

# INTEGRATED DEVELOPMENT TOOLS SUITE FOR THE SPACEWIRE RTC ASIC

**Session: SpaceWire Test and Verification**

## **Short Paper**

Walter Errico (1), Jørgen Ilstad (2), Annamaria Colonna (1), Fabrizio Bertuccelli (1)

*(1) Aurelia Microelettronica Srl, Via Vetraia 11, I-55049 Viareggio, Italy*

*(2) ESA-ETEC, Keplerlaan 1, NL-2201 AZ Noordwijk, The Netherland*

*E-mail: walter.errico@aurelia-microelettronica.it, Jorgen.Ilstad@esa.int*

### **ABSTRACT**

The SpaceWire RTC device is a fully integrated LEON2-FT based System on Chip that among other features, provides the capability to bridge traffic between the SpW network and the CAN bus. The SpW RTC Development Suite is the collection of HW/SW components designed to stimulate all the RTC interfaces and to drive all its communication links.

In this paper a brief description of the RTC ASIC and the Development Suite Tool are presented before focusing on the SpaceWire aspects of the two devices and their test chain.

## **1 SPACEWIRE RTC**

The SpaceWire Remote Terminal Controller (RTC) device is a system on chip for space application, including the LEON2-FT processor unit with on chip memory, a memory controller for external banks, two SpW Links, a CAN bus controller, ADC/DAC interfaces for analogue acquisition/conversion, FIFO controller, standard interfaces and resources (UARTs, timers, general purpose input output).

The embedded LEON2-FT microprocessor, its abundance of interfaces and the possibility to access the system via SpW and CAN makes the RTC suitable for many onboard applications. i.e as an intelligent node due its processing capability, or it can be remotely managed via its SpW link interfaces using the Remote Memory Access Protocol (RMAP)

The SpW-RTC can play the role of payload data processor inside the Instrument Controller Unit (ICU). It can receive payload data from instruments, process them and send down via SpW. Its several programmable interfaces allow RTC to acquire analogue and digital data, generated by peripherals. The on-chip interfaces have DMA capability, thus data transfers between SpW links and i.e FIFO can sustain high bitrates. The CAN bus allows the On-board Computer (OBC) to monitor and control the SpW-RTC device that acts as a remote terminal.

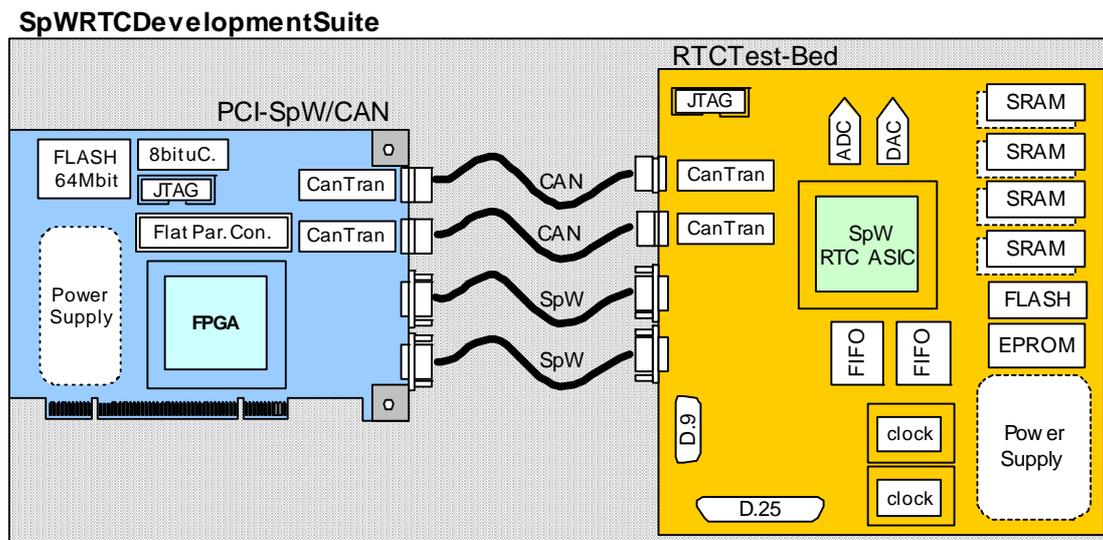
Alternatively, the networking features of the RTC may be exploited even if the device is integrated in the OBC. The CAN controller capability can be utilised for node

