SYSTEM ASPECTS OF SPACEWIRE NETWORKS

Session: SpaceWire missions and applications

Short Paper

Paul Rastetter, Dr. Stephan Fischer, Uwe Liebstückel, Richard Wiest

Astrium GmbH, 81663 Munich, Germany

E-mail: paul.rastetter@astrium.eads.net, stephan.fischer@astrium.eads.net, uwe.liebstueckel@astrium.eads.net, richard.wiest@astrium.eads.net

1 ABSTRACT

The well established SpaceWire standard provides a point-to-point communication with hundreds of Mega-bit per second data transfer in full-duplex operation. To efficiently use SpaceWire devices in an on-board system many aspects have to be taken into account already early in the system definition phase. Among others the following points have to be considered: provided data rates, supported protocols, provided interface to configure and control the SpaceWire devices, interaction with software, compatibility of equipments from different suppliers assuming master/slave concepts, etc..

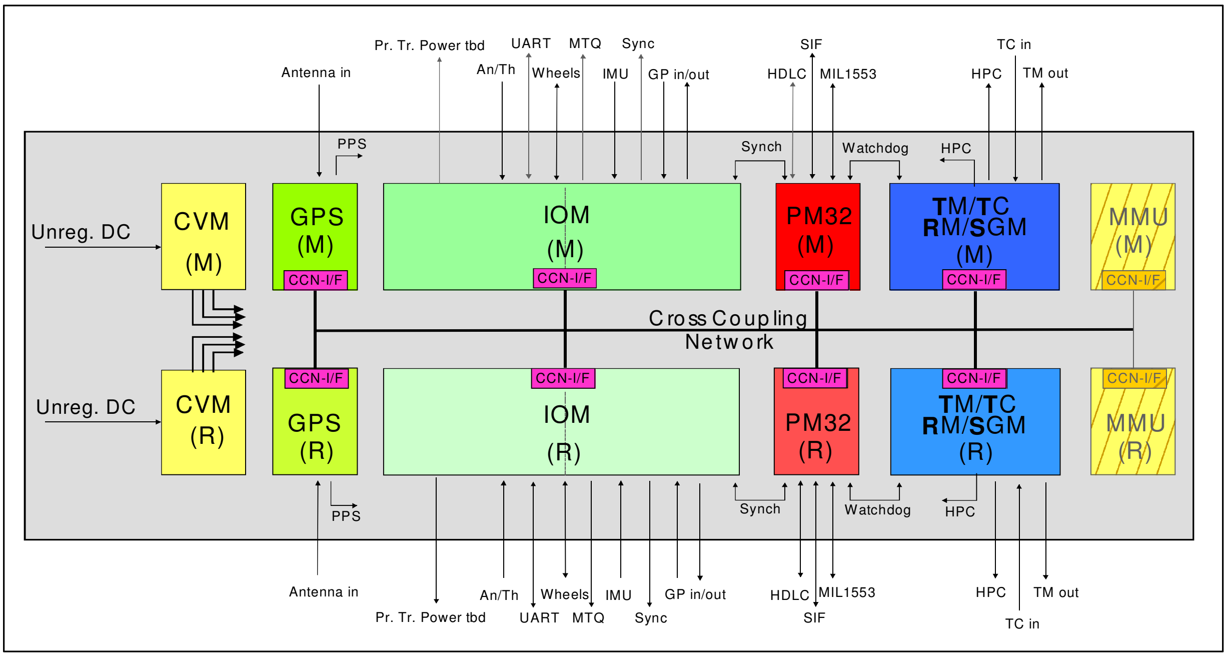
The paper will give an overview about some system concepts and their implementation using e.g. the SMCS332SpW, the SMCS116SpW and special FPGA implementations including a SpaceWire interface in a typical topology.

2 SYSTEM ASPECTS

In the definition phase of a SpaceWire based system several aspects have to be considered. One aspect is the proposed or required net work topology. Another is the selection of available devices to implement the required functions and interfaces.

