Real-Time Triggering applied to new Tracking Systems

Ian Tomalin (RAL/UK)

Real-time track-finding: talk Outline

- **Introduction**
  - HL-LHC, CMS upgrade & trigger basics.

- **Real-time tracking at HL-LHC.**
  - Why bother?
  - Tracker design & readout considerations.
  - Track reconstruction: strengths & weakness of various successful solutions.
    - — Using FPGAs or custom integrated circuits.

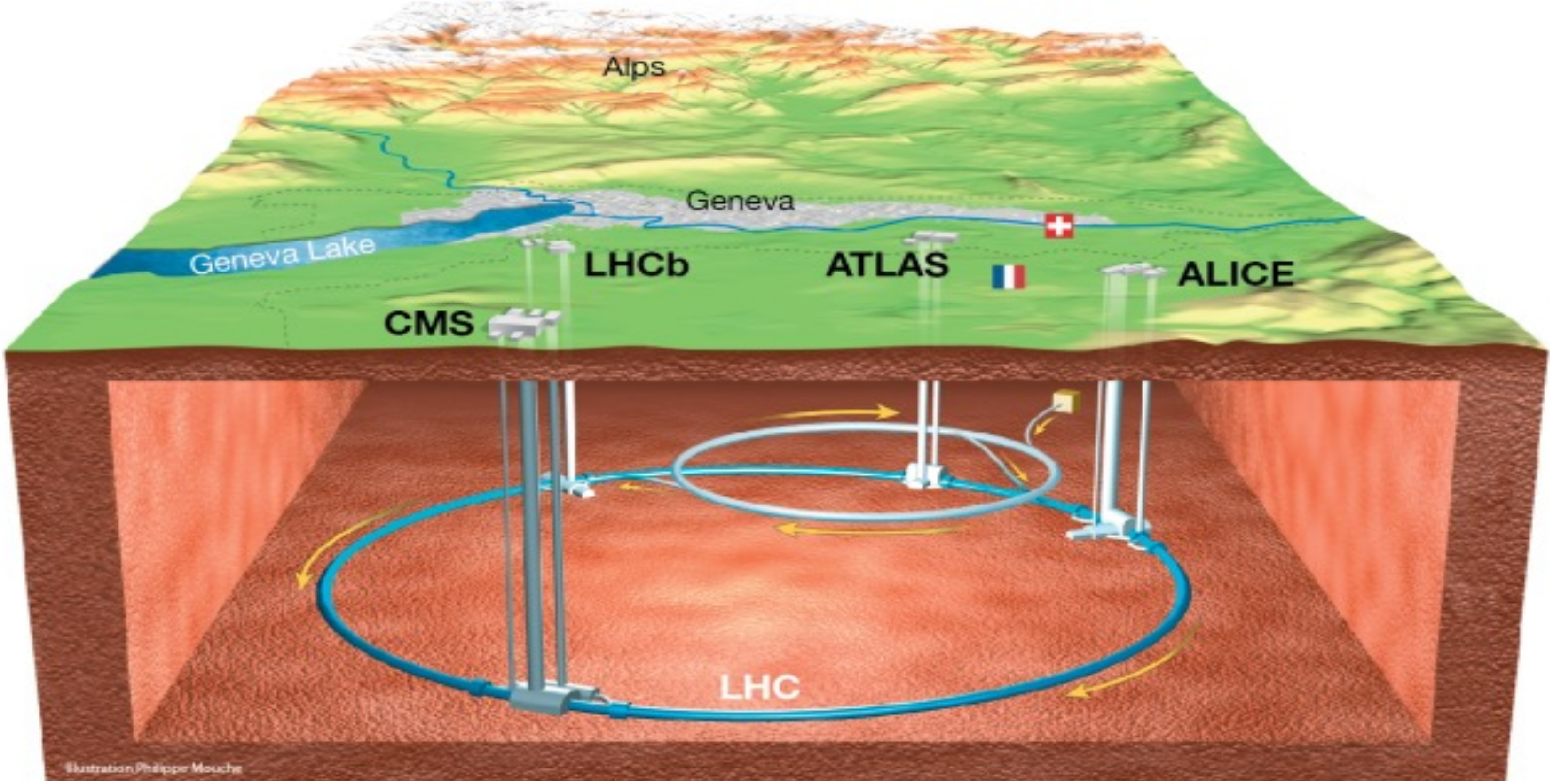
I'll focus mainly on CMS upgrade for the High-Luminosity-LHC.

- *Most advanced & challenging real-time tracking use case.*
- *ATLAS no longer using real-time tracking in L1 trigger.*

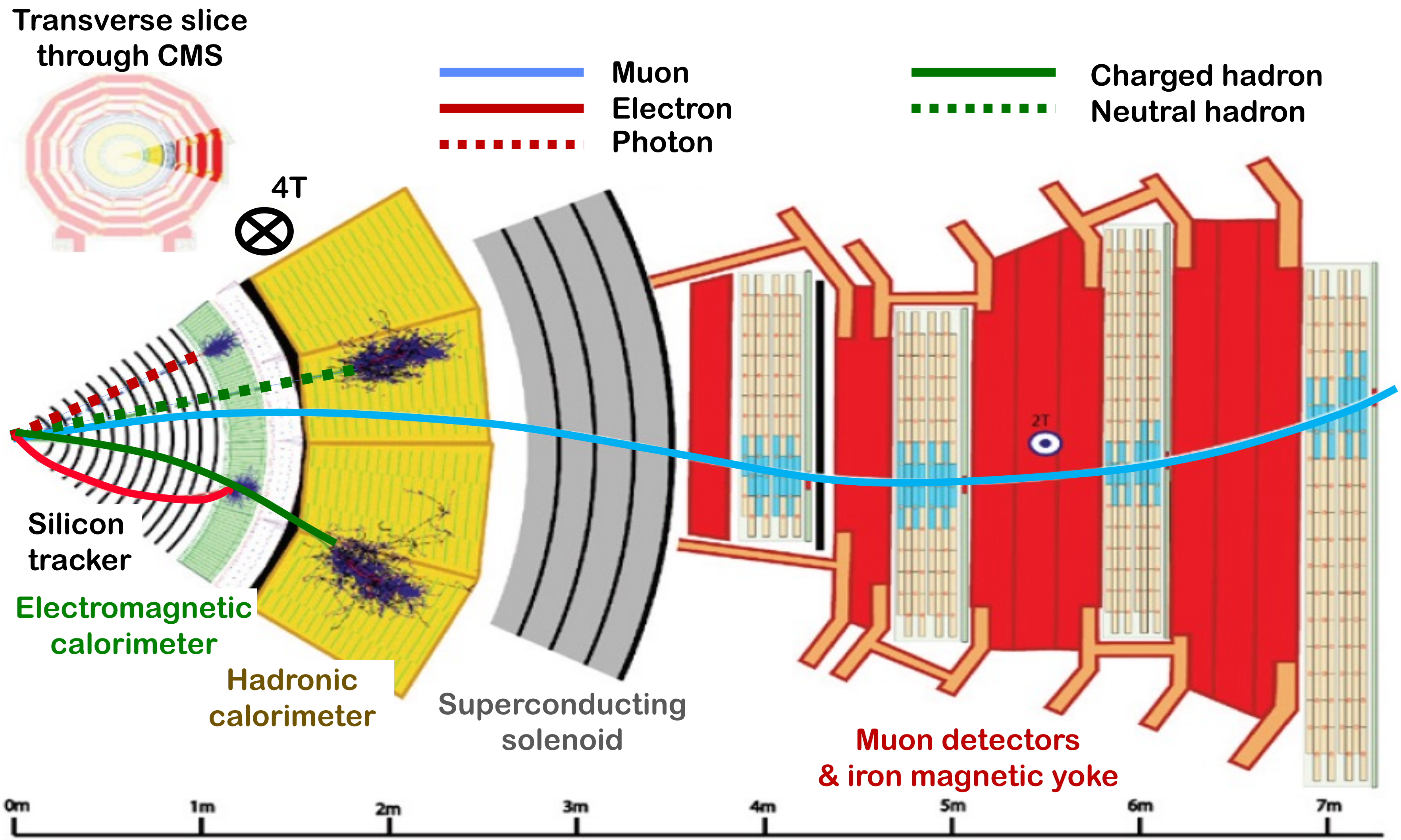
Introduction

The Large-Hadron-Collider (CERN)

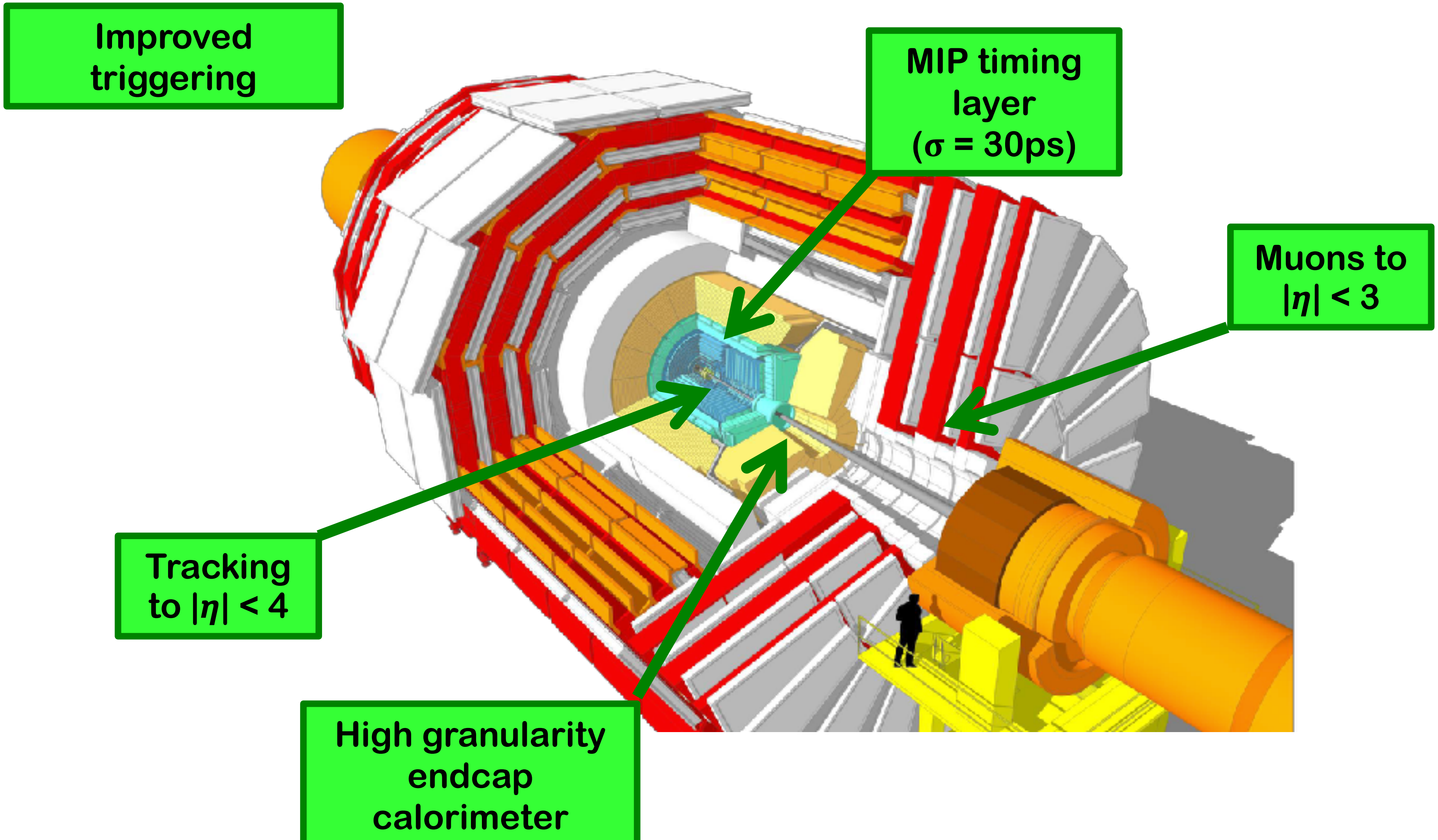
- 14 TeV proton-proton collider, famous for Higgs boson discovery.
  - HL-LHC to start in 2027, with 5 – 7.5 times nominal LHC luminosity.
- CMS & ATLAS = general purpose detectors.



