White Rabbit Applications for Data Acquisition Systems

Dimitris Lampridis

CERN BE-CO
Hardware and Timing section

DAQ Meeting, IN2P3, 01 June 2016

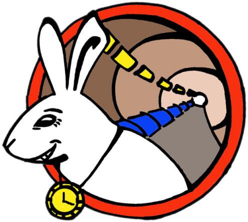
1. White Rabbit
2. WR Demo
3. WR for Distributed DAQ
4. Adding WR to your Design
5. Conclusions

Outline

1. White Rabbit
2. WR Demo
3. WR for Distributed DAQ
4. Adding WR to your Design
5. Conclusions

What is White Rabbit?

- A protocol to synchronize nodes in a large-scale network with sub-ns accuracy
- Open Hardware and Open Software with commercial support
- International collaboration



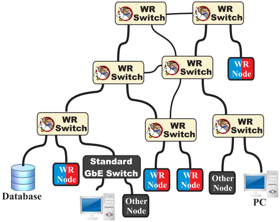
Why we use Open Hardware ?

	Commercial	Non-commercial
Open	**Winning combination. Best of both worlds.**	Whole support burden falls on developers. Not scalable.
Proprietary	Vendor lock-in.	Dedicated non-reusable projects.

- Get a design just the way we want it
- Peer review and design re-use
- Healthier relationship with companies

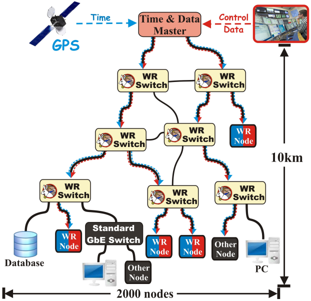
White Rabbit: an *extension* of Ethernet

- Standard Ethernet network
- Ethernet features (VLAN) & protocols (SNMP)



White Rabbit: an *extension* of Ethernet

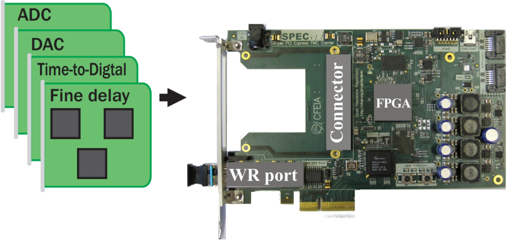
- Standard Ethernet network
- Ethernet features (VLAN) & protocols (SNMP)
- High accuracy synchronization
- Reliable and low-latency Control Data



White Rabbit Switch

- Central element of WR network
- 18 port gigabit Ethernet switch with WR features
- Optical transceivers: up to 10 km, single-mode fiber
- Fully open design, commercially available

White Rabbit Nodes



- Carrier boards in PCI-Express, VME, PXIe
- Equipped with a WR port and FMC connector(s)
- Mezzanines use the WR clock signal and timing interface
- All sources available in the OHWR:

http://www.ohwr.org

White Rabbit technology

Based on

- Gigabit Ethernet over fiber
- IEEE-1588 (PTP) protocol

White Rabbit technology

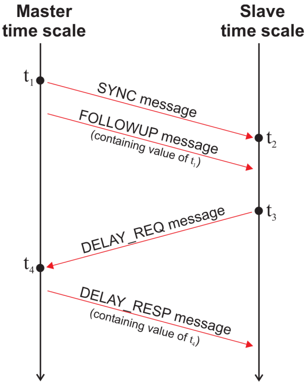
Based on

- Gigabit Ethernet over fiber
- IEEE-1588 (PTP) protocol

Enhanced with

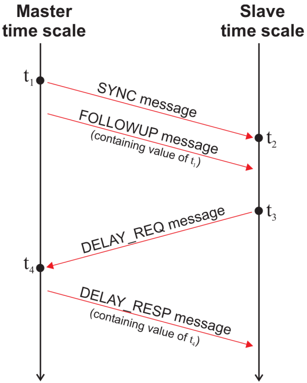
- Layer 1 syntonization
- Digital Dual Mixer Time Difference (DDMTD)
- Link delay model

Precision Time Protocol (IEEE 1588)



- Frame-based synchronization protocol
- Like NTP but in hardware
- Simple calculations:
  - link *delay*$_{ms}$ $\delta_{ms} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$
  - clock *offset*$_{ms}$ $= t_2 - t_1 + \delta_{ms}$

