

FPGA-based Networking Systems for High Data-rate and Reliable In-vehicle Communications

**Sergio Saponara, ^AEsa Petri, ^AMarco Tonarelli, *Iacopo Del Corona, *Luca Fanucci*

**Dept. Information Engineering, University of Pisa, Via Caruso 16, 56122, Pisa, Italy*

^AConsorzio Pisa Ricerche – Divisione Applicazioni Microelettroniche (CPR-TEAM), C.so Italia 116, 56125, Pisa, Italy

Abstract

The amount of electronic systems introduced in vehicles is continuously increasing: X-by-wire, complex electronic control systems and above all future applications such as automotive vision and safety warnings require in-car reliable communication backbones with the capability to handle large amount of data at high speeds. To cope with this issue and driven by the experience of aerospace systems, the SpaceWire standard, recently proposed by the European Space Agency (ESA), can be introduced in the automotive field. The SpaceWire is a serial data link standard which provides safety and redundancy and guarantees to handle data-rates up to hundreds of Mbps. This paper presents the design of configurable SpaceWire router and interface hardware macrocells, the first in state of the art compliant with the newest standard extensions, Protocol Identification (PID) and Remote Memory Access Protocol (RMAP). The macrocells have been integrated and tested on antifuse technology in the framework of an ESA project. The achieved performances of a router with 8 links, 130 Mbps data-rate, 1.5 W power cost, meet the requirements of future automotive electronic systems. The proposed networking solution simplifies the connectivity, reducing also the relevant volume and mass budgets, provides network safety and redundancy and guarantees to handle very high bandwidth data flows not covered by current standards as CAN or FlexRay.

1. Introduction

Automotive vehicles are nowadays complex distributed computing systems with various and evolving demands on networking capabilities [1-3]. Emerging applications such as automotive vision subsystems for intelligent driver assistance and safety warnings [4,5] require in-vehicle high data-rate and reliable communication backbones. Different videos acquired by imaging sensor devices distributed through the vehicle have to be transferred, safely and with a bandwidth need of tens of Mbps, to a central digital processing unit in charge of data fusion and analysis for different purposes: collision avoidance, obstacle detection, improving vision during night or in case of bad lighting conditions, parking aid, etc. State-of-the-art standards for in-vehicle communication do not fulfill the requirement of a reliable communication protocol for high-bandwidth subsystems. According to the

Society for Automotive Engineers (SAE), communication protocols can be grouped in 4 classes based on data transmission speeds [1]. Class A networks, e.g. LIN, have a data rate lower than 20 kbps and are used to transmit simple control data with low-cost technology. Class B networks, e.g. low-speed CAN, support data exchanges between Electronic Control Units (ECUs) in order to reduce the number of sensors by sharing information. They operate up to 125 kbps. Class C networks, such as TTP/C and high-speed CAN, are used for the powertrain and currently for the chassis domains up to 1 Mbps. A Class D is not formally defined, but it is generally considered that networks over 1 Mbps belong to class D. In this class we can find fault tolerant protocols e.g. the emerging FlexRay. However, FlexRay targets X-by-wire applications and its bandwidth, in the order of Mbps, is not enough for complex vision systems. High data-rate automotive networks exist, as the media-oriented system transport (MOST) supporting up to roughly 25 Mbps, or the IDB-1394 (automotive FireWire) up to 100 Mbps, but they are missing specifications such as fault-tolerance and reliability. Indeed they are devoted to transport of large amount of data for infotainment applications and not for safety or driver assistance. A similar scenario characterizes aerospace systems where the requirement of high rate and reliable protocol is further increased by the use of high bandwidth instruments for aerial surveillance or space scientific missions such as radars, image cameras, X-ray detectors. For the space scenario beside fault tolerance also radiation tolerance is a main issue.

