

Level-1 Tracking at CMS for the HL-LHC

Sara Fiorendi (University of Tennessee) on behalf of the CMS Collaboration

> Connecting the Dots 2023 October 10-13th

HL-LHC opportunities and challenges

The High-Lumi LHC will provide the experiments with **unprecedented high statistics data**

- extend discovery reach in searches for new physics & rare SM processes
- improve Higgs boson and SM precision measurements

This will happen in a **very challenging environment** for the experiments

- instantaneous luminosity of 5-7 x 10³⁴ cm⁻² s⁻¹
- **•** expected average pileup of 200, resulting increase of particle density
- radiation damage to the detector

Phase-II upgrades of the CMS detector were designed to maintain excellent detection ability, and even improve performance wrt current detector

• **including tracking in hardware trigger** plays a crucial role

• Current trigger systems

• L1 trigger

- Particularly challenging for *• Hardware-based, implemented in custom-built electronics*
- asea, impiermentea ir
vrimatar infarmation v • Muon & calorimeter information with reduced granularity, no tra

CMS trigger uppgrade Journey Continued to Algements Continued Continued

- *• Tracking information & full detector granularity*
- *• ATLAS use level-2 & event filter, CMS single-step HLT*
- The entire trigger system will be replaced for HL-LHC **Tellet in the Sunch spacing and Telec**
- Still based on a 2-level trigger approach to reduce the 40MHz collision rate down to 7.5 kHz
	- hardware Level 1 (L1) trigger
	- software High Level Trigger (HLT)

- **Significant challenge** in data processing • **<PU> = 140, Peak PU = 192 (increase × 6) Example in data processing**
	- huge amount of input data bandwidth (~63Tb/s) mount of input data bandwidth (~63Tb/s)
————————————————————
	- decision window of 12.5μs (4μs for track reconstruction)
- **Tracking information will be used for the first time at L1!**
	- On-detector filtering to reduce hit rate
	- Off-detector track finding algorithm implemented on Xilinx FPGAs

CMS L1 trigger scheme @HL-LHC

Benefits of tracking @L1

- Usage of tracking information in hardware trigger allows to
	- improve **p_T resolution and particle identification** \rightarrow lower trigger thresholds
	- identify primary interaction vertex, **mitigating the pileup** effects
	- **associate objects** to a common vertex
	- perform Particle Flow reconstruction already at L1 (also thanks to the fine calorimeter granularity)

Phase 2 Outer Tracker

- Entire tracker detector will be replaced during LS3
	- increased granularity and pseudo-rapidity acceptance, radiation tolerance, and lower mass
- Outer Tracker (OT) will consist of 6 barrel layers and 2 x 5 disks
	- **tilted geometry** for better trigger performance and reduction in number of modules
	- **PS** and 2S modules provide p_T discrimination in front-end electronics through hit correlations between two closely spaced sensors

Tracker input to the L1 trigger \sim Signals from each sensor are correlated by \sim

• Two kinds of modules (PS and 2S) will be used in different regions of the detector

- Correlated pairs of clusters consistent with a $p_T > 2$ GeV track form a **stub**
	- input to the track finding algorithm
	- cut at 2 GeV will allow a factor ~10 data reduction

L1 tracking system over the VV $\sum_{\text{A. Hart}}$

- Extensive **parallel processing** to cope with high data rate and large combinatorics Extensive **parallel processing** to cope with high data rate and large combinatorics
- takes advantage of natural detector segmentation (9 sectors in φ) $\frac{1}{2}$ σ the Throughout algorithm, σ bectors σ , τ \in segmentation (9 sectors in Φ) $\hspace{1cm}$ layer/disk pair \mathcal{L} viding communication channels to \mathcal{L} subset contraction blades. These contractions \mathcal{L} \bullet takes advantage or natural detector segmentation (9 sectors in ϕ)
- further within-sector parallel processing dividing ϕ into "virtual modules" • further within-sector parallel processing dividing φ into "virtual module
- use of time-multiplexing (x18) to implement multiple identical key to minimize combinatorics $\mathcal{L}_{\mathcal{A}}$ firmware combinatorics $\mathcal{L}_{\mathcal{A}}$ firmware combinatorics $\mathcal{L}_{\mathcal{A}}$ • use of time-multiplexing (x18) to implement multiple identical $\left\langle \left. \right\vert \left. \right\rangle \left. \right\rangle$
- Flexible and scalable architecture \overline{Eff} by \overline{Eff} by \overline{Eff}

