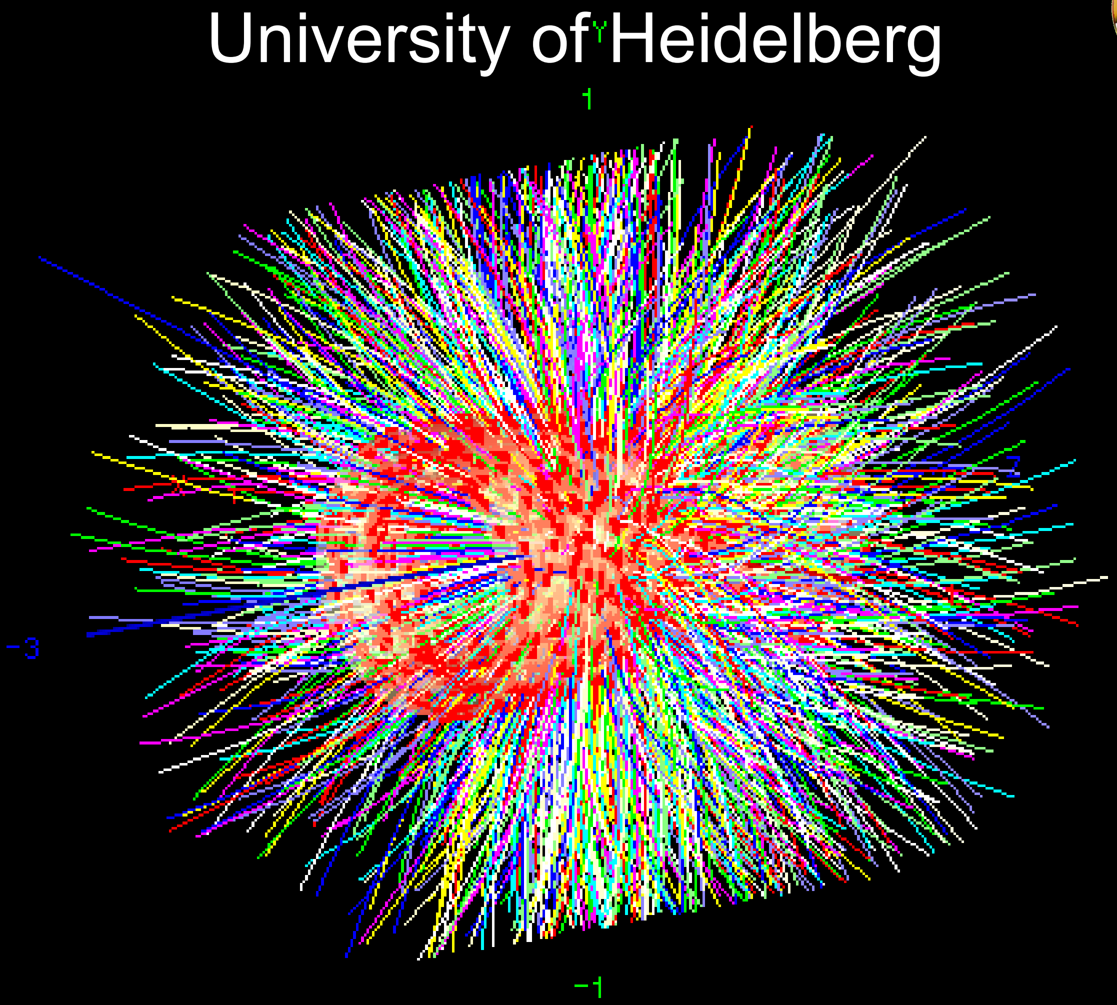
A First Level Track Trigger for Atlas at Super-LHC