Particle detection in CMS detector
(ATLAS similar)

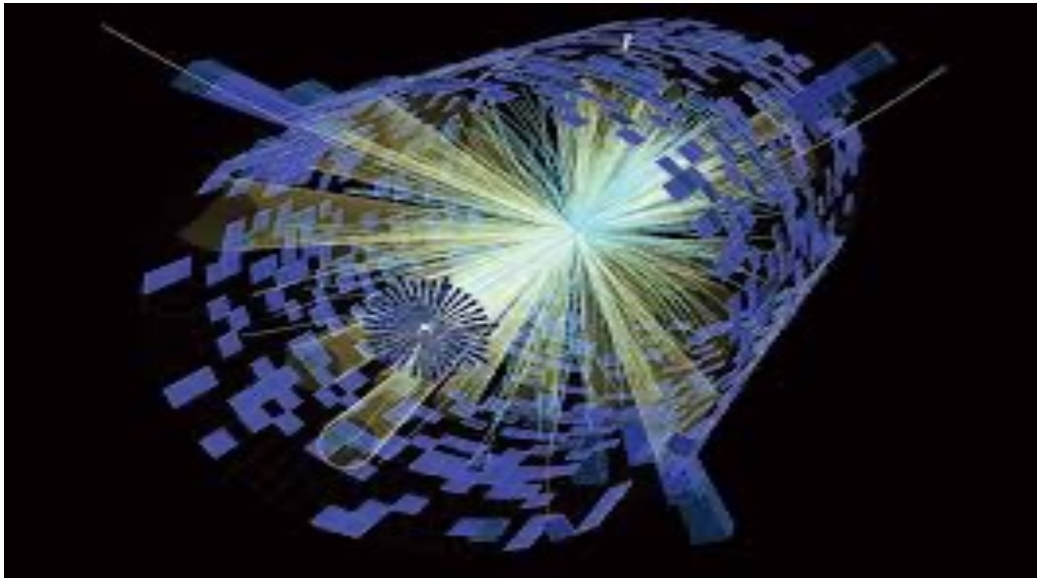


Key CMS upgrades for HL-LHC



How CMS trigger at HL-LHC triggering works
(ATLAS similar)

- LHC proton-proton bunch collisions every 25ns.
- CMS event size ~1 Mb.



- Impossible (& undesirable) to write every event to disk
  - And don't have opto-fibre bandwidth to transmit all data off-detector.

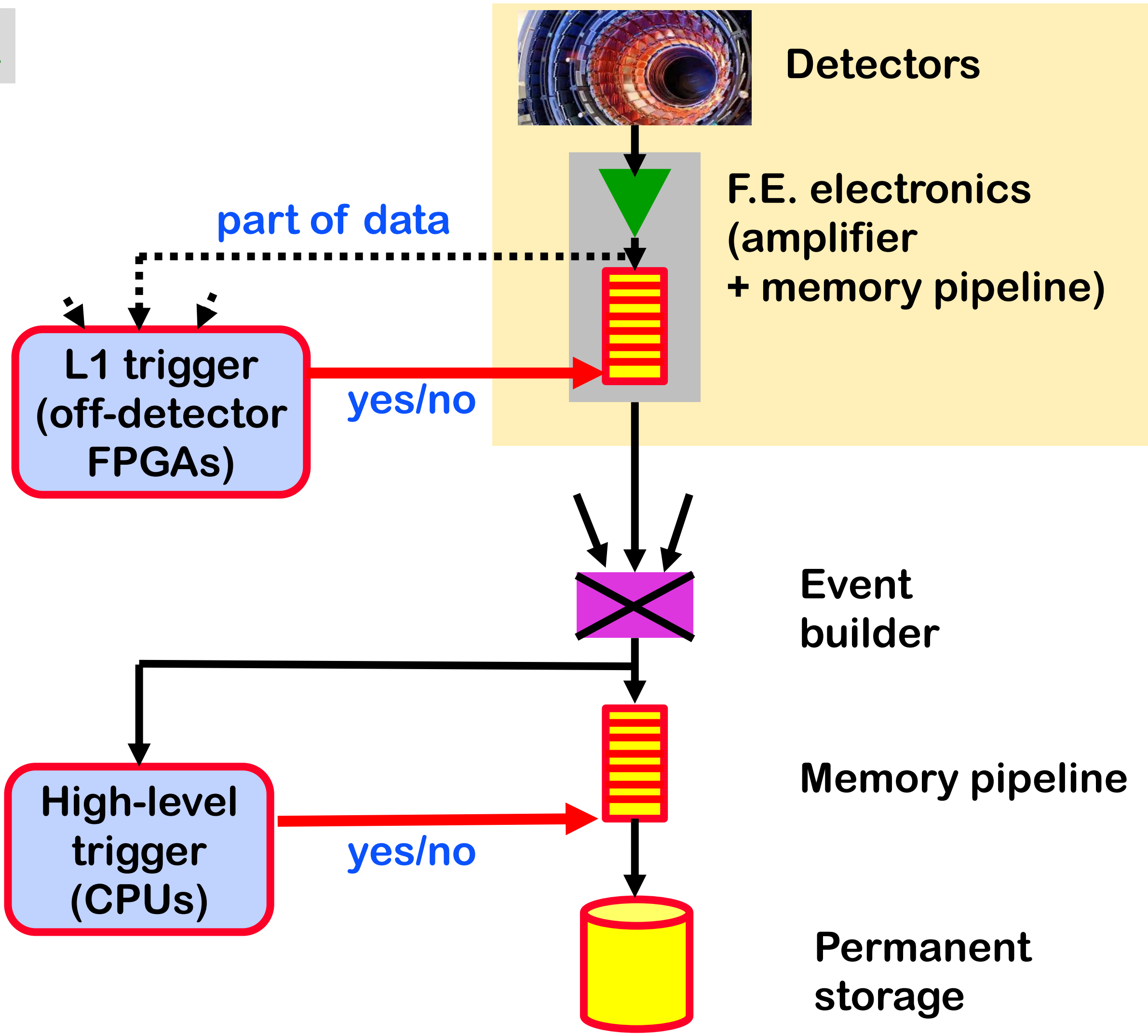
- Two-stage trigger system selects interesting events:
  - Level 1: Using electronics (FPGAs …)
  - High-Level Trigger: Using CPU/GPU farm.

How CMS triggering at HL-LHC triggering works (ATLAS similar)

LHC collisions at 40MHz

L1 trigger:
decision time=12.5 μs
"yes" rate=750kHz

HLT:
decision time ~300ms
"yes" rate=7.5 kHz

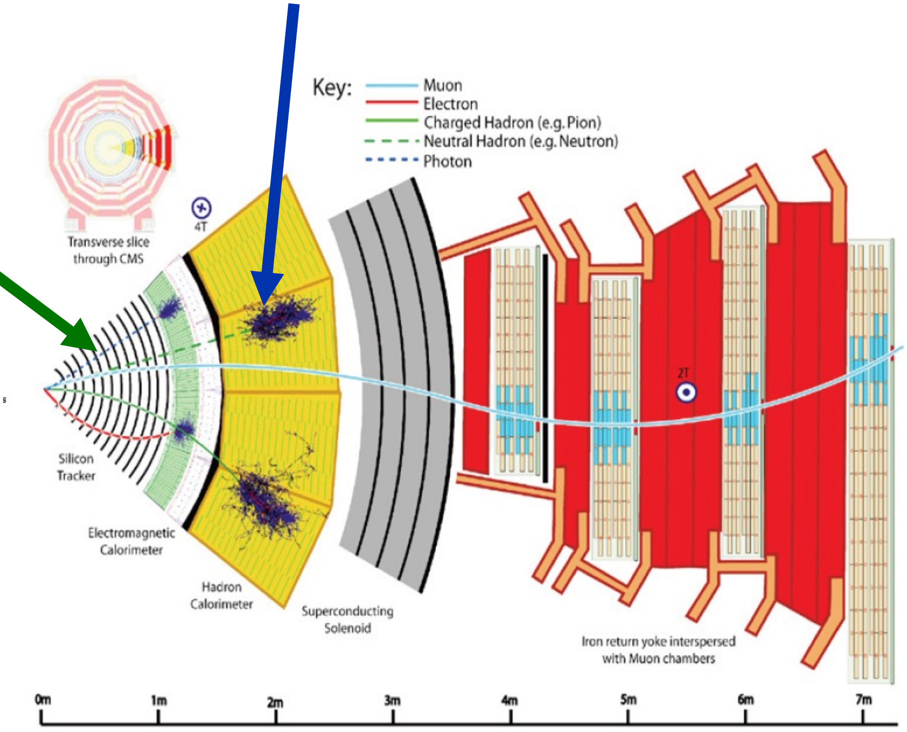


Ingredients of CMS/ATLAS L1 triggers

- Traditional L1 trigger decision uses **Calorimeter** & **Muon Chamber** data. Conceptually easy:
  - e.g. γ trigger: require energy in few neighbouring ECAL towers to exceed threshold.

- ***New*** **for CMS at HL-LHC**:
  - L1 trigger uses Tracker tracks.
  - Difficult!

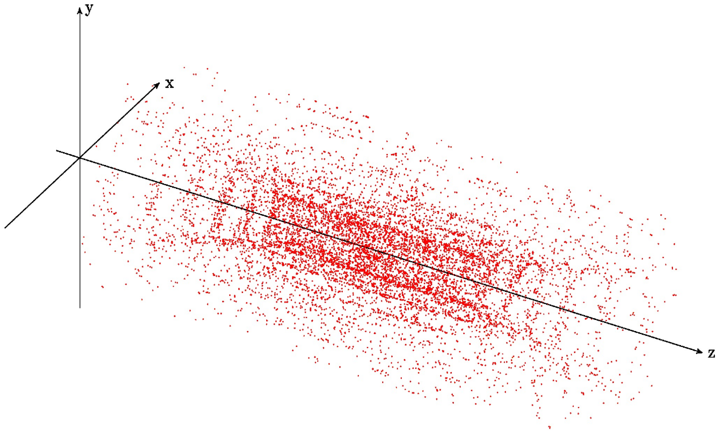
- How cope with data rate?
  - Use fast FPGA electronics.
  - Parallel processing of small angular regions of CMS/ATLAS.
  - Parallel processing of successive events



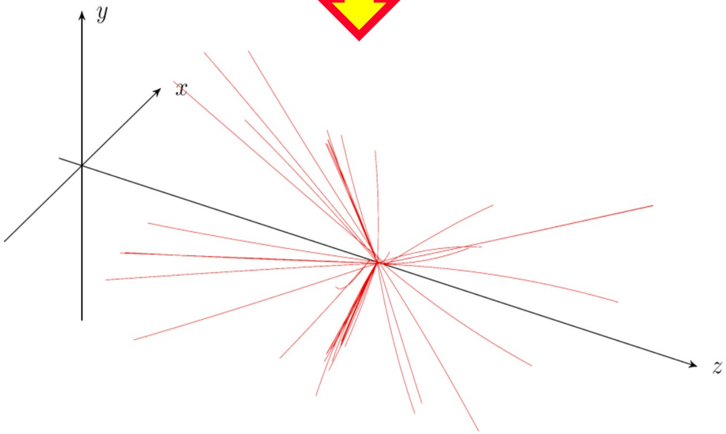
Real time-tracking with CMS at HL-LHC

What is real-time tracking at CMS?

- Giant game of join the dots.
  - Goal to reconstruct all particles (Pt > 2 GeV) from all LHC collisions (40MHz).



- Each collision event has:
  - ~200 particles of Pt > 2 GeV.
  - ~20k tracker measurement points ("stubs").



- 12.5μs / event available for L1 trigger decision:
  - tracking can use 4μs / event.
  - c.f. offline tracking 1 million times slower!

Challenging!!!

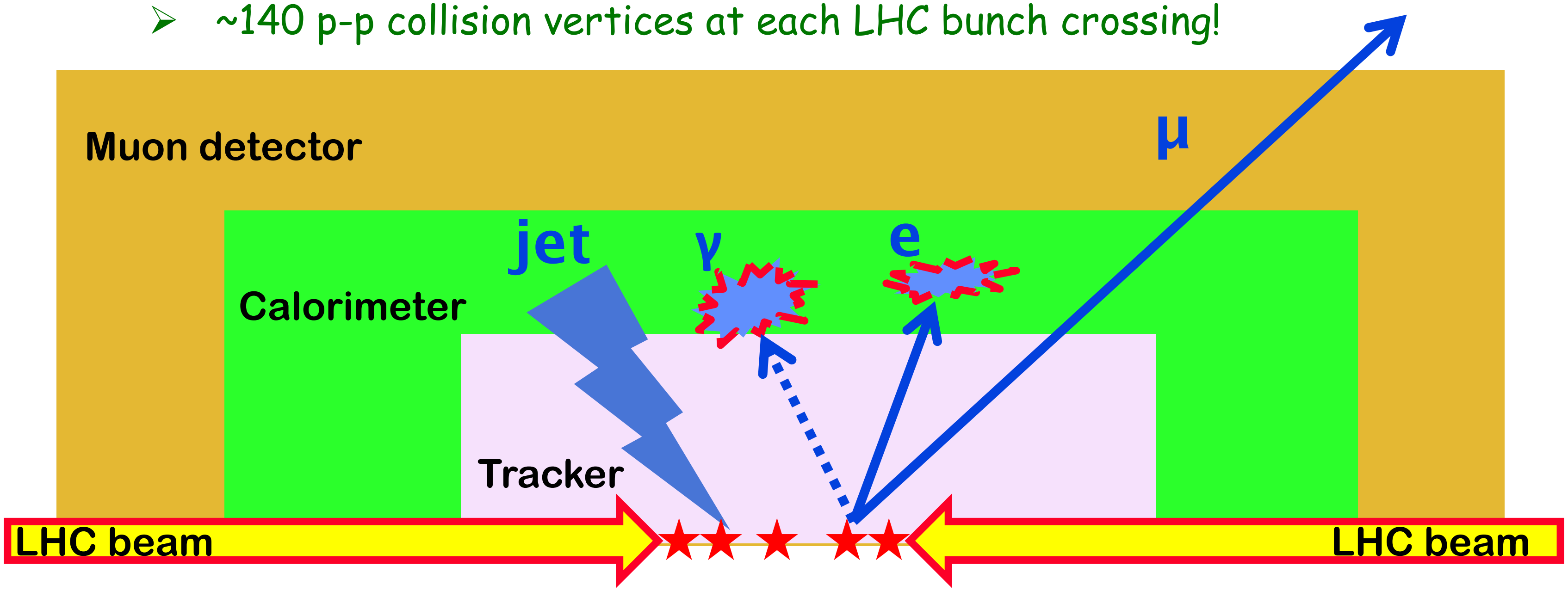
Motivation for including Tracker in L1 trigger decision

❖ **Muons**: tracks improve P_T resolution.

