Title: SpaceWire in the Joint Architecture Standard

Author(s): Leonard Burczyk, Justin W. Enderle, Daniel Gallegos, Paul S. Graham, Richard D. Hunt, Jeffrey L. Kalb, David S. Lee, Jacob E. Leemaster, John M. Michel, and Justin L. Tripp

Intended for: International SpaceWire Conference 2011

SPACEWIRE IN THE JOINT ARCHITECTURE STANDARD

Session: SpaceWire missions and applications

Long Paper

Leonard Burczyk[1], Justin W. Enderle[2], Daniel Gallegos[2], Paul S. Graham[1], Richard D. Hunt[2], Jeffrey L. Kalb[2], David S. Lee[2], Jacob E. Leemaster[2], John M. Michel[1], and Justin L. Tripp[1]

[1]*Los Alamos National Laboratory, Los Alamos NM 87545*

[2]*Sandia National Laboratories, Albuquerque, NM*

E-mail: rdhunt@sandia.gov, jtripp@lanl.gov

ABSTRACT

The Joint Architecture Standard (JAS) is a joint project between Los Alamos National Laboratory and Sandia National Laboratories to provide a common processing and communication infrastructure upon which to more quickly develop payload sensing and processing capabilities. JAS offers a flexible, scalable, and reliable solution to space-based processing for our customer's applications. This standardized architecture is a modular design that allows for rapid prototyping and provides faster system integration and testing that reduces development and integration time and costs. The adaptable architecture meets a wide range of performance requirements including: throughput speeds; reliability; Size, Weight and Power (SWaP) reduction; and mechanical and electrical interfaces. The architecture also allows for evolving design changes while minimizing impacts to established interfaces.

The primary capability enabling technologies in JAS are packet-switched network connectivity and reconfigurable computing. The fundamental architecture of packet-switched networks in JAS are serial interconnects. Because JAS has a broad range of data rate requirements and has the added challenge of providing reliable command, control and data handling in a space environment, this architecture has employed two network tiers connected using Consultative Committee for Space Data Systems (CCSDS) and European Cooperation for Space Standardization (ECSS) communication protocol standards. One of these tiers is driven by high performance gigabit-per-second class communication for high bandwidth sensors and data processing. The other tier is driven by reliable command and control that can also support moderate data transfer rates. SpaceWire is an excellent candidate and is the serial interconnect of choice for the latter tier.

BACKGROUND

The JAS hardware architecture defines several standard hardware *nodes* connected through a minimal number of serial and discrete interconnects. Each of these nodes provides a fundamental capability such that a set of them can be combined to form the basis of a payload data processing system.

The JAS node types are shown in Figure 1. The Configuration and Host Interface Node (CH) provides the interface between the payload and Host Platform. This node contains a radiation-hardened processor to allow it to reliably boot and operate when power is applied to the payload. It runs the application software used for configuring, controlling and monitoring the payload and provides the interfaces to connect the payload to the ground system. The Non-volatile Mass Storage Node (NV) contains non-volatile memory, such as a flash memory, for storing applications and data. The SDRAM Mass Memory Node (SD) contains a large amount of fast and dense memory. It provides a temporary storage capability for processing nodes to manipulate payload data as well as being a communication buffer between nodes. The Reconfigurable Processing Node (RP) and Reconfigurable Sensor Interface Node (RS) contain a large reconfigurable logic device, such as the Xilinx Virtex 5, that can be configured to run hardware applications or a soft-core CPU that runs software applications. They are general purpose, high performance processing node intended to process payload data. The network interface node (NI) provides network connectivity for slower sensors and less performance critical processing. The RS and NI nodes contain an I/O interface that allows program-specific interface boards to be developed to connect to payload hardware devices. The intent of these nodes is to provide the interface to sensors and perform any necessary pre-processing of their data prior to passing it to other RP or SD nodes for final processing and downlink.