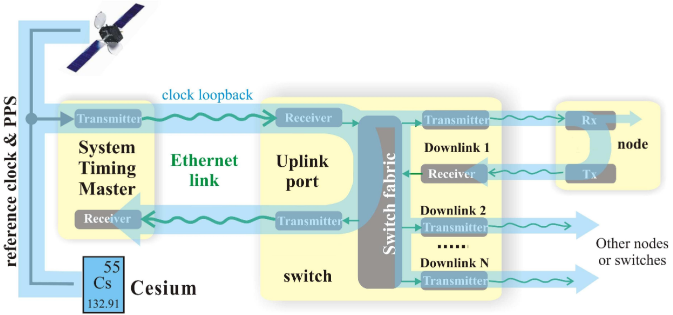
Precision Time Protocol (IEEE 1588)



- Frame-based synchronization protocol
- Like NTP but in hardware
- Simple calculations:
  - link $delay_{ms}$ $\delta_{ms} = \frac{(t_4 - t_1) - (t_3 - t_2)}{2}$
  - clock $offset_{ms} = t_2 - t_1 + \delta_{ms}$
- Can be further improved
  - assumes symmetry of medium
  - all nodes have free-running oscillators
  - frequency drift compensation vs. message exchange traffic

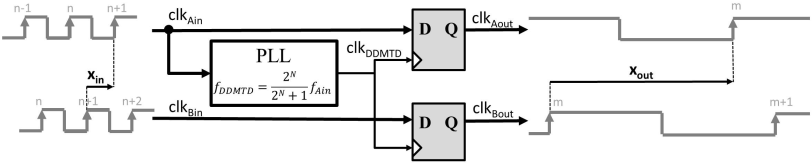
Layer 1 Syntonization

- All network devices use the same physical layer clock.
- Clock is encoded in the Ethernet carrier and recovered by the receiver chip.
- Clock is looped back, phase detection allows sub-ns delay measurement.



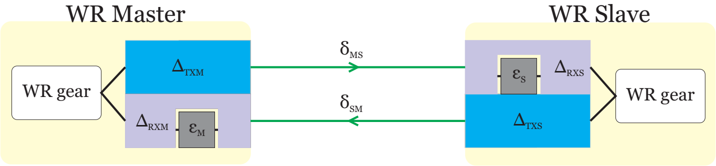
Digital Dual Mixer Time Difference

DDMTD



- Used for precise phase measurements
- Outputs are at much lower frequencies, easier to measure

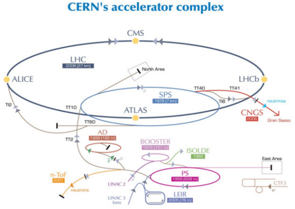
Link delay model

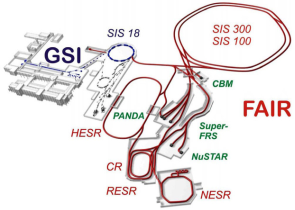


- static hardware delays: Δ_{TXM}, Δ_{RXM}, Δ_{TXS}, Δ_{RXS}
- semi-static hardware delays: ϵ_M, ϵ_S
- fiber asymmetry coefficient: $\alpha = \frac{\delta_{MS} - \delta_{SM}}{\delta_{SM}}$

White Rabbit application examples

- CERN and GSI





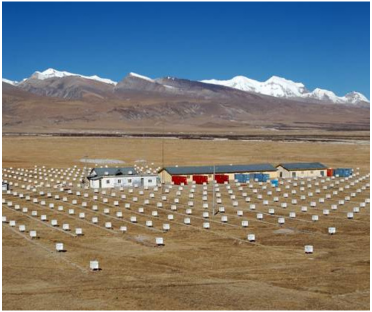
White Rabbit application examples

- CERN and GSI
- HiSCORE: Gamma&Cosmic-Ray experiment

> Institute for Nuclear Research of the Russian Academy of Sciences
> Moscow State University
> Irkutsk State University

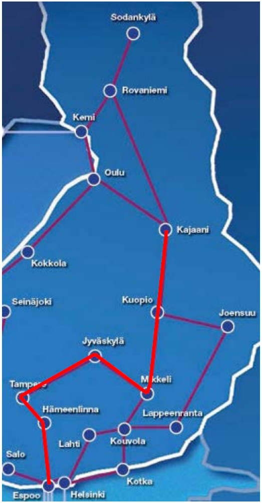
White Rabbit application examples

- CERN and GSI
- HiSCORE: Gamma&Cosmic-Ray experiment
- The Large High Altitude Air Shower Observatory



