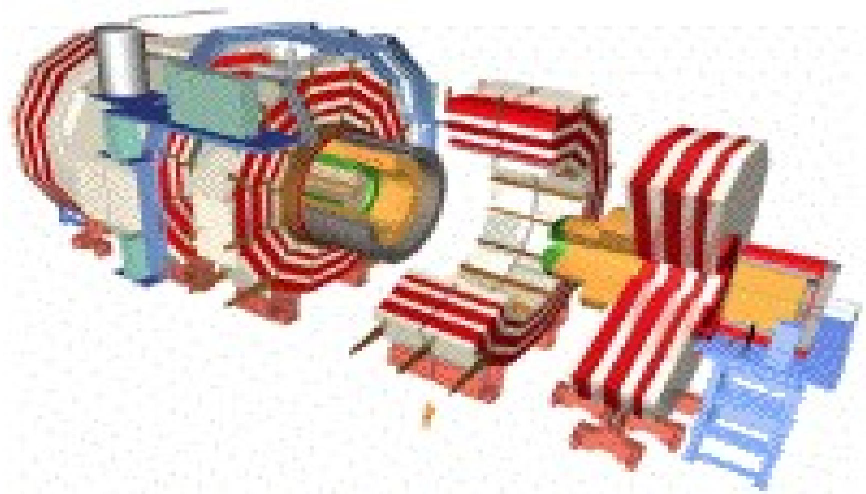
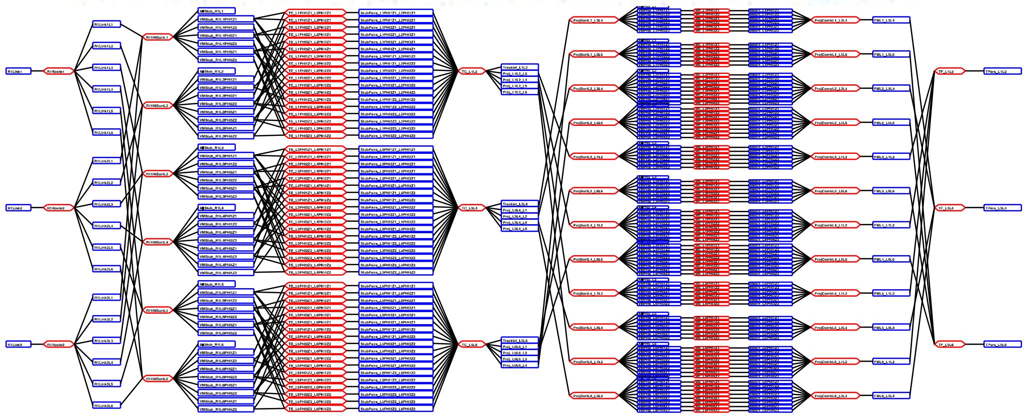
FPGA Based Tracklet Approach for Future L1 Track-Finding Applications





Anders Ryd
(Cornell University)
On behalf of the
CMS Tracklet Group

Presented at CPAD

Oct. 6, 2015

Introduction

- Motivation for tracking in L1
  - CMS phase-2 tracker
- Tracklet algorithm
  - Simulation and challenges
- Tracklet Demonstrator
  - Status and plans
- Next steps and outlook

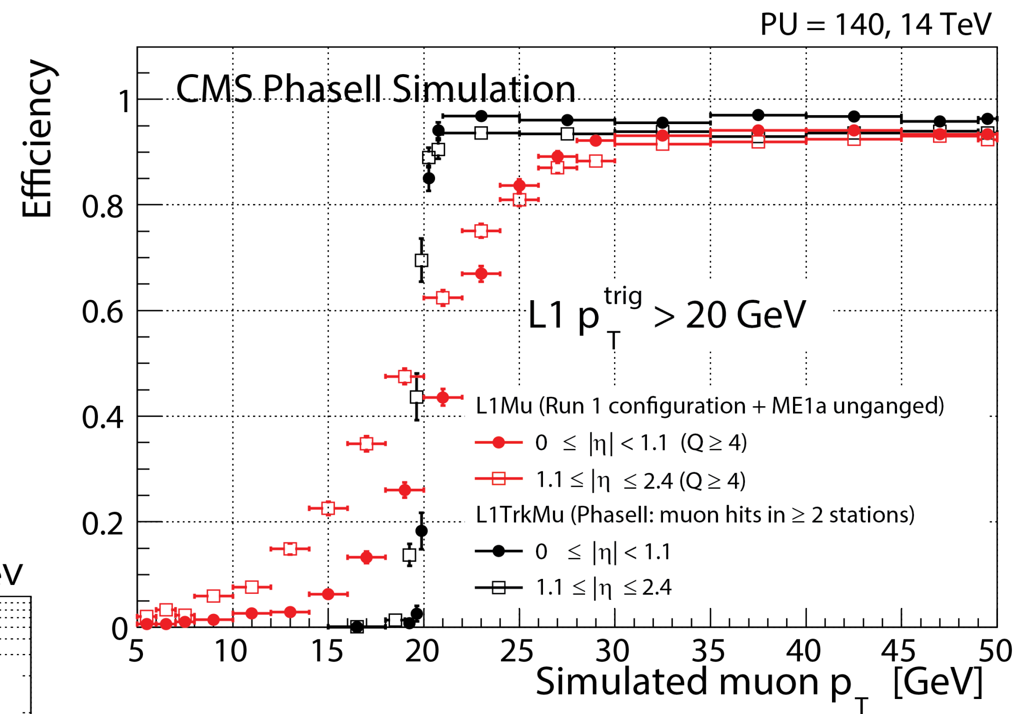
This project has involved developing tracking algorithms suitable for execution on FPGAs

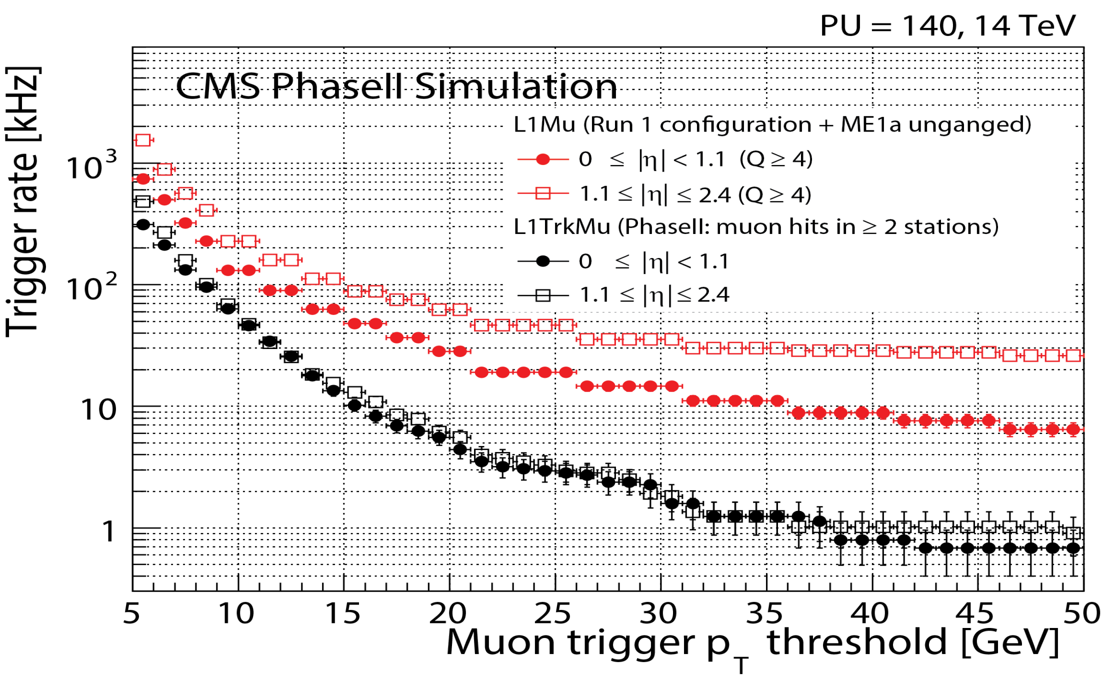
- Student contributions to this effort has been critical
  - Firmware written by students with engineering 'guidance'

The current goal of this work is to demonstrate the feasibility of tracking at 40 MHz by the fall of 2016 for the CMS tracker TDR

Muon Triggers

- Track matching to muon candidates has high efficiency
- Muons+L1Tracks provide much sharper threshold



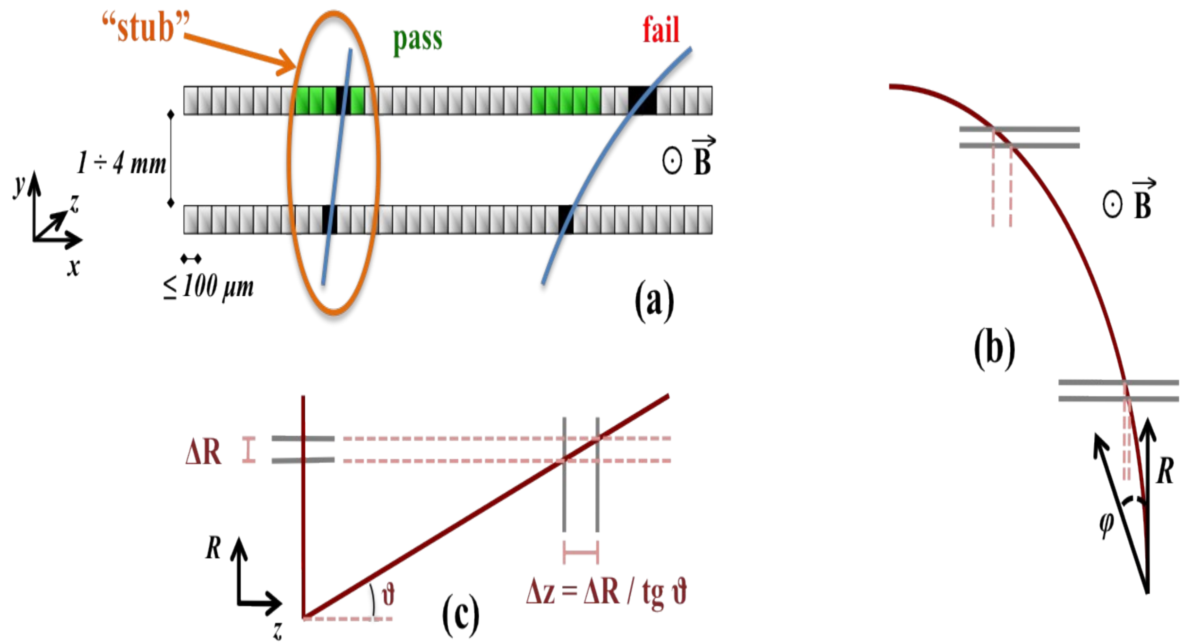


- Sharp threshold allows a significant rate reduction:
  - Factor ~10 reduction @20 GeV

Tracking at L1 is also powerful tool for electrons, taus, jets, photons as detailed in the CMS TP: https://cds.cern.ch/record/2020886

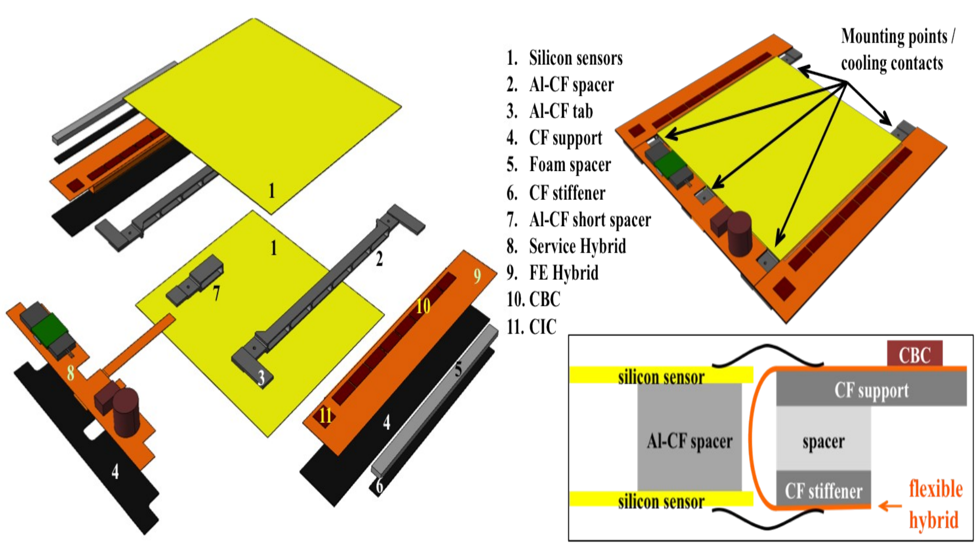
p_T Modules

- Correlating hits in closely spaced sensors give p_T discrimination
- Correlations formed on module – data reduction for trigger readout



