SpaceWire Plug 'n' Play

Glenn Rakow
NASA GSFC
Code 561
Greenbelt, MD
301-286-5993

Patrick McGuirk
Micro-RDC
Albuquerque, NM
505-294-1962

Clifford Kimmery
Honeywell Inc.
Clearwater, FL
727-539-4234

Paul Jaffe
NRL
Code 8243
Washington, D.C.
202-767-6616

Glenn.P.Rakow@nasa.gov Patrick.McGuirk@micro-rdc.com clifford.kimmery@honeywell.com paul.jaffe@nrl.navy.mil

Abstract—The ability to rapidly deploy inexpensive satellites to meet tactical goals has become an important goal for military space systems. In fact, Operationally Responsive Space (ORS) has been in the spotlight at the highest levels. The Office of the Secretary of Defense (OSD) has identified that the critical next step is developing the bus standards and modular interfaces. Historically, satellite components have been constructed based on bus standards and standardized interfaces. However, this has not been done to a degree, which would allow the rapid deployment of a satellite [3]. Advancements in plug-and-play (PnP) technologies for terrestrial applications can serve as a baseline model for a PnP approach for satellite applications. Since SpaceWire (SpW) has become a de facto standard for satellite high-speed (>200Mbp) on-board communications, it has become important for SpW to adapt to this Plug and Play (PnP) environment. Because SpW is simply a bulk transport protocol and lacks built-in PnP features, several changes are required to facilitate PnP with SpW. The first is for Host(s) to figure out what the network looks like, i.e., how pieces of the network, routers and nodes, are connected together; network mapping, and to receive notice of changes to the network. The second is for the components connected to the network to be understood so that they can communicate. The first element, network topology mapping & change of status indication, is being defined (topic of this paper). The second element describing how components are to communicate has been defined by ARFL with the electronic data sheets known as XTEDS. The first element, network mapping, is recent activities performed by Air Force Research Lab (ARFL), Naval Research Lab (NRL), NASA and US industry (Honeywell, Clearwater, FL, and others). This work has resulted in the development of a protocol that will perform the lower level functions of network mapping and Change Of Status (COS) indication required by Plug 'n' Play over SpaceWire. This work will be presented to the SpaceWire working group for standardization under European Cooperation for Space Standardization (ECSS) and to obtain a permanent Protocol ID [2]. The portion of the Plug 'n' Play protocol that will be described in this paper is how the Host(s) of a SpaceWire network map the network and detect additions and deletions of devices on a SpaceWire network.

[1]

[1] 1-4244-0525-4/07/$20.00 ©2007 IEEE.

[2] IEEEAC paper #1211, Version 6, Updated January 8, 2007

TABLE OF CONTENTS

1. INTRODUCTION

Plug 'n' Play (PnP) has traditionally been associated with consumer electronics to support rapid and seamless integration among commercial products or to incorporate add-on components. The space industry led by the US Department of Defense is now adopting the PnP approach to address the need for quicker integration time of intelligence, communications, and other satellites. The ultimate goal is less than one week from identified need to launch. The effort falls under the umbrella of the Office of Force Transformation's (OFT) Operationally Responsive Space (ORS) initiative. The Air Force Research Lab (AFRL) has been refining concepts and developing hardware to support PnP for network interfaces designed for satellites. Much of the earlier work has focused on the USB 1.1 standard upon which the electronic data sheet (XTEDS) would be layered to provide the required information necessary for devices to communicate. This concept allows a host or multiple hosts

to read the device driver from a component and start communicating. This XTEDS work has been standardized by AIAA.

SpaceWire's growing popularity has resulted in great interest in incorporating PnP features. As background, SpW (ECSS-E-50-12A) is a simple and flexible protocol that is both small in area and low in power enabling it to be embedded in FPGAs. It is currently the only standard that was developed from the ground up for spacecraft applications. Additionally, the possibility of layering changes onto the bare protocol is made feasible through the use of the Protocol ID, as defined in a new standard, ECSS-E-50-11. With the Protocol ID, various protocols may be layered on SpaceWire and advanced features can be introduced.

This is the approach that has been taken to support the lower level features necessary to support PnP. These low level features include the ability to map the initial network topology and subsequently indicate the Change of Status (COS) that would result with the addition or deletion of network resources. These are basic functions that must be in place before upper layer functions can read information such as the device identity and the device driver.

This paper focuses on these low level features, network mapping and COS because they must be done in hardware, whereas other PnP functions may be offloaded to software. It is important for a single SpW Protocol ID to be permanently assigned to support these two low level features.