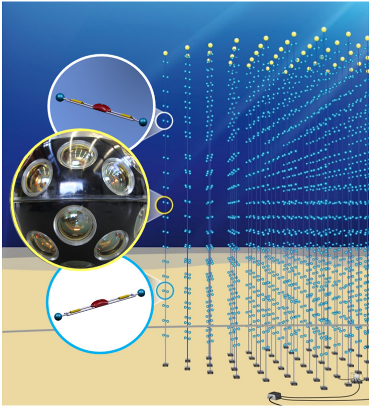
White Rabbit application examples

- CERN and GSI
- HiSCORE: Gamma&Cosmic-Ray experiment
- The Large High Altitude Air Shower Observatory
- MIKES: Centre for metrology and accreditation



White Rabbit application examples

- CERN and GSI
- HiSCORE: Gamma&Cosmic-Ray experiment
- The Large High Altitude Air Shower Observatory
- MIKES: Centre for metrology and accreditation
- KM3NET: European deep-sea research infrastructure



More WR users:

http://www.ohwr.org/projects/white-rabbit/wiki/WRUsers

Outline

1. White Rabbit
2. WR Demo
3. WR for Distributed DAQ
4. Adding WR to your Design
5. Conclusions

WR Demo

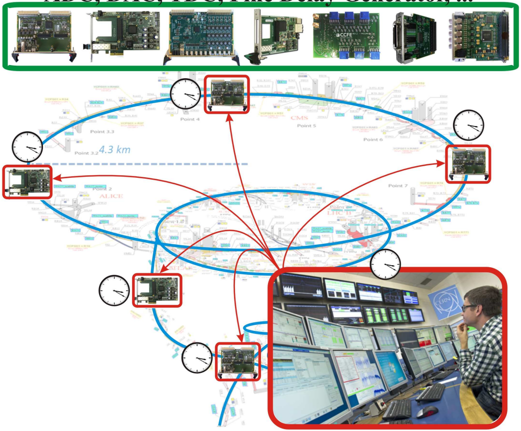
Demo in progress. . .

Outline

1. White Rabbit
2. WR Demo
3. WR for Distributed DAQ
4. Adding WR to your Design
5. Conclusions

Purpose

ADC, DAC, TDC, Fine Delay Generator, ...

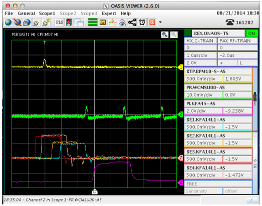


- Provide a communication protocol for distributed instrumentation over WR

Motivation

OASIS: Open Analog Signals Information System

- Distributed oscilloscope
- 1000s of signals
- 100s of triggers
- Unidirectional
- Hard-wired


OASIS VIEWER (2.6.0)
File General Scope1 Scope2 Scope3 Expert Help
PLS
☎ 161202
PSB:EAST1 (4) CPS:MD7 (4)
1
2
7
3
9-16

08/21/2014 18:38

BEX.ONAOS-TS		ON
MX.C-TRAIN	PAX.RF-TRAIN	
0	0	
1.0us/div	-2.0us	
2.0V	+	L
BTP.BPM10-S-AS		
500.0mV/div	1.603V	
PR.WCMSU00-AS		
10.0mV/div	0.0V	
PI.KFA45-AS		
2.0V/div	-9.238V	
BE1.KFA14L1-AS		
500.0mV/div	-1.5V	
BE2.KFA14L1-AS		
500.0mV/div	-1.5V	
BE3.KFA14L1-AS		
500.0mV/div	-1.5V	
BE4.KFA14L1-AS		
500.0mV/div	-1.473V	
FREE		
Sensitivity	offset	

18:35:04 - Channel 2 in Scope 1: PR.WCMSU00-AS

4.3 km
Point 3.3
Point 3.2
Point 4
CMS
Point 5
Point 6
Point 7
Point 2
ALICE
LHC-b
ATLAS

Existing Solution: LXI

Nearest existing solution is **LXI**

- Designed for instrumentation
- Works over Ethernet
- Plug & Play
- Has extensions for synchronisation, timestamping and message exchanging