❖ **Electrons**: tracks distinguish them from photons.

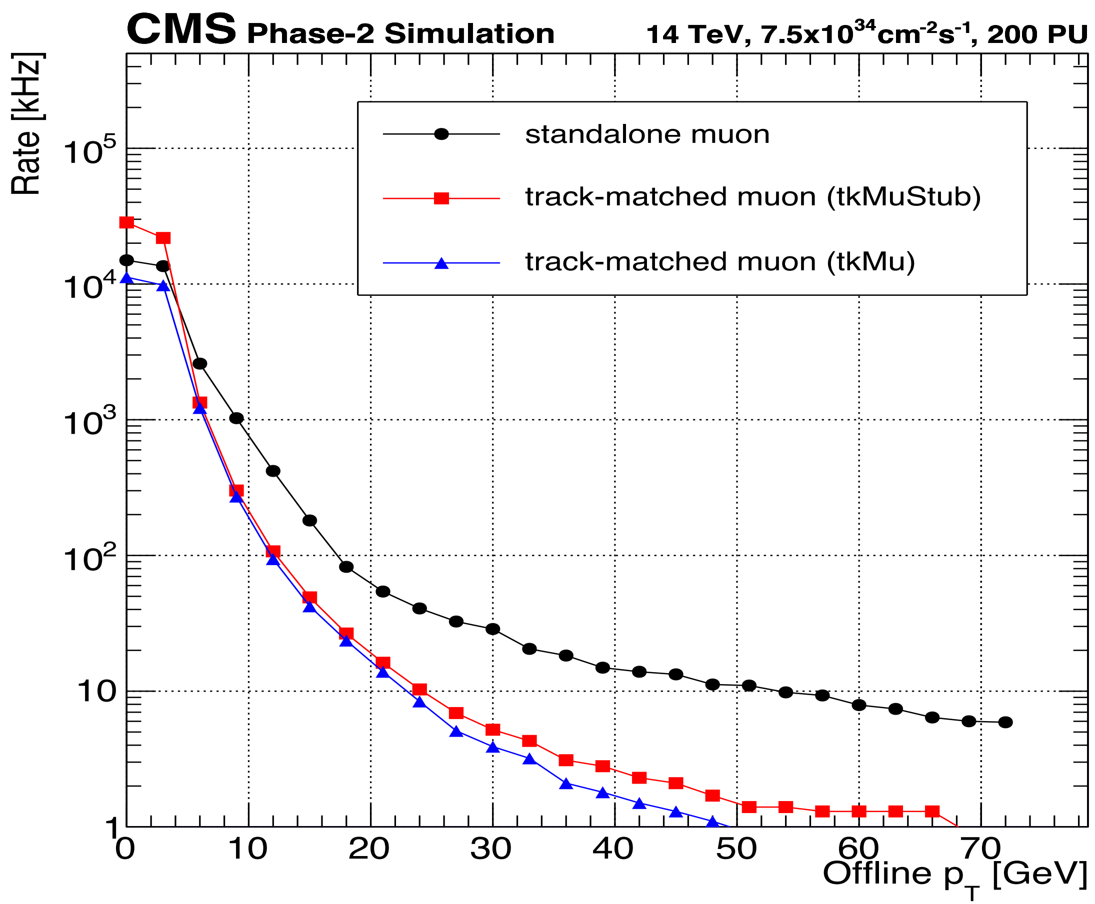
❖ **Jets**: use tracks to help reconstruct jets:

1) Better p_T resolution than calorimeters alone. ("particle flow")
2) Rejects of jets/particles from boring "pile-up" p-p collision points.
   - ~140 p-p collision vertices at each LHC bunch crossing!



Examples of how real-time tracking improves CMS HL-LHC L1 trigger performance

Total L1 trigger rate would be factor ~5 higher without Tracker tracks, for given efficiency.

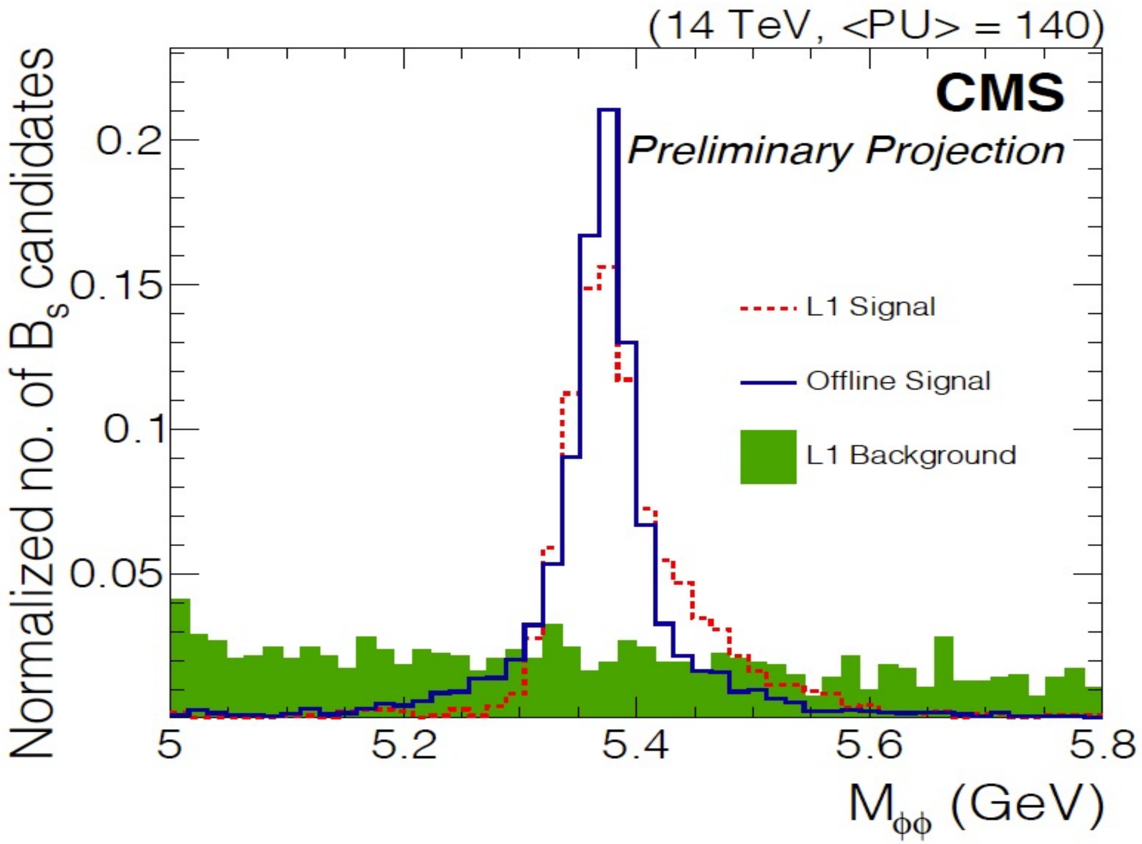


Example:

Reduction in muon trigger rate when using Tracker tracks.

Examples of how real-time tracking improves CMS HL-LHC L1 trigger performance

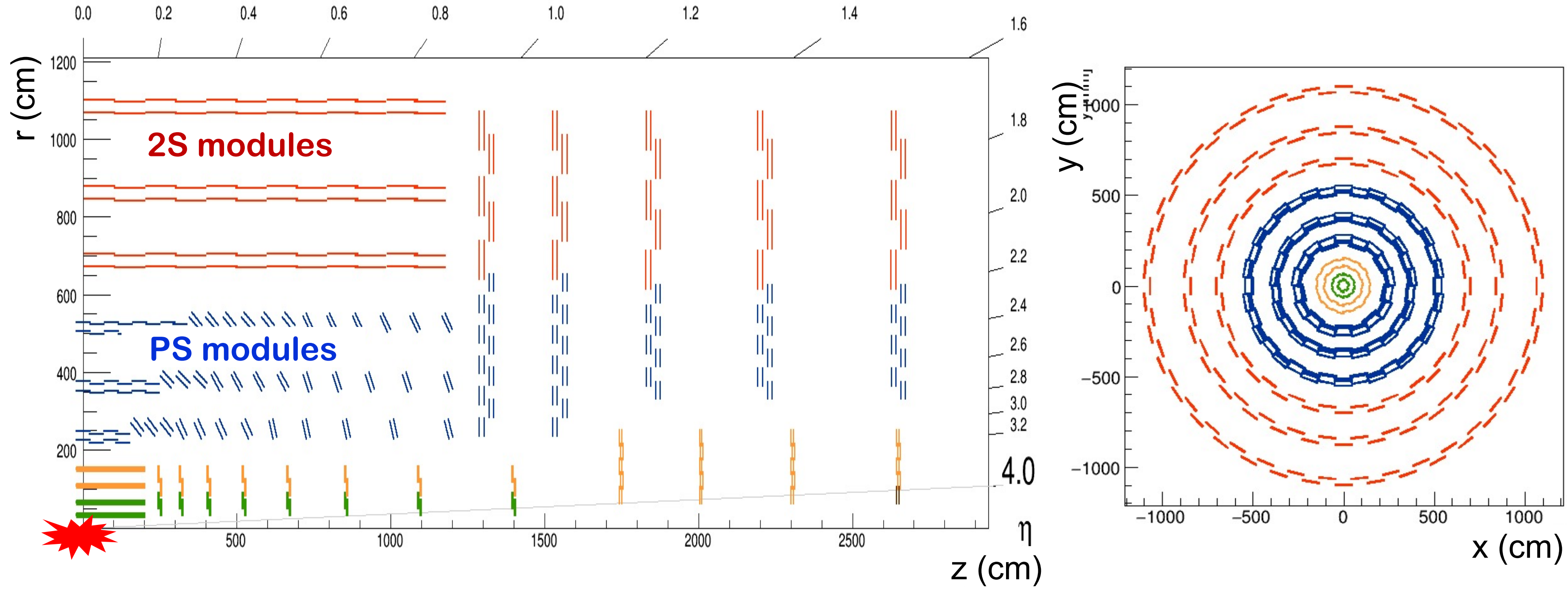
- Illustration: $B_S \rightarrow \Phi\Phi \rightarrow 2(K^+K^-)$
  - No leptons, photons or high energy jets, so conventional L1 triggers fail!
  - But Tracker tracks let trigger reconstruct mass resonance!



HL-LHC CMS tracker

CMS (& ATLAS) has new silicon tracking detector for HL-LHC.

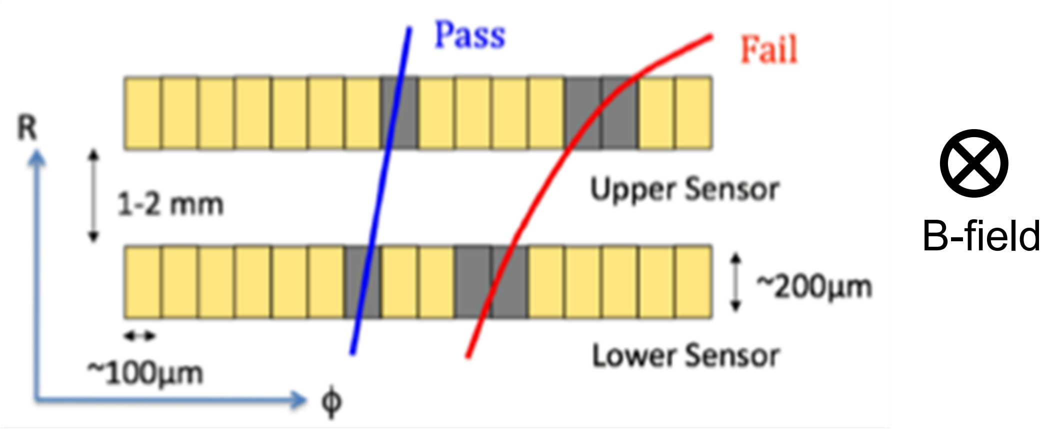
- In outer tracker, each module has 2 closely-spaced silicon sensors.
  - 110 > r > 60cm: 2S modules (each with 2 strip sensors)
  - 60 > r > 20 cm: PS modules (each with 1 strip + 1 macro-pixel sensor)
- Pixel tracker: r < 20cm (*not* used for real-time tracking).



