UNIVERSAL SPACEWIRE INTERFACE TO/FROM VME AND TO/FROM PCI

Session: Poster Session

Short Paper

ir. G.J. Vollmuller, ing. A. Pleijsier

National Aerospace Laboratory NLR

Anthony Fokkerweg 2, 1059CM, Amsterdam / Voorsterweg 31, 8316PR, Marknesse

The Netherlands

E-mail: vollmull@nlr.nl, pleysier@nlr.nl

ABSTRACT

NLR developed in co-ordination with Satellite Services (Katwijk, Netherlands) an universal SpaceWire interface that is fully ready for production. This SpaceWire interface module is a PCI Mezzanine Format (PMC) and accommodates 3 DS SpaceWire links (Data Strobe encoding). It is connected to the PCI local bus of the PowerPC board by a high-performance 132 MB/s PCI interface.

Special features of this SpaceWire interface are:

- transmit speed up to 200 Mbit/s
- time-tagging for both incoming and outgoing packets
- wormhole routing,
- segmenting of large data structures
- priority settings for each channel
- and channel routing

This interface focuses on usage in EGSEs and SCOEs. It can be used in a PC environment (PCI bus) aswell as in an embedded PC environment (VME / compact PCI), have maximum performance and is flexible in use.

1 BLOCK DIAGRAM AND FEATURES

The block diagram of the SpaceWire PMC module is depicted in figure 1.

The central part of the interface module is the Xilinx Virtex4 FPGA that contains the ESA SpaceWire cores and the VHDL, developed by NLR, to incorporate the rest of the functionality.

The module has three SpaceWire DS links routed via FIN1102 LVDS repeaters. The Dual ported RAM is of size 512k x 36 and the module has one PCI accelerator PLX9056. The Pulse Per Second interface can be provided via a LVTTL/RS422 receiver. Power is extracted from the carrier board (3.3 Volt); the local voltages are generated on the board itself (including the start-up controller). The FRAM memory can be used for storing settings and parameters.

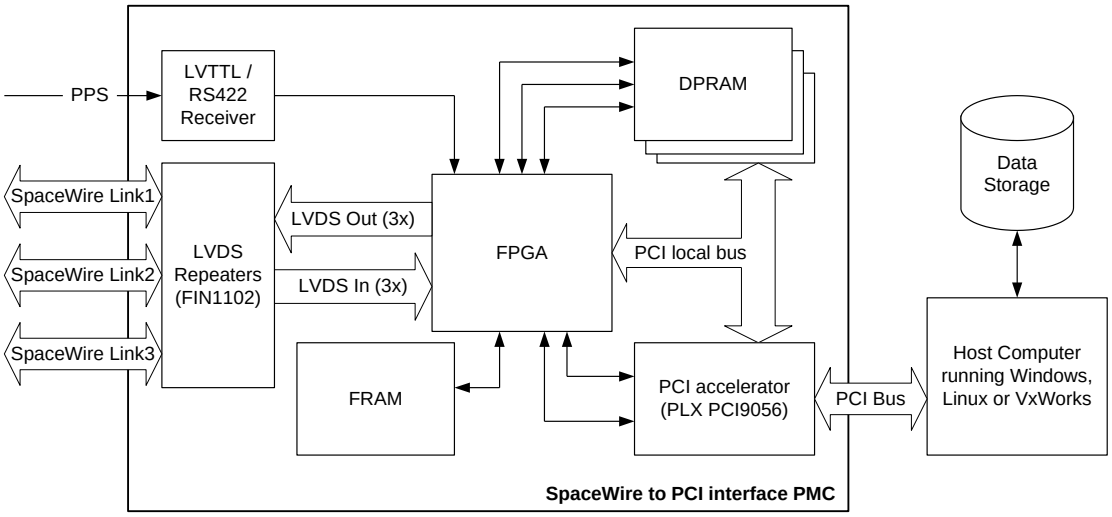

Figure 1: Block diagram of the SpaceWire to PCI interface module

The main features of this SpaceWire interface PMC module are:

- Provides three bi-directional asynchronous SpaceWire DS links via 3 ITT Cannon D-miniature connectors
- Maximum data rate of 200 Mbits/s sustained throughput on three links (measured: 260 Mbits/s)
- Provides 33/66 MHz 32 bits PCI interface
- PCI accelerator PXI9056 to facilitate PCI burst read and write cycle between Dual Ported RAM and host memory
- On-board Dual Ported RAM (512k x 36) as FIFO for temporary storage of data-to-be-send or data-received
- Pulse Per Second synchronisation (LVTTL or RS422)
- CUC timer (CCSDS Unsegmented Time Code) for time stamp of received and sent data packets
- General purpose FRAM memory (among others for storing settings)
- Temperature sensor, JTAG interface, Reset button on front panel and internal loop-back functionality for test purposes.

