SWIFU: SPACEWIRE INTERFACE UNIT FOR VERSATILE SENSOR INTEGRATION ON THE EXOMARS CHASSIS BREADBOARD

Session: Spacewire Missions and Applications

Short Paper

Lee C. G.-Y., Obstei R.

von Hoerner & Sulger GmbH, Schlossplatz 8, Schwetzingen, D-68723, Germany

E-mail: lee@vh-s.de, obstei@vh-s.de

ABSTRACT

This paper describes a versatile interface unit which will allow quick integration of a distributed sensor system into a Spacewire network. Each unit contains two Spacewire interfaces plus an RMAP decoder and functions both a simple router and a data interface: Spacewire packets are accepted by the unit and passed to the RMAP decoder only if the packets logical address matches the units node ID. Otherwise the packet is passed to the alternate interface for transmission to the next node in the chain. In this way the network is built up by daisy chaining units together.

Accepted packets are decoded using the RMAP protocol which access a generic IO bus. Various different IO peripherals may be attached to this bus as required by the system and thus the unit can act as both a gateway to other protocols such as RS232 and CAN as well as direct interface to sensors from which data is generated. As well as servicing RMAP requests, the unit may also be configured to initiate RMAP requests upon specific events – data arriving on a port or if a given Spacewire time code is received. In this way resources may be free from the task of polling all the SWIFU units for data.

The current implementation of the unit is target to mixed signal FPGA technology to allow a single chip sensor solution and is being used as an integral part of the Exomars Chassis Breadboard being built by von Hoerner & Sulger GmbH.

1 INTRODUCTION

1.1 MOTIVATION

Over the last 30 years von Hoerner & Sulger GmbH (vHS) have designed and built custom space instruments for a large number of space missions belonging to US, European and Russian space agencies. Examples of which are COSIMA on ESA Rosetta and CIDA on NASA's successful Stardust mission. Each agency and even each mission has required different TM/TC protocols and the company has had to been able support a variety of interface protocols from simple RS-232 style protocols to more sophisticated MIL-STD-1355 interfaces.

The use of Spacewire within spacecraft data system data architecture has now become more and more common and it increasingly required that instruments and subsystems interface to directly to the on-board data handling system through this medium. It is highly likely that projects that the company are currently involved in, such as BELA, the laser altimeter on ESA Bepi-Columbo and Exomars Chassis and Locomotion subsystem will require Spacewire interfaces to able to communicate and with this in mind it was decided to develop the company expertise in this area by initiating this internal development project.

1.2 CONCEPT

The aim was the development project was to raise the expertise in the Spacewire area and to provide a cost effective method to implement a prototype Spacewire compatible systems. The project has two design goals:

- Development of a RMAP request server module with versatile peripheral interface.
- Development of simplistic routing module to allow a daisy chain network topology to be created.

A single Spacewire Interface Unit (SWIFU) will contain both modules and all the necessary peripheral interface logic implemented on a single FPGA device.

2 NODE DESIGN

The functional block diagram of a node is shown in

Figure 1. Each node contains two space wire CODEC link interfaces. The two interfaces are connected together by routing logic that is also attached to the RMAP decoder. Each node also possesses an 8 bit unique identifier set in hardware which acts as a logical address for the Spacewire network. Requests to the RMAP decoder may access any of the IO modules via a peripheral address bus.

