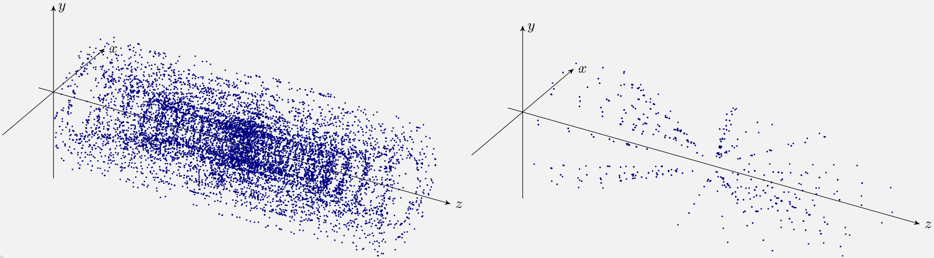
CMS Phase-2 Trigger Upgrade

Using a track finder at Level-1

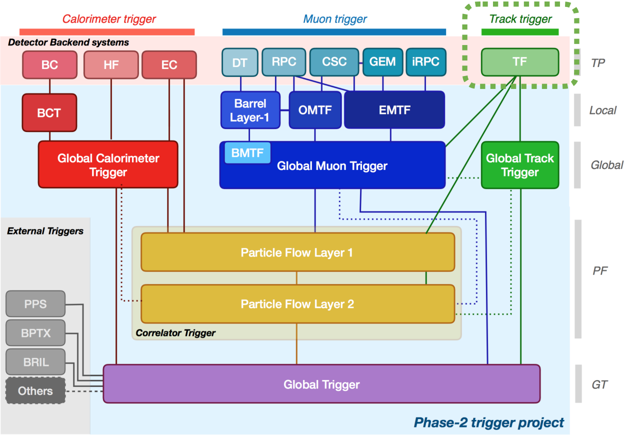
Thomas Schuh

2nd March 2023

RUTHERFORD APPLETON LABORATORY



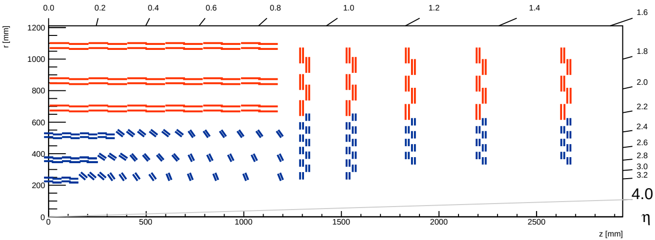
CMS L1-Trigger Phase 2



- L1 tracks must be used to associate trigger objects with single vertices
- vertices are formed in global track trigger
- global trigger may associate trigger objects with single vertices
- additional Particle Flow begins with tracks and vertices to create trigger objects

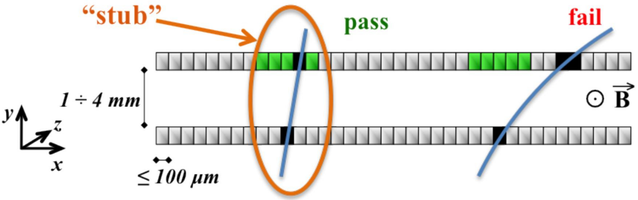
Phase II Outer Tracker

- replacement of entire outer tracker
- 13 200 modules, 190 m^2 silicon, 213 M channel (legacy: 10 M channel)
- 6 barrel layer plus 2×5 endcaps discs



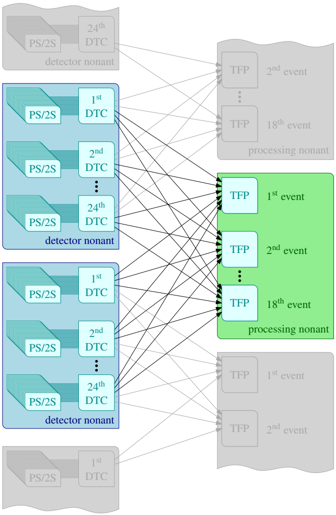
- two kinds of p_{T}-modules:
  - strip-strip (2S): 5 cm×90 µm
  - pixel-strip (PS): 1.6 mm / 2.5 cm×100 µm
- constructed to enable online track reconstruction above 2 GeV