Note that for verification, two of these SpaceWire PMC modules are placed on a commercial Motorola PowerPC VME board; providing 6 SpaceWire links in total.

2 FPGA ARCHITECTURE

The FPGA architecture is depicted in figure 2.

The module allows communication between three SpaceWire link interfaces and a host processor. Data is exchanged between host memory and PMC using hardware initiated DMA. Data format exchange is based on so-called segments (enclosed packets or packet parts). The module allows a sustained throughput of 200 Mbits/s over three links.

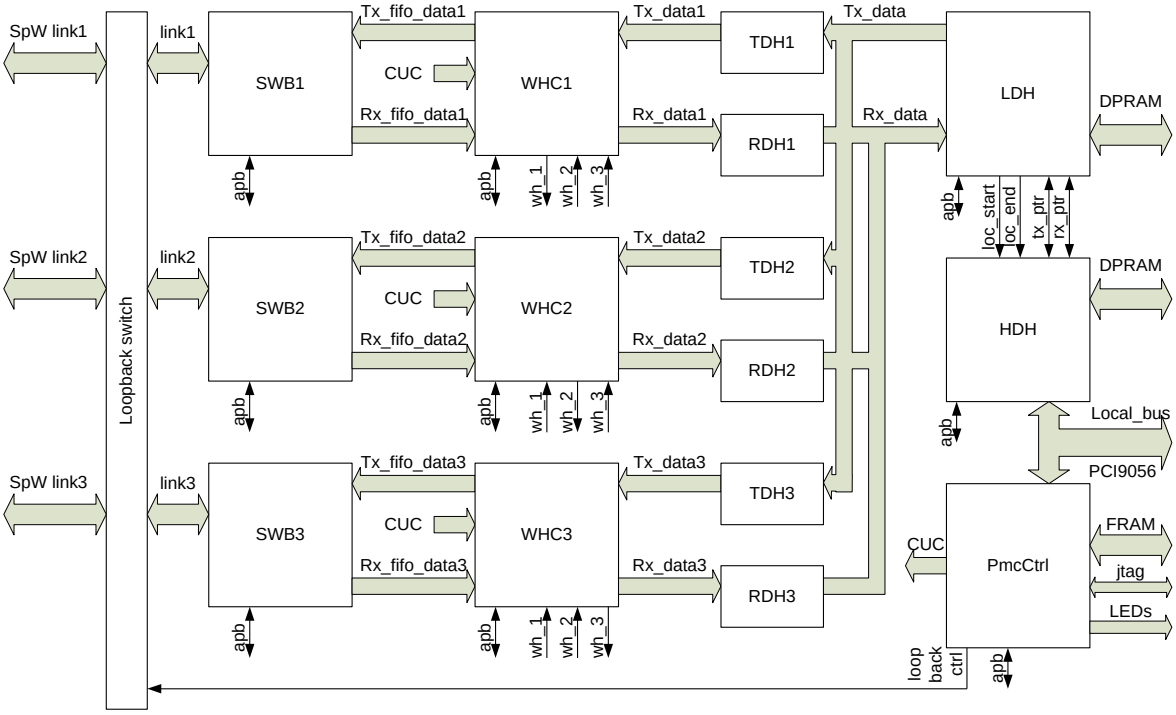

Figure 2: Block diagram of the FPGA

The Loopback switch, on the left side, offers the possibility to (re)route SpaceWire output data of a specific link to the input part of another link (or directly to its own input part). This feature can be used for PMC testing.

The SpaceWire Block (SWB) contains the ESA IP core. It allows SpaceWire packet transmission according to the SpaceWire specification ECSS-E-50-12A see [1]. The IP core also contains an AHB interface for communication to the host and uses the APB interface for internal register I/O. For the SpW PMC module the AHB (high performance bus) interface is not used for host I/O communication, but instead the FIFO interface is used to interface with the other VHDL components.

The Wormhole Controller (WHC) controls the transmitted data to a link (output packet if CUC time arrived) and controls the received data from a link (generate CUC time in segment header). It performs the route check (destination can be the host, or another link or both) and this component adds CUC times to the segment header. When the packet from the link is larger than the maximum segment size (defined by the host) the packet will be split up into segments of max_segment size. Per link interface a CUC time is maintained, based on the external CUC clock, the PPS input and host read/write CUC time capabilities. Output of a SpW packet is delayed until the requested CUC time (in the segment header) has come.