1.1. Paper Outline

To cope with the above issue in both the automotive and aerospace scenarios, in this paper we propose the design of new intellectual property (IP) macrocells, implementing the emerging standard ECSS-E-50-12A known as SpaceWire (SpW) [6]. The SpW standard can facilitate the set up of in-vehicle high-speed and reliable networks, reducing system integration costs by the re-use of digital interfaces across different applications: see Fig. 1 for comparison with other in-vehicle bus. In its basic version SpW covers six communication protocol levels: physical, signal, character, exchange, packet and network levels. Currently two extensions of the standard, the RMAP [7] and the PID [8], are under development covering the transport layer protocol. Several space missions in Europe,

US, Japan and Korea have planned the use of the SpW standard [9-13] and it has been also adopted in avionic applications [14]. To simplify the development of complex SpW-based communication systems, it is of paramount importance the availability of reusable and low-complexity IP macrocells for its building blocks: routers and interfaces. The state of the art [9-14] is characterized by the implementation of SpW interfaces or routers compliant only with the basic standard version. No SpaceWire router IP core is at present already available supporting the latest standard extensions, PID and RMAP, covering the transport layer. In this paper we present the design of a complete SpW routing and interfacing solution, compliant with RMAP and PID, achieved through novel hardware macrocells targeting FPGA (Field Programmable Gate Array) devices. Different technologies, anti-fuse or SRAM based, radiation-tolerant or not, have been considered for the different application fields: automotive, avionics, space. FPGA devices have been preferred to the ASIC (Applied Specific Integrated Circuit) approach for these reasons:

- As discussed in Sections 5 and 6, today FPGAs provide enough logic and memory resources and performance (timing, power, operating point) to meet the challenging requirements of aerospace or automotive systems.
- The FPGA non recurrent cost and development/time-to-market time, much lower than the ASIC ones, are more suitable for a scenario where the standard is not frozen [6-8]. Moreover a large market is not foreseen for space and avionic fields and also the diffusion of automotive vision systems is still in a low-volume prototyping phase.

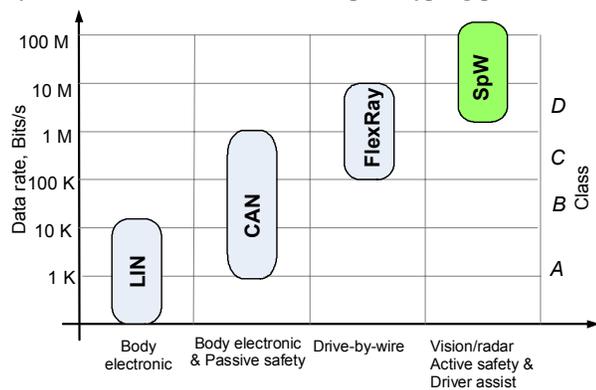


Figure 1: Bus comparison for in-vehicle networking

Hereafter, Section 2 briefly reviews the SpW standard with reference also to high data-rate serial links as IEEE-1355 and IEEE-1394. Section 3 details the architectures of router and interfaces. The validation flow is described in Section 4. Implementation results in different FPGA technologies and comparisons to state-of-art SpW devices are shown in Section 5. Conclusions are in Section 6.

2. SpaceWire Networking

2.1. Basic Standard

SpW is a bi-directional, full-duplex, point-to-point, high-speed, serial data communication link. It was born from the IEEE-1355 terrestrial standard [16] and is based on Low Voltage Differential Signaling (LVDS) physical layer, resulting in a low-power high-speed link. The Standard is defined at six different levels of protocol [6]:

- **Physical level**, defining the cables and connectors.
- **Signal level**, covering signal voltage levels, noise margins and signal encoding. The SpW signal level is similar to other high-rate serial standards such as IEEE-1394. It also has data and strobe signals carried by twisted pair wires and with LVDS bus drivers. Unlike IEEE-1394, SpW specifies only a minimum bit-rate of 2 Mbps and the twisted pairs do not carry bidirectional signals. To achieve full duplex, 4 twisted pairs are needed for each link. At 100 Mbps, SpW cables up to 10 m are envisaged.
- **Character level**, defining the character format. Two types of characters are supported: 10-bit data characters and 4-bit control ones, the latter used for bus management and control. A data character contains a parity bit, a data control flag set to 0 and an 8-bit data word. A control character contains a parity bit, a data control flag set to 1, and two bits identifying a normal End Of Packet (EOP), an Error End of Packet (EEP), a Flow Control Token (FCT) or an escape (ESC). Special control codes are the 8-bit NULL, obtained by an ESC followed by FCT, and the time-codes, obtained by an ESC followed by a data character. Time-codes, which are new in SpW vs. IEEE-1355, permit to support the distribution of system time. A SpW network is event-driven, e.g. like CAN, but through time-codes distribution time-triggered communications can be set up, e.g. as in TTCAN, where a single node acts as time master while the others are time slaves.
- **Exchange level**, defining the protocol for initialization of connection, flow control, errors detection and link error recovery. Particularly FCT and NULL characters are exchanged for connection initialization between SpW nodes. The FCT is also used for flow control, by sending it, a requesting node indicates to the destination node that it has space in its receiving buffer for 8 more data characters. The exchange of NULL is used to maintain the connection between the two nodes when there are no other characters to transmit.