HL-LHC CMS tracker

- FE electronics reconstructs **clusters** = particle crossing a sensor.
  - Transmits them off-detector when L1 trigger = "yes" (1 event in 40).
  - Used for track reco by HLT & Offline.

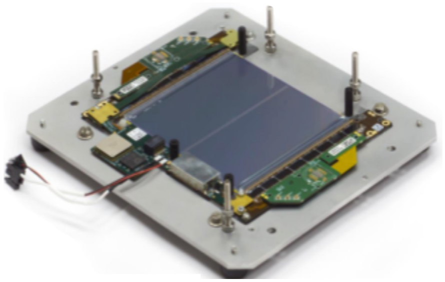
- FE electronics reconstructs **stubs** = pairs of neighbouring clusters in module.
  - Each stub measures p_T of particle that created it. Kill those below 2 GeV.
    - Factor 30 data rate reduction, compared to clusters!!!

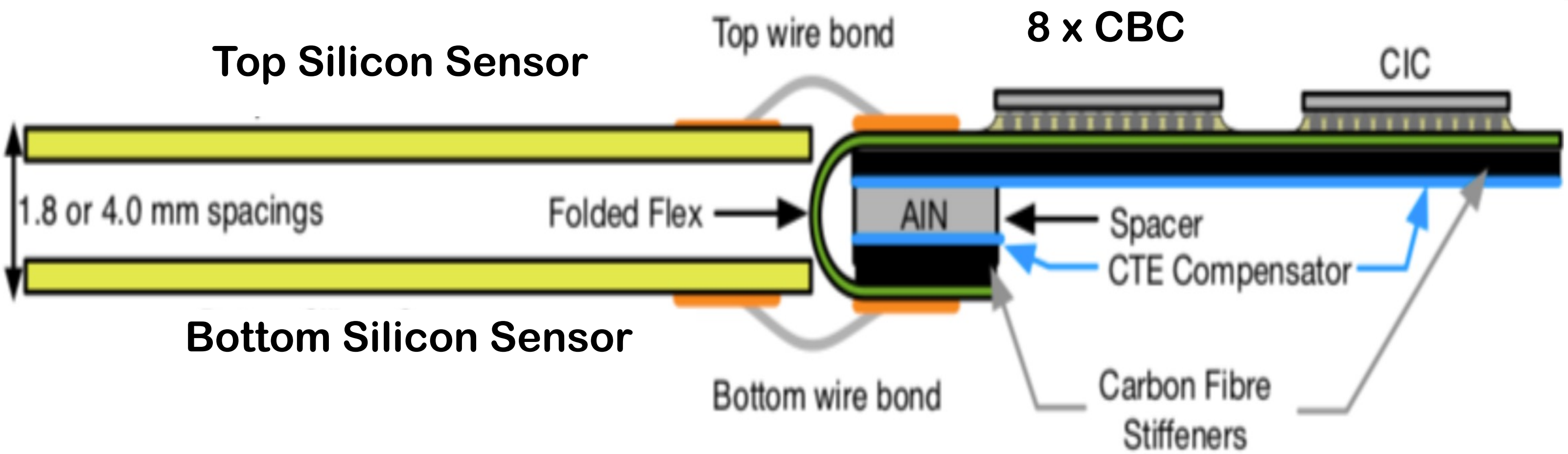


  - Allows transmission of stubs off-detector for *every* LHC collision event.
  - Reduced data rate simplifies real-time track reco.

HL-LHC CMS outer-tracker modules

- Each **2S module** is 10x10 cm²:
  - Both its strip sensors have 5cm x 90μm pitch.
  - Read out via optical link.

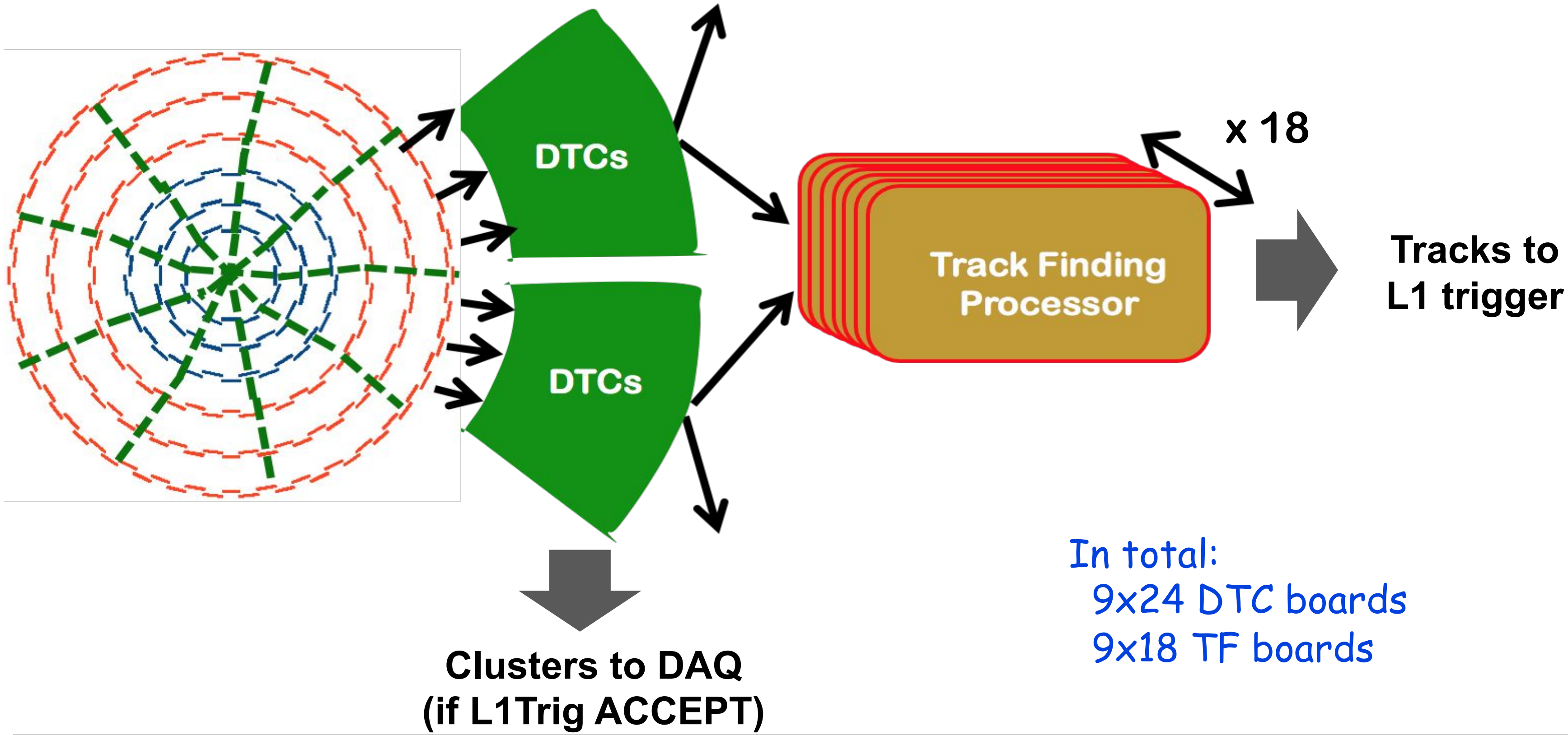




- Each **PS modules** is 5x10cm²:
  - One strip sensor with 2.5cm x 100μm pitch.
  - One macro-pixel sensor with 1.5mm x 100μm pitch.

CMS tracker back-end electronics overview

- 24 off-detector DTC (FPGA) boards read one φ nonant of Tracker.
  - Send stubs to Track Finder (TFP) board,
- Each TFP finds tracks in just one φ nonant (= angular multiplexing).
- 18 TFP in each φ nonant, take it in turn to process events (= time multiplexing).



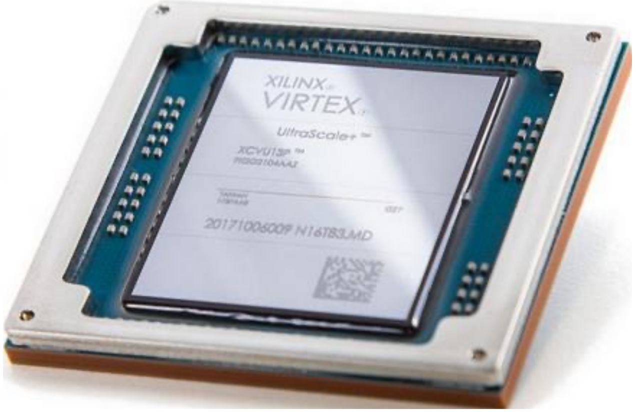
In total:
9x24 DTC boards
9x18 TF boards

FPGA-based Track-Finding Processor

- L1 tracking done in board: with 1-2 large FPGA (VU13P).
  - 48 input optical links (25 Gb/s) reading stubs from DTCs.
  - 3 output optical links (25 Gb/s) writing tracks to L1 trigger system.

- Each TFP receives 1 event in 18, so new event every 18*25ns = 450ns.
  - But TFP takes 4.0 μs to reconstruct an event.
  - Must process new event, before finished with last one!
  - Achieved by "data pipelining", explained in this web page.
  - FPGA runs at 240-360 MHz, reading new stub from each link every clock cycle.

Quick FPGA lesson



- Big commercial FPGAs market: good value.
  - Programmable chip, so can modify tracking algo.
  - Can rapidly do many calculations in parallel.

- Xilinx VU13P contains resources:
  - 12k DSP (each multiplies 27 x 18 bit numbers)
  - 5k BRAM (each providing 18 kb memory)
    - *But can only write (or read) to a BRAM once per clock cycle.*
  - 2M LUT (each providing 64b memory/logic table).
    - *For hard calcs (e.g. division) may store precalculated answers in memory.*

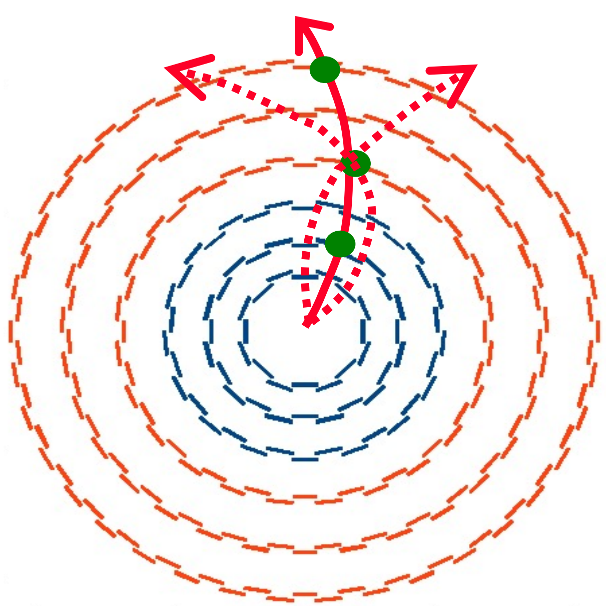
- Traditionally programmed in VHDL (Europe) or Verilog (USA).
  - Unlike software, specify which clock cycle each operation occurs in, & design algo to match FPGA resources.

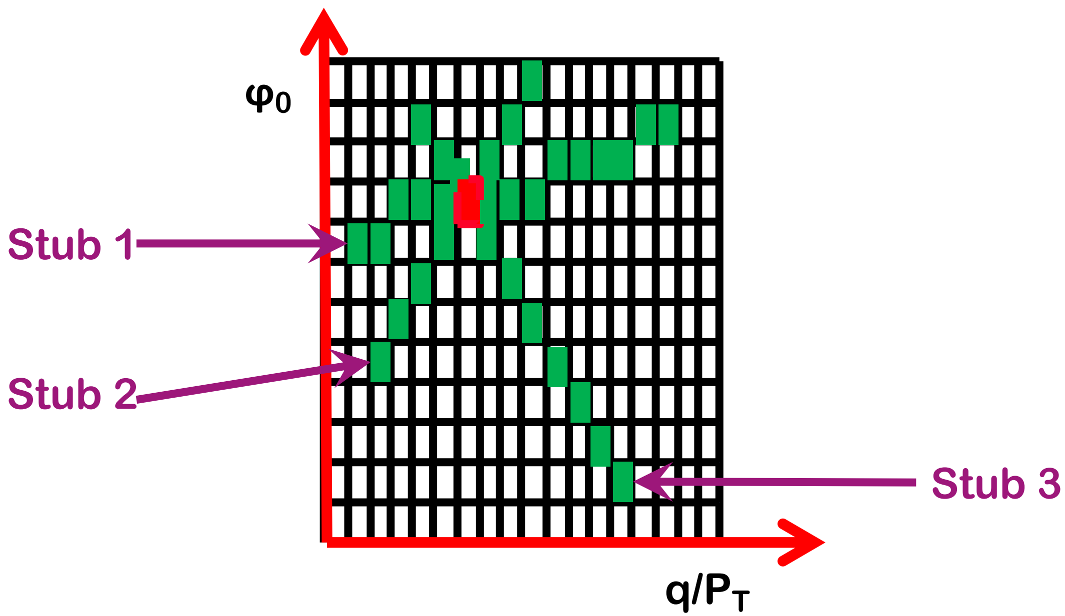
- Modern alternative to program in HLS (essentially C++).
  - Compiler converts it to VHDL/Verilog. (Sometimes works, sometimes not)
  - Can call HLS from CMS simulation program to predict tracking performance.

