New Developments in the MDT Trigger Processor for the ATLAS Level-0 Muon Trigger at the HL-LHC

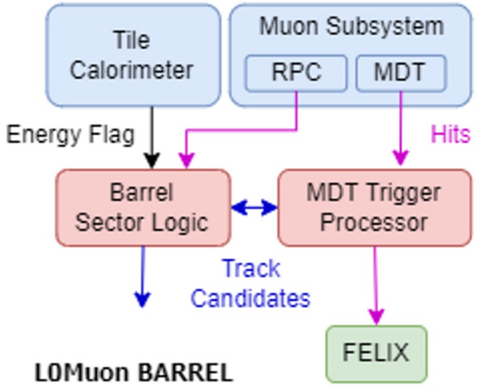
Priya Sundararajan, Kostas Ntekas

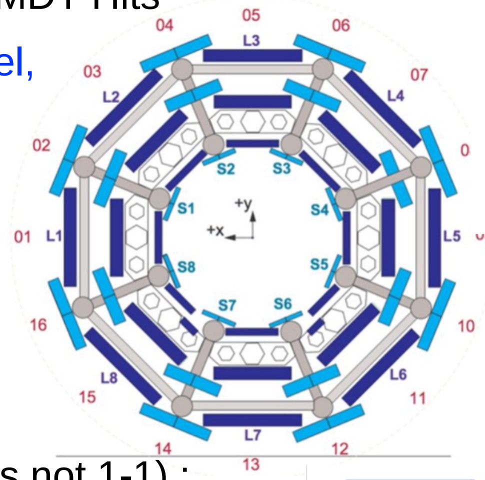
On behalf of the ATLAS TDAQ Collaboration

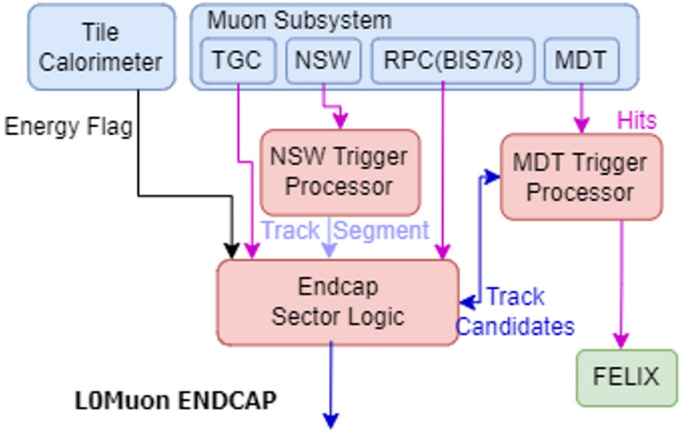
September 20th, 2022

Level-0 Muon Trigger Architecture

- The biggest update in the ATLAS Level-0 (L0) Muon Trigger for HL-LHC is the introduction of the MDT Trigger Processor (MDTTP)
  - Provides Muon track candidates with improved pT resolution, reduces fake triggers
  - Responsible for MDT front end (FE) configuration as well as read out of MDT Hits
- The Muon spectrometer is partitioned into 16 sectors in phi for endcap and barrel,
  - 64 sectors in total (A and C side)
- One or more Sector Logic (SL) per sector
- One MDTTP per sector that interfaces via optical links with
  - MDT Front end electronics: Chamber Service Module (CSM)
    - Receives MDT hits, Configures front end electronics
  - Primary and neighboring SL Board (mapping between SL-MDTTP boards not 1-1) :
    - Receives maximum 5 (barrel) or 8 (endcap) SL track candidates per bunch crossing
    - Transmits reconstructed track candidates at a fixed latency of 1.7us
  - FELIX Board (**ATL-DAQ-SLIDE-2021-157**)
    - Receives LHC clock and synchronizations commands
    - Transmits MDT Hits on a L0 accept

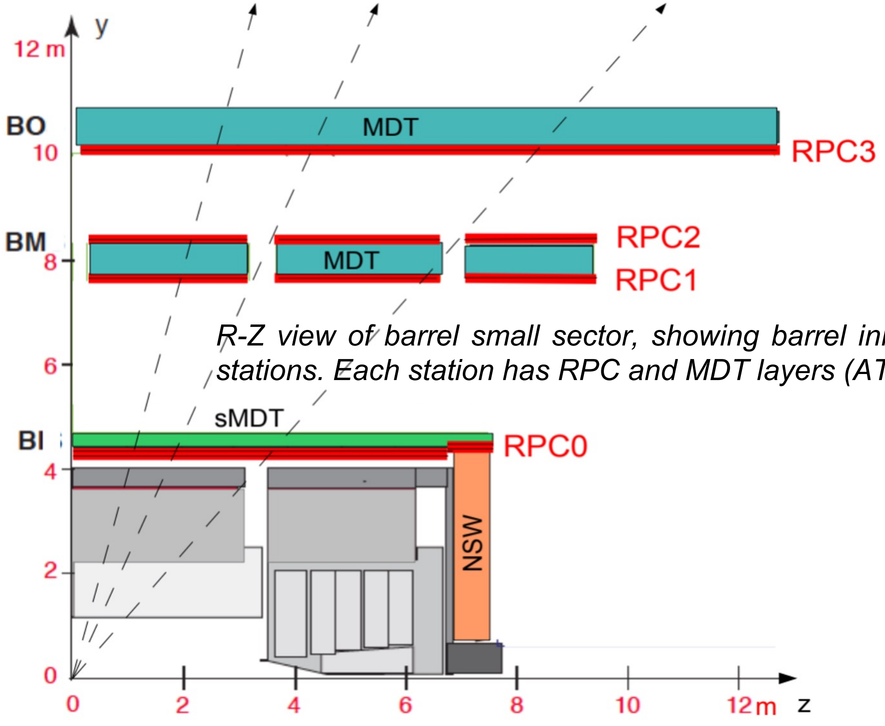




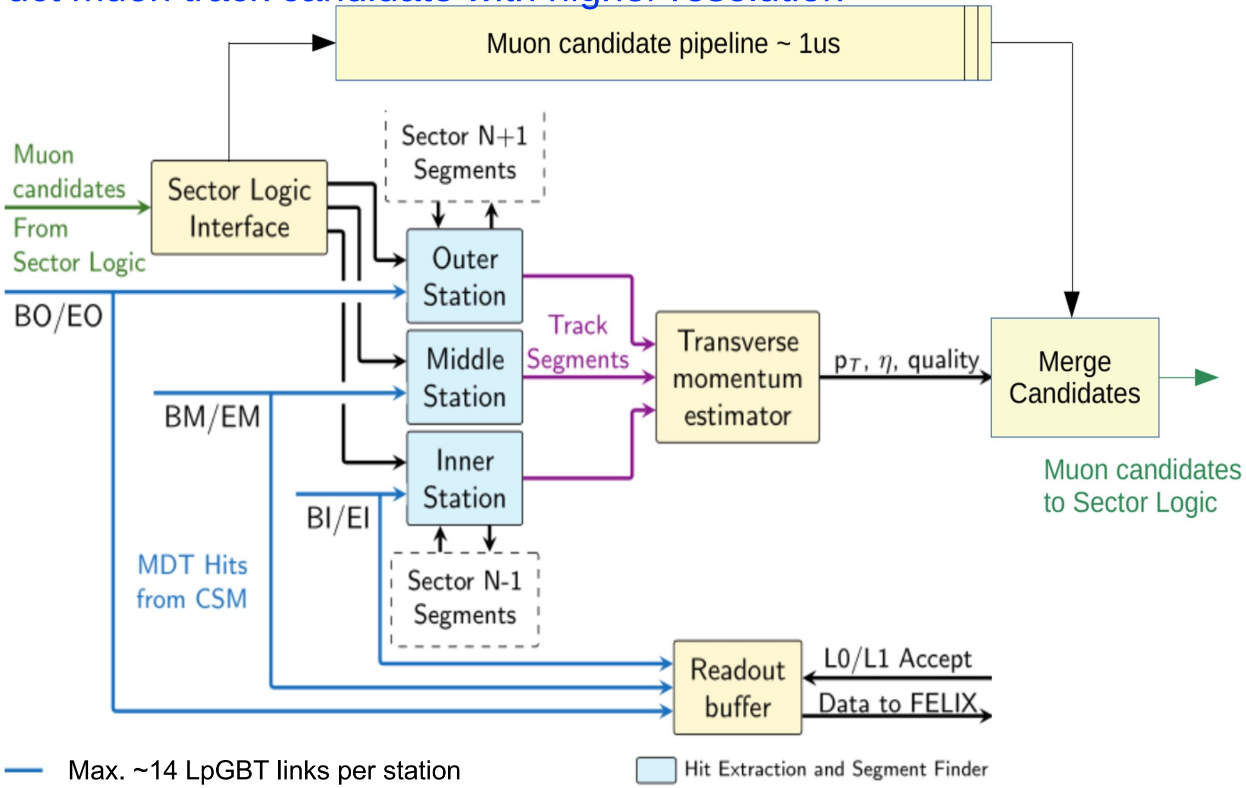


MDTTP Data Flow

- The Muon subsystem has three layers of MDT chambers for barrel and endcap
- MDTTP processes hits from each station (inner, middle, outer) independently
  - Muon track candidate from SL board is used to define a region of interest (RoI) for each station
  - MDT Hits matched to RoI are used to construct track segment
- Track Segments from the different stations are combined to reconstruct muon track candidate with higher resolution
- Reconstructed track candidate are sent back to SL at fixed latency