2. SpaceWire Basics

In order to understand the technical aspects of this paper, a basic understanding of SpaceWire from the Network and Packet level is required. SpaceWire is really a Data Link protocol as defined by the Open Systems Interconnection (OSI) and Consultative Committee for Space Data Systems (CCSDS) protocol stacks. Within SpaceWire, a separate definition of levels that are encompassed by the Physical Layer and Data Link Layer are defined. These are as follows: Physical Level; Signal Level; Character Level; Exchange Level; Packet Level; Network Level. See section 9, "SpaceWire Network and Packet Level Overview" for more details.

3. Requirements

Below are the requirements to support network mapping and COS indication.

The requirement for mapping the network is that the SpW router shall provide the following upon request by the host(s): unique Router ID; number of ports attached to router; and the status of ports attached to the router. This is the basic information necessary for the host(s) to determine the interconnectivity of the routers and nodes in an SpW network.

The requirement for COS indication, which is the ability of the SpW router to detect an attachment/detachment event and notify the host(s) is as follows:

The SpW network shall:

A) recognize the attachment/detachment of nodes/subnets as a notification event.

B) distribute notification events to the host(s) with best effort.

The notification packet shall:

A) identify the type of notification event (attachment or detachment)

B) identify the attached/detached port

4. Assumptions

The assumptions for the implementation of these requirements are as follows:

A) The protocol can require modification of router designs, but not end point designs

B) The protocol should:

1) support path addressing
2) support logical addressing
3) minimize reservation of route resources
4) minimize notification packet size

B) Notification should be inactive in un-initialized SpW networks

5. General Approach

The general approach for the PnP low-level functions of network mapping and COS is to put the logic in the router hardware. The end nodes do not have to be changed. Using the Protocol ID format to specify the packet as a PnP packet, a router can be queried by a host to determine the identity of the router, i.e., Router ID; the number of ports and number of active ports attached to the router. This information will allow mapping of the network by the host.

When attachment or detachment events occur, a notification packet for the COS will be generated and sent by the router hardware to the hosts.

6. Proposed Solution

The specification for the network mapping packet format is straightforward as it is only sent when requested by a host and contains the general information necessary for network mapping as described above.

The solution for the notification of COS is much more complex and several approaches have been devised and evolved to come up with the solution.

Each router will reserve a number of slots, N, that are used for the return addresses to the hosts. Each host has a slot where it's return address is written. Hosts fill these slots in any arbitrary order. Note that one slot provides enough bytes to properly define the return address. This may be a single byte in the case of logical addressing or multiple address bytes in the case of path addressing, a scheme that uses concatenated address bytes and header deletion. The number of reserved slots is implementation specific, but consider that 32 hosts with 16 paths per slot would amount only to a manageable 512 bytes.

When an attachment or detachment event occurs, the router that detected the COS issues a notification packet identified by the PnP Protocol ID to all valid slots defined as hosts. The router that detected the COS event includes its unique Router ID within the notification packet so that the host(s) can determine where the network change occurred.

6.1 Slot Usage

Slots are used to specify the address to the hosts. Because Path addressing may be used, multiple bytes may be needed for the return address. For logical addressing only one byte

	Notification Slots				Memory Mapped Address
Slot 0	P3	P2	P1	P0	0xC00
	P7	P6	P5	P4	0xC04
Slot 1	P3	P2	P1	P0	0xC08
	P7	P6	P5	P4	0xC0C
Slot 2	P3	P2	P1	P0	0xC10
	P7	P6	P5	P4	0xC14
Slot 3	P3	P2	P1	P0	0xC18
	P7	P6	P5	P4	0xC1C

Memory Size is 8 by 32

Memory mapping to an address space is optional

Figure 1 Slot Definition

is required (see Figure 1). At power-up all slot table values are initialized to zeros. The router marks valid slots as they are programmed. Leading zeros in the address are deleted before including the address in the SpW packet. Slot size can be tailored by the router implementation. Both number of paths and number of slots are independent.

6.2 Slot Considerations

Memory storage for the slot table is user definable and Remote Memory Access Protocol (RMAP) and PnP Protocols can co-exist for slot and Router ID housekeeping, i.e., map PnP parameters to RMAP memory space (see Figure 2).