WRXI

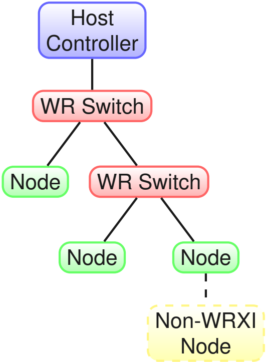
White Rabbit eXtensions for Instrumentation

- A communication protocol for distributed instrumentation over a White Rabbit (WR) network
- Inspired by LXI
- Leverages the high accuracy and precise synchronisation offered by WR
- Augments WR with complex event scheduling, timestamping and real-time message exchanging across the network
- Designed in an application-agnostic way, so that it can be adopted and re-used by others
- Fully open design and implementation

Vision

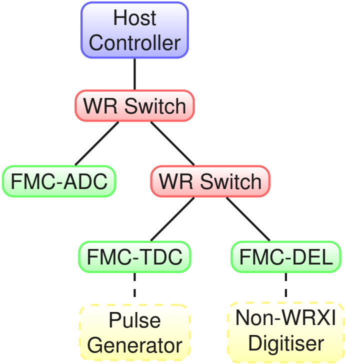
Design a new protocol for instrumentation

- Flexible
- Robust
- Scalable
- Re-usable
- Sustainable
- Fully open
- On top of WR



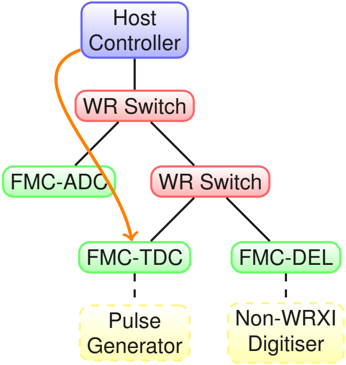
- The network will be built on top of WR switches, with distributed instrumentation nodes, under the supervision and control of a host controller. The host controller can be linked to an external network. Non-WRXI instrumentation can be attached to special nodes (eg. GPIB bridges, external trigger generators, etc.)

WRXI Example 1



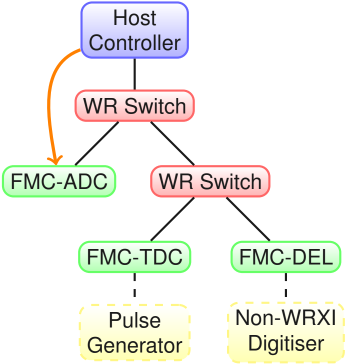
WRXI Example 1

1. FMC-TDC: generate message #1 upon reception of external TTL pulse



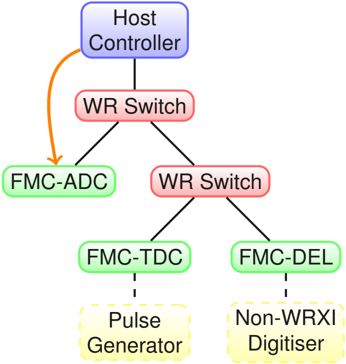
WRXI Example 1

1. FMC-TDC: generate message #1 upon reception of external TTL pulse
2. FMC-ADC: get message #1 and arm



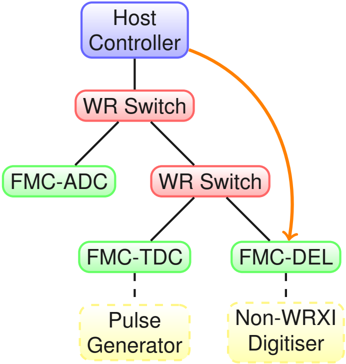
WRXI Example 1

1. FMC-TDC: generate message #1 upon reception of external TTL pulse
2. FMC-ADC: get message #1 and arm
3. FMC-ADC: generate message #2 on trigger



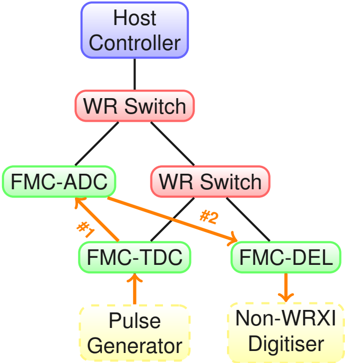
WRXI Example 1

1. FMC-TDC: generate message #1 upon reception of external TTL pulse
2. FMC-ADC: get message #1 and arm
3. FMC-ADC: generate message #2 on trigger
4. FMC-DEL: get message #2 and generate pulse