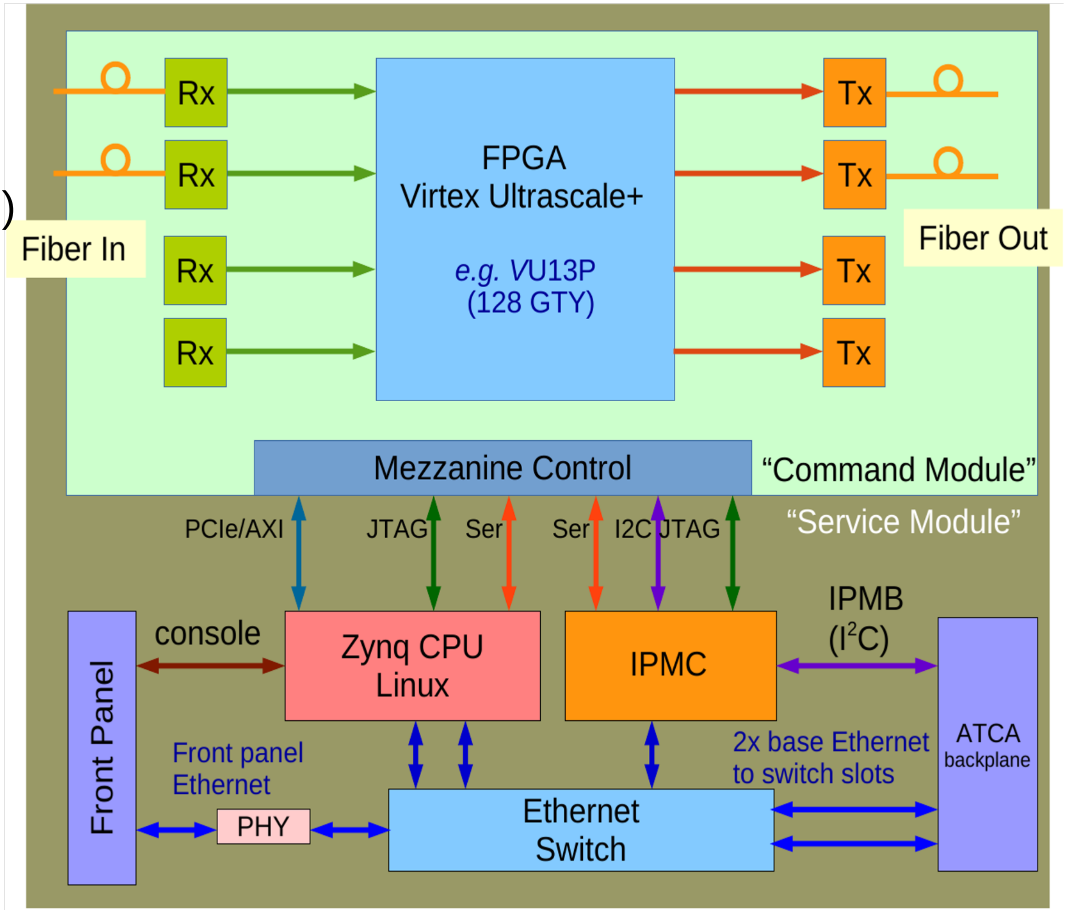


R-Z view of barrel small sector, showing barrel inner, middle and outer stations. Each station has RPC and MDT layers (ATLAS-TDR-026)



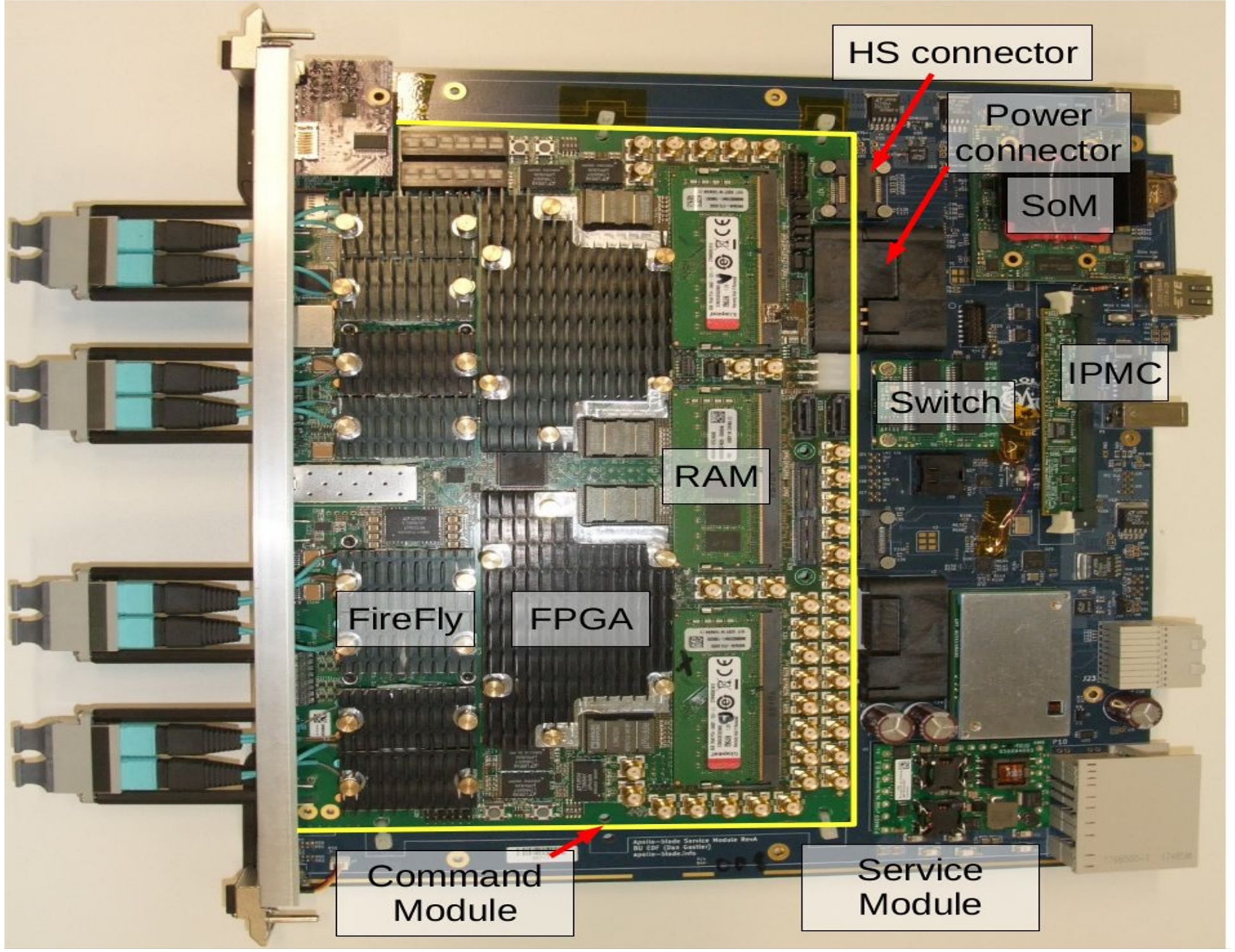
MDTTP Hardware - Service Module

- Apollo blade is based on a standard ATCA platform. Each blade is divided into Service Module (SM) and Command Module (CM)
- SM board
  - Common board for different applications
  - Handles ATCA blade infrastructure and communications
  - Hardware Monitoring and interface to Detector Control System (DCS)
  - Main components include
    - Power Management
    - Intelligent Platform Management Controller (IPMC)
    - Ethernet Switch
    - Xilinx Zynq System-on-chip (SoC) FPGA
      - Setting up Microcontroller in CM
      - Programming CM FPGA
      - Has access to CM trigger path
        - » Control and status registers
        - » Fast Monitoring Logic



