SpaceWire Protocol ID: What Does It Means To You?

Glenn Rakow	Richard Schnurr	Daniel Gilley	Steve Parkes
NASA GSFC	NASA GSFC	Lockheed Martin	University of Dundee
Code 561	Code 561		
Greenbelt, MD	Greenbelt, MD	Sunnyvale, CA	Dundee, Scotland, UK
301-286-5993	301-286-1852		0044 1382 34 5194
Glenn.P.Rakow@nasa.gov	Richard.G.Schnurr@nasa.gov	daniel.gilley@lmco.com	sparkes@computing.dundee.ac.uk

Abstract— SpaceWire is becoming a popular solution for satellite high-speed data buses because it is a simple standard that provides great flexibility for a wide range of system requirements. It is simple in packet format and protocol, allowing users to easily tailor their implementation for their specific application. Some of the attractive aspects of SpaceWire that make it easy to implement also make it hard for future reuse. Protocol reuse is difficult because SpaceWire does not have a defined mechanism to communicate with the higher layers of the protocol stack. This has forced users of SpaceWire to define unique packet formats and define how these packets are to be processed. Each mission writes their own Interface Control Document (ICD) and tailors SpaceWire for their specific requirements making reuse difficult. Part of the reason for this habit may be because engineers typically optimize designs for their own requirements in the absence of a standard. This is an inefficient use of project resources and costs more to develop missions.

A new packet format for SpaceWire has been defined as a solution for this problem. This new packet format is a compliment to the SpaceWire standard that will support protocol development upon SpaceWire. The new packet definition does not replace the current packet structure, i.e., does not make the standard obsolete, but merely extends the standard for those who want to develop protocols over SpaceWire.

The SpaceWire packet is defined with the first part being the Destination Address, which may be one or more bytes. This is followed by the packet cargo, which is user defined. The cargo is truncated with an End-Of-Packet (EOP) marker. This packet structure offers low packet overhead and allows the user to define how the contents are to be formatted. It also provides for many different addressing schemes, which provide flexibility in the system. This packet flexibility is typically an attractive part of the SpaceWire.

The new extended packet format adds one new field to the packet that greatly enhances the capability of SpaceWire. This new field called the Protocol Identifier (ID) is used to identify the packet contents and the associated processing for the packet. This feature along with the restriction in the packet format that uses the Protocol ID, allows a deterministic method of decoding packets that was not before possible. The first part of the packet is still the Destination Address, which still conforms to the original standard but with one restriction. The restriction is that the first byte seen at the destination by the user needs to be a logical address, independent of the addressing scheme used. The second field is defined as the Protocol ID, which is usually one byte in length. The packet cargo (user defined) follows the Protocol ID. After the packet cargo is the EOP, which defines the end of packet. The value of the Protocol ID is assigned by the SpaceWire working group and the protocol description published for others to use.

The development of Protocols for SpaceWire is currently the area of greatest activity by the SpaceWire working group. The first protocol definition by the working group has been completed and is now in the process of formal standardization. There are many other protocols in development for missions that have not yet received formal Protocol ID assignment, but even if the protocols are not formally assigned a value, this effort will provide synergism for future developments.

TABLE OF CONTENTS

1. INTRODUCTION

This paper describes a new addition to SpaceWire packet format that will compliment the existing SpaceWire standard, ECSS-E-50-12A, 24 January 2003. This new packet definition is necessary so that applications using SpaceWire can be more powerful. The new Protocol ID now allows different protocols to be transported over SpaceWire and to be easily identified in a standardized way. It also allows these protocols to be reused because they may be standardized.

2. BACKGROUND

National Aeronautics and Space Administration (NASA), European Space Agency (ESA), and Japanese Aerospace Exploration Agency (JAXA) are just a few of the many groups using SpaceWire. NASA has developed four satellite architectures around SpaceWire with one on-orbit (Swift) and with many more in the formulation phase. ESA is the agency that standardized SpaceWire and has all their spacecraft missions base-lining the use of SpaceWire. The JAXA is also very active in developing SpaceWire hardware and using it on spacecraft applications.

SpaceWire is a simple protocol, which is one reason that it is gaining popularity. Some times simplicity can cause more difficulty in later stages of the system engineering process. This is the case for SpaceWire's simple packet format, because the packet contents needs to be defined along with it's processing. Each mission that uses SpaceWire needs to develop unique packet formats and protocols. These packet formats, and the necessary processing are naturally optimized for a particular application. This is done perhaps because there are not many other alternatives available. There are some applications that are so specialized that a previous solution does not work, but this is not always the case. Usually, if a standard exists and can be made to work for an application, it is a benefit.

3. SPACEWIRE NETWORK & PACKET LEVEL OVERVIEW

This section describes the basic elements of SpaceWire Network Level and Packet Level, which is additional background context for the Protocol ID. The Packet level describes the existing packet format for SpaceWire. The Network level describes how packets are routed over a network of routers and nodes

3.1 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 truncated 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)

3.2 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.

3.3 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's inputs to any other port's output. As long a port's output is available, it may be used. If two or more port's inputs are requesting the same port's output than arbitration occurs. The arbitration outcome will result in one connection between an input and an output. The arbitrator holds off the other port's inputs and they must wait so their flow of data stalls.

