Comments on NASA Space Telecommunications Radio System (STRS)

Open Architecture Specification

Approved 8 November 2007

SDRF-07-W-0013-V1.0.0

Table of Contents

1. Introduction

The Software Defined Radio Forum (SDRF) is pleased to provide comments and recommendations to the National Aeronautics and Space Administration (NASA) on Version 1.0 of the Space Telecommunications Radio System (STRS) specification, released December 2005. These comments are respectfully submitted to NASA Glenn Research Center for consideration of incorporation into future versions of the STRS specification.

The comments and recommendations contained herein have been prepared by a cross section of industry personnel as part of the Space Working Group (WG) at the SDRF. As a key element of its charter, the Space WG is intended to supply the Software Defined Radio (SDR) community with a venue to evaluate and provide commentary and recommendations for change proposals the specification. This document represents the first such submission to NASA Glenn Research Center on behalf of the SDR Forum.

1.1. STRS Background

The STRS was initiated by NASA to define a standard architecture for space-qualified radios in support of future NASA missions. The objectives of the STRS are to:

1. Support near-term communications needs and enable support for new communications needs as the nature and scope of NASA's missions evolve,
2. Provide a common open architecture definition across NASA missions and services,
3. Promote reuse of hardware and software components,
4. Provide an initial set of waveform Application Programmer Interface (API) descriptions, and
5. Define an initial Hardware Interface Description (HID) as a baseline hardware abstraction.

Industry feedback was initially provided at a workshop NASA in February 2006. Considerable discussion has been held regarding the use of existing standards, specifically the Software Communications Architecture (SCA) developed as part of the Joint Tactical Radio System (JTRS) program and the Software Radio (SWR) specification developed within the Object Management Group (OMG).

1.2. Document Organization

The document starts with more general high level comments and recommendations concerning standards and process in sections 2, 2.4 and 3. Then sections 4, 4.3, and 6 have more specific recommendations and technical discussions. Please note that section 6's detailed discussions are preceded by a summary in section 4.3. Contributing member companies are listed in the acknowledgements. And finally, an appendix contains an STRS acronym list for reference.

2. Standards Organization Involvement

There are several paths that may be taken to developing a Space SDR standard. These paths are dependent on the focus and scope of the specification. There are three key components to standards development that have been considered: 1) Requirements, 2) Validation, and 3) Process. These components have become core competencies in different standards development organizations, including the SDR Forum, the Object Management Group, and the Institute of Electrical and Electronic Engineers (IEEE).

"Each of these groups[1] and their respective initiatives have high potential of giving new birth to the SDR and Cognitive Radio (CR) technologies, including requirements, technology development, and standards processing. It is recognized that, conflict and cost will both rise without agreeable coordination. The benefits of convergent opportunity with fresh perspective, with more minds focused on common objectives, each contributing to a common SDR/CR standards set based on their core competency(ies) are significant" [2].

To that end, leads of these groups meet periodically to discuss methods to improve SDR standards collaboration. The groups are now actively exchanging memberships, engaging liaisons, and forming a common process to enable collaborative standards development.

The strength of the SDR standards collaboration, combined with the collective expertise and core competencies of these several consortia are positioned to distribute the burden of non-recurring expenses (NRE), reduce recurring expenses (RE), and improve the breadth and depth and quality of the SDR standards that will be openly available to NASA.

The following subsections provide additional background on these organizations.

2.1. Software Defined Radio Forum[3]

NASA has solicited the expertise of the SDR Forum to review the STRS Architecture, and to provide recommendations for improvements to this architecture. The SDR Forum has responded to NASA's request, and expresses interest in continuing to provide expertise and perspectives to NASA for further development and refinement of the STRS Architecture. The SDR Forum membership has a substantial pool of knowledge related to the analysis, design, and development of software radio systems. In particular, system level knowledge is critical in order to capture the systems constraints and radio-specific knowledge that must be part of the foundation of a software specification of the Space SDR.

The requirements levied by space deployment drive certain key areas of the overall STRS architecture. They include, but are not limited to traditional areas such as Size, Weight, and Power (SWaP) and tolerance to radiation effects experienced in space. In fact, each mission typically imposes requirements that add additional unique considerations on communication systems. These constraints typically affect a systems level view of the SDR as opposed to a software-only view. The SDR Forum helps derive, analyze and recommend these and other requirements through broad industry consensus rendering further strength and quality to the STRS initiative.

[1] Refers to the OMG, SDR Forum, IEEE SCC41, the NCOIC and others
[2] Article: The Software-Defined Radio & Cognitive Radio Inter-Consortia Affiliation, Mark Scoville, Stephen Berger, Richard C. Reinhart, Dr. Jeffery E. Smith (http://www1.coe.neu.edu/~jsmith/Publications//MILCOM2006.pdf)
[3] http://www.sdrforum.org

2.2. Object Management Group [4]

"The OMG is dedicated to solving complex industry problems through the development of software specifications. OMG members develop these specifications through a mature, proven technology adoption process.[5] That process is summarized in the Hitchhikers Guide,[6] that serves as an aid to navigating through and complying with the OMG technology adoption process, and is an interpretation of the formal OMG Policies and Procedures document. The RFI, RFC, and RFP processes are key in the OMG technology roadmaps. Organizations, including other consortia, contribute to and have voting privilege on specification development and approval".[7]

OMG Task Forces, through this well-defined process, develop standards for a wide range of technologies and industries. These include well-known standards such as: the Common Object Request Broker Architecture® (CORBA®), the Unified Modeling Language™ (UML®) and Model Driven Architecture® (MDA®).

This proven methodology combined with the expertise of the SDR Forum brings process and requirements for SDR together, and creates a reliable roadmap for STRS development and evolution.

2.3. Institute of Electrical and Electronic Engineers[8]

The SDR Forum recommends that the STRS align with IEEE standards where applicable. There are multiple groups within the IEEE that could be considered as information sources and participants in the STRS efforts. The first to note is the new Standards Coordinating Committee (SCC41, formerly P1900). The SCC41, created in March 2007, has increased authority and control over the P1900 efforts. The objective of the SCC41 (IEEE 1900)[9] is to develop supporting standards dealing with new technologies and techniques being developed for next generation radios and advanced spectrum management. This includes the standardization of terminology and concepts, which is key in coordinating the standards development within NASA and the consortia circles.

An additional IEEE area considered is the 1003.1 Portable Operating System Interface (POSIX) standard. However, there is not consensus in the SDR Forum's Space WG (with regard to STRS) how much to leverage this standard.

2.4. Process and Relationships

The relationships between existing specifications, organizations, and how these may be leveraged as part of the STRS development are illustrated in Figure 2-1 below.

[4] http://www.omg.org
[5] http://www.omg.org/memberservices/TechAdoptProcess.pdf (Brief overview of the OMG process)
[6] http://www.omg.org/cgi-bin/doc?hh - The Hitchhikers Guide can be downloaded from this location.
[7] Article: The Software-Defined Radio & Cognitive Radio Inter-Consortia Affiliation, Mark Scoville, Stephen Berger, Richard C. Reinhart, Dr. Jeffery E. Smith (http://www1.coe.neu.edu/~jsmith/Publications//MILCOM2006.pdf)
[8] http://www.scc41.org/
[9] http://grouper.ieee.org/groups/emc/emc/1900/files/1900_Committee_Overview.ppt

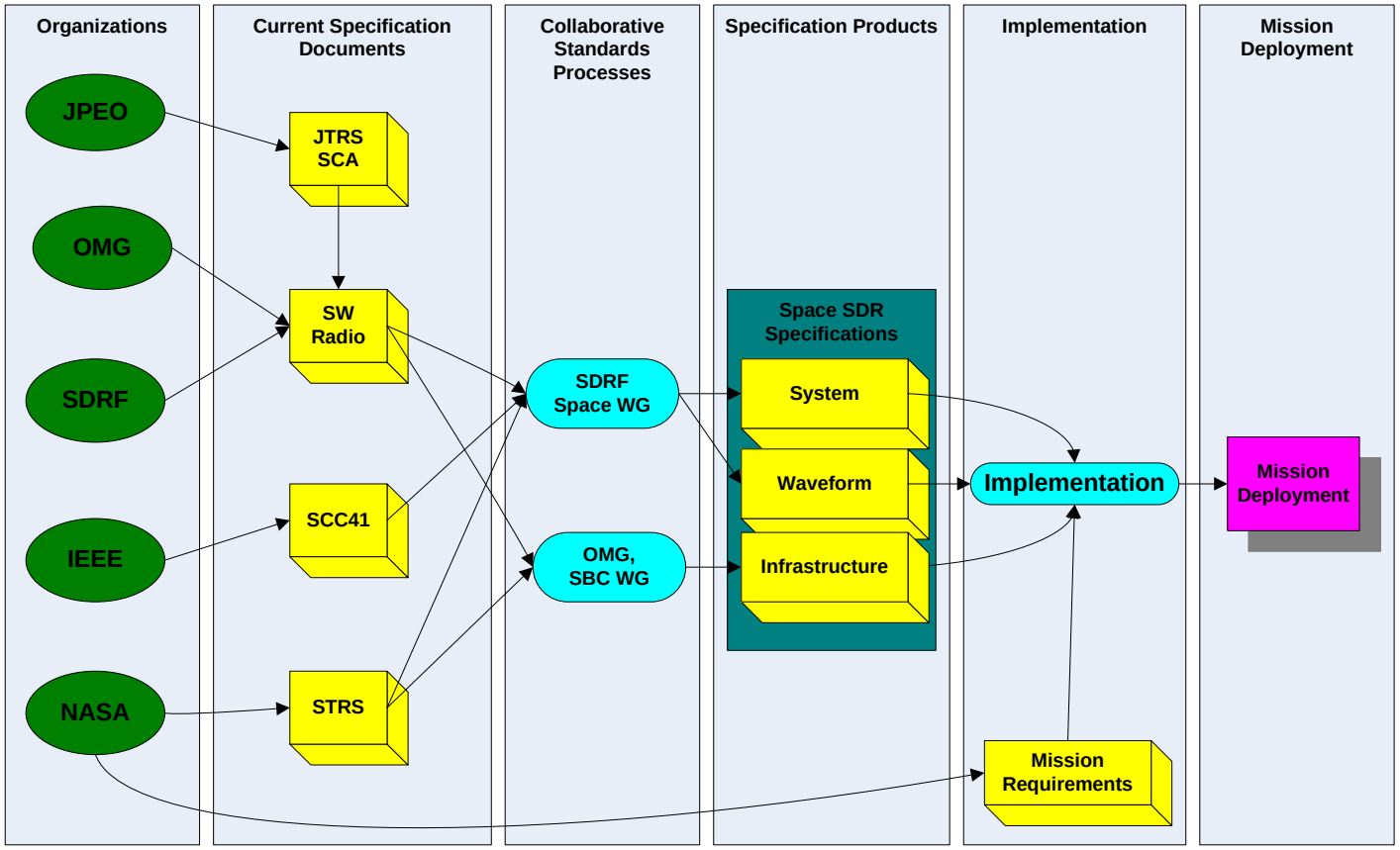

Figure 2-1 Candidate Specification Process and Relationships

As shown in the figure, existing specifications and current work to date provide foundational components upon which a comprehensive space SDR specification may be developed.

3. General Recommendations

The following sections identify general questions and provide recommendations regarding the STRS specification and process.

3.1. Open Architecture

The consensus of the Space WG is that the STRS should continue to evolve towards an open standard rather than a NASA unique standard. An open standard would promote wider acceptance and relieve NASA of the entire burden of maintaining a standard, while still allowing NASA to influence the content and direction. Furthermore, the development of an industry standard would provide a forum for wider contributions and comments.

The SDR Forum recommends that the STRS align with the SDR Forum, the OMG, and the IEEE SCC41 for purposes of distributing the burden and cost of non recurring engineering (NRE) across NASA and all consortia members contributing to the STRS, and to further broaden and enhance the quality of the implementation and deployment of STRS-based standards. Such quality will be realized through the development of tools implementing STRS modeling which is viable because of the market that is created based on the collaborative efforts of multiple consortia working together to establish Space SDR standards.