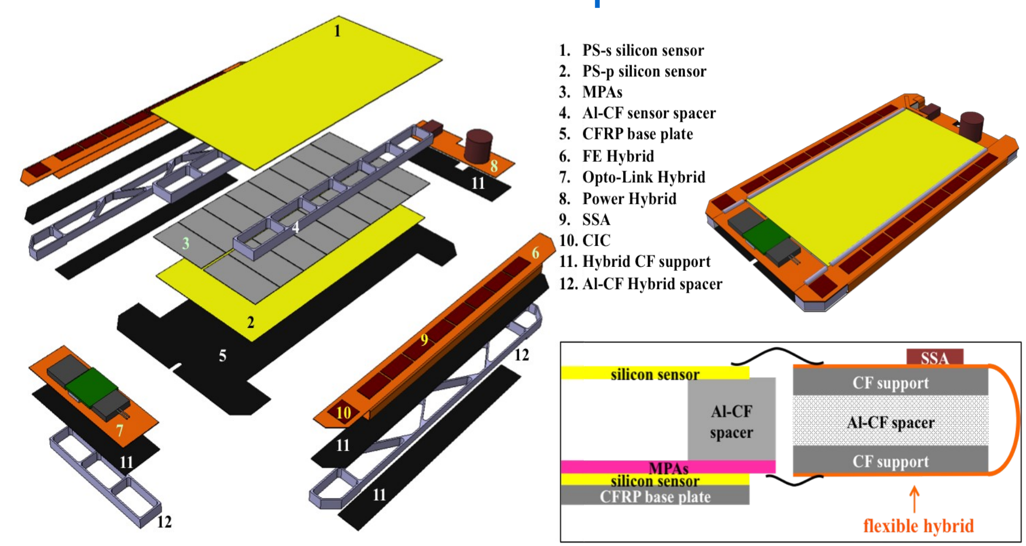
Strip-strip (2S) Modules

2x5 cm strips 90 um pitch

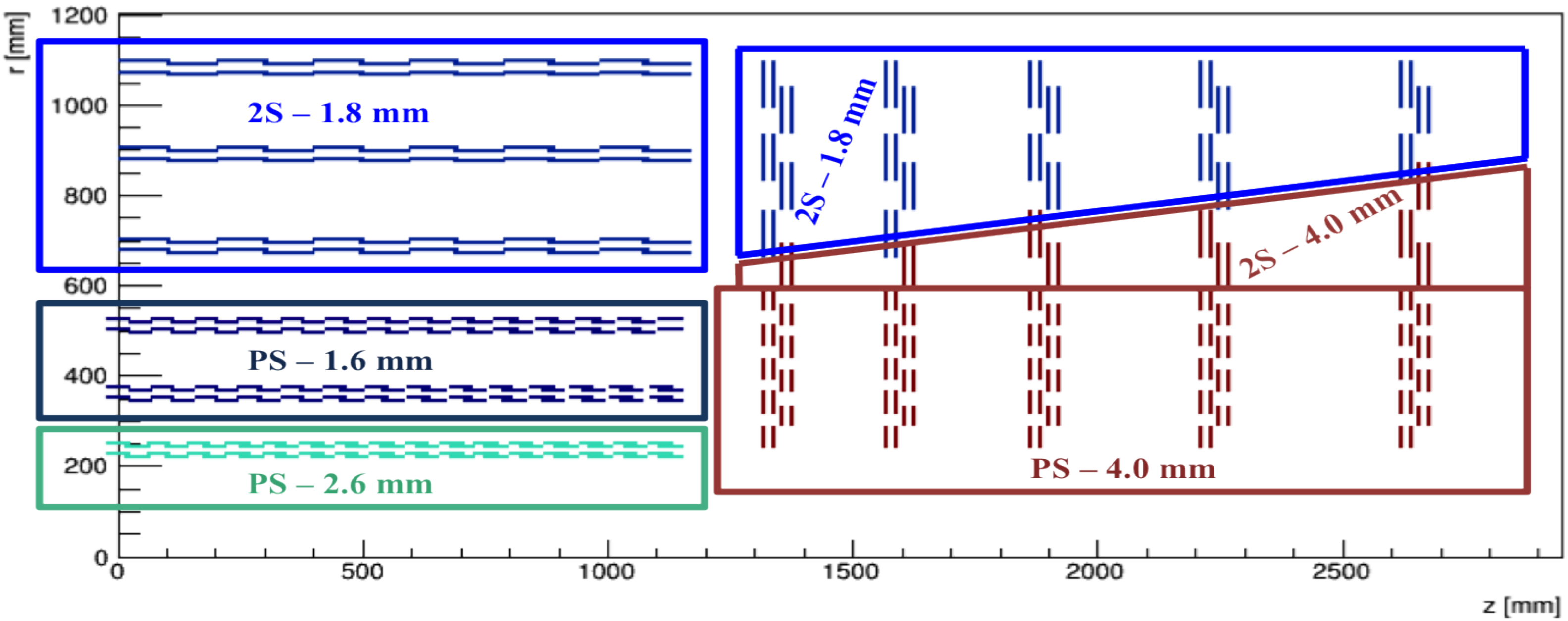


Pixel-strip (PS) Modules

2x2.5 cm strips 100 um pitch
1.5 mm macro pixels



CMS Phase-2 Tracker Layout

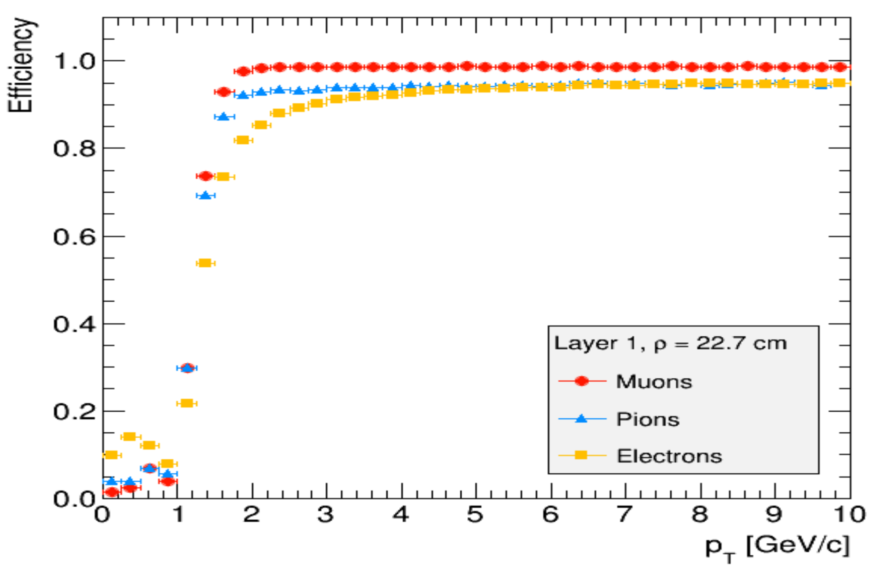


¼ of Outer Tracker

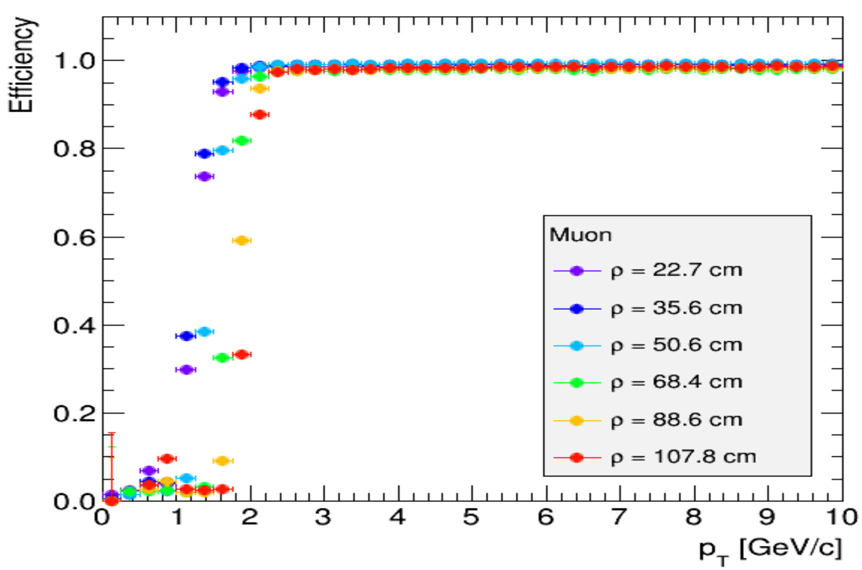
6 Layers

6 Disks

Layer 1 Stub Finding Efficiency



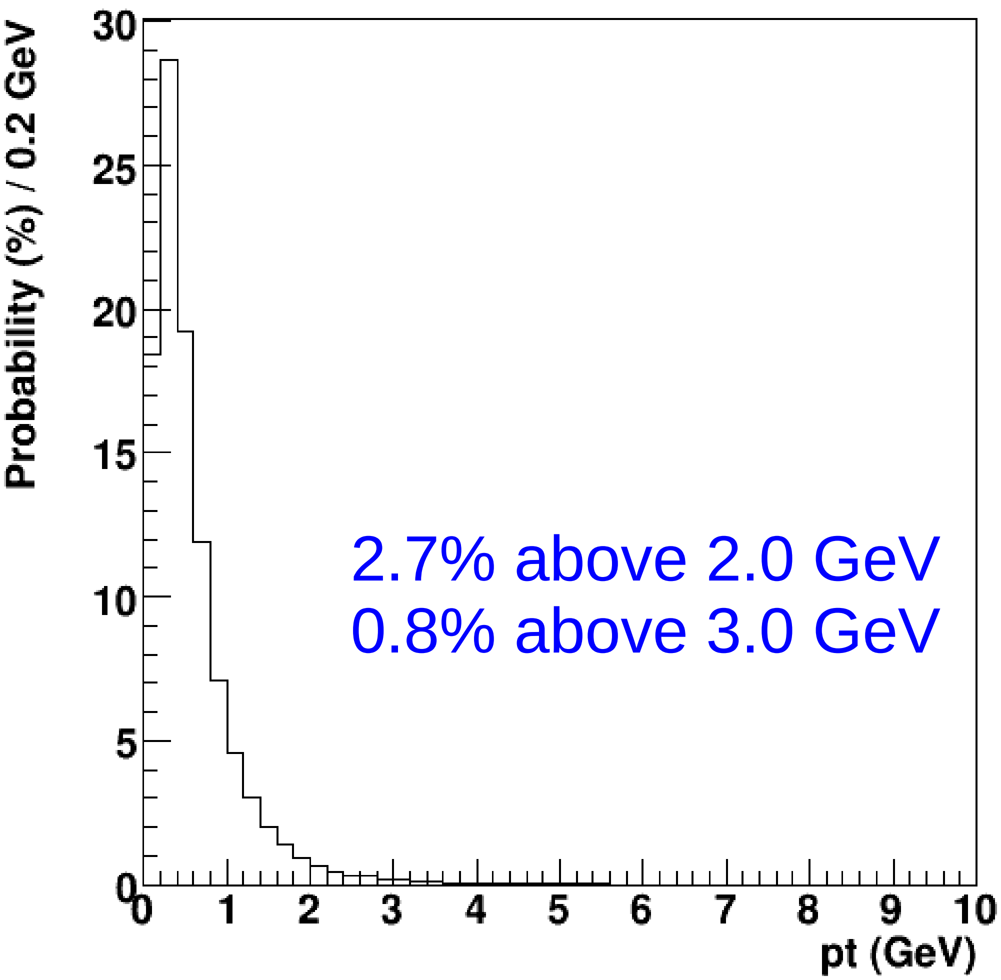
Stub Finding Efficiency per Layer



HL-LHC Environment: Minbias

- 14 TeV minbias:
  - 6.5 charged particles, mostly pions, per unit of rapidity or 33 charged particles in the tracking volume $|\eta|<2.5$
  - With 140 PU → 4600 charged particles per bunch crossing.
  - Soft spectrum – peaks at p_T of about 200 MeV.
- The average min bias event has $33\times2.7\%=0.89$ tracks with $p_T>2.0$.
  - For PU=140 we expect ~125 tracks with $p_T>2.0$.

p_T distribution in 14 TeV minbias



Data volume:
- In 140 PU average of about 12,000 stubs/BX
- Each stub is ~36 bits
  - At 40 MHz BX rate we have 17 Tbits/s

Only ~5% of stubs are from tracks with $p_T>2$GeV

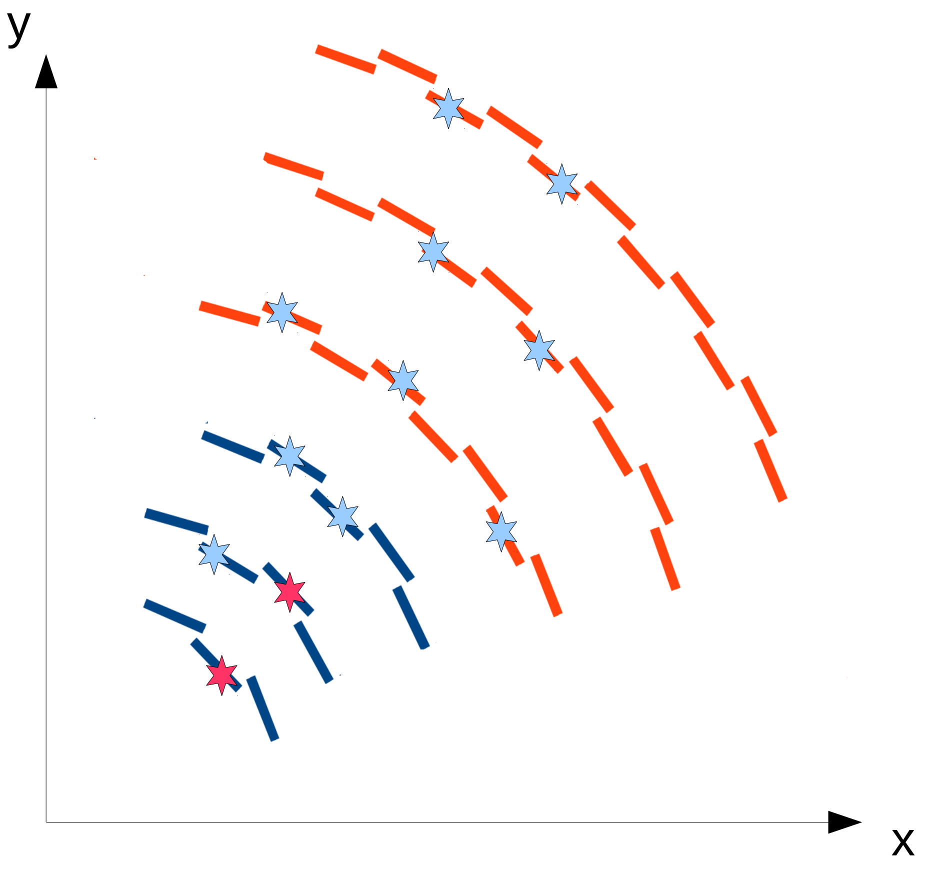
L1 Tracking Goals

- Highest possible efficiency for high p_T isolated tracks
  - Lepton triggers: e, μ, and τ
- Good z resolution for pileup suppression
  - Veto objects from different pp interactions
- Good efficiency for tracking in jets
  - Need to be able to identify jet vertex for pileup suppression
- Efficiency down to p_T=2 GeV
  - Low momentum tracks used for track based isolation
- Good track purity
  - Important e.g. for τ identification

- All this delivered in ~5 μs to meet overall L1 trigger latency requirements

Tracklet Based Track Finding

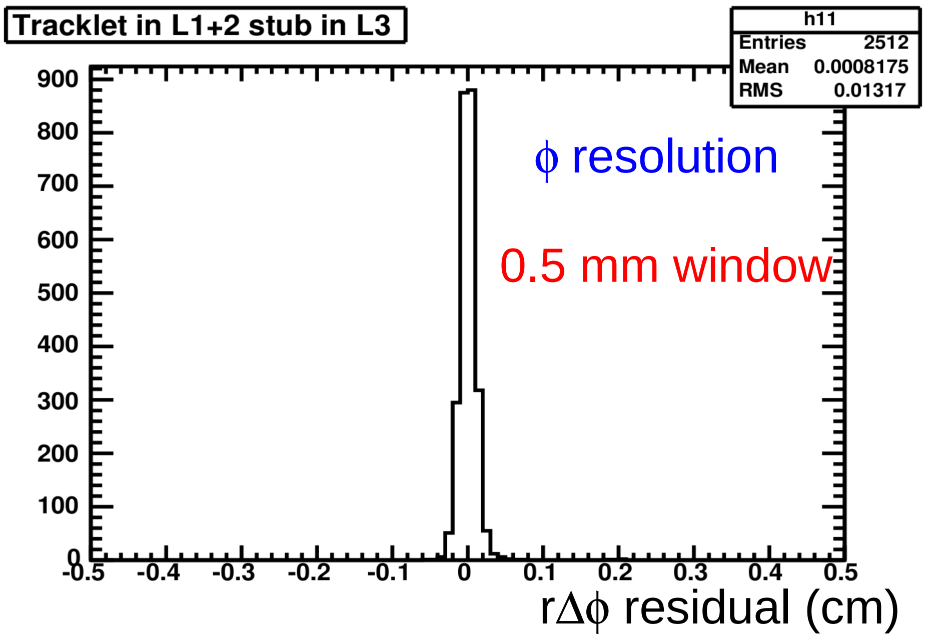
- Form track seeds, tracklets, from pairs of stubs in neighboring layers

