

Thoughts about CMS SLHC Level 1 Track Trigger Architecture

Ulrich Heintz (*Brown University*)

for

M. Narain (*Brown U*)

R. Lipton, M. Johnson (*Fermilab*)

S.X. Wu, E. Hazen (*Boston U*)

the challenge

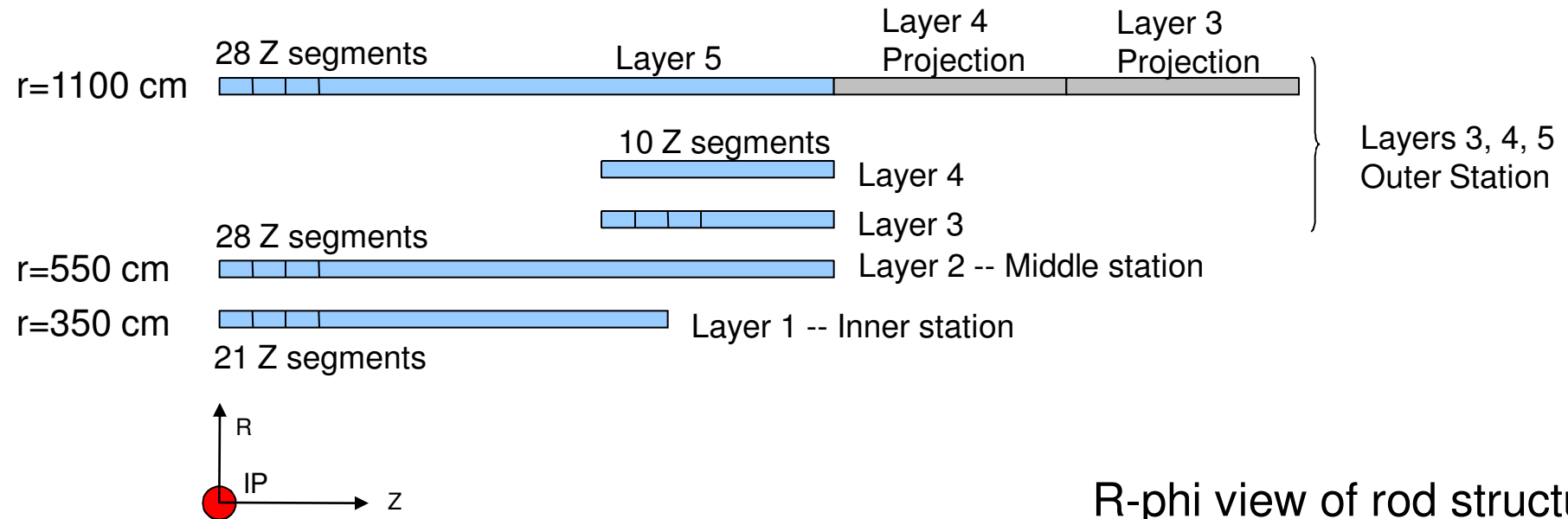
- how can we bring the signals from the CMS tracker together to find tracks
- with $p_T > 2.5 \text{ GeV}$
- within L1 latency



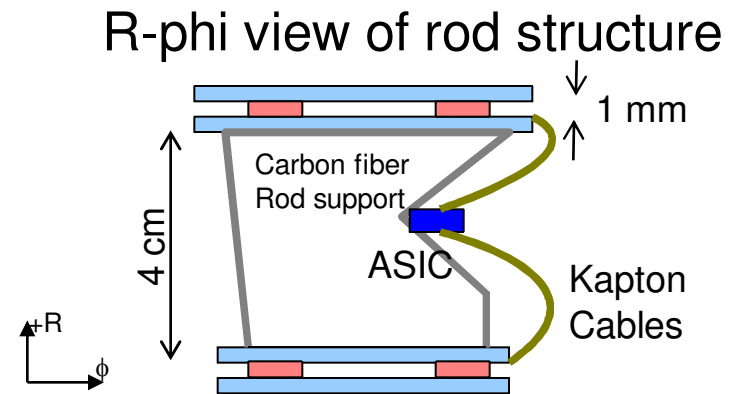
- two strawman scenarios
 - logic equations á la D0 L1 track trigger (Marvin)
 - tracklet extrapolation (Shouxiang)

geometry

- assume long barrel design for tracker



BUT:
many ideas would also apply to other geometries and to different low p_T hit rejection schemes (e.g. cluster width)