While the STRS defines a software radio infrastructure, it will be deployed as part of the mission critical communications system of the space vehicles. The SDRF does not believe that the specification will require International Trafficking and Arms (ITAR) restrictions, much as the SCA developed as part of the Joint Tactical Radio System (JTRS) is an open standard. However, the SDRF recommends that NASA be proactive to ensure that the STRS will be an open standard.

3.2. Leverage Existing Standards

With the combination of the standards development process and the mindshare of industry in the SDR Forum, OMG, and IEEE standards development organizations (SDOs), NASA has identified a powerful collaborative core competency that it does not have to duplicate. The NASA/SDO affiliation relieves NASA of the time consuming, costly, and very complex responsibilities of standards development and maintenance, (i.e., the SDO Business Model) allowing NASA to be both surgical and comprehensive in its business of Space Communications applications – and still strongly influence the standards process.

There are several paths that may be taken to developing a Space SDR standard. These paths are dependent on the focus and scope of the specification. There are three key components to standards development that have been considered: 1) Requirements, 2) Validation, and 3) Process. These components have become core competencies in different SDOs (including the SDR Forum, the OMG, and the IEEE P1900), and are all being leveraged by NASA.

a. **Organizations & Initiatives**: SWRadio, IEEE 1900.1, NCOIC MBWG, 1900.4 (Networking), Open Group – Open Architecture Framework (Enterprise), DODAF, AFRL – Common Bus for the space mission domain (USB)
b. Vendors of standards products: http://www.omg.org/mda/committed-products.htm, http://www.cetus-links.org/oo_object_request_brokers.html,

c. OMG Specification Summary:
http://www.omg.org/technology/documents/spec_summary.xls

The OMG's SWRadio specification[10] is focused on the portability of waveforms across software defined radios. It does this by adding communications and Open Systems Interconnection (OSI) components and facilities and a model/technology separation to the SCA[11]. Additionally, it creates a new standardized UML profile for the software radio. Considerable effort has already been invested in the development of these OMG standards, and there is benefit to leverage these efforts to the greatest degree possible where applicable for the space domain. The potential to leverage the existing software radio specification provides a valuable starting point.

3.3. Develop an Integrated Set of Specifications

Segregating the specification into a cohesive set of specifications covering the system, infrastructure, and waveform would provide both complete coverage of the Space SDR system and promote more concise and clear definition of each of the areas by limiting assumptions and implementation approaches within the specification document and forcing a specific set of interfaces and protocols to be defined.

The Architecture Description Document that was released and reviewed at the NASA STRS industry day currently addresses the Software Infrastructure, Hardware Architecture, and Waveforms within a single specification. Consequently, the specification has tightly intertwined dependencies and implementation assumptions within each of these three areas. While there are certainly dependencies and relationships between each of these components that must be addressed in a comprehensive SDR design, each area should be developed as an independent specification to the greatest extent possible.

For example, the software infrastructure specification should focus on those aspects of the SDR infrastructure that must be provided, e.g. overall radio management, waveform loading, Built-in-Test (BIT), waveform management, etc. The hardware specification should define the system level requirements, physical interfaces and protocols between radio system components, and other physical requirements and constraints. Similarly, the waveform specification should define the interfaces and control points it must provide to the infrastructure and the target processor architectures and constraints.

1. **System Context/External Interface Specification –** A concise description of the system context and external interface specifications for the Space SDR should be provided as a top-level system document. This would provide the form, fit, and function information for the radio system integration with the space platform on which it is to be located.

2. **SDR Systems Specification**: The Systems specification provides a systems view of the radio. This captures radio system requirements, use cases, and quantitative information regarding the space SDR.

[10] Enhancing the Platform Independent Model (PIM) and Platform Specific Model (PSM) for Software Radio Components Specification Version 1.0 – a.k.a. SWRadio specification – (OMG document number: dtc/06-04-17, et.al)

[11] The goal of the SCA (as stated in the SCA 2.2.2 specification) is to provide for the deployment, management, interconnection, and intercommunication of software components in embedded, distributed-computing communication systems.

Industry is increasingly moving towards representation of systems and software using System Modeling Language (SysML) and UML respectively. Collectively these modeling languages provide the basis for modeling tools (Rose, Rhapsody, Enterprise Architect, and many others) which facilitate formal methods to create systems and software views of the radio system. This recommendation does not preclude the use of general block diagrams produced by tools like Visio, but does encourage SysML and UML to produce views that are subject to standards-based interpretation, which is likely to produce implementations with fewer requirements or interpretive errors.

The recommendation therefore is that the STRS should be provided using SysML or an equivalent system modeling language. SysML has the advantage of integrating with the UML thereby providing a common, integrated model approach for defining both systems and software engineering aspects.

3. **Software Infrastructure Specification:** The software infrastructure should be defined by the management infrastructure and services provided by the space SDR. This should be developed using the UML. The UML specification of the management infrastructure should be linked to the SDR system specification identified above.

4. **Waveform Implementation Guideline:** The waveform specification should define the parameters, guidelines, and constraints that should be followed when developing a waveform for the space SDR. The waveform specification should have a format of an Interface Definition Document (IDD), which has a dependency on the system specification to identify baseline processing capabilities, interconnections, and protocols and the infrastructure specification to identify the common waveform control interfaces. It will also be dependent on the actual radio system for a particular mission. Thus, this will be a top level specification from which specific implementation specifications must be derived based on the actual radio system.

3.4. STRS Architecture Conformance

In considering the factors that constitute conformance with the STRS Architecture, the relevance of a variety of market and application concerns is recognized. In general, the requirements for conformance to or compliance with the STRS Standard should not imply or mandate an explicit or de facto implementation, or force or promote reliance on a design tool set or design process. The primary criteria for assessing conformance/compliance should be satisfaction of the behavior specification of the STRS APIs. One implication that emerges from this conformance philosophy is relaxation of the commitment to mandate POSIX as a requirement for conformance to STRS. Although many of the capabilities of POSIX can benefit STRS architecture implementations, this is not the case for every application. A mandate to conform to POSIX in every instance precludes highly-efficient implementations where the application does not require POSIX; forcing the implementation simply diminishes its efficiency. Furthermore, a mandate has the undesirable impact of forcing platform vendors to select from a very limited set of RTOS solutions that provide POSIX for their platform, and may in turn inhibit innovation and slow the adoption of STRS. Without a clearly identifiable return on investment, space radio providers are not expected to expend resources to add a POSIX interface to an existing platform. It is much less costly to simply add the abstracted STRS APIs that are used for creating and deleting task resources, especially considering that adding the POSIX interface requires expertise outside their core competency to capture share in a comparatively small market segment.

3.5. Reduce Review Cycle

It is recommended that NASA reduce the time to respond to industry input and release of documents related to the space software radio specifications. This will promote more timely input and feedback from industry and standards organizations and help achieve the deployment of the technology in the time-frame required for future missions.

4. Specification Recommendations

The following recommendations address areas that relate to specific technical items associated with the STRS specification.

4.1. Define External Control Functions

Specify the top level control capabilities required for configuration, control and monitoring of a Space SDR. This would provide initial guidance to the development of a standard for the infrastructure and facilitate achieving short-term NASA objectives.

4.2. Extend the HAL Concept

The waveform implementation and deployment is illustrated as a component-based implementation that may be deployed across the suite of processing resources within the Space SDR.

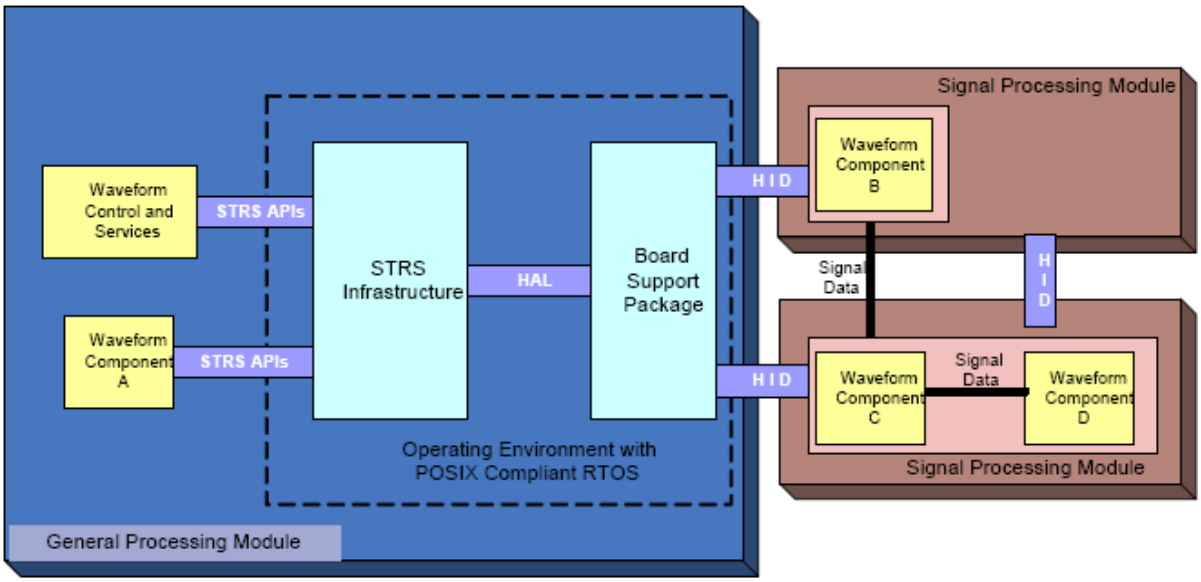

Figure 4-1 STRS Waveform Component Distribution

As illustrated in Figure 4-1 above, the hardware abstraction is provided only within the General Purpose Processing Module (GPM). In order to promote waveform portability, interconnection abstractions for data and control must also be provided as part of the waveform components in the Signal Processing Module (SPM). If this is not provided, the cost of waveform porting will be driven up significantly.

4.3. Security

The STRS Architecture should address controls to mitigate the risk of unauthorized reconfiguration of STRS radio software. The threat is unauthorized access of a spacecraft's communications functionality from Earth, which could change the spacecraft's performance characteristics or even render it unusable. Countermeasures

could include authentication, code signing, etc. The SWG recognizes the extensibility of the STRS architecture to include additional services to support security controls.

5. Selected Topics Highlights and Recommendations

The following is a summary of the recommendations resulting from the Space Working Group selected topic analysis and discussions. The background and details on which these recommendations are based can be found in section 6.

5.1. Module Definition Cautions

This review shows that most of the module definitions provide sufficient detail to support interface discovery and abstraction, with the exception of the SEC Module, where further detail is required. Module partitioning is defined in a manner sufficient to support interface abstraction. Modules are suitably defined in the STRS architecture as logical divisions of functionality, but it should be highlighted that these logical divisions do not imply a physical implementation, nor do they imply a physical partitioning.

5.2. Change Mission Class to Radio Class

Consider renaming “Mission Class” to “Radio Class”. The term “Mission Class” speaks to the type of spacecraft or application. However, a spacecraft may have L, M and H class communication equipment according to mission requirements. The term “Radio Class” seems to better describe the level of capability of a particular communication sub-system.

With advances in FPGA technology, it may be unnecessary to restrict the type of FEC coding capability in the class L platform class.

Existing L and M platform class radios have transmitter power capabilities > 10 watts. The STRS transmit power specification seems low for the L and M classes. Consider having a high and low power subset for these classes. For example, M2-H may indicate medium class high power >10 W.

Existing L platform class radios have transmitter and receiver bandwidths of 8 MHz and the capability to transmit and receive 4 Mbps. The STRS specification for the L platform class radio is restrictive and should be expanded.

The STRS specification has H class radios with transmit bandwidth of 600 MHz and data rates up to 1000 Mbps but the maximum receive bandwidth for this class is 50 MHz. Transmit and receive bandwidths and data rates should match in the H class to accommodate reception at the highest rates. Consider expanding the receive bandwidth for H class radio.

Generally, H class should have hardware interface definitions based at the Card or Module level while Small/Medium Radio Class definition should be at the Radio Level.

5.3. Add Hardware Interface Characterizations

It is recommended that hardware interface characterizations of section 6.3 be incorporated into the STRS Architecture. They should be continuously reviewed and revised to remain current with advances in applicable technologies and is relevant to advances in software defined radios and space communications.

5.4. Enabling Reliability

The STRS Architecture APIs enable vendors to implement reliability based on mission requirements. This approach mandates that modules support diagnostics and reporting mechanisms to validate their operation.