management and time distribution, while the SpW links are utilized to communicate and manage the SpW network itself.

## 2 DEVELOPMENT SUITE

The SpW RTC development suite is composed of two HW modules; PCI-SpW/CAN board, SpW RTC ASIC test bed, and a SW library for generating traffic on all of the SpW RTC communication ports, independently of the LEON2 debugger tool.

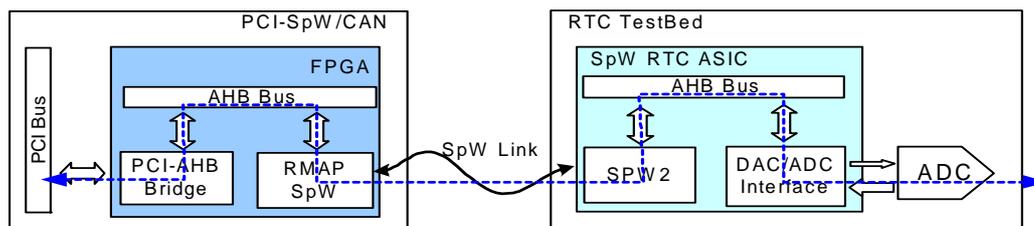
The PCI-SpW/CAN is a standard PCI peripheral FPGA based board housing 2 SpW links plus 2 CAN ports and a large parallel service connector. Ones plugged into a Linux PC it is used as a stimuli source for the RTC test bed.



**Fig 1 SpW RTC Development Suite HW modules**

The “RTC test-bed” is the external test board where the SpaceWire RTC ASIC is mounted in addition to peripheral units such as SRAM, FLASH, EEPROM, ADC-DAC, FIFO etc. The testbed includes possibility to interface custom made daughterboards via mezzanine connectors. Users can take advantage of the FIFO interface or ADC/DAC interfaces for breadboarding of i.e. instrument control modules or custom made FPGA designs.

Communication with the PCI-SpW/CAN board is achieved through the CAN and SpW links. The RTC test-bed is provided with the standard serial (UART) connector to be directly connected to the test PC where also the LEON2 GRMON debugger tool may be independently running.



**Fig 2 TestBed ADC measurement controlled via SpW**

The development suite SW library incorporates drivers and comm. functions to perform data exchange between the PCI-SpW/CAN and the testbed via SpW and CAN links. The comm. functions for the SpW links utilize RMAP which enables access to all internal registers and memory space of the SpW RTC ASIC.

Fig 2 shows an example of measurements done on the RTC ADC interface directly transferred via SpW to the test PC.

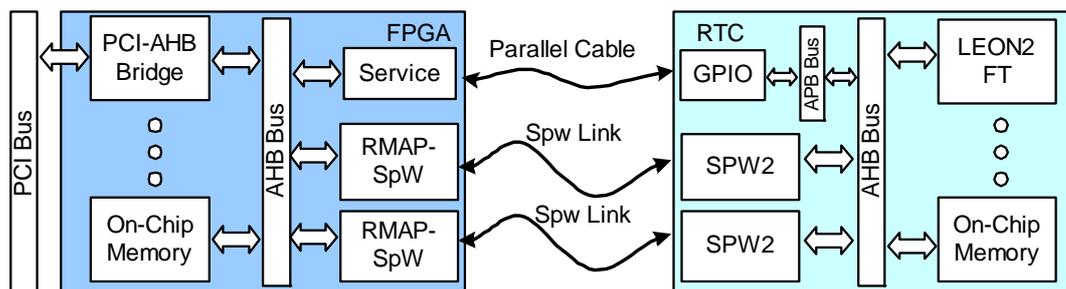
The division of the development suite in two distinct parts leaves the user with the possibility to combine the test suite with other 3rd party equipment.