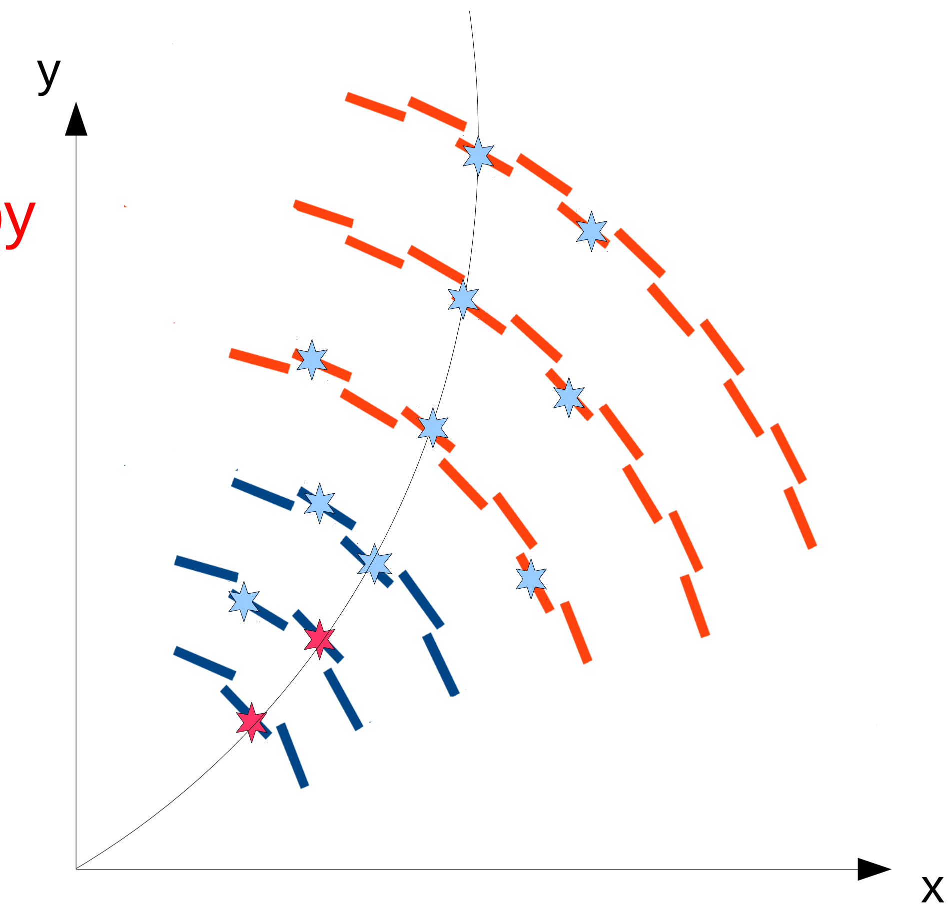


Tracklet Based Track Finding

- Form track seeds, tracklets, from pairs of stubs in neighboring layers
- Match stubs on road defined by tracklet and IP constraint

Matching resolutions for tracklets in L1+L2 projected to L3

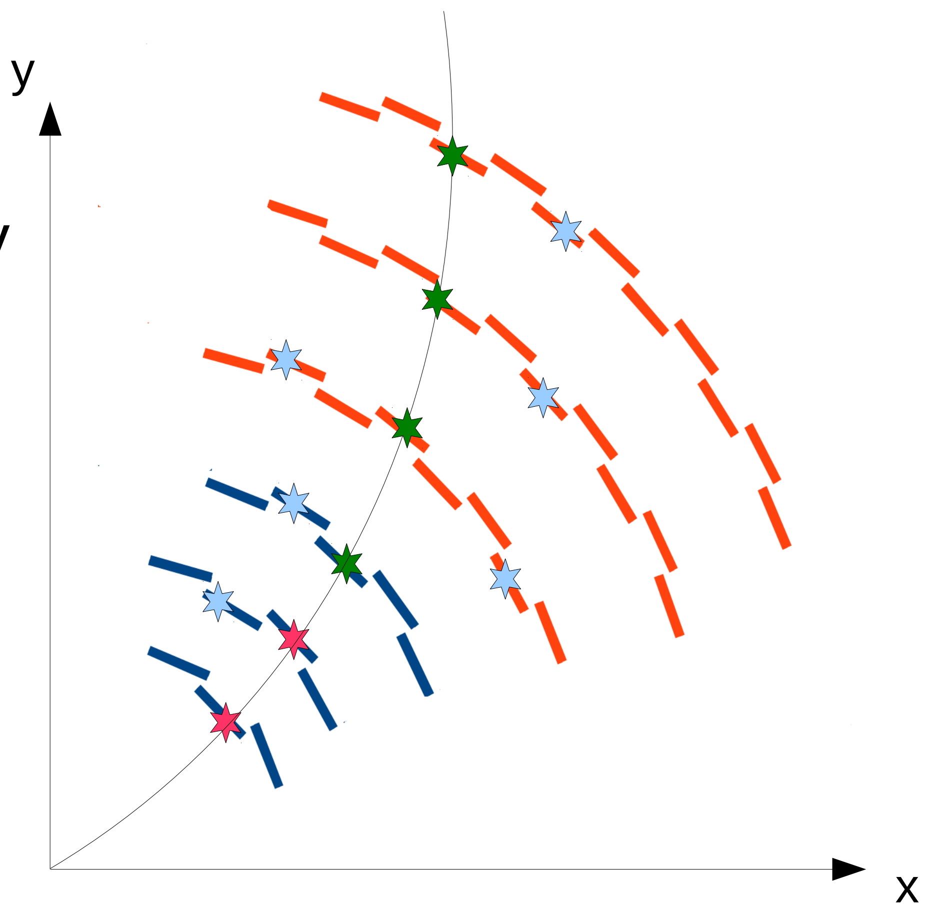




Residual from matching used in final trackfit

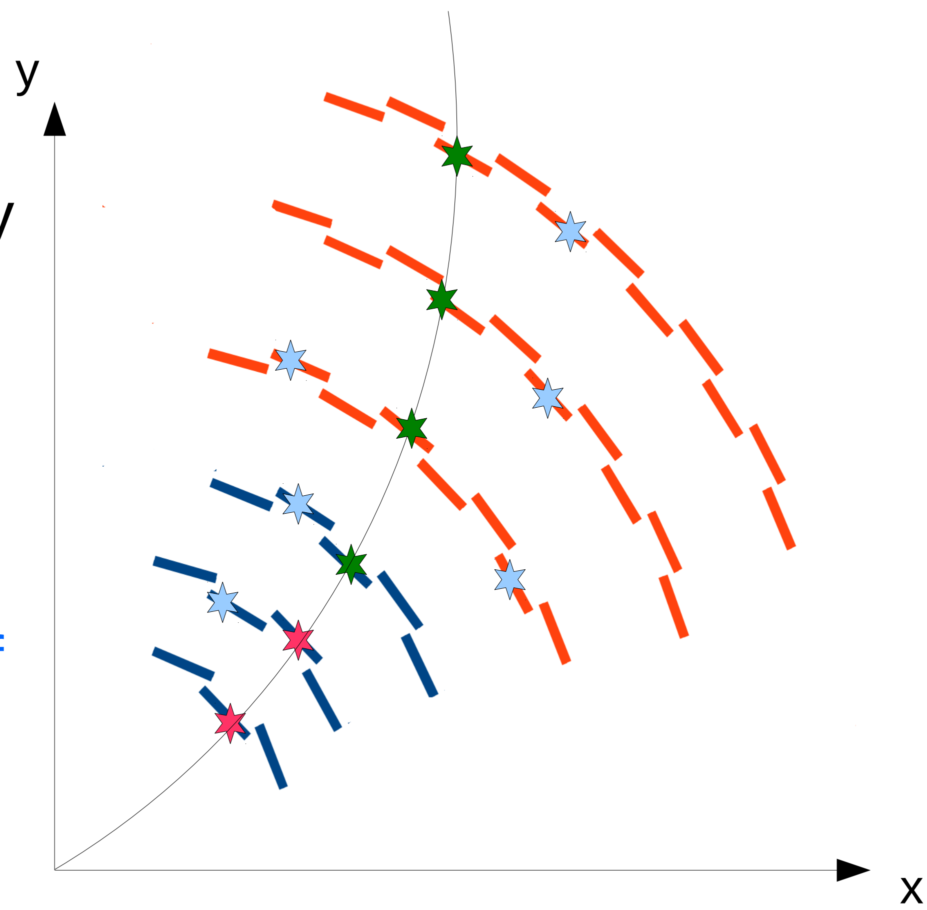
Tracklet Based Track Finding

- Form track seeds, tracklets, from pairs of stubs in neighboring layers
- Match stubs on road defined by tracklet
- Fit the hits matched to the tracklet using a linearized χ^2 fit
  - Tracklet parameters good – linear fit works very will



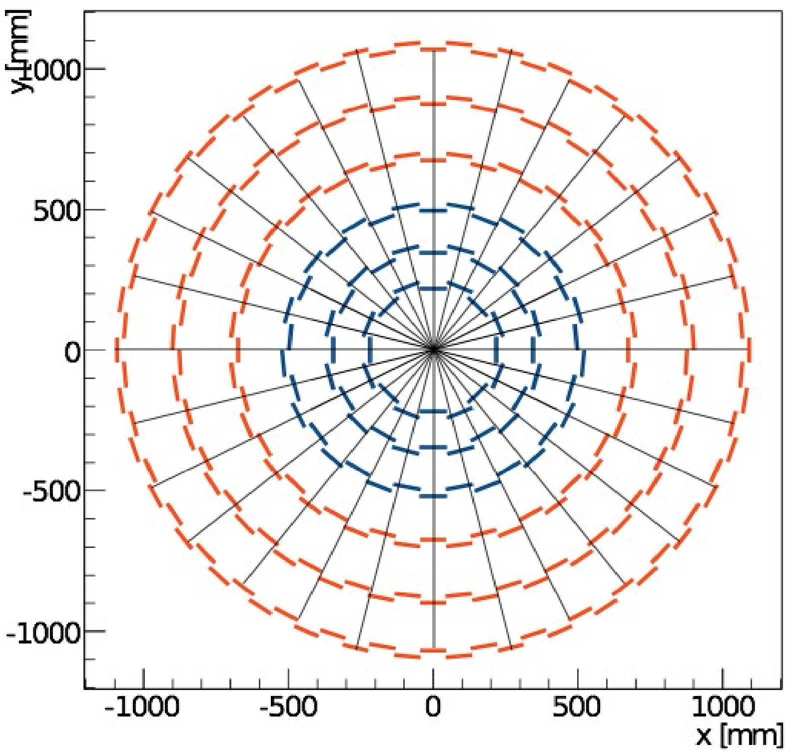
Tracklet Based Track Finding

- Form track seeds, tracklets, from pairs of stubs in neighboring layers
- Match stubs on road defined by tracklet
- Fit the hits matched to the tracklet using a linearized χ^2 fit
- Seeding is done in parallel in different layers
- Duplicate tracks are removed if they share 2 or more stubs



Implementation Approach

- Tracker divided into 28 sectors
  - Maximum number of sectors we can have and contain tracks down to p_T=2 GeV in a sector+nearest neighbor
  - Seeding (tracklet finding) local within sector - ~10% stub duplication
- From simulations (at 140 PU which is the nominal HL-LHC luminosity) we have per sector:
  - ~400 stubs
  - Form ~60 tracklets
  - Find ~10 track (including duplicates)



- What computational resources are need to implement this tracking in and FPGA?

Resource Estimates

- Approximate number of DSP operations required for the different steps in the algorithm

Task	# objects	DSP operations per object	Total DSP operations
Tracklet Parameters	60	20	1200
Tracklet Projections (4 per tracklet)	60	4×10	2400
Matching	200	4	800
Track fit	10	20	200
Total			4600

- Assuming a factor of 4 time multiplexing (100 ns between events) and a 300 MHz project clock each DSP in the FPGA can perform 30 operations. A vertex 7 (690T) has about 3600 DSPs and can perform ~100,000 operations per event
  - We need about 5% of the resources
- The newly announced Ultrascale+ FPGAs will have up to 12,000 DSPs and O(1%) would be needed for the L1 tracking
- Challenge is to handle combinatorics and tails

Baseline Implementation

We have established a 'baseline project':

- Factor of 4 time MUX: Each board receives a new event every 100 ns (LHC bunch crossing frequency is 40 MHz)
- 28 sectors - Tracking performed for p_T>2 GeV
  - Tracklets formed locally in sector
  - Projections sent to neighboring sectors
  - A sector covers the full η range
- Seeding (tracklet finding) done in
  - 3 pairs of layers (L1+L2, L3+L4, and L5+L6)
  - 2 pairs of disks (D1+D2 and D3+D4)
  - Barrel disk overlaps (L1+D1 and L2+D1)
- Find tracks with 4 or more hits
  - Should explore use of tracks with 3 stubs
- Perform track fit
  - With tracklet seed a linearized χ^2 fit works very well
- Remove duplicates – implementation under study