Various real-time tracking methods

A simple tracking algorithm in r-φ plane
Hough transform

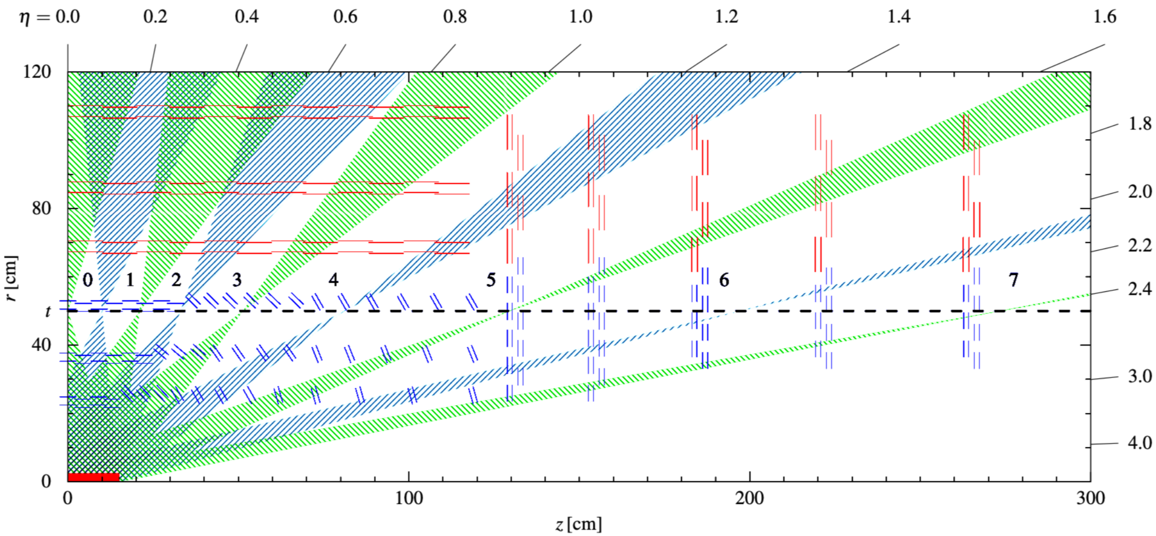
- CMS studied *FPGA-friendly* "Hough transform" algo = 2D array in track (q/p_T, φ_0).
  - For each stub, & all q/p_T columns *in parallel*, add entry to any φ_0 bin, if track of that (q/p_T, φ_0) would pass through stub.
    - *(Trick: stub's p_T measurement vetoes some q/p_T columns).*
  - If bin has stubs from ≥ 5 tracker layers, we've found a track!
  - Processing time *linear* with no. of stubs.





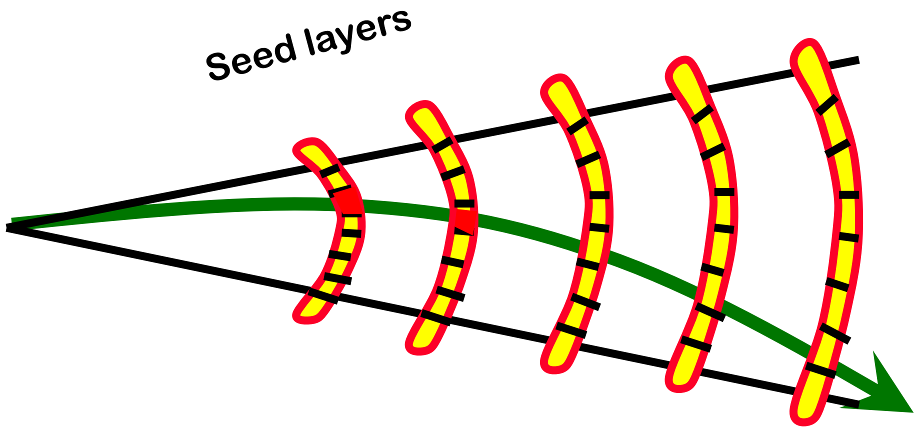
A simple tracking algorithm in r-φ plane
Hough transform

- Need additional trick, so HT copes with HL-LHC events.
  - Divide tracker φ-nonant into 16 sectors in r-z plane, each processed by one HT.
    - Gives crude track finding in r-z plane.
    - But sectors must overlap to allow for 5cm length of LHC beam-spot.



- **HT strength**: very fast (gives tracks in << 4 us)
- **HT weakness**: hard to find tracks below 3 GeV.
  - If particle jet in sector, input data rate so high, may get truncation.
  - Minimum useful HT bin size set by particle scattering, worse at low p_T.

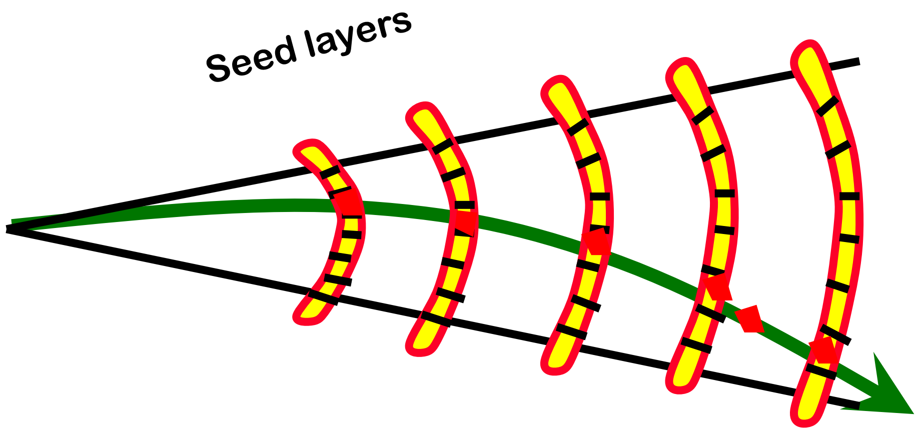
Track reconstruction: "Hybrid" algorithm (= CMS baseline)



1) Divide each tracker layer into small φ regions ("VM" = ~1/16 of φ-nonant), each with memory storing stubs.

2) **Seed track candidates** with pair of stubs in neighbouring layers.
   - FPGA has multiple logic blocks *running in parallel*, each operating on only 1 VM in each seeding layer.

Track reconstruction: "Hybrid" algorithm (= CMS baseline)



3) Extrapolate helix through each seed to other tracker layers, seeking compatible stubs.

- First determine which VM's seed extrapolates to.
- Each VM has its own FPGA logic that examines stubs there.
  - All VMs of interest examined *in parallel*.

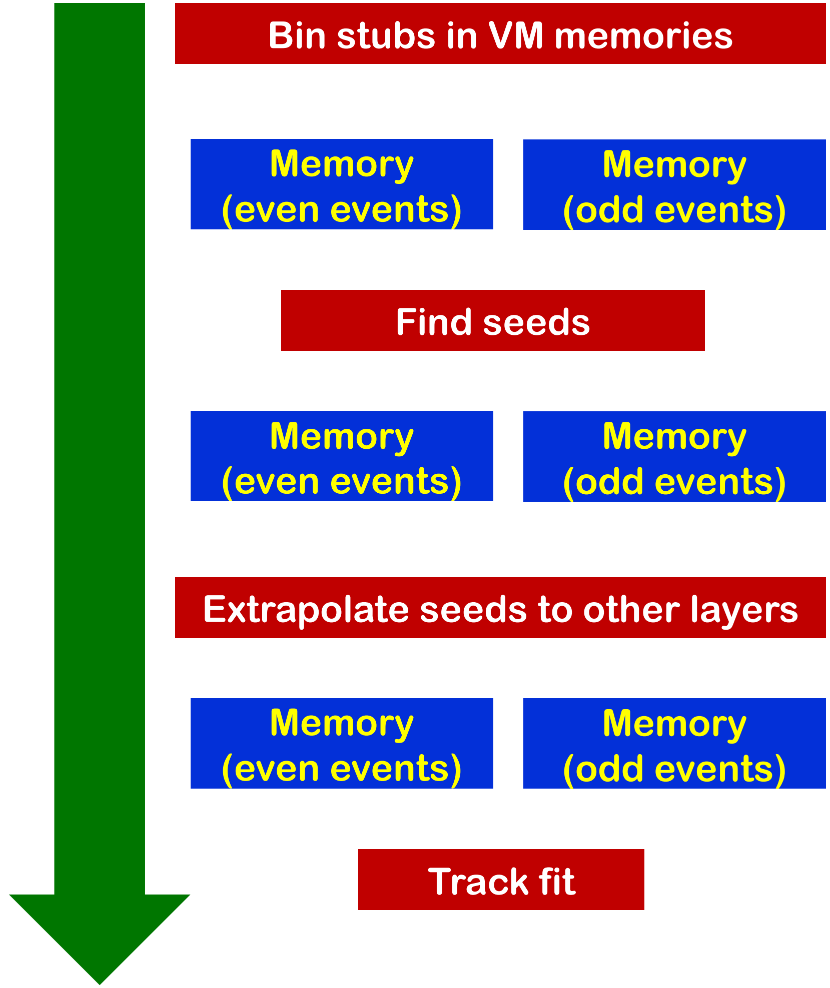
Strengths:

- Each VM contains few stubs, so fairly fast.
- VM size « particle jet size, so small data rate fluctuations due to jets.

Track reconstruction: "Hybrid" algorithm (= CMS baseline)

Simplified algorithm structure:

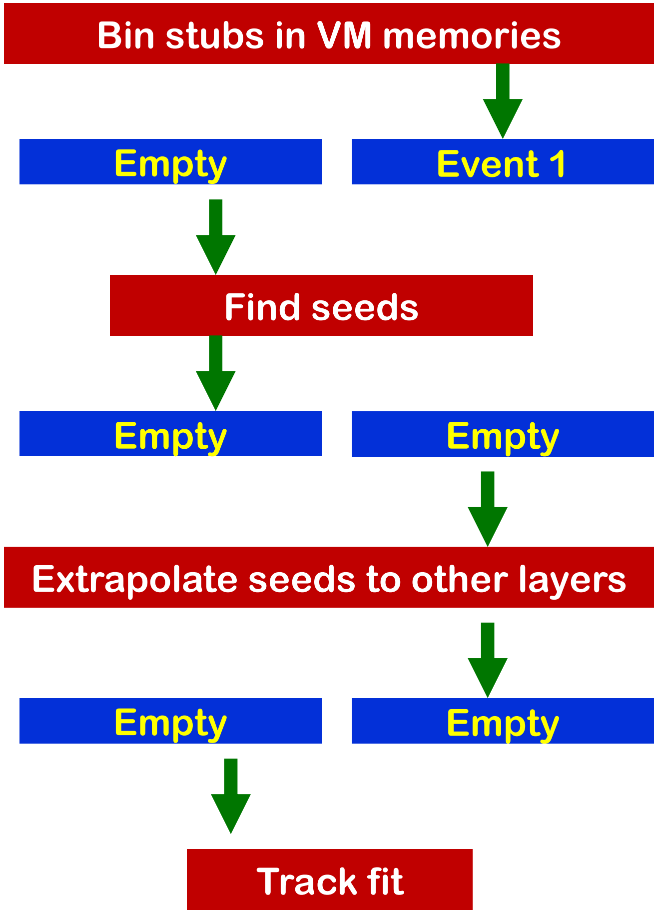
- Algo steps (HLS) in red.
- BRAM memories in blue.
  - 2 pages for even & odd numbered events.
  - Common trick, so all algo steps can be run in parallel. (Also used by HT algo).
- All instantiated in top-level firmware (VHDL).