3.4 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" it's 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 will not result in the lost of data because Flow Control Tokens (FCTs) will 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 for the other end of the link's receiver. Each FCT represents space for eight (8) bytes of data. When the FCT credit is zero at a nodes receiver that nodes transmitter may not send data until an FCT is received. This will result is a stalled link but no data loss.

3.5 Arbitration

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

3.6 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.

3.7 Path Addressing

Ports associated with a non-blocking cross-bar switch, i.e., router, are assigned a hardwired port number, which is used by the router for identification. Up to 31 ports may be attached to a SpaceWire router. Each 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 represents 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.

3.8 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. Logical Address is distinguished from 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 looked up and mapped to a 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 may be useful for programming a router during operation, see Configuration Space, section 3.9).

3.9 Configuration Space

Destination address "0" is a reserved destination address in the SpaceWire network. Address '0' represents a router's configuration space. 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 routing table's port mapping and other configuration information, i.e., control of link speed, etc.

3.10 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. The link will re-initialize per the SpaceWire Link Initialization state machine. When a link re-initializes, the next packet sent by the transmitter will be the beginning of a good known packet. If a packet was in transit while the error occurred and the EOP maker did not pass the transmitter yet, than the transmitter will have to "spill" or "consume" the remainder of the old partial packet up to the EOP marker. After it does this, it will than send the next good packet it has to send. Likewise, after link re-initialization, the receiver must be ready to accept the beginning of a good known packet. 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 the packets quality. In this way a link error does not propagate to the entire switched fabric (network)

4. PROBLEM

With this background, it may be understood that one problem with the SpaceWire standard is that it is not easy for the destination user of a packet to know the contents of the packet. This is especially true if multiple packet structures exist. In most systems more than one packet format is necessary for each user and these packets usually require different processing per packet type. More

specifically, it is not known if the SpaceWire destination address is present or not. Does the packet presented to the user contain the logical destination address? The system may use logical addressing but the address may be stripped off at the last route. Path addressing may be used, in which case no destination address information will be present. This requires the system engineer to restrict a prior the addressing scheme so that it is known what to expect. This is not a useful situation for system reuse. The power of a standard is that it may be reused by different systems. There needs to be a packet format that all users can interrupt in a standardized way. However one of the nice features of SpaceWire is that the packet has low overhead and it is not very restrictive in packet format, i.e., it does not define packet length and it has a very simple header. So it is necessary to impose only the minimum amount of information necessary to accomplish the goal of packet identification. With this ability, there can be development and reuse of upper layer protocols over SpaceWire from a large community.

5. SOLUTION

5.1 New Packet Format

To solve these problems, i.e., no knowledge of logical address presence, no ability to identify packets, and to promote reuse, a new packet format has been defined. The new packet definition is an extension of the present SpaceWire packet format. It will allow a deterministic method to identify the packet contents so that upper level protocols may be developed in a standardized way. This new packet format does not affect the existing standard, ECSS-50-12A, if it does not want to be used. It only affects how the user interrupts the packet, i.e., layers above the SpaceWire standard.

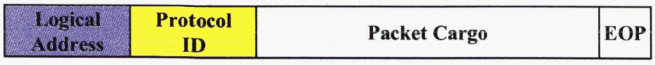


Figure 1. Logical Address with Protocol ID

5.2 Logical Addresses

The new packet format requires that the first byte that arrives at the user destination be a logical address (see Figure 1). This does not mean that Path Addressing cannot be used, as it will be necessary especially at start-up for initializing the various routers' configuration spaces' across the network. All present addressing schemes in the SpaceWire standard may be used (see Figure 2). If the logical destination address does not exist, or if the source of the packet does not know it, than the logical address of 254 decimal is reserved as a dummy value for this case. Since logical address 254 Decimal is a default address when the destination is not known, this value may be ignored by the destination. The source therefore just needs to insert into the packet format a logical address weather it is used to route the packet or not so that is will appear first at the destination. This new packet format uses the Logical Address because most systems will probably use the logical address as it provides useful information to be decoded by the destination. It may be used to decode a particular function within the destination or it may be used as a filter or even as a source address if desired.

5.3 Protocol Identifier

The second byte called Protocol Identifier (ID) is the new field that was not defined in the current standard. It is used for decoding the packet format and defining the processing necessary for the packet. With the Protocol Identifier, different upper layer protocols may use the same SpaceWire network and be interpreted by the user. This is a powerful new addition to SpaceWire, which will encourage protocol development and reuse. The protocol identifier is a value that will be assigned by the SpaceWire working group and be published with a specification of the packet format and processing.