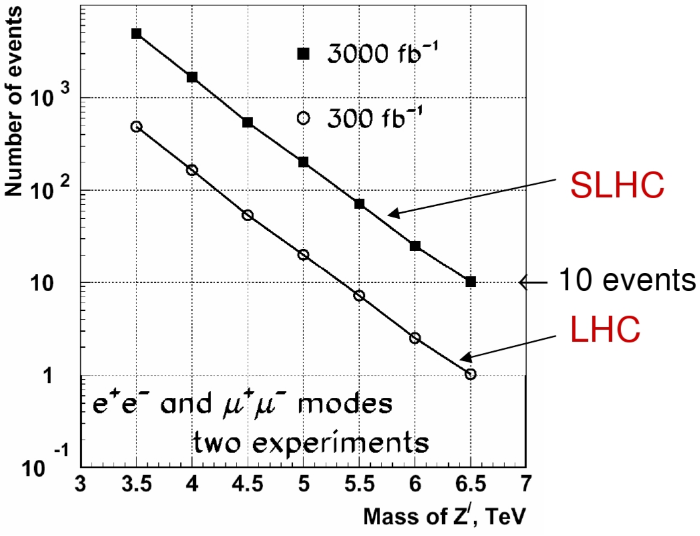
Physikalisches Institut
University of Heidelberg

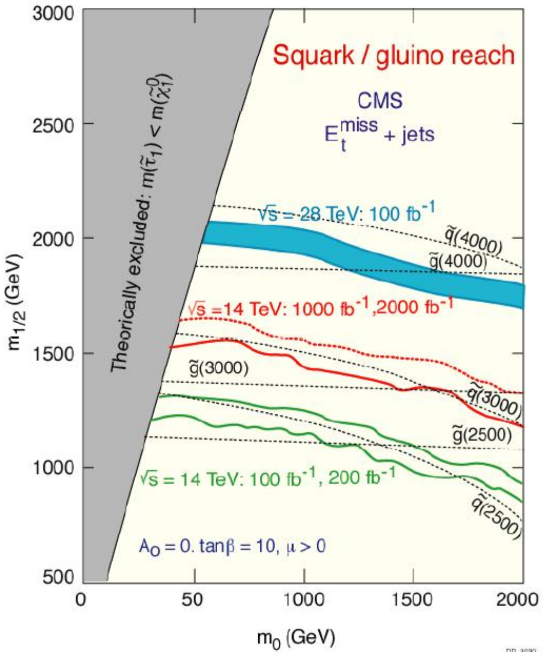


S. Schmitt, A. Schöning

Super-LHC: The LHC is not enough

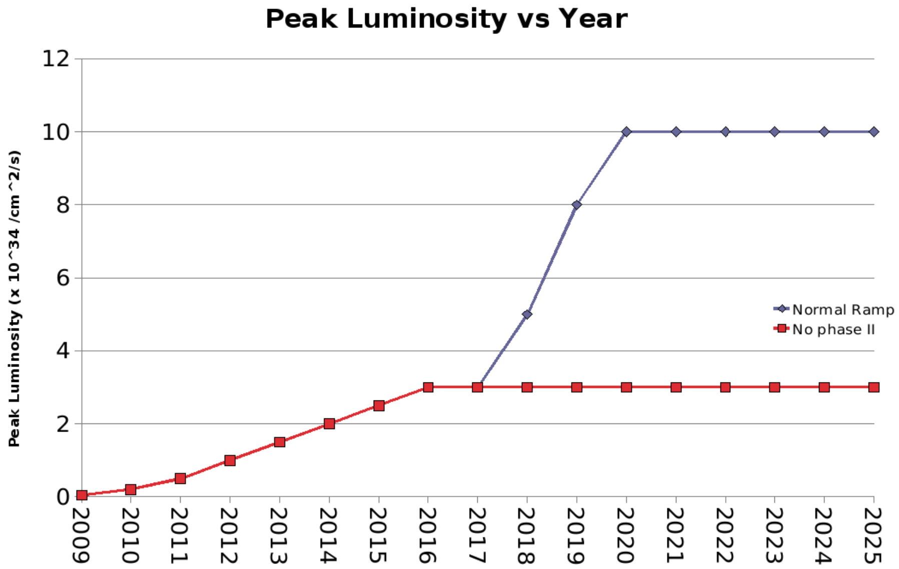
- **What LHC will discover is unknown, but:**
  - Properties of potentially new particles (e.g. Higgs couplings) need to be exactly measured (no ILC soon)
  - New heavy particles like e.g. Z' can only be seen with enough statistics
  - More statistics gives also better exclusion limits