The STRS Architecture has the inherent capacity for robust fault detection and management. The STRS interfaces can be used to distribute or propagate fault information under the supervision of a Fault Management Service Application operating via the STRS Infrastructure, APIs, and HAL. Such a Fault Management Service Application should encompass fault alarm generation, fault alarm polling, or other means of fault alarm detection.

Similarly, a STRS Service Application can be developed to manage redundancy between multiple radios or radio modules. Equipment vendors should specify the reliability and degree of support for redundancy on a module or radio-system level for each discrete equipment item.

5.5. HID and HAL

Emphasize that vendors are responsible for publishing the HID. Each vendor will provide the Hardware Abstraction Layer (HAL) device driver that provides the logical (software) to physical (hardware) interface layer to integrate the module into the STRS architecture.

5.6. Provisions for Technology Insertion

In order to enable advancement of SDRs in space it is recommended that NASA consider engaging in the following activities in the area of electronic device capabilities:

1. Develop and publish a NASA roadmap for space processor needs and developments
2. Accelerate technology readiness level with an on-going radiation test/evaluation program
3. Develop and fly a technology flight test vehicle to secure flight history for these technologies
4. Identify conditions under which commercial or military parts may be used for space environment
5. Identify space-proven parts categorized by radiation performance

6. Selected Topics Review

Part of the Space Working Group's review of the STRS Architecture v1.0 involved hardware centric topics. These were formulated at a workshop held with NASA in August 2006, as being of specific interest for the Working Group. NASA requested that these comments be responsive to the constraints of space flight hardware and systems (e.g., size, weight, and power of space radios are highly constrained compared to terrestrial systems), and reflect the heritage and culture of space flight. The group focused on the following topics:

Module Definitions – The review and assessment has been conducted to determine whether the module definitions and logical partitioning are appropriate to support identification and abstraction of interfaces, and to determine whether the set of modules identified in the STRS Open Architecture Description document is sufficiently complete.

Platform class mapping to requirement sets – This task was comprised of mapping mission platform classes to typical NASA applications to validate the preliminary STRS Architecture Description Document class definitions. Accordingly, mission requirements, applications, and class definitions were reviewed based on the NASA mission matrix summary[12]

Interfaces – Assessments of the STRS Hardware Architecture interfaces were made to identify a representative set of interfaces (both internal – between modules, and external – to other subsystems), evaluate them for completeness, contemplate the possible role of existing standards in expressing the requirements for describing these interfaces, and assess whether the level of interface descriptions in the STRS Open Architecture Description document is appropriate.

Redundancy/reliability requirements – The task group focused on a close examination of the STRS Hardware Architecture to confirm that it is consistent with design for reliable operation. The STRS Hardware Architecture was assessed in the context of two key dimensions of a reliable design: fault detection/recovery, and redundancy.

Hardware Abstraction Layer – The task group assessed the tradeoffs, potential benefits, and derivative requirements of the STRS Open Architecture approach to hardware abstraction and hardware interface definition.

Survey Device Capabilities – The task group was asked to assess current space-qualified electronic device technology capabilities, limitations, and roadmap gaps relevant to SDR technology needs and bring forth recommendations for action to enable advancement of space SDRs.

Comments and discussion for each of these areas is in the following subsections. Each topic concludes with related recommendations for the STRS Architecture.

[12] "SDR Application Trade Study," D. Israel and W.L. Thompson, Goddard Space Flight Center, October 12 – 2005.

6.1. Module Definitions

6.1.1. STRS Architecture / Interfaces / Hardware Module Definitions

The STRS Architecture Standard defines external radio interfaces and also describes the connections between radio components. The STRS open architecture definition identifies interfaces and applies rules for the hardware and software to realize the benefits of SDR. The radio functions are distributed among different modules, to organize platform services and waveform functions within the radio. The GPM (General Purpose Module) is a required module within the STRS radio that supports execution of the software-based operating environment. This environment is responsible for waveform instantiation and execution, certain radio services, and hardware abstraction. The SPM (Signal Processing Module) conducts high speed data and signal processing, clock distribution, and may provide an external interface to the payload data, for example, a SpaceWire interface. The RFM (Radio Frequency Module(s)) provide RF front-end functions for waveforms anticipated to operate in the UHF, S, X, Ku, and Ka- frequency bands, which are allocated for use by NASA in space. These modules are depicted in Figure 6-1, with overlays depicting the operating environment, waveform components, and interfaces.

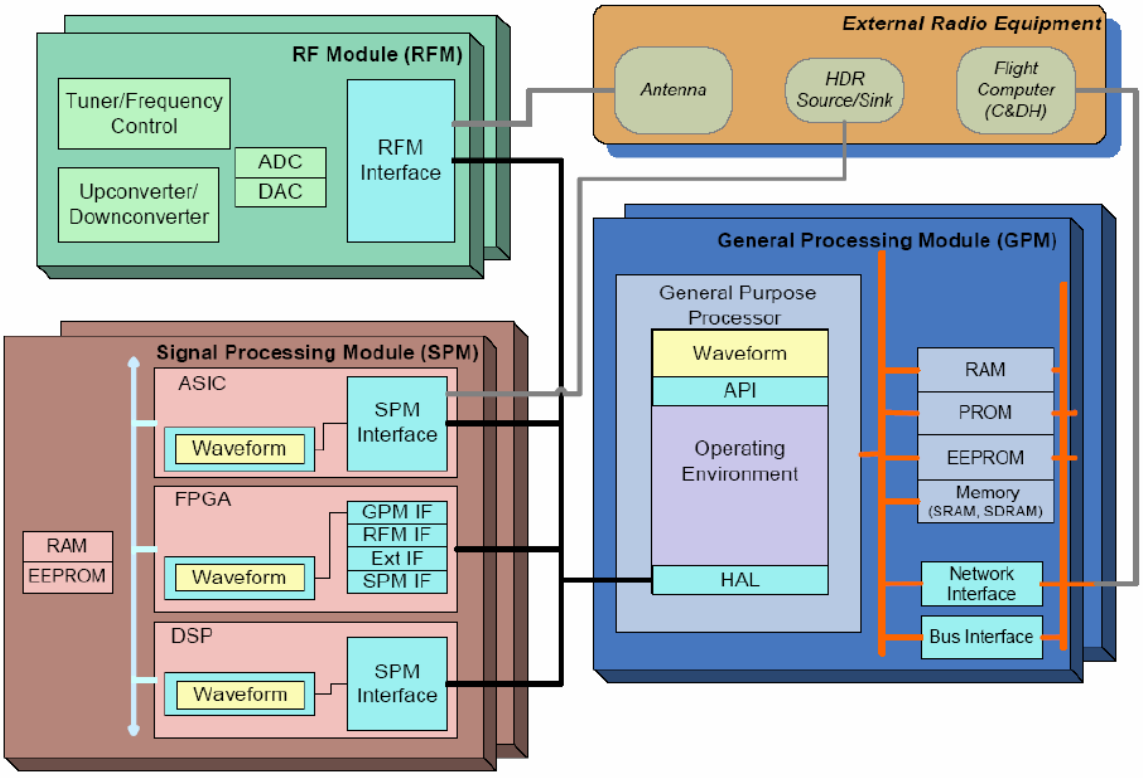

Figure 6-1. STRS Hardware Architecture Module Definition

Modules are defined in the STRS architecture to be a logical division of functionality. The Hardware Architecture Review highlights that this logical division does not imply a physical implementation, and explicitly stipulates that the modular logical definition is not a physical partition. Modules are identified in STRS to maintain common interface descriptions, terminology and documentation among SDR developments. The requirement for STRS Modules is that they are sufficiently defined to support interface

discovery and abstraction. Review of the STRS Modular logical partitioning shows that this is the case for the module definitions and descriptions identified in the STRS Architecture Description Document.[13]

6.1.2. Security Module Definitions and ArcFigure 6-2hitecture Impact

The imposition of a Type 1 security module, and its impact on the other modules, is depicted in Figure 6-2.

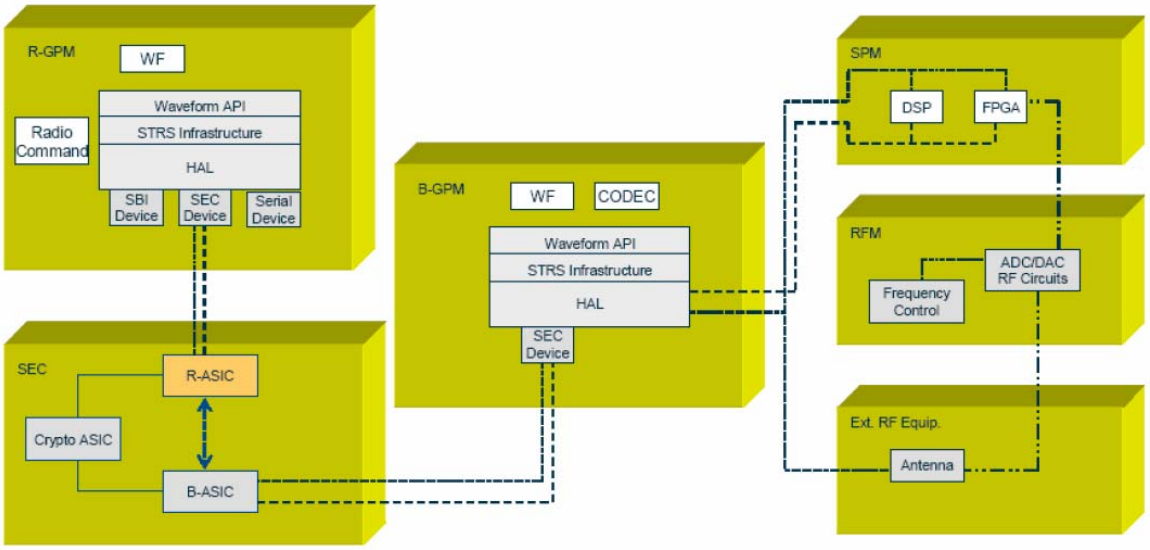

Figure 6-2 STRS Security Module Impact

The STRS Architecture Description Document identifies two security types for STRS, with a statement about the architecture impact for each:

> Type I cryptography is a specification that provides requirements and behaviors for handling distribution of classified material. It requires a distinct separation between the encrypted and unencrypted data transfer bus. To achieve the Type I requirement, a separate security module, as well as separate radio backplane must be developed.
>
> Type III cryptographic messaging can be on a single CPU bus, but would still be required to meet certification requirements from NIST. Thus, a separate Security Module and backplane would not be required. The requirements for Type III are specified in the FIPS140-2 documentation.

SEC Module Interface Development Parameters are identified as TBD.

Review of the impact of security module insertion into the STRS Architecture confirms that there is sufficient definition to support interface discovery and abstraction, although the STRS Open Architecture Description does not have a complete description of the impact depicted in Figure 6-2, and the SEC Module Interface Development Parameters should be included. It is recommended that the diagram of Figure 6-2 or a similar diagram be included in the STRS Open Architecture Description, and that the section describing SEC Module Interface Development Parameters be completed.

[13] NASA, "STRS Open Architecture Description, Release 1.0", Cleveland, Ohio, December 12, 2005

6.1.3. Recommendations

The Hardware Architecture Review addressed several aspects of module definition: 1) whether the STRS modules are sufficiently defined to identify and extract interfaces, 2) module partitioning, and 3) completeness of the module descriptions in the STRS Open Architecture Description document.

This review shows that most of the module definitions provide sufficient detail to support interface discovery and abstraction, with the exception of the SEC Module, where further detail is required. Module partitioning is defined in a manner sufficient to support interface abstraction. Modules are suitably defined in the STRS architecture as logical divisions of functionality, but it should be highlighted that these logical divisions do not imply a physical implementation, nor do they imply a physical partitioning.

6.2. Map Platform Classes to Requirements Sets

6.2.1. General Discussion

The task of mapping mission platform classes to typical NASA applications attempts to validate the preliminary STRS Architecture Description Document class definitions.