Eff by

 $\Delta \varphi$ (R=R

A. Hart

critical radius

'F⊥∏TDF8JPU+FSP

Muon Track Finding

duplicated

 $p_T > 2$ GeV

unique

Track finding algorithm (1) \bullet is to reconstruct all \bullet \bullet all \bullet 2.4 \bullet 2.4

Road search algorithm based on tracklet seeds • *Hybrid algorithm*

Emulation ⇔ Firn

- 2. **Track parameters** initially estimated from *tracklet* + beamspot constraint **• Goal: Systematic** large-scale **comparison between firmware & emulation**
	- only combinations with $p_T > 2$ GeV kept ‣ Now **large-scale, sequential event processing**, **Inations with** $p_T > Z$ **GeV kept**oject tracklets to other layers
edisks to soarch for matching stubs reproject trackiets to other layers
& disks to search for matching stubs f kantaigat tradition to other
- 3. **Project** potential track to other layers/disks **and a constants compatible stubs within predefined narrow windows** otential track to other layers/disks fracks a the bigger of Search windows derived from tracklets & stubs *<u>&* search</u>
	- propagation both inward and outward Both inside-out & outside-in
	- minimum number of stubs required • Develop SW tools for large-scale comparisons where \mathcal{L} bitwise comparisons in the set of the set

 $\frac{1}{2}$

Track finding algorithm (2)

Puplicate removal and fitting track can be merginal track can be seen and fitting TRACK FINDING ALGORITHMS 5

4. **Removal of duplicate tracks 3D KALMAN FILTER (KF)**

x

- pattern recognition produces multiple track candidates per each \sim Similar performance \sim demonstrated fearing feasibility, denote the phase-2 Tracker Tra \cdot **measurements** containing inaccuracies and noise ->
- **•** redundant seeds ensure high efficiency, but lead to duplicate tracks en de search algorithm based on "tracklet" seeds on "tracklet" seeds the seeds of tracklette seeds the seeds o
Tracklette seeds the seeds of tracklette seeds the seeds of tracklette seeds the seeds of the seeds of the see

litional duplicates may originate from combinatorial stubs $\textsf{H}^\textsf{t}_\textsf{t}$ candidates may originate from combinatorial stubs

ερμ<mark>, B. Yates</mark>, replicated tracks are joined into a "merged" track candidate $\sum_{k=1}^{\infty}$ **Tracklet Based Track Finding**Ls Yates,

‣ Previously single event comparisons from pairs of stubs in nom pane or etabe in
neighboring layers _o track ‣ (1) Compare emulation vs Vivado simulation • iterative appro Ω^{no} Candidate tracking • iterative approach: starts with tracklet parameters & y Form track seeds, tracklets, neighboring layers e track is finally^{fited} with a Kalman Filter algorithm y iterative approach, starts with trackles parameters α
uncertainties, then use matched stubs to update the

tracklet

stub pair

J. Chaves, LS

tracklets & study and tracks track parameters

4. Repeat until all stubs are added Tracklet seed & search Kalman Filter fitting

Emulation ⇔ **Firmware Comparisons (1)**

• Goal: Systematic large-scale **comparison between firmware & emulation**

Expected performance

- Expected tracking performance estimated on simulated events
	- high efficiency across η and p_T
	- precise z_0 resolution (~1mm in the barrel), necessary for vertex association

Track quality

- An additional track quality module will be run after the Kalman Filter step to reduce number of tracks not coming from genuine charged particles
- Using a ML approach to classify real/fake tracks, outperforms simple cut based selection (★)
	- features from reconstructed track parameters: φ , η, z_0 , n_{stub}, n_{misslayer}, χ^2 _{bend}, χ^2 _{rz,} χ^2 _{r φ}
	- GBDT chosen over NN as less FPGA-resource hungry