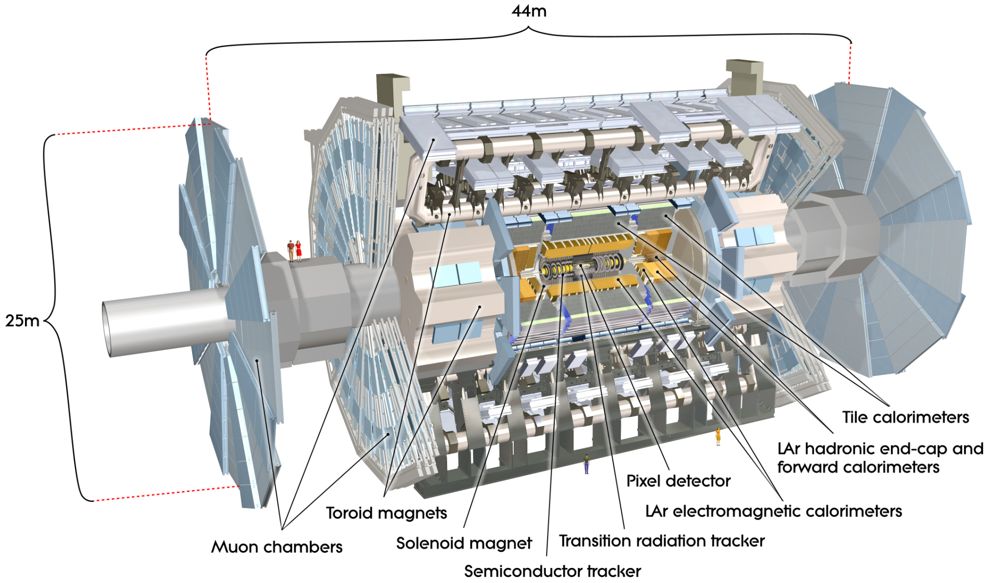


Super-LHC and Atlas

- **SLHC**
  - Plan: 10 times LHC luminosity, keep 14 TeV cms
  - Exploit full pp collider physics potential
  - Error halfing time becomes soon too large



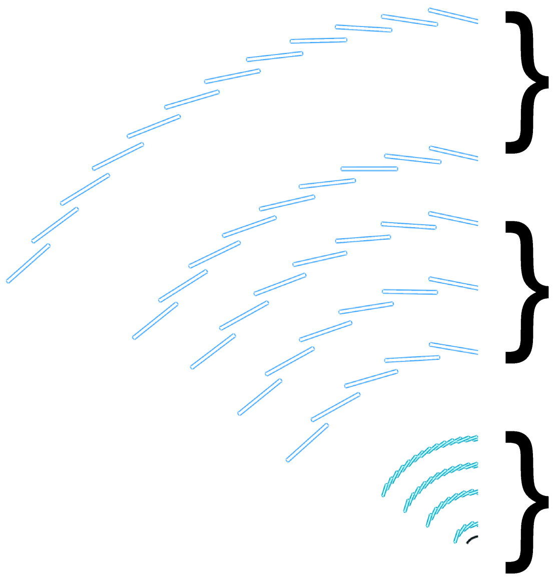
- **Atlas Upgrade**
  - Design life time of certain part reached after 10 years due to radiation damage
  - Replace inner detector completely by all silicon tracker
  - Improvements of muon and calorimeter system
  - Trigger system must be adapted for higher rates


44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Muon chambers
Solenoid magnet
Transition radiation tracker
Semiconductor tracker

