SpaceWire in the Joint Architecture Standard

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

\textsuperscript{1}Los Alamos National Laboratory, Los Alamos NM 87545
\textsuperscript{2}Sandia National Laboratories, Albuquerque, NM 87185

November 9, 2011
Agenda

- Background
- Overview of JAS
- Use of SpaceWire in JAS
- JAS SpaceWire Protocols
- JAS Packet Data Formats
- JAS Packet Services
Background

- Joint Architecture Standard (JAS) – Jointly developed by Los Alamos National Laboratories and Sandia National Laboratories
- Common processing and communication infrastructure
- Flexible, scalable and reliable solutions for DOE customers
- Wide range of performance, throughput, reliability, etc.
- Inexpensive environment for rapid prototyping and development of data processing platforms
- SpaceWire provides packet switched serial interconnect backbone
Features of JAS

- JAS is a modular, node-based architecture that uses:
  - High-speed serial data interfaces
  - Industry standard protocols
  - Hardware and software building blocks

- Nodes
  - Several processing nodes
  - Mass SDRAM & non-volatile memory nodes
  - Number and type are determined by system requirements

- JAS supports:
  - Rad-hard ASIC processors
  - Reprogrammable FPGAs for HDL algorithm implementations or soft-core processors

- JAS offers COTS-based development and test environment for rapid system demonstration.
Node Overview

Common node elements:

- System Monitoring and Communications device with SpaceWire router
- Processing Element (FPGA/CPU)
- Local POLs for power conversion
- Supporting Electronics (ADCs, Memories, etc.)
SMAC Device

- System Monitor and Communication device is a FPGA on each node that provides:
  - Physical Interfaces
    - SpaceWire Network
    - Serial/Parallel I/O
  - Reusable IP
    - SpW Router (NASA)
    - RMAP
    - SpW Broadcast
    - SelectMAP for Xilinx configuration
    - Remote JTAG over SpaceWire
    - IPMI data for node identification
JAS SpaceWire Protocols

- RMAP is defined in ECSS Space Engineering Standard (ECSS-E-ST-50-52C)
- JAS has defined a number of project-specific protocols
  - JPP is a “best effort” packet protocol
  - GMAP is used to configure NASA-Goddard SpW Router core
  - JRDDP is a more flexible version of NASA’s “guaranteed delivery” packet protocol RDDP
  - Time Protocol provides 1Hz time synchronization between nodes
  - Broadcast Protocols provide a network broadcast capability
CCSDS SOIS with JAS Protocols

- Applications use a communication stack based on CCSDS Spacecraft Onboard Interface Services Standard (SOIS)
- Subnetwork Layer services can be implemented in software or hardware (VHDL)
- Protocols are implemented in routers and end-points as needed
- Applications use RMAP, JRDDP and JPP to communicate
SpaceWire Regional Addressing

- Implements a subnet capability within the network allowing a significant increase in potential endpoints
- Uses SpaceWire logical routing with a two-byte addressing scheme
  - Regional address routes packet to a particular router and is removed
  - Endpoint address routes packet to final endpoint
- Logical Addresses 32-63 are reserved for endpoints and 64-254 are used for routers
- Route tables can be manually loaded or automatically generated using a network discovery algorithm
JAS Packet Format

- Used for all packet-based communication protocols
- Based on CCSDS Space Packet Protocol and ECSS Packet Utilization Standards
- Specifies command and telemetry packet formats
  - Source and destination APID and Transaction ID fields track communication between applications
  - Service Type and Subtype fields identify packet content
Packet Services

- Services identify data contents and format
- Based on the ECSS Packet Utilization Standard
- JAS defines additional service types for onboard communication
- Additional services can be added based on program-specific needs

<table>
<thead>
<tr>
<th>Service Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>Device Access Service</td>
</tr>
<tr>
<td>129</td>
<td>File Access Service</td>
</tr>
<tr>
<td>130</td>
<td>Platform Management Service</td>
</tr>
<tr>
<td>131</td>
<td>Time Management Service</td>
</tr>
<tr>
<td>132</td>
<td>Sensor X Service</td>
</tr>
<tr>
<td>133</td>
<td>Sensor Y Service</td>
</tr>
<tr>
<td>134</td>
<td>Test Service</td>
</tr>
</tbody>
</table>
### A Service Example

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Service Subtype</th>
<th>Subtype Description</th>
<th>Cmd</th>
<th>Tlm</th>
<th>Service Parameters</th>
<th>Data Types and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>1</td>
<td>Device Parameter Request</td>
<td>X</td>
<td></td>
<td>Device ID, Parameter</td>
<td>Device_ID is an unsigned integer that identifies the device. Parameter is an enumerated value that identifies the parameter to read.</td>
</tr>
<tr>
<td>128</td>
<td>2</td>
<td>Device Parameter Report</td>
<td>X</td>
<td></td>
<td>Device ID, Parameter, Value</td>
<td>Device_ID is an unsigned integer that identifies the device. Parameter is an enumerated value that identifies the parameter to read. Value is the parameters value and its type is obtained from a configuration table.</td>
</tr>
</tbody>
</table>

- Services consists of a set of subtypes that define the packet data content
- Each subtype translates to a specific JAS command or telemetry packet
- Service parameters define the content and can be parsed based on standard data types (char, int, float, ...)

---

**Data Types and Description**

- **Device_ID**: An unsigned integer that identifies the device.
- **Parameter**: An enumerated value that identifies the parameter to read.
- **Value**: The parameter's value, obtained from a configuration table.

---

**Device Access Service**
SNL/LANL Interoperability Demo

- Nodes from both LANL and SNL have been combined and interconnected with SpaceWire for the command and control network.
- Hardware was based on a combination of Xilinx evaluation boards (ML-507, ML-605) and JAS prototype nodes.
- Software was independently implemented by each lab based on the JAS Communication Specification.