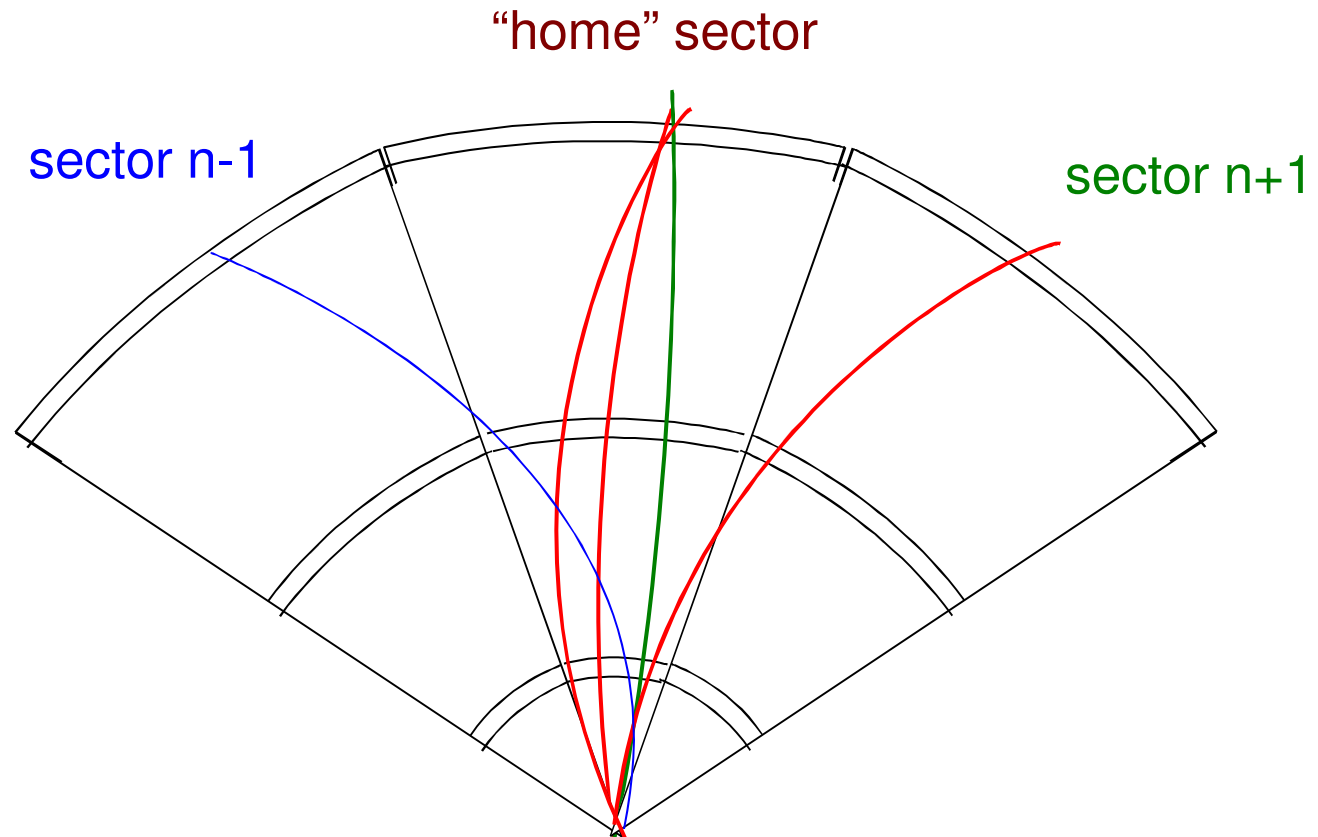


design 1 (à la D0 L1 track trigger)