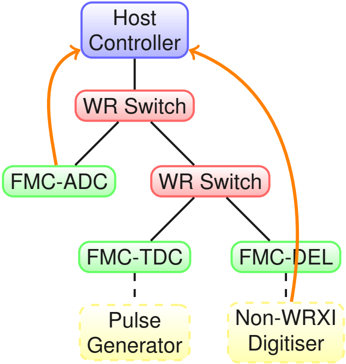
WRXI Example 1

1. FMC-TDC: generate message #1 upon reception of external TTL pulse
2. FMC-ADC: get message #1 and arm
3. FMC-ADC: generate message #2 on trigger
4. FMC-DEL: get message #2 and generate pulse
5. Execute

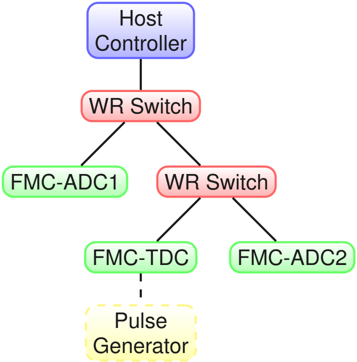


WRXI Example 1

1. FMC-TDC: generate message #1 upon reception of external TTL pulse
2. FMC-ADC: get message #1 and arm
3. FMC-ADC: generate message #2 on trigger
4. FMC-DEL: get message #2 and generate pulse
5. Execute
6. Retrieve data

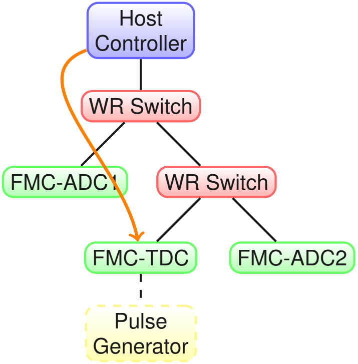


WRXI Example 2



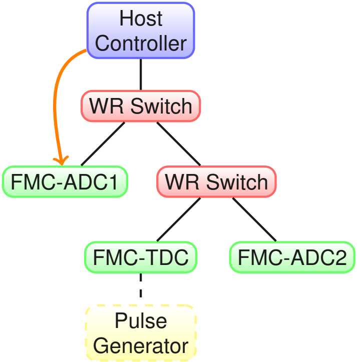
WRXI Example 2

1. FMC-TDC: record pulse, generate message #1



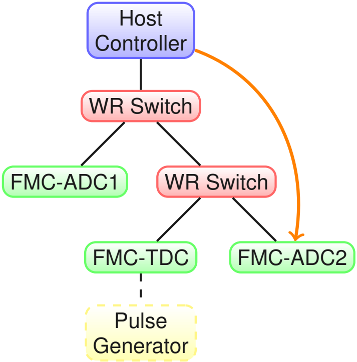
WRXI Example 2

1. FMC-TDC: record pulse, generate message #1
2. FMC-ADC1: in free-running mode, get message #1 and trigger



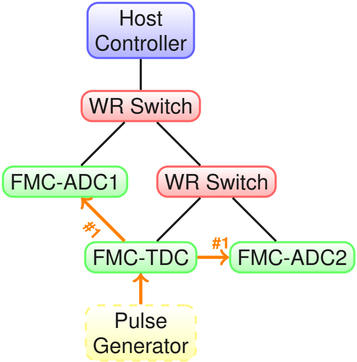
WRXI Example 2

1. FMC-TDC: record pulse, generate message #1
2. FMC-ADC1: in free-running mode, get message #1 and trigger
3. FMC-ADC2: in free-running mode, get message #1 and trigger



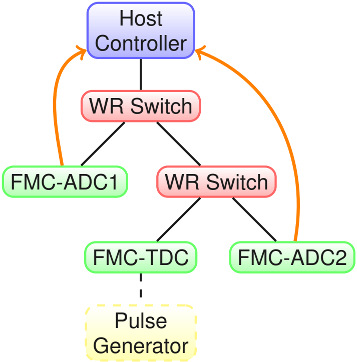
WRXI Example 2

1. FMC-TDC: record pulse, generate message #1
2. FMC-ADC1: in free-running mode, get message #1 and trigger
3. FMC-ADC2: in free-running mode, get message #1 and trigger
4. Execute