New Inner Detector

- **Different designs under study**
- **No Transition Radiation Detector anymore**
- **Possible layout:**
  - Volume up to solenoid filled with silicon layers
  - Pixel: 50 µm x 250 µm
  - Short and long strips:

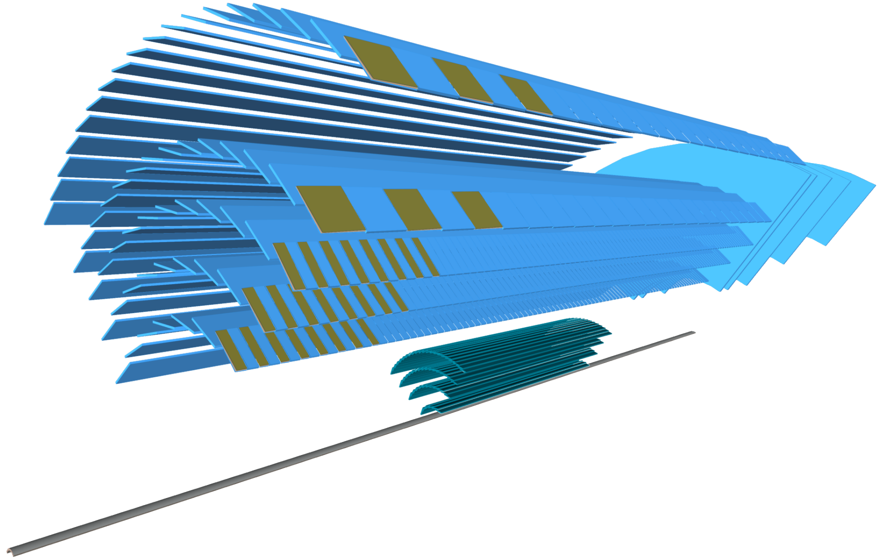
    75 µm x 2.4 cm (10 cm)