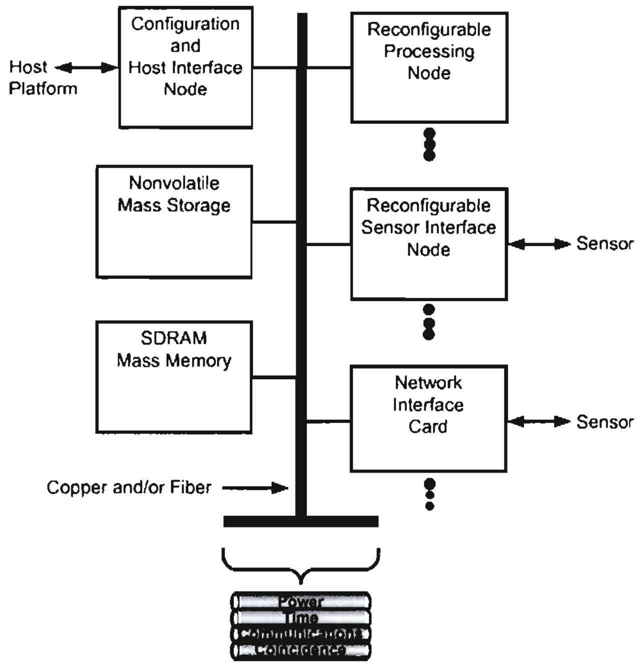


Figure 1: JAS Node Architecture

The number and type of nodes to use in a JAS based payload are determined by program system requirements. Complex custom backplanes are eliminated by having a minimal number of physical interconnects between nodes which allows this node-based architecture to scale to most applications. SpaceWire can be used for routing payload command and state-of-health data as well as moderate bandwidth mission data. For high bandwidth requirements, additional networks based on PCIe, SRIO or custom fast serial can be used.

Each JAS node contains a programmable logic device (FPGA) that provides a set of common functions to the node. This FPGA is referred to as the System Monitoring and Communications (SMAC) device. As shown in Figure 2, the SMAC provides a standard set of physical interfaces for communicating with devices both on and off the

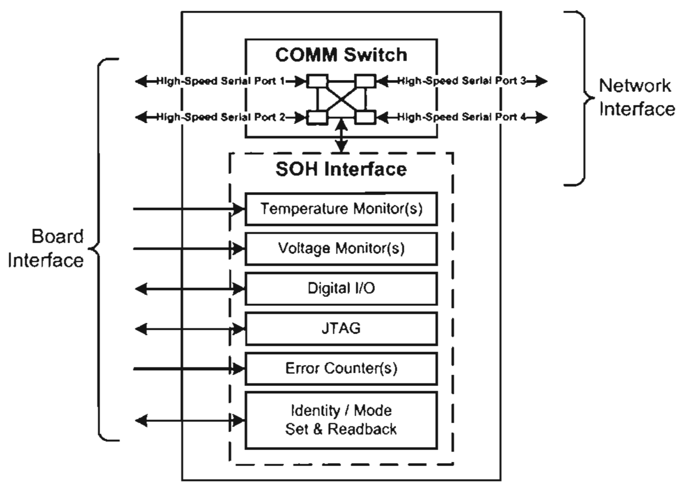


Figure 2: System Monitor and Control Device

node. It includes a SpaceWire network router with at least 5 ports and a suite of serial and parallel I/O interfaces.

The SMAC runs a suite of firmware intellectual property (IP) that provides a standard set of services. SpaceWire communication is provided by router and endpoint cores obtained from NASA Goddard. A Remote Memory Access Protocol (RMAP) core provides a common interface to read and write to memory-mapped peripherals connected to the SMAC. Standardizing on RMAP as a communication protocol reduces the number of protocols that must be supported to communicate with hardware peripherals connected to JAS-based payloads. In addition to the I/O interfaces, RMAP is also used to communicate with other standard devices on JAS nodes such as Point-of-Load (POL) power converters and EEPROM storage devices containing Intelligent Platform Management Interface (IPMI) based node identification records. This storage format defines items like, node capabilities, product and firmware versions, and a unique device identifier. By accessing this information over the SpaceWire network using RMAP, the configuration host node can gather detailed information about each node using a common standard.

