

A VERSATILE SPACEWIRE CODEC VHDL IP CORE

Session: SpaceWire Components (Poster)

Short Paper

Marko Isomäki, Sandi Habinc, Jiri Gaisler

Aeroflex Gaisler, Kungsgatan 12, SE-411 19 Göteborg, Sweden

marko@gaisler.com, sandi@gaisler.com, jiri@gaisler.com

ABSTRACT

Aeroflex Gaisler is specialized in developing IP cores and System-on-chip solutions based on the AMBA [1] on-chip bus. A library of IP cores called GRLIB [2] is provided where two SpaceWire IP cores, the GRSPW and GRSPW2, are available for AMBA bus based systems providing a master interface, advanced DMA and a RMAP [3] target.

There is a need for a more basic SpaceWire core with a simpler host interface and this has been addressed with the new SpaceWire Codec IP core. The core has the same SpaceWire link interface as the GRSPW2 but a simplified FIFO based host interface.

This paper presents the requirements driving the design of the core and features of the final core.

INTRODUCTION

Aeroflex Gaisler has been providing SpaceWire interfaces intended for AMBA on-chip bus based systems for several years. Two IP-cores have been available, the GRSPW and GRSPW2, which contain advanced functionality such as DMA channels and an RMAP target. It has become evident from several projects that this functionality is not needed in many cases and the inherent area overhead narrows the range of suitable technology alternatives in these cases.

For example, there might be only one hardware unit requiring SpaceWire in a system and it does not have an AMBA AHB on-chip bus. In that case it would more efficient to let it access data directly to/from the transmitter and receiver FIFOs for example, when implementing router functions [4] or additional SpaceWire protocols such as the CCSDS packet transfer protocol [7].

To provide a core suitable for these applications the development of a stand alone SpaceWire codec IP core was started.

This paper will first go through the basic properties of the core followed by a deeper analysis of the external SpaceWire interface, the dual ports and finally make a comparison with existing SpaceWire codecs.

BASIC PROPERTIES

Most parts of the SpaceWire codec are identical to the encoder-decoder used in the GRSPW2 core. This has the benefit of large parts of the core already being verified, validated and proven in hardware. The difference is in the host interface. Where the

GRSPW2 has an AMBA based DMA engine and RMAP target processing the incoming data the SpaceWire codec instead has FIFOs in both the transmit and receive directions containing the SpW characters including the control bit to be able to differentiate between EOP/EEP and normal characters [2]. This makes for a significant area reduction and simplification of the host interface. Since the user has full-control of the order and timing of the characters it gives more flexibility on packet generation compared to a bus based DMA solution where the implementation of the SpaceWire core restricts what can be accomplished.

The transmit host interface contains a data signal, write signal, FIFO count and a FIFO full indication. Correspondingly the receive interface contains a data signal, read signal, FIFO count and FIFO empty indications. This is shown in figure 1. This is a simple interface but still powerful enough to enable the implementation of most applications. Since the FIFO contents are transmitted and received in order without any additional processing a lot of freedom is left to the entity controlling the host interface.

The core also provides a time-code interface which basically transmits what is presented on the time input when the tickin input is asserted. For received time-codes tickout is asserted and the received time-code is presented on the time-output. No time-code validity checking is done (except for parity). This is left up to the user to implement externally if needed.

Separate control signals are also provided for the frequency during initialization and run-state. The former also controls the time-out periods used in the link interface FSM.

As for all other cores in the GRLIB IP library the core is written in a technology independent way with wrappers for the instantiation of technology dependent modules if needed by the target technology.

The presented features in total gives a fully compliant codec implementation. The following sections will show new features which make the core more versatile.

