WHITE RABBIT STATUS AND PROSPECTS

J. Serrano, M. Cattin, E. Gousiou, E. van der Bij, T. Włostowski, CERN, Geneva, Switzerland
G. Daniluk, AGH University of Science and Technology, Krakow, Poland
M. Lipiński, Warsaw University of Technology, Warsaw, Poland
D. Beck, J. Hoffmann, M. Kreider, C. Prados, S. Rauch, W.W. Terpstra, M. Zweig
GSI, Darmstadt, Germany

Abstract

The White Rabbit (WR) project started off to provide a sequencing and synchronisation solution for the needs of CERN and GSI. Since then, many other users have adopted it to solve problems in the domain of distributed hard real-time systems. The paper discusses the current performance of WR hardware, along with present and foreseen applications. It also describes current efforts to standardise WR under IEEE 1588 and recent developments on reliability of timely data distribution, finishing with an outline of future plans.

INTRODUCTION

The White Rabbit (WR) project [1] was initiated in 2008 and will serve as the basis for the renovation of the existing accelerator timing systems at CERN and a new clock and event distribution system for the upcoming FAIR facility at GSI. A WR network is basically a switched Ethernet network in which nodes automatically get sub-ns synchronisation. WR switches allow users to build highly deterministic data networks by having different internal queues for Ethernet frames of different priorities, as established by the priority header defined in IEEE 802.1Q. The combination of deterministic latencies and a common notion of time to within 1 ns allows WR to be a suitable technology to solve many problems in distributed real-time controls and data acquisition. For example, commands instructing nodes to carry out a task can be sent with an execution International Atomic Time (TAI) stamp attached. The actual reception time in the nodes will have some jitter, but that jitter will have an upper bound. Provided nodes have received the message in a timely manner, the execution of the tasks is synchronised using the attached TAI stamp with sub-ns accuracy.

The main technologies involved in WR are the Precise Time Protocol (PTP, IEEE 1588), layer-1 syntonization and phase tracking, and have been described elsewhere [2]. In the following sections, we describe the basic building blocks of a WR network.

THE WHITE RABBIT SWITCH

WR is a switched network. At its heart lies its most important component: the WR switch, which provides 18 ports in a 1U 19” rackable enclosure. It is made of open source hardware, gateware and software, and it is sold and supported by a commercial company. WR switches are fully compatible with Ethernet, and can identify if a WR node or another WR switch is hooked to one of their ports by using the WR extension [3] to the IEEE 1588 protocol at link establishment time. This extension is also designed to be backwards-compatible with standard PTP, so it is possible to connect existing PTP gear to a network made with WR switches, along with WR nodes. In this case, the WR nodes will benefit from the extension and therefore achieve better accuracy, while the standard PTP nodes will run only the standard protocol and feature reduced accuracy.

Architecture

Figure 1 shows a high-level block diagram of the WR switch. Ethernet frames are exchanged through 18 ports equipped with Small Form-factor Pluggable (SFP) sockets which can host optical transceivers. The reference implementation uses SFP modules for one single mode fibre, using one wavelength for TX traffic and a different one for RX. The use of a single fibre ensures that the symmetry in TX and RX paths – after mathematically compensating for fibre dispersion – is robust, in particular against changes in cabling not notified to the final user. The SFPs are connected directly to a Xilinx Virtex-6 Field Programmable Gate Array (FPGA).

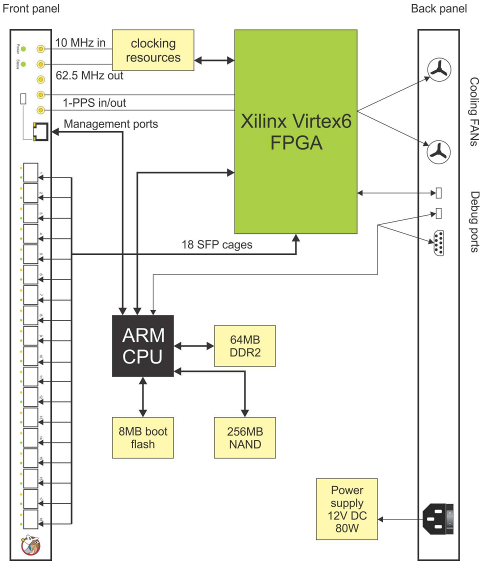

Figure 1: High-level block diagram of the WR switch.

Ethernet frames get switched inside the FPGA with very low latency. An ARM CPU running Linux helps with less time-sensitive processes like remote management and keeping the frame filtering database in the FPGA up to date.

The clocking resources block contains PLLs for cleaning up and phase-compensating the system clock.