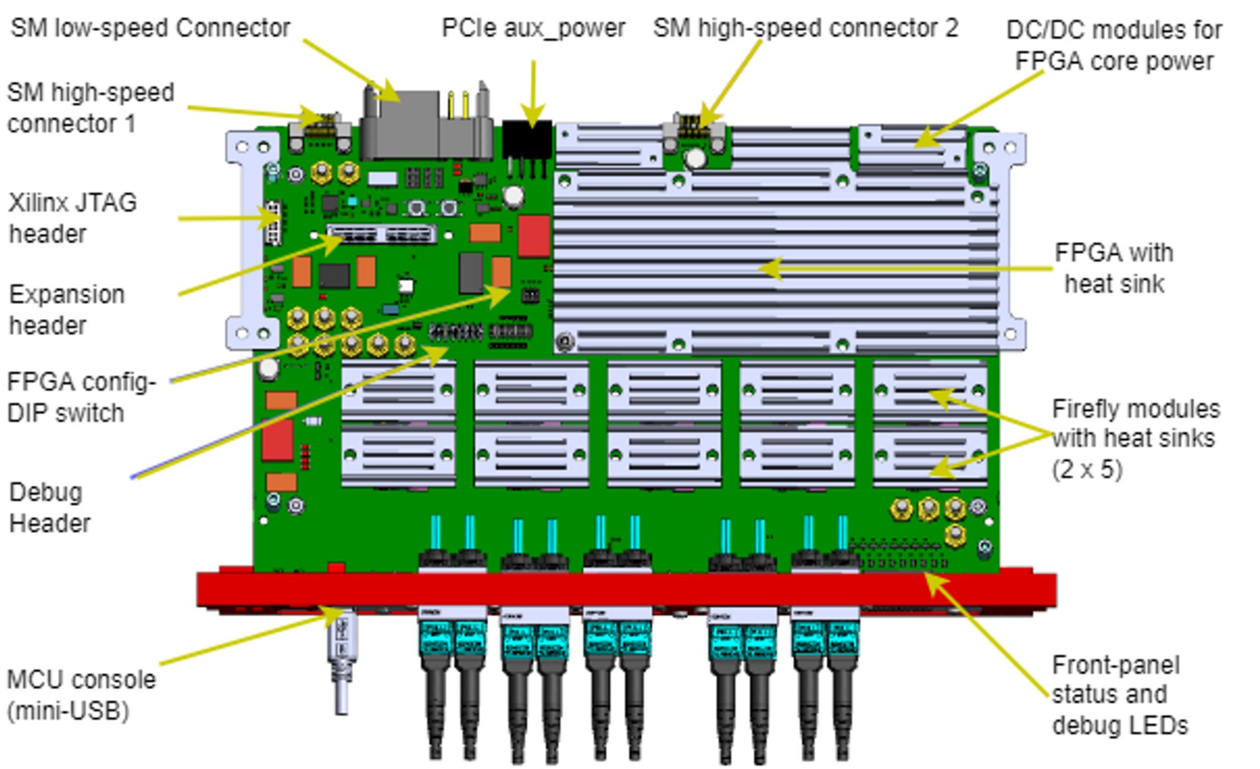
MDTTP Hardware - Command Module (Demonstrator)

- Command Module is application specific. It implements the trigger and readout logic
- Demonstrator board
  - 2-FPGA Solution (KU15P & ZU11EG)
  - Board level tests done to validate
    - optical links
    - SM-CM AXI C2C link
    - clocks
    - power
    - Micro Controller Unit



MDTTP Hardware - Command Module (Prototype)

- Prototype board
  - Single FPGA solution (VU13P) → Simplifies firmware logic
  - Removed unused features (DDR4, Zync MPSoc)
  - Simpler configuration and power scheme
  - Ten identical Firefly modules capable of line rate up to 14Gbps



SM low-speed Connector

PCIe aux_power

SM high-speed connector 2

DC/DC modules for FPGA core power

SM high-speed connector 1

Xilinx JTAG header

Expansion header

FPGA config-DIP switch

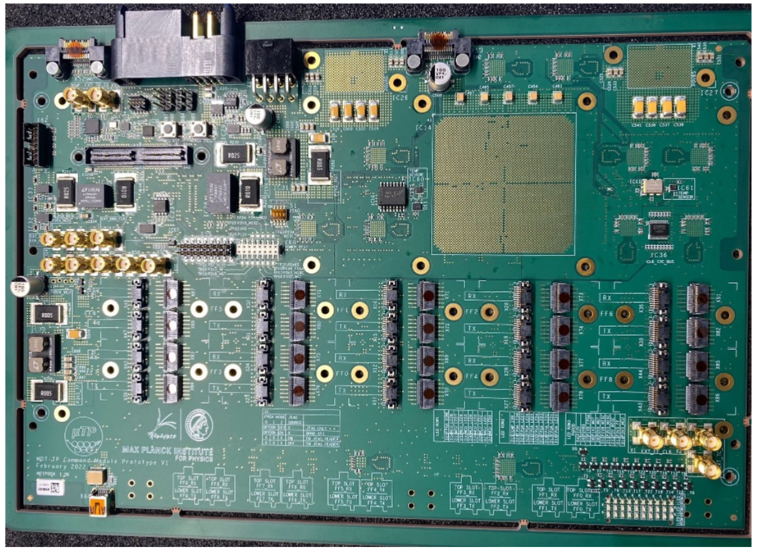
Debug Header

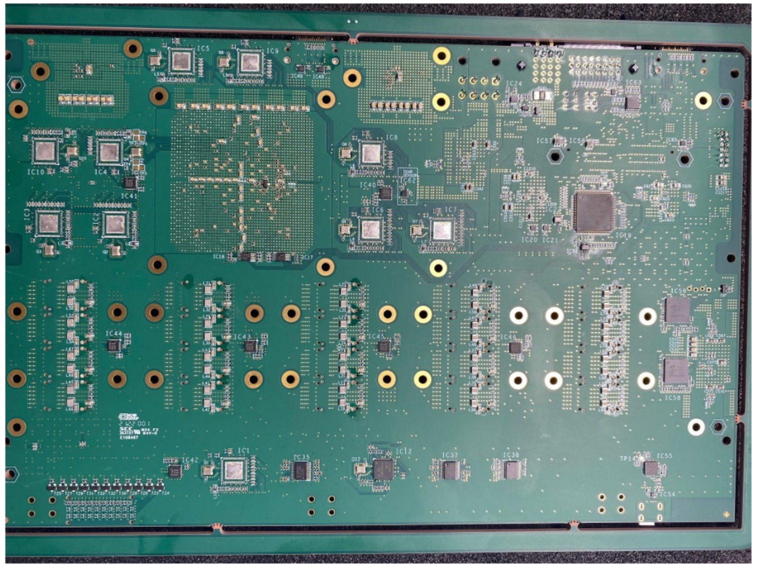
MCU console (mini-USB)

FPGA with heat sink

Firefly modules with heat sinks (2 x 5)