two long strip layers
75 µm x 10 cm

three short strip layers
75 µm x 2.4 cm

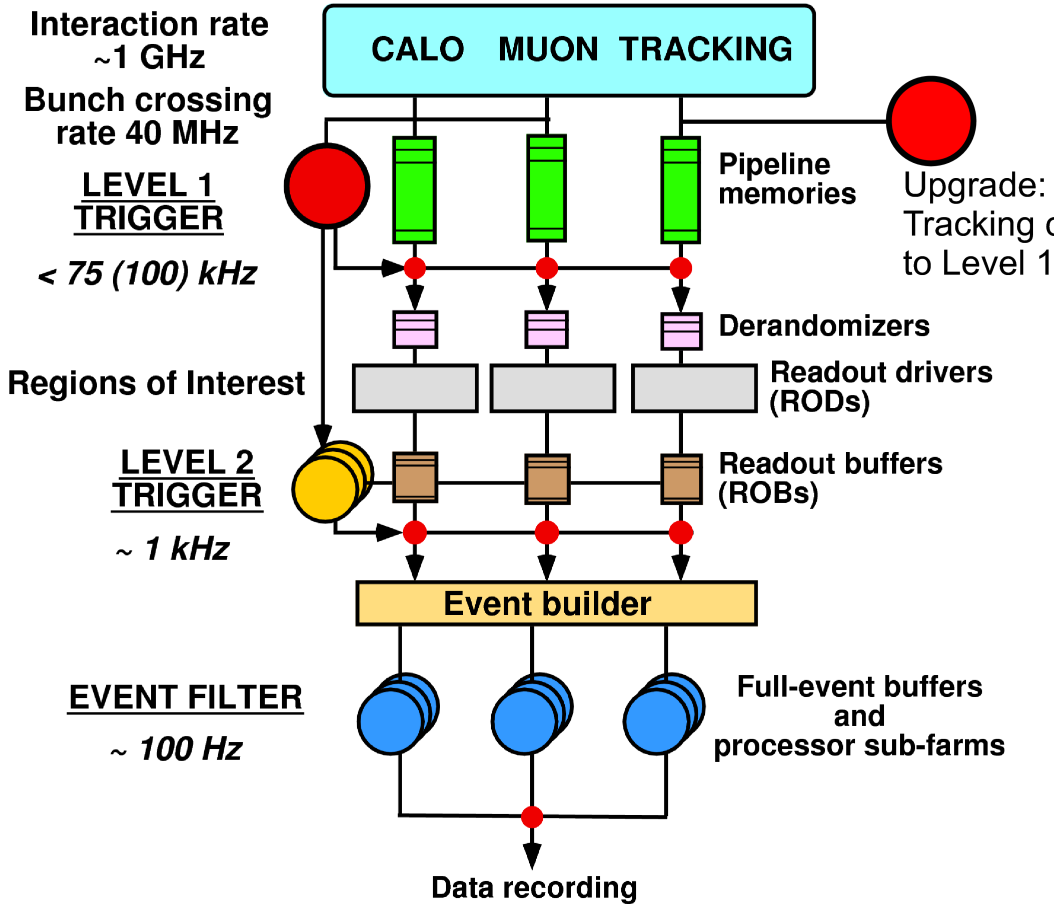
four pixel layers
50 µm x 250 µm



➔ Our Reference Design

Trigger Upgrade

- **Current Design**
  - L1 Trigger decision based on jets and leptons (calorimeter, muon system)
  - Lepton ID using central tracking information at level 2
- **Problem:**
  - Higher rate at SLHC -> higher L1 fake rate
  - Need better lepton ID at L1 or increase L1 bandwidth
- **Approaches to include Tracking**
  - Regional Readout + Regional Track Trigger (L1 - L2)
  - Early Regional L2 Track Trigger (FTK processor)
  - Fast Track Trigger (L1)



Interaction rate ~1 GHz

Bunch crossing rate 40 MHz

LEVEL 1 TRIGGER

< 75 (100) kHz

Regions of Interest

LEVEL 2 TRIGGER

~ 1 kHz

EVENT FILTER

~ 100 Hz

CALO MUON TRACKING

Pipeline memories

Upgrade: Tracking c

to Level 1

Derandomizers

Readout drivers (RODs)

Readout buffers (ROBs)

Event builder

Full-event buffers and processor sub-farms

Data recording

Requirements Fast Track Trigger

- **Should work on Level 1**
- **Latency below 1 µs**
- **Must be fully pipelined → no dead time**
- **Reconstruction of all tracks with pT > 10 GeV**