- **Packet level**, following the one defined in IEEE-1355, establishes how data are encapsulated in packet to transfer from source to destination. A packet comprises a destination address, a cargo and an end of packet marker. The destination address contains, depending on the used packet-addressing scheme, either the identity of the destination node or the path that the packet will take to reach the destination node. The cargo comprises one or more data characters to be transferred. The end of packet marker is a control character, EOP or EEP, indicating the

end of packet and that the next received data character will be a destination identifier.

- **Network level**, describing the structure of a SpW network, the routing techniques and the addressing schemes. It also defines network level errors and the correspondent error recovery protocols.

A SpW network is made up of links, nodes and routers. Nodes can be either directly connected by links or via routers to form a network. SpW routing switches are dynamic since they can change the routing frequently, usually on a packet by packet basis. A SpW router is based on wormhole routing in order to reduce (i) the size of the required memory buffer for transmitting/receiving SpW interfaces and (ii) the latency of the communication. Each packet contains a header holding the address of the destination node. When a packet is received, the router determines the output port to send the packet out of. If the requested output port is free, the whole packet is immediately routed to that port without waiting the reception of the entire packet, otherwise the incoming packet is halted until the output port becomes free. The SpW router implements the following packet-addressing schemes: Path, Logical and Regional addressing. In Path addressing the destination address is specified as a sequence of valid output port numbers used to drive the packet across the network. At each routing switch passed through by the packet, a port identifier is stripped off (header deletion) in the destination address, so that the subsequent switch takes the routing decision upon the following port identifier. In Logical addressing each destination node has a unique logical address and each network router has a routing table binding each logical address with a physical output port. Regional logical addressing is based on logical addressing in conjunction with header deletion, allowing packet transfer across an arbitrary sized network. In regional logical addressing the routing table also holds information about the headers to delete or to keep, for each logical address. To guarantee path redundancy and tolerance to link failure, Group Adaptive Routing (GAR) is supported. This is a means of routing packets to a requested destination over different paths through a network. Two or more output ports can be grouped together so that, if a link in the group is busy or unavailable due to failure, a packet is sent through the next available link in that group. The GAR feature, together with the support of dynamic routing topology, leads to an increased robustness of SpW vs. other serial protocols, such as IEEE-1394, where any branch node failure will partition the bus.

2.2. RMAP and PID extensions

To define and standardize higher level protocol layers and to increase the hardware and software compatibility among SpW systems, the PID has been introduced in 2005 [8]. The PID scheme enables many different protocols to operate concurrently over a SpW network without them

interfering with each other. Currently the standard is under extension with the introduction of RMAP protocol [7] to which PID=1 has been assigned. This high level protocol specifies procedures and command/answer message formats (verified or not, acknowledged or not, CRC type, error types and relevant answers) which allow a SpW network to be configured, SpW nodes to be controlled and data and status information from those units to be gathered. RMAP may operate alongside other custom protocols (with PID≠1) running over SpW. The access procedures from a requesting node to a destination one can be a read, a write or a read-modify-write. For each procedure two different CRC checks are performed, on the command (mandatory) and on the data (option verified/not verified) fields of the RMAP message. Moreover, for each access procedure an RMAP reply message, containing the resulting status of the access, can be optionally generated by the destination node. As an example, Fig. 2 shows the write command sequence. A similar structure characterizes read command messages where the data field is empty since data are contained in the reply message from the destination to the source. The size of an RMAP message is variable since the sizes of transmitted data, source address and destination address are variables. In the reply message a status field 0 indicates that all was OK, otherwise different codes can be used to signal different types of errors: data length error, buffer overrun, not supported or not defined command, authorization failure, invalid CRC, late or early EOP or EEP.