Stubs – on detector data reduction



- new modules allowing rough p_{T} estimate
- cut at 2 GeV reduces data rate by $\sim$ one order of magnitude
- p_{T}-information is useful during track reconstruction

Outer Tracker Back-End System



outer tracker

- organised in detector nonants
- each nonant read out by 24 DTCs
- each DTC connected with up to 72 modules

track trigger

- organised in processing nonants
- shifted by half a nonant w.r.t. detector nonants
- each nonant processed by 18 TFPs
- each TFP connected with up to 48 DTCs

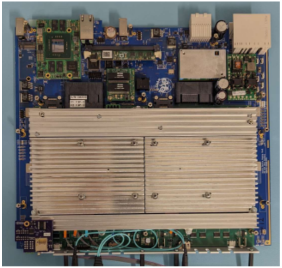
Data, Trigger & Control Board (DTC)

Serenity-S1

- hosts a VU13P, -2, A2577 (128 GTY)
- uses 72 links @2.56, 5.1/10.24 gbps towards detector
- and 36 @25 gbps towards back/end

- I am not covering: lpGBT, DAQ, slow Control, calibration, . . .
- focus on stub processing:
  - sensor modules sending stubs in 8 BX boxcars
  - DTC repacks them into 18 BX long trains containing only stubs from single events
  - and assigns stubs to processing nonants

Track Finding Processor (TFP)



Apollo

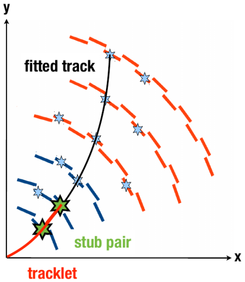
- hosts up to two VU13P, -2, A2577 (128 GTY)
- gets stub over 48 links @25 gbps from DTCs
- sends tracks over 3×2 links @25 gbps downstream

CMS L1 track trigger requirements

- reconstructs prompt tracks above 2 GeV
- covering $|\eta| \leq 2.4$
- allowing beam spot window of ± 15 cm in beam direction
- within 4 µs latency

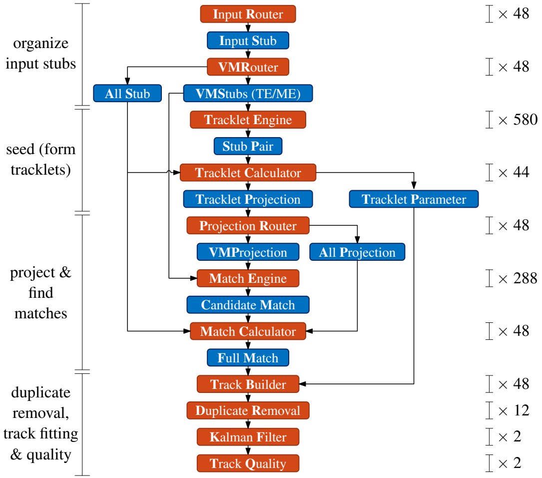
CMS L1 track trigger algorithm

- track finding through road search based on tracklet seeds
- multiple seeding layer combinations used for redundancy and high efficiency
- found tracks sharing stubs get merged (duplicate removal)
- track fit uses kalman filter
- BDT provides qualifiers for improved fake reduction



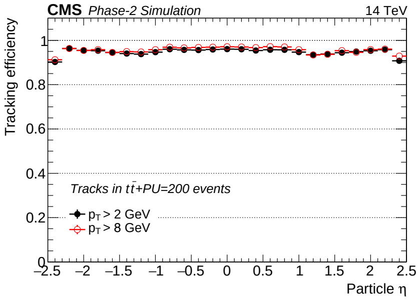
CMS L1 track trigger algorithm – Implementation