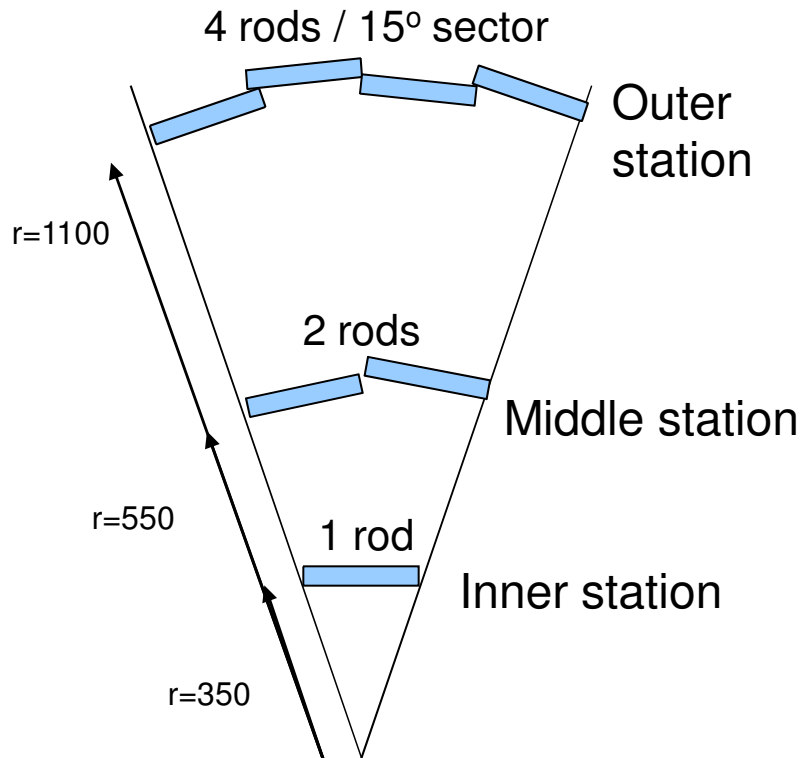
- represent each possible hit combination by a logic “equation”:
(layer 1) AND (layer 2) AND (layer 3)
- require 2 of three layers for robustness
- create a table of all possible equations in FPGAs
- feed all hits for an event into the FPGAs and evaluate all the equations simultaneously.
- the equations which are satisfied are the reconstructed tracks
- tracks are available one clock cycle later.

sector structure

- determined by minimum p_T acceptance requirement
- tracks with $p_T > 2.4$ GeV can traverse at most 2 15° sectors
- trigger logic must handle inputs from “home” sector plus $n-1$ and $n+1$