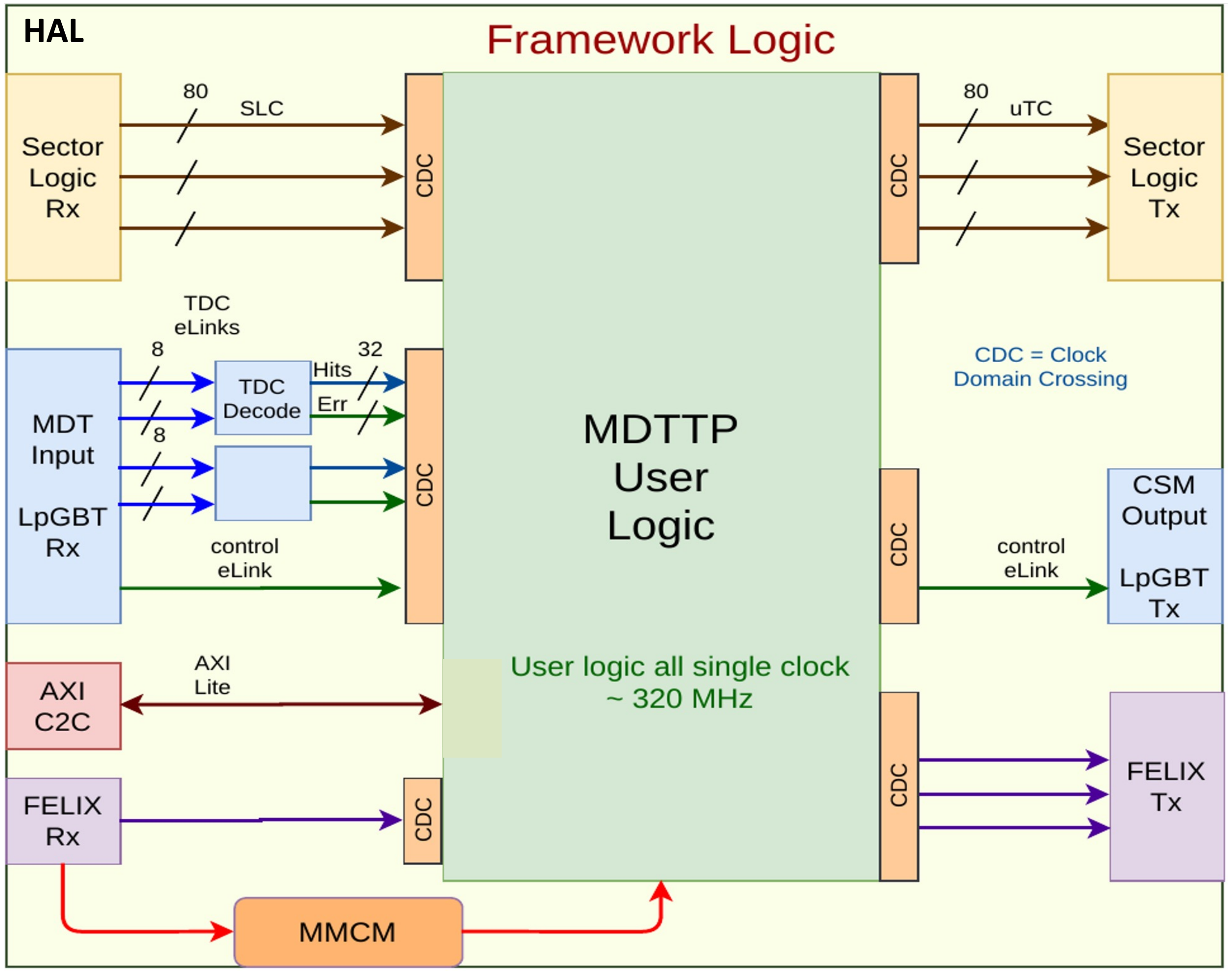
Front-panel status and debug LEDs





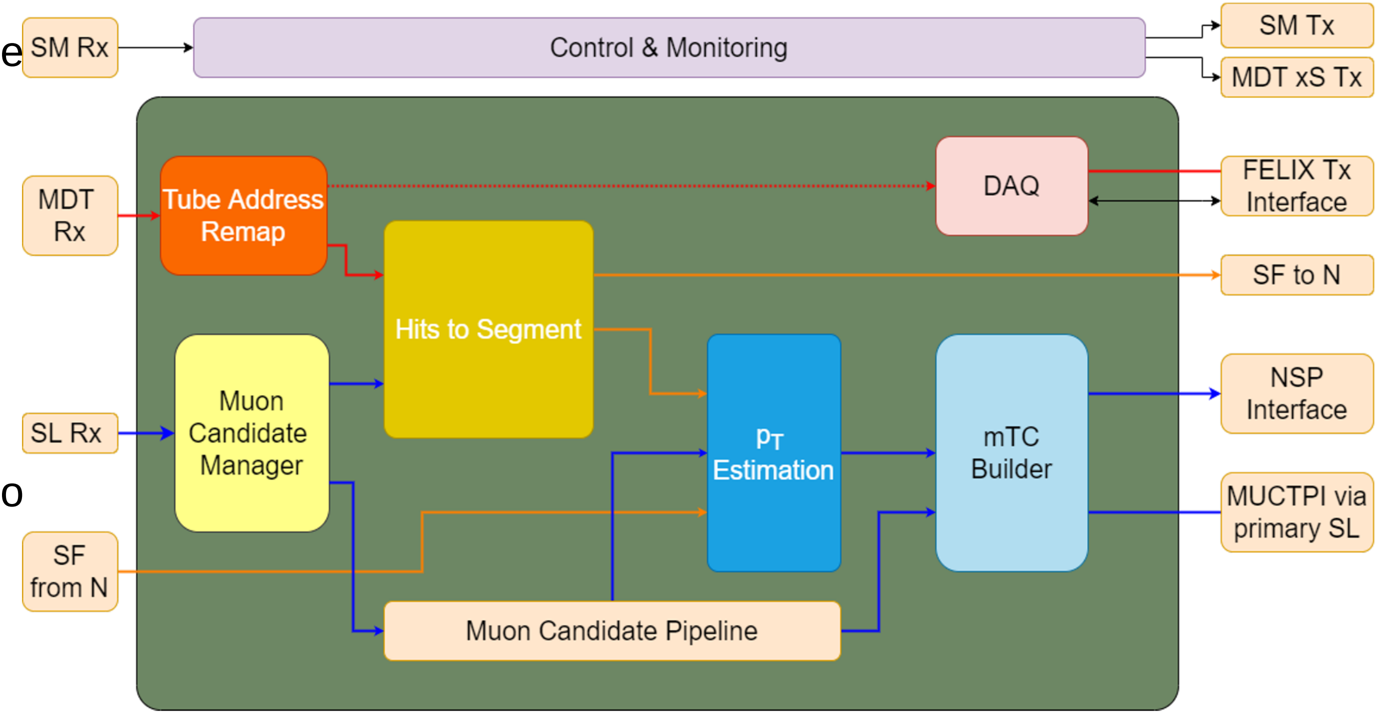
Command Module Firmware Architecture

- Firmware functionally divided into
  - Hardware Abstraction Layer (HAL)
    - IO interfaces to SM , SL, CSM, FELIX
    - Clocking : Logic to control clock domain crossing from LHC (~40MHz) to User Logic (320MHz)
    - Slow Control and Monitoring
  - User Logic - Single Clock domain (320MHz)
    - Trigger Data Path
    - DAQ
    - Fast Monitoring Logic
- Data format definitions exist for all interfaces
  - External links to SL, CSM
  - HAL interface with User Logic
  - Block interfaces within User Logic



User Logic Sub-Blocks

- Muon Candidate Manager (MCM)
  - Uses incoming SL trigger candidate to compute SL vectors per station
- Tube Address Remapping (TAR)
  - Synchronize MDT Hits with SL candidates
  - Map the MDT tube info to their global coordinates
- Hits to Segment (one per station)
  - Calibrate and filter hits based on their position
  - SL vector and associated hits are used to reconstruct track segment at the station
    - Two segment finding algorithms implemented
      - Legendre Segment Finder (LSF)
      - Compact Segment Finder (CSF)
- Muon Candidate Pipeline (MCP)- Delay SLC information
- DAQ - Sends relevant hits to FELIX on Level 0 Accept
- Transverse momentum (pT) Estimation - Combine segments reconstructed at each station
- Muon trigger candidate (mTC) Builder - sends response to SL board

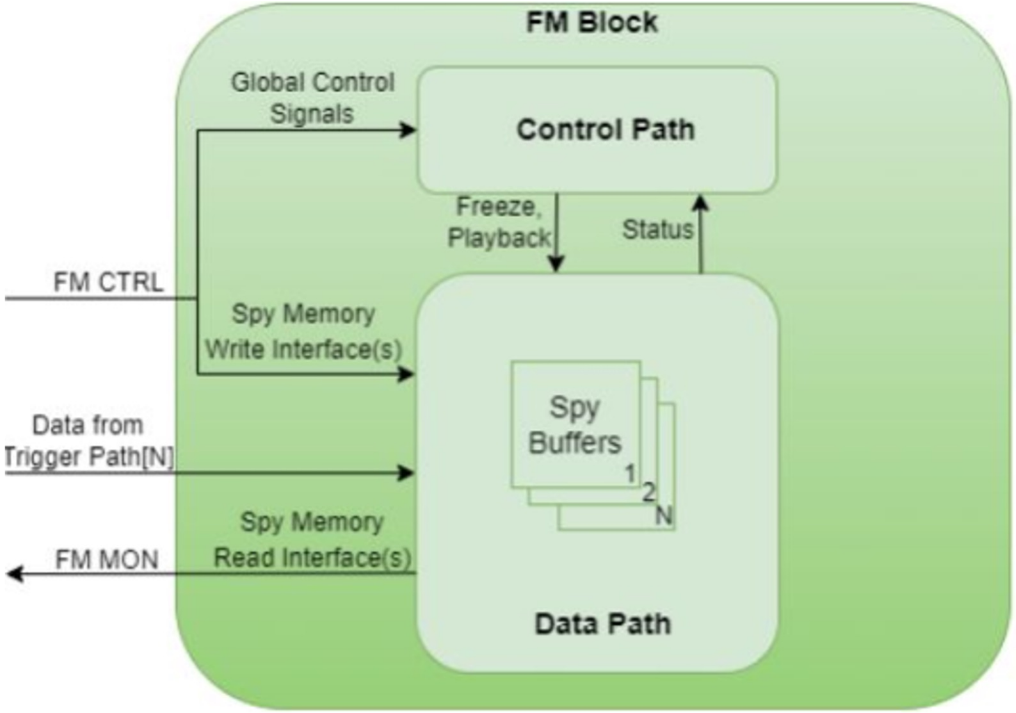


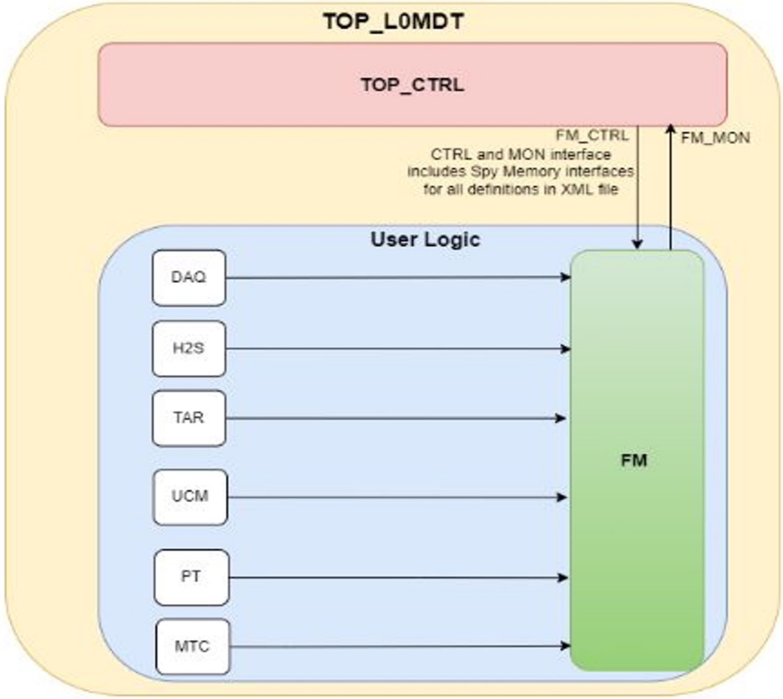
__Note__

- *MDTTP processes 3 SL candidates at a given time*
- *For every incoming SL candidate a response is sent back in fixed latency*

