40MHz trigger-less readout of the CMS Drift Tube muon detector

Matteo Migliorini[1], Andrea Triossi[1], Jacopo Pazzini[1], Marco Zanetti[1]
on behalf of the CMS Muon Group

[1]University and INFN Padova

Introduction

Electronics of the CMS Drift Tubes detector (DT) will be replaced for the High Luminosity phase of LHC [1]

- New on-detector electronics for DT (**OBDT**) based on radiation-tolerant FPGAs
- Responsible for the time digitization of the DT signals (TDC)
- Up to 240 channels per board with a least significant bit of the TDC of 25/32 ns
- Data streamed to backend boards in the service cavern via lpGBT link

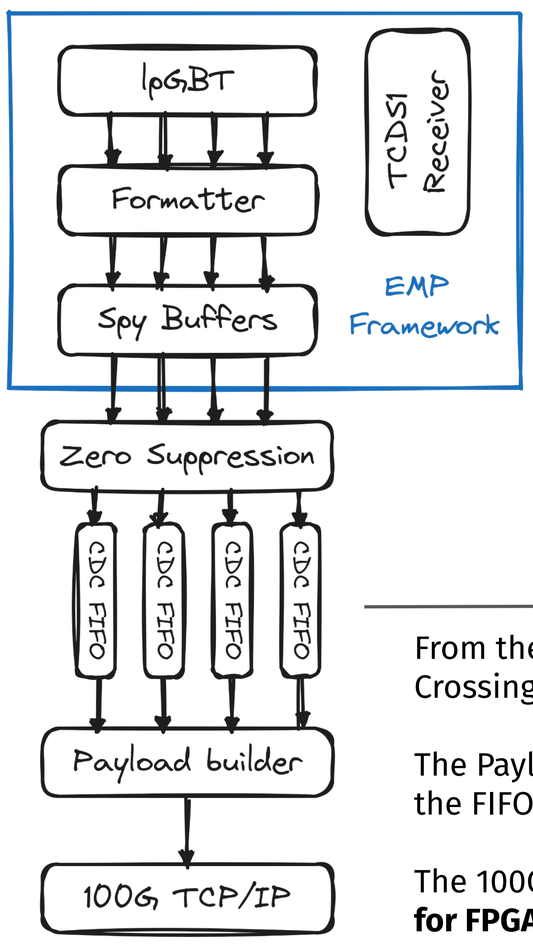
One sector of the DTs, 4 chambers, has been equipped with final prototypes of the OBDT working in parallel with the legacy electronic system (**Slice Test**)

- Used to validate the OBDTs in situ

In this context, a system capable of reading **triggerless** the OBDTs is proposed

- Use spare links to send ALL hits to a readout system capable to collect all data
- Useful for evaluating and debugging the new front-end boards and detector studies

EMP Framework



Readout of the OBDTs secondary links is based on Xilinx VCU118 evaluation board equipped with a 6-port QSFP FMC+ module

- Can read up to 24 lpGBT links (1 link per OBDT)

Firmware based on the **CMS EMP Framework** [2]

- Provides common infrastructural firmware, control and monitoring software
- Extended to include CMS TCDS1 receiver capable of accepting CMS phase-1 TTC

Used existing lpGBT module, payload formatter and spy buffers to capture the data stream for debugging

TCP/IP

From the EMP buffers, hits are written into Clock Domain Crossing (CDC) FIFOs to decouple ReadOut and transmission

The Payload builder process is in charge of reading hits from all the FIFOs, one per channel, and merge them in a single stream

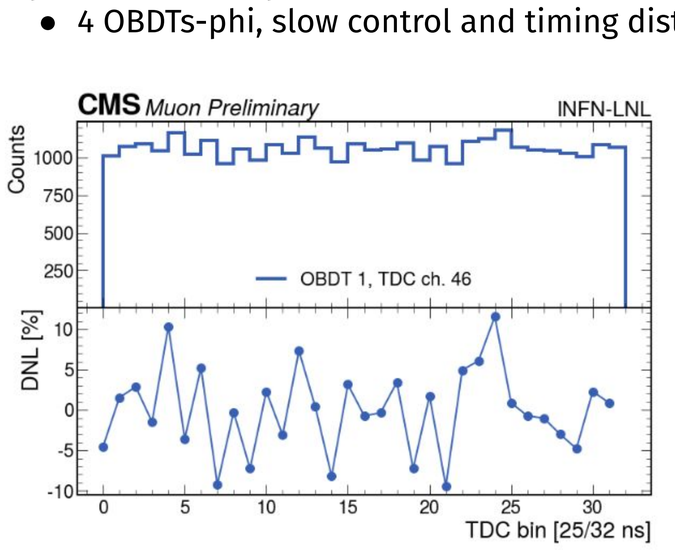
The 100G TCP/IP is implemented using a **scalable network stack for FPGA** developed by ETH [3]

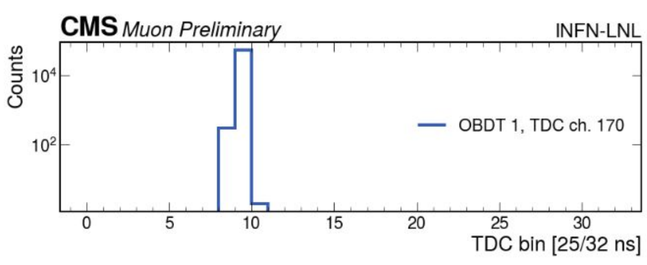
- Full TCP/IP, buffering for retransmission on external DDR4

