UPGRADE OF THE FIRST-LEVEL MUON TRIGGER FOR THE ATLAS EXPERIMENT AT THE HL-LHC

Davide Cieri* on behalf of the ATLAS collaboration

*Max Planck Institute for Physics, Munich

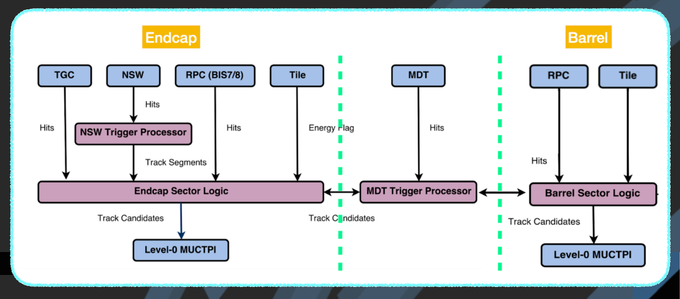
15th Pisa Meeting on Advanced Detectors • 22-28 May 2022

The First-Level Muon Trigger

The novel ATLAS first-level Monitored Drift Tube (MDT) trigger will be responsible to reduce the output trigger rate by a factor 3-5 at HL-LHC conditions, and to reconstruct muon track transverse momentum with a 6% resolution w.r.t. offline reconstruction at 20 GeV.

The new MDT Trigger Processor (MDTTP) board will use MDT hits to improve the muon track candidate parameters, coming from the RPC and TGC Sector Logic (SL) boards. Refined candidates are then sent back for the final trigger decision.

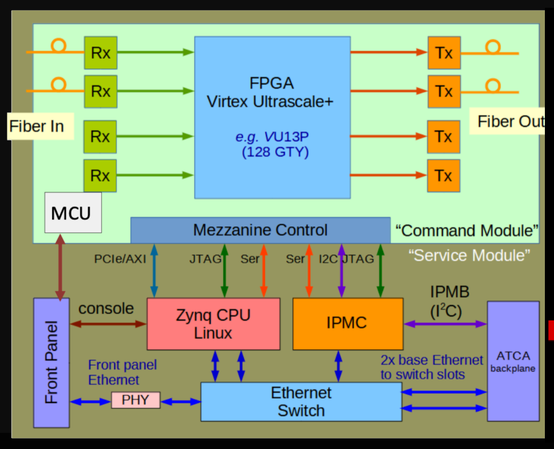
The MDTTP is also responsible for the transmission of MDT hits to the FELIX Data Acquisition system and for the monitoring and configuration of the MDT chamber front-end electronics. 64 MDTTP boards, one per MDT sector, will be installed.

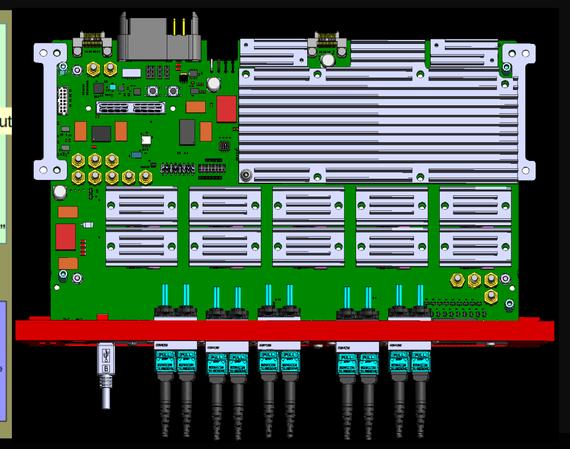


Performance

The MDT trigger rejects fake SL candidates from combinatorial background, reducing the output rate by ~50-70%, while keeping a high efficiency plateau, for a muon pT threshold of 20 GeV.

The MDTTP Command Module Prototype (2022)

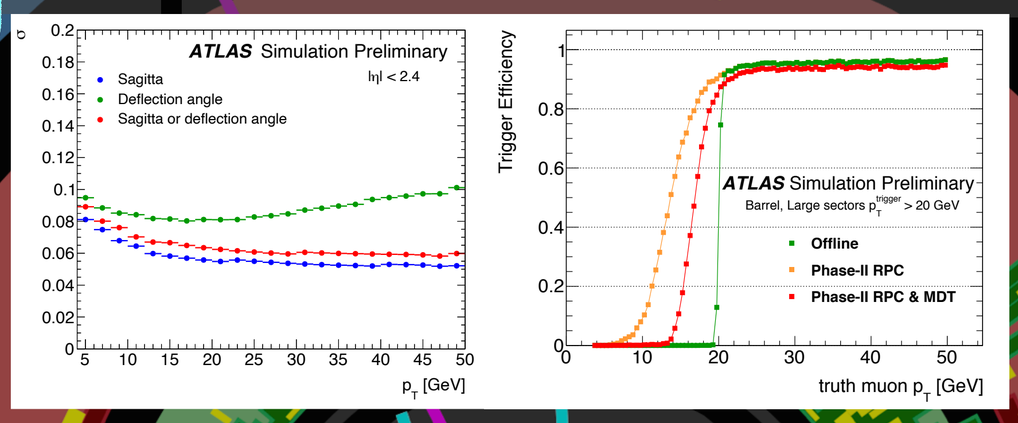




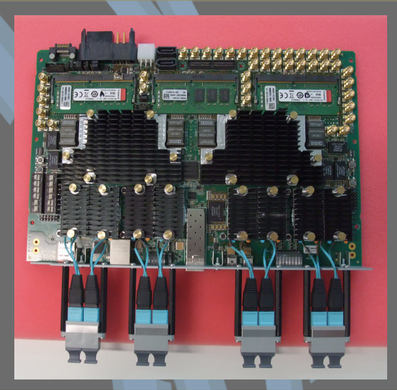
MDTTP Hardware Implementation

The MDTTP is based on the common APOLLO platform. It consists of an ATCA blade, separated in two modules. A general-purpose Service Module (SM), common to all APOLLO applications, which handles the ATCA infrastructure, the slow control and the monitoring; and an application-specific Command Module (CM), designed for the ATLAS muon trigger application.