Fast Monitoring

- Fast Monitoring (FM) via Spybuffers can be used for firmware, hardware testing and for online data quality
    - Store trigger path data into SpyBuffer Memory
    - Freeze store operation and analyze/debug data in SpyMemory
    - Freeze store operation and inject test vectors into trigger path
- Spybuffer memory sized to hold data for minimum 5 BCIDs (est. 18% of VU13P BRAMs)
- Framework developed to integrate FM with User Logic and Control Firmware




TOP_L0MDT
TOP_CTRL
FM_CTRL
FM_MON
CTRL and MON interface includes Spy Memory interfaces for all definitions in XML file
User Logic
DAQ
H2S
TAR
UCM
PT
MTC
FM

Latency

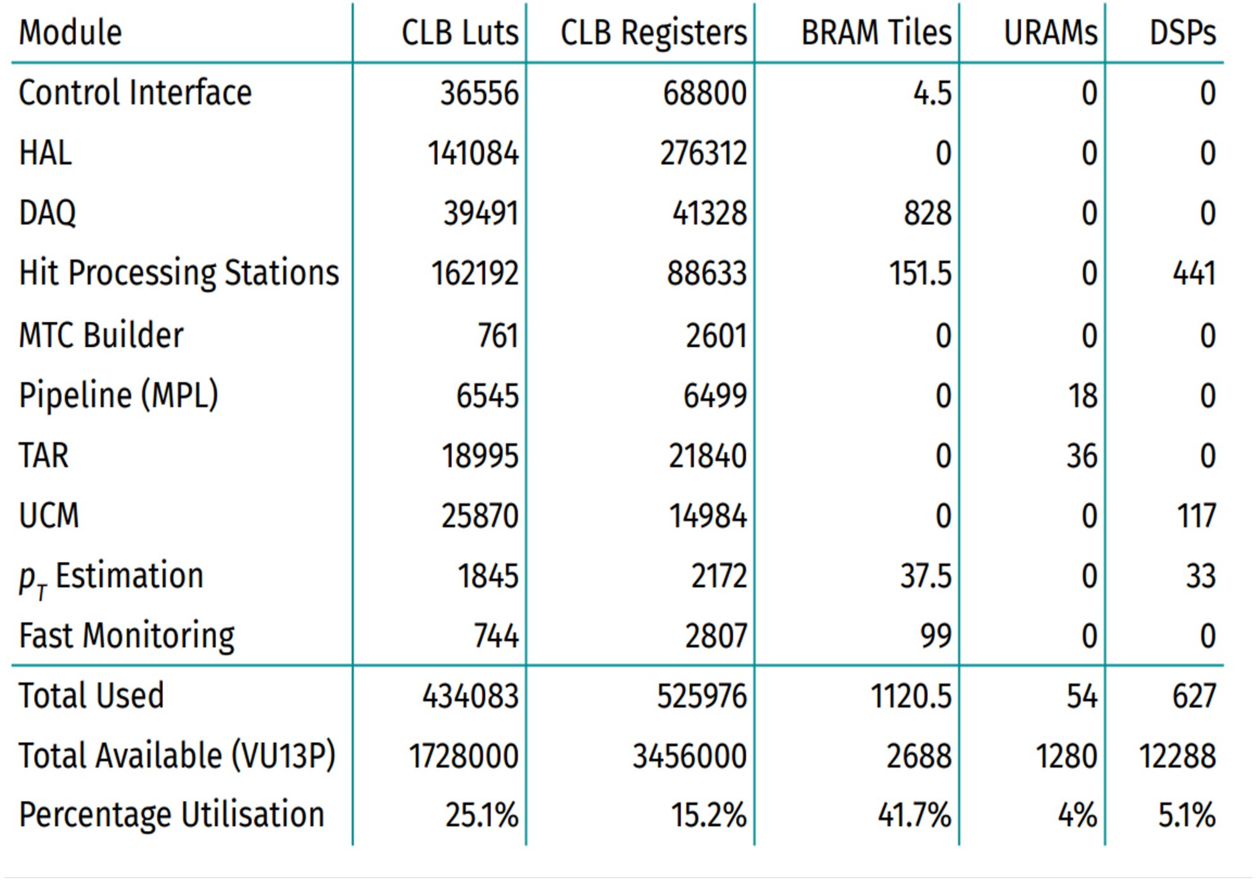
- Hit extraction, segment reconstruction and momentum determination must be done within 1.764 us
- Firmware Registers in place to control termination of hit processing to meet latency requirements
  - Requirement is to process 99.5% of real muon hit in the highest background environments
  - This termination could happen in advance for neighboring candidates to compensate for delay introduced when sending reconstructed segment to primary board

Description	Max Latency [ns] CSF	Max Latency [ns] LSF
Input Delay for SL candidates	1207	1229
Segment Finder Finalization	181	75
Rest of trigger path	330	330
Total	1718	1634

MDTTP latency estimates well within project requirements

Firmware Status & Resource Estimation

- Firmware blocks integrated and implemented with Vivado tools to estimate resources in VU13P (utilizes stacked silicon interconnect technology and has four dies)
  - Firmware partitioned across Super Logic Regions and latency of signals cross SLR regions have been studied and included in latency calculations
  - Resource estimation meets <50% requirement
  - DAQ is likely to be updated to use URAMs
  - Integration time of firmware blocks eased by definition of data format between each internal block interface
- Functional Validation in Progress

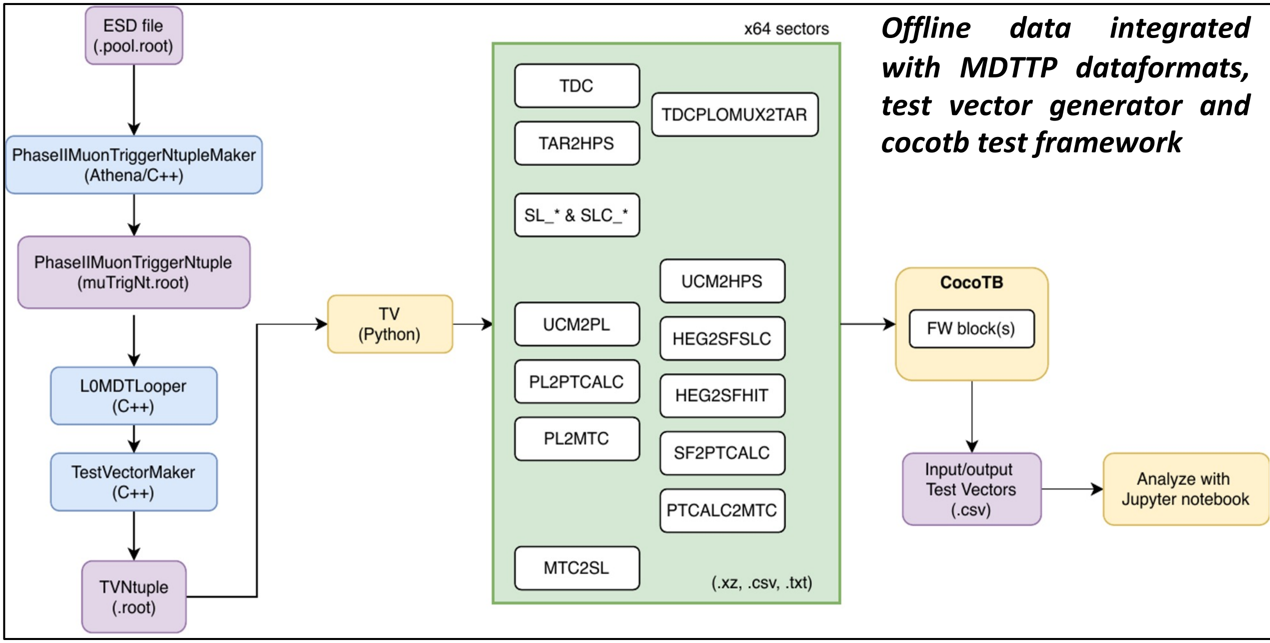


Module	CLB Luts	CLB Registers	BRAM Tiles	URAMs	DSPs
Control Interface	36556	68800	4.5	0	0
HAL	141084	276312	0	0	0
DAQ	39491	41328	828	0	0
Hit Processing Stations	162192	88633	151.5	0	441
MTC Builder	761	2601	0	0	0
Pipeline (MPL)	6545	6499	0	18	0
TAR	18995	21840	0	36	0
UCM	25870	14984	0	0	117
p_T Estimation	1845	2172	37.5	0	33
Fast Monitoring	744	2807	99	0	0
Total Used	434083	525976	1120.5	54	627
Total Available (VU13P)	1728000	3456000	2688	1280	12288
Percentage Utilisation	25.1%	15.2%	41.7%	4%	5.1%