WRXI Example 2

1. FMC-TDC: record pulse, generate message #1
2. FMC-ADC1: in free-running mode, get message #1 and trigger
3. FMC-ADC2: in free-running mode, get message #1 and trigger
4. Execute
5. Rewind and retrieve data



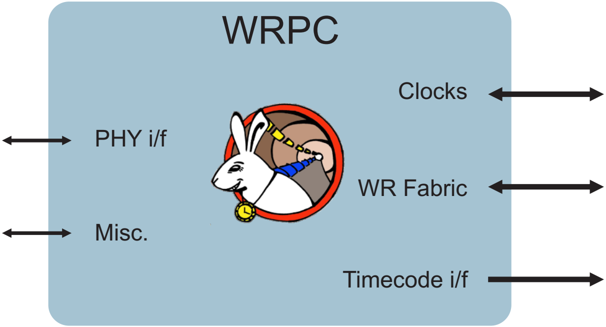
Outline

1. White Rabbit
2. WR Demo
3. WR for Distributed DAQ
4. Adding WR to your Design
5. Conclusions

Many possibilities

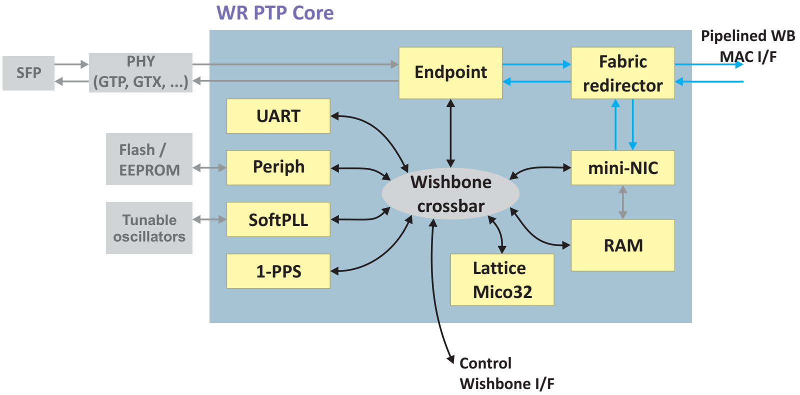
- Make use of one of the provided carriers and selection of mezzanines
- Include an HDL core in your design
- Use a standalone WR node implementation

WR PTP Core - overview



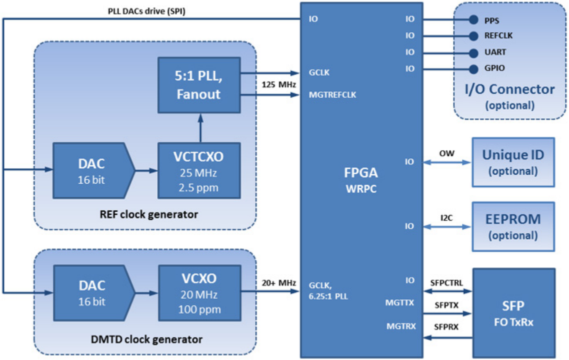
- HDL core with soft CPU
- Ethernet MAC with WR features
- WR implementation for the nodes

