Applications of White Rabbit Technologies

White Rabbit Trigger Distribution

Dimitris Lampridis

CERN, BE-CO-HT

BE/CO Technical Meeting, 14 May 2020
Outline

1. Introduction
2. WRTD
3. Demonstrator
4. Current Status and Outlook
Outline

1. Introduction
2. WRTD
3. Demonstrator
4. Current Status and Outlook

Applications of White Rabbit Technologies
Scientific experiments often require correlated data acquisition.

Synchronisation in large scale installations can be a serious challenge.

Triggers may be external events, but also “internal”.

In any case, triggers will not arrive simultaneously on all acquisition nodes.
What is WRTD?

**White Rabbit Trigger Distribution**

- A system to distribute events (aka triggers) over a White Rabbit (WR) network.
- An event is simply an ID with a timestamp (“what” and “when”).
- Leverage WR to provide deterministic delivery of events to all nodes.
- An evolution of the LIST project.

**Applications of White Rabbit Technologies**

- CERN, BE-CO-HT
Outline

1. Introduction
2. WRTD
3. Demonstrator
4. Current Status and Outlook
How it Works

Scenario #1: Event Distribution

Node #1

Node #2

Node #3
How it Works
Scenario #1: Event Distribution

Event

Node #1

Node #2

Node #3
How it Works

Scenario #1: Event Distribution

- Event
- $T_0$ Node #1
- $T_S = T_0 + \Delta$
- Node #2
- Node #3

Applications of White Rabbit Technologies

CERN, BE-CO-HT
How it Works

Scenario #1: Event Distribution

- **Event**
  - Node #1
  - $T_0$

- **Event Tx**
  - Node #1
  - $T_1$

- $T_0 = T_0 + \Delta$

- **Node #2**

- **Node #3**
How it Works

Scenario #1: Event Distribution

- Node #1: Event $T_0$ to Event $T_1$
- Node #2: Event $T_1$ to Event Rx $T_2$
- Node #3: Event Rx $T_2$ to $T_3$

$T_3 = T_0 + \Delta$
How it Works

Scenario #1: Event Distribution

- Event
- Event Tx
- Event Rx

Node #1: $T_0 \rightarrow T_1 \rightarrow T_5$

Node #2: $T_2 \rightarrow T_5$

Node #3: $T_3 \rightarrow T_5$

$T_5 = T_0 + \Delta$
How it Works

Scenario #1: Event Distribution

Event

Node #1

T₀

T₁

Event Tx

T₂

T₃

T₄ = T₀ + Δ

Event Rx

Node #2

Action

Node #3

Event Rx

Action

T₅

T₆
How it Works

Scenario #2: Distributed Data Acquisition

- Node #1
  - Trigger
  - $T_0$ to $T_1$
  - $T_1$ to $T_5$

- Node #2
  - Trigger
  - $T_2$ to $T_5$

- Node #3
  - Trigger
  - $T_3$ to $T_5$
How it Works

Scenario #2: Distributed Data Acquisition

Applications of White Rabbit Technologies
How it Works

Scenario #2: Distributed Data Acquisition
Inside a WRTD Node

- A Node contains a list of Rules.
- A Rule associates an incoming event to an outgoing one.
- Local channels connect the Node to its environment.
- Network messages send/receive events of the WR network.
- Alarms can also be used as event sources.
# WRTD Event Message Format

<table>
<thead>
<tr>
<th>Offsets</th>
<th>Octet</th>
<th>Bit</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 1234567</td>
<td>89101112131415</td>
<td>1617181920212223</td>
<td>2425262728293031</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0x4c ('L')</td>
<td>0x58 ('X')</td>
<td>0x49 ('I')</td>
<td>0x00</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td>Event ID [0:3]</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td>Event ID [4:7]</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>96</td>
<td></td>
<td></td>
<td></td>
<td>Event ID [8:11]</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>128</td>
<td></td>
<td></td>
<td></td>
<td>Event ID [12:15]</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>160</td>
<td></td>
<td></td>
<td></td>
<td>Sequence number</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>192</td>
<td></td>
<td></td>
<td></td>
<td>Event timestamp seconds (32 lower bits)</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>224</td>
<td></td>
<td></td>
<td></td>
<td>Event timestamp nanoseconds</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>256</td>
<td></td>
<td></td>
<td>Event timestamp fractional nanoseconds</td>
<td>Event timestamp seconds (16 upper bits)</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>288</td>
<td></td>
<td>0x00</td>
<td>0x00</td>
<td>0x00</td>
<td>0x00</td>
</tr>
</tbody>
</table>
Standardisation Effort