The preliminary document identifies five mission platform classes. The currently defined classes include:

- Class L: Low intrinsic complexity and low data rate signals
- Class M1: Moderate complexity and medium data rate signals
- Class M2: Moderate complexity with at least one high-data-rate transmit signal
- Class H1: High functional complexity with mixture of low, medium, and high data rate signals
- Class H2: Same characteristics as H1 with at least one ultra-high data rate transmit signal

The L class describes a single band transmitter, receiver or transceiver communication sub-system with a typical receive bandwidth of 1 MHz, transmit data rate of 2 Mbps, transmit power 3 W, simple FEC (e.g., convolutional FEC), low rate network interface, receive command authentication and legacy relatively low rate spacecraft interfaces.

M1 and M2 class radios are basically identical with operation in two RF bands, typical maximum receive bandwidth of 4 MHz, typical maximum transmit data rate of 20 (M1) or 100 (M2), transmit power 5 W, multiple FEC capabilities, receive command authentication, transmit encryption, low to high rate network interfaces and legacy to more contemporary spacecraft interfaces.

The highly capable H1 and H2 class radio sub-systems extend to operation in four or more bands, typical maximum receive bandwidths of 50 MHz, typical maximum data rates of 100 (H1) or 1000 (H2) Mbps and the additional features as described in the M class radios.

6.2.2. Applications Definitions and Examples

Virtually every NASA application requires a communication sub-system for spacecraft to earth and/or spacecraft to spacecraft communication. In order to map platform classes it is useful to describe the wide range of NASA applications. Following is a listing of potential applications for future STRS compliant communication equipment:

Robotic Spacecrafts: Spacecraft with some level of autonomy that are typically used for exploration. Examples include Voyager 1, Cassini - Huygens, Deep Impact

Rover / Surface Elements: Spacecraft that operate on the surface of bodies other than earth. Examples include MER Spirit and Opportunity, Phoenix and Lunar Rovers.

EVA Radios: Extra Vehicular Activity Radios used for communicating between an astronaut and a ship or orbital platform station during a space walk.

Orbiting Relays: A craft in orbit that relays communications such as TDRSS, MRO and Iridium.

Launch Vehicles: A vehicle used to accelerate a payload and/or astronauts into space. Examples include Ares, Orion, Shuttle, Atlas and Delta.

Sub-Orbital Vehicles: A spacecraft that remains in space for less than one orbit like the SpaceShipOne.

Space Stations / Outposts: An artificial structure designed for humans to live such as the ISS, Skylab or future lunar bases.

Ground Stations: Provide telemetry, tracking, and control of spacecraft from earth based communication centers. Examples of ground stations include KSC, JSC, LGS, DSN, and White Sands Ground Terminal.

6.2.3. Small Mission Class (L) Criteria

The Small Mission Class communication sub-systems will be utilized on low mass, power constrained applications such as Robotic Spacecrafts, Rover/Surface Elements and EVA Radios.

Communication equipment required for this type of mission is typically custom designed with the primary driving factors being mission communication requirements, reliability and SWaP. Applying SDR to radios of this class will have the advantage of providing a common platform between missions, mission reconfigurability, phase reconfigurability, upgradeability, autonomous operation, and advanced DSP integration.

Typical characteristics of the Small Mission Class radio include:

- Single Vendor Radio
- Minimal Scalability
- Non-Standard Form Factor
- Non-standard Hardware Architecture
- Vendor Publishes Abstract Layer Interface
- Hardware Interface Definition at Box, Module, or Component Level
- Highly Constrained SWaP
- Small Footprint (Probe/Rover)
- Single Frequency Band Supported
- Few Duplex Links
- Data Rates from Low to Medium

Review of the STRS Architecture Description Document for L class does expose a few potential issues relative to missions of this type. For example, the L class maximum receive and transmit data rates, transmit power and FEC seem inconsistent when compared to applications of this type already planned.

6.2.4. Medium Mission Class (M1 & M2) Criteria

Medium Mission Class applications include larger footprint applications such as Orbiting Relays, Launch Vehicles and Sub-Orbital Vehicles.

These applications typically require multiple communication waveforms and frequency bands but have larger power and mass budgets. The advantage of using STRS compliant communication equipment for medium mission class applications will provide a common communication platform permitting reuse of hardware designs among multiple missions, mission reconfigurability, phase reconfigurability, upgradeability, autonomous operation and advanced DSP integration.

The Medium Mission Class radio characteristics include:

- Supports Multiple Vendor Radio
- Some Scalability
- Form Factor Standardization Optional
- Non-standard Hardware Architecture
- Vendor Publishes Abstract Layer Interface
- Hardware Interface Definition at Box, Module, or Component Level
- Relaxed SWaP Constraint
- Multiple Frequency Bands
- Multiple Simultaneous Links
- Data Rates from Low to High

Review of the M1 and M2 platform classes is consistent with these applications with the exception of the 100 Mbps maximum transmit rate and 5 W transmit power. These types of spacecraft typically have downlink rates exceeding 400 Mbps and power levels of 10 W or greater.

6.2.5. Large Mission Class (H1 & H2) Criteria

The Large Mission Class applications are typically large platforms with long duration missions such as Space Stations, Lunar Outpost or Martian Outpost.

These mission applications have multiple communication requirements including earth ground links, crew ships, supply ships, EVA radios and surface exploration communications. The ability of SDR to utilize multiple waveforms and frequency bands would require fewer radio sub-systems and would have the ability to utilize common replacement assemblies. Additionally, Large Mission Class equipment would be used in Ground Stations permitting equipment reuse over multiple missions with a low risk of obsolescence.

Like the Small and Medium class radios, the Large Mission Class communication sub-system SDR advantages include a common communication platform, mission reconfigurability, phase reconfigurability, upgradeability, autonomous operation and advanced DSP integration.

Large Mission Class radio characteristics include:

- Support for Multiple Vendor Radio
- Highly Scalable
- Hardware Interface Standardization
- Hardware Interface Definition at Box, Module, or Component Level
- Standard Form Factor Cards

- Backplane Bus
- Interchangeable Modules
- Minimal SWaP Constraint
- Multiple Frequency Bands
- Multiple Simultaneous Links
- Data Rates from Low to Very High

The H1 and H2 platform classes are appropriate for this mission type with the exception of receiver bandwidth. Given that the STRS Architecture Description Document has transmit data rates up to 1000 Mbps, there is a requirement to have STRS compliant communication equipment in ground stations that can receive these signals. Additionally, it seems likely that future space station and outpost missions may require higher bandwidth receive capabilities.

6.2.6. Recommendations

Consider renaming “Mission Class” to “Radio Class”. The term “Mission Class” speaks to the type of spacecraft or application. However, a spacecraft may have L, M and H class communication equipment according to mission requirements. The term “Radio Class” seems to better describe the level of capability of a particular communication sub-system.

With advances in FPGA technology, it may be unnecessary to restrict the type of FEC coding capability in the class L platform class.

Existing L and M platform class radios have transmitter power capabilities > 10 watts. The STRS transmit power specification seems low for the L and M classes. Consider having a high and low power subset for these classes. For example, M2-H may indicate medium class high power >10 W.

Existing L platform class radios have transmitter and receiver bandwidths of 8 MHz and the capability to transmit and receive 4 Mbps. The STRS specification for the L platform class radio is restrictive and should be expanded.

The STRS specification has H class radios with transmit bandwidth of 600 MHz and data rates up to 1000 Mbps but the maximum receive bandwidth for this class is 50 MHz. Transmit and receive bandwidths and data rates should match in the H class to accommodate reception at the highest rates. Consider expanding the receive bandwidth for H class radio.

Generally, H class should have hardware interface definitions based at the Card or Module level while Small/Medium Radio Class definition should be at the Radio Level.

6.3. Hardware Interfaces

6.3.1. Interfaces between Modules

Background

The purpose of this section is to provide a mechanism for specifying inter-module interfaces within the context of the STRS architecture that will facilitate the integration of STRS radio equipment modules from

multiple vendors to realize the STRS benefits of scalability and extensibility. This is an expansion of Section 8 of the STRS Architecture 1.0.

One of the key goals of such a mechanism is that it be **generic**; too great a degree of specificity is likely to limit its applicability and range of utility. Accordingly, another goal is that the mechanism be **inclusive**; a mechanism that is usable for a wide range of applications is deemed to be more valuable. It is also desired that the mechanism for specifying these interfaces be **exhaustive**; it must accommodate the full definition of how interfaces can be accessed, either by an integrator, or by independent developers creating improved or otherwise competitive radio system modules.

The mechanism for specifying the inter-module interfaces should also be **"implementation independent;"** the scope of applicability of the interface definition should not be limited by a particular implementation. It is also important that the mechanism for specifying these interfaces have the capacity to **leverage existing industry standards**, or to **adapt to custom designs** that can be opened up via interfaces that permit integration with third-party equipment. The interface specifications contemplated in this discussion are envisioned as a framework to define all interfaces and as a starting point for adaptation between interfaces. It is further envisioned that these interface descriptions/specifications will be used with a set of standard APIs to transfer information between modules regardless of the actual physical interface used.

STRS Interface Characterization Table

Table 6-1 identifies major characteristics that must be considered in identifying the interfaces between modules for the STRS.

Table 6-1. STRS Module Interface Characterization Table

Parameter	**Description / Comments**
Name	
Interface type	Point to point, point-multipoint, multipoint, serial, bus, other
Implementation level	Component, module, board, chassis, rack, remote node
Reference documents / Standards	
Note / Constraints	
Transfer speed	Clock speed, throughput speed
Signal definition	
Physical Implementation	
Technology	
Connectors	Model number, number of pins, physical dimensions
Data plane	Width, speed, timing,
Control plane	Control signals, control messages or commanding, interrupts
Functional Implementation	
Models	Data plane model, control plane model, test bench model

Table 6-1. STRS Module Interface Characterization Table	
Parameter	**Description / Comments**
APIs	
Logical implementation	
Addressing	
Channels	Open, close,
Connection type	Forward, terminate, test
Implementation Library	
Hardware / software	Model, physical, software drivers, software APIs for a given OS environment
Implementation summary	Size, weight, power, technology, radiation assurance level, reliability

Recommendations

It is recommended that these interface characterizations be incorporated into the STRS Architecture. They should be continuously reviewed and revised to remain current with advances in applicable technologies and is relevant to advances in software defined radios and space communications.

6.3.2. External Interfaces

Background

The purpose of this section is to provide a mechanism for specifying physical external interfaces within the context of the STRS architecture that will allow third party hardware developers and platform integrators to integrate their products and platforms with a specific STRS configuration. It is an alternative view and expansion of the HID discussed in section 8 of the STRS Open Architecture Description release 1.0.

The goal of this section is to provide external interface descriptions that are:

- ✓ Generic
- ✓ Inclusive
  - Mechanism must be such that it can be used for a wide range of physical implementations
- ✓ Exhaustive
  - Full definition on interface access by an independent developer

- ✓ Implementation Independent
- ✓ Compatible with current and future platforms
- ✓ Has mechanism for leveraging industry standards or for adapting to custom designs
- ✓ Can be used as an framework to define all external interfaces and as a starting point for adaptation between interfaces

Functional Diagram

The functional diagram showed in Figure 6-3 provides a general classification of the STRS external interfaces. More detail on the specific interfaces is provided in paragraph 4.4 which provides the functional hardware external interface taxonomy.

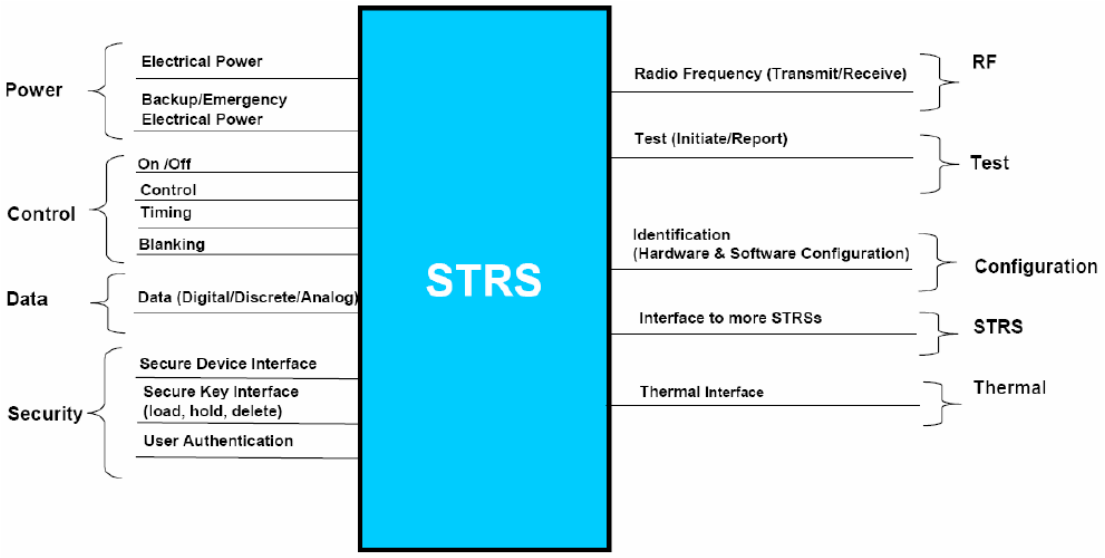

Figure 6-3 STRS External Interfaces

Interface Characterization Table

This interface characterization table identifies major characteristics of the interface that must be considered in identifying the external interfaces for the STRS.