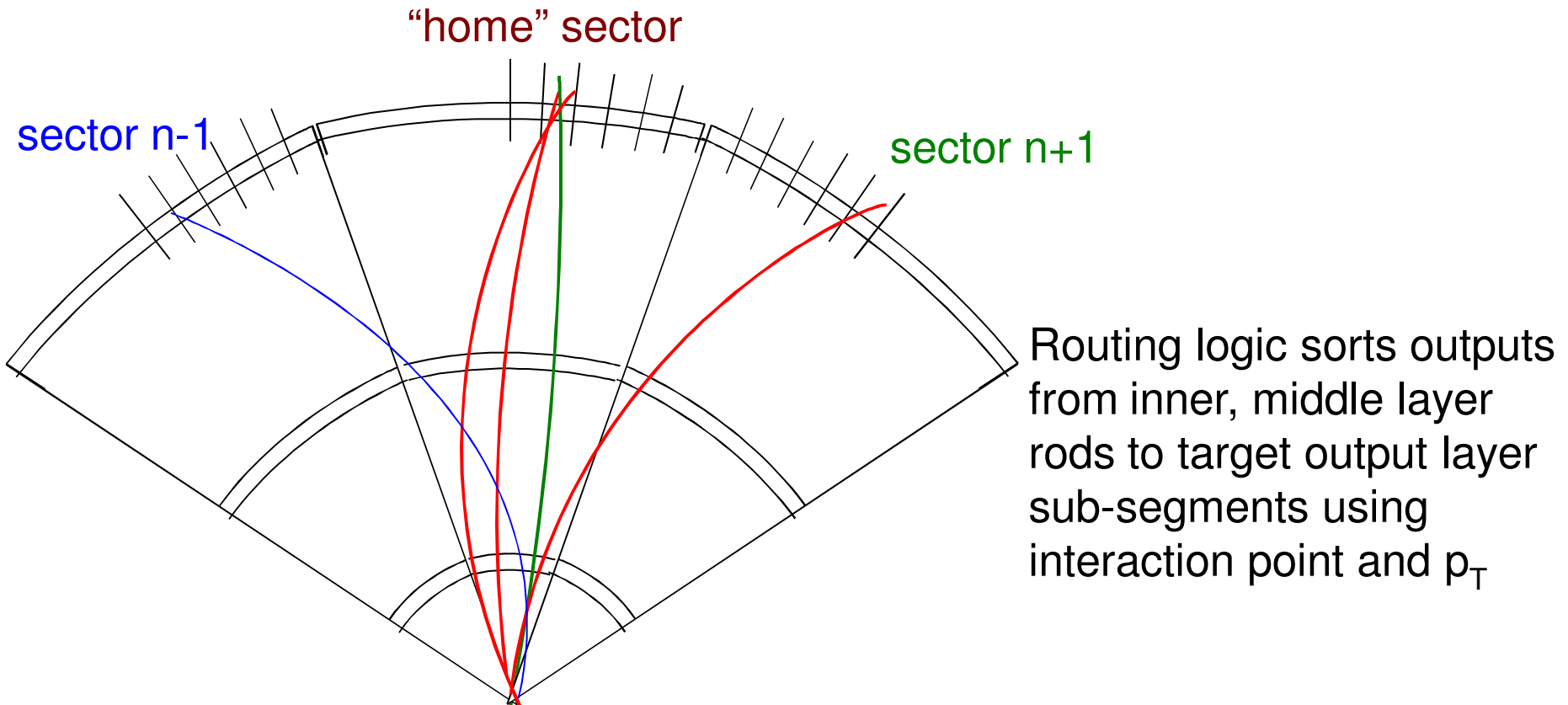
data rate estimate



- for each hit
 - 20 bits
 - 10 bits ϕ
 - 6 bits z
 - 2 bits curvature (+/0/-)
 - 2 spare bits
- for each sensor in inner station
 - $8 \times 10 \text{ cm} = 80 \text{ cm}^2$
 - 2 particles/cm²
 - Monte Carlo estimate
 - 160 hits/BX
- for each stack
 - 10x reduction due to 2-layer coincidences
 - 16 hits/BX
- data rate for each z-segment
 - $16 \text{ hits} * 20 \text{ bits} * 40 \text{ MHz} = 12.8 \text{ Gb/s}$
 - 2 fibers

equation count

- number of equations $\approx 100 * 200 * 400 * 2 = 16,000,000$
- too large by order of magnitude for one FPGA
- divide outer station into 12 sub-sectors



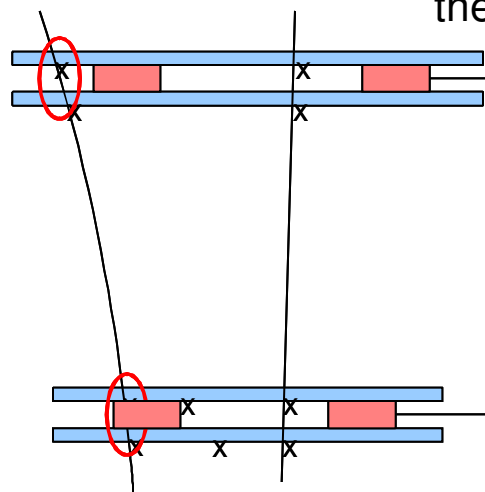
processing for each z-segment

doublets formed by readout chips

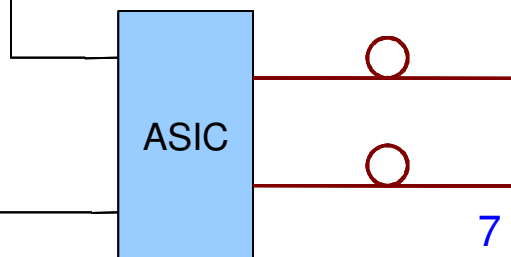
tracklets may be formed either on-detector by ASIC or in USC by the trigger processor.

sorter block sorts tracklets using IP, pT into sub-sector of outer station

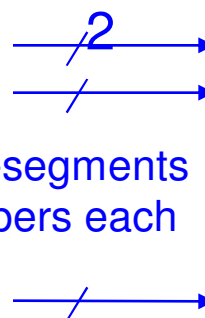
Doublet
(2-layer coincidence)