Firmware Testing – Simulation and Hardware Testing

Simulation

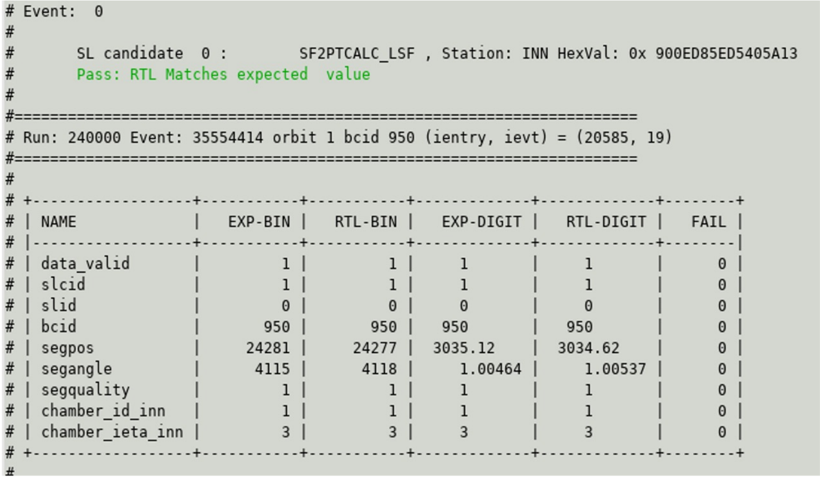
- Firmware validation requires test vectors to be available for all interfaces between the different MDTTP blocks
- Python utilities developed (DataFormats, TVMaker, TVReader)
- Functional simulation done using Questa simulator
- Two test frameworks available
  - CocoTB based test framework built to automate test creation and validate sections or entire trigger path
  - VHDL based framework can validate entire trigger path and individual blocks



Offline data integrated with MDTTP dataformats, test vector generator and cocotb test framework

Demonstrator CM Hardware Testing

- BUTool used to program CM FPGA, access CM registers, Block RAM memory
- Example CM firmware written for testing AXI C2C link between SM and CM
- Validation of MDTTP bare metal firmware in progress



```
# Event:  0
#
#       SL candidate  0 :        SF2PTCALC_LSF , Station: INN HexVal: 0x 900ED85ED5405A13
#       Pass: RTL Matches expected  value
#
#===============================================================
# Run: 240000 Event: 35554414 orbit 1 bcid 950 (ientry, ievt) = (20585, 19)
#===============================================================
#
# +-------------------+-----------+-----------+-------------+-------------+--------+
# | NAME              |   EXP-BIN |   RTL-BIN |   EXP-DIGIT |   RTL-DIGIT |   FAIL |
# |-------------------+-----------+-----------+-------------+-------------+--------|
# | data_valid        |         1 |         1 |   1         |   1         |      0 |
# | slcid             |         1 |         1 |   1         |   1         |      0 |
# | slid              |         0 |         0 |   0         |   0         |      0 |
# | bcid              |       950 |       950 |   950       |   950       |      0 |
# | segpos            |     24281 |     24277 | 3035.12     | 3034.62     |      0 |
# | segangle          |      4115 |      4118 |     1.00464 |     1.00537 |      0 |
# | segquality        |         1 |         1 |   1         |   1         |      0 |
# | chamber_id_inn    |         1 |         1 |   1         |   1         |      0 |
# | chamber_ieta_inn  |         3 |         3 |   3         |   3         |      0 |
# +-------------------+-----------+-----------+-------------+-------------+--------+
#
```

Cocotb test framework: Table generated to compare Segment Finder block results against offline

Conclusion

- MDTTP hardware and firmware have passed Product Design Review (PDR)
  - Final Design Review planned for Spring of 2023

- Next Steps
  - Firmware
    - Complete functional validation against offline test vectors with focus on block integration
      - Fine tuning firmware performance with offline will be second step
    - Analyze timing and placement in Vivado implementation
  - Software development on Zyng SM SoC
    - FM
    - Slow control and monitoring
    - Trigger Path control
    - DAQ control
  - MDTTP Demonstrator Board
    - Validation in progress of L0MDT HAL firmware
    - Use FM firmware and offline test vectors to validate firmware trigger blocks in stages

- MDTTP Prototype board planned to arrive in September 2022

BACKUPS

Baseline Upgrade Programme of LHC

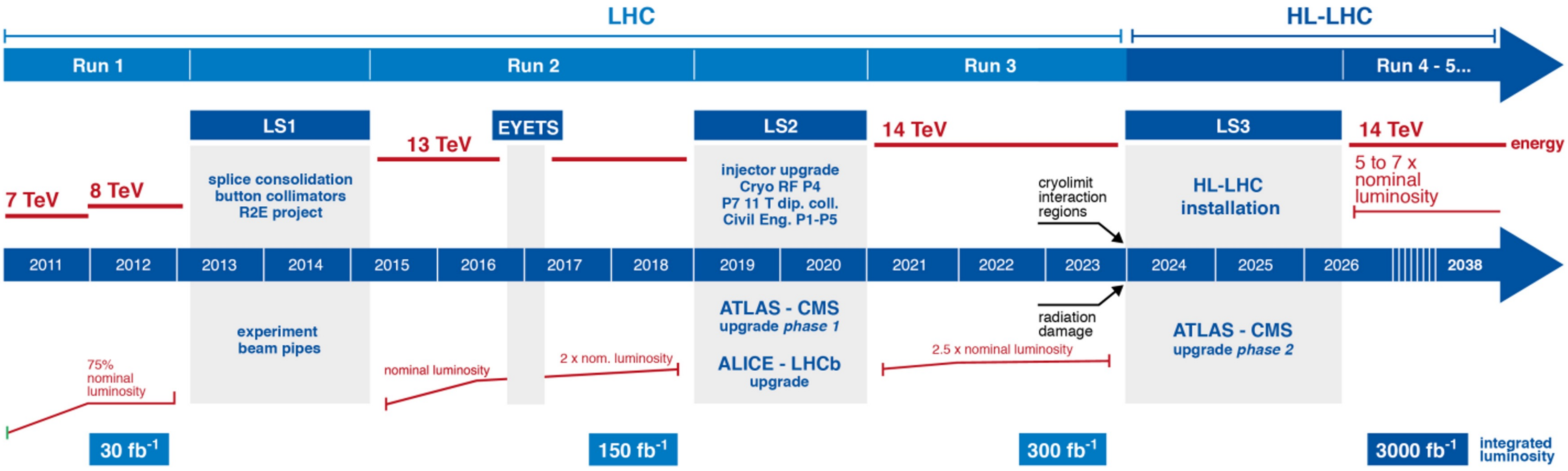
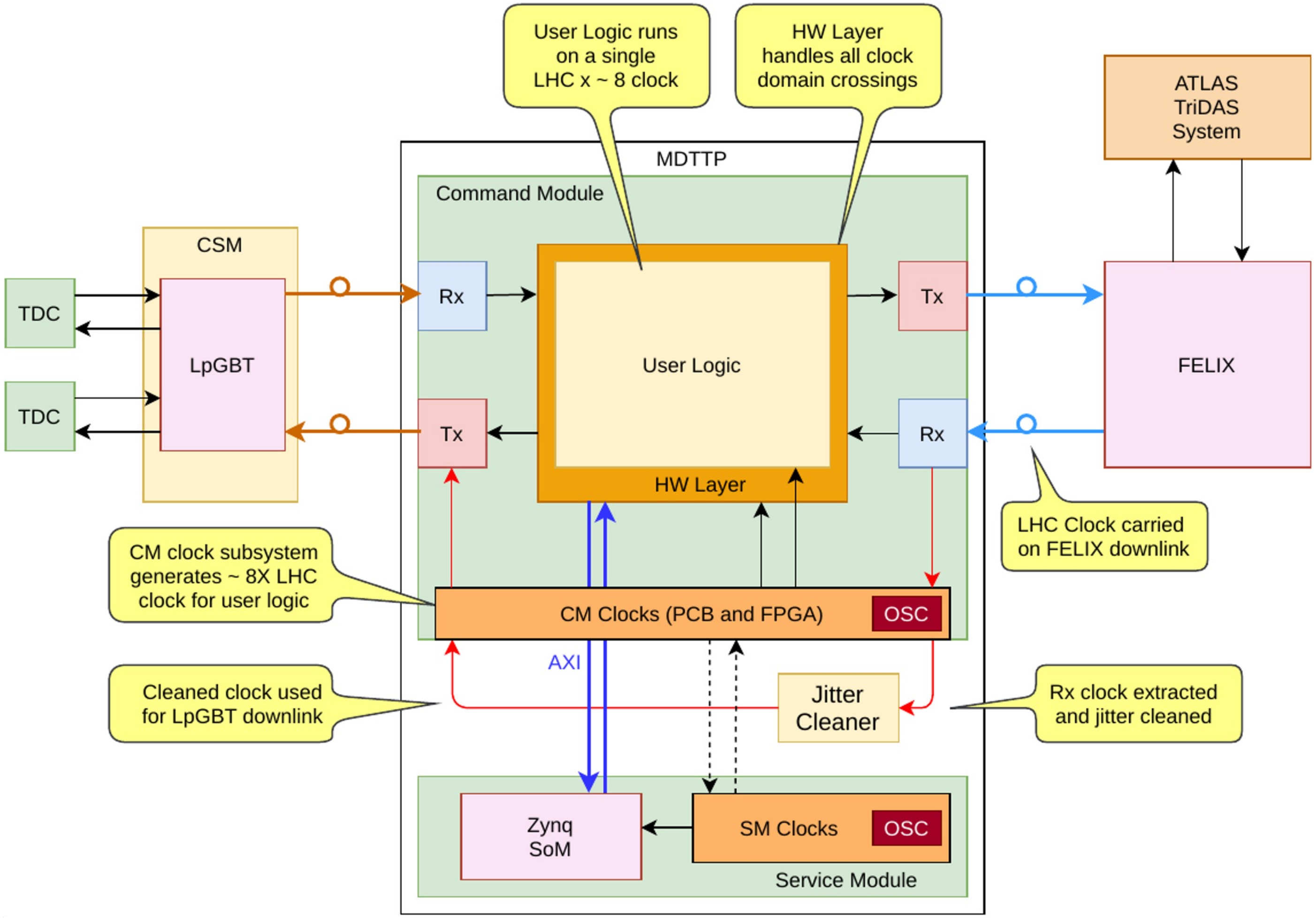