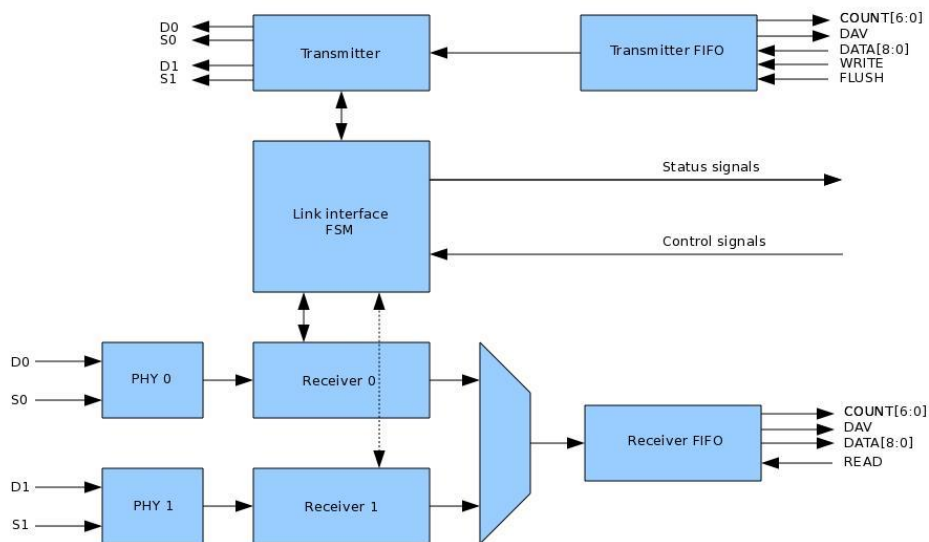


Figure 1: SpaceWire Codec block diagram

EXTERNAL SPACEWIRE INTERFACE

The signal interface to the SpaceWire network consists of a set of data and data-valid signals. These signals should be connected to a GRSPW2_PHY IP core also provided by Aeroflex Gaisler which handles the actual low-level clock and signal recovery or connecting to an external PHY component. This core is also used by the GRSPW2.

The benefit of having this in a separate core is that the data is always transferred in the same manner at the core interface not requiring any changes in the main core if the low-level implementation needs to be changed.

Another important issue to consider is the application of constraints during synthesis and place and route. Names of instances often change between versions of tools and the deeper in the hierarchy the instance is the more often the constraints need to be changed. By keeping the parts with a high implementation complexity in a single entity at the top-level of the design makes it easier to apply constraints.

On the receiver side the SpaceWire codec supports the standard self-clocked version with clock-recovery using an XOR gate. In addition to this there are three alternatives: single data rate (SDR) sampling, double data rate (DDR) sampling and an interface to the Aeroflex SpW transceiver [6]. The latter is a standard part provided by Aeroflex Colorado Springs which enables the codec to run at a reduced clock frequency compared to the SpaceWire link.

The sampling versions sample the incoming strobe and data signals and the sampling frequency must be at least 1.5 times higher than the maximum bit rate on the SpaceWire link. In DDR mode the clock frequency can be reduced compared to the

SDR version with the same sampling rate. The benefit of the sampling version is the much simpler constraint requirement. Only a single clock frequency constraint is needed to get a working implementation.

On the transmitter side SDR, DDR and Aeroflex SpW transceiver interfaces are supported. The DDR interface offers a reduced clock frequency compared to the SDR at the same bitrate but is not available for all technologies.

DUAL PORTS

The SpaceWire codec also provides an optional second SpaceWire port. The core can automatically switch between the ports depending on link activity or it can be forced using a specific link using a signal. The use of dual ports requires an additional GRSPW2_PHY IP core and internal receiver logic giving a small area penalty. For the Aeroflex SpW transceiver it would also require a second transceiver chip externally.

CONCLUSION

The SpaceWire codec presented is a technology independent and versatile core. It provides a fully standard compliant codec, and in addition to this several features such as dual ports and support for several different SpaceWire physical layers. The latter two features are not found in most other encoder-decoder implementations and make the core an area efficient alternative for redundancy applications, and easily movable between technologies.

REFERENCES

1. AMBA specification, Rev 2.0, ARM IHI 0011A, Issue A, ARM Limited, www.arm.com
2. GRLIB IP Core User Manual, Aeroflex Gaisler, www.gaisler.com
3. Remote Memory Access Protocol, ECSS-E-ST-50-52C
4. Marko Isomäki et al., “A Configurable SpaceWire Router VHDL IP Core”, ISC 2010
5. SpaceWire – Links, Nodes, Routers and Networks, ECSS-E-ST-50-12C
6. UT200SpWPHY01 SpaceWire Physical layer transceiver, Aeroflex Colorado Springs, www.aeroflex.com
7. SpaceWire - CCSDS packet transfer protocol, ECSS-E-ST-50-53C