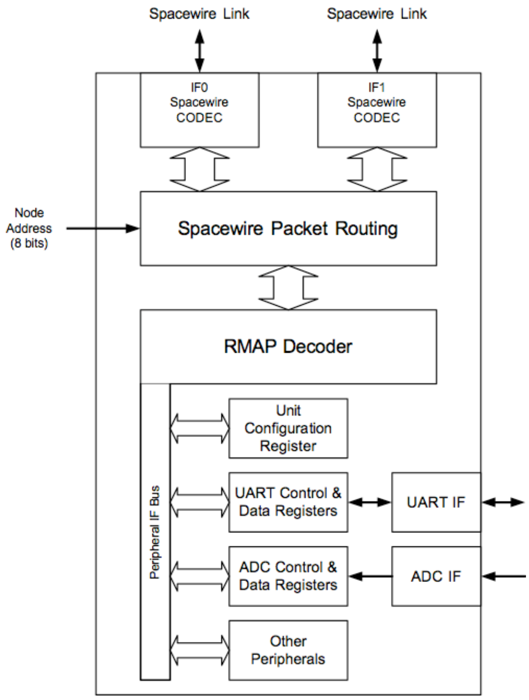

Figure 1 SWIFU block diagram

2.1 ROUTING LOGIC

When a packet arrives at a node, the node stores the first byte inside the receiving interface and checks where the packet should be routed to on the basis of the destination byte. If the destination address matches the node's identifier, the packet routing module directs the packet to the RMAP decoder if it is idle. If the RMAP decoder is busy with another request, Spacewire's flow control mechanism is used to hold the packet until the current request is completed.

If the destination address does not match the node's identifier then the packet will be transmitted through to the alternate Spacewire interface using a wormhole routing mechanism. In the special case that the alternate interface has not established an initialised link the packet is looped back on the receiving link.

The value of the received time codes are stored in the node and made available all parts of the node. Received time codes are also routed to the alternate interface as prescribed by the standard – i.e. codes which have time values equal to current time code value are suppressed.

2.2 RMAP REQUEST DECODER

As the RMAP decoder receives a packet, the header CRC is validated and the header is stored. If the packet contains a read request, the decoder uses the header information to retrieve the requested data into the data buffer, which is then used to form the data block in response to the request. Conversely for a write command, the packet's data block is stored in the data buffer and if no errors detected by the data CRC is written to the location defined in the header information. In either case, the response modifies the stored header information and a new CRC calculated for the header and data block if necessary. The response if then transmitted via the same interface as the request was received on.

2.3 PERIPHERAL INTERFACE

The RMAP decoder is connected to the node peripherals using a bus system following the Wishbone specification. In addition to the signals defined by the Wishbone specification, the bus includes discrete signals to allow each peripheral to indicate an occurrence of an event and to interpret information relating to the receipt and value of Spacewire time codes.

The peripheral address map is divided into 8 separate address spaces of 12 bits each. The first address space is reserved for configuration and status registers of the node. Other address spaces are then available for any peripheral module that may be required for a given the application. Each peripheral module must respect provide the following rules:

- The peripheral may utilise as little or as much of the address range 0x000 to 0x7FF.
- Address 0x800 to 0x803 Event Description registers (all other addresses are ignored) which must provide valid data if accessed.
- Access to an invalid address location is signalled with using the ERR signal in the Wishbone bus.

Apart from these rules the peripheral model may comprise of type of peripheral that can be implemented in programmable logic such as CAN interfaces, UART and lower level digital and analog input/output. In all cases the Spacewire network is transparent to the peripheral.

2.4 EVENT DATA TRANSFER

The design also allows the node to act as a client and to generate a RMAP write request on the occurrence of predefined event. The destination address for the request is predefined in the nodes configuration registers whilst the circumstances that initiate the event are defined with by the peripheral module logic itself. Typically an event could be generated upon the receipt of a specific Spacewire time or when the peripheral itself contains a certain amount of data.

When an event occurs, the peripheral signals the RMAP decoder that it has a pending event. The decoder then generates the RMAP message using information contained in the event configuration registers at locations 0x800-0x803 to collect the specified data from the peripheral.

3 NETWORK TOPOLOGY

By connecting several of these node units together a simple daisy chain network can be built. An operational network would consist not only of these slave nodes (SN) which receive and process normal RMAP requests but also a master node (MN) which generates the requests. The master node is also configured as the destination of RMAP packets generated by events at the SN's. Typically the MN would be the application's processing environment and the SN's are remote peripherals to which it communicates with. The advantage of using such a daisy chain system is that it is possible to build a space wire system without the need for a dedicated router. Once implemented it is also simple to expand the number of SN with out the need of expensive routing components.