The SMAC may contain additional features as well. A SpaceWire broadcast capability can be used by any endpoint to deliver a single SpaceWire message to a variable number of other nodes in an efficient manner. This broadcast capability can be used in conjunction with SpaceWire timecodes to achieve coarse-grained time synchronization between nodes without the use of discrete signals. SelectMAP and JTAG interfaces provide remote configuration and debugging of Xilinx FPGAs over SpaceWire. This library of services will continue to grow as JAS evolves and the SMAC is a versatile and critical component in standardizing the JAS architecture.

Protocol Name	***Value***
Remote Memory Access Protocol (RMAP)	*1*
Reliable Data Delivery Protocol (RDDP)	*238*
JAS Packet Protocol (JPP)	*240*
Goddard Memory Access Protocol (GMAP)	*241*
JAS RDDP (JRDDP)	*242*
Time Protocol	*243*
Broadcast	*245*
Broadcast	*246*

Table 1: SpaceWire Network Protocols

JAS Communication Protocols

There are a number of communication protocols being used on a JAS Spacewire network. Table 1 shows a list of these protocols and their Protocol ID (PID) values. There are two protocols being used from the defined set of ECSS SpaceWire standard protocols [3]. The rest were developed for JAS and assigned values in the user-defined range.

Software applications built on JAS will use a service oriented architecture based on the CCSDS Spacecraft Onboard Interface Services standard [4]. This standard specifies a layered architecture for communicating with devices and applications over serial data links. A representation of the CCSDS SOIS architecture showing the JAS data links and protocols is shown in Figure 3. Services can be implemented as hardware (FPGAs) or as software based on the needs of the application.

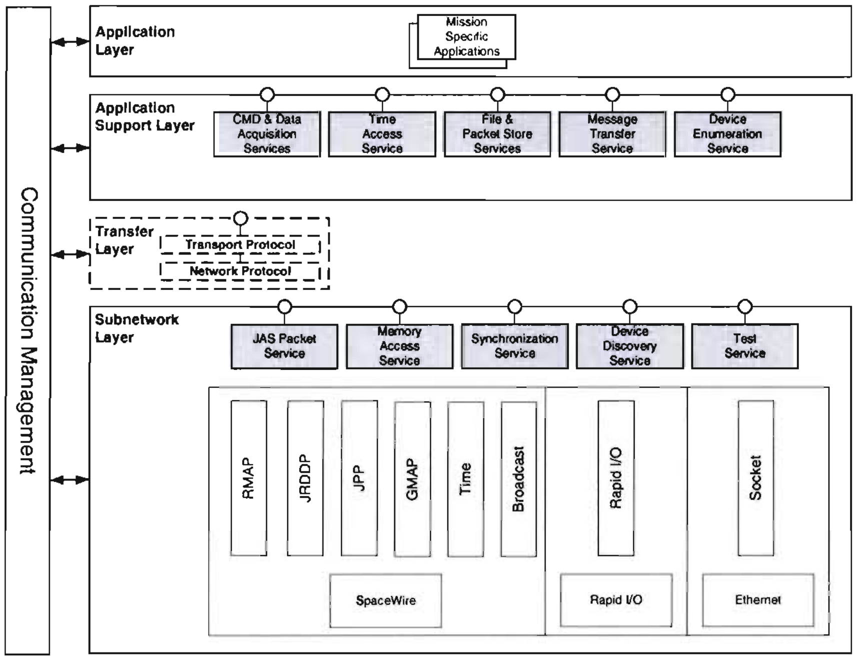


Figure 3: CCSDS SOIS with JAS Protocols

SpaceWire is used by JAS for payload command and control as well as low rate mission data routing. Applications will use one of three fundamental communication protocols for sending data over SpaceWire: the Remote Memory Access Protocol (RMAP); the JAS Reliable Data Delivery Protocol (JRDDP); or the JAS Packet Protocol (JPP). Other SpaceWire protocols are used for specific JAS functions such as router configuration or broadcasting time between the nodes.

RMAP is used to access remote memory based devices across a SpaceWire data link. This protocol is based on the standard managed by the ECSS committee [5]. The protocol itself supports three primary operations: read, write and read- modify-write. While this is not a reliable delivery protocol in that it will not retransmit the commands if there is an error, it does support the ability to notify the sender that the operation was successful. It has the capability of supporting write operations that both verify the data prior to writing it as well as acknowledge that the data was written.