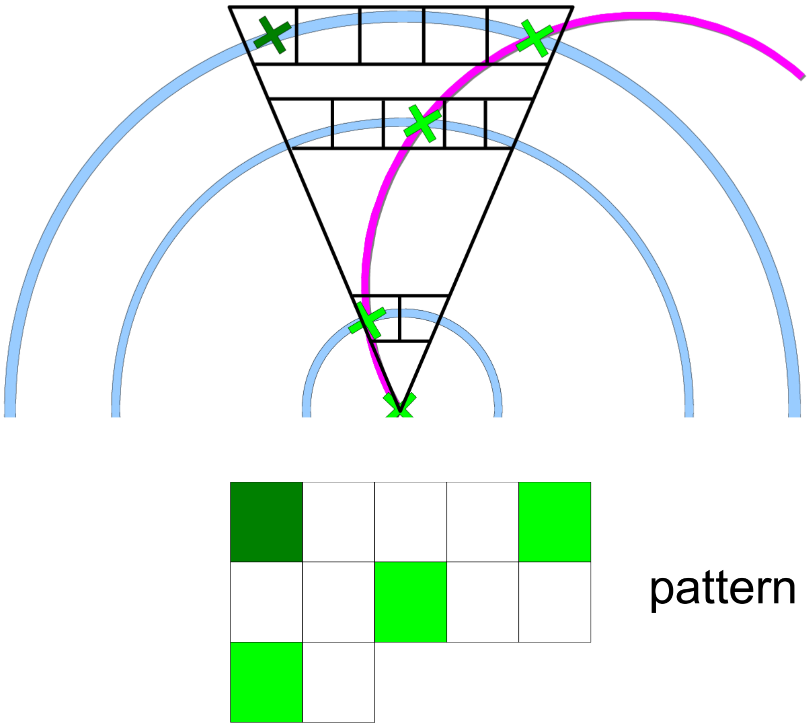
→ Possible solution: use lookups only

In contrast to FTK processor which combines lookup and fitting techniques.

Lookup Technique

- **Event described by a pattern of hits**
- **Lookup template ("valid" or "false") using the pattern as address identifier:**

$$addr = \sum 2^{\text{integer identifier of hit}}$$



Problem:

> 100 millions of identifiers

→ Not possible to address all possible outcomes by a single lookup

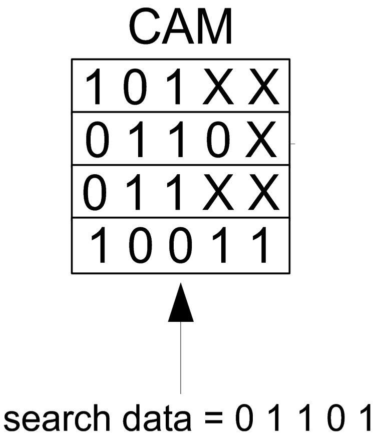
Definitions:
- Hit: signaled cell or channel
- Pattern: pattern of signals in all channels
- Combination: combination of hits between layers
- Template: combination belonging to a valid track

Smart Lookup

- **Problem: Too many valid patterns**
- **Solution: Ignore not important hit combinations**
- **Possible implementation: Ternary CAM (“Don't Care bits”)**
- **Content Addressable Memory (CAM)**
  - Offers storage and in-built compare
  - Input patterns are compared simultaneously against all stored templates

e.g. internet routers

X = don't care



CAM

1 0 1 X X
0 1 1 0 X
0 1 1 X X
1 0 0 1 1

search data = 0 1 1 0 1

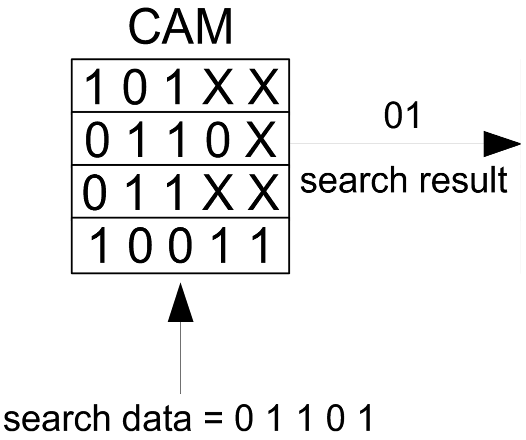
- Available as
  - discrete CAM (e.g. Network Search Engine) 64k x 586 bit
  - embedded FPGA 10k x 64 bit

Smart Lookup

- **Problem: Too many valid patterns**
- **Solution: Ignore not important hit combinations**
- **Possible implementation: Ternary CAM (“Don't Care bits”)**
- **Content Addressable Memory (CAM)**
  - Offers storage and in-built compare
  - Input patterns are compared simultaneously against all stored templates