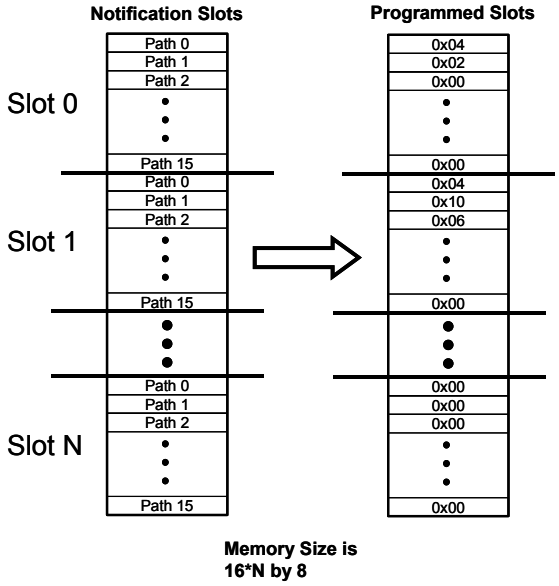

Figure 2 Slot Considerations

6.3 Attachment Event

Figure 3 shows an attachment event to illustrate how a notification packet is sent to the hosts. The green node triggers the notification event from its attached router. The router builds the notification packet using the PnP protocol ID and then sends it to the hosts using the addresses as described in the slot descriptions. Upon receiving the notification, the host examines the Router ID inside the packet to determine which router issued the event and takes appropriate action. This may include some unspecified higher layer action such as reading the device driver, etc.

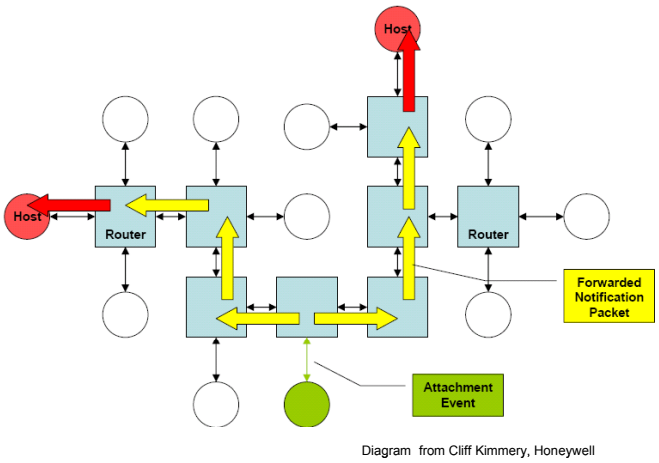

Figure 3 Attachment Event

This approach requires each router to have a unique ID that is incorporated into the notification packets. It also assumes that all host nodes have access to the network map.

6.3 PnP Packet Structure

The packet format of the PnP packet conforms to the Protocol ID format now being standardized under ECSS-E-50-11. The packet contains a Packet Type byte after the Protocol ID byte that specifies different packets used by the PnP Protocol ID (see Figure 4).

The different packet formats necessary for a PnP protocol are as follows:

A) Ping/Query message. The Host polls routers for information for network mapping (status of ports and router ID).
B) Notification Event. Router generates to notify host(s) of an attachment/detachment event.
C) Slot Housekeeping. Supports slot writes, slot read requests and slot return reads.
D) Router ID Housekeeping. Supports Router ID writes, read requests and return reads.
E) Programmable detachment inactivity time. Could be used to disable notification messages.
F) Router Diagnostics. Used to gather additional information from the router for debugging.

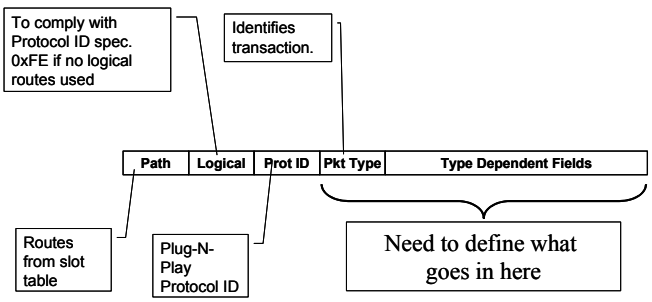

Figure 4 PnP Packet Format

7. SUMMARY

A general approach to solving the lower level utilities necessary to support a PnP network over SpW has been proposed that will leverage the Protocol ID format now being standardized. This approach is implemented in the SpW routers only. It is meant to be used with an upper layer software interrogation mechanism. This operation is relegated to higher-level network functions, thus leaving the SpW portion of the node design intact.

8. DEFINITIONS

Detachment – a link becomes inactive (unrelated to error conditions) after some period of activity

Attachment – a link becomes active after some period of inactivity unrelated to error conditions

Node – an addressable network entity that is not a router

Host node – a node that is the destination for notification packets

Subnet – a separable collection of internally connected routers and nodes

Change of Status (COS) – either an attachment or detachment event

Network mapping – the ability of the host or hosts to determine the connections in the network between routers and nodes

9. SPACEWIRE NETWORK AND PACKET LEVEL OVERVIEW

This section describes the basic elements of the SpaceWire Network Level and Packet Level. The Packet level describes the packet format for SpaceWire. The Network level describes how packets are routed over a network of routers and nodes [1].