Track reconstruction: "Hybrid" algorithm (= CMS baseline)

Illustration of data flow
& use of BRAM memories

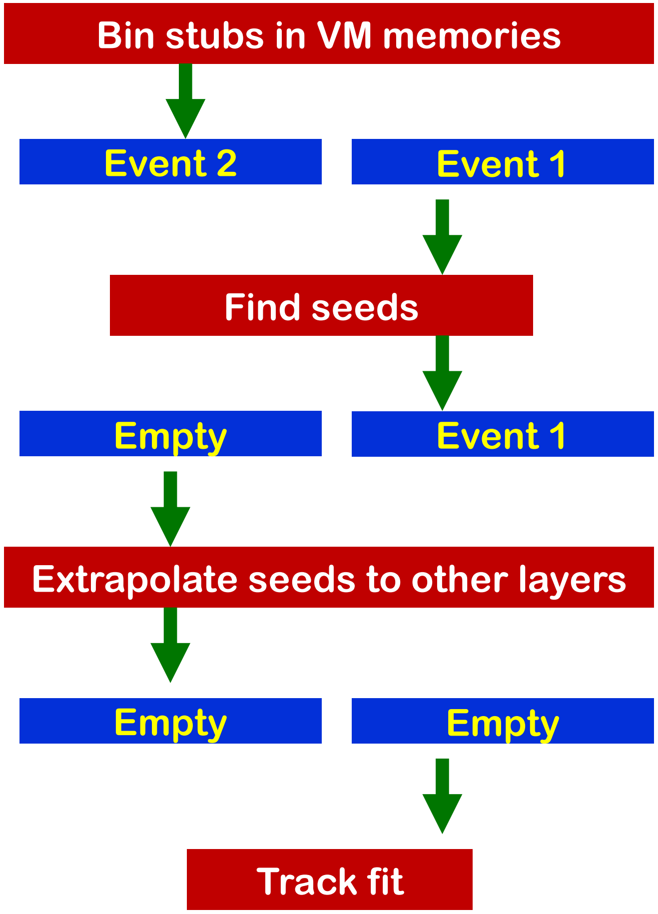
Time: t = 0



Track reconstruction: "Hybrid" algorithm (= CMS baseline)

Illustration of data flow
& use of BRAM memories

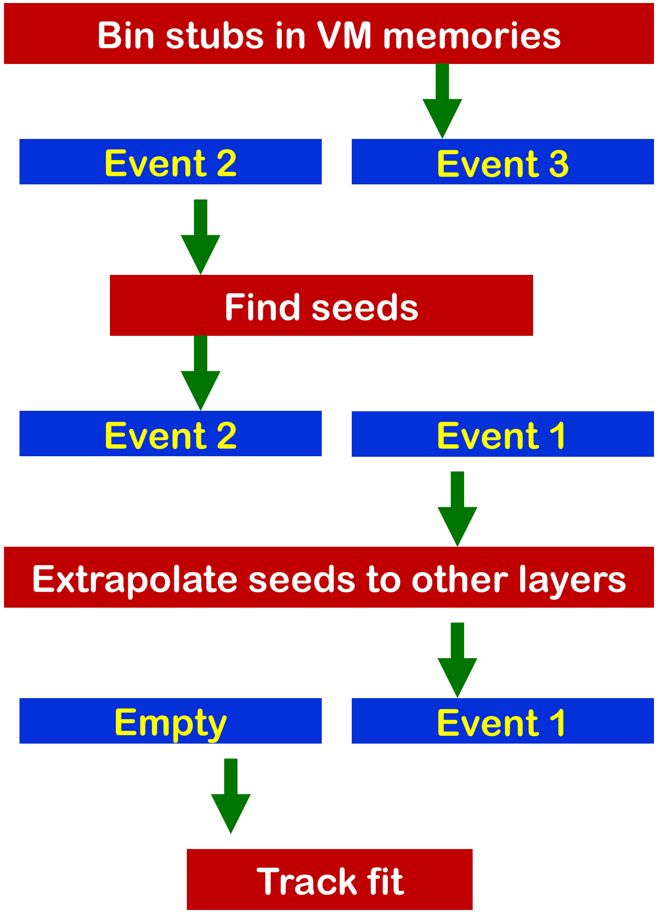
Time: t = 18*25ns



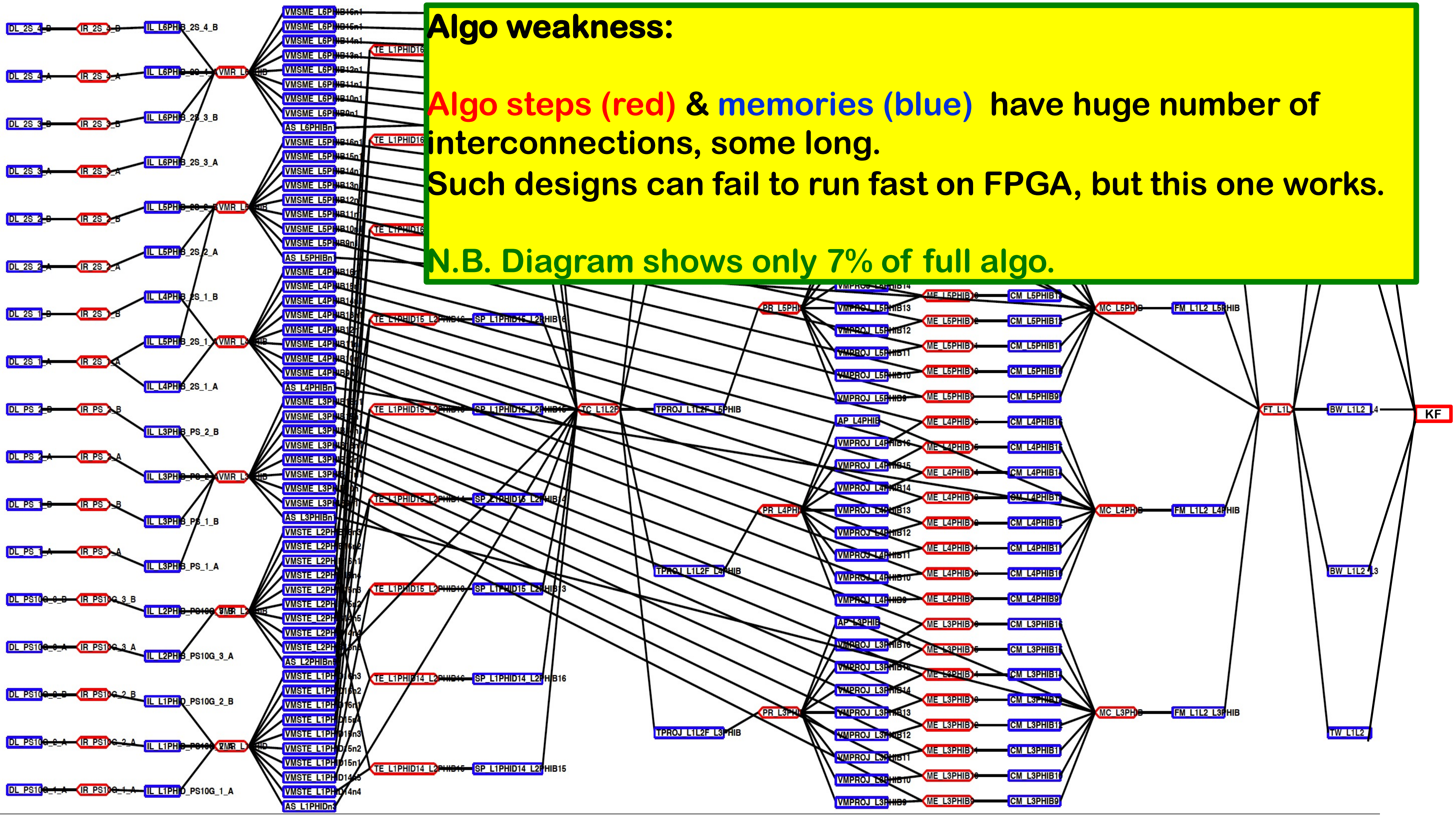
Track reconstruction:
"Hybrid" algorithm (= CMS baseline)

Illustration of data flow
& use of BRAM memories

Time: t = 2*18*25ns



Track reconstruction: "Hybrid" algorithm (= CMS baseline)



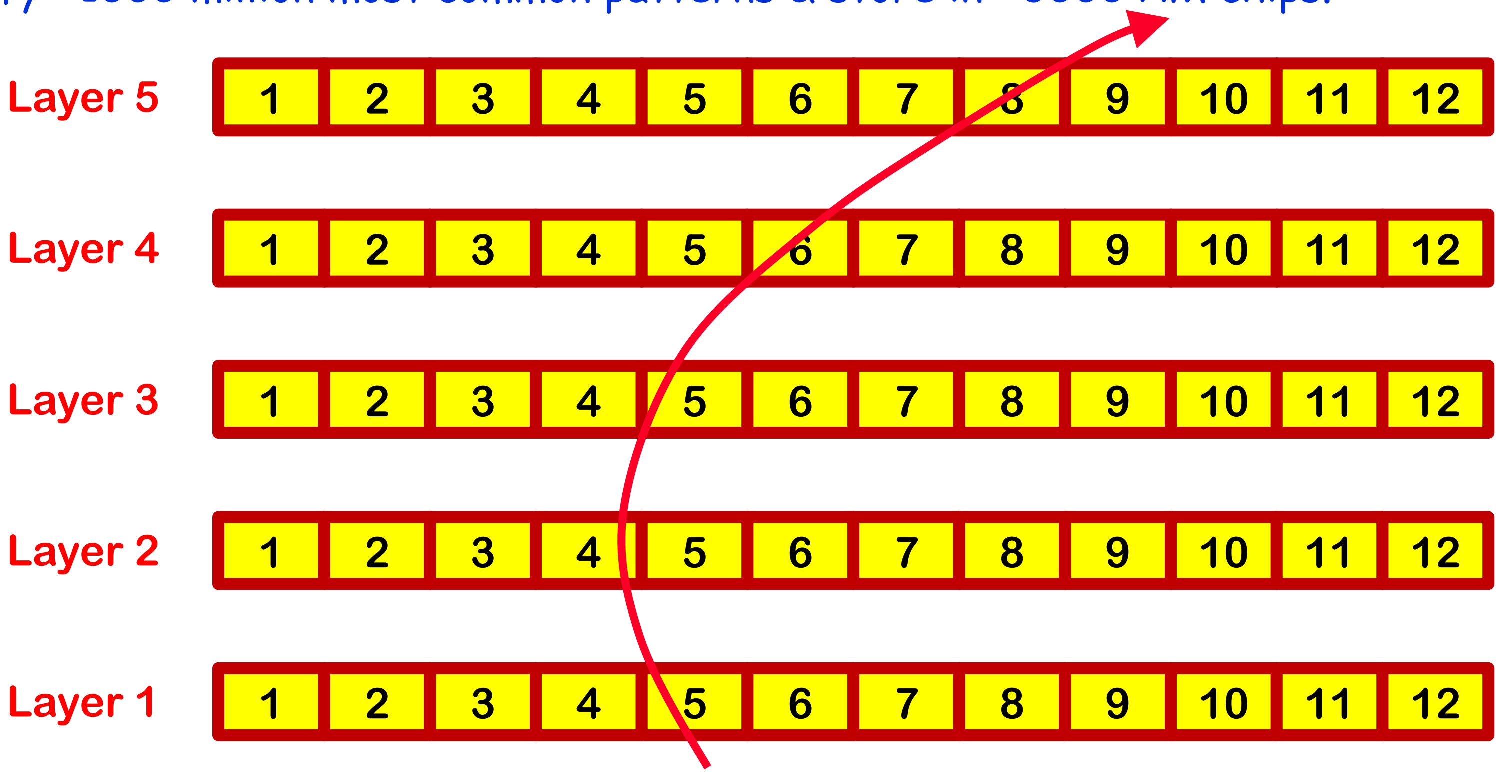
Algo weakness:

Algo steps (red) & memories (blue) have huge number of interconnections, some long.
Such designs can fail to run fast on FPGA, but this one works.

N.B. Diagram shows only 7% of full algo.

