Trigger and RF distribution using White Rabbit
Outline

- A very quick introduction to White Rabbit
- Trigger Distribution system
- Radio Frequency Distribution system
- Status & outlook
White Rabbit – A quick recap

- **Based on Gigabit Ethernet**
  - > 2000 nodes in a network
  - > 10 km distance (single mode fiber)
  - All nodes synchronized to less than 1 ns
  - With jitter of < 20 ps
  - Deterministic data transfers

- **Data and timing in the same network**

- **Using standards:**
  - IEEE1588 (Precision time Protocol)
  - Synchronous Ethernet

- **WR PTP Core: embedded WR stack**
  - Single VHDL module
  - Provides 125 MHz, PPS and TAI time
  - ... and Ethernet MAC functionality

T. Włostowski
Trigger and RF distribution using White Rabbit
White Rabbit – A quick recap

• Based on Gigabit Ethernet
  • > 2000 nodes in a network
  • > 10 km distance (single mode fiber)
  • All nodes synchronized to less than 1 ns
  • With jitter of < 20 ps
  • Deterministic data transfers

• Data and timing in the same network

• Using standards:
  • IEEE1588 (Precision time Protocol)
  • Synchronous Ethernet

• WR PTP Core: embedded WR stack
  • Single VHDL module
  • Provides 125 MHz, PPS and TAI time
  • ... and Ethernet MAC functionality
The LHC Instability Studies Project

- Instruments detect the onset of a beam instability.
- Generate a trigger.
- Distribute the trigger to other instruments and acquire a massive amount of data for offline study.
- Exchange triggers between any pair of nodes.
Trigger Distribution - Background

The LHC Instability Studies Project

- Instruments detect the onset of a beam instability.
- Generate a trigger.
- Distribute the trigger to other instruments and acquire a massive amount of data for offline study.
- Exchange triggers between any pair of nodes.
A trigger pulse comes in and gets timestamped.

The timestamp is broadcast in a UDP packet with metadata identifying the trigger source.

Any number of devices can subscribe to the trigger and reproduce it with a fixed delay thanks to network-wide synchronization provided by White Rabbit.
Trigger Distribution – Implementation

- Based on the CERN FMC Kit
  - SVEC Carrier (VME64x)
  - Input: FMC TDC
  - Outputs: FMC Fine Delay

- FPGA: the *Mock Turtle* core
  - Based on deterministic CPU cores
  - One core takes care of the inputs, the other – of the outputs
  - No specialized HDL needed (reused standard TDC & Fine Delay cores)

- Software
  - Real-time CPU cores programmed in bare metal C
  - Generic Linux device driver
  - Application-specific user space libraries and front end software.
Trigger Distribution – Features

- **Accuracy**: < 1 ns network-wide, jitter < 100 ps rms (largest jitter contribution from the TDC).
- **Throughput**: 1 trigger every 80 µs per each input/output (capable of distributing the LHC revolution frequency as a series of pulses).
- **Worst case latency**: < 100 µs + fiber
- **Single shot and continuous** triggering modes.
- **Delay** configurable independently for each input/output.
- Each output can subscribe to up to 128 triggers.
- **Conditional triggering**: a trigger arms an output to produce a pulse when another trigger comes.
- **Logging** of each sent, executed and missed trigger.
- **Standard network diagnostic tools** (Wireshark).
RF distribution – Introduction

- Direct Digital Synthesis: standard method to generate RF in accelerators.
  - RF is generated centrally.
  - Distribution using traditional, coax cabling or fibers.
  - Cabling is expensive. DDS chips are cheap.

- As the DDS output frequency and phase depend on:
  - Control word (tune) value
  - Reference clock frequency and phase

- The synthesizers set up with the same control word and same reference clock will produce identical RF signals.
RF distribution – Idea

• All nodes have the same reference frequency and time.
• Master phase locks its DDS to the RF input.
• Broadcast the DDS control words, including a TAI timestamp.
• All receivers update their DDSes with the received control word at the same moment (+ some fixed delay)
• Thanks to WR synchronization, we get identical RF signals at all nodes.
RF distribution – Implementation

- Hardware based on the SVEC carrier and the DDS600M FMC
- HDL implemented with *Mock Turtle* (all DSP and networking in software)
- Additional features:
  - RF Counter synchronization
  - Pulse generation and time stamping using the RF clock
  - Simple timing event distribution (proof of concept)
RF Distribution – Performance

- **Accuracy**: < 1 ns
- **Jitter**: < 20 ps rms
  - Carrier: 44 MHz (RF @ 352 MHz), divided by 8
  - 2.6 ps rms for 1 kHz – 1 MHz
  - 16 ps rms for 10 Hz – 20 MHz
  - Significant high frequency noise contribution from the DDS
  - Additional PLL to clean up the synthesized clock
- **Tuning bandwidth**: ~ 1 kHz
- **Latency**: 200 µs
- **RF Range**: 10 – 500 MHz
Status & outlook

- **Trigger Distribution**: production
  - Operational in the LHC (8 crates)
  - **2017**: new trigger system for distributed signal acquisition at CERN

- **RF Distribution**: advanced prototype
  - In phase RF recovery and counter sync working
  - Event distribution demonstrated
  - Jitter optimization ongoing
  - **2016**: beam-synchronous data acquisition in SPS
  - **2016**: proof of concept timing for Synchrotron Light Sources

- Both designs done using **reusable** hardware, gateware and software.

Sources available
at the Open Hardware Repository: ohwr.org
We invite you to our presentation on development of hard-real time systems using FPGAs and soft CPU cores.

Thursday, 9:30, Hardware Track (2nd floor)