- WRTD library looks and feels very similar to an IVI Extension API.
- Inspired by IVI-3.15 IviLxiSync specification.
- Uses the same event message format as LXI Event Messages.
- Ongoing discussions with the IVI Foundation and LXI Consortium to merge WRTD with the IVI standard as an extension.
- It would allow us (and everyone else) to buy off-the-shelf WRTD-enabled products.
WRTD Reference Designs

Quad channel, 14-bit, 100MSPS Digitiser
- PCI Express format (SPEC)
- 5x inputs, 1x output
- 12 µs input latency
- 12 µs output latency
- Distributed Oscilloscope

Time-to-Digital Converter (TDC) + Fine Delay Generator
- VME format (SVEC)
- 5x inputs, 4x outputs
- 20 µs input latency
- 40 µs output latency
- Generic trigger distribution

More to come:
- Dual channel, 8-bit, 1GSPS Digitiser
- ...

Applications of White Rabbit Technologies
What is it made of?

- Application Core(s)
- WR Core
- Application Firmware
- WRTD Firmware
- MockTurtle Firmware
- MockTurtle CPU(s)
- Node FPGA
- WR Tools
- WRTD Python
- WRTD Library
- MockTurtle Library
- MockTurtle Driver
- Host

Applications of White Rabbit Technologies
Using the WRTD Tool

# Declare and configure the event source
wrtd-tool add-rule 1 rule1
wrtd-tool set-rule 1 rule1 LC-I2 NET0
wrtd-tool enable-rule 1 rule1

# Declare and configure the event destination
wrtd-tool add-rule 2 rule2
wrtd-tool set-rule 2 rule2 -d 50u NET0 LC-01
wrtd-tool enable-rule 2 rule2

- The tool also provides easy access to the event log, as well as statistics and diagnostics.
- The Library, Python wrapper and tool are fully documented in:

  https://wrtd.readthedocs.io
Outline

1. Introduction
2. WRTD
3. Demonstrator
4. Current Status and Outlook
WRTD Demonstrator
The “Distributed Oscilloscope”

- Use WRTD to create the illusion of a “benchtop” oscilloscope with signals coming from various distributed digitisers.
- Use it as our playground for testing new ideas.
- Make it a portable system, easy to replicate outside of CERN.
- Use it as a compatibility tester for third-party, WRTD-enabled digitisers.
Demonstrator Setup

- **Host PC**
- **2x FMC-ADC**
- **Signal Generator**
- **2.5km fiber**
- **50m fiber**
- **WR Switch**

Applications of White Rabbit Technologies
Demonstrator GUI
Outline

1. Introduction
2. WRTD
3. Demonstrator
4. Current Status and Outlook
v1.0 released in September 2019.
Used in-house for the “Distributed Oscilloscope” Demonstrator.
  Demo video available at Indico.
To be used by CERN in OASIS trigger distribution over WR.
Used in the Shanghai soft X-ray Free-Electron Laser User Facility (SXFEL-UF).
Under evaluation by the Shanghai HIgh repetition rate XFEL aNd Extreme light facility (SHINE).
Under evaluation by Fermilab for their Neutrino detection experiment.
Summary

- WRTD provides sub-nanosecond accurate event scheduling for large-scale experiments.
- WRTD is stable and fully documented, providing a variety of configuration and control methods (C, Python, CLI).
- To be used in and outside of CERN.
- The WRTD reference designs act as turn-key solutions for pulse distribution and distributed data acquisition applications, or as solid examples for other applications.
- More reference designs will soon follow.
Thank you for your attention!

<table>
<thead>
<tr>
<th></th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wiki</td>
<td><a href="https://ohwr.org/project/wrtd/wiki">https://ohwr.org/project/wrtd/wiki</a></td>
</tr>
<tr>
<td>Forum</td>
<td><a href="https://forums.ohwr.org/c/wrtd">https://forums.ohwr.org/c/wrtd</a></td>
</tr>
<tr>
<td>Docs</td>
<td><a href="https://wrtl.readthedocs.io">https://wrtl.readthedocs.io</a></td>
</tr>
</tbody>
</table>