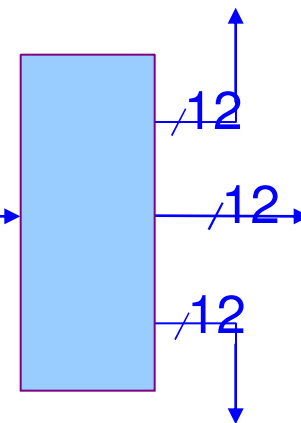
Tracklet
(4-layer coincidence with P_T validation)



7 z-segments
2 fibers each



tracklet block converts *doublets* into *tracklets*



tracklet data sent to home sector, plus 2 neighbors

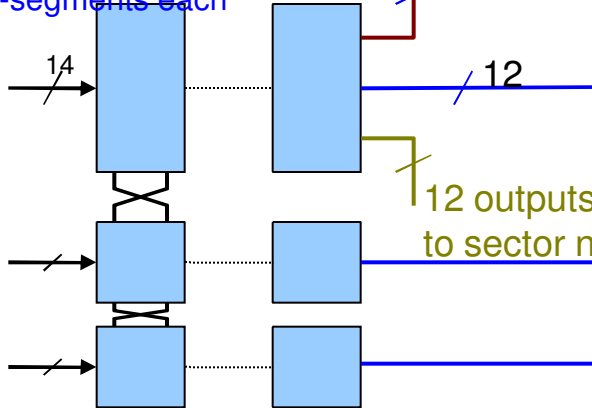
If ASIC sends doublets off-detector, then each station requires two fibers per z-segment per sector

If ASIC forms tracklets, this is reduced by about 2X

trigger processor (1 sector, 1/2 barrel)

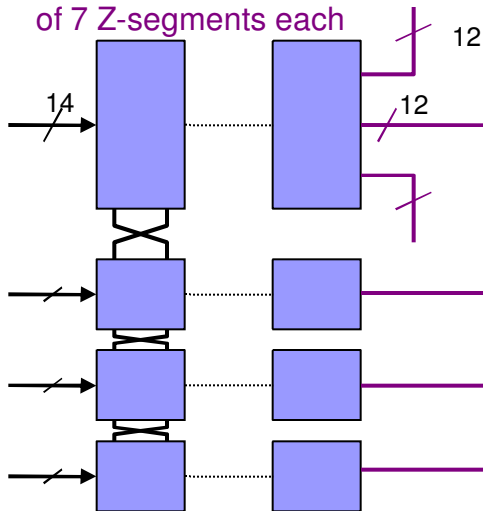
inner station logic

processes 3 Z-groups of 7 Z-segments each



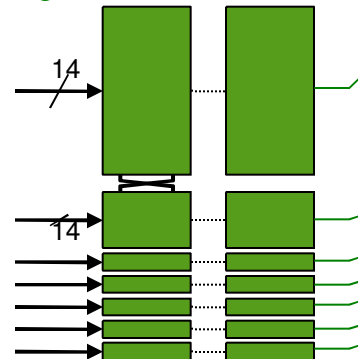
middle station logic

Processes 4 Z-groups of 7 Z-segments each



outer station logic

Processes 7 Z-groups of 7 Z-segments each



Inputs from sector n+1

12 subsector processors

7 * 12

7 groups of 12 links

from 7 Z-groups to 12 subsector processors

7 * 12

output: track candidate found

Inputs from sector n-1

summary in numbers

- 3 trigger stations provide full coverage in 15 degree sectors
- Hits collected in real time, sent off-detector on 5000-10,000 optical fibers at 6 Gbit/s
- All possible track equations for each sector evaluated in parallel using FPGAs in USC to produce ROI inputs for L1
- Total of 1000-2000 Virtex-6 class FPGAs needed to build this using current technology