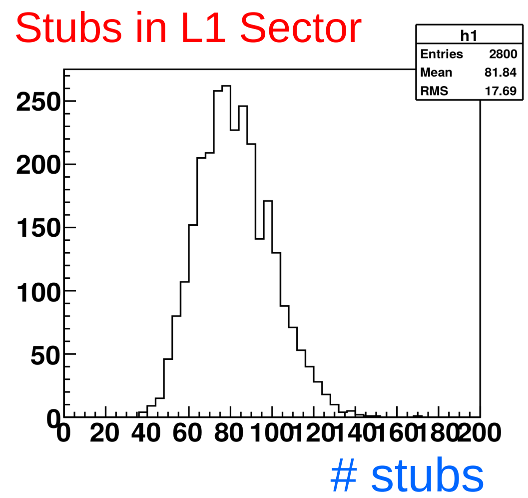
Combinatorics

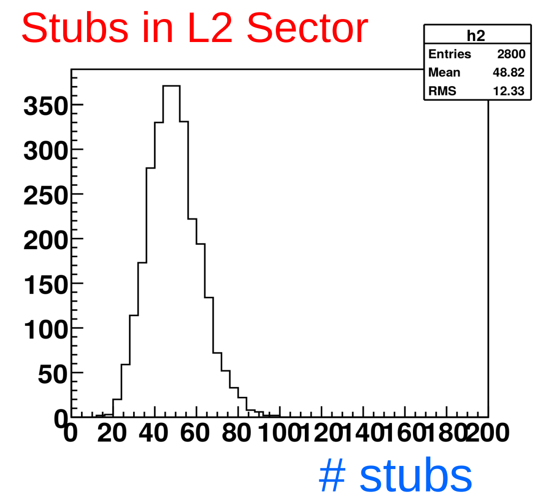
- The obvious challenge for implementing a tracking algorithm with a fixed latency is how to handle the combinatorics.
- The tracklet approach described here has two key steps were combinatorics is an issue:
  - Forming tracklets from pairs of stubs
  - Matching tracklets projected to other layers (or disks) to stubs
- Both cases are addressed in a similar manner:
  - The detector is divided into smaller regions and the combinatorics problem is solved by massive parallelism

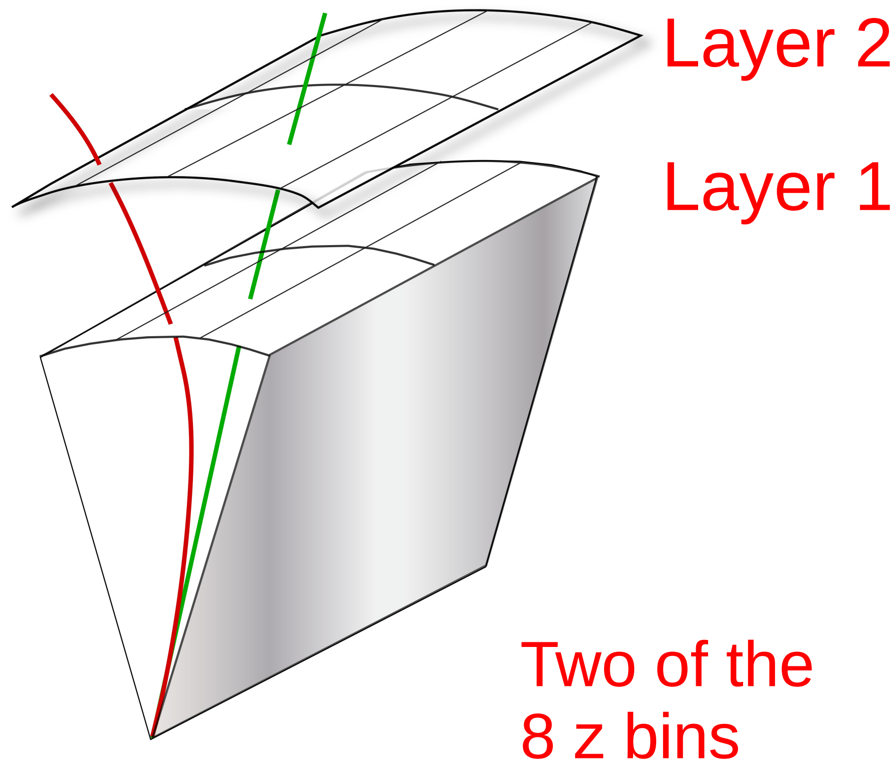
- To illustrate this we consider the tracklet finding in some detail
  - Remember that we work here in one sector, so 1/28 of the full detector
  - The plots of the occupancy is taken from a sampler of $t\bar{t}$ with 140 minbias events superimposed

Tracklet Formation

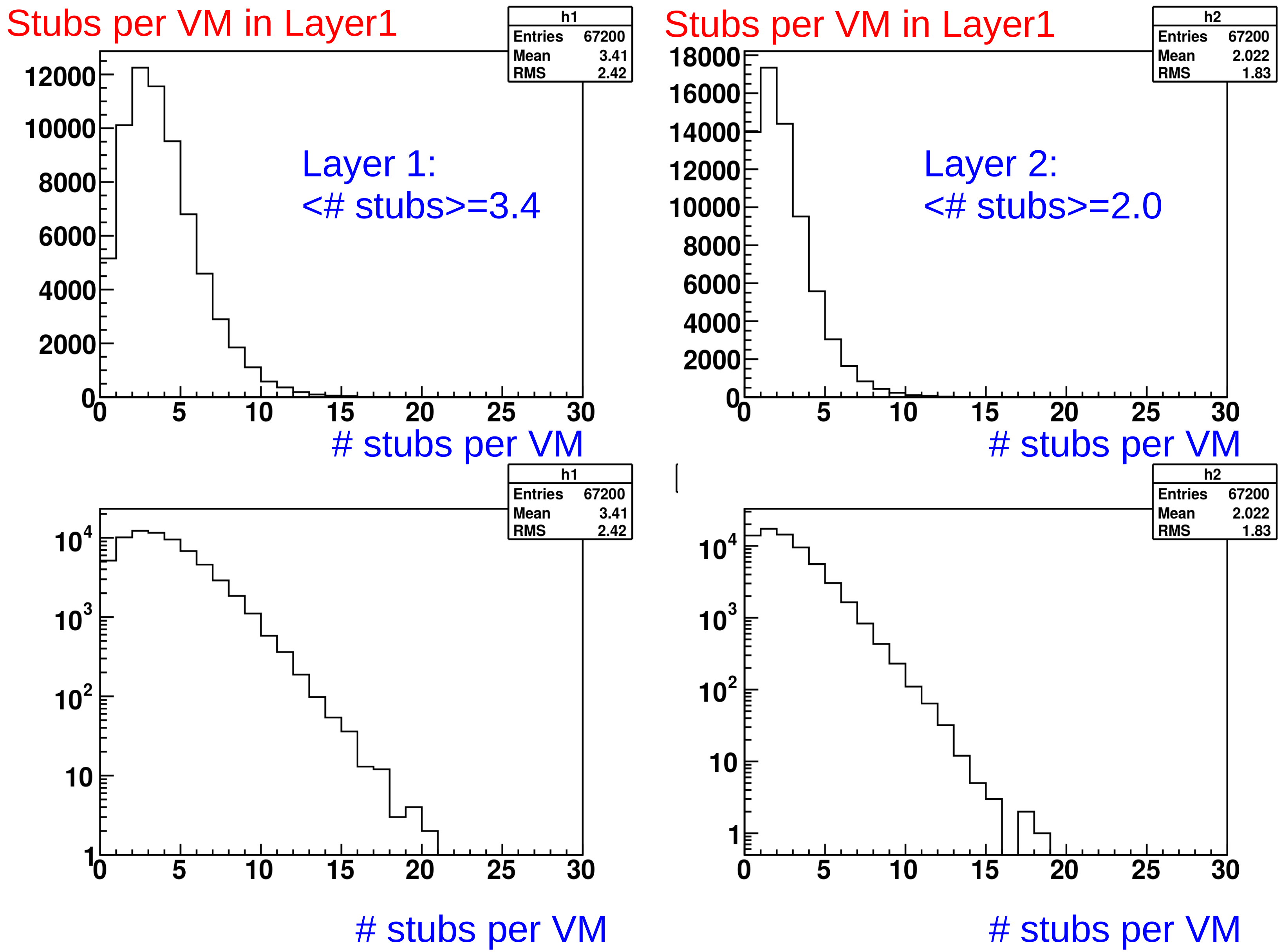
- Average number of stubs in a sector:
  - Layer 1: ~80 stubs
  - Layer 2: ~50 stubs
- Total of ~4000 possible pairs
  - Most of these don't satisfy:
    - $|z_0|<15$ cm
    - $p_T>2$ GeV
- Divide each layer into sub regions ('Virtual Modules' VMs)
  - Layer 1: $8\ z \times 3\ \phi = 24$ VMs
  - Layer 2: $8\ z \times 4\ \phi = 32$ VMs
- Total of $24 \times 32 = 768$ pairs of VMs.
  - 96 of these can form tracklets
  - Search for tracklets in parallel in these 96 pairs of VMs.







Virtual Module Occupancy

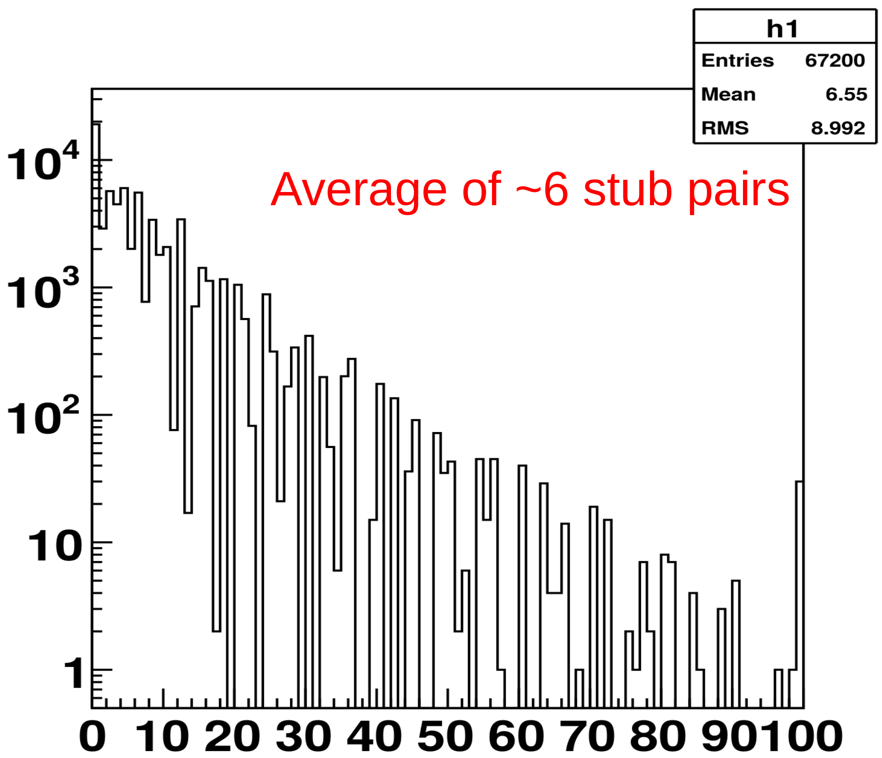


Linear scale

Log scale

Tracklet Formation Combinatorics

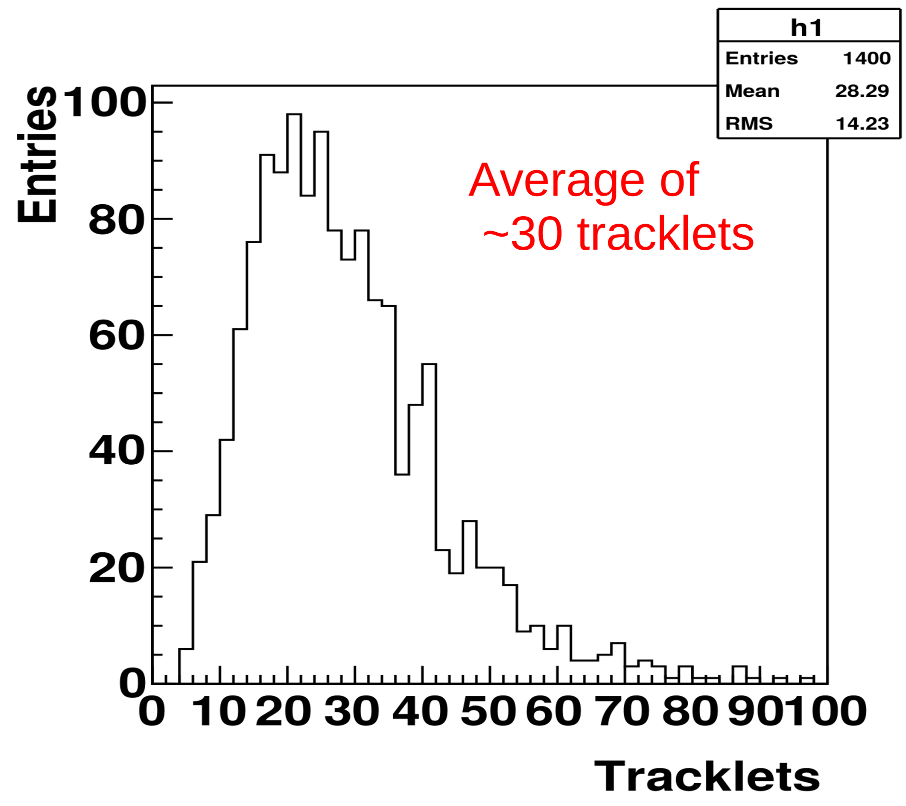
Stub pairs per virtual module combination



stubs pairs

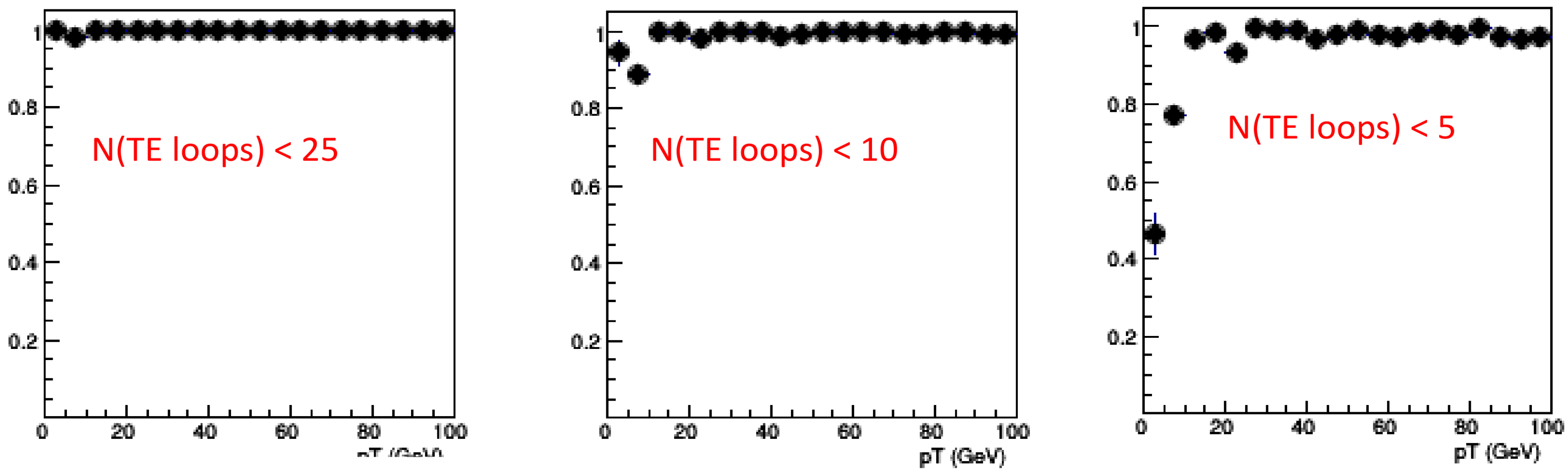
Tracklets found in L1+L2:

- $|z_0|<15$ cm
- $p_T>2$ GeV



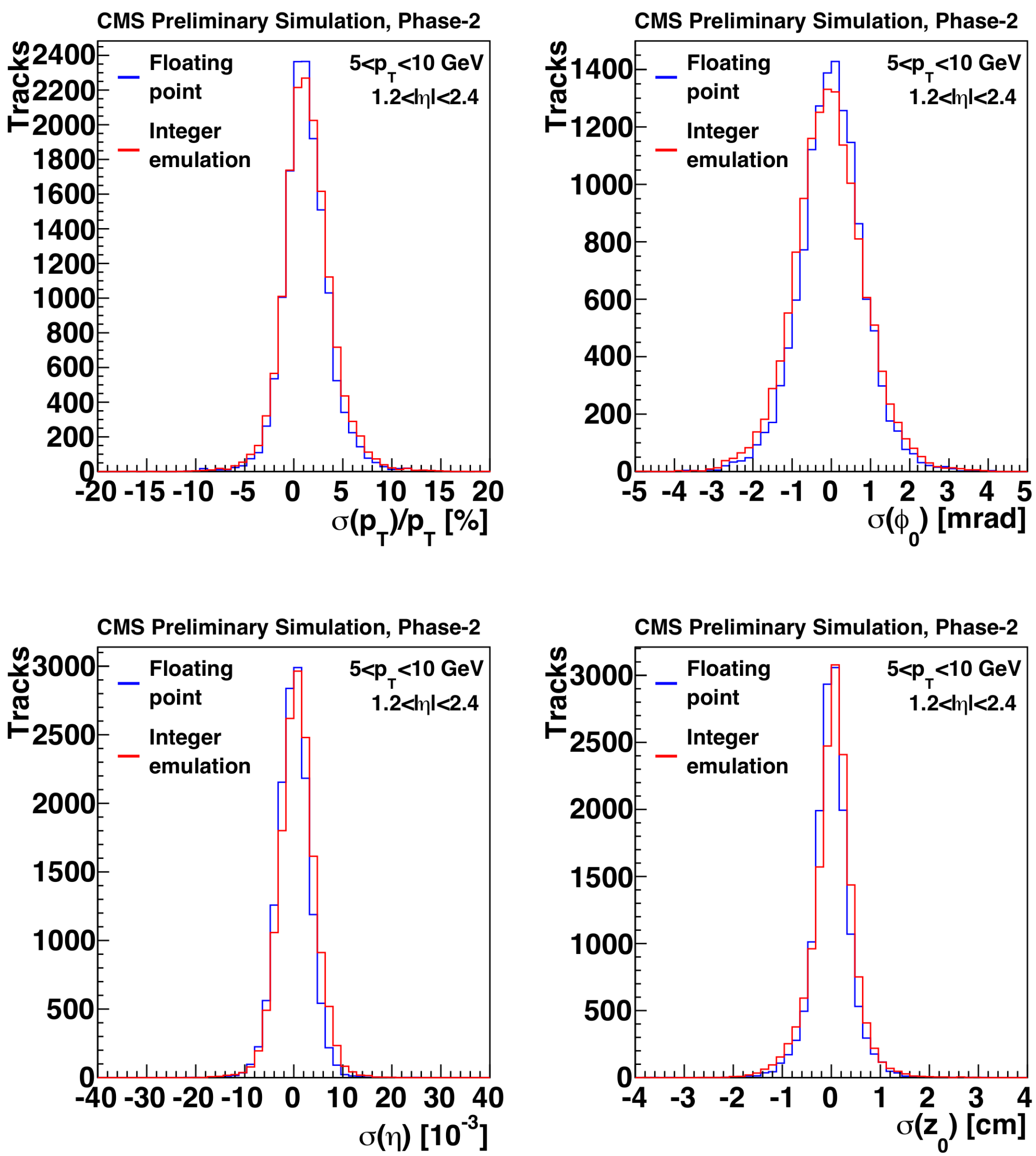
Truncating Tracklet Finding

- There will be a limit to how many stub pairs we can try in the tracklet engine (TE)
- The effect of the truncation for finding isolated muons in events with 140 PU is illustrated below



- Currently our implementation can comfortably process 25 stub pairs per 100 ns (factor of 4 time multiplexing)
- Redundancy from multiple seeding layers gives good performance for isolated tracks.
  - Performance studies in jets are underway

Track Fit Performance



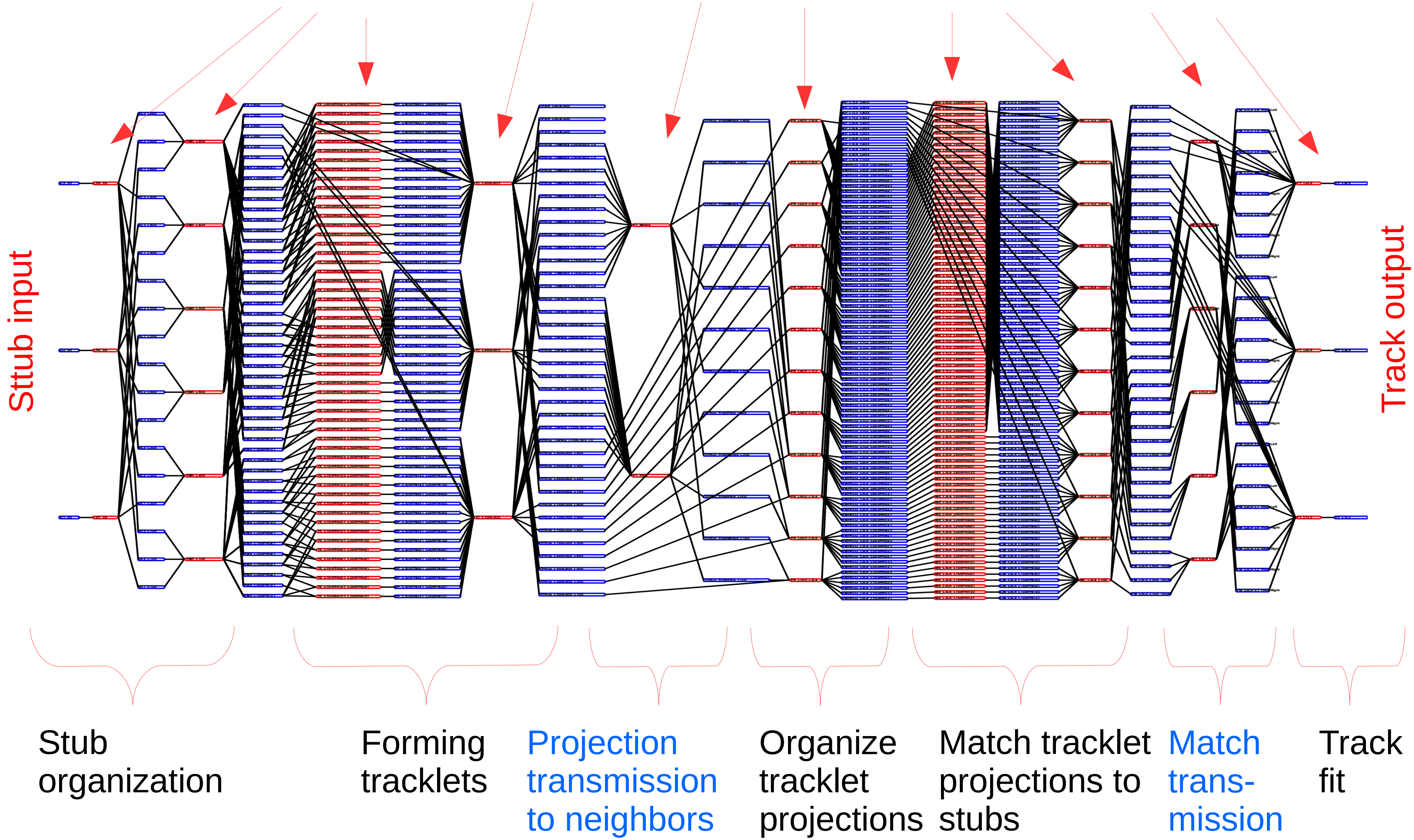
The integer performance and full floating point calculation gives the same track parameter resolution

Implementation Status

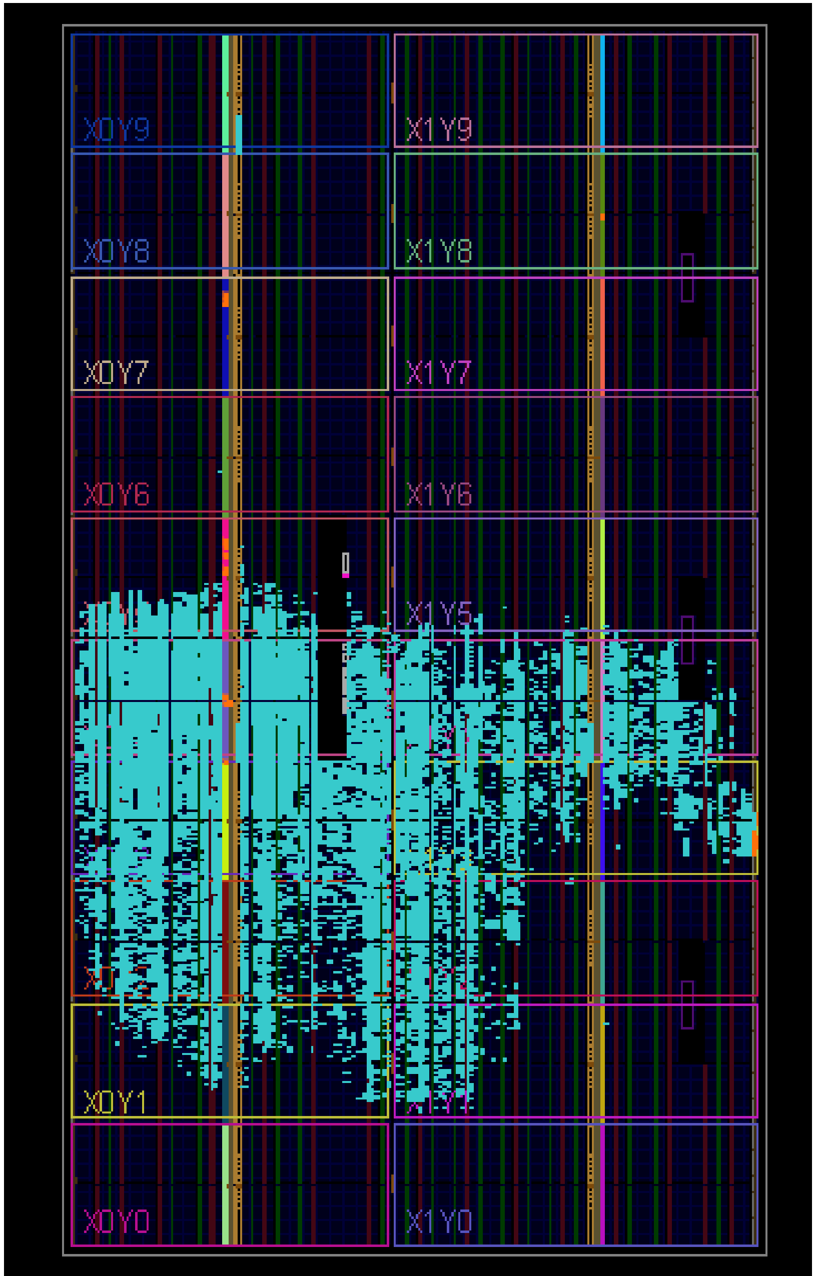
- The Tracklet algorithm is implemented in firmware in 8 core processing steps
  - These 8 steps do not include the duplicate removal
    - Duplicate removal is currently under study
- The barrel part of the algorithm is well developed:
  - Firmware written in verilog
  - C++ emulation – reproducing bitwise result from firmware
  - Firmware for ¼ of barrel sector uses about ~25% of FPGA resources in the 690T Virtex7 FPGA
  - Firmware is auto generated from high level routing information

Overview of Project (¼ of Barrel)

Eight processing steps + two transmission (red) implements the algorithm



Firmware Status



- Current project runs at 320 MHz
- Left: Picture of the resources used (cyan) in the Virtex 7 FPGA (690T)
  - 25% BRAM
  - 10% LUT
  - 22% BUFG
  - 6% DSP
- Higher occupancy of the chip will impose tighter timing constrains due to routing.