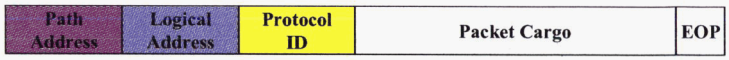


Figure 2. Path Address with Protocol ID

5.4 Extended Protocol Identifier

The Protocol ID value of "00"Hex is reserved and may be used to extend the Protocol ID field. When the first byte of the Protocol ID is "00" Hex than two additional bytes are added to make the Protocol ID field three bytes in length. This will allow a 16 bit extended protocol ID field to permit up to 65535 protocols to be sent over SpaceWire. Implementation of the extended Protocol ID field is not required and if not implemented than a value of "00" Hex shall be ignored. Protocols that have the same value weather they are extended or non-extended shall be interpreted as the same protocol. Lastly, an extended Protocol ID of "000000" Hex shall be ignored.

5.5 Unknown Protocols

Packets that arrive at the user destination with an unknown Protocol ID or an unsupported protocol shall be ignored and they shall be consumed at the user destination. The user shall also keep a count of unknown protocols that are received.

5.6 Protocol ID Allocation

Protocol ID values in the range of 1 to 239 Decimal shall be assigned by the SpaceWire working group. Protocol ID values in the range of 240 to 254 Decimal shall be available for experimental use and will not be assigned protocols definition by the SpaceWire working group. Proven protocols may be recommended for adoption to the SpaceWire working group and may be assigned a permanent value and documented in the new standard, ECSS-E-50-11.

6. SPACEWIRE PROTOCOLS

This section describes just some of the protocol developments. There are other informal proposals in Europe and the United States that are not mentioned in this paper.

6.1 Remote Memory Access Protocol

The first protocol assigned a Protocol ID is called the Remote Memory Access Protocol (RMAP). RMAP describes a method for writing and reading memory mapped locations over a SpaceWire network. It has features that allow it to be used for reliable transfers with additional definition at the next higher layer, as well as optional Cyclic Redundancy Code (CRC) checks for header and data separately. It also allows DMA accesses. This was the first protocol developed by the SpaceWire working group and was assigned the Protocol ID value of 1. This protocol will be standardized into a new document by ECSS. This new standard, ECSS-E-50-11, will document the use of the Protocol ID and describe RMAP. Future protocols developed by the working group will probably be added to this standard document.

6.2 GOES-R Reliable Transport Protocol

The GOES-R project at NASA's Goddard Space Flight Center (GSFC) has developed a protocol under the experimental Protocol ID range that will provide reliably transport of packets between a source and destination over a SpaceWire network. The protocol uses a positive acknowledgment mechanism on a per packet basis with retransmission of the packet if the acknowledgment has not been received for a defined amount of time. It provides for an alternative path for retransmission to facilitate physical level redundancy.

6.3 LRO Protocols

The LRO mission is developing several protocols to use over SpaceWire. Two of the protocols are Consultative Committee for Space Data Systems (CCSDS) formats. One is a CCSDS packet and the other is a CCSDS transfer frame using the Advanced Orbiting Systems (AOS) Virtual Channel Data Unit (VCDU). The other protocol is an instrument specific format. These Protocol IDs are in the experimental range, as they have not been officially assigned a Protocol ID by the SpaceWire working group. The CCSDS protocol ID will probably be assigned permanent numbers.

6.4 Protocol Development

The SpaceWire working group encourages recommendations for new protocol developments and welcomes all those interested. Industry and government participation is necessary to develop useful protocols for Space applications

7. SUMMARY

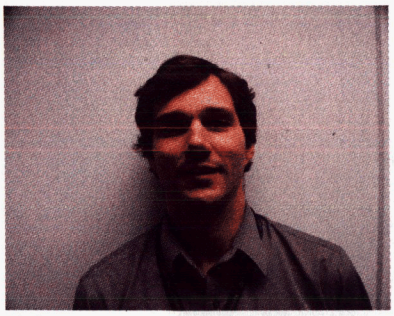
With the large user community for SpaceWire, the use of the Protocol ID will increase synergism around the standard. This should translate into savings in satellite development costs.

8. ACRONYM LIST

AOS	Advanced Orbital System
CCSDS	Consultative Committee for Space Data Systems
CRC	Cyclic Redundancy Code
ECSS	European Cooperation for Space Standardization
EOP	End-of-Packet
EEP	Error-End-of-Packet
ESA	European Space Agency
FCT	Flow Control Token
GOES	Geostationary Orbiting Environmental Satellite
GSFC	Goddard Space Flight Center
Hex	Hexadecimal
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
ICD	Interface Control Document
JAXA	Japanese Aerospace Exploration Agency
JWST	James Webb Space Telescope
LRO	Lunar Reconnaissance Orbiter
NASA	National Aeronautics and Space Administration
RMAP	Remote Memory Access Protocol
VCDU	Virtual Channel Data Unit

9. 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 and GOES-R 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, Collage Park, in 1988, and a Master of Science in Electrical Engineering from George Washington University, Washington D.C., in 1997.



Richard Schnurr is the GSFC representative to the CCSDS Standard On board Interfaces (SOIF) panel as well as the Chief Architect of the Electrical Engineering division. In these roles Mr. Schnurr is working with many missions in early formulation to design standard on-board networks meeting specific mission requirements. Mr. Schnurr has worked at GSFC 19 years. He graduated from the University of Maryland with a BS in Electrical engineering in 1984. Rick worked on JWST as the IC&DH study lead during formulation.

No biography for Daniel Gilley

No biography for Steve Parkes