and size. In this case, the additional power is consumed by the large RAM bank (16 Mb). This RAM is required to support the SCA operating environment and the more powerful general-purpose processor (a RAD 750). Based on this analysis, the power consumption of the SCA platform could be dramatically reduced, either by obtaining a more power-efficient memory part or by creating an SCA operating environment that required far less memory (for example, 4 Mb).

TABLE 4.—LANDSAT 7 ANALYSIS RESULTS

Transceiver	Size (in.^{3})	Weight (lb)	Power (W)	Increase size %	Increase weight %	Increase power %
Fixed	75.7	38.3	122.8	0.0	0.0	0.0
Reconfig_small	81.9	39.4	135.6	8.3	2.8	10.4
Reconfig_large	94.6	41.5	170.2	25.0	8.4	38.6

4. Conclusions

As designers approach missions such as Landsat and more complex missions envisioned for future Exploration Systems, trades in communication capability and implementation must be judged by resource consumption and capability versus requirements. Software defined radios offer the promise of reconfigurability and flexibility. SDRs designed and built on an open architecture offer disparate vendor participation for both software and hardware elements and software reuse. The work conducted by GRC and its industry partners is a first step towards looking at alternative open architectures such as the JTRS SCA and defining an open architecture more unique to meet NASA’s needs.

While advancements are needed in reconfigurable logic and mission assurance concerns remain, the trend both within NASA and industry appears to be the continued use of reconfigurable FPGA-based platforms. According to the modeling results, both NASA and industry have accepted a 10 percent increase in resources for the benefit of FPGA-based reconfigurability for an optimized platform compared to a fixed, non-reconfigurable platform. The second result indicates an additional 26 percent increase in resources from an optimized reconfigurable platform to instantiate the SCA on a reconfigurable platform. This increase in resources is used to compare with the benefits and savings of design and software reuse and interoperability enabled by using the SCA. It also provides a ceiling from which comparisons can be made as the SCA is optimized and improvements or tailoring in third-party software products are made for space or other resource-constrained platforms such as handheld units. As NASA continues to investigate its open architecture for space, future analyses will be conducted to assess resource consumption for alternative architectures and implementations or alternative functional allocations and hardware architectures and designs.

STRS requires objective and quantitative analysis of the tradeoffs between various approaches toward implementing SDR to support the communications requirements of current and future space missions. It is important that architectural choices serve both small and large missions and NASA’s long-term interests.

An important observation of this study is that the STAT supports a thorough, objective, and quantitative analysis of the design alternatives for space transceivers. Often, when critical decisions are made about future design strategies (e.g., software versus hardware, middleware versus traditional software approaches, open standards versus vendor-centric designs, one language or software architecture versus another), both the lack of objectivity of vendors and the size and complexity of the design space can result in sub-optimal decisions. Quantitative comparisons of the costs and benefits of the alternatives are necessary for optimal design decisions. As real transceivers are implemented, the models that were used to predict the size, weight, power, and resource utilization of the design can be updated to reflect the measurements of the actual as-built system, thereby improving the fidelity of future analyses. As the STAT is used over time, the model database will grow and become more accurate and will become a key part of the transceiver design process.

5. Directions Forward

Advancements to the STAT may include an ability to conduct parametric analysis by automatically varying channel requirements or design choices such as waveform algorithm selections, hardware component properties, software to hardware allocation, and infrastructure properties. This could be accomplished by leveraging contemporary work in design space exploration (ref. 1) combined with the plans in (ref. 2) and extending the tool with the ability to automatically identify valid designs (those which meet requirements and have acceptable resource utilization margins) that optimize application specific metrics. Integration of STAT with such a design space exploration engine would enable the designer to more efficiently identify valid designs and identify where technology gaps occur, indicating possible investment opportunities. Other advancements to STAT may include interfacing to a commercial modeling and simulation tool such as Simulink (The MathWorks, Inc.) to provide a more general and configurable signal flow specification.

References

1. Brandon Eames, “Design Space Exploration,” Ph.D. Dissertation, Vanderbilt University, 2005.
2. James S. Schier, et al., “Space Communication Architecture Supporting Exploration and Science: Plans and Studies for 2010-2030,” 1st Space Exploration Conference: Continuing the Voyage of Discovery, Orlando, FL. 2004.

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)* 21-06-2007		**2. REPORT TYPE** Technical Memorandum				**3. DATES COVERED** *(From - To)*
4. TITLE AND SUBTITLE Modeling and Analysis of Space Based Transceivers						**5a. CONTRACT NUMBER**
						5b. GRANT NUMBER
						5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Moore, Michael, S.; Price, Jeremy, C.; Abbott, Ben; Liebetreu, John; Reinhart, Richard, C.; Kacpura, Thomas, J.						**5d. PROJECT NUMBER**
						5e. TASK NUMBER
						5f. WORK UNIT NUMBER WU-22-041-20-06
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-16020
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-2007-214828
12. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified-Unlimited Subject Category: 32 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 paper presents the tool chain, methodology, and initial results of a study to provide a thorough, objective, and quantitative analysis of the design alternatives for space Software Defined Radio (SDR) transceivers. The approach taken was to develop a set of models and tools for describing communications requirements, the algorithm resource requirements, the available hardware, and the alternative software architectures, and generate analysis data necessary to compare alternative designs. The Space Transceiver Analysis Tool (STAT) was developed to help users identify and select representative designs, calculate the analysis data, and perform a comparative analysis of the representative designs. The tool allows the design space to be searched quickly while permitting incremental refinement in regions of higher payoff.						
15. SUBJECT TERMS Software defined radio; Modeling						
16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES** 16		**19a. NAME OF RESPONSIBLE PERSON** Richard C. Reinhart
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U				**19b. TELEPHONE NUMBER** *(include area code)* 216-433-6588