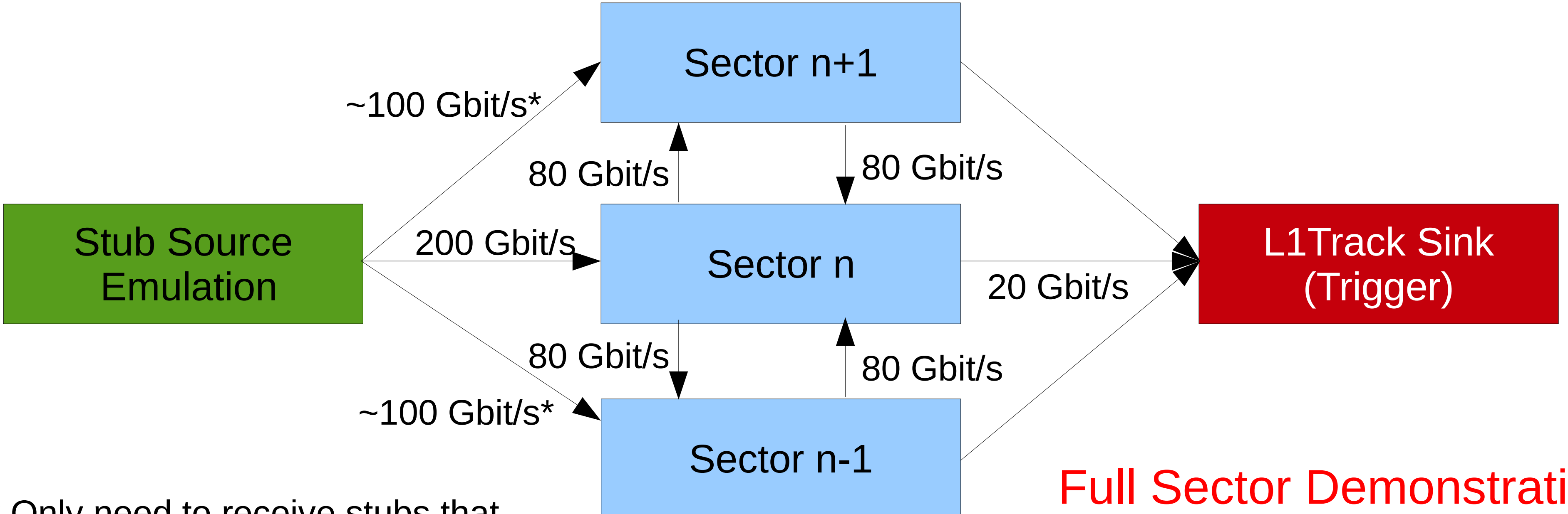
Latency Budget

- Assume 10 processing steps
  - 8 for track finding/fitting and 2 for duplicate removal
- Assume 3 data transfers
  - Tracklet projections, stub matches, and duplicate removal

Task	Time Multiplexing	
	4	8
10 Processing Steps	1000 ns	2000 ns
Processing Latency	450 ns	450 ns
3 Data transfer steps	300 ns	600 ns
3 Link latencies	300 ns	300 ns
Total Latency	2050 ns	3350 ns

- With a latency goal of about 4 μs for the L1 tracking a TMUX of 8 still fits within the latency
  - With a TMUX of 4 we are well within the target latency

Tracklet Demonstrator



* Only need to receive stubs that needed to demonstrate functionality of the central sector

Full Sector Demonstration:

- Covers |η|<2.4
- Uses factor 4 time MUX
- p_T>2 GeV

Sector board

Input: 360 Gbit/s

Output: 180 Gbit/s

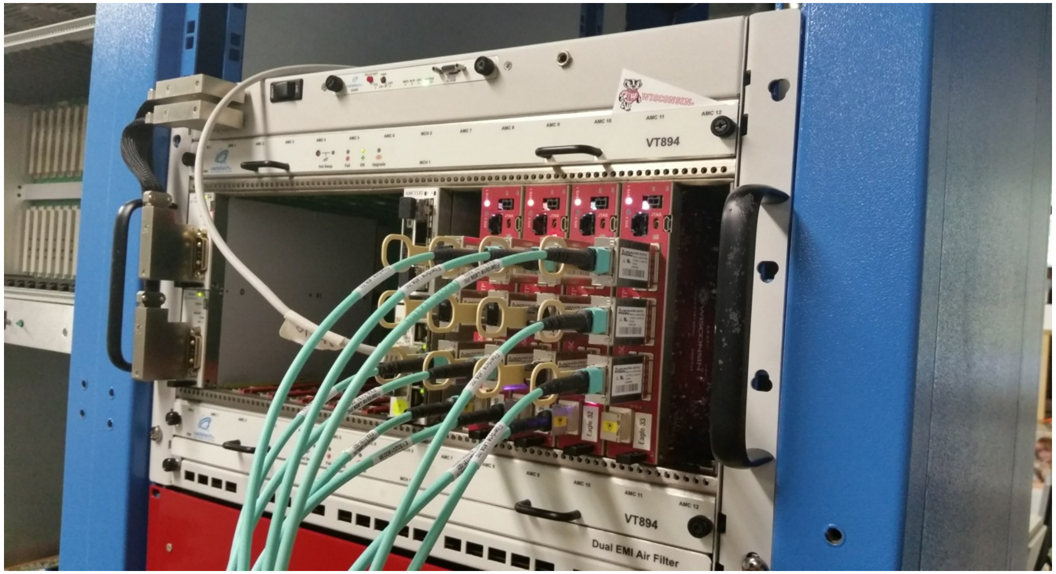
Board used for stub source emulation and the L1Track receiver can be the same

Allow measurement of latency from stub input to tracks available

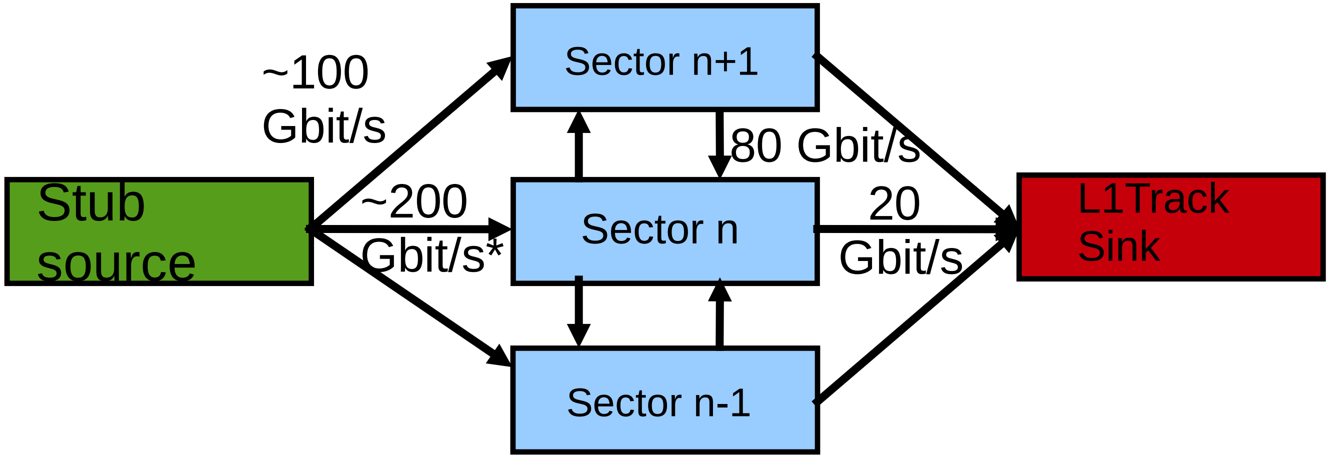
Sector Demonstration

- Goal: Establish overall viability of tracklet approach.
  - Firmware/Algorithm
  - I/O Tests (input, neighbor, output)
  - Latency
  - Stress tests with data volumes
- Setup:
  - TFirmware/Algorithm
  - I/O Tests (input, neighbor, output)
  - Latency
  - Stress tests est crate at CERN
  - Four µTCA cards (CTP7)
    - 4th acting as data source/sink.
    - Each board has a Xilinx Vertex-7 FPGA and a Zynq chip.
    - Core sector board + 2 neighbors
  - AMC13 card for clock dist.

Test crate at CERN



µTCA boards (CTP7) developed by University of Wisconsin for the current 2016 Level-1 Trigger upgrade.



Demonstrator Status

- Firmware:
  - Project representing ¼ of tracker barrel has been implemented in a Vertex-7 chip
  - Running with 4x TMUX (100 ns/event)
- Milestones achieved:
  - Sector boards synchronized
  - Input stubs sent from input source
  - Projections to adjacent sector boards sent & received
  - Full tracks found and results sent to output sink

- Future milestones
  - Send/receive matches from adjacent sector boards
  - Scale to larger area
  - Include forward disks

Track Info: 24 bits/link

Base RX capture memory address for channel 16: 0x61010000
Number of BXs to readout: 5

BXID	Word0	Word1	Word2	Word3	Word4	Word5
0000	0x000000bc	0x00000000	0xb7796900	0x0047f800	0x00000000	0x00000001
0001	0x00000001	0x00000001	0x00000001	0x00000001	0x00000001	0x00000002
0002	0x00000002	0x00000002	0x00000002	0x00000002	0x00000002	0x00000003

Base RX capture memory address for channel 18: 0x61012000
Number of BXs to readout: 5

BXID	Word0	Word1	Word2	Word3	Word4	Word5
0000	0x000000bc	0x00000000	0x00949900	0xdfffce00	0x00000000	0x00000001
0001	0x00000001	0x00000001	0x00000001	0x00000001	0x00000001	0x00000002
0002	0x00000002	0x00000002	0x00000002	0x00000002	0x00000002	0x00000003
0003	0x00000003	0x00000003	0x00000003	0x00000003	0x00000003	0x000000df
0004	0x0000003c	0x00000004	0x00000004	0x00000004	0x00000004	0x00000005
0005	0x00000005	0x00000005	0x00000005	0x00000005	0x00949905	0xdfffce06

Example: 0x000000_000038_009499_b77969

Stub Indices p_T ϕ_o $\tan(\theta)$ z_o

Summary/Outlook

- We are proceeding with the implementation of the demonstrator
  - No major issues encountered so far
- The current generation of DSPs we are using (Virtex 7, 690T) will not allow us to scale up to a full sector for the full 'baseline' project
  - A full sector with a reduced number of seeding layers should be possible and will be explored
  - Much more powerful FPGAs already available or announced

FPGA	Memories	DSPs
Virtex 7	~50 Mbits	3,600
Ultrascale	~130 Mbits	5,500
Ultrascale+	~500 Mbits	11,900

- The next generation FPGAs should allow implementation of the full tracklet project
- In the process of implementing the current project we have realized several optimizations that will be explored later

Backup

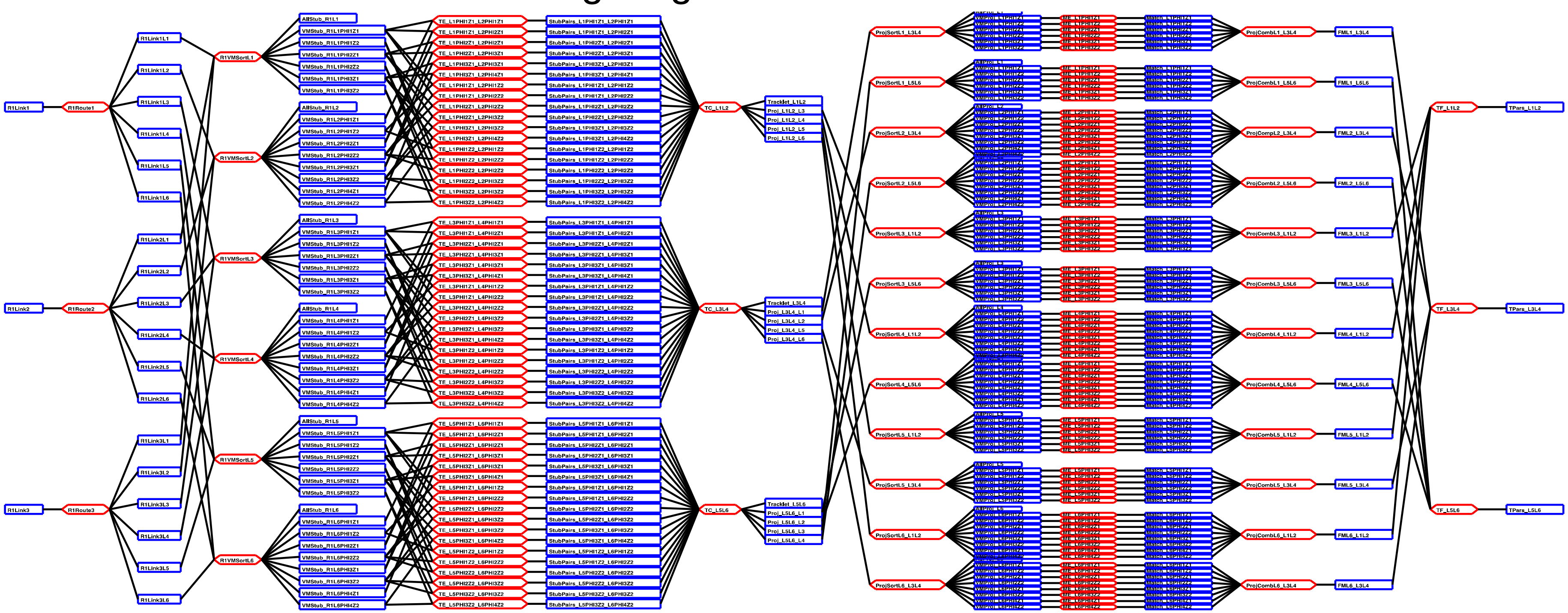
Projections to Neighbor Sectors

- Tracklet projections to a neighboring sector needs to be sent to this sector to be matched with stubs
  - We send a tracklet ID, and projection information to the neighboring sector
    - About 50 bits are required per projection
  - In the worst case about 100 projections needs to be sent to each neighboring sector
    - See B. Winer's presentation
- With our baseline with a factor of 4 time MUX we need 50 Gbits/s
  - About 8 links (at 10 Gbits/s) will be needed

Scaling Project beyond ¼ Sector

- Software to generate the high level routing diagram for the project has been extended beyond ¼ of a sector (at 'DCT region')
  - Includes full barrel
  - Includes endcap disks
  - Includes overlap regions (seeds formed in layer+disk)
  - Includes additional modules required for neighboring sector communication

Routing diagram for ¼ sector

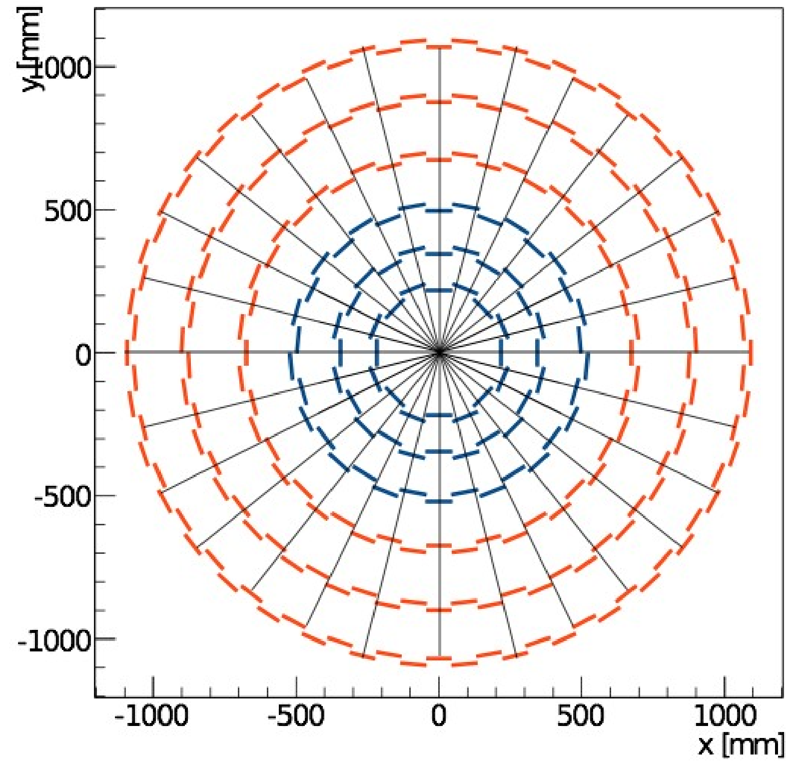


DCT Functionality

- The DCT receives the stubs from the front ends on one optical link per module
  - The stub data are averaged over 8 BXs
- The functionality we require from the DCT for the tracklets is:
  - Extract the stub data from the payload
  - Identify the correct BX for each stub
  - Translate stub from local to global (r,ϕ,z) coordinates
  - Route the stubs to the correct sector board based on BX (to implement the time multiplexing) and global coordinate
- We want to understand the latency and resources required to implement these steps
- We will not address any other functionality of the DCT

Hardware Configuration

- Parallel Processing
  - ➡ Divide tracking system into 28 sectors in φ.
    - ▶Min P_T: 2 GeV
    - ▶Tracks contained within a sector and nearest neighbor, which simplifies communications.
  - ➡ Each sector has a dedicated processor board.
  - ➡ 4-8x time multiplexing
    - ▶@ 4x TM: each sector receives new event every 100 ns.
    - ▶Current spec: Latency < 4 μs.