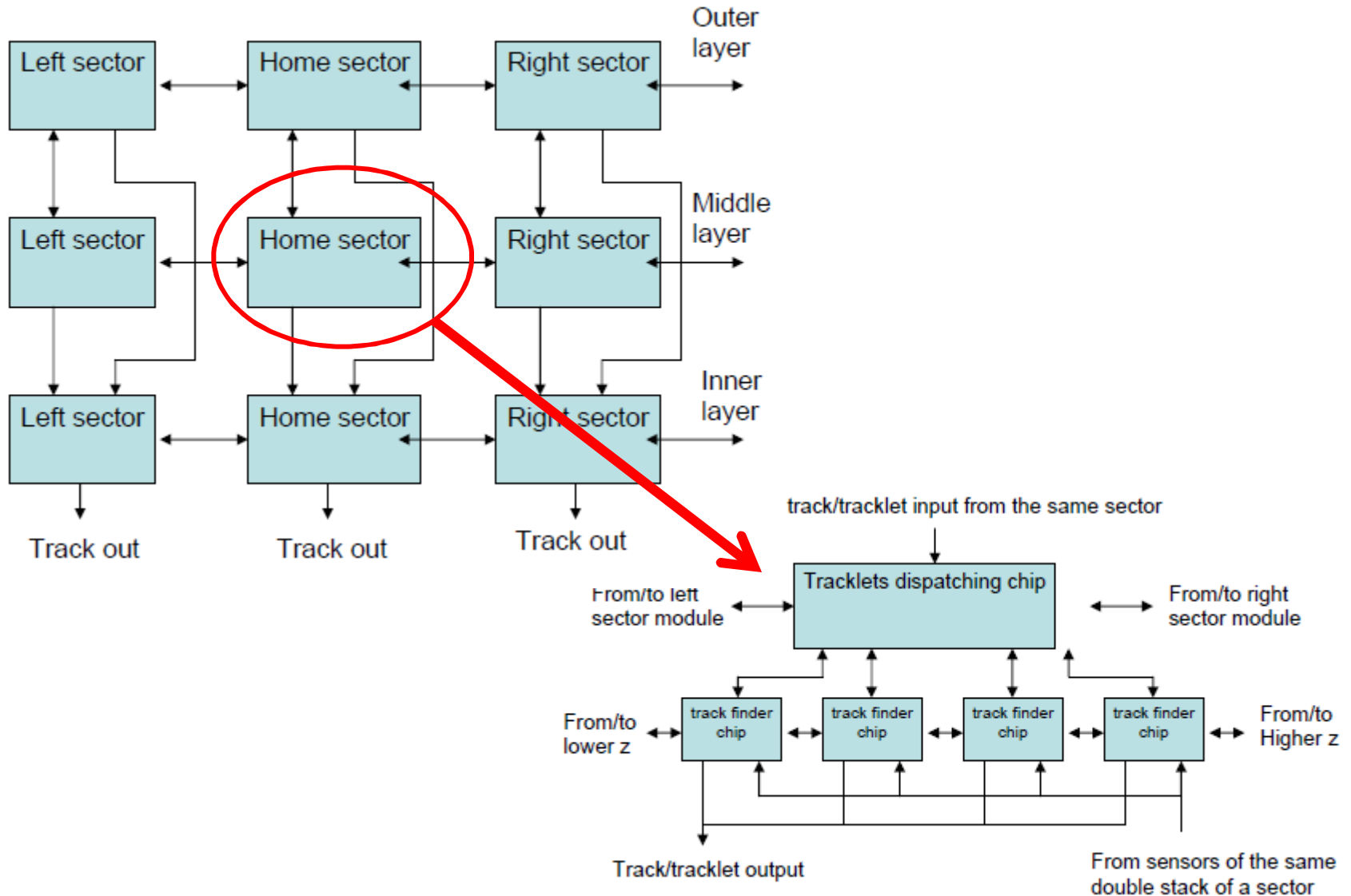
design 2 (tracklet extrapolation)

- possible shortcomings of design 1:
 - no use of z-segmentation
 - large number of inputs into single FPGA
 - power consumption of FPGAs and links for one sector may be too large for single board
- possible solution:
 - find tracklets in middle and outer stations
 - extrapolate to other layers in ϕ and z to find matching hits
 - only send tracklets between processing blocks

trigger processor

- find doublets on detector
- send doublet hit data to processor
- find tracklets in middle/outer station
- extrapolate in ϕ, z to outer/middle station
- recompute track with matching hits
- extrapolate tracks in ϕ, z to inner layer
- recompute track with matching hits
- cut on number of doublet hits on track
 - do not look for tracklets in high occupancy inner layer
 - robust against inefficient stacks

block diagram of trigger processor



next steps

- need MC simulation
 - include 200 pile-up events/bx
 - include inefficient layers
- validate assumptions about data volume/reduction rates
- check performance of both designs
 - efficiency
 - rejection
 - timing