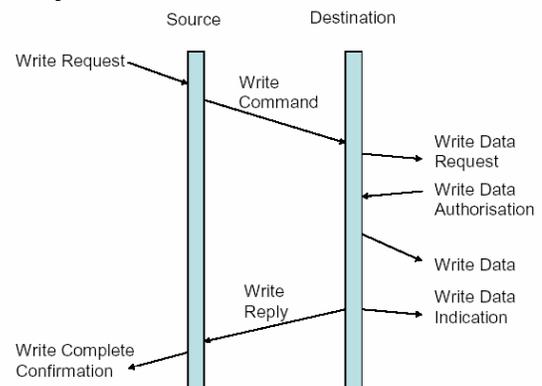


Figure 2: RMAP write command/acknowledge sequence

3. Architectural Design

The SpW router IP in Fig. 3 is a technology-independent VHDL macrocell providing a complete SpW routing and interfacing solution compliant with the basic standard plus RMAP and PID. The router IP features a parametrizable number of SpW interfaces (SpW itf) plus relevant switching matrix, a programmable router table, a dedicated decoder for RMAP, a time-code interface, a control/status interface and a data interface, all accessible by an external

host through a 32-bit AHB bus wrapper. All building blocks are detailed hereafter.

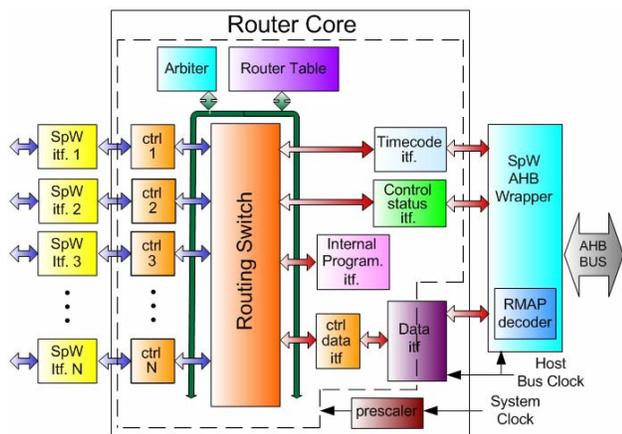


Figure 3: Architecture of the SpaceWire router macrocell

3.1. SpW Router Core

The SpW router core is a switching matrix connected to the external world via N SpW interfaces, 1 time-code interface, 1 control/status interface and 1 host data transfer interface. The VHDL model is parametric in terms of number of ports N, address intervals and size of receiving/transmitting buffers. The control/status interface provides a host unit read (for status check) and write (for configuration) access to internal registers of the SpW router and interfaces. The time-code interface distributes system time over a SpW network. Only one node in the network manages the distribution of time, providing the master time reference for the entire network, while the other nodes are time-slaves. Our SpW router can be configured to be time-master or time-slave. The router, like any other node in a SpW network, contains one six-bit time-counter. In master mode, a tick_in input is asserted periodically by the external host system. When it happens, the router updates its internal time-counter and asserts a tick_out signal on each port, so that all the ports emit the time-code (which is the new value of the time-counter). Two different modes of time-master operations are possible for the router: when its time-code interface receives a tick, the time-code that is sent out is either internally generated (by incrementing the time-counter) or provided by the host system. When the router, in time-slave mode, receives a valid time-code (i.e. one more, modulo 64, than its current time-counter) from one of the SpW interfaces, it increments its time-counter, emits a tick signal and sends out the time-code. The tick signal and the time-code propagate to all the ports of the router except the one on which the time-code was received. If the received time-code is not valid, the time-counter is updated but the tick_out signal is not asserted.

The switching matrix and the ctrl units in Fig. 3 connect an input port to any output port. Data transfer through the switch is accomplished after an arbitration phase which

grants multiple requests at the same port according to a round robin priority scheme. The routing table provides the physical address of the destination, header deletion flag and group information for GAR. Following the standard, the routing table is organized as a memory space of 224 locations (corresponding to 224 logical addresses) each 10-bit wide storing the following data: the number of output port bound to the accessed logical address, the header deletion and the group information. The SpW router is configured by the internal programming interface accessible via the switching matrix from the AHB bus or from the SpW bus. The RMAP decoder in Fig. 3 allows the hardware management of the RMAP protocol, defined for remotely writing/reading memories, registers, FIFOs and mailboxes. It can be used for configuring a SpW network, controlling the units that make it up (i.e. its nodes) and collecting data from them. The RMAP hardware decoder implemented in our SpW router allows a remote control of the AHB bus from the SpW side. All the accessible registers, FIFOs and memories are seen as mapped to a memory space. In this space, a specific address is reserved for the router, in order to allow for programming it by means of RMAP commands. The default SpW logical address of the RMAP decoder is 254. After initialization, packets with a logical address of 254 are either accepted or rejected depending on the value of a settable flag. When a non-RMAP packet (i.e. a packet with a PID≠1) is received by the RMAP hardware decoder, it is passed on to the host system. This allows other high level protocols to be handled software-wise by the host unit. From an architectural point of view the RMAP decoder & SpW/AHB wrapper unit consists of a finite state machine, implementing the RMAP protocol rules, plus 1.5 Kbytes of local memory resource and dedicated units for CRC calculation and AHB wrapper.