Performance

In order to characterise the performance of the WR switches, a system was set up in a laboratory consisting of four cascaded WR switches. The master switch was connected to a first slave switch through a 5 km fibre roll. Similar fibre rolls were used to connect the first switch to the second one, and then the second one to the third one, for a total of 15 km of fibre. Adverse conditions were simulated by heating the fibre rolls with a hot air gun.

Since the four switches were all in the same laboratory, it was easy to monitor their Pulse Per Second (PPS) outputs with an oscilloscope and draw histograms of the offsets between the PPS output in each switch and the PPS output in the master switch. The results of these measurements can be seen in Fig. 2.

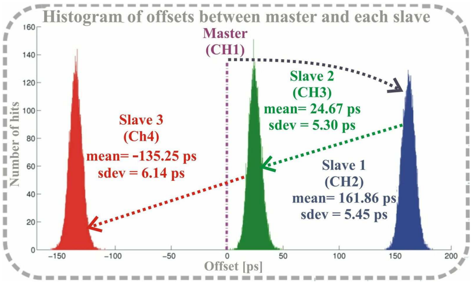

Figure 2: Histograms of PPS output offsets of three cascaded WR switches with respect to the PPS pulse output in the master switch.

As can be seen in the plots, accuracy always stays within ± 200 ps, and typical precisions are in the order of 6 ps. The same type of accuracy and precision is found with WR nodes, since the technologies involved for delay measurement and compensation are exactly the same as those used in the switches.

WHITE RABBIT NODES

Figure 3 shows a simplified block diagram for a WR node based on the Simple PCI Express Carrier (SPEC) board. This card can host mezzanines conforming to the FPGA Mezzanine Card (FMC) VITA 57 standard. We have developed Analogue to Digital Converter (ADC), Time-to-Digital Converter (TDC) and programmable delay generator FMCs. By plugging these cards in a WR-enabled carrier such as the SPEC, and appropriately configuring the FPGA in the carrier board, one can enhance their functionality with features such as synchronous sampling clocks in remote nodes and precise TAI stamps.

In order to enable users to easily build nodes for WR-based applications, we have developed a core which takes care of all WR data transmission and reception, along with all synchronisation tasks. This WR PTP Core (WRPC) [4] – an Ethernet Medium Access Control (MAC) unit with enhancements for timing – contains an LM32 soft CPU inside which runs the whole PTP stack. Frames which are identified as non-PTP are forwarded downstream to the user logic. Conversely, the core also accepts frames from user logic, which can for example be used to stream data acquired in an ADC FMC. The WRPC takes care of controlling the programmable oscillators on the SPEC or any other WR-enabled board. An Etherbone slave core [5] can optionally be instantiated between the WRPC and the user logic. Etherbone is an independent project led by GSI which can work in conjunction with any Ethernet MAC core, not necessarily with the WRPC. It aims at providing a way to trigger reads and writes in a remote Wishbone bus through carefully defined payloads in an Internet Protocol (IP) packet. With Etherbone, a complete network of sensors and actuators looks like a big memory map to a master/management node.

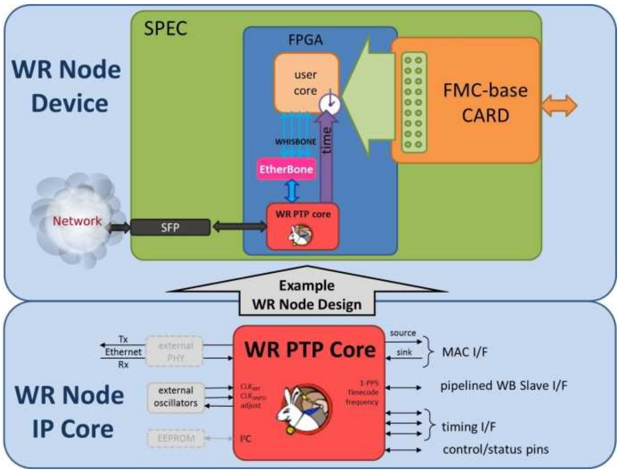

Figure 3: An example WR node.

WR-capable nodes have been designed in PCIe, PXIe, VME64x and μTCA form factors. These designs have varying degrees of maturity and commercial support. The most mature one is the PCIe SPEC board, and different WR-based gateware designs have been successfully targeted at it, including a Network Interface Card with a Linux network device driver.

APPLICATION EXAMPLES

WR technology provides users with a common notion of TAI in every node and with a deterministic network in which an upper bound for latencies is guaranteed by design. This opens up many possibilities in different fields, such as Multiple Input Multiple Output (MIMO) feedback systems. In this section, we describe just two of the multiple possible applications of WR.