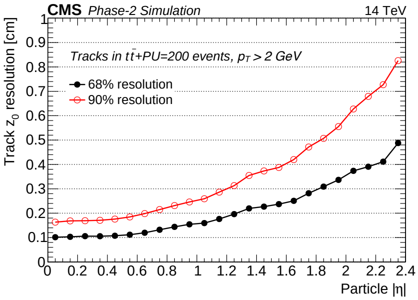
- track finding organized in alternating processing and memory modules
- multiple copies of each module run in parallel
- most processing modules written in HLS
- memory modules, kalman filter and top level written in VHDL
- targeting 240 MHz
- f/w nearly complete



CMS L1 track trigger algorithm – Performance

- examples of expected L1 tracking performance based on simulation
- high efficiency across p_T/η
- precise z_0 resolution for vertex association



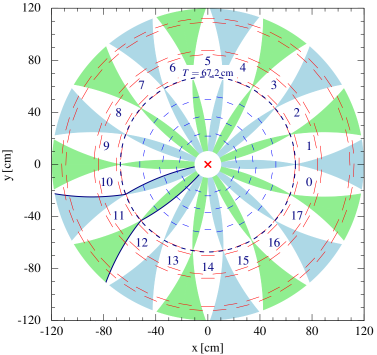


Exploration of an alternative L1 Tracking Algorithm

- benefit of an all FPGA system: one can explore alternatives
- alternative track reconstruction chain:
  - GP: spatial sectoring
  - HT: track finding in r-ϕ
  - MHT: finer track finding in r-ϕ
  - ZHT: finer track finding in r-z
  - TB: track builder
  - KF: kalman filter fit
  - DR: duplicate removal
- currently targeting single VU13P running @ 360 MHz
- simpler and faster as CMS L1 track trigger algorithm
- based on already working f/w
- reconstructs however only prompt tracks above 3 GeV

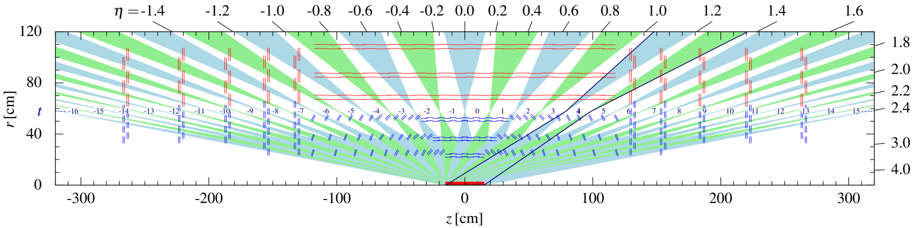
the approach described in the following is one possible implementation that is not at this moment the CMS baseline

Geometric Processor – Divide and Conquer

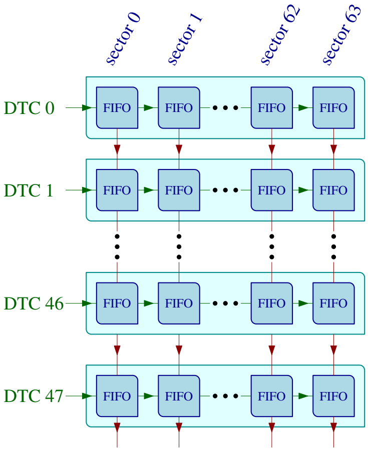


- division of each nonant in $2\,\phi \times 32\,\eta$ sectors
- stub duplication at sector boundaries leads to:

  13 k stubs $\rightarrow$ 26 k stubs in $t\bar{t}$@200PU
- enables parallel track-finding per sector



Geometric Processor – Implementation



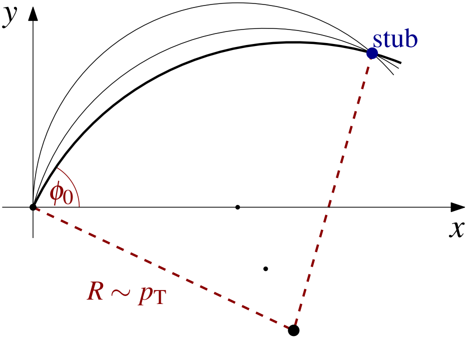
- 336 ns latency
- utilization:

LUT	18 %
LUTRAM	23 %
FF	11 %
BRAM	2 %
DSP	1 %

- bit and clock accurate emulator written
- ~ 1 ‰ stub loss due to truncation in $t\bar{t}$@200PU events

Hough Transform – Theory

- search for primary tracks in the r-ϕ plane
- infinite number of circles $\left(\phi_0, \frac{q}{p_\mathrm{T}}\right)$ consistent with beam-line & any individual stub position (r, ϕ)

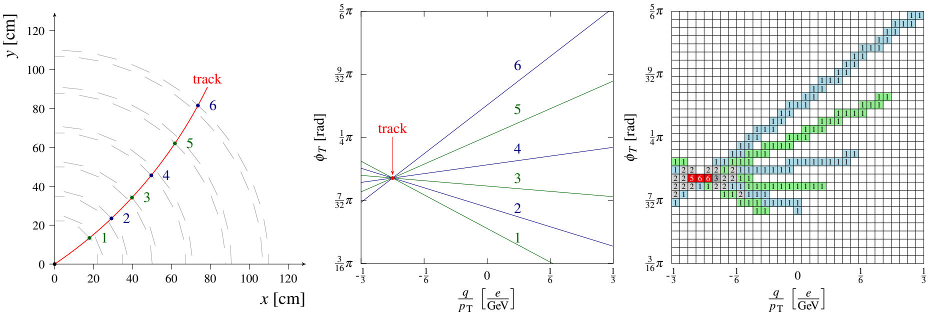


- they must obey constraint:

$$\phi_0 \quad \approx \quad \phi + \frac{q}{p_\mathrm{T}} \times r$$

- **stub positions corresponds to straight lines in the track parameter plane**

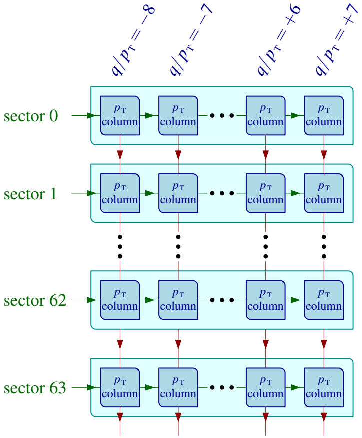
Hough Transform – Algorithm



per sector:

- calculate $\phi_0\,(q/p_{\mathrm{T}})$ for all stubs within the sector
- fill stubs in the corresponding cells of a ϕ_T - q/p_{T} - histogram (32×16)
- define cells with stubs from at least 5 layer as track candidate

Hough Transform – Implementation



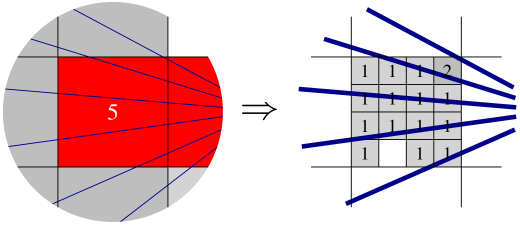
- 706 ns latency
- utilization:

LUT	20 %
LUTRAM	17 %
FF	13 %
BRAM	20 %
DSP	8 %

- bit and clock accurate emulator written
- data reduction: 25.6 k stubs → 2.7 k stubs

Mini Hough Transform (MHT)

- analyze found tracks with 4×4 $r - \phi$ mini array
  - adaptive track parameter space of mini array covers candidate's rough HT cell
  - takes stub ϕ uncertainty into account
  - uses 4 layer threshold to define found tracks
- does not create a track for each cell above layer threshold
  - instead removes stubs which are not found in the finer track collection
  - removes track entirely when no finer track has been found



- 67 ns latency
- utilization:

LUT	<1 %
LUTRAM	≪1 %
FF	<1 %
BRAM	1 %

- 2.7 k stubs → 1.3 k stubs
- 351 tracks → 196 tracks

Z Hough Transform (ZHT)