The JAS Reliable Data Delivery Protocol (JRDDP) is a reliable packet transmission protocol used for guaranteed data delivery between two applications over a SpaceWire data link. It is based on the RDDP protocol created by NASA for the GOES-R program and has been modified to make it more flexible so it can meet the needs of JAS payloads [6]. JRDDP consists of essentially two parts, a sender and a receiver. The sender accepts data from a user application, segments into smaller pieces in accordance with the Maximum Transmission Unit (MTU) of SpaceWire, packetizes it for transmission, and then sends it over the data link. Transmission includes a closed loop acknowledgement packet that is returned to the sender to confirm correct delivery of the packet to the remote application. The receiver accepts SpaceWire packets read in off the network and reassembles them to create the original data message. Once reassembled, the data is delivered to the receiving application in the identical form as originally sent. If any errors occur in this transmission, timers will expire on the transmission side, and the sender will try and resend the data for a user-definable number of times.

The JAS Packet Protocol (JPP) provides the capability to send a JAS data packet over a SpaceWire data link. It is a best-effort protocol provides little error checking and no retransmission capabilities. Because of that, it requires little processing overhead which also makes it useful to implement in hardware or for testing purposes. JPP supports sending a single JAS packet within a single SpaceWire packet. The maximum JAS packet size is 64K. Since JAS packets contain a Cyclic Redundancy Check (CRC) as part of their definition, this CRC can used to check the integrity of the JAS packet by receiving applications. The CRC combined with a packet sequence counter, provides the tools necessary for reliable data transfer. In the future, if JAS continues to use this protocol, a segmentation capability will be added for the case that a maximum SpaceWire MTU size is enforced.

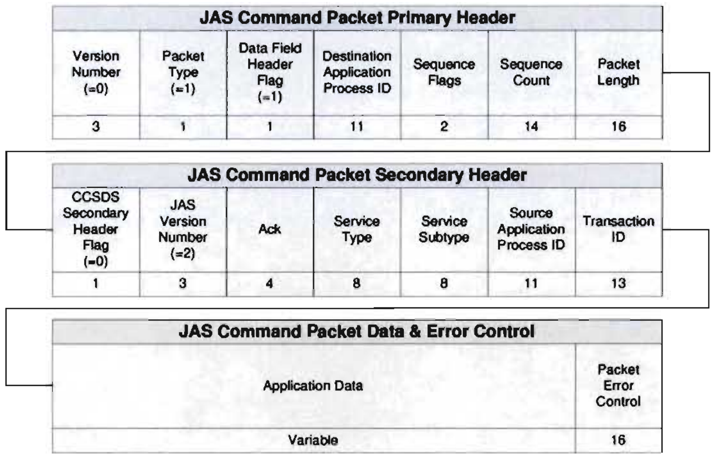
The Goddard Memory Access Protocol (GMAP) is used specifically to configure the Goddard SpaceWire routers, and the GMAP packet format expands on the SpaceWire defined packet. There are three GMAP functions: GMAP Write, GMAP Read, and GMAP Read Response. When sending a GMAP Read request to a Goddard router, the GMAP protocol inserts a variable length reply address field into the packet which the router copies byte-for-byte to the address field of the Read Response packet. The router will also insert a SpaceWire protocol ID into the response packet as well. The return address and protocol ID allow the Read Response packet to be routed to the node that originated the read request and processed by GMAP protocol service. GMAP writes occur without any response from the router so there are no additional capabilities in the router to support this function.

The Time Protocol is used to send an absolute time message from one node to another within a payload. Typically, a single node will maintain the reference time and broadcast it out to all other nodes. This protocol is intended to be used along with SpaceWire timecode packets which are used as the low-latency epoch for telling the receiving nodes that the previously delivered time is valid. This protocol is also intended to be used with the broadcast protocol to enable a coarse-grained time distribution solution for payloads.

The Broadcast Protocol provides the capability for a single node to send a SpaceWire packet to any number of nodes in the system. It uses two different SpaceWire protocol IDs to accomplish this. The combination of the two packet types handles the broadcasting of the packet to all SpaceWire routers and endpoints while eliminating any duplicate deliveries.