First tests

System developed and tested at the INFN Legnaro National Laboratories (LNL) where a replica of the components installed at CMS for the slice test is available

- 4 OBDTs-phi, slow control and timing distribution system





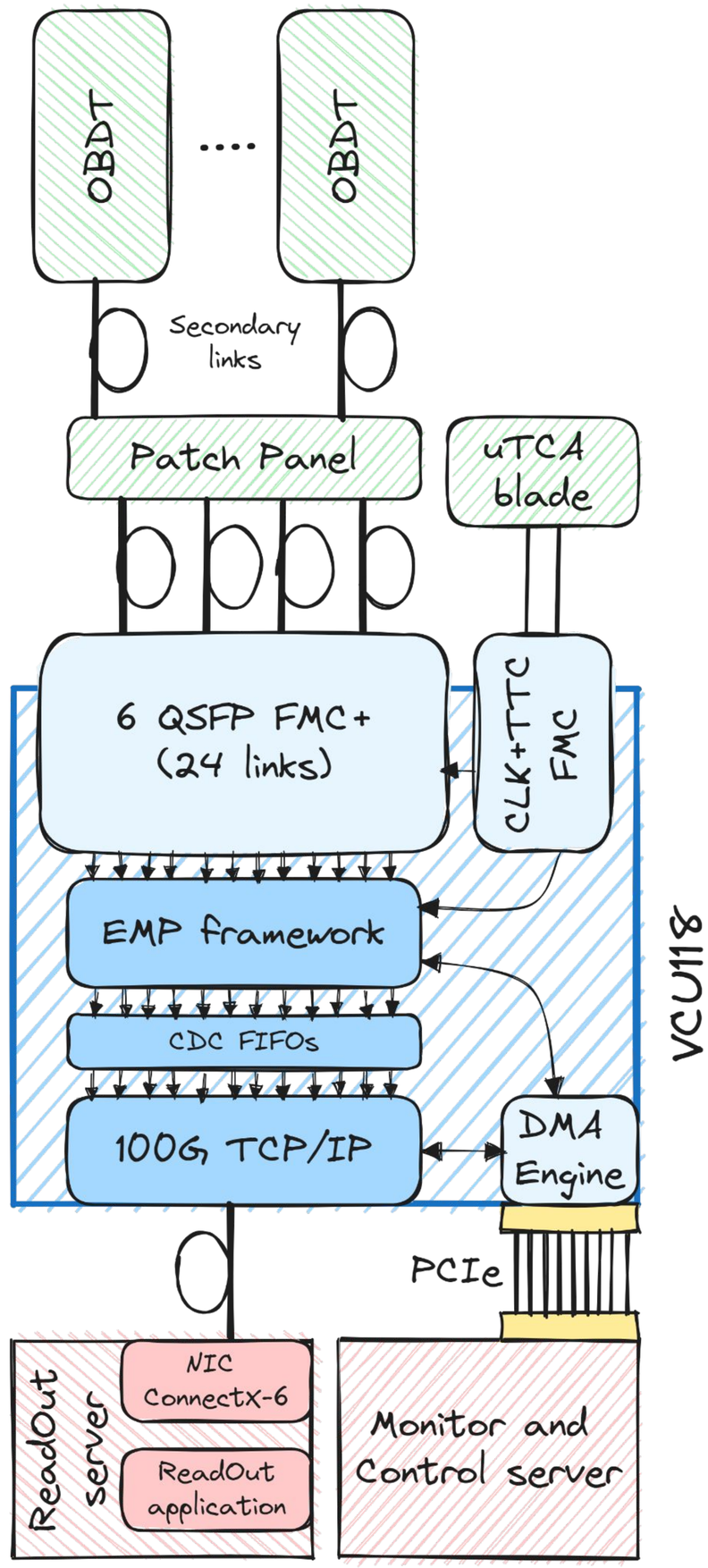
Data collected in two scenarios

Electronics Noise:
Set the threshold of the chambers' front-end comparator to 0mV

- Collected pure noise
- In case of white noise, expected a flat distribution in the TDC bins

Possible to check the differential nonlinearity (DNL) of the TDCs, measuring how close each bin is to its ideal width of 25/35ns. DNL should be within +-10% for each channel.

Test Pulse:
The OBDT board has the capability of generating periodic signal stimulating the the chamber's frontend output (Test Pulse, TP). This reflects into a signal spanning, ideally, one TDC bin. The spread is caused by the convolution of TP generation and TDC uncertainties



Software

The readout server hosts a NVIDIA Mellanox ConnectX-6 Network Interface card with a point-to-point connection with one of the VCU on-board QSFP. Stream is received with a standard TCP socket and written in a RamDisk.

The board is mounted on the PCIe slot of a server in charge of monitor and control, which consists of standard operations such as resetting the links and checking their status. This has been implemented using IPBus registers and EMP software.

Future work

Currently deployed at CMS in the context of DT Slice Test

- Allows to read and validate data from the OBDTs
- The use of a similar system to monitor the entire data stream produced by the OBDTs is foreseeable

TCP/IP implementation as a starting point, moving to RoCEv2 (INFN feROCE project)

- Implementation available in the network stack
- Developed a Real-time firmware simulation [4]

References

[1] A. Triossi et al., "The OBDT board: a prototype for the Phase2 Drift Tubes on-detector electronics."
[2] https://serenity.web.cern.ch/serenity/emp-fwk/
[3] https://github.com/fpgasystems/fpga-network-stack
[4] Front-End RDMA Over Converged Ethernet, real-time firmware simulation, G. Bortolato (TWEPP2023)