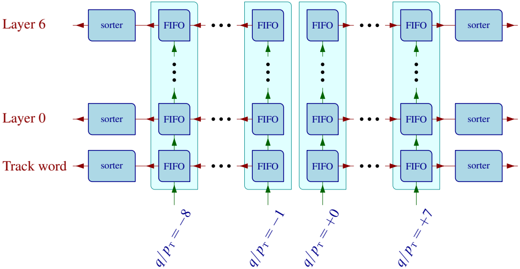
- identical to MHT but looking at $r-z$ plane instead of $r-\phi$
- adaptive track parameter space of mini array covers $r-z$ sector
- 67 ns latency
- utilization:

LUT	<1 %
LUTRAM	≪1 %
FF	<1 %
BRAM	1 %

- 1.3 k stubs $\rightarrow$ 1.1 k stubs
- 196 tracks $\rightarrow$ 159 tracks

Track Builder (TB)

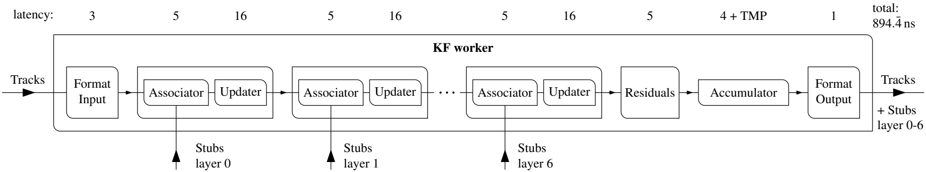
- track builder only restructures the data flow
- input: 16 streams of stubs
  - one stub per clock tick
  - a track is a sequence of stubs
  - contains gaps
- output: 2 streams of tracks
  - a track consists of a track word and 7 stub words (one per layer)
  - layer id are counted along found track from inside out
  - we allow up to 4 stubs per layer: one track takes between 1 and 4 clock ticks
  - gap-less



- 572 ns latency
- utilization:

LUT	1 %
LUTRAM	<1 %
FF	<1 %
BRAM	3 %

Kalman-Filter (KF)



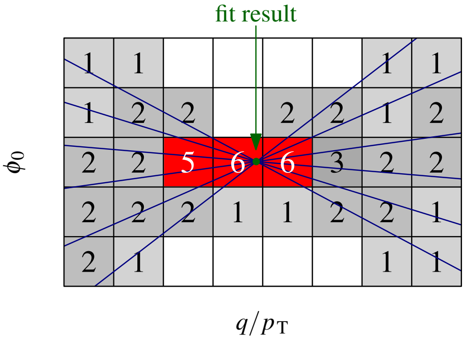
- iterative track fit also used offline and HLT
- tuned for CMS L1 track trigger algorithm
  - no cuts during iterations
  - no final cuts
  - good efficiency with CMS L1 track finding
- need re-implementation of combinatorics reduction strategies

- 894 ns latency
- utilization:

LUT	1 %
LUTRAM	<1 %
FF	1 %
BRAM	6 %
DSP	3 %

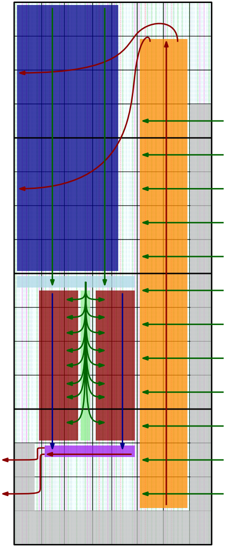
Duplicate Removal (DR)

- example: HT finds 3 candidates out of 6 stubs from one particle
- track fit delivers almost same track parameter for these 3 candidates



- DR removes candidates with inconsistent fitted and found parameter
- much simpler as conventional algorithms (comparison of track pairs)

Alternative L1 Tracking Implementation



- VU13P floorplan
  - Geometric Processor
  - Hough Transform
  - Mini Hough Transforms
  - Track Builder
  - Kalman Filter (FW being refined)
  - Duplicate Removal (FW being refined)
- total latency: $\sim 2\,750\,\text{ns}$

Conclusion

- this talk focused on alternative track finding implementation
- ongoing CMS L1 track trigger algorithm work and improvements are not covered
- targeting complete alternative track reconstruction chain demonstration in next months
- costs, latency and performance of alternative TFP seems to be in required margin
- alternative TFP is neither baseline nor backup though