JAS provides standard computing and data services for a wide range of sensor systems that may consist of a small simple network of only a few nodes, to a large complex network of hundreds of nodes. To support this flexibility, regional SpaceWire addressing is used to identify the routers. Each router then uses local SpaceWire addressing to access end endpoints for that node. By knowing the topology of the network and thus which routers are neighbors to each other, routing tables can be established which cause the routers to strip the regional address from the SpaceWire packets intended for its neighbors. This enables remote nodes to be able to access other nodes using a combination regional and local addressing scheme with all routers executing standard SpaceWire routing.

To establish the topology and routing tables, JAS implements two options. The first option is a manual process and requires a priori knowledge of the network. Details of the topology which include the physical addressing paths to each router, along with the individual routing tables can be uplinked to the CH node. Using physical addressing along with either the GMAP or RMAP protocol, the CH node loads all the routing tables. The second option uses a network discovery algorithm which, by using physical addressing and polling out of each port of a router, establishes the topology. Then, a routing algorithm establishes the routing tables. This auto-discovery method allows for a quick way to establish the SpaceWire network as nodes are added or deleted giving the system a level of plug-and-play capability.



JAS Command Packet Primary Header						
Version Number (=0)	Packet Type (=1)	Data Field Header Flag (=1)	Destination Application Process ID	Sequence Flags	Sequence Count	Packet Length
3	1	1	11	2	14	16

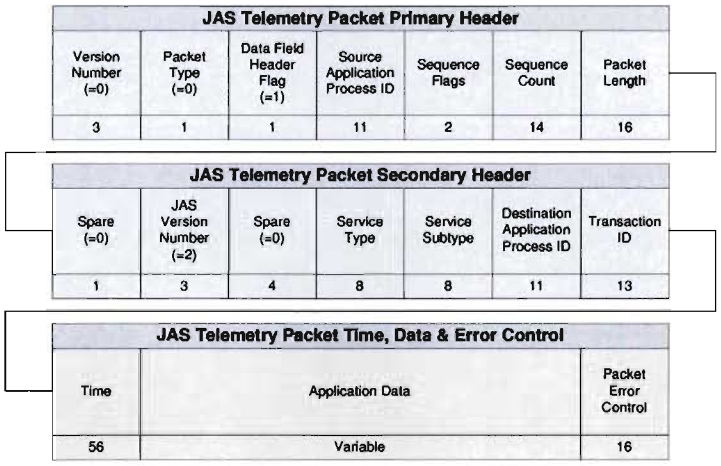
JAS Command Packet Secondary Header						
CCSDS Secondary Header Flag (=0)	JAS Version Number (=2)	Ack	Service Type	Service Subtype	Source Application Process ID	Transaction ID
1	3	4	8	8	11	13

JAS Command Packet Data & Error Control	
Application Data	Packet Error Control
Variable	16

Figure 4: JAS Command Packet

JAS Data Formats

The JAS architecture is designed to be a collection of nodes interconnected through a peer-to-peer network topology based on SpaceWire. Transferring data across the network requires a packetized data format. A logical choice was to use a data format based on the CCSDS Space Packet Protocol Standard [1] and ECSS Packet Utilization Standard (PUS) [2]. A combination of these



JAS Telemetry Packet Primary Header						
Version Number (=0)	Packet Type (=0)	Data Field Header Flag (=1)	Source Application Process ID	Sequence Flags	Sequence Count	Packet Length
3	1	1	11	2	14	16

JAS Telemetry Packet Secondary Header						
Spare (=0)	JAS Version Number (=2)	Spare (=0)	Service Type	Service Subtype	Destination Application Process ID	Transaction ID
1	3	4	8	8	11	13

JAS Telemetry Packet Time, Data & Error Control		
Time	Application Data	Packet Error Control
56	Variable	16