RF Distribution

WR distributes a 125 MHz clock – typically TAI-related – for free. A WR user gets access to this clock, or derivatives of it, just by hooking a WR node to a WR network. However, in some cases users care more about synchronising to Radio Frequency (RF) signals related to e.g. the accelerating structures in a particle accelerator. Generating phase-compensated RF signals in different locations can be useful in other domains as well, such as in radar applications.

Figure 4 shows a block diagram of how one can use a WR network to distribute RF clocks, through a scheme called Distributed Direct Digital Synthesis (Distributed DDS or D3S).

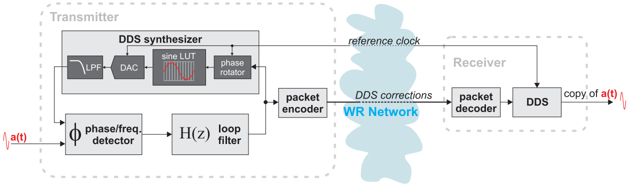

Figure 4: Distributed DDS in a WR network.

The reference clock line in the drawing is just conceptual. Nodes get the reference clock from the WR network itself. There is no need for additional connections. The transmitting node tracks an RF signal connected to its input with a PLL in which the role of the voltage-controlled oscillator is fulfilled by a DDS block. The control words for that DDS, along with the TAI at which they were applied, are encoded and broadcast through the WR network. Receiving nodes can then apply a fixed offset to the TAI stamps and replay the RF waveform with their local DDS blocks, with just a fixed delay. In most situations the RF is stable enough for this fixed delay to be of no concern.

This scheme has several advantages over traditional RF distribution systems. There is no additional cabling to be done. The same network can handle more than one distributed RF. In addition, all waveforms are played using a TAI-related clock, which is very useful for diagnostics. A first crude implementation has been demonstrated at CERN, with a jitter – defined here as the integral of the Power Spectral Density of the phase noise, integrating between 10 Hz and 5 MHz – in the replayed RF below 10 ps. Better jitter can be achieved by carefully tuning the digital PLL filter and cleaning the output of the DDS with an analogue PLL including a low phase noise oscillator.

Distributed Oscilloscope

Figure 5 shows a conceptual representation of a distributed oscilloscope using several of the building blocks we have described so far.

ADC nodes sample analogue signals synchronously in remote locations. The synchronous phase-compensated sampling is facilitated by the WR-derived clock. The ADCs can store their samples in rolling buffers, where each location is known to contain the sample corresponding to a precise TAI. As soon as an ADC node detects a condition upon which it should trigger, it can broadcast a trigger message through the WR network. All nodes will have received that message after a guaranteed worst-case delay. These nodes can then stop sampling and rewind their buffers to the TAI specified in the triggering message. External trigger pulses can also be accommodated through the use of TDC nodes, and sampling with non-TAI-related clocks can be done using D3S nodes.

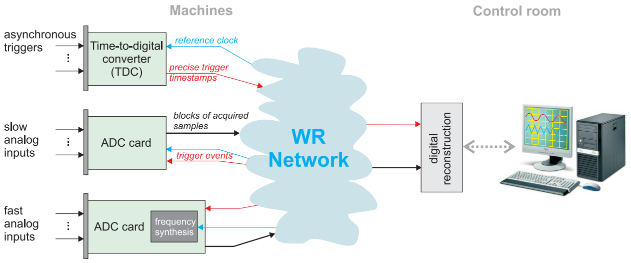

Figure 5: Distributed oscilloscope using a WR network.

A computer in the control room then gets all the TAI-stamped acquisition data and displays it coherently on a single screen. The operators then see this whole distributed system as a simple oscilloscope to which all their signals are connected. This would of course be impossible with a real oscilloscope, but it can be done with WR's phase-compensated distribution of TAI, appropriate WR-enabled nodes and software.

PROJECT STATUS AND OUTLOOK

At the time of this writing (August 2013) the WR switch is a mature commercially-supported hardware product with a stable release for its gateware and software. WR nodes have been designed and validated in a variety of formats, and they have consistently shown sub-ns synchronisation accuracy. This section provides a quick overview of some of the most important upcoming efforts in the project.

Synchronisation Performance

The results in Fig. 2 show that the original goals for WR in terms of synchronisation have been achieved. In some applications though, it is important to improve accuracy as much as possible. While the WR delay model [6] and compensation mechanism cover changes in fibre temperature appropriately, they do not compensate for the delay fluctuations induced by temperature variations in the nodes and switches themselves. We believe these effects can account for tens of picoseconds or even more in particularly adverse circumstances.

Tests in a climatic chamber [7] have shown that the dependency of delay with respect to temperature is mostly linear, so an evolution of the WR delay compensation algorithm which takes into account on-board temperature measurements should help in this respect. The modified algorithm could include a linear model of the dependency or a table of delay vs. temperature values resulting from offline calibration.