Segments to transmit are stored into DPRAM by the **Host Data Handler (HDH)**, and retrieved from DPRAM by the Local Data Handler (LDH) into the transmit FIFO of the Transmit Data Handler (TDH).

The Transmit Data Handler will read the FIFO and strip the segment header from the segment data (which is the SpaceWire packet). The output packet data (converted to 9 bits wide) is offered to the WHC, preceded by the packet status and CUC time bytes

The Receive Data Handler (RDH) controls the reception of packets. It stores the received data until the LDH is able to read this link interface. Since three channels can offer data to be stored in the DPRAM, the LDH decides (arbitrates) which channel is permitted to store data in DPRAM.

The Local Data Handler (LDH) interfaces between the link interfaces (RDH/TDH) and the DPRAM. The LDH is responsible for correct segment storage into DPRAM when SpaceWire packets from a link are received. Segment data from host to a link is retrieved from DPRAM and send to the TDH. Due to each link interface acting concurrently, the LDH arbitrates which link interface is able to send or receive DPRAM segment words.

The Host Data Handler (HDH) interfaces between PCI accelerator (PCI9056) and DPRAM in case of data exchange, and interfaces between PCI accelerator and PMC registers in case of PMC control (register read/write actions).

The PMC control (PMCCtrl) module is used for general PMC control, like controlling the Non link specific registers that are allocated in this module and handling the PMC interrupt sources. Also the CUC clock is generated in this module, using the external 33.554432 clock input signal. It also contains the FRAM I/O control.

3 API

Next to the hardware (and VHDL) development, the NLR has written the Application Programming Interface (API) for VxWorks for this SpaceWire interface module. The API contains the function calls to the PMC module. No direct (register) access to the SpW_PMC module is foreseen.

The following function calls are available:

- SpWCardOpen/Reset/Close/Status
- SpWNodeOpen/Close/Control/Status
- SpWNodeSetTime/GetTime
- SpWLinkOpen/Close/Control
- SpWLinkStart/Stop
- SpWLinkReadPacket/WritePacket/WriteStop/Status
- SpWCucControl/CucReadTime (per link)
- SpWPpsControl/LedControl/JtagStatus
- SpWRegisterRead/RegisterWrite (Enables PMC register reading/writing)
- SpWFramRead/FramWrite
- SpWDpramRead/Write
- SpWReadSegment/WriteSegment

4 REFERENCES

1. ESA-ESTEC ECSS-E-50-12A, “SpaceWire - Links, nodes, routers and networks”, Jan-2003.

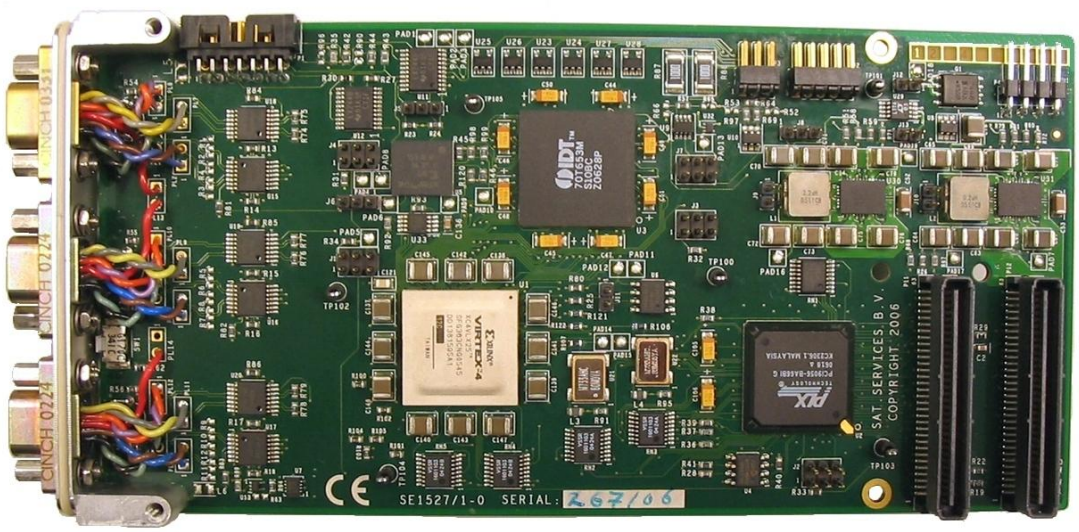

Figure 3: Picture of the SpaceWire PMC module