Figure 5: JAS Telemetry Packet

standards where used to create the JAS command and telemetry packet formats (JAS Packets) shown in Figure 4 and Figure 5.The structure of JAS packets are identical to those defined for a CCSDS/PUS packet with the optional fields defined, or additional fields added as needed by JAS. The source and destination Application Identifiers (APIDs) are used describe the source and destination recipients for the packet. The Transaction_ID can be used as a sequence counter by applications that wish to maintain a list of outstanding command requests in which a telemetry response is expected. One might think that the sequence count that is part of the packet primary header could also be used for this but this sequence counter is used when a large data set must be segmented into a number of JAS packets so they can be reassembled on the receive end. The last two important fields are the Service Type and Service Subtype fields. These are used to identify the packet data contents and follow the services that are described in the PUS specification. The only other modification to the standards was to replace the PUS packet version number with the JAS packet version number in the secondary header.

Service Number	*Description*
128	*Device Access Service*
129	*File Access Service*
130	*Platform Management Service*
131	*Time Management Service*
132	*Sensor X Service*
133	*Sensor Y Service*
134	*Test Service*

Table 2: JAS Packet Services

JAS SERVICES

Using JAS packets as the data format for communicating between applications across a SpaceWire network, a set of services were defined to identify the data contents and format. The JAS packet services are based on the PUS service concept. There are a set of standard services described within PUS and these can be used if appropriate but they are targeted more for communication between the payload and ground system. A set of additional on-board services is needed for communication between payload applications. Table 2 shows a subset of these additional services, defined for JAS, along with a brief description of each. The service numbers were chosen based on the user-definable range defined within the PUS specification. Within each service, subtypes were also defined as needed. For example, Table 3 shows two service subtypes defined for the File Access Service. These subtypes correspond to a command packet for requesting the contents of a remote file system and the associated telemetry packet that contains the response.

JAS services provide a straight forward interface to develop applications and provide for a set of standard reusable services as well as extension to mission specific services.

File Access Service						
Service Type	***Service Subtype***	***Subtype Description***	***Cmd***	***Tlm***	***Service Parameters***	***Data Types and Description***
129	*1*	*File System Directory Listing Request*	*X*		*File_System_ID*	*File_System_ID is an unsigned integer that identifies the file system. It is assumed there is only the root directory in the file systems for JAS so a directory identifier is not required*
129	*2*	*File System Directory Listing Report*		*X*	*File_System_ID, Directory_List*	*File_System_ID is an unsigned integer that identifies the file system. Directory_List is a null-terminated string which is contains a list of each file and attributes. Each file is a record separated by a '|' (pipe) character and ends with a '\n' (newline) character. The fields and format of a single entry would look like "file_name|size|modification_time|create_time\n".*

Table 3: File Service Subtypes

CONCLUSION

To date, networks ranging from two stand alone nodes, a configuration host node and a network interface node, to a network of a dozen nodes including multiple reconfigurable processing nodes in a VPX chassis have been demonstrated. Nodes from both Los Alamos and Sandia have been combined and interconnected with SpaceWire for the command and control network. SpaceWire nodes can both be directed from either simulated ground control or networked CH nodes. The rich protocol support and extensibility has made SpaceWire an excellent candidate for reliable communication and is the serial interconnect of choice for reliable command and control at moderate data transfer rates.

REFERENCES

1. The Consultative Committee for Space Data Systems (CCSDS), "Space Packet Protocol", CCSDS 133.0-B-1, September 2003

2. European Cooperation for Space Standardization (ECSS), "Ground Systems and Operations – Telemetry and Telecommand Packet Utilisation", ECSS-70-41A, 30 January 2003

3. European Cooperation for Space Standardization (ECSS), "SpaceWire Protocols", ECSS-E-ST-50-11C Draft 1.3, July 2008

4. The Consultative Committee for Space Data Systems (CCSDS), "Spacecraft Onboard Interface Services", CCSDS 850.0-G-1.1, May 2010

5. European Cooperation for Space Standardization (ECSS), "SpaceWire – Remote Memory Access Protocol, ECSS-E-ST-50-52C, February 2010

6. Sandia National Laboratories, "Joint Architecture Standard Reliable Data Delivery Protocol", JRDDP-001 Version D

7. Sandia National Laboratories, "Joint Architecture Standard Communication Services Specification", JAS-CSS-00001 Version C Draft 4