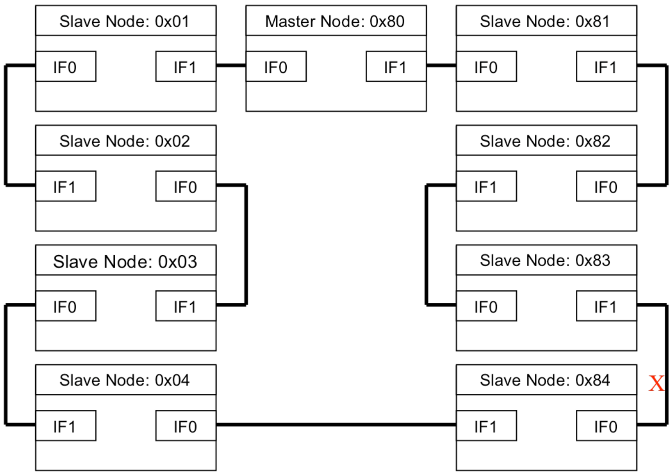

Figure 2 SWIFU Ring Network

A weakness of such a daisy-chained system is reliability. If a node or link fails then the any SN further down the chain from the MN is cut off. One method to improve reliability is to form a complete ring with the chain. A possible implementation is shown in Figure 2. Here the MN is designated with address 0x80. The SN addresses are set so that if the number of steps to the MN via interface 0 is less than interface 1, the most significant bit of the node's identifier is set. The master node is initialised so that it has a routing table where all packets destined to addresses whose MSB is set is sent to interface 1and otherwise to interface 0. If a link failure occurs at point X in the diagram, RMAP requests destined for Node 0x84 would be "looped back" at Node 0x83. Thus the MN will be able to detect the broken network and modify its routing for Node 0x84 to go via IF0. Such a routing table would require little resources in the MN as each entry can be represented with a single bit or two if unreachable addresses were also to be identified.

4 ROVER APPLICATION

The current SWIFU development will be used as part of the Phase B1 Exomars Chassis Breadboard by the Oerlikon-Space led consortium of which vHS is the major German partner. It is required that the electrical data system is implemented as a Spacewire network. The current design comprises of 6 steerable wheel drives each containing a single node providing an RMAP data interface to:

- COTS steering and wheel motor controller
- Steering position sensors
- Force and Torque sensors on the wheels
- General voltage, current and temperature monitoring

Additional nodes provide interfaces to system housekeeping, suspension position sensors and the IMU. The Master node is provided by the chassis's Onboard Computer.

Although this implementation is for the breadboard only, some advantages could be seen for a flight implementation. An over-riding requirement for the Exomars rover is mass which should be minimised. For a system critical function such as rover motion it would also be desirable to have fully redundant space wire links to the motor controllers so that a single electrical harness failure would not cause the rover to be immobile. However if these controllers were located on the suspension members the amount of harness required for a standard star network would be excessive. The use of a daisy chain network such as this would provide some fault tolerance to such failures whilst economising on mass.

5 FUTURE

The development of the node unit is nearing completion and hardware will soon be fabricated for use on the Exomars breadboard.

It also is hoped that SWIFU will be used on other future projects and in other fields. It is also being considered to extend its features to allow small ring networks to be incorporated into networks which utilise the path addressing mechanism.

6 ACKNOWLEDGEMENTS

The Exomars Breadboard Consortium is comprises of Oerlikon Space (lead) von Hoerner & Sulger GmbH (Germany) , Bluebotics SA (Switzerland), DLR-RM Oberpfaffenhofen (Germany), and ETHZ (Swiss Federal Institute of Technology Zurich).