3.2. SpW Interface

Each SpW interface, whose architecture is sketched in Fig. 4, is composed of a SpW Encoder-Decoder (Codec) and a SpW I/O wrapper. The SpW Codec is made up of three main parts: an encoder (or transmitter), a decoder (or receiver), and a finite state machine which implements the Exchange Level Protocol. The SpW I/O wrapper connects the SpW Codec to the router core through transmitting (TX) and receiving (RX) FIFOs. To avoid receiver FIFO overflow and loss of data, data flow across a link is controlled using the FCT control characters [1]. Through FCT a SpW interface indicates to an external transmitting link the amount of free space in its receiving FIFO. To be noted that SpW FIFO buffers have been implemented through dedicated memory resources available on the target FPGA to save logic resources. The SpW interface is also responsible for error detection and recovery at exchange level. The link detects: Parity error, Char Sequence error, Escape Sequence error, Disconnect error and Credit error. The Parity error event occurs if a parity

error is detected by the link; the parity bit for a data or control character is contained in the following character. The Character Sequence error is detected by the state machine in the link interface when an incorrect sequence is received, for example during link initialization. The Escape error occurs if an ESC character is followed by any control character other than an FCT. Disconnect error is detected when no new data bit is received within a link disconnect timeout window (850 ns) set by the standard. Credit error occurs if data is received when the host system is not expecting any more data or if an unexpected FCT is received (a maximum of 7 FCTs can be outstanding at any time). If an embedded system needs SpW connections without routing capability, one or more SpW interfaces can be integrated and used in standalone mode through a direct connection to AMBA bus by a wrapper. AHB bridge in Fig. 3 is a wrapper between the SpW router and the AHB AMBA bus, a de-facto standard in embedded system design. This module can act as both master and slave on the AMBA bus. Moreover, it can ask the control of the bus to execute a DMA transfer into memory devices, or to initiate a SpW remote control session to a host system connected at AHB. To interface the SpW Router of Section 3.1 to pre-existing automotive networks a CAN module can be connected as a slave on the AHB Router interface. This way the Router will act also as a gateway between the CAN and the SpW networks. To this aim it is possible to reuse one of the IP macrocells, available in the market, implementing a CAN node with AMBA interface.

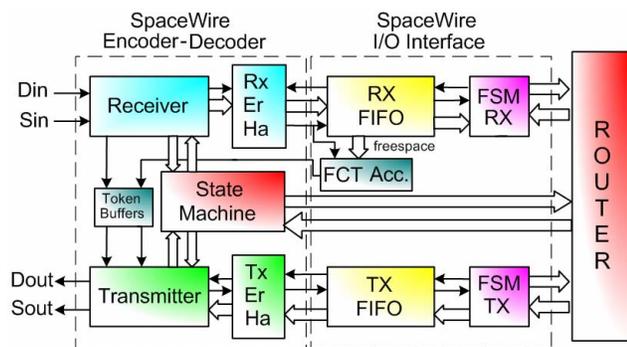


Figure 4: Architecture of the SpW interface

4. Design Validation Flow

The above IP cores have been used to build a complete SpW network solution within the ESA project IPPM: Integrated Payload Processing Module, carried out by Aurelia Microelettronica (prime), Caen Aerospace and Consorzio Pisa Ricerche with the University of Pisa [17]. The SpW IP cells have been tested at two levels:

- 1) test of the developed VHDL code using the ESA SpW IP Codec as reference link [18];
- 2) test of the FPGA prototyped router using as reference a commercial version of SpW links (a PCI-SpW card [19] in

Xilinx FPGA featuring 3 unrouted SpW links) and a SpW link analyzer provided by ESA, see Fig. 5.