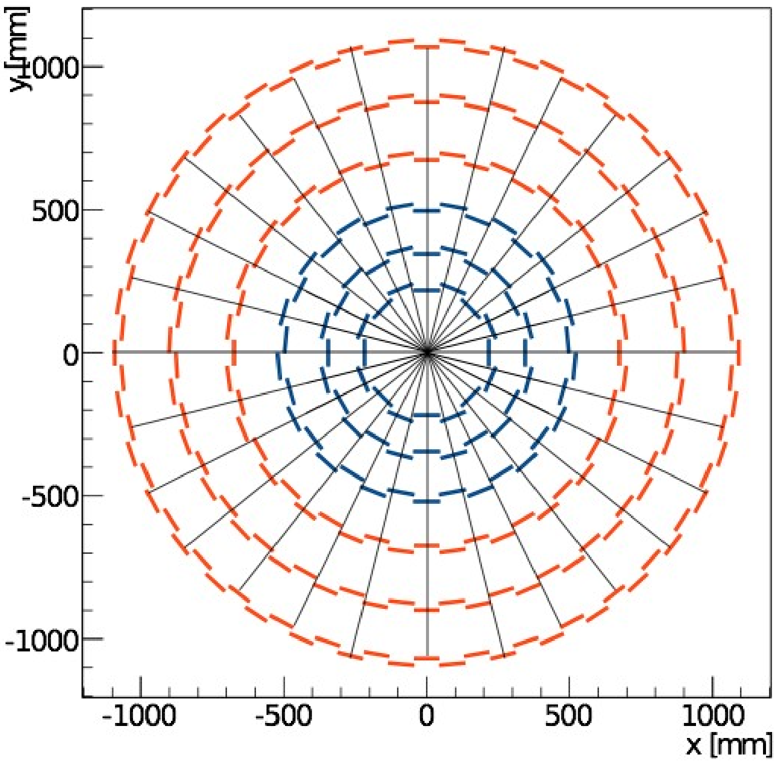


- Sector Processors:
  - ➡ Target large commercial FPGAs
  - ➡ I/O:
    - ▶Stub Input: ~200 Gbits/s
    - ▶Neighbor Boards: 2 x 80 Gbits/s
    - ▶Track list: ~20 max trk/sec/event = ~20 Gbits/s

Downstream trigger systems will associate tracks with calorimeter and muon system objects to form L1 trigger objects.

Hardware Configuration

- Parallel Processing
  - ➡ Divide tracking system into 28 sectors in φ.
    - ▶ Min P_T: 2 GeV
    - ▶ Tracks contained within a sector and nearest neighbor, which simplifies communications.
  - ➡ Each sector has a dedicated processor board.
  - ➡ 4-8x time multiplexing
    - ▶ @ 4x TM: each sector receives new event every 100 ns.
    - ▶ Current spec: Latency < 4 μs.
- Sector Processors:
  - ➡ Target large commercial FPGAs
  - ➡ I/O:
    - ▶ Stub Input: ~200 Gbits/s
    - ▶ Neighbor Boards: 2 x 80 Gbits/s
    - ▶ Track list: ~20 max trk/sec/event = ~20 Gbits/s



Downstream trigger systems will associate tracks with calorimeter and muon system objects to form L1 trigger objects.

Baseline Assumptions for Demonstration

Full system (our baseline):

- 28 ϕ-sectors – full η range
  - Sectors communicate with nearest neighbors
- Factor 4 time multiplexing
- DCT transforms local to global coordinates and performs first step of data routing

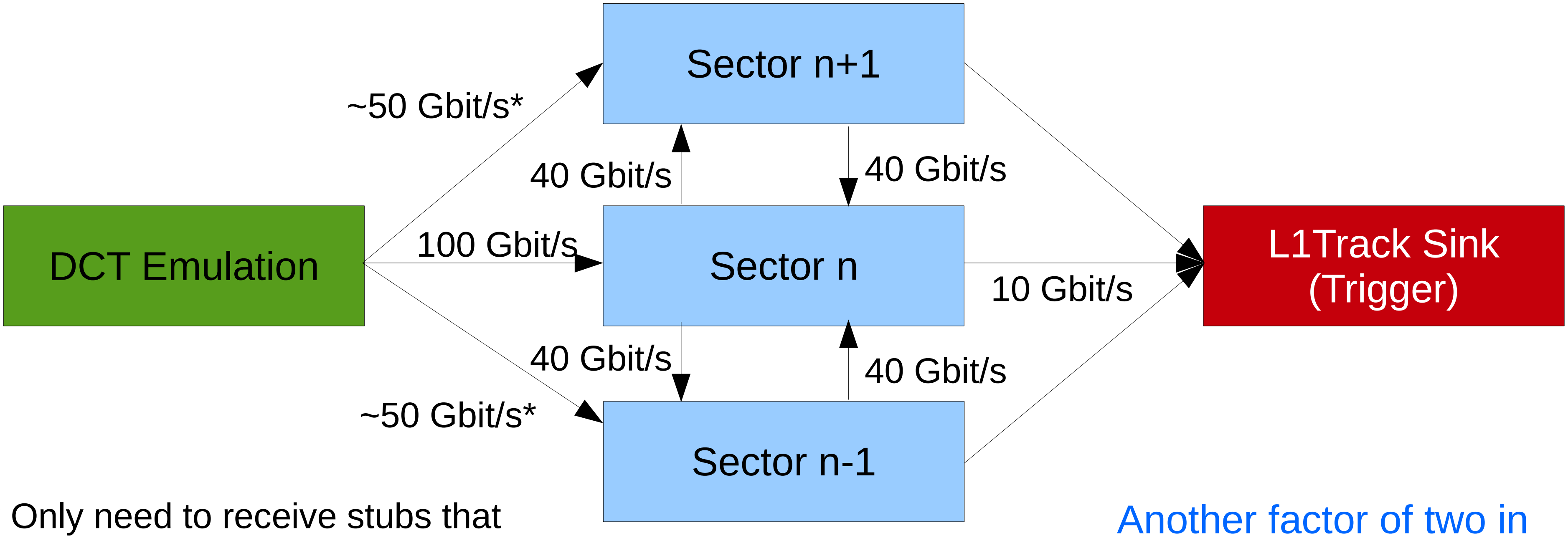
Demonstrators:

- We will separately demonstrate the functionality of:
  - L1Track finding/fitting
  - Functionality of DCT for trigger data
- We will measure the latency separately for each of these steps.
  - Combined with the time it takes for stub data to be sent from the front end to the input of the DCT we can establish the latency of the full chain

Tracklet Demonstrator - Reduced Scope

A reduction by a factor of two of the I/O needs is easily obtained by:

- Implementing half a sector ($\eta>0$)
- Using factor 8 time MUX



* Only need to receive stubs that needed to demonstrate functionality of the central sector

Sector board

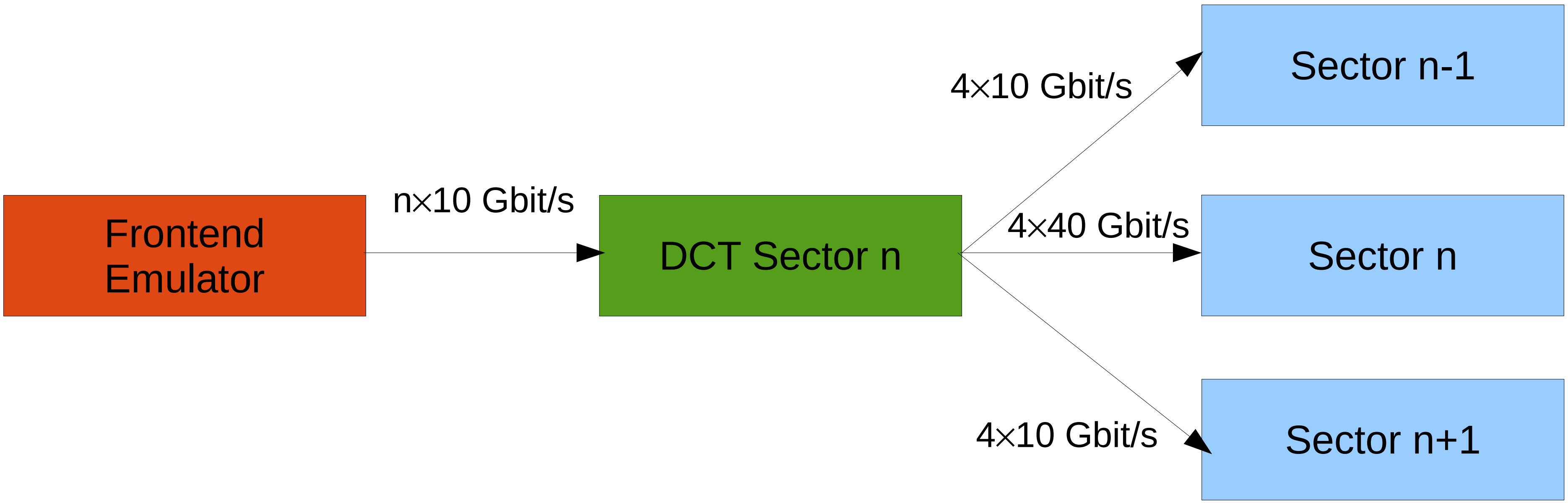
Input: 180 Gbit/s

Output: 90 Gbit/s

Another factor of two in performance should be available with the next generation FPGAs and links

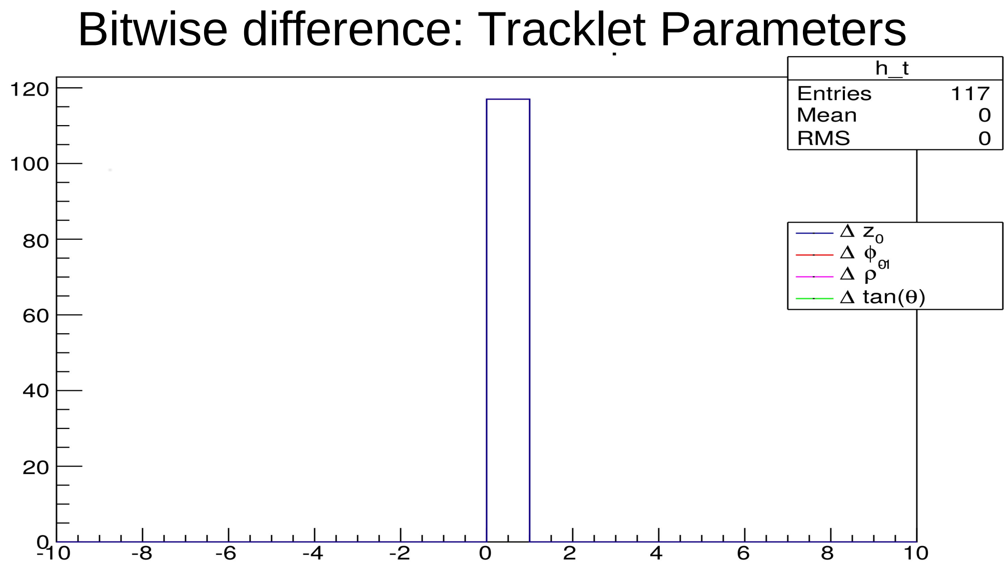
Still provides a substantial demonstration of the algorithm

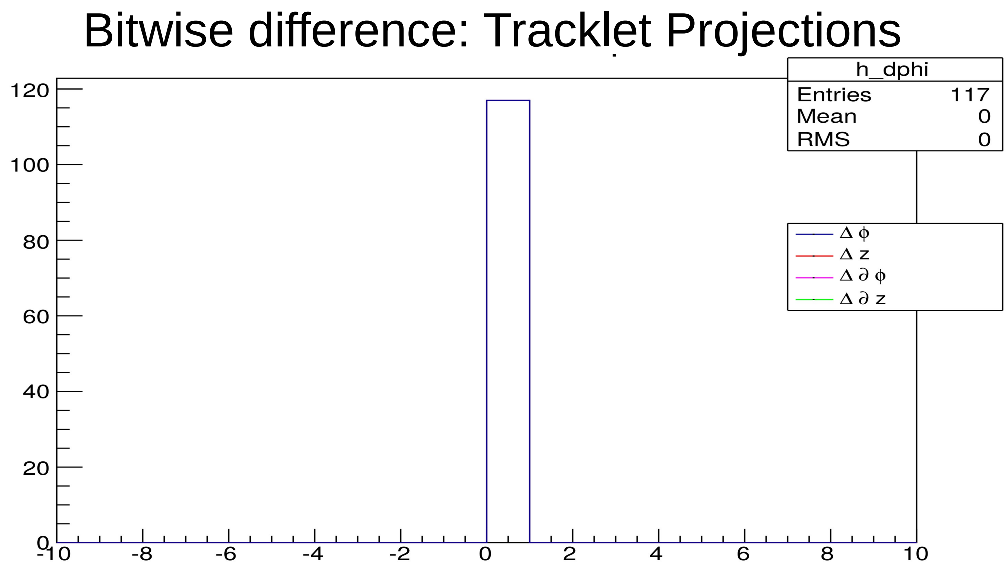
DCT Demonstrator



- Demonstrate DCT functionality for handling the trigger data. (Other DCT functionality is not part of this demonstrator)
  - Unpacking of data package, local to global translation, and data routing (to correct sector and time MUX slice)
- The frontend emulator and the sectors can be implemented on the same board: Need two boards for this demonstration
- Measure DCT latency

Firmware vs C++ Emulation



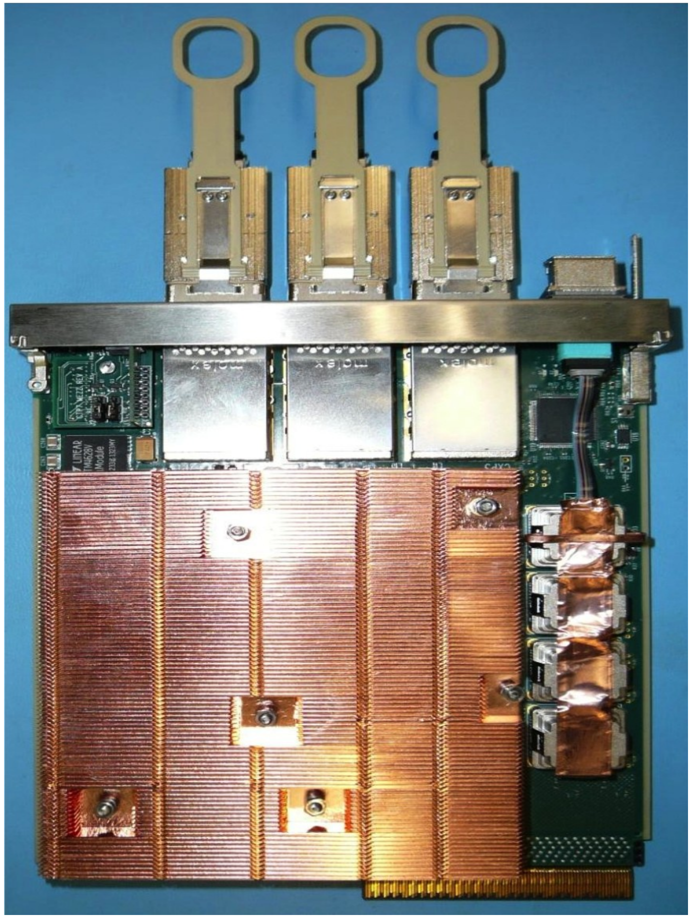


At the same time as we have optimized the firmware we have resolved some differences between the firmware and C++ emulation.