In addition several innovative concepts are implemented in the proposed architecture of the PCI-SpW/CAN board:

- *Compactness:* Two CAN Controllers and two SpW ports are hosted on the same single board.
- *Flexibility:* CAN and SpW modules are RTL Core instances embedded in large FPGA. They can be easily modified and/or enhanced to follow any new protocol updates and features.
- *Maintainability:* A custom FPGA programming scheme (Aurelia A-AltPrg1) is implemented on the board to allow the users to update the FPGA image without need of any vendor specific SW/HW tool or licence. The users can load the new programming image directly through the PCI bus.
- *Simplicity:* The software developer is provided with a single API for all the communication links. All the internal buffers and control registers are easily accessible as they are mapped in a uniform memory segment in the PCI space.

### 3 SPACEWIRE TEST

The development suite structure foreseen for the SpaceWire Test is shown below. The RTC ASIC SPW2 modules are connected to the Aurelia RMAP-SPW interfaces implemented in the FPGA of the PCI-SpW/CAN board. Both types of modules are provided with an AMBA AHB interface.



**Fig 3 SpaceWire test chain**

The SPW2 has access to the full range of all RTC registers and memory space via the AHB bus, thus its DMA engines can perform transmission and receptions of data without involving the LEON2-FT processing unit support during the transfers.

<sup>1</sup> A-AltPRG is the Aurelia integrated solution for Altera FPGA programmability on-board without need of proprietary software and external cables.

Due to the complex memory hierarchy of the Linux operating system the reception of RMAP packets is buffered in the on-chip memory of the FPGA. A DMA controller is included in the PCI-AHB bridge that optimizes large data transfers to and from this buffer.

#### 4 SPACEWIRE RTC SPW2 MODULE

The RTC ASIC is equipped with two instances of the SPW2 Interface module. The SPW2 module supports RMAP (Remote Memory Access Protocol) commands and the VCTP (SpW Virtual Channel Transfer Protocol) encapsulating the University of Dundee SpW CODEC for the low layers SpW link protocol. High-level non-supported commands and protocols are redirected to the software.

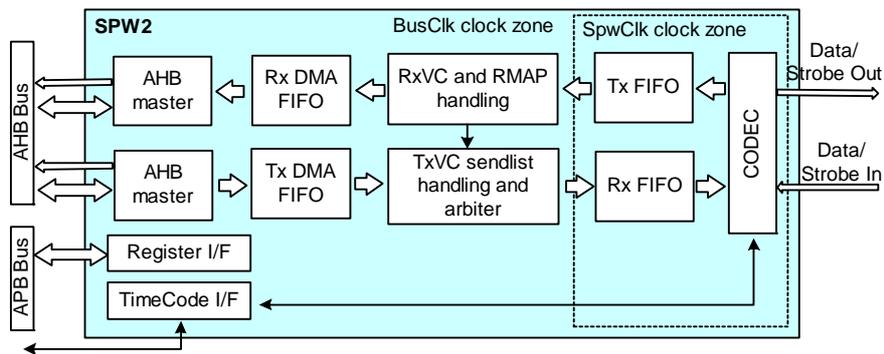


Fig 4 SpaceWire RTC SPW2 module

The RMAP allows remote users to access the entire RTC memory space using the explicit AMBA address to access also the system control and configuration registers. The RMAP is robust in that it offers CRC protection of the cargo and destination key check. It effectively allows for system control via SpW.