9.1 Packet Level

The SpaceWire packet level defines a packet to have a Destination Address at the front of the packet, to be followed by one or more data bytes and to be terminated by an End-Of-Packet (EOP) marker. The packet length is not limited. The Destination address may be one or more bytes depending upon the addressing scheme (see Path Addressing, section 3.7 and Logical Addressing 3.8 in ECSS-E-50-12A).

9.2 Network Level

SpaceWire networks consist of point-to-point links interconnected between routers and nodes (end users). This

interconnection media or switched fabric provides the network over which packets flow. The Network Level describes the routing, addressing, arbitration and error recovery of SpaceWire packets.

9.3 SpaceWire Router Switch

A SpaceWire Router is a non-blocking cross bar switch that allows connections among a group of ports (up to 31). A port is defined as either a SpaceWire serial link or local parallel port. Connection may be made between any port input and any other port output. As long as a port's output is available, it may be used. If two or more port's inputs are requesting the same destination port, then arbitration occurs. The arbitration outcome will result in one connection between an input and destination. The arbitrator ceases the data flow from the losing port, which must wait for the destination port to be available.

9.4 Wormhole Routing

SpaceWire implements a routing scheme called "wormhole" routing. It may be described by comparing it to another scheme called "store and forward". In "store and forward" routing, the whole packet must be buffered at a receiver and processed before it may be passed to the next destination in the network. In "wormhole" routing, only the destination address of a packet must be received and processed before the connection can be made to pass data to the next destination in the network. This may result in a packet being physically located in many different buffers at the various routers in the network. This has the benefit of reduced packet latency across the network and allows very small buffering in the receive ports through the network. As the packet "worms" its way through the network, the EOP marker at the tail of the packet (worm) is pulled through routing switches. When this happens, connections in the switch are broken thus allowing re-arbitration for the connection resources.

"Wormhole" routing may cause congestion in the network when a packet stalls due to the unavailability of a port. However, this behavior does not result in the loss of data because Flow Control Tokens (FCTs) prevent data loss. FCTs are exchanged between links in the network to control the flow of data across the different network hops. The number of FCTs received represents the amount of buffer space in the receiver at the other end of the link. Each FCT represents space for eight (8) bytes of data. When a nodes FCT credit is zero, that node may not send data until an FCT is received. This results in a stalled link but no data loss.

9.5 Arbitration

The routing switches may perform Priority, round robin or random arbitration schemes. Most implementations provide for at least round-robin (fair) arbitration.

9.6 Routing Scheme

The main routing schemes that SpaceWire provides are Path Addressing and Logical Addressing. All other schemes are based upon these two basic methods with additional processing. Addressing schemes may be mixed within the same packet.

9.7 Path Addressing

Ports associated with a non-blocking crossbar switch, i.e., router, are assigned a hardwired port number that is used by the router for port identification. Up to 31 external ports may be included in a SpaceWire router. Each external port is associated with a port number starting from 1 up to 31. Path Addressing is a mechanism that specifies the destination port number directly in the value of the Path Address byte. This method does not require the use of a look-up table. For example, destination address 5 would be routed to port number 5. Path addressing requires the destination address to be deleted upon leaving the router, i.e., header deletion. Thus, in order to transverse multiple routers, the destination address must have multiple bytes, each byte defining a Path address for the next router in the path the packet must travel. Path Addressing therefore may have a large overhead. Path Addressing is very useful for network initialization when routing information is not present in the network.

9.8 Logical Addressing

A lower overhead solution when compared to Path Addressing is Logical addressing. Logical Addressing provides an association between a unique number and the physical port number (Path Addressing range). Logical Addressing therefore uses a look-up table in the router to provide the association. A Logical Address is distinguished from a Path Address by the address range. Logical addresses are defined to be in the range of 32 to 254 (255 is reserved). Upon a logical address entering a router, it is used to look up the appropriate physical port number. Logical addresses may be deleted or not depending upon the information in the look-up table. This is a more bandwidth efficient method of routing packets if multiple routers are in the path. Logical Addressing may be used with Path Addressing in the same packet. This can be useful for configuring a router during operation.

9.9 Configuration Space

Destination address "0" is a reserved destination address in the SpaceWire network. Address '0' identifies the configuration space port in a router. A packet that arrives at a router with a destination address of "0" is directed to the configuration space of the router. This packet format is not defined in the standard and it is therefore implementation specific. The router configuration is necessary for programming the port mapping and other configuration information, i.e., control of link speed, etc.