Hardware platforms

Dual VU13P

• Hardware for track-finding based on ATCA platform (standard for HL-LHC upgrade)

[PoS TWEPP2018 \(2019\) 115](https://serenity.web.cern.ch/TWEPP18_Serenity_Rose.pdf) 2022 JINST 17 C04033

APOLLO: track finding processing boards

- Service Module provides infrastructure components
- Flexibility via pluggable Command Module: contains two large FPGAs, optical fiber interfaces & memories

Figure 4. The Apollo revision 2 CM witho[ut](https://iopscience.iop.org/article/10.1088/1748-0221/17/04/C04033/meta) [any](https://iopscience.iop.org/article/10.1088/1748-0221/17/04/C04033/meta) [FPGAs.](https://iopscience.iop.org/article/10.1088/1748-0221/17/04/C04033/meta) [Maj](https://iopscience.iop.org/article/10.1088/1748-0221/17/04/C04033/meta)or components are labeled and highlighted

ards)

Firmware implementation complete track the complete track track track the complete track that \mathcal{L}

- Implemented as alternated processing and memory modules memory modules
- Multiple copies of each module run in parallel **module for the control**
- Seeding & propagation steps written using Xilinx Vivado HLS $\frac{1}{2}$ proposation stops amg & propagaa
tan using Yiliny V
- Memory modules, Kalman Filter and top level written in VHDL memory modules, kalman filter nory modules, Kamilan inter
Level with video in VHDL
- Targeting 240 MHz FPGA clock

Narrow slice project \blacksquare

- End-to-end demonstration of the track finding chain on a narrow ϕ slice
	- based only on one (barrel) seed • Goale: based only on one (barrel) seed
	- does not include the duplicate removal step
- Demonstrated on Apollo board rev1 $\frac{1}{2}$

VU7P

Filter *Filter*

Full barrel project Full project and project and the set of \mathbf{P}

- Seeding & stub matching in barrel layers, **~2/3 of the full project**
	- implemented in single VU13P FPGA
		- final project will use two VU13P
	- **meeting timing requirements was challenging**
		- exploited machine learning based Vivado firmware implementation strategy
	- **floorplanning** to avoid signals crossings regions with dead silicon interconnections
	- using **combined modules** to reduce latency

• Currently working on integrating the full chain of modules for the entire detector **Figures** for the entire detector

Summary

- **L1 track finding will be crucial @HL-LHC** to maintain acceptable trigger rates while successfully pursuing CMS physics goals
- Main challenges related to the large combinatorics and latency
	- CMS will use a **unique detector design with p_T modules** providing on-detector data filtering
	- **extensive parallelisation** being exploited for the off-detector track finding algorithm (on FPGAs)
- Current status:
	- **reduced configuration firmware was successfully tested**
	- **ongoing** work to integrate the **full chain** covering the entire detector on two FPGAs

backup

Combined modules $\mathcal{P}(\mathcal{$

• Moving towards combined modules → fewer processing modules help in reducing the latency • Moving towards co mouur

Displaced tracking

- Extended tracking being studied in order to reconstruct trajectories not pointing to the PV
- Changes wrt baseline tracking algo impact:
	- seeding step: triplets instead of doublets + origin
	- Kalman filter: 5-parameter fit instead of 4-par. (+ *d0*)

Track quality

• Resource usage for NN and GBDT **Figure 5:** Difference between the reconstructed and simulated pri-Figure 6. Published CDDT arce usage for this a

 $\overline{\mathcal{L}}$ with 150 trees and a maximum depth of 4 that is only trained on displaced tracks. This only tracks. T

https://agenda.infn.it/event/28874/contributions/168841/attachments/93290/127232/ICHEP_2022_Poster.pdf

• Performance on displaced tracks of the baseline GBDT, compared to a possible dedicated displaced GBDT **Track Quality Application: Primary Vertex Reconstruction** GBDT has the same features as the classifier in Section 4 with the addition of a displacement \mathbf{T} and the chain chain: Track variative 30. applaced tracks of the baseline GBD i, compared
GRDT