The topology shown in the diagram, called "Cross Coupling Network", is a star topology. Star topology means that one module is connected directly to all other modules. In the diagram the module PM32 is the one which is connected to all other modules. The other modules like IOM, TM/TC or GPS have one main SpW link and one redundant SpW link. In a star topology each SpW link can operate on a different transmit rate depending on the necessarity of the connected module.

The aspects of interfacing between software and SpaceWire device, hardware-software interacting, protocol support and remote controlled SpaceWire devices will be discussed in the following chapters.



2.1 Interfacing between software and SpaceWire-device (on "intelligent" modules)

SpaceWire devices are often used to transfer data between modules. The data interfacing can be done in several ways which has different impacts on software. One way is to use a simple FIFO-like interface and another way is to implement DMA (Direct Memory Access). The FIFO-like interfaces require less or even no external hardware but put the load of transfer to software. Each byte/word (depending on interface width) has to be read/written by a software task, which limits the throughput.

Another way is to implement DMA capability into the SpW-device, but DMA requires memory. The memory can be on the board or inside the SpW-device itself. The data transfer is done in the following way. The processor writes the complete transmit SpW packet into the memory. After that the processor sets the transmit pointers inside the SpW device. From now on the device takes control of transmitting the SpW packet without any further interaction from the processor. The maximum length of one SpW packet depends on the implementation, e.g. length of pointers or additional control bits.

The receipt of a SpW packet with DMA works similar. The processor sets the receive pointers which define a receive area in the memory and it is informed by interrupt when a packet is received. The length of a receive packet is defined by the sender.

DMA itself is more complex in hardware, but offers high throughput and simple interface to software. SMCS332SpW and SMCS116SpW (in master mode) provide DMA capability.

2.2 Hardware-Software interacting

The interacting between hardware and software is important for the system and is done via interrupts and/or status bits. Therefore the interrupt handling is a major task in software. To simplify the life for SW, a SpW device should support several interrupt/events/status bits. The interrupts should be maskable to allow a selection of bits of interest at the current time.

In the case of a FIFO-like interface the interrupts could be e.g. link connected, link error, byte, word transmitted/received. Using DMA the interrupts could be e.g. link connected, link error, packet transmitted, EOP/EEP received, receive area full. The advantage of the DMA is the fact that the interrupts can be issued on packet level instead of byte/word level. This reduces the interrupt frequency in comparison with a FIFO-like interface.

SMCS332SpW and SMCS116SpW provide all necessary interrupts to signal the events to software at packet level.

2.3 Protocol support

There are several protocols available on top of SpW. All protocols are characterized by a header part and a data part. The length of the header part may be not always aligned with 32 bits, which is mostly used by processors. Therefore an easy way to assemble the header part and data part to a SpW packet at the transmitting side of a SpW-link is necessary in DMA mode.

The SMCS332SpW supports the protocol packet generation by the header field bit. The bytes to be transmitted via SpW are always read in 32 bit mode. If the header field bit is set, the value of the first byte which is read from memory is used as a counter value. This value defines the amount of header bytes to be sent. The counter byte itself is not transmitted via SpW. This mechanism allows the software to generate headers with a length from 1 to 15 bytes. The complete length of a SpW transmit packet is defined by the transmit pointers.

The same mechanism could also be used if path address bytes have to be put in front of a SpW packet.

2.4 Remote controlled SpaceWire devices (without local processor)

A remote controlled SpW device is characterized by containing a SpW protocol and several functions/interfaces like event counters, timers, RAM- or FIFO-interfaces etc. which are controlled and configured via a defined protocol. The protocol is defined by the remote SpW device itself. It can be an open/general protocol like RMAP or STUP or a propriety one.

3 Summary

As discussed in the paper the performance of a system depends on several issues. There are several SpW devices available, like SpW-Router, RTC (Remote Terminal Controller), SMCS332SpW and SMCS116SpW, which can be used to design and implement a SpW-based system.

If there are specific requirements which can not be fulfilled by these existing devices, the SpW-Codec from ESA can be included in a FPGA together with the specific requirements.