Table 6-1. STRS External Interface Characterization Table

Parameter	**Description / Comments**
Name	
Interface type	Point to point, point-multipoint, multipoint, serial, bus, radio frequency, power, test, thermal, other
Implementation level	
Reference documents / Standards	E.g.. MIL-STD-1553, Ethernet, IEEE1394B, Space Wire (IEEE 1355), Time Triggered Protocol (TTP) etc.
Note / Constraints	

Table 6-1. STRS External Interface Characterization Table	
Parameter	**Description / Comments**
Transfer speed	Clock speed, throughput speed
Signal definition	
Physical Implementation	
Technology	
Connectors	Model number, number of pins, physical dimensions, special features (e.g. filter pins)
Power, Power Factor, Waveform, Current, Voltage, Frequency	
Impedance, Voltage Standing Wave Ratio (VSWR)	
Functional Implementation	
Models	Data plane model, control plane model, test bench model
Logical implantation	
Addressing	
Channels	Open, close,
Connection type	Forward, terminate, test
Implementation Library	
Hardware / software	Model, physical, configuration?
Implantation summary	Size, weight, power, heat, technology, radiation level, periodic maintenance, reliability etc.

Functional Hardware External Interface Taxonomy

The following taxonomy classifies the specific external interfaces and describes the external interfaces. This section can be revised to reference NASA STRS Architecture Taxonomy Table when these classifications are incorporated into that table.

- **Power Interface**
  - ✓ The Power interface supports Primary, Emergency and Backup Electrical Power for operating the STRS as well as Power Control, Monitoring and Telemetry (TM).
- **Control Interface**

- ✓ The Control interface controls the STRS on and off selection, controls selection of frequency, waveform, modulation, data format, timing, blanking and the use of time and frequency references.

- **Data Interface**
  - ✓ The Data interface transfers all digital, analog and discrete data being transmitted or received by the STRS. The data can be transferred as analog audio signals, digitally formatted information etc. and sent on buses, discrete signals, local area nets etc.
- **Security Interface**
  - ✓ The Security interface provides compatibility with all approved external devices used to secure data and provides secure key interfaces for loading, holding or deleting keys. In addition, the security interface authenticates users that operate the STRS.
- **Radio Frequency (RF) Interface**
  - ✓ The radio frequency (RF) interface refers to the carrier frequencies for transmitting and receiving STRS data, IF (intermediate frequencies) for connectivity to RF (radio frequency) up converters and down converters as well as carrier frequencies used for other equipment internal to the STRS. This interface may be used to connect the STRS to an antenna, RF filter, RF switch, up converters, down converters, RF power amplifier, low noise amplifier etc.
- **Test Interface**
  - ✓ The test interface provides the ability to initiate and report tests of the STRS. The interface supports stimulus response testing as well as provides a high bandwidth capability greater than that supported on other STRS interfaces.
- **Configuration Interface**
  - ✓ The Configuration Identification Interface provides a means to identify the STRS hardware and software configuration without applying normal electrical power to the STRS. This interface will provide identification of waveforms programmed into the STRS radio. This interface can be used in warehouses, on remote satellites and vehicles, spacecraft, space stations, etc. for trouble shooting, waveform selection and configuration control.
  - ✓ The Configuration Identification Interface may be a passive or active interface. Examples of the Identification Configuration Interface are: a) Passive- identification resistors, grounded connector pins, radio-frequency identification (RFID) etc. b) Active - serial bus providing low power to access STRS low power internal memory device, active RFID etc.
- **STRS Interface**
  - ✓ The STRS interface provides the STRS with the capability to share control data etc. with other STRS units for expansion or fault tolerant redundancy.

- **Thermal Interface**
  - ✓ The thermal interface transfers thermal energy between the STRS and the platform to provide an acceptable thermal environment for storing and operating the STRS. The interface supports the use of liquid, gas or contact as a means of providing convective, conductive or radiated thermal energy transfer.

Recommendations

It is recommended that these External Interface characterizations be incorporated into the STRS Architecture. They should be continuously reviewed and revised to remain current with advances in applicable technologies and is relevant to advances in software defined radios and space communications.

6.4. Reliability

6.4.1. Introduction

The first task of assessing the capacity for reliable operation of an STRS-based SDR is to identify the requirements for reliable operation in the space environment, and to discover the implications of these requirements on the hardware architecture. Accordingly, a set of primary reliability factors are identified here relative to the STRS Hardware Architecture. Since reliability must be "designed in" to the system, a close examination of the STRS Hardware Architecture is necessary to confirm that it is consistent with design for reliable operation. Two key dimensions of a reliable design are fault detection/recovery, and redundancy. The STRS Hardware Architecture is therefore assessed in the context of these measures.

6.4.2. Discussion

Persistent Storage – Reliable SDRs for space must be configured with adequate persistent storage capacity to store as many images of each configurable device as are identified in the mission's fault management system. The location of such persistent storage may be dependent on the hardware modules present in the transceiver, i.e. a digital processing module may support its own memory to hold FPGA images.

Diagnostic Interface – Reliable operation is also dependent on support within the architecture for a diagnostic interface which can be queried by a reliability of fault management application to produce telemetry. A common API for these services is mandated.

Process Protection – Process Protection is another dimension of reliability that an SDR must support in the space environment. Isolation between waveform applications is mandated in this case, to prevent program failures in one waveform from causing a failure in another waveform. The existence of multiple applications running at multiple priorities imposes the need to provide enough isolation such that an exception that occurs does not corrupt the execution of other waveforms or applications. Hardware and software modularity can provide the required degree of isolation.

Reliable Default Waveform – A reliable space SDR must also be equipped with a reliable default waveform that is used in the event that mission waveforms are not operational. The radio must be able to protect itself

and the mission by retaining a default configuration that can be installed if the primary waveform implementation is faulty. This default waveform will be used by the radio's fault management service. The enabling of this waveform can be control internally to the radio or commanded by spacecraft bus controller. This capability can also be handled operationally if the spacecraft flies with more than one radio

Power-on Modes – Power-on modes must be defined in the architecture that can support independent diagnostic recovery sequences. Each radio mission will have different requirements on how it will behave when power is applied. The STRS architecture can provide configurable interfaces to support this capability.

Default Initialization – Reliable initialization dictates that the architecture support configurable initialization sequences for automated recovery and initial power-on. As is the case with the power-on modes, other fault scenarios can require different capabilities when attempting a recovery.

Redundancy – The APIs should permit and support development of redundancy services to sustain communication links in the event of a single fault failure within the digital components. This requirement may not be required for all STRS applications, but will depend on mission requirements.

Waveform Upload Reliability – Reliable uploads to the radio are mandatory. Two key points governing reliability for uploads are 1) the upload process must not affect currently operational waveforms; and 2) the radio will validate and authenticate the upload before allowing reconfigurations. Furthermore, the radio system must have the capacity to recover from improper configurations.

6.4.3. Recommendations

The STRS architecture must be flexible in allowing spacecraft system engineers to choose the redundancy and reconfigurability behaviors consistent with mission requirements. As an example of this, considerable additional resources may be required (e.g., persistent storage) in order to reliably switch from one configuration to another without any disruption. Alternatively, disruption may be traded for hardware resources. The STRS Architecture APIs enable vendors to implement reliability based on mission requirements. This approach mandates that modules are required to support diagnostics and reporting mechanisms to validate their operation.

The STRS Architecture has the inherent capacity for robust fault detection and management. The STRS interfaces can be used to distribute or propagate fault information under the supervision of a Fault Management Service Application operating via the STRS Infrastructure, APIs, and HAL. Such a Fault Management Service Application should encompass fault alarm generation, fault alarm polling, or other means of fault alarm detection.

Similarly, a STRS Service Application can be developed to manage redundancy between multiple radios or radio modules. Equipment vendors should specify the reliability and degree of support for redundancy on a module or radio-system level for each discrete equipment item. Typical cases comprise single-string redundancy, as well as redundancy at either the radio or module level.

6.5. HAL definition

6.5.1. STRS Goals

The emergence of SDRs for space offers NASA the opportunity to improve the way space missions develop and operate space transceivers for communications and navigation. Software defined radios provide the capability to change the functionality of the radio during mission development or after launch. The ability to change the operating characteristics of a radio through software once deployed to space offers the possibility to reduce development cost and risk by adapting generic space platforms to meet specific mission requirements. The STRS Architecture Standard can reduce NASA's dependence on ad hoc SDR implementations, provide reliable, flexible and extensible systems and make the economies that arise in an open-standard environment accessible to NASA.

The benefits that the STRS standard should provide include 1) a scalable architecture supporting small- to large-scale space radio systems, 2) an open architecture, with published specifications that do not provide artificial advantages to using a single-source procurement, 3) extensibility, promoting innovation and technology insertion to extend the lifecycle of NASA radios and reduce average lifecycle costs, 4) platforms that promote waveform portability, requiring minimal effort to port waveforms between different implementations of STRS Radio Platforms, and 5) interoperability, ensuring that radios support existing waveforms while being adaptable to future waveform specifications, and allowing space radios to provide services to and accept services from other systems, and to use the services to enable them to coexist effectively together.

6.5.2. Waveform Portability

An application is portable across a class of environments to the degree that the effort required to transport and adapt it to a new environment in the class is less than the effort of redevelopment. Waveform portability relies on development of a consistent API set that is waveform-independent, and a platform configuration that identifies component location and type to ensure that the waveform applications view is consistent. Portability is more likely when the waveform can make use of simple interfaces for distributing data and controlling hardware, which is the function of a Hardware Abstraction Layer. All Operating Environments are affected by functionality implemented in hardware. Porting functions with HW components is more difficult; for space radios, high data rates and environmental factors often dictate a hardware or hybrid implementation.

STRS provides HAL interfaces to limit the impact of functionality implemented in hardware on the application portability. Since implementations of the STRS Architecture on a physical platform will be vendor-specific, vendors must publish inter-module communications spec to permit 3rd Party Hardware and SW development and thereby ensure waveform (application) portability across platforms.

6.5.3. HAL Definition

The STRS Open Architecture Description notes that "the function of the HAL, which is a higher level abstraction, is to decouple the infrastructure from the specialized hardware." This is consistent with common definitions for the hardware abstraction layer of a programmable computing machine:

> A hardware abstraction layer (HAL) is an abstraction layer, implemented in software, between the physical hardware of a computer and the software that runs on that computer. Its function is to hide differences in hardware from most of the operating system kernel, so that most of the kernel-mode code does not need to be changed to run on systems with different hardware. A HAL allows instructions from higher level computer languages to communicate with lower level components, such as directly with hardware.
>
> Hardware abstraction layers are of an even lower level in computer languages than application programming interfaces (API) because they interact directly with hardware instead of a system kernel, therefore HALs require less processing time than APIs. Higher level languages often use HALs and APIs to communicate with lower level components.
>
> Operating systems having a defined HAL are easily portable across different hardware. This is especially important for embedded systems that run on dozens of different microcontrollers.[14]

As a key component of the STRS Architecture, the HAL specification defines the physical and logical interfaces for inter-module and intra-module integration.