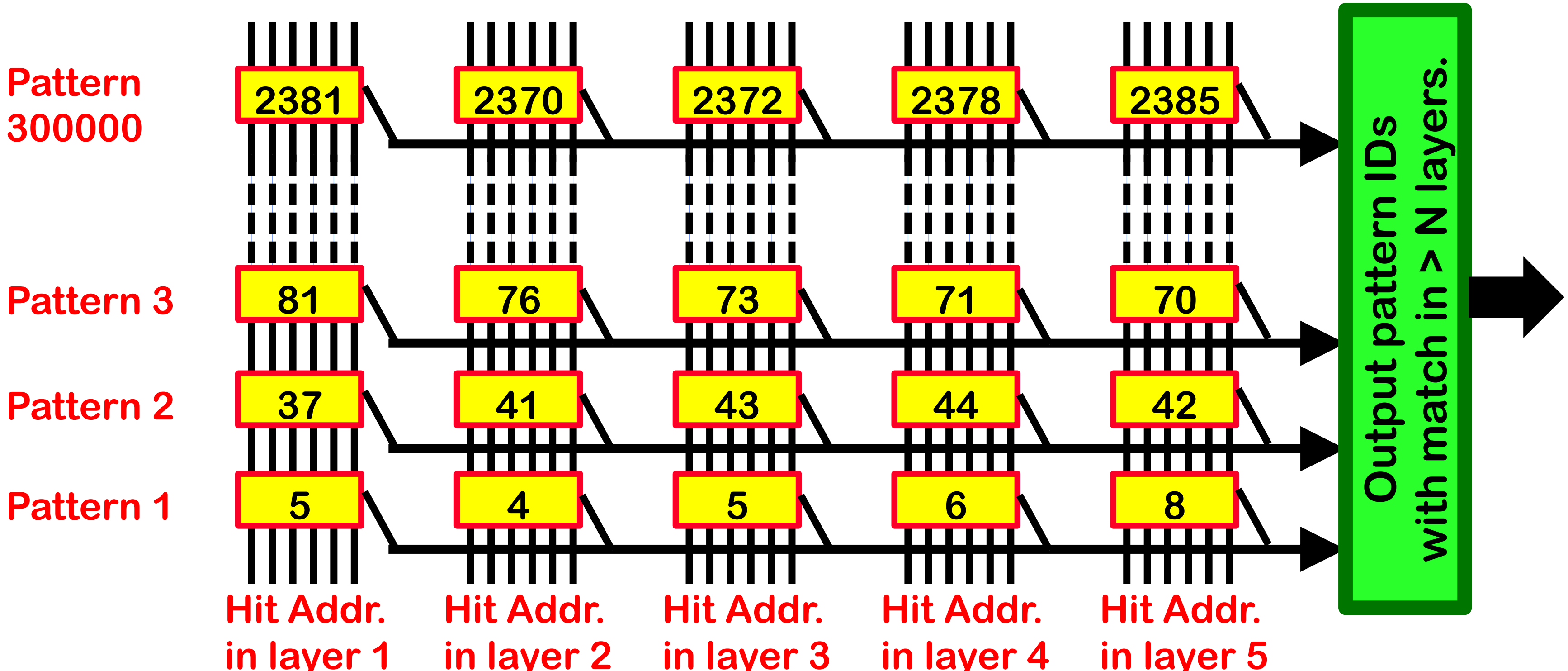
ATLAS & CMS track reconstruction: *custom integrated circuit: AM chip*

- ATLAS & CMS showed possible tracking with "Associative Memory" (AM) chip.
  - Divide each tracker layer into small (~1cm) numbered regions.
  - Simulated events say which patterns of these are fired by each particle. e.g. (L1,L2,L3,L4,L5). have pattern (5,4,5,6,8)
  - Identify ~1000 million most common patterns & store in ~3000 AM chips.



ATLAS & CMS track reconstruction: *custom integrated circuit: AM chip*

- AM chip is custom "Content-Addressable-Memory" chip,
  - Weakness: not as flexible as FPGA. And hard work to design.
- Stores 300k patterns. Each input hit checked against all patterns **simultaneously.**
  - Entire tracker uses ~3k chips = 1000M patterns!
- Patterns with matches in > N layers are track candidates.



Final step: Kalman Filter track fit

- Hough Transform, Hybrid & AM algos produce *rough* tracks.
  - Must then clean them (kill incorrect hits) & fit track trajectory.

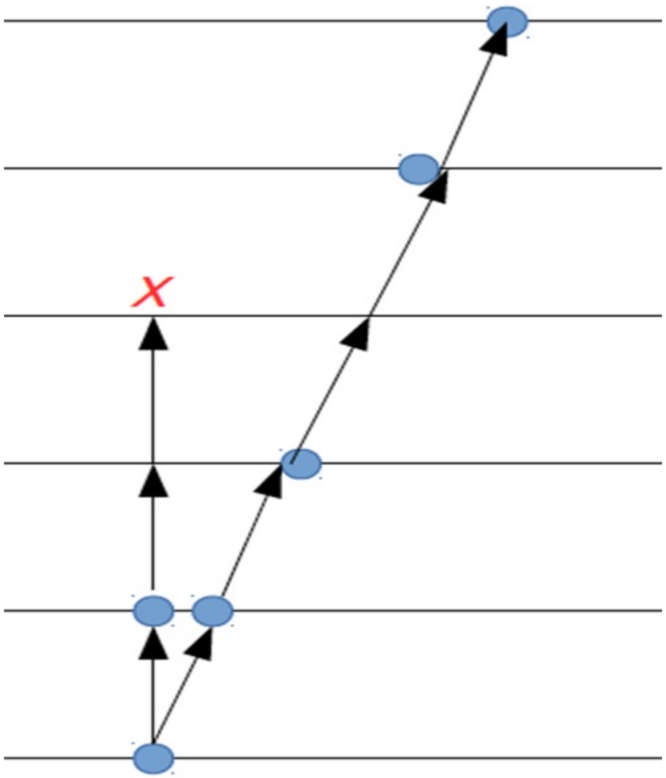
e.g. **Kalman Filter track fit:**

1) Initial track trajectory estimate from HT (or Hybrid or AM), but inflate its uncertainty to infinity.

2) Update trajectory with stub from layer 1, then layer 2 etc.

3) If multiple stubs per layer, create one trajectory from each.

4) Keep best trajectory found from each track.



FPGA-unfriendly algo, but possible as HT etc. reduce enough data rate.

Summary

- Real-time track-reco in CMS greatly improves real-time event selection (trigger).
  - Factor 5 trigger rate reduction.

- CMS real-time tracking will use programmable FPGA chips.
  - Successfully demonstrated in HW.
  - Can reconstruct tracks from *all* LHC collisions in < 4μs.
    - — possible in thanks to clever double-sensor tracker modules.

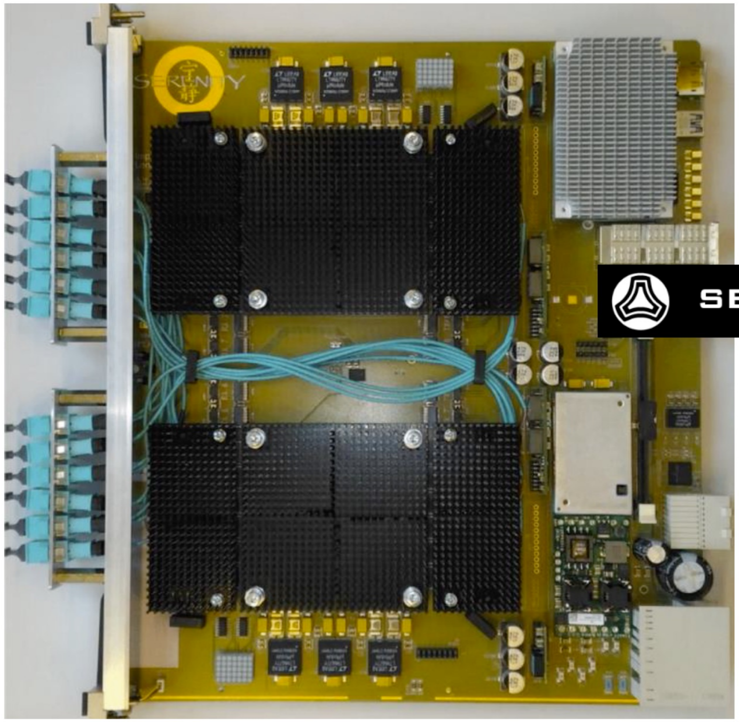
- ATLAS & CMS explored custom ASICs (AM chip) solution.
  - Successful, but rejected,
  - More work & less flexible than FPGA.

More info: nice review in web link

Backup

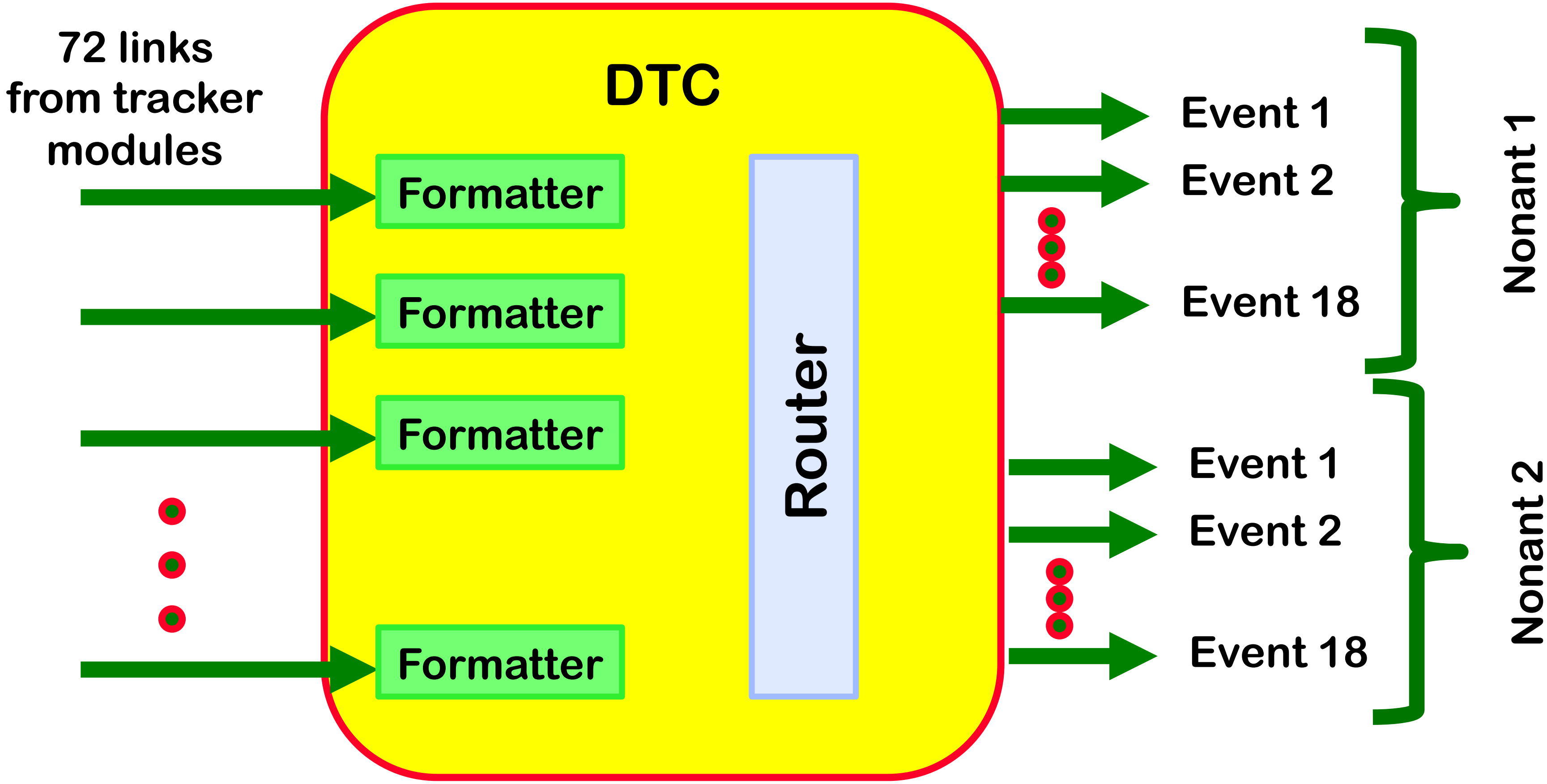
DTC Board

- The DTC board is of type "Serenity"
  - Will be equipped 1 large FPGA (VU13P)