The first set of tests have been conducted on the VHDL code at functional, post-synthesis and post-fitting level of SpW router vs. the HDL SpW Codec provided by ESA. The SpW router IP has been synthesized on several FPGA devices, e.g. SRAM Stratix II Altera and anti-fuse Actel Accelerator families. Exhaustive tests have been performed in a simulation environment created using the SpW router IP cell with 8 links each connected with an external ESA SpW IP Codec. On the host side the SpW router IP has been interfaced with VHDL models of AMBA AHB master and slave units. Performed tests include for example transmission and reception of SpW packets between the SpW links using different types of addressing (Path, Logical and Regional) and communication speeds, programming phase of the router via a SpW link or AHB, remote access to host via a SpW link and error detection and recovery protocols.

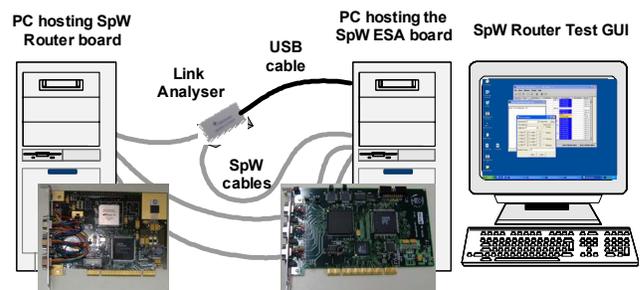


Figure 5: Emulation test environment

After validating the VHDL code of the SpW router the IP macrocell has been implemented in FPGA technology. As reported in Fig. 5, the FPGA has been connected to a commercial SpW card provided by ESA. At FPGA level, test procedures similar to the aforesaid VHDL level ones have been applied but with a much larger volume of data (emulation is much faster than simulation). The relevant timing results take into account also the contribution of FPGA I/O, board layout, SpW cables and connectors. Such tests have been successfully performed on a board by Aurelia Microelettronica hosting a Stratix II EP2S60 FPGA. Currently we are repeating the set of tests on the anti-fuse Actel (RT)AX2000 device. Being reprogrammable, the EP2S60 makes easier the VHDL design refinement phase before porting it on the one-time programmable Actel device. On the other hand anti-fuse devices outperform SRAM-based FPGA for their tolerance to Total Ionizing Dose (TID) and Single Events Effects (SEE) and for their higher temperature working range and thermal reliability [15]. Moreover anti-fuse ensures low-power dissipation. Hence anti-fuse is the first choice for avionic, military and space applications while for automotive the choice depends on the desired trade-off between cost and the level of reliability required by the automotive sub-system.

5. Implementation Results

Anti-fuse Actel technology does not need any mitigation techniques, since antifuses are not upset or damaged by heavy ions at the energy levels found in space or in atmosphere and triple-module redundancy is implemented at component level. The RTAX devices, space-qualified versions of the AX devices, bear a TID up to 300 Krad with SEU immunity up to 60 MeV·cm²/mg. They also feature embedded RAM with EDAC capability and integrated SRAM scrubber and an upset rate minor than 10⁻¹⁰ errors/bit-day. The 8-link SpW router with AMBA AHB interface results in a complexity of about 180 K gates for the logic and 2800 bytes of RAM. With reference to Fig. 3, roughly 85 % of the logic complexity is due to the Router core and the SpW interfaces while the rest is due to the RMAP decoder plus SpW/AHB wrapper. The complete 8-link SpW router with AMBA AHB interface fits on AX2000, a device providing 2 million gates equivalent to roughly 250K ASIC gates. For the customized 8-link SpW router, data-rates higher than 130 Mbps are supported in worst conditions on the AX2000, speed grade -3. The occupation of available logic is about 80%. For the EP2S60 SRAM FPGA, providing roughly 725K equivalent ASIC gates for the logic, the occupation of the 8-link SpW router IP is about 25% of the available resources while the transmission rate increases up to 200 Mbps. These performances meet the requirements of space, avionic and emerging automotive systems. To be noted that the circuit complexity of the router macrocell depends on the number of supported SpW links: e.g. a 4-link router can be also fitted on a smaller (RT)AX1000 device providing roughly 125K ASIC gates.