Switch Evolution

The efforts on switch gateware and software will go in the direction of providing better support for remote management, diagnostics and robust data transmission in a WR network.

A Port Statistics block will be added in the switch FPGA to provide counts of transmitted, received and dropped frames per port, along with other counts relevant for ascertaining the state of health of the network. A Time-Aware Traffic Shaper Unit (TATSU) will allow blocking selected output queues for some time so that high-priority frames can go through the switch without colliding with low-priority frames being sent at that moment through a port. This will avoid having to wait until the end of transmission of those frames before taking ownership of the port.

In addition, a Topology Resolution Unit (TRU) will add hardware support to the process of providing redundant loop-free topologies for a network. This will result in a faster topology switch-over compared to traditional software-based methods such as the Rapid Spanning Tree Protocol (RSTP). Figure 6 illustrates why fast switch-over can be important in a WR network.

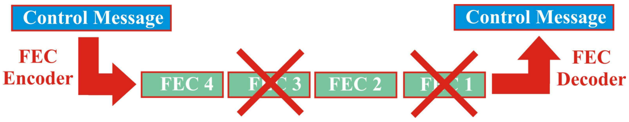

Figure 6: Forward Error Correction used to correct for frame loss.

In this example, a WR node decomposes a control message into four Ethernet frames which have been encoded using Forward Error Correction (FEC), in such a way that receiving any two of those frames allows the receiving node to re-constitute the original control message. The use of this type of FEC scheme is well adapted to networks in which a late frame is a wrong frame, such as those used for accelerator timing at CERN and GSI. This precludes the use of protocols which require re-trials in case of transmission errors, such as the Transmission Control Protocol (TCP).

If the switch can enable a redundant port very quickly after detecting a fatal condition in another port, and if the switch-over time is lower than that corresponding to the transmission of one of the frames in the figure, the receiving node will be able to re-constitute the message with the remaining frames. Tests at CERN have shown that this kind of switch-over speeds are indeed possible with appropriate hardware support.

On the software front, most of the activity will be focused on providing the switch with good Simple Network Management Protocol (SNMP) support so that all control and diagnostics can happen remotely using standard switch management software. Another important development will be the replacement of the current PTP stack running in the ARM processor by the PTP Ported to Silicon (PPSi) stack. PPSi [8] is a portable PTP daemon which can be targeted at bare-metal systems, such as the LM32 processor inside the WRPC in the WR nodes, but can also run under an operating system, such as the Linux running in the ARM processor in the switch. PPSi has been developed in the frame of the WR project but can be used in any project requiring PTP support. It is licensed under LGPL.

Standardisation

Using standards is good for many reasons. A standard is more likely to be used for a long time, so using it reduces long-term risks. Standardisation bodies typically invest a big effort in making sure standards are robust, so adopting them also saves time. Finally, companies are more inclined to participate in a development project if it is based on standards, because markets are typically larger in that case and also because the rules of governance and evolution of the standard are clear from the outset.

In WR, we are developing functionality which is not made available by any existing standard. However, it was felt from the beginning of the project that WR ideas could constitute the basis for the evolution of the IEEE 1588 standard. The original WR specification was already worded with this in mind. Now the IEEE P1588 Working Group has opened the process for the periodic revision of the standard, and the WR project is represented in the subcommittee for high accuracy enhancements. The aim of this effort for the WR team is to end up in a situation where WR is just a particularly accurate and precise implementation of the IEEE 1588 standard. The work of the subcommittee has just started, and the outlook is very promising.

WR is also concerned with determinism, and the WR team keeps an eye on the efforts of the Time Sensitive Networks (TSN) Task Group inside the IEEE 802.1 Working Group. The ideas discussed in this Task Group are of great relevance to WR. Conversely, the TRU and TATSU blocks in the WR switch FPGA can be a very convenient testing ground for these ideas.

REFERENCES

[1] White Rabbit project website, `http://www.ohwr.org/projects/white-rabbit/wiki`

[2] J. Serrano et al., “The White Rabbit Project,” IBIC 2013.

[3] M. Lipiński et al., “White Rabbit: a PTP Application for Robust Sub-nanosecond Synchronization,” ISPCS 2011.

[4] WR PTP Core, `http://www.ohwr.org/projects/wr-cores/wiki/Wrpc_core`

[5] Etherbone Core project, `http://www.ohwr.org/projects/etherbone-core/wiki`

[6] White Rabbit Specification, `http://www.ohwr.org/documents/160`

[7] WR torture report, `http://www.ohwr.org/documents/190`

[8] PPSi, `http://www.ohwr.org/projects/ppsi/wiki`