6.5.4. The HAL in STRS

In STRS, the HAL offers a platform-independent view of the specialized hardware implementations (e.g. FPGA) by abstracting the physical hardware interfaces. It implements the software that is directly dependent on the underlying hardware. STRS should require that developers publish hardware interfaces (i.e. HAL API) such as FPGA data and control interfaces. The HAL API must include a description of each method/function used, including its calling sequence; return values, and an explanation of its functionality. This permits NASA to access the developer's proprietary, intellectual property associated with the waveform algorithms by exposing the interfaces used in the FPGA or other hardware for subsequent developments or corrections without relying on the continuing involvement of the original developer, as traditional radios require, and without compromising the integrity of the developer's intellectual property rights.

6.5.5. The STRS Architecture HAL Context

The STRS Architecture defines APIs with Radio Set view. Radio Set APIs exist independent of waveforms and thus provide access to a common set of services and devices that may be used by any waveform. The main function of the STRS Architecture is to serve as a conduit for data transfer. Figure 6-4 provides a graphical depiction of the STRS Architecture, showing the HAL and its relationship to the other STRS layers and interfaces. Table 6-2 provides a brief description of these layers and interfaces relevant to the HAL. *A table that provides detailed descriptions of these layers and interfaces relevant to the HAL and HID should be added to the STRS Architecture document.*

[14] http://en.wikipedia.org/wiki/Hardware_Abstraction_Layer

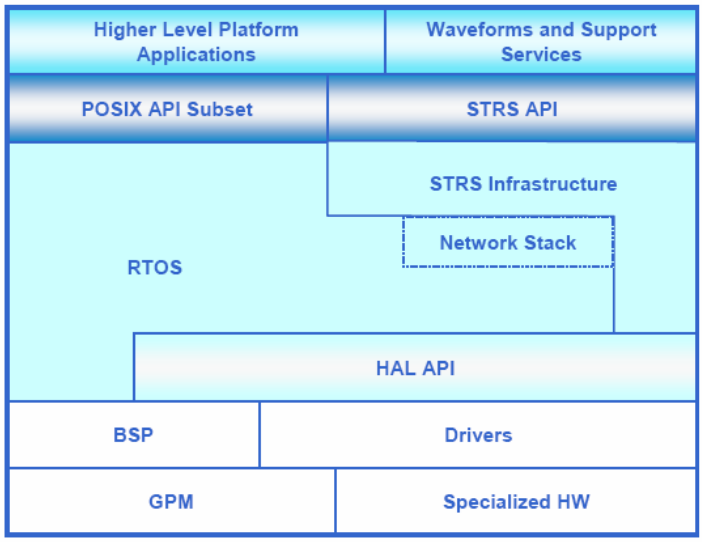

Figure 6-4. STRS Architecture HAL Context[15]

Table 6-2. STRS Software Architecture Descriptions

Layer	Description
STRS API	The STRS API provides a consistent interface for executing the applications and services. The associated Device Control interfaces provide a hardware abstraction layer (HAL) for the waveform applications to interact with the hardware.
BSP	The Board Support Package (BSP) provides the hardware abstraction of the GPM module. A BSP contains source files, binary files, or both and an OEM Adaptation Layer (OAL), which includes a boot loader for initializing the hardware and loading the operating system image. The OAL is the software that is hardware specific and is compiled and linked into the embedded operating system. STRS Applications can interact with the GPM hardware via the STRS Infrastructure and the HAL.
HW Drivers	The hardware drivers provide the platform independence to the software and infrastructure by abstracting the physical hardware interfaces into a consistent Device Control API.
Logical HAL Interfaces	Provides the Device Control interfaces that are responsible for all access to the hardware devices in the STRS radio.

[15] An augment view of this relationship among the components is something deferred to future working group activity.

6.5.6. Modular STRS HW/SW Configuration

Figure 6-5 depicts the modular STRS platform configuration, showing the role of the HAL. This figure highlights the necessity for an effective HAL supported by a detailed HID (Hardware Interface Definition). The HID documents the physical interfaces of the individual modules through abstraction and definition of the module data flow functionality. Among the primary modules defined by the STRS Architecture are the general purpose processor module, a specialized signal processing module, and a radio frequency module. The hardware architecture does not specify an internal physical implementation on each module. The STRS vision anticipates that the radio developer will combine modules as necessary during the radio design process to meet the specific mission requirements. Module developers can incorporate proprietary circuitry or software, as long as the modules meet the architecture rules and interface specifications defined for each module. NASA can obtain the desired benefits of the STRS Architecture, including scalability, extensibility, and waveform portability, with access to the interface information required to produce substitute or additional modules that will interoperate with the existing or core modules of the radio. The HID defines the physical interfaces that allow third party hardware developers to integrate their products with a specific STRS platform. The HID for each module abstracts and defines the module functionality for data flow, enabling multiple vendors to provide different modules or add modules to existing radios. The HID specifies the electrical interfaces, connector requirements, and physical requirements necessary to create the HAL abstraction.

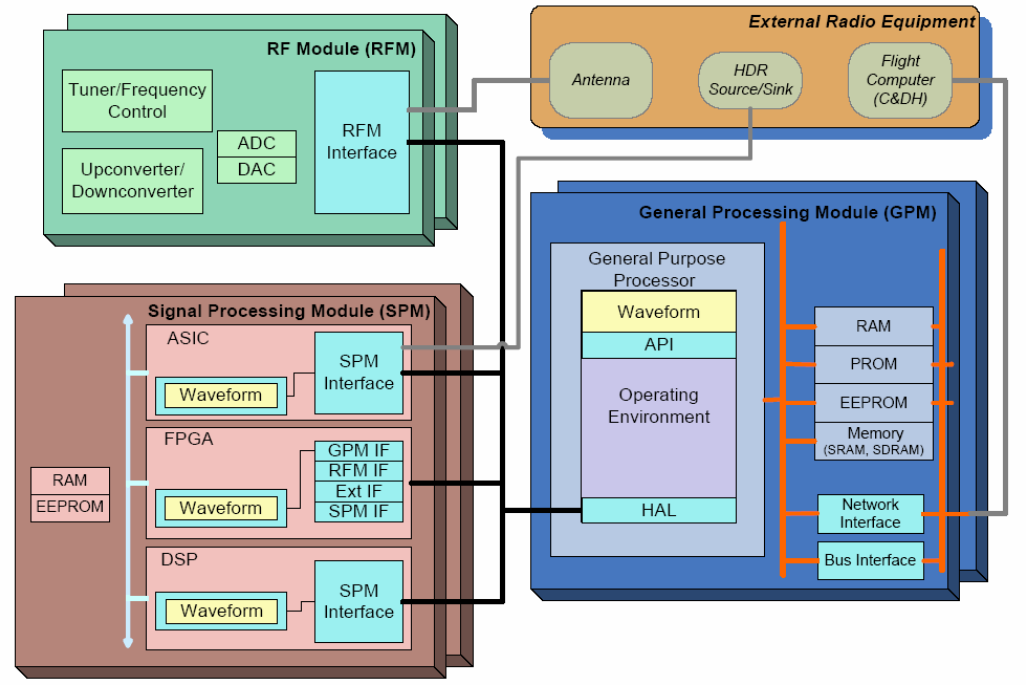

Figure 6-5. STRS HW/SW Configuration Showing HAL

6.5.7. STRS HAL

The Hardware Abstraction Layer should provide the STRS Architecture with a degree of "future-proofing," and should promote innovation in technology while protecting infrastructure from obsolescence. To achieve

these objectives, hardware component vendors must conform to strict interfaces defined by platform supplier (HID) and the HAL should use arguments compatible with STRS Functional APIs for its transport function. An example of this is the nomenclature:

HALSend(deviceID,funcID,&funcData,noBytes);

Figure 6-6 depicts the high level relationships between modules in a STRS radio. The application in the GPM will use STRS Device Control APIs that interface to the device drivers associated with the SPM and RFM modules. The device drivers communicate via the physical interface specification defined by the HID in transferring command and data information between the modules. On the SPM, front-end interfaces provide connection between the DSP and FPGA components with the External HID. An internal HID can be used to provide application developers with the capability to exchange data between components on the SPM. For modules such as the RFM, these interfaces can be a memory mapped registers, serial, parallel, and GPIO.

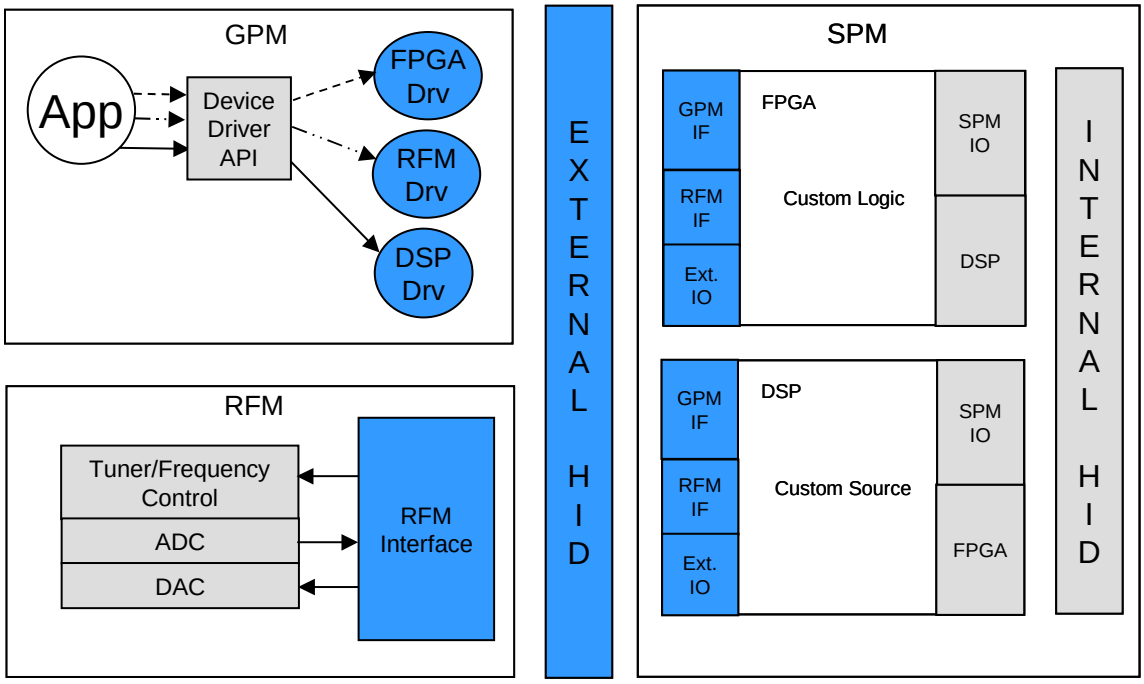

Figure 6-6. Detailed STRS HAL Diagram

6.5.8. The HAL and the HID in STRS

In STRS, *the HAL should offer a platform-independent view of the specialized hardware implementations (e.g. FPGA) by abstracting the physical hardware interfaces.* It implements the software that is directly dependent on the underlying hardware. *STRS should require that developers publish hardware interfaces (i.e. HAL API) such as FPGA data and control interfaces. The HAL API documentation must include a description of each method/function used, including its calling sequence; return values, and an explanation of its functionality.* This permits NASA to access the developer's proprietary, intellectual property associated with the waveform algorithms by exposing the interfaces used in the FPGA or other hardware for subsequent developments or corrections without relying on the continuing involvement of the original developer, as

traditional radios require, and without compromising the integrity of the developer's intellectual property rights.

The STRS Open Architecture Description defines the HID, and notes the requirements imposed on the radio supplier to publish the HID as a condition of STRS-compliance:

> The radio supplier shall publish a Hardware Interface Description (HID), which defines the physical interfaces that allow third party hardware developers to integrate their products with a specific STRS platform. The HID specifies the electrical interfaces, connector requirements, and physical requirements for the delivered radio… Each module's HID abstracts and defines the module functionality for data flow enabling multiple vendors to provide different modules or add modules to existing radios.