WR PTP Core - inside

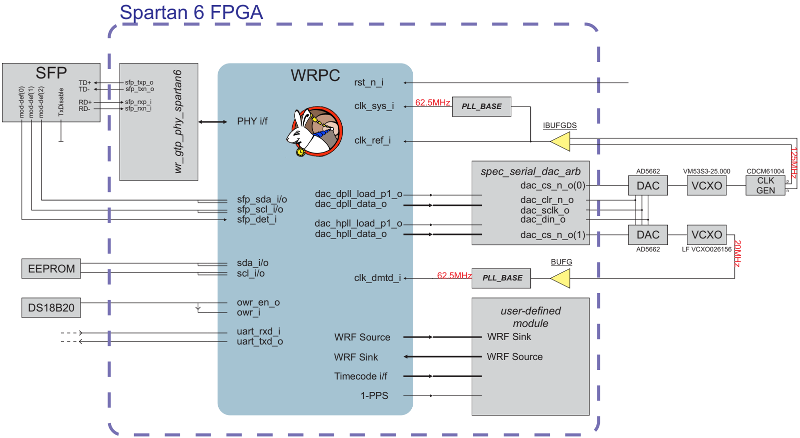


WR PTP Core - clocks

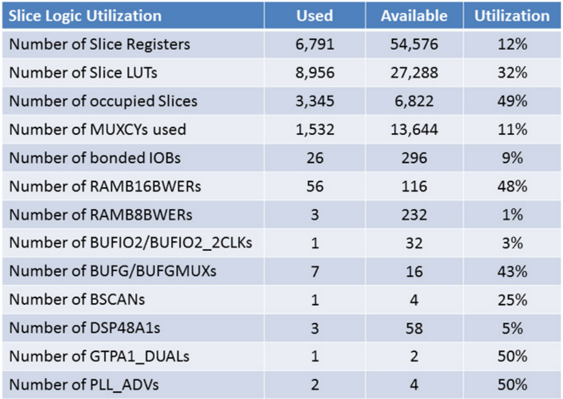
- 125 MHz reference clock
- 62.5 MHz DDMTD clock
- system clock ($\leq$ ref. clock)
- aux clocks



WR PTP Core - how to integrate



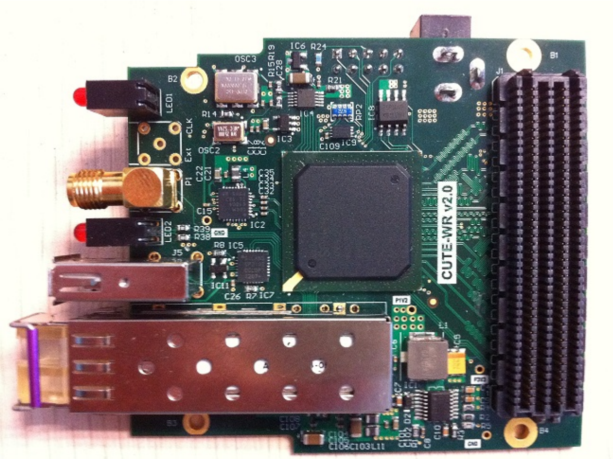
WR PTP Core - resource utilization



Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	6,791	54,576	12%
Number of Slice LUTs	8,956	27,288	32%
Number of occupied Slices	3,345	6,822	49%
Number of MUXCYs used	1,532	13,644	11%
Number of bonded IOBs	26	296	9%
Number of RAMB16BWERs	56	116	48%
Number of RAMB8BWERs	3	232	1%
Number of BUFIO2/BUFIO2_2CLKs	1	32	3%
Number of BUFG/BUFGMUXs	7	16	43%
Number of BSCANs	1	4	25%
Number of DSP48A1s	3	58	5%
Number of GTPA1_DUALs	1	2	50%
Number of PLL_ADVs	2	4	50%

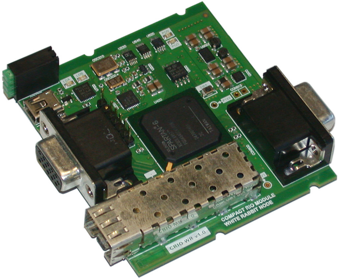
- *Xilinx Spartan-6, XC6SLX45T-3FGG484*

Standalone WR node 1



http://www.ohwr.org/projects/cute-wr/wiki

Standalone WR node 2



http://www.ohwr.org/projects/crio-wr/wiki

Outline

1. White Rabbit
2. WR Demo
3. WR for Distributed DAQ
4. Adding WR to your Design
5. Conclusions

Summary

- Open (H/W & S/W)

Summary

- Open (H/W & S/W)
- Commercial support

Summary

- Open (H/W & S/W)
- Commercial support
- More applications than ever expected

Summary

- Open (H/W & S/W)
- Commercial support
- More applications than ever expected
- A versatile solution for general control and data acquisition

Summary

- Open (H/W & S/W)
- Commercial support
- More applications than ever expected
- A versatile solution for general control and data acquisition
- Standard-compatible and standard-extending

Summary

- Open (H/W & S/W)
- Commercial support
- More applications than ever expected
- A versatile solution for general control and data acquisition
- Standard-compatible and standard-extending
- Active participation in IEEE1588 revision process

Join the development!

http://www.ohwr.org/projects/white-rabbit/wiki