New hardware solutions for the control of LHC injectors

Plus some reflections on Open Hardware

J. Serrano

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

Soleil, 30 June 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

J. Serrano

New HW solutions for LHC injectors
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

J. Serrano
New HW solutions for LHC injectors
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

- Designed, built and tested by someone else (resource gain).
- Immediately available.
- Hardware proven by many users in many different applications.
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

J. Serrano
New HW solutions for LHC injectors
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.
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

- Power Supplies
- DDR Memory
- EEPROM Memory
- ZBT Memory
- Ethernet Interface
- Clock Generation
- System FPGA
- Application FPGA
- VME Interface
- FMC Connectors

To ALL the modules
Ongoing developments

WR-enabled carriers

- VME with two single-width (one double-width) slots.
- PCIe with one single-width slot.
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:
  - Fine delay generator.
  - Time-to-Digital Converter (TDC).
  - Four-channel 100 MS/s DAC.
  - 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**

**White Rabbit**
- Switch
- SFP
- Carrier
- FMC
- ADC

**White Rabbit**
- Network

**Distributed oscilloscope**

**Distributed feedback system**

**J. Serrano**
**New HW solutions for LHC injectors**
Gateware and software
Some initial ideas

Gateware
- Internal bus is Wishbone-based (open standard with IP cores available).
- Try to automate repetitive code through scripts.
- Auto-discovery of Wishbone cores by Linux kernel would be nice.

Software (very preliminary ideas!)
- Define Wishbone as a bus in Linux.
- Write Linux modules and interconnect them 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
   - New HW solutions for LHC injectors
What is White Rabbit?

- Ethernet
  - + synchronism
  - + determinism
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

10/125 MHz
1PPS
UTC timecode

GPS/cesium reference clock

WR master or WR switch
(configured as a PTP grandmaster)

backup WR master and reference clock

downlink ports

uplink ports

multiple uplink paths for timing redundancy

data-only routes outside the tree topology, managed by the Spanning Tree protocol

uplink ports

downlink ports

timing and data routes
data-only routes

J. Serrano
New HW solutions for LHC injectors
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.
Requirements
Common platform for machine instrumentation
White Rabbit
Open Hardware

Overview
Technical concepts
Work so far

Synchronous Ethernet

System Timing Master
GPS
Cesium

Sync-E switch

Switch fabric

Uplink port

Transmitter
Receiver

Downlink 1
Receiver
Transmitter

Downlink 2
Transmitter

Downlink N
Transmitter

Sync-E node

Other nodes or switches

Reference clock & PPS

Ethernet link

clock loopback

J. Serrano
New HW solutions for LHC injectors
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)

master time

slave time

\[ \delta t_{\text{coarse}} = (t_4 - t_1) - (t_3 - t_2) \]

digital DMTD (fine delay)

reference clock

master PHY

link

slave PHY

recovered clock

DMTD

RX clock recovered by the CDR from incoming data is fed back as a reference clock for TX path of the PHY

The changes of fine delay value are continuously tracked and the overall link delay is updated without a need for any extra network traffic (e.g. PTP)

J. Serrano

New HW solutions for LHC injectors
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.
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

<table>
<thead>
<tr>
<th>HP frame</th>
<th>8 bytes</th>
<th>6 bytes</th>
<th>6 bytes</th>
<th>4 bytes</th>
<th>2 bytes</th>
<th>4 bytes</th>
<th>20 bytes</th>
<th>up to 1500 bytes</th>
<th>4 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>DST MAC</td>
<td>SRC MAC</td>
<td>optional 802.1q header</td>
<td>ethertype 0xa0a0</td>
<td>header CRC</td>
<td>LT header</td>
<td>Payload</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SP frame</th>
<th>8 bytes</th>
<th>6 bytes</th>
<th>6 bytes</th>
<th>4 bytes</th>
<th>2 bytes</th>
<th>8 bytes</th>
<th>up to 1500 bytes</th>
<th>4 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>DST MAC</td>
<td>SRC MAC</td>
<td>optional 802.1q header</td>
<td>ethertype 0xa0a1</td>
<td>mandatory continue header</td>
<td>Payload</td>
<td>CRC</td>
<td></td>
</tr>
</tbody>
</table>
The WR protocol

Preemption mechanism (preliminary ideas)

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.
Switch design

White Rabbit switch MCH
- Uplink ports LC/E2K
- Front panel
- Backplane connector
- IPMI-B
- fabric A
- TCLKA
- fabric D
- Power supply/fan management
- uTCA crate
- Redundant PSU

White Rabbit switch AMC
- Downlink ports LC/E2K
- Downlink ports twisted pair
- Front panel
- Backplane connector
- AMC port 0/1
- WR switch slave module FPGA
- Fat Pipe 0

White Rabbit switch AMC
- Downlink ports LC/E2K
- Downlink ports twisted pair
- Front panel
- Backplane connector
- WR switch slave module FPGA

Other slave card
- Anything you like here :)
- Backplane connector
- Optional slave card timing
- Slave card Ethernet controller

cooling stuff

Technical concepts
Work so far

New HW solutions for LHC injectors
The switch prototype
Proof of concept design
Tests
80 ps accuracy over 5km fiber link
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 neither free nor 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.
Getting organized

“*It’s all about interfaces*, *Bob Dalesio*

Your piece of HW can speak to others if you can agree on a set of interfaces. Examples (currently in OHR) include Ethernet, VME, PCIe, FMC...

**Design compromises**

The price to pay for sharing (and saving time and money) is to choose sub-optimal technical solutions from time to time. We did not choose to write an OS more suited to our needs than Linux, did we?
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.
Some business model examples for commercial partners

**IBM-style**
Become part of a larger OH team, fully respecting OH practice. Sell full systems based on OH kit.

**Red Hat-style (kind of)**
Sell manufacture, test and support of individual boards along with a guarantee. Participate in design if needed.

**Oracle-style**
Support OH kit and build a closed solution on top with added value.
Licensing
A quick landscape tour

Hardware is not like software

- 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.
Requirements
Common platform for machine instrumentation
White Rabbit
Open Hardware

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.
What about free riders?

Free riders are fine
People and companies who take open designs and do not contribute anything back do not pose a problem to us.

But what about *mean* free riders?
If somebody takes OH and uses it to build a closed solution for a profit, that is fine as well, but we would not be clients.
“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, work in progress with CERN’s KTT group.
- We still need a clear collaboration model with companies.

First HW due end of 2010, stay tuned!

J. Serrano
New HW solutions for LHC injectors