With respect to the state of the art, the proposed SpW routing solution is the first IP macrocell, mapped in anti-fuse and SRAM technology, and compliant with the complete standard: 6 basic protocol levels plus PID and RMAP extensions. The state of the art is characterized by radiation-tolerant implementation of SpW interfaces only or by SpW routers not including PID and RMAP extensions. The future implementation of an ASIC router in ATMEL CMOS technology, with a 300 Krad radiation tolerance and 8 SpW links has been announced in [9] but is not available yet. The target for the ASIC is 200 Mbps but for a power consumption budget of 4 W. The anti-fuse FPGA router power consumption for typical applications is about 1.5 W using a 1.5 V supply for the core and 2.5 V supply for the LVDS I/O, with a saving factor 2.5 vs. the ASIC router. Our SpW interface and router IPs allow for higher flexibility vs. the announced ASIC. Thanks to parametric HDL design and AHB interface, our IPs can be easily integrated and customized in any embedded system where the designer can define its desired target technology and performance/complexity trade-off. On the contrary, the ASIC, which has a non standard host interface, will

provide a fixed solution in terms of complexity, energy cost, number of SpW links.

6. Conclusions

Driven by the experience of aerospace systems, this paper introduces the SpaceWire standard to the automotive field. The proposed SpaceWire networking solution simplifies the connectivity, provides network safety and redundancy and guarantees to handle very high bandwidth data flows, up to hundreds of Mbps, not covered by current standards such as CAN or FlexRay. The design of configurable SpaceWire router and interface hardware macrocells, the first in state of the art compliant with the newest standard extensions, PID and RMAP, are detailed. The macrocells have been integrated and tested on SRAM and antifuse FPGAs in the framework of an ESA project. The achieved performances meet the functional requirements of space, avionic and future automotive electronic systems.

References

- [1] N. Navet et al., "Trends in automotive communication systems", Proc. of the IEEE, Vol. 93, n. 6, June 2005, pp. 1204 – 1223
- [2] T. Nolte et al., "Automotive communications: past, current and future", IEEE ETFA, vol. 1, Sept. 2005, pp. 985 - 992
- [3] G. Leen, D. Heffernan, "Expanding automotive electronic systems", IEEE Computer, 2002, pp. 88-93
- [4] S. Muramatsu et al., "Image processing device for automotive vision systems", IEEE Intelligent Vehicle Symp. 2002, vol.1, pp. 121 – 126
- [5] Z. Nan-Ning et al., "Toward intelligent driver-assistance and safety warning system", IEEE Int. Syst. and Their Appl., vol. 19, n. 2, 2004
- [6] ECSS-E-50-12A "Space engineering, SpaceWire - Links, nodes, routers and networks", 24 Jan. 2003
- [7] ECSS-E-50-12 Part 2 Draft E, 21 Dec. 2005
- [8] ECSS-E-50-12 Part 2 Draft B, January 2005
- [9] S. Parkes et al., "SpaceWire: a spacecraft onboard network for real-time communications", IEEE-NPSS Real time conf., June 2005 pp. 6-10
- [10] G. Rakow et al., "Reliable transport over SpaceWire for James Webb space telescope", IEEE Aerospace Conf. 2003, vol. 5, pp. 2289 - 2302
- [11] Day-Young Kim et al. "Design of a new on-board computer for the new KOMPSAT bus", IEEE Aerospace Conf. 2005, vol. 1, pp. 1-12
- [12] J. Marshall et al., "Reconfigurable processing systems in spaceborne applications", Proc. IEEE Aerospace Conf. 2004, vol. 4, pp. 2354 – 2363
- [13] S. Yesil et al., "GOLGE: a case study of a secure data communication system for micro-satellites", IEEE RAST 2005, pp. 438-441
- [14] S. N. Chau et al. "A design-diversity based fault-tolerant COTS avionics bus network", IEEE Pacific Rim Int. Symp. 2001, pp 35 – 42
- [15] M. Mason et al., "FPGA reliability in space-flight and automotive applications", FPGA and programmable logic Journal, 2005
- [16] J. T. Giasch, S. Parkes, A. Christem, "From IEEE 1355 high-speed serial links to SpaceWire", DASIA'99, pp. 1-6
- [17] http://www.caen.it/micro/rd_IPPM.php
- [18] Spw12, SpaceWire IP core hardware manual, Astrium, 2003
- [19] www.star-dundee.com