Figure 1.1: *Baseline upgrade programme of the LHC accelerator complex over the next two decades. The plan shows the scheduled long shutdowns for the machine and experiment upgrades, and the projected luminosities delivered by the LHC in a three-year run period following each long shutdown [1.2].*

ATLAS-TDR-029.pdf

MDTTP CM Prototype - Clock Infrastructure

Simplified clocking for MDT readout system. The master clock originates in the ATLAS TTC system, and is distributed to the MDTTP via the downlink fiber of the FELIX system. The red arrows show how the clock flows from the FELIX receiver, out of the FPGA through a jitter cleaner, and back into the FPGA for use as the lpGBT transmitter reference clock


User Logic runs on a single LHC x ~ 8 clock
HW Layer handles all clock domain crossings
ATLAS TriDAS System
MDTTP
Command Module
CSM
TDC
TDC
LpGBT
Rx
Tx
User Logic
HW Layer
Tx
Rx
FELIX
LHC Clock carried on FELIX downlink
CM clock subsystem generates ~ 8X LHC clock for user logic
CM Clocks (PCB and FPGA)
OSC
AXI
Cleaned clock used for LpGBT downlink
Jitter Cleaner
Rx clock extracted and jitter cleaned
Zynq SoM
SM Clocks
OSC
Service Module

MDTTP Demonstrator Hardware Testing

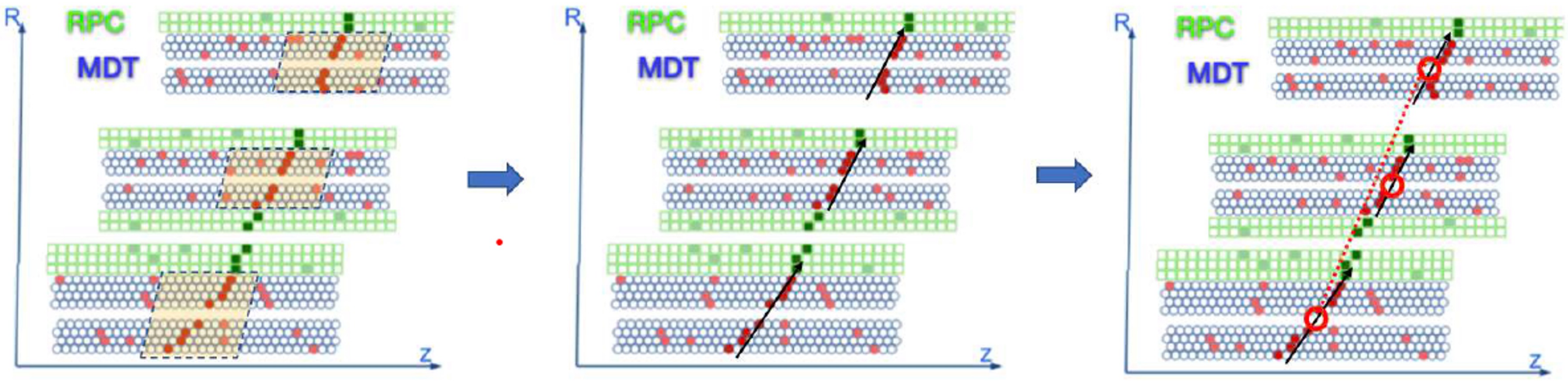
Many useful tests have been done with the demonstrator hardware. The design of the CM prototype V1 is based on knowledge gained with the CM demonstrator.
Points that were found to be working OK on the demonstrator:

- SM-CM interface (handshaking, AXI Chip2Chip, UART).
- CM power: Stability for fast transients in current consumption.
- CM MCU firmware download / update.
- CM FPGA firmware download via on-board JTAG header and SM SoM.
- CM slow control: I^2C, UART, temperatures, serial number, FireFly configuration
- Optical IOs:
  - Very low bit error rate of less than 10^{-14} at 14 Gbps and at typical line rates for the attached peripherals: 2.56, 4.80, 10.24 Gbps.
  - Stable operation with recovered reference clock.
  - Stable operation with walking LHC clock in range of 39.7 … 40.3 MHz.

These items showed the need for improvement:

- Variation of clock phases of the recovered LHC clock and its multiples due to missing zero-delay feedback option at the clock generator ICs.
- Cooling tests showed temperatures of ≈ 90 °C at moderate power of 70 W.
  => Need for well-designed cooling for CM prototype.

MDTTP Algorithms



Hit Extraction

- Reconstruct SL vectors per MDT station
- Define RoIs around these vectors in the different MDT stations
- Match the MDT hits to SL input in space and time

Segment Finding

- Receive matched MDT hits and SL vector from Hit Extraction
- Reconstruct segments in each MDT station
- 2 different algorithms have been implemented
  - Legendre Segment Finder (LSF)
  - Compact Segment Finder (CSF)

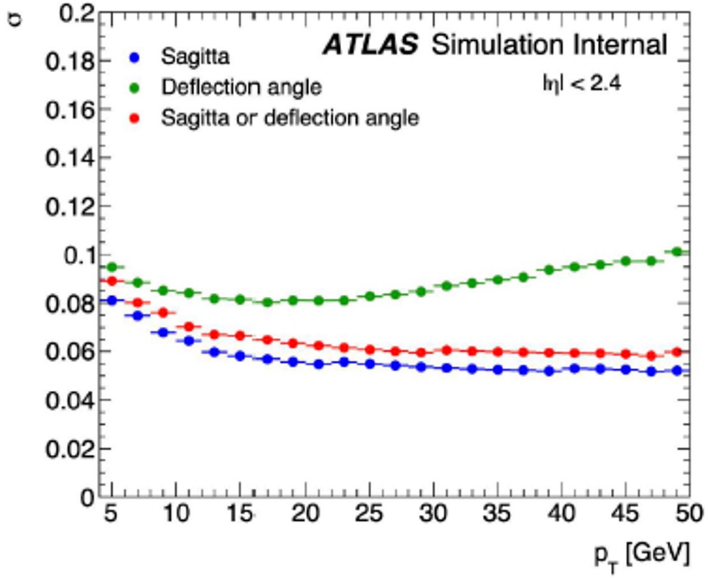
pT estimation

- Calculate the muon candidate pT and charge by measuring the deviation from straight track due to the B-field using the reconstructed segments
- Update position of the candidate with η of the innermost segment