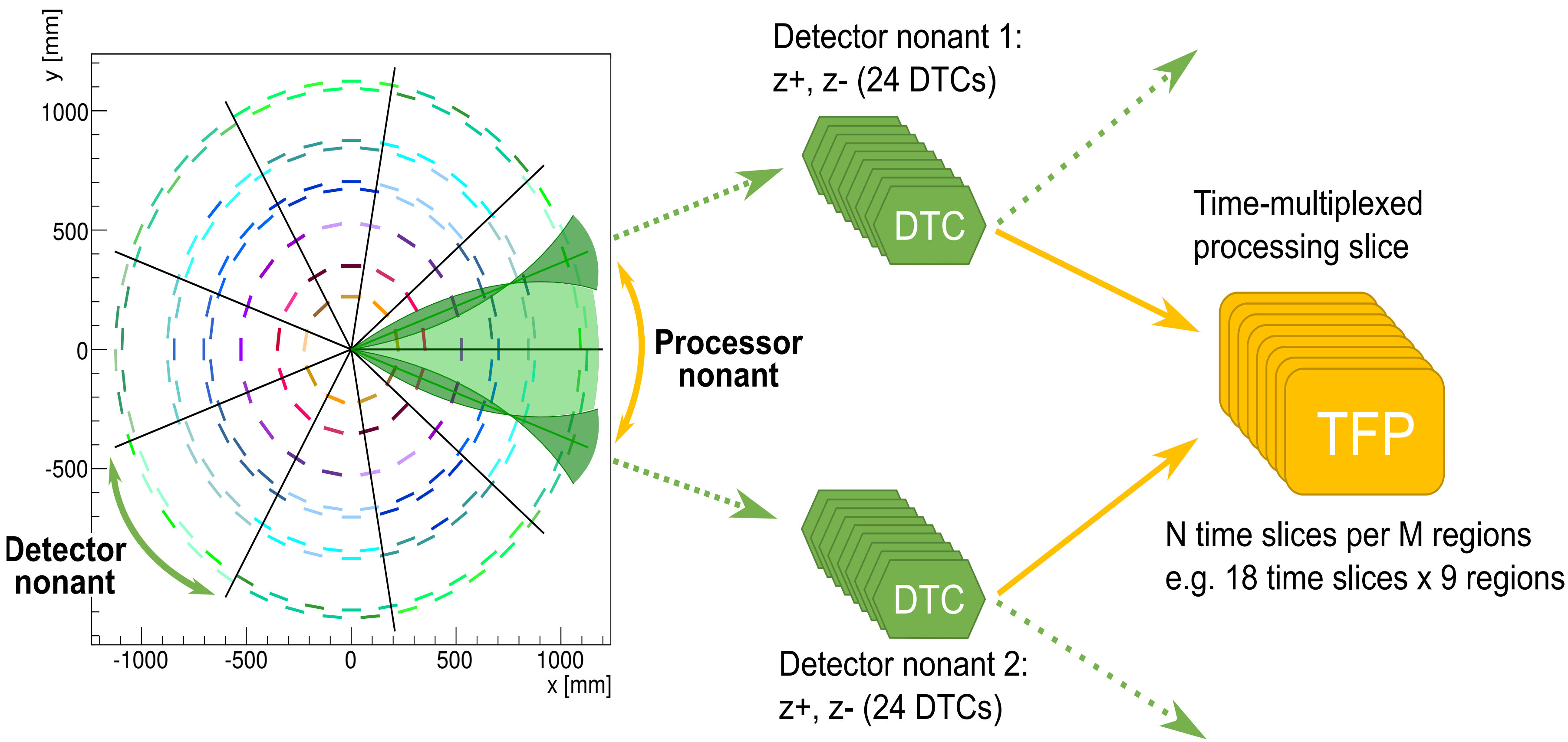


DTC Firmware & Software

- UK responsible for stub processing in DTC.
  - Formatter: Converts stub strip → coords & assigns to φ nonant(s).
  - Router: Sends stub to correct output link

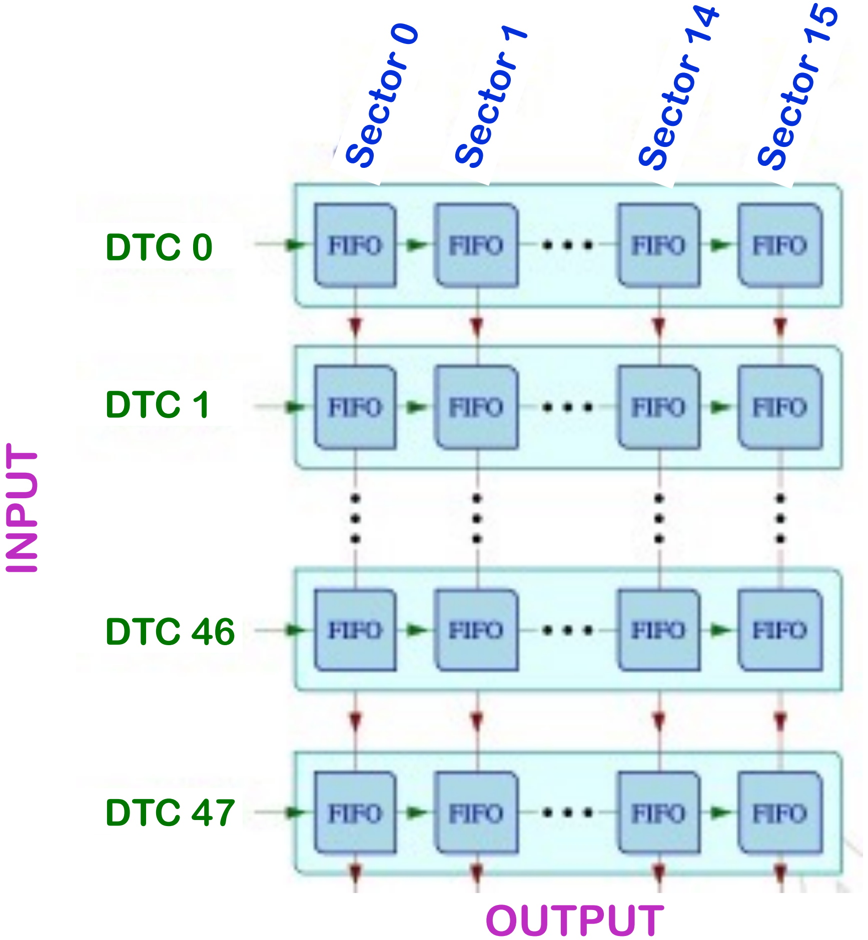


Use of φ nonants in Tracker readout



Hough transform input firmware: routing stubs from 48 DTCs to 16 r-z sectors

- Uses a systolic array.
  - First-in-first-out memories (FIFO) handle traffic jams.



A simple tracking algorithm in r-φ plane
Hough transform firmware

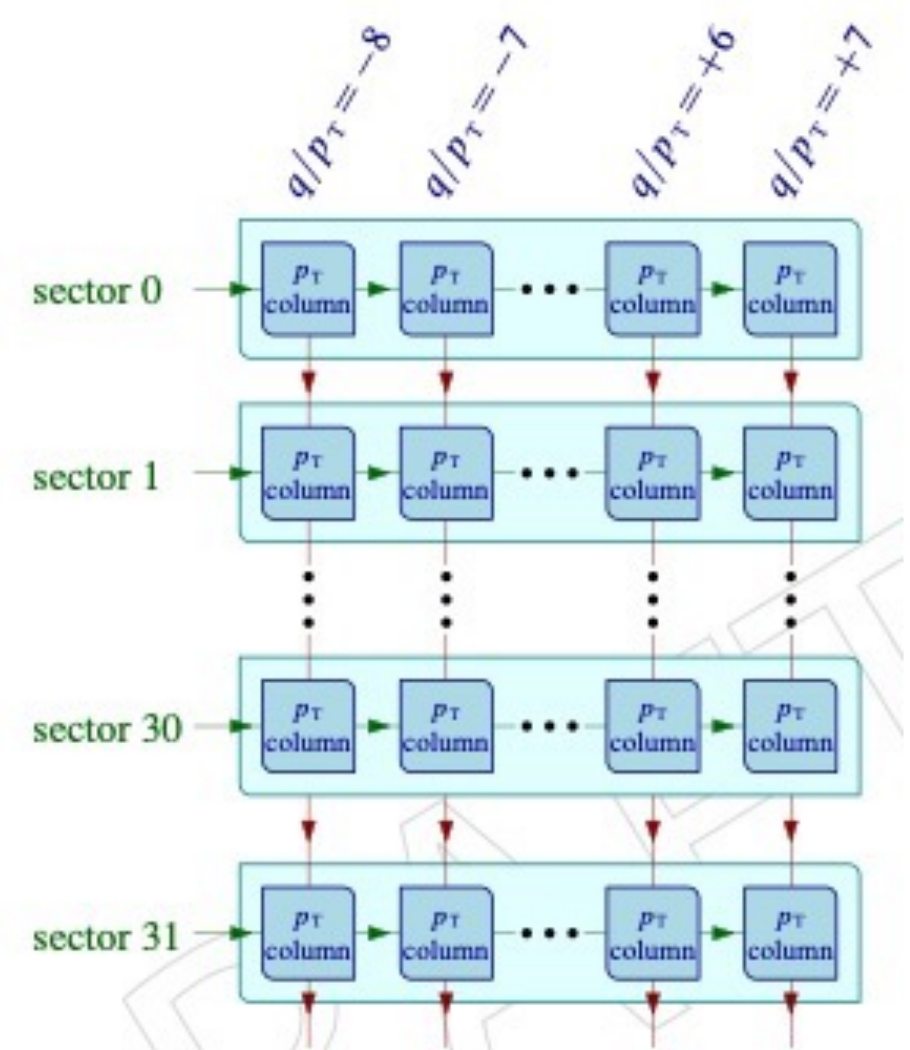
1) Stubs for HT in sector 0 go to each q/P_T column in turn, where firmware adds stub to correct φ bin.

2) After all stubs from event read in, firmware for q/P_T column identifies φ bir with tracks (stubs in ≥ 5 tracker layers).

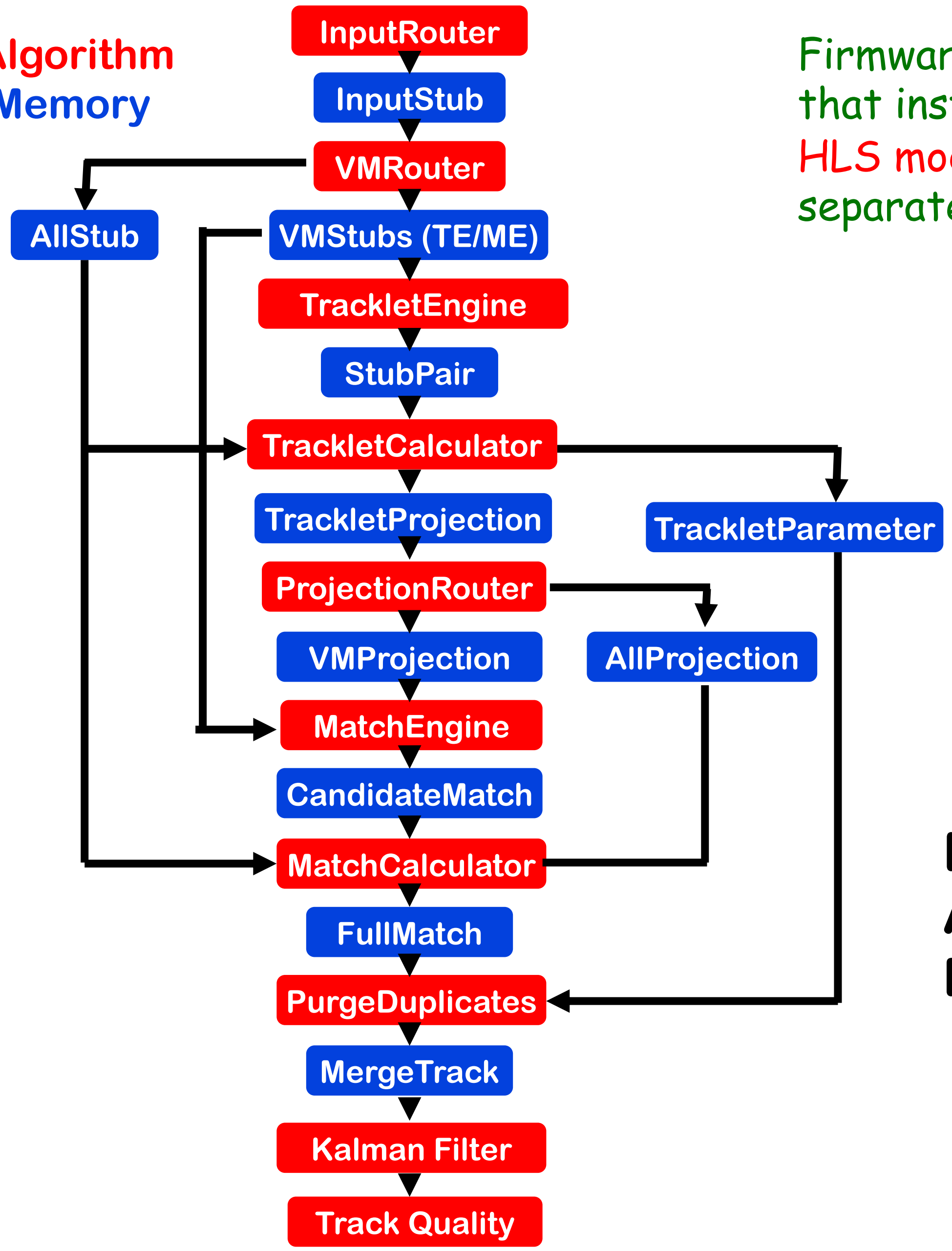
   - q/p_T column stub storage uses 2 pages of BRAM memory. Odd events written to one & even to the other. Allows (1) & (2) to run in parallel.

3) Tracks output to link based on their q/p_T bin, not sector.

   - Avoids hot links due to particle jets.



Key:
Red=Algorithm
Blue=Memory



Firmware consists of **top-level VHDL** that instantiates chain of HLS modules (red) separated by memories (blue).

C++ emulation can call HLS modules.

HYBRID L1 TRACKING ALGORITHM FPGA FIRMWARE DESIGN