



MicroTCA-based Global Trigger Upgrade project for the CMS experiment at LHC

Babak Rahbaran*, Bernhard Arnold, Herbert Bergauer, Markus Eichberger, Dinyar Rabady

Institute of High Energy Physics of the Austrian Academy of Sciences

*babak.rahbaran@cern.ch

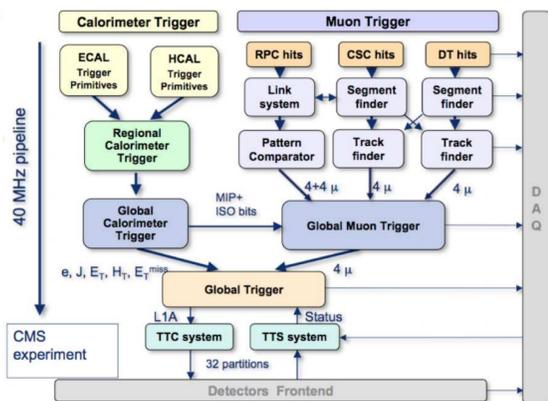
Abstract

At LHC 40 million collisions of proton bunches occur every second, resulting in about 800 million proton collisions. The Level-1 trigger, a custom designed electronics system based on FPGA technology, performs a quick on-line analysis of each collision every 25 ns and decides whether to reject or to accept it for further analysis. As part of the Global Trigger Upgrade, the current system will be redesigned and implemented for MicroTCA based technology, which allows engineers to detect all possible faults on plug-in boards, in the power supply and in the cooling system. Additionally, reconfigurability and testability will be supported based on the next system generation.

1. Current implementation

The Global Trigger electronics for the CMS is hardware based:

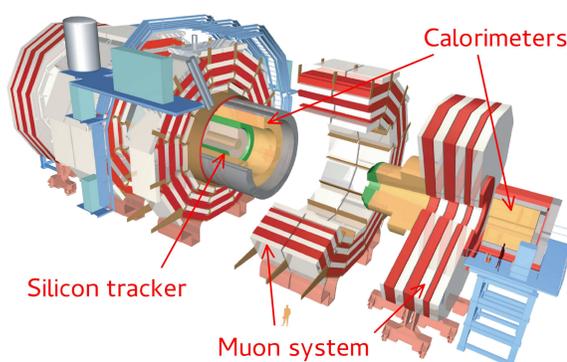
- **Dominated** by FPGAs
- **Fixed** latency
- **Guaranteed** behaviour per event
- **Receives** trigger objects from the Global Calorimeter Trigger, the Global Muon Trigger and calculates in parallel up to 128 Algorithms
- **Flexible** to adapt to changing conditions: beam type (protons, heavy ions), higher rates
- **Generate** L1_Accept signals for starting the Data Acquisition System and the High Level Trigger software



2. Physics requirements for a new trigger system

Future physics requirements for an improvement of the Level-1 Trigger:

- **Physicists** will require more complex algorithms
- **Improve** flexibility in dealing with changing requirements, i.e growing luminosity
- **Significantly** more channels when the Silicon Strip Tracker is added as a trigger source during the Phase-2 upgrade of the CMS experiment



3. New development for Trigger electronics

The next generation of the Global Trigger is based on MicroTCA and takes advantage of:

- **Scaleable** modern architecture
- **High speed** serial connectivity available in MicroTCA (fat pipes)
- **Remote** management system that detects all possible faults on plug-in boards, in the power supply systems and in the ventilation system
- **Development** and implementation of just one card type for the whole system
- **Reconfigurability, Testability, Usability**
- **Use** more algorithms in a more flexible way and apply any required prescale factors
- **Sending** read-out records to DAQ over GbE through backplane



4. Hardware architecture

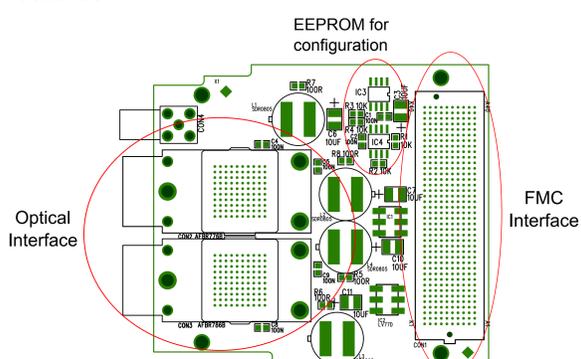
The next generation of GT uses the advantages of:

- **Dual star** MicroTCA backplane
- **Redundant** MCH slot used to distribute/receive control and feedback information
- **Commercial** AMC FPGA Carrier (AMC514-Xilinx Virtex-6 FPGA) from VadaTech with customized FMC (designed by HEPHY)
- **Cross Bar Switch** for 48 inputs and 48 outputs (12 AMCs with 4 ports each) with 6.5 Gbps data bandwidth per channel
- **Distribute** LHC clock over backplane



5. Customized hardware development

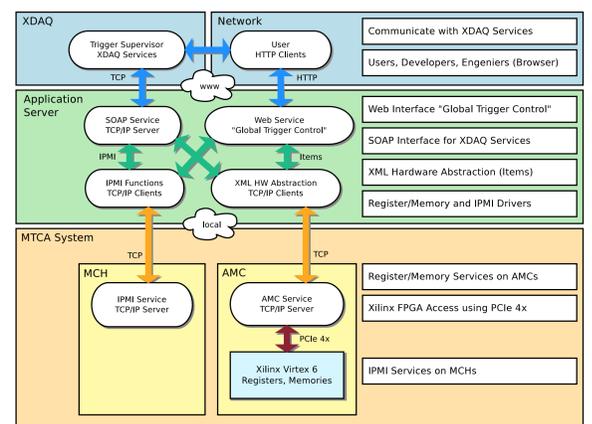
We are developing a customized FPGA Mezzanine Card (FMC) for AMC514 with the following features:



6. Software architecture

XML hardware abstraction

- **API** for resolution of name to register address
- **Tree** organization of registers
- **Integrated** datatype conversion functions
- **Configuration** by XML file



GT Control Web interface:

- **Full** display of register tree possible
- **Partial** tree display by single click
- **Collapse** and expand function
- **Offline** mode with hardware emulation



7. Software performance analysis

Current tested AMC bandwidth ↔ Global Trigger Control (System manager):

- Two obvious candidates: TCP and UDP

TCP/IP:

1. R/W → Avg:2.13s, Max:2.16s
2. Write → Avg:1.08s, Max:1.09s
3. Read → Avg: 1.08s, Max:1.08s

UDP:

1. R/W → Avg:0.16, Max:0.18s
2. Write → Avg:0.08s, Max:0.08s
3. Read → Avg: 0.08s, Max:0.08s

- **Currently** UDP outperforms TCP by a factor of ~10

- **There are some possibilities** to improve TCP performance