e.g. internet routers

X = don't care



CAM

1 0 1 X X
0 1 1 0 X
0 1 1 X X
1 0 0 1 1

01

search result

search data = 0 1 1 0 1

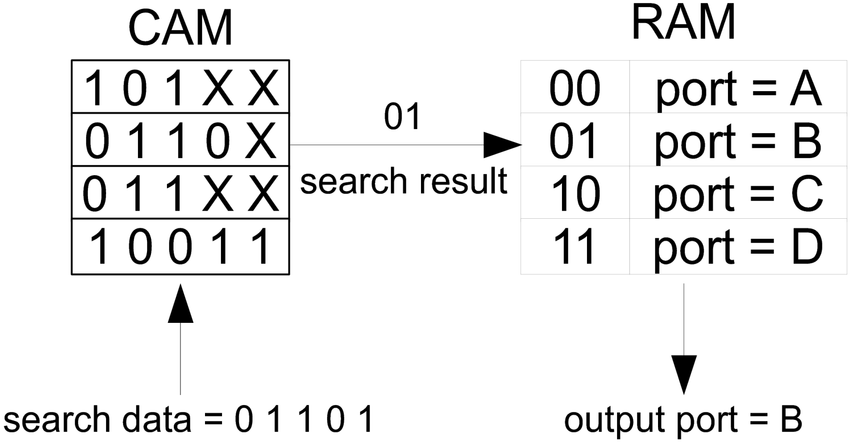
  - Available as
    - discrete CAM (e.g. Network Search Engine) 64k x 586 bit
    - embedded FPGA 10k x 64 bit

Smart Lookup

- **Problem: Too many valid patterns**
- **Solution: Ignore not important hit combinations**
- **Possible implementation: Ternary CAM (“Don't Care bits”)**
- **Content Addressable Memory (CAM)**
  - Offers storage and in-built compare
  - Input patterns are compared simultaneously against all stored templates

e.g. internet routers

X = don't care



CAM
1 0 1 X X
0 1 1 0 X
0 1 1 X X
1 0 0 1 1

01

search result

RAM

00	port = A
01	port = B
10	port = C
11	port = D

search data = 0 1 1 0 1

output port = B

- Available as
  - discrete CAM (e.g. Network Search Engine) 64k x 586 bit
  - embedded FPGA 10k x 64 bit

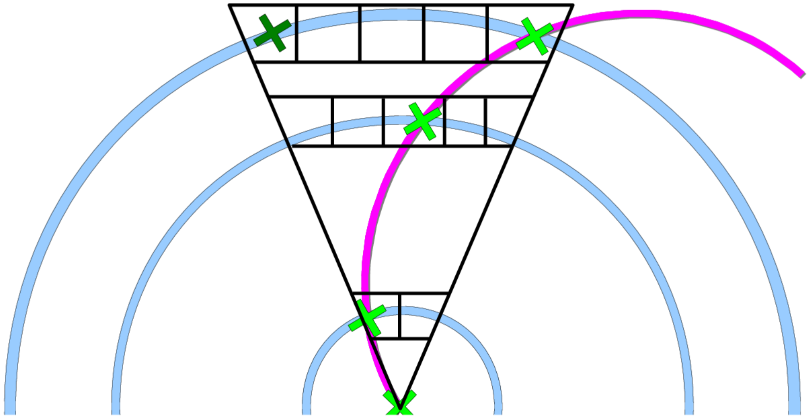
Hardware Implementation

currently available circuits:

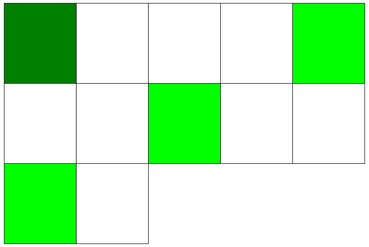
	Pros	Cons
discrete CAM Netlogic	• **lots of memory and large input bandwidths**	• **slow multiple matches** • **price**
FPGA Alterate	• **embedded CAM** • **flexible, integration**	• **smaller memory** • **price**
ASIC	• **associated memory chip, specialized and optimized design** • **e.g. AM chip (FTK)**	• **smaller memory** • **speed**

Track Trigger based on CAMs

- **Procedure**
  - Compare patterns in strips or pixels with precomputed templates
  - Templates are stored in Ternary-CAMs providing storage and compare functionality
  - Every part of the detector is special, no symmetry exploited
- **Main questions**
  - How many templates needed depending on detector geometry?
  - Fake rate?
  - Hardware implementation?



pattern



template