Similarly, the STRS Open Architecture Description defines the HAL, and notes the requirements imposed on the radio supplier to thoroughly document and publish the HAL API as a condition of STRS-compliance:

> The HAL API defines the physical and logical interfaces for inter-module and intra-module integration. The HAL API documentation must include a description of each method/function used, including its calling sequence, return values, an explanation of its functionality, any preconditions before using the method/function, and the status after using the method/function. Examples should be included where helpful.
>
> The HAL API documentation shall also contain information about the underlying hardware such as address and data interfaces, interrupt input and output, power connections, plus other control and data lines necessary to operate in the STRS platform environment. The electrical interfaces, connector requirements, and physical requirements are specified by the platform provider. Information on a module's use of data in the specification will be made available to waveform developers either directly from the manufacturer (specific types of components) or from the platform provider (memory maps based on positions within chassis/enclosure). The STRS Infrastructure will use this information to initialize the hardware drivers such that the control and data messages will be appropriately delivered to the module.

The STRS HAL must adequately and completely specify the communication and integration specification between physical hardware modules. *The STRS Infrastructure should abstract this implementation in providing the services specified with the STRS API.* To effect this specification, the HAL for the STRS Architecture is decomposed into two components, the Logical (software) Interfaces, and the Physical (Hardware) Interfaces. This separation is depicted in Figure 6-7.

Compliance with STRS requires developers to provide a description of the physical hardware interfaces (the HID) used in the implementation and a mapping of the control interfaces to each of the modules. *These HIDs should be provided by the respective module developers to permit NASA to augment or replace modules from in-house and outside sources, possibly from other than the original vendor.* The ability to compete existing radio modifications or additions as opposed to sole-source contracts to legacy providers offers an opportunity to achieve lower costs and improve capabilities.

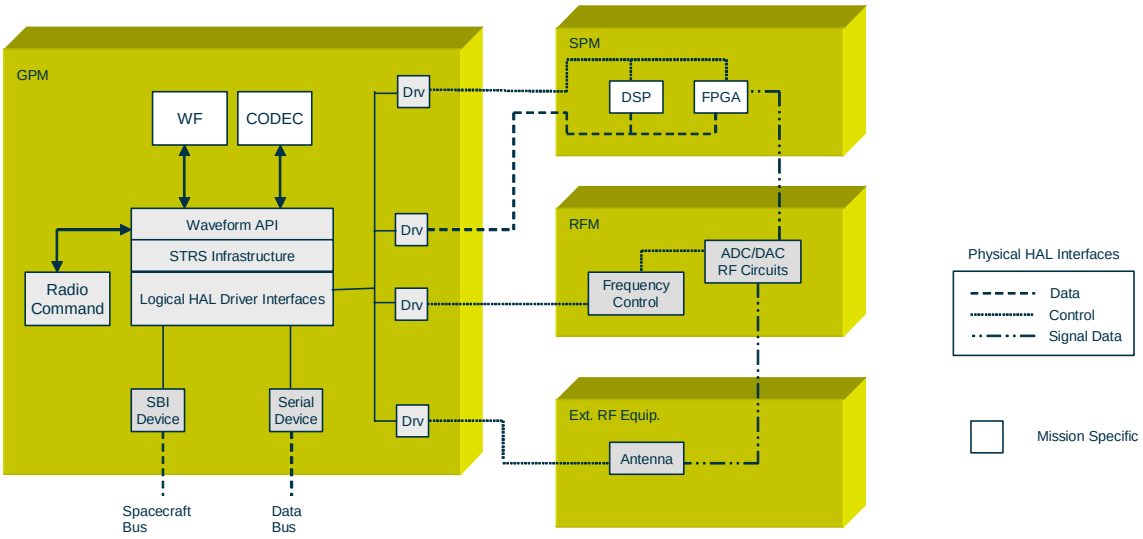

Figure 6-7. STRS Architecture HAL Diagram

Examples of the HID includes interface type, transfer speeds, signal definition, addressing, data width, timing, control signals, messages, interrupts, hardware/software boundary (model, drivers, custom interfaces, operating environment) and implementation summary (size, weight, power consumption, radiation level, and reliability).

6.5.9. HAL Benefits

The use of a HAL promotes extensibility by adapting existing software and hardware for technology insertion (infrastructure portability). By publishing the Hardware Interface Definition (HID) after the platform has been constructed facilitates hardware technical insertion from multiple vendors.

The HID provides 3rd-party developers the structure under which they can develop new modules for a platform. In such a case, the HID should specify bus configurations as well as GPIO pin assignments for the backplane (if there is one). The GPIO pin assignments are typically allocated based on the associated functionality. For example, a set of pin assignments may be dedicated to Channel-1 Receiver data stream.

The distinction of functional data provides two benefits, 1) it provides a set of distinct pins to a 3rd-party developer which to provide the module functionality that insures a mechanism for integration with the other modules in the system, and 2) it provides platform developers and system designers the capability to respond to off-nominal conditions that can be mitigated with the STRS infrastructure. For example if a channel in the SPM module fails, waveform implementations that permit the channel to operate at a lower data rate using the GPM for all signal processing could be developed and uploaded to the radio. *Each vendor should provide the Hardware Abstraction Layer (HAL) device driver that provides the logical (software) to physical (hardware) interface layer to integrate the module into the STRS architecture.*

6.5.10. Recommendations

It is recommended that NASA modify the STRS Open Architecture Description pertaining to the HAL and HID definitions as follows:

- Reflect these comments into Section 7.2 of OA
- Emphasize that vendors are responsible for publishing the HID
- Add HAL definition to STRS Taxonomy document
- Each vendor will provide the Hardware Abstraction Layer (HAL) device driver that provides the logical (software) to physical (hardware) interface layer to integrate the module into the STRS architecture.

6.6. Device Capabilities Survey

6.6.1. Introduction

The following sections present an assessment of the constraints on SDRs currently imposed by technologies that are suitable for space. Since the constraints may have different implications depending on application and architecture, specific implications of device performance and availability on implementation capabilities are not explicitly stated. Developers of space hardware and software systems will make the inferences most suitable to the applications and environments for which they are concerned.

However, it is highlighted that these constraints must be observed in formulating any hardware and software architectures for space SDRs.

Key performance parameters evaluated for technology availability include gate density, supply voltage level, gate delay, and power consumption, which are closely related to process node feature size. Typically, new process technologies are augmented by additional layers of metalization to interconnect the transistors, permitting more densely packed gates, logic elements, and memory cells. Similarly, core voltage (transistor supply voltage) tracks feature size linearly. At 350 nm, 3.3 volts represents the typical core voltage, while at 250 nm, the core voltage is 2.5 volts. Gate delay is a primary parameter for assessing potential operating clock speed. As feature size has decreased, capacitance between transistors (or gates, etc.) has come to dominate the overall performance capability of the technology, although it is somewhat offset by the additional layers of interconnect metalization. For this reason, migration to lower-resistivity interconnect materials has been a key technology enabler. Most commercial semiconductor producers have already migrated from tungsten or aluminum to copper, although this migration is still underway in rad-hard foundries.

6.6.2. ASICs and FPGAs

The left-hand panel of Figure 6-8 depicts the evolutionary trend of ASIC technology, expressed by several key performance parameters. This forecast relies on several important assumptions

1. A new process generation emerges every 24-36 months[16]
2. Radiation tolerant processes have historically lagged commercial process introductions by 2-3 process generations

[16] Developing Science and Technologies List, Section 19: Space Systems Technology, DoD Defense Threat Reduction Agency, October 2002.

3. Leading commercial semiconductor companies entered production with the 90 nm process node in 2003

These assumptions indicate that the leading-edge process in the radiation-tolerant ASIC market in the 2001-2004 time period is at the 250 nm node, although many radiation-tolerant ASIC foundries were still producing designs on older processes (350 nm and 500 nm) during that period.

The improvement in TID tolerance relies on two emerging factors: 1) TID tolerance naturally increases as feature size decreases, accounting for the improvement in to 300 krad (Si) for rad-tolerant devices, and 2) rad-hard foundries are committing to "strategic" levels of TID tolerance, accounting for the persistence of 1 Mrad (Si) TID capability. Decreasing intrinsic gate delay, as well as the reduction in power (W/g-MHz) are results of reductions in capacitance and lower voltage associated with finer geometry processes.

ASICs	2001 - 2004		2005 - 2007	
Gate Density	1M gates		6M gates	
I/O Levels	3.3 V	2.5 V	2.5 V	1.8 V
Gate Delay	125 ps	165 ps	80 ps	
Power	140 nW / g-MHz		45 nW / g-MHz	
TID Tolerance	300 k / 1 Mrad		100 k / 1 Mrad	
SEU	< 1E-11		< 1E-10	
LET	> 125 MeV-cm^2/mg		> 125 MeV-cm^2/mg	
Process Node	250 nm		180 nm	

FPGAs	2001 - 2004	2005 - 2007	
Gate Density	1M / 300k gates	5M / 1M gate	
Internal Speed	200 MHz	300 MHz	
Core Voltage	2.5 V	1.8 V	1.5 V
LE Delay	12 ns	8 ns	
Power	3 µW / LE / MHz	1.5 µW / LE / MHz	
TID Tolerance	100 k / 1 Mrad	200 k / 1 Mrad	
SEU	< 1E-10	< 1E-10 / 1E-12	
LET	> 125 MeV-cm^2/mg	> 150 MeV-cm^2/mg	
Technology	SRAM / AF	SRAM / AF	
Process Node	250 nm	180 / 150 nm	

Figure 6-8. Device Capabilities – ASICs and FPGAs

The right-hand panel of Figure 6-8 shows the trend for space-qualifiable FPGA technology over the same time period. Again, key performance parameters and their projected values are catalogued in this figure. The forecast shown in this table relies on three additional assumptions:

1. FPGA production generally lags commercial process introduction by 1-2 process generations.
2. Currently, radiation-hardening and/or space qualification of the FPGA product family imposes an additional lag of 1-2 process generations (2-4 years)[17].

As validation, Xilinx is the current commercial process leader, introducing its newest FPGA product family the Virtex-4, on the 90 nm process node in 4Q2004, which represents a 1-generation lag (Intel began production at 90 nm in 1Q2003). These factors lead to the conclusion that space-qualified and/or radiation-tolerant FPGAs lag commercial process introduction by 6-8 years. This conclusion indicates that these FPGAs will not be produced on the 90 nm node until almost 8 years after the introduction of 90 nm silicon.

Three FPGA vendors (Xilinx, Aeroflex, and Actel-BAE) currently lead the market, which is reflected in the two columns, covering the time periods 2001-2004 and 2005-2007. These vendors produced FPGAs for space applications at the 250 nm node bin the period 2001-2004, with product releases in 2005 at the 180/150 nm process node(s).

[17] Xilinx, "New Technologies & Trends in Programmable Logic Devices," October 2004.

Decreasing intrinsic gate delay is reflected in decreasing values for LE (logic element) delay, while the reduction in power (W/g-MHz) results from reductions in capacitance and lower voltage associated with finer geometry processes. It is expected that a low-resistivity interconnect technology such as copper will be introduced by all FPGA vendors to enhance speed and reduce power dissipation.

6.6.3. SRAMs and Processors

The left panel of Figure 6-9 depicts the technology availability for space-qualifiable SRAM technology. This data is based on radiation-hardened CMOS or SOI processes using a 6-transistor memory cell that occupies 0.6 μm^2 at 65 nm. The chip size is assumed to be less than 100 mm^2. These assumptions permit estimates to be constructed for storage density per chip. Decreases in intrinsic gate delay and faster interconnect technologies (e.g., copper) are reflected in decreasing values for access speed. The reduction in active power represents a balance between increased number of power-consuming circuits per chip and reductions in capacitance and lower voltage associated with finer geometry processes. Reduction in standby power reflects advanced circuit techniques for leakage current reduction. Often, rad-hard foundries will use SRAM as a pilot vehicle for bringing a new process on-line. Accordingly, SRAM process introductions are slightly advanced in comparison to rad-hard ASIC foundry process introductions.