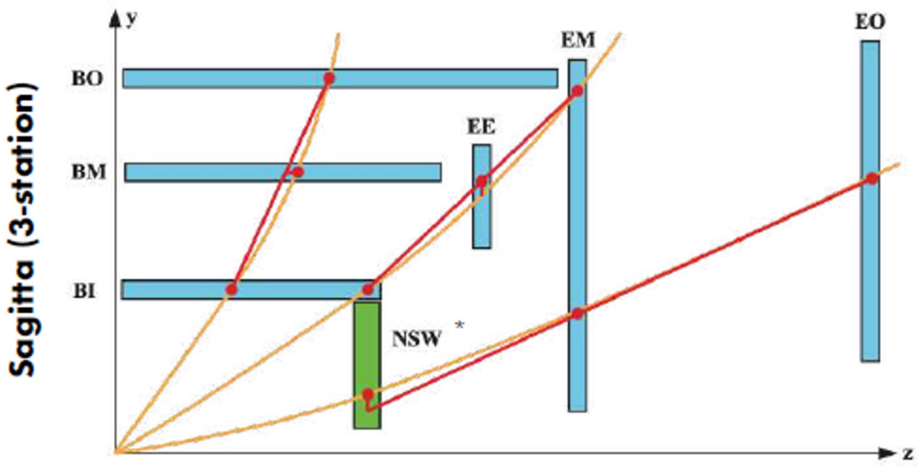
MDTTP Algorithms - pT calculations

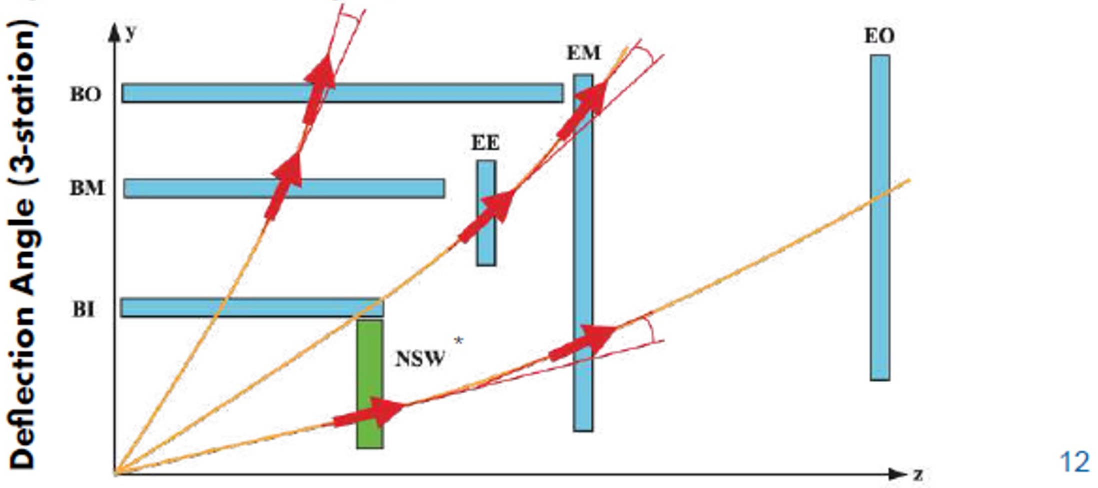
- **Two methods for estimating the pT using the MDT segments**
  - 3-station : Calculate sagitta using the positions of 3 segments (applicable to 73% of the muons)
  - 2-station : Calculate deflection angle between 2 segments (applicable to 94% of the muons)
- **Both quantities correlated with muon pT and B-field**
  - The momentum dependency on the deflection of the muon trajectory can be parametrised with η and ϕ
  - Separate parametrisation extracted for each chamber combination (>450 combo IDs with valid stats)
- **Method had been implemented and studied for the TDR**
  - Several updates/improvements over the last 2 years (better performance)

pT calculation accuracy



* NSW vector is provided to the MDTTP from the endcap SL (currently using MDT/CSC segments for the NSW region)

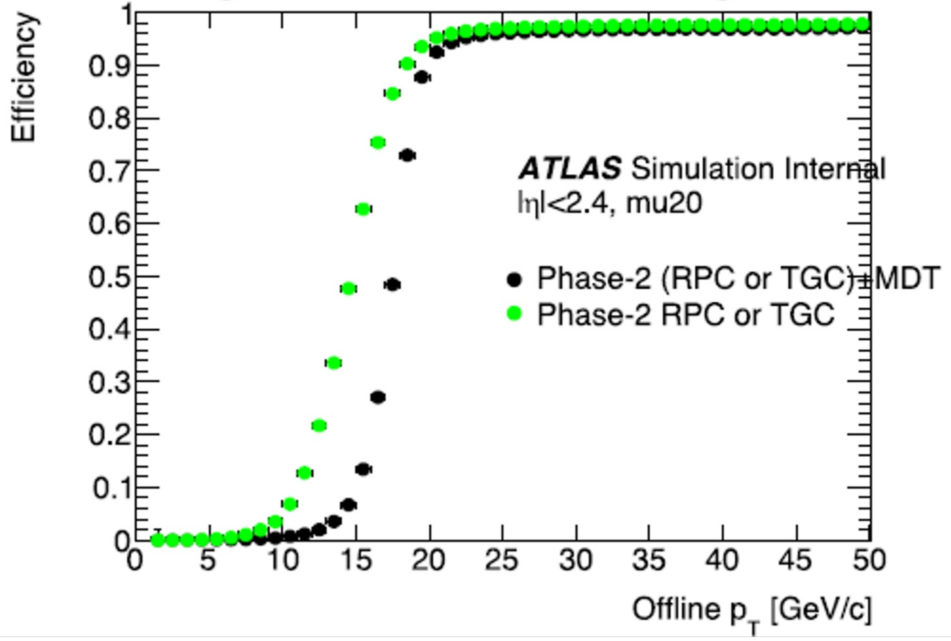




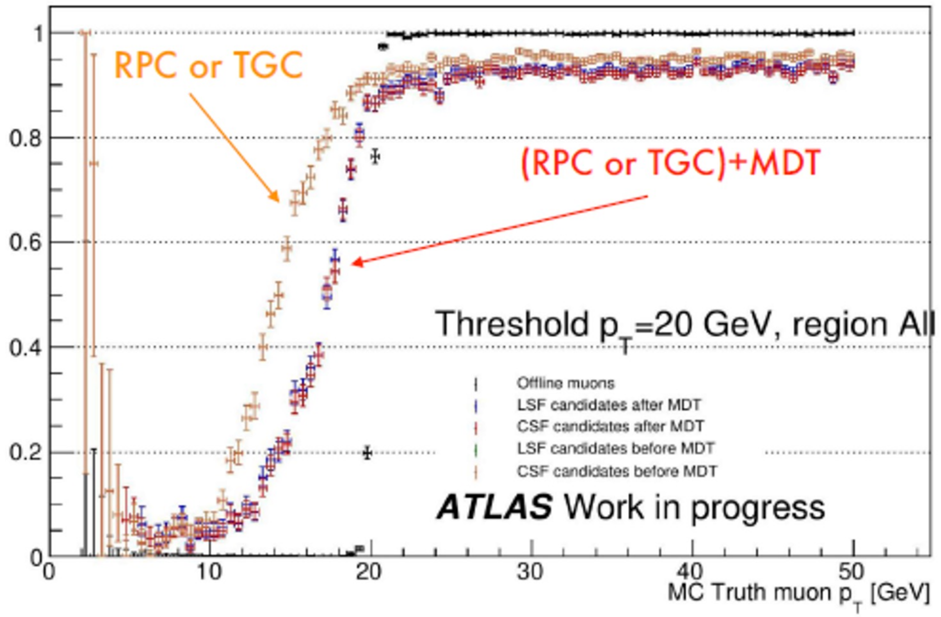
MDTTP Trigger Performance - Efficiency

- **pT extracted from the parametrisation is compared to a pT threshold for the candidate to be accepted/rejected**
  - pT threshold < nominal threshold (i.e. 20GeV) to allow for high efficiency at the plateau
  - pT threshold calculation done separately per combo ID taking into account also the pT resolution for that combination
- **Significant improvement in the sharpness of the efficiency turn-on curve achieved with the introduction of the MDTTP in the L0Muon trigger**
  - Improved performance compared to the TDR plots (new pT parametrisation) and included also the online segments from the MDTTP segment finders
- **Plan to study additional (lower) threshold values**

Single muon + offline MDT segments



Z->μμ + PU + Bkg + online MDT segments



Level-0 MDT Trigger Processor
Firmware Testing–Data Format definition and Test Vector generation

- Firmware validation in simulation or hardware require test vectors to be available for all dataformat definitions in trigger path
- Python utilities have been developed
  - TVMaker: Generate test data from Level 0 Muon offline package suite
  - DataFormat: Package test data into test vectors (for all dataformat definitions in firmware)
    - Test vector file generated for all command module configurations (64 sectors)
  - TVReader: Read required dataformat from a testvector file
- These python utilities are integrated into the L0MDT git repository
  - Dataformat definitions are used in firmware code
  - Test vectors are used in the functional validation framework