The VCT protocol utilises a pre defined memory area within the RTC memory space, defined using configuration registers. This effectively establishes a virtual channel which is well adapted to transfer larger data payloads.

#### 5 DEVELOPMENT SUITE RMAP-SPW CORE

The RMAP-SpW core internal structure is drawn in Fig 5. The module is composed of a SpW link interface, a RMAP receiver and two independent AMBA AHB Master and Slave interfaces.

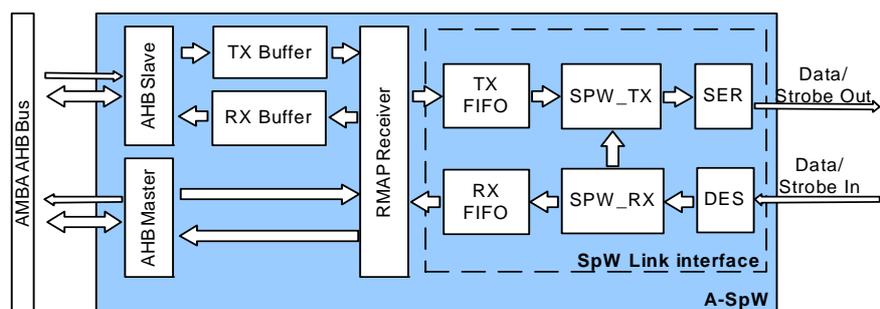


Fig 5 RMAP-SpW core block diagram

The RMAP receiver filters all the incoming packets. It recognizes the remote memory access requests and transfers them on the AMBA bus through the AHB Master interface. Then it automatically sends back the required acknowledges.

The incoming non-RMAP (or RMAP response) packets are transparently written in the RX Buffer, and outgoing data are fetched from the TX Buffer and forwarded toward the SpW link without any filtering. TX and RX buffers are accessed from the AMBA bus through the AHB Slave interface.

The TX Buffer is designed for efficient transmission of outgoing packets without any limitation on format and length. A 1024 x 32 transmission buffer is managed as an outgoing FIFO connected to the physical link like a free running output pipe. Specific locations are dedicated to transmit non d-word (32 bit) aligned bytes and End Of Packets flags. The RX Buffer is capable of simultaneously manage up to 8 packets. A 1024 x 32 memory buffer temporarily stores input data in an “almost-FIFO” structure. The data inside the buffers are not automatically removed after reading, but each application has to explicitly notify the number of data and packets that have been consumed; afterward logic shift take place in the buffer so that the offset of the first valid entry is always 0x000. The packets delimiter pointers are automatically updated.

## **6 CONCLUSIONS.**

The ever-increasing focus on reducing cost for space missions by reducing development time and time-to-test puts the SpW-RTC ASIC together with the development suite in the spotlight. The device is well suited for a number of applications, both at platform and payload level. The development suite toolbox allows users to start breadboarding activities at an early stage as both S/W and H/W tools included in the suite contributes to rapid prototyping.

The new SpaceWire interface (SPW2) integrated in RTC devices includes the Star Dundee CODEC core connected with the HW implementation of the two RMAP and VCTP high-level protocols. Both protocols permit the direct connection (DMA) between the system memory and the SpW link relieving the processing unit from the transfer data flow management. In addition, the RMAP protocol can be utilized also for the system remote control via SpaceWire.

On the Development suite side the RMAP-SpW core embedded in a fast FPGA is used. The RMAP-SpW interface operates up to 400 Mbit/sec and includes a RMAP receiver capable of accessing all the FPGA AMBA memory space. A PCI-AHB bridge makes the FPGA AMBA space available on the host Linux PCI space and includes a DMA controller for fast memory transfers. The task to fully utilize the new protocols features of the SPW2 is assigned to the Suite SW library that will be running on the Linux test PC.

The SpW implementations of the device under test and the tester tool come from separated sources and for several aspects follow different approaches. This matter increases the validity of the test activity reducing the risk that common problems mask each other.