SRAMs	2001 - 2004		2005 - 2007	
Density	4 Mb		4M / 16Mb	
Access Speed	25 ns		15 ns	
Voltage	3.3 V	2.5 V	2.5 V	1.8 V
Active Power	500 mW		200 mW	
Stdby Power	50 mW		20 mW	
TID Tolerance	100 k / 1 Mrad		> 1 Mrad	
SEU	< 1E-10		< 1E-10	
Technology	SOI / CMOS		CMOS / SOI	
Process Node	350 / 250 nm		180 / 150 nm	

Processors	2001 - 2004		
Performance MIPs	240	167	60
Clock MHz	133	100	65
Power (W)	5	3	1.5
Bus Width	32b	32b	32b
L1 Cache (kB)	32	16	8
TID (krad)	200	100	300
SEU (upsets / b-d)	E-10	E-5	E-11
Architecture	750	603	CF
Proc Node (nm)	250	350	350

∗750⇨PPC 750
603⇨PPC603
CF ⇨ ColdFire

Figure 6-9. Device Capabilities – SRAMs and Processors

The right-hand panel of Figure 6-9 depicts the availability of three different classes (S, M, L) of radiation-hardened GPP (general-purpose processor). One processor defines each class. These processors are: 1) the PowerPC 750 (PPC 750), produced by BAE, 2) the PowerPC 603 (PPC 603) produced by Honeywell, and 3) the General Dynamics Coldfire (CF), based on the Motorola Coldfire V2 core. The rows show the relevant performance parameters of the processors, with one column devoted to each processor class. The Coldfire processor is representative of a class in which architecture and process techniques are leveraged to achieve very low power consumption. At the other end of the scale, the PPC750 class is dedicated to achieve maximum operating performance (MIPS), with power efficiency a secondary objective. Although the design space for processors permits a far greater range of parameter values than shown in the figure, the processors shown are representative of achievable performance in the power-constrained environment space environment, where a suitable radiation assurance level is the predominant selection criteria.

6.6.4. Non-Volatile Memories

Figure 6-10 depicts two main categories of nv- (non-volatile) memories: 1) EEPROM, represented for the 2001-2004 time period in the left-hand panel of the figure, and 2) PROM, represented in the 2001-2004 time-period in the right-hand panel of the figure. These two technologies are shown separately based on differences in their relative access speeds, radiation tolerance, densities, and power consumption.

eePROMs	2001 - 2004	
Density	1 Mb	
Access Speed	120 ns	200 ns
Voltage	5 V	3.3 V
Active Power	20mW	20 mW
Stdby Power	100 µW	72 µW
TID Tolerance	100 krad (Si)	
LET / SEU	90 / 40 MeV-cm^2/mg	
Technology	EEPROM	
Process Node	500 nm	350 nm

PROMs	2001 - 2004	
Density	256 kb	
Access Speed	40 ns	65 ns
Voltage	5 V	3.3 V
Active Power	400mW	150mW
Stdby Power	15 mW	5 mW
TID Tolerance	> 1 Mrad (Si)	
LET / SEU	70 / 50 MeV-cm^2/mg	
Technology	PROM	
Process Node	500 nm	350 nm

Figure 6-10. Device Capabilities – eePROMs and PROMs

Figure 6-11 shows the introduction of two new nvRAM technologies in the 2005-2007 time-period, magneto-resistive RAM (MRAM), and chalcogenide RAM (CRAM). These technologies both have very good tolerance to radiation TID, high bit densities, and low power consumption, but diverge when compared on the basis of access speed. Based on the asymmetry in access speed (Read versus Write), CRAM technology is assigned to the left-hand panel of the figure (compare to EEPROM), while MRAM is assigned to the right-hand panel of the table (compare to PROM). However, both use a single transistor per memory cell, and therefore track closely on bit density and power consumption. Note also, that MRAM is introduced on a more advanced process node than CRAM (based on manufacturers published data). Both technologies use a single-transistor circuit topology.

CRAM	2005 - 2007
Density	4 Mb
Access Speed	R / W = 50 / 500 ns
Voltage	3.3 V
Active Power	50 nW / bit (200 mW)
Stdby Power	12 pW / bit (50 µW)
TID Tolerance	> 1 Mrad (Si)
LET / SEU	< 1E-11
Technology	CRAM
Process Node	250 nm

MRAM	2005 - 2007
Density	4 Mb
Access Speed	25 ns
Voltage	3.3 / 1.8 V
Active Power	20 nW / bit (80 mW)
Stdby Power	5 pW / bit (20 µW)
TID Tolerance	> 1 Mrad (Si)
LET / SEU	< 1E-11
Technology	MRAM
Process Node	180 nm

Figure 6-11. Device Capabilities – CRAMs and MRAMs

6.6.5. Recommendations

In order to enable advancement of SDRs in space it is recommended that NASA consider engaging in the following activities in the area of electronic device capabilities:

1. Develop and publish a NASA roadmap for space processor needs and developments
2. Accelerate technology readiness level with an on-going radiation test/evaluation program
3. Develop and fly a technology flight test vehicle to secure flight history for these technologies
4. Identify conditions under which commercial or military parts may be used for space environment
5. Identify space-proven parts categorized by radiation performance

7. Epilogue – Comments on Standardization Philosophy

Assessment of the impact of the review comments and recommendations highlights two distinct philosophies in standards development that participants in the marketplace bring to the standards-development process. Both philosophies represent valid interpretations of the concerns of the marketplace, and both provide valuable perspectives for standards development.

One philosophy promotes the virtues of rule-based processes and procedures for development, and interprets a successful standard as one which ensures that there are few or no variations between implementations. This category of standards finds its usefulness in applications where uniform product behavior is important. A common goal of such a standard is to remove proprietary design advantage from the marketplace, replacing it with proprietary advantages in manufacturing or production efficiency as the basis of competition. Consistent with this philosophy, a useful standard identifies and enforces preferred practices for development and operating capability through mandates and specifications embodied in the standard.

An alternative philosophical interpretation regards a successful standard as a set of practices for coordinating interaction between participants in the marketplace to enable specialization and to promote innovation, usually represented by the emergence of new technologies and applications. This interpretation does not take a view on whether proprietary design advantage is good or bad, but regards it as a phenomenon that is effective in promoting innovation and specialization. In this philosophical interpretation, a successful standard is one which identifies and describes practices for coordinating action between a diverse set of marketplace participants with the ultimate purpose of supporting specialization, and robust and efficient competition.

It is important to emphasize that neither interpretation should be viewed as “good” or “bad.” Each has evolved to address different marketplace concerns. In a marketplace where stability and vertical integration are desired virtues, the first interpretation is effective. In a marketplace in which the participants desire to promote change, specialization, competition derived from design innovation, and horizontal market organization, the second interpretation is effective. The SDR Forum STRS review participants seek to capture the best consequences from both philosophies.

In documenting its review of the STRS Standard, the SDR Forum participants have endeavored to record all perspectives represented in its membership, in order to provide NASA with the insights that arise from each of these complementary philosophical interpretations. It is recognized that the recommendations and interpretations contained in this document, some of which may be contradictory, are made from different points of view, motivated by different concerns in the marketplace.

8. Acknowledgments

The following SDR Forum members contributed directly to this document.

STRS Architecture Review Task Group

Appendix A – STRS Acronyms

Working Group Updates

µm	micrometer
µw	microwatt
ADC/DAC	Analog to Digital Converter/Digital to Analog Converter
AF	
b	bit
CE	
CF	ColdFire™
CLV	Crew Launch Vehicle
cm	centimeter
CMOS	Complementary Metal Oxide Semiconductor
CRAM	Chalcogenide Random Access Memory
DSP	Digital Signal Processing
EEPROM	Electrically Erasable Programmable Read Only Memory
EVA	Extra Vehicular Radios
EXT	External
g	gram
GPM	General Processing Module
HAL	Hardware Abstraction Layer
HID	Hardware Interface Description
IEEE1394B	Institute of Electrical and Electronic Engineers 1394B bus
IEEE 1355	Institute of Electrical and Electronic Engineers 1394B bus
IF	interface, intermediate frequency
ISS	International Space Station
JPL	Jet Propulsion Lab
K	thousand
LE	logic element
LET	Linear Energy Transfer
M	million
Mb	megabit
MER	
MeV	Million electron volts
Mg	milligram
MHz	megahertz
MIPS	million instructions per second
Mrad	million radian
MRAM	magneto-resistive Random Access Memory
MIL-STD-1553	Military Standard 1553 bus
nm	nanometer, nautical mile
ns	nanoseconds
nv	non volatile

PC	personal computer
PPC	Power PC
PROM	Programmable Read Only Memory
RAM	random access memory
RFM	Radio Frequency Module
R-GPM	
RTOS	real time operating system
SBI	
SDR	Software Defined Radio
SEC	Security Module
SEU	Single Event Upset
Si	Silicon
SOI	Silicon-On-Insulator
SPM	Signal Processing Module
SRAM	Static Random Access Memory
Stdby	standby
SWaP	Size, Weight and Power
TID	total ionizing dose
TDRSS	Tracking and Data Relay Satellite System
TTP	Time Triggered Protocol
V	volt
W	watt
WF	Waveform

Original STRS Acronym List

802.x	IEEE network interface standards
AEP	Application Environment Profile
API	Application Program Interface
ASIC	Application Specific Integrated Circuit
BIT	Built-In Test
BSD	Berkeley Software Distribution
BSP	Burst Schedule Packets
BTS	Base Transceiver Station
C++	a computer programming language
C4I	Command, Control, Communications, Computers and Intelligence
CF	Core Framework
CFG	Configuration
COMSEC	Communication Security
CORBA	Common Object Request Broker Architecture
COTS	Commercial off the Shelf
CPU	Central Processing Unit
DCD	Device Configuration Descriptor
DMD	*Domain Manager* Configuration Descriptor
DoD	Department of Defense

DPD	Device Package Descriptor
DSP	Digital Signal Processor
DTD	Document Type Definition
FPGA	Field Programmable Gate Array
GIOP	General Inter-ORB Protocol
GPP	General Purpose Processor
GPS	Global Positioning System
HCI	Human-Computer Interface
HF	ALE High Frequency – Automatic Link Establishment
HH	Hours
HQ	Have Quick, an electronic counter-countermeasures waveform
HW	Hardware
I/O	Input/Output
ICD	Interface Control Document
ID	Identification, Identifier
IDL	Interface Definition Language
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
IIOP	Internet Inter-ORB Protocol
INFOSEC	Information Security
I/O	input/output
IOR	Interoperable Object Reference
IP	Internet Protocol
ISO	International Standards Organization
Java	Computer Programming Language
JPO	Joint Program Office
JTA	Joint Technical Architecture
JTR	Joint Tactical Radio
JTRS	Joint Tactical Radio System
LAPx	Link Access Protocol x (where x represents 1 of several protocols defined by industry
MAC	Medium Access Control, a sublayer of the OSI Data Link Layer
MIB	Management Information Base
MLS	Multi-Level Security
MM	Minutes
MSB	Most Significant Bit
MSRC	Modular Software-Programmable Radio Consortium
MISSI	Multilevel Information System Security Initiative
N/A	Not Applicable
NAPI	Networking Application Programming Interface
NSA	National Security Agency
OE	Operating Environment
OMG	Object Management Group
OO	Object Oriented
ORB	Object Request Broker

ORD	Operational Requirements Document
OS	Operating System
OSD	Operational Security Doctrine
OSI	Open System Interconnection
OTAR	Over-the-air Re-key
PCI	Peripheral Component Interconnect (bus)
PMCS	Programmable Modular Communication System
PN	Pseudo random Noise
POSIX	Portable Operating System Interface
PPP	Point-to-Point Protocol
PSE52	Real-time Controller System Profile, defined in IEEE Std 1003.13
QoS	Quality of Service
RAM	Random Access Memory
RF	Radio Frequency
RS-232	Electronic Industries Alliance interface standard
RS-422	Electronic Industries Alliance interface standard
RS-423	Electronic Industries Alliance interface standard
RS-485	Electronic Industries Alliance interface standard
SA	Situation Awareness
SAD	Software Assembly Descriptor
SCA	Software Communications Architecture
SCD	Software Component Descriptor
SDD	Service Definition Description
SINCGARS	Single Channel Ground/Airborne Radio System
SLIP	Serial Line Internet Protocol
SNMP	Simple Network Management Protocol
SPD	Software Package Descriptor
SRD	Support and Rationale Document (for the SCA)
SW	Software
TBD	To Be Determined
TBR	To Be Reviewed
TCP	Transmission Control Protocol
TOD	Time of Day
TRANSEC	Transmission Security
UML	Unified Modeling Language
UNIX	A computer operating system developed by AT&T Bell Laboratories
UUID	Universally Unique Identifier
VME	VersaModule Eurocard, a 32 bit data bus standard
XML	eXtensible Markup Language