9.10 Packet Recovery from Error

Error recovery on a network needs to be handled in a graceful and predictable way. This is very straightforward with a "store and forward" system where the packet contents are completely buffered and checked before being sent to the next hop in the network. For a "wormhole" routing scheme it is a little more complicated since parts of the packet may be physically located in many different buffers across the network. When the error condition occurs, it happens on one link and it is handled between the ends of that physical link, so that the rest of the network, which is "worming" the packet, has no knowledge (nor does it require knowledge) that the error occurred. The actions taken by the two ends of the link where the error occurred are as follows.

1. The link re-initializes per the SpaceWire Link Initialization state machine.
2. When a link re-initializes, the next packet sent by the transmitter is the beginning of a good known packet.
3. If a packet was in transit while the error occurred and the EOP maker did not pass the transmitter yet, then the transmitter has to "spill" or "consume" the remainder of the old partial packet up to the EOP marker.
4. After it does this, it sends the next good packet it has to send.
5. Likewise, after link re-initialization, the receiver must be ready to accept the beginning of a good known packet.
6. So if the receiver did not receive the EOP for the packet that was in transit when the error occurred, it must mark the end of that incomplete packet with an Error-End-of-Packet (EEP) maker. This EEP will transverse the network just as if it were a good EOP marker and be interrogated by the destination to determine packet quality.

In this way a link error does not propagate to the entire switched fabric (network).

10. ACRONYM LIST

AFRL	Air Force Research Lab
CCSDS	Consultative Committee for Space Data Systems
COS	Change Of Status
COTS	Commercial Off The Shelf
ECSS	European Cooperation for Space Standardization
EOP	End-of-Packet
EEP	Error-End-of-Packet
ESA	European Space Agency
FCT	Flow Control Token
FPGA	Field Programmable Gate Array
GSFC	Goddard Space Flight Center
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
NASA	National Aeronautics and Space Administration
NRL	Naval Research Lab
OFT	Office of Force Transformation
ORS	Operationally Responsive Space
OSI	Open Systems Interconnection
PnP	Plug and Play
RMAP	Remote Memory Access Protocol
SpW	SpaceWire
XTEDS	Electronic Data Sheet

REFERENCES

[1] Doyne, T., Wegner, P., Riddle, R., Hurley, M., et al. (2006, April). A TacSat and ORS Update Including TacSat-4. 4th Responsive Space Conference, Los Angeles, CA, USA

[2] Glenn Rakow, Richard Schnurr, Steve Parkes, "SpaceWire Protocol ID: What Does It Mean To You?", IEEE Aerospace, Big Sky, Montana, March 2006

[3] ECSS-E-50-12A, "Space Engineering: SpaceWire – Links, nodes, routers and networks" ESA-ESTEC, 24 January 2003

BIOGRAPHY

Glenn Rakow is the NASA representative to the SpaceWire working group. Since 1998 he has worked with government and the US industry in the development and enhancement of SpaceWire. He has been part of SpaceWire efforts with Swift, JWST, LRO, GOES-R and MMS as well as formulation mission support and technology developments incorporating SpaceWire. He earned a Bachelor of Science in Electrical Engineering from the University of Maryland, College Park, in 1988, and a Master of Science in Electrical Engineering from George Washington University, Washington D.C., in 1997.

Patrick S. McGuirk is a digital design engineer at Micro-RDC, a company specializing in custom hardware development and radiation hardened electronics. He has designed and implemented FPGA solutions for a variety of applications, including SpaceWire, USB, FFTs, digital filters, correlators, image compression & processing, and general command & control functions. He earned his B.S. degree in electrical engineering from the United States Air Force Academy in 1992, and his M.S. degree in electrical engineering from the University of Florida in 1994.

Cliff Kimmery, a Fellow at Honeywell in Clearwater, Florida, has worked on a variety of government-sponsored projects and space missions. Currently he is developing hardware architectures for advanced satellite payload applications. His areas of interest are system architectures, system performance modeling, communications protocols, processor architectures and fault-tolerant architectures. He earned a Bachelor of Science in Electrical Engineering from the University of South Alabama in 1979.

Paul Jaffe is an electronics engineer at the Naval Center for Space Technology at the Naval Research Laboratory. He has worked on several NASA and Department of Defense space missions, including STEREO and TacSat-1. Currently he is developing hardware and standards that include SpaceWire for TacSat-4 as part of the Office of Force Transformation's Operationally Responsive Space effort. He earned a Bachelor of Science in Electrical Engineering from the University of Maryland, College Park in 1996, and he is currently pursuing a Master of Science in Electrical Engineering at the Johns Hopkins University.