The goal is to have firmware and emulation software that produces the same results.

CTP7

- Calorimeter Trigger Processor Virtex7
  - Developed for the phase-1 trigger
- Based on the XC7VX690T FPGA
- The board provides 67 optical inputs and 48 outputs
  - Sufficient for the demonstrations we want to do
- Boards are currently in production
  - We expect that we can start working with this hardware for our demonstration in ~May 2015
    - We will use 4 boards (on loan) for our demonstration
  - We are currently not limited by lack of hardware – we can continue to develop the firmware and algorithms using the evaluation boards (vc709)



Developed by the U. Wisconsin

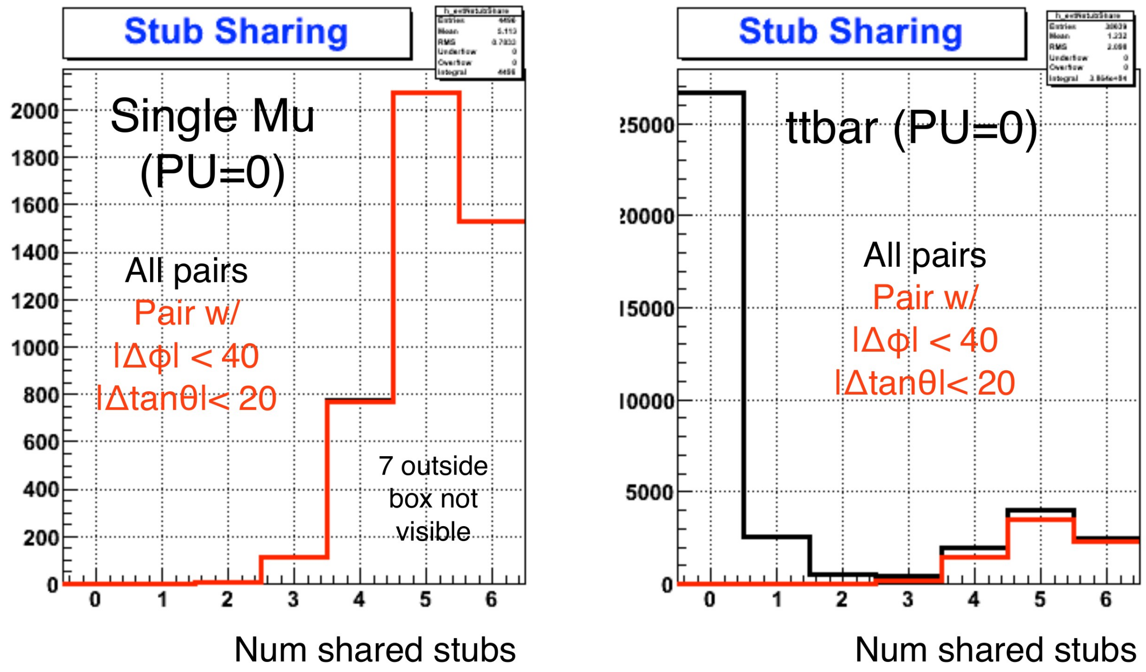
Forward Disk Implementation

- The use of the forward disks has now been implemented in the emulation software.
  - The relevant integer based calculations have been studied to give sufficient precision
  - Appropriate coordinates for these calculations have been defined
- Need to translate this into firmware
  - Calculation of tracklet parameters from pairs of stubs
  - Projection of stubs to disks
  - Implementation of modified trackfit for hits in disks

- One challenge for the disks is the use of PS hits in the outer disks. The challenge here is that the strips do not point to the IP (except for the center strips) and hence the phi position is a fcn of the radial position.

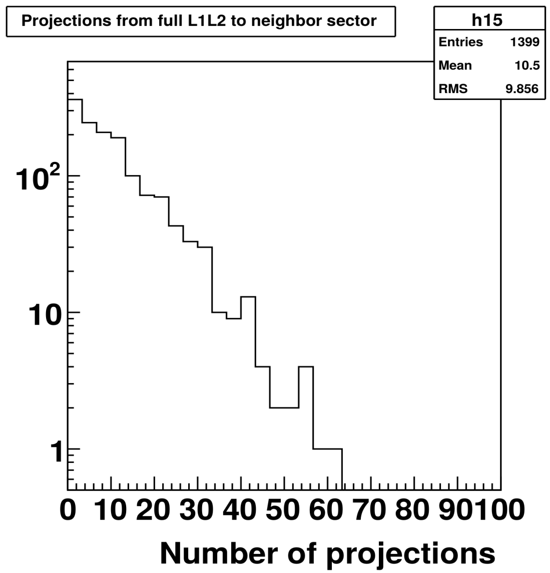
Duplicate Removal

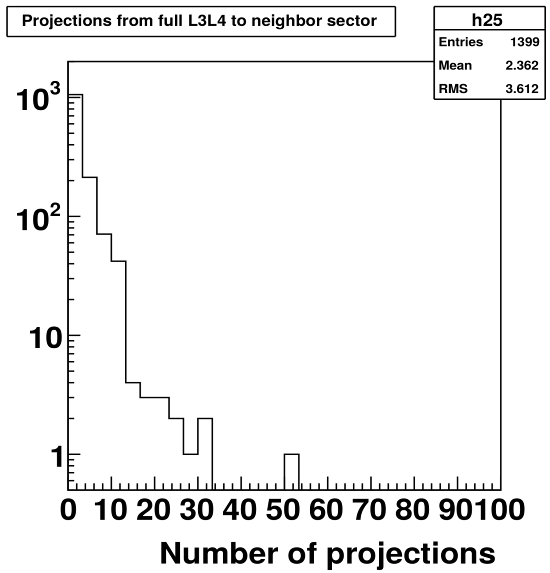
- Started a systematic study of how to identify and remove duplicate tracks (B. Winer)
- Starting by understanding the properties of the duplicates:

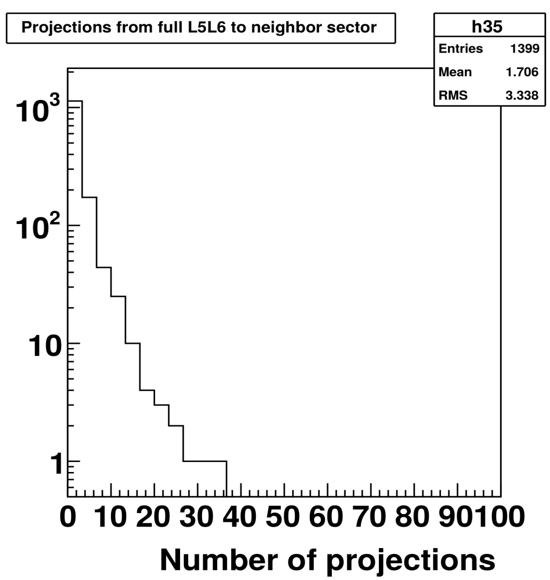


- The goal is to arrive at an algorithm that can be implemented in firmware

Projections to Neighboring Sector

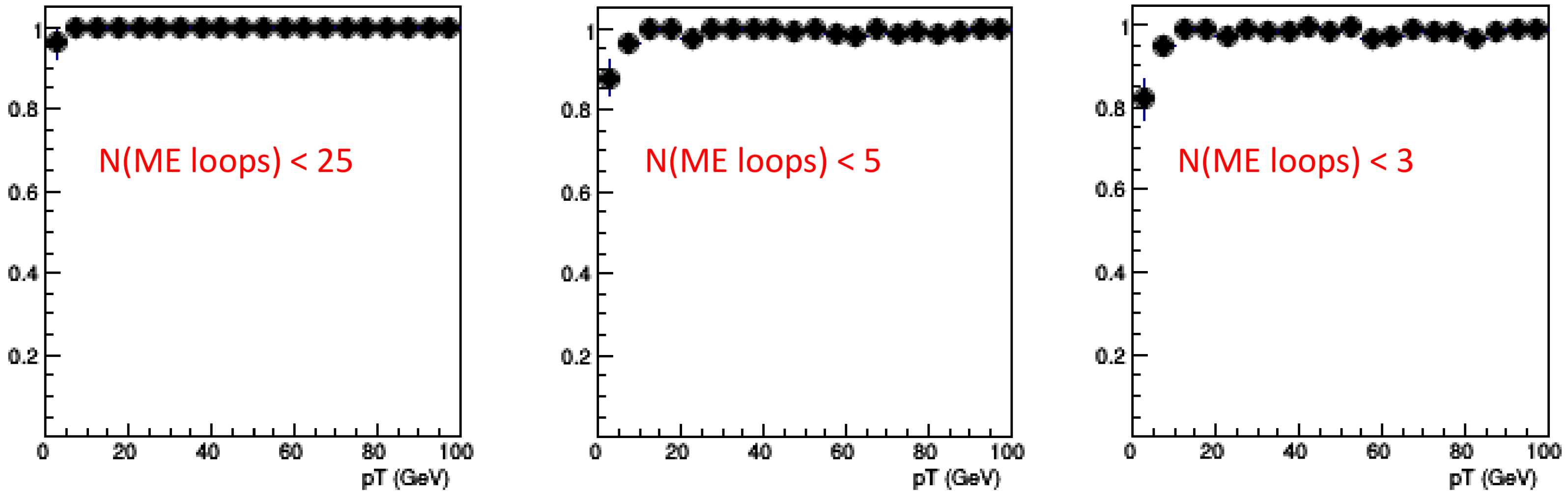






Number of projections from each of the three seed layers to neighboring sectors.

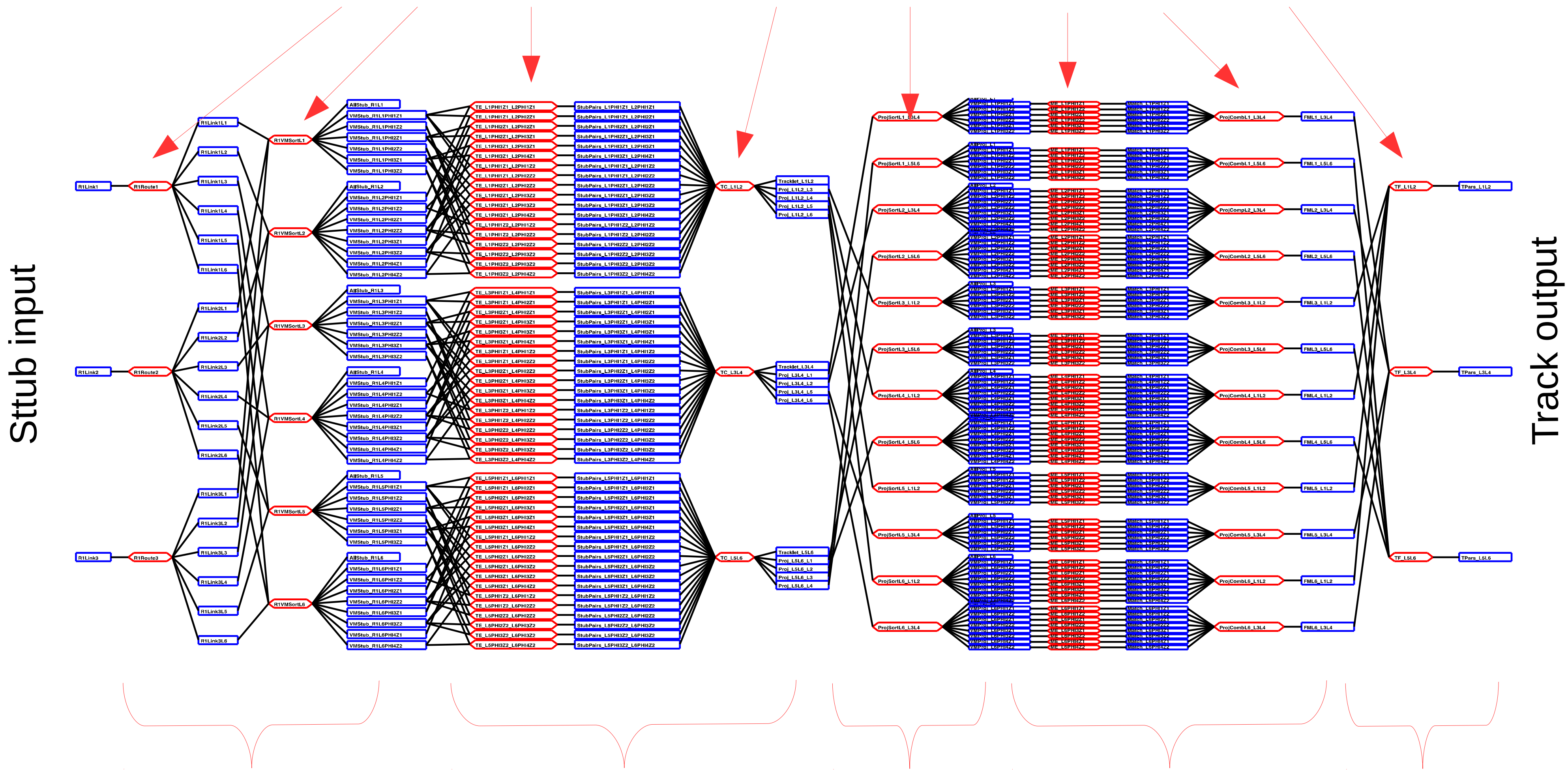
Truncating Match Engine



Effect of truncating the projection-stub matching

Overview of Project (¼ of Barrel Sector)

Eight processing steps (red) implements the algorithm



Stub organization

Forming tracklets

Organize tracklet projections

Match tracklet projections to stubs

Track fit

Seeding Redundancy