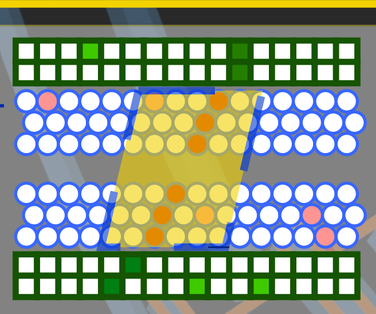
The MDTTP CM design is based on the MDTTP firmware requirements and the experience gained from the MDTTP demonstrator. It uses a large Xilinx Virtex UltraScale+ VU13P FPGA, running the trigger and data aquisition algorithms, and ten identical FireFly modules with 12 RX/TX channels capable of a line speed up to 14 Gbps, which are used communicate with the MDT



The MDTTP Command Module Demonstrator (2020)

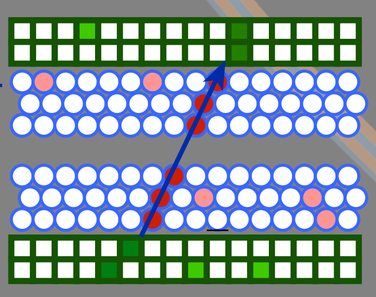


MDTTP Trigger Algorithm



1. Hit Extraction

- Uses Sector Logic candidates to open Region-of-Interest (RoI) windows in each MDT station
- Matches MDT hits to window in space and time



2. Segment Finding

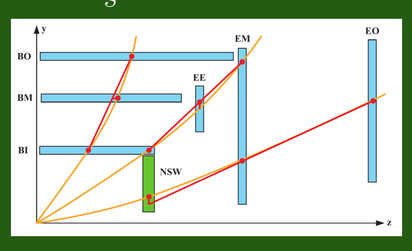
- Reconstructs muon track segments in each MDT station using matched MDT hits

3. Momentum Estimation

- The muon pT is calculated by measuring the deviation from straight track due to the magnetic field using the reconstructed MDT segments

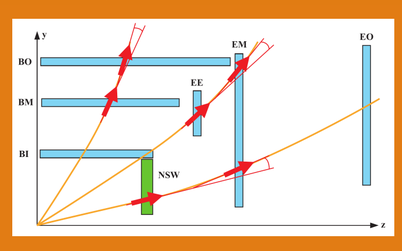
Sagitta Method

$$p_T = \Sigma_{i=0}^{2} \frac{a^i}{s^i} + \Sigma_{i=0}^{2} b_i \cdot \phi^i + \Sigma_{i=0}^{1} c_i \cdot \eta^i$$



Δβ Method

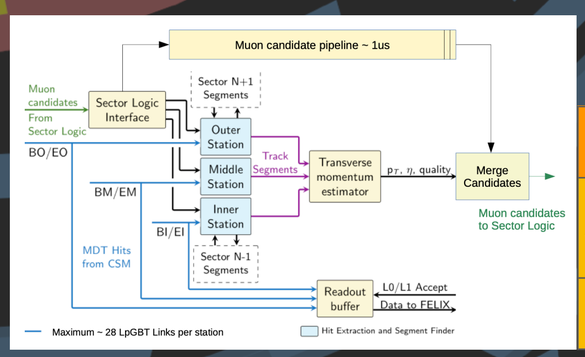
$$p_T = \Sigma_{i=0}^{2} \frac{a^i}{\Delta\beta^i} + \Sigma_{i=0}^{2} b_i \cdot \phi^i + \Sigma_{i=0}^{1} c_i \cdot \eta^i$$



Firmware Implementation

The MDTTP algorithms have been implemented in firmware targeting the VU13P FPGA. The design is divided in two major blocks:

- the Hardware Abstraction Layer (HAL), responsible for the hardware-dependent and low-level object interfaces;
- the User Logic (UL), hardware-independent, containing the actual MDTTP algorithm, running with a single clock domain.



	CLB LUTs	CLB Registers	BRAM Tiles	URAMs	DSPs
Total Available VU13P	434083	525976	1120.5	54	627
MDTTP Usage	25.1%	15.2%	41.7%	4.0%	5.1%

The design is capable of processing up to three SL candidates per bunch-crossing simultaneously. It is divided in macro-blocks, to ease the development. SL candidates are first processed in the Sector Logic Interface, which determines the RoI windows. MDT hits are then matched to these windows in each station, and used to reconstruct the MDT segments. Finally the segment data is combined in the pT estimator block, which calculates the final muon quantities, transmitting them back to the SL board and then to the Global trigger for the final decision. Upon the arrival of a L0 trigger accept signal, MDT hit data matching a certain time window are sent from the readout buffer to FELIX.

The implemented design fits comfortably in the chosen FPGA, satisfying at the same time the stringent ATLAS latency constraints.