

White Rabbit Time Distribution System and Common Platform for Machine Instrumentation

Plus some reflections on Open Hardware

J. Serrano

BE-CO Hardware and Timing section
CERN, Geneva, Switzerland

Berkeley Lab, 26 February 2010

Outline

- 1 Requirements
- 2 Common platform for machine instrumentation
 - Introduction
 - The FMC standard
 - CERN's implementation
- 3 White Rabbit
 - Overview
 - Technical concepts
 - Synchronous Ethernet
 - PTP Protocol
 - The WR protocol
 - Work so far
- 4 Open Hardware

Outline

- 1 Requirements
- 2 Common platform for machine instrumentation
 - Introduction
 - The FMC standard
 - CERN's implementation
- 3 White Rabbit
 - Overview
 - Technical concepts
 - Synchronous Ethernet
 - PTP Protocol
 - The WR protocol
 - Work so far
- 4 Open Hardware

Desirable HW features of a distributed control system

1/2

The good things of custom HW

- Function is exactly what you need.
- Can change easily if you find a bug. Or have it changed!
- Peer review. Potential for really good designs.
- Not tied to a single company (you never know).

The good things of commercial HW

- Built and tested by someone else.
- Supported by someone else.
- Guaranteed.

Desirable HW features of a distributed control system

2/2

Modular

- Re-use components easily.
- Have different people in an organization do what they do best.

Interconnect!

- Allows to build distributed systems easily.
- Based on communication standards.
- Good sync capabilities. Transparent common notion of time.

Outline

- 1 Requirements
- 2 **Common platform for machine instrumentation**
 - Introduction
 - The FMC standard
 - CERN's implementation
- 3 White Rabbit
 - Overview
 - Technical concepts
 - Synchronous Ethernet
 - PTP Protocol
 - The WR protocol
 - Work so far
- 4 Open Hardware

Background

- CERN's BE Controls group supports a kit of standard hardware modules.
- Support includes stocks management, help in debugging and low level software:
 - Linux Device Drivers.
 - C/C++ libraries with usage examples.
 - Test programs for drivers and libraries.
- With the injectors renovation project, supported platforms will include PCI and PCIe in addition to VME.
- A carrier/mezzanine strategy has been adopted.

Advantages of the carrier/mezzanine approach

Re-use

One mezzanine can be used in VME, PCI and PCIe carriers.

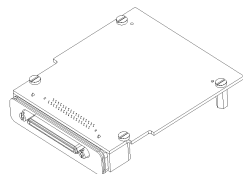
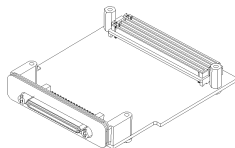
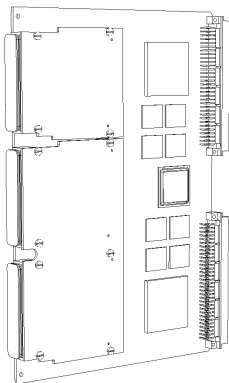
Reactivity

No need to place and route a complex FPGA PCB for every new user need.

Rational split of work

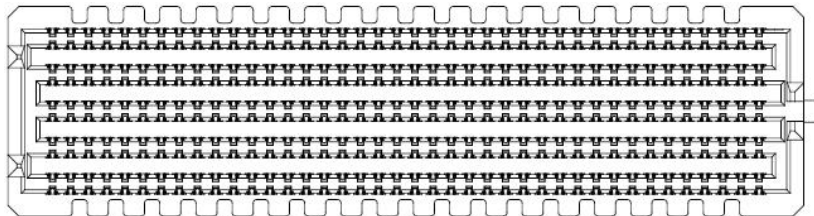
Controls can design the carrier, Instrumentation an ADC mezzanine, RF a DDS one, etc.

Outline



Courtesy of VITA: <http://www.vita.com/fmc.html>

Connectors



- Ball Grid Array (BGA) characterized for high bandwidth applications.
- Low Pin Count (LPC) and High Pin Count (HPC) variants with 160 and 400 contacts respectively.

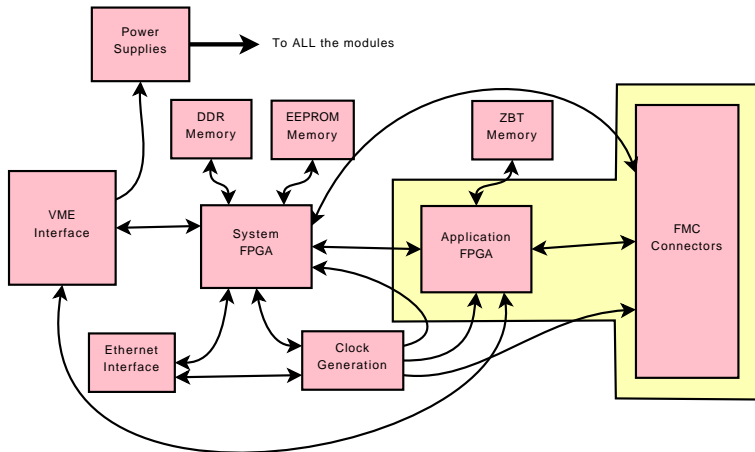
Physical Dimensions

- Small dimensions for thermal reasons.
- Keep all digital circuitry in the carrier.

Agnosticism

- Pin function, sense – input or output – and electrical standard are defined at FPGA configuration time.
- Carrier reads FMC identity through an I2C serial bus and configures the FPGA accordingly.

Carrier design



Ongoing developments

1/2

Carriers

- VME with two single-width (one double-width) slots.
- PCIe with one single-width slot.

Ongoing developments

2/2

Mezzanines

- Four-channel 100 MS/s 14-bit ADC with oscilloscope-type analog front end.
- Eight-channel 100 kS/s 16-bit sampler.
- Simple parallel digital I/O.
- Under discussion:
 - Four-channel 16-bit 120 MS/s sampler.
 - Four-channel 16-bit 120 MS/s DAC.
 - Clock generator based on 1 GS/s 32-bit DDS.
 - One-channel 24-bit 2 MS/s sampler.

A critique of FMC

What we can say after 6 months of experience

Pros

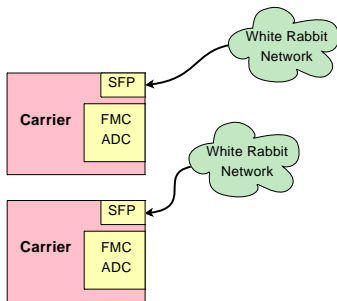
- It's a standard! Somebody thought hard about mechanics, dissipation, connectors...
- The only agnostic standard for FPGA mezzanines.
- Auto-discovery of card type very useful.

Cons

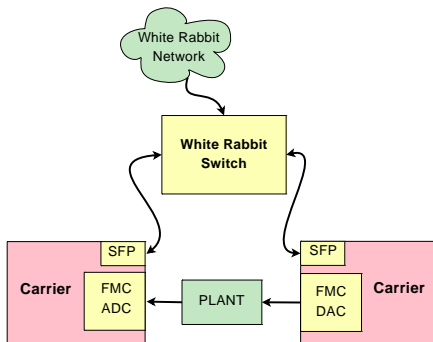
- Real estate a potential problem in some applications.
- Little consideration given to Carrier-to-mezzanine clocks.

Overall a good choice in our opinion.

Use cases



Distributed oscilloscope



Distributed feedback system

Gateway and software

Some initial ideas

Gateway

- Wishbone-based.
- Try to automate repetitive code through scripts.
- Auto-discovery of cores by Linux kernel would be nice.

Software (very preliminary ideas!)

- Kernel modules, one per core.
- Interconnected by a driver representing the whole board.
- Integration into official kernel desirable.

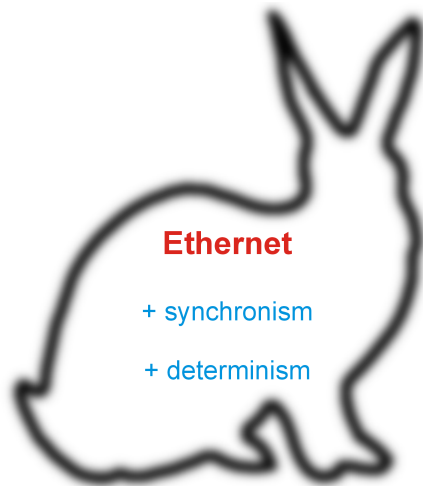
Part 1 Summary

- The first **agnostic standard** to interface mezzanines and FPGAs.
- CERN's BE-CO group will adopt it to **improve support** of hardware and **reduce maintenance** costs.
- Combined with Open Hardware paradigm and collaborations, it can **reduce duplication** and **improve design quality**.
- Outlook
 - Finish carriers (and some mezzanines) designs before end 2010.
 - Start collaboration with companies for series production of carriers and FMCs.

Outline

- 1 Requirements
- 2 Common platform for machine instrumentation
 - Introduction
 - The FMC standard
 - CERN's implementation
- 3 White Rabbit**
 - Overview
 - Technical concepts
 - Synchronous Ethernet
 - PTP Protocol
 - The WR protocol
 - Work so far
- 4 Open Hardware

What is White Rabbit?



Design goals

Scalability

Up to 2000 nodes.

Range

10 km fiber links.

Precision

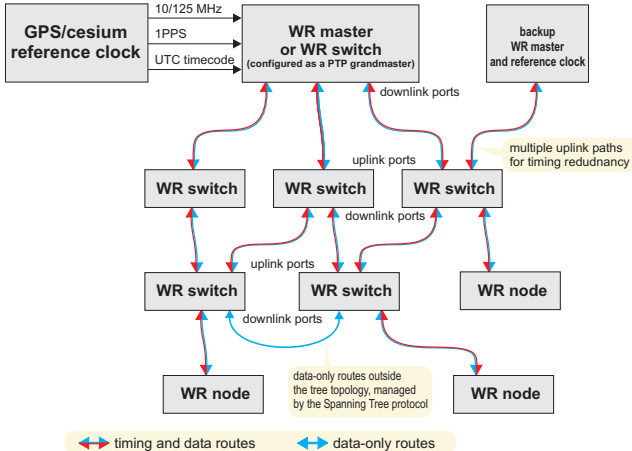
1 ns time synchronization accuracy, 20 ps jitter.

What is White Rabbit?

An **extension** to **Ethernet** which provides:

- **Synchronous mode** (Sync-E) - common clock for physical layer in entire network, allowing for precise time and frequency transfer.
- **Deterministic routing** latency - a guarantee that packet transmission delay between two stations will never exceed a certain boundary.

Network topology



Technical concepts in White Rabbit

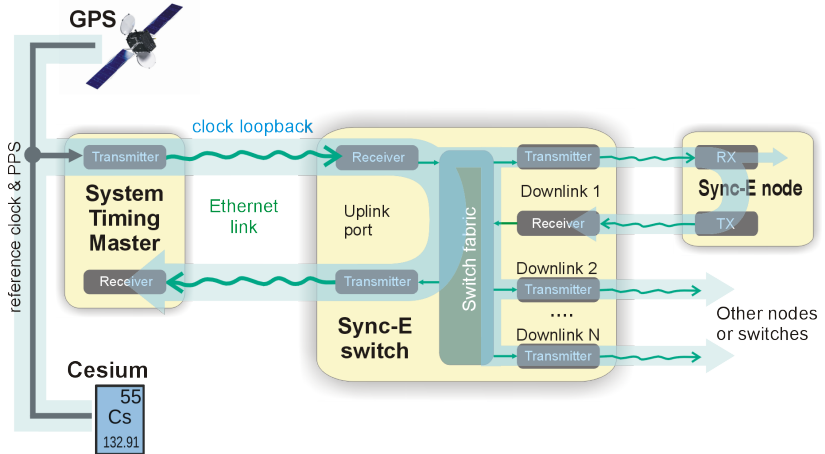
- Synchronous Ethernet
- Hardware-assisted PTP (IEEE1588 - Precision Time Protocol)
- Packet preemption and deterministic protocol

Synchronous Ethernet

Common clock for the entire network

- All network nodes use the same physical layer clock, generated by the System Timing Master
- Clock is encoded in the Ethernet carrier and recovered by the PLL in the PHY.

Synchronous Ethernet

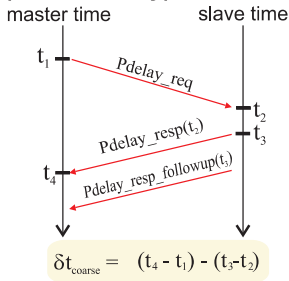


PTP Protocol (IEEE1588)

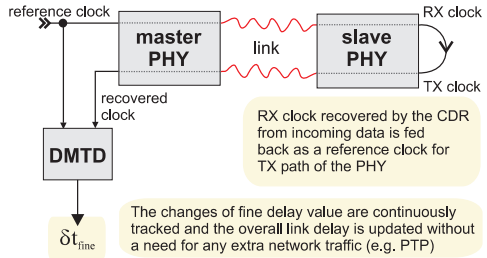
PTP

Synchronizes local clock with the master clock by measuring and compensating the delay introduced by the link.

PTPv2 protocol (coarse delay)



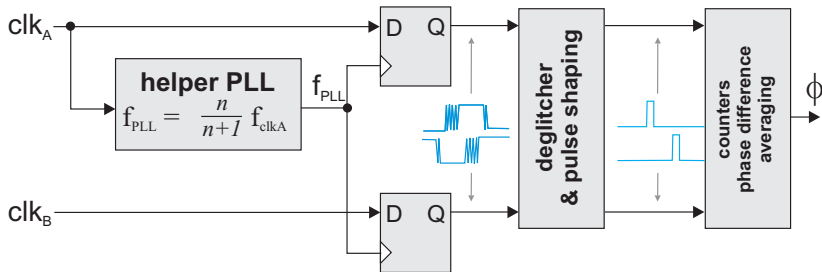
digital DMTD (fine delay)



Enhanced PTP

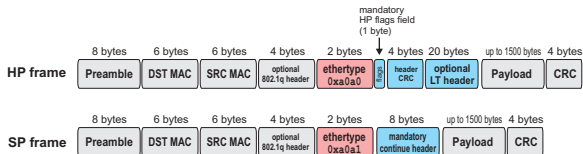
- Monitor phase of bounced-back clock continuously.
- Non-invasive: piggy backs on any type of traffic, including an idle link.
- Every 125 MHz tick is put to good use: performance is equivalent to PTP with messages exchanged every 8 ns.
- Compatibility: works with any PTP-enabled network, but with superior performance in WR mode.

Digital Dual Mixer Time Domain (DMTD) phase detector



- Fully digital, so fully linear.
- In a loop, it becomes a linear phase shifter.

The WR protocol

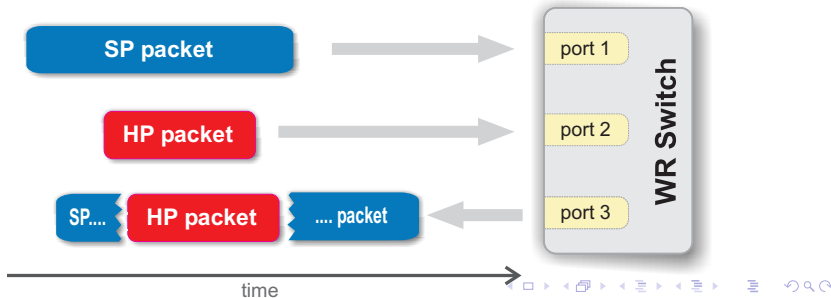


- Traffic divided into High Priority (HP) packets and Standard Priority (SP) packets.
- HP packets use a special value in the Ethertype field of the frame.
- Quality of Service (QoS) in the 802.1Q VLAN standard does this and more ⇒ will study full compliance in the future.
- HP packets can preempt other types of packets “on-the-fly”.

The WR protocol

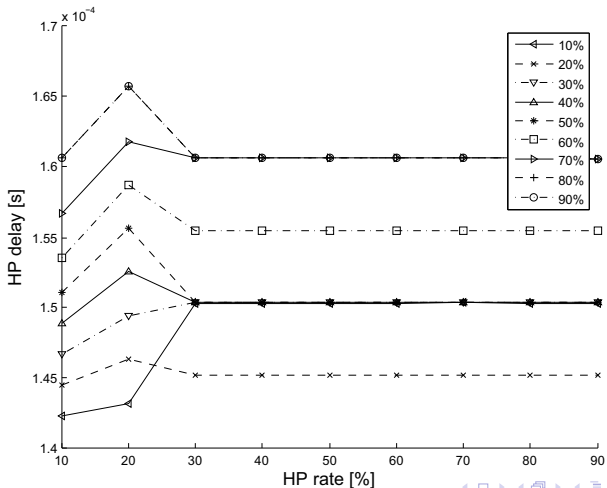
Preemption mechanism

When a HP packet arrives at the switch, SP packet currently being routed is terminated so the HP packet can be sent out with minimal latency. The remaining part of terminated SP packet is sent later.

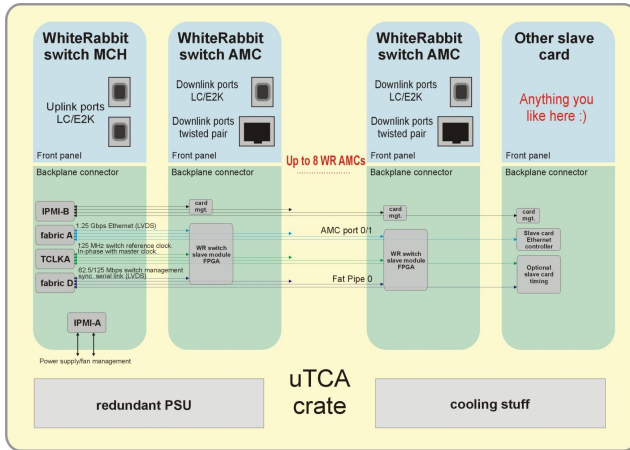


Simulation results

High throughput and determinism achieved simultaneously

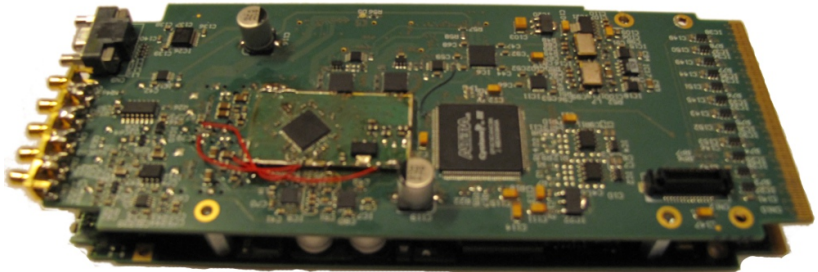


Switch design



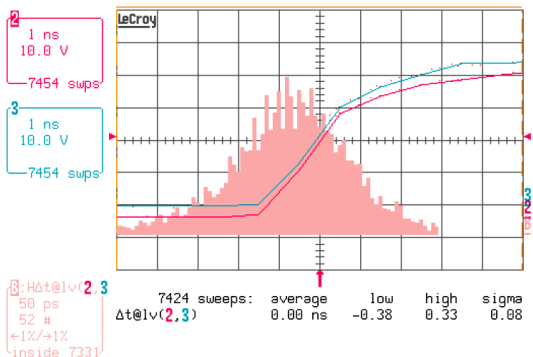
The switch prototype

Proof of concept design



Tests

80 ps accuracy over 5km fiber link



- 1 ns
- 1 trig only
- 2 1 V DC $\frac{10}{10}$
- 3 1 V DC $\frac{10}{10}$
- 4 trig only



2 DC 24.8 V
no limits set

1 GS/s

NORMAL ⋮ ▶ ⋮ ↺ ↻

Part 2 Summary

- A data link fulfilling all our needs in **synchronization** and determinism.
- Fully based on **standards** like Synchronous Ethernet and PTP.
- A successful **collaboration** including institutes and companies.
- Outlook
 - Establish a community of developers.
 - Deliver working prototypes by the end of 2010.

Outline

- 1 Requirements
- 2 Common platform for machine instrumentation
 - Introduction
 - The FMC standard
 - CERN's implementation
- 3 White Rabbit
 - Overview
 - Technical concepts
 - Synchronous Ethernet
 - PTP Protocol
 - The WR protocol
 - Work so far
- 4 Open Hardware

Open Hardware: our definition

Publish everything needed to review

Specifications, discussions, schematics and layouts in some human-readable format, HDL, etc. Publish universally, no NDAs.

Publish everything needed to modify

Schematics and PCB layout files for your favorite EDA tool. Unfortunately the best ones are not free...

Publish everything needed to produce

Manufacturing files, bill of materials, etc.

Advantages

Peer review

Get your design reviewed by experts all around the world, including companies!

Design re-use

How many people are designing a 100 MS/s ADC independently, making the same – or different – mistakes?

Healthier relationship with companies

No vendor-locked situations. Companies selected solely on the basis of technical excellence, good support and price.

Role of companies

Design partners

Pay a company specialized in a given topic to design a specific card with/for you.

Commercial partners

Buy the cards you designed from a company that will take the charge of manufacturing, testing, managing stocks and providing support.

Licensing

A quick landscape tour

HW is not like SW

- Copyright protects the expression of an idea, not the idea itself.
- For a schematic (and even HDL), GPL is easily bypassed.

Options

- OHL (viral). If you take my design and use it, you promise not to sue me for patent infringement.
- BOHL (viral). Design files are not released.
- MIT/BSD (non-viral). Do what you like, don't blame me in case of problems.

Licensing

Our thoughts so far

LGPL for HDL

- It's very easy to turn a "used in" into a "connected to" situation in HDL, so GPL would not help.
- We do want to be informed and profit if our cores are improved.

MIT/BSD-style for the rest

- Not clear how OHL, BOHL and others would perform in court. And don't want to find out!
- Viral licenses scare some of our potential commercial partners. Could do more harm than good.

Mad patent disease and patent trolls

See <http://www.eetimes.com/showArticle.jhtml;?articleID=216600017>

“In this climate, many fear being charged with willfully infringing patents or omitting prior art in patent applications, a charge known as inequitable conduct. So Intel and other companies have put strict procedures in place to control which patents its engineers can read.”

Opening up your designs does make you more vulnerable to this disease.

Open Hardware Repository: <http://www.ohwr.org>

A very useful tool

A web-based collaborative tool for electronics designers.

Made itself of open software

- Redmine for wiki and task/issue management.
- Sympa mailing list manager.
- SVN/GIT for version management (integrated in Redmine).

Other possible uses

- Traceability for Technology Transfer departments.
- Prove prior art with UTC time stamps in SVN, GIT, wiki...

Conclusions

Open Hardware looks like a good idea so far

- We can get the best of the custom and COTS worlds.
- We are learning a lot, even electronics! ;)
- Definitely more fun than closed HW.

Some things not completely clear yet

- Legal framework.
- We still need a clear collaboration model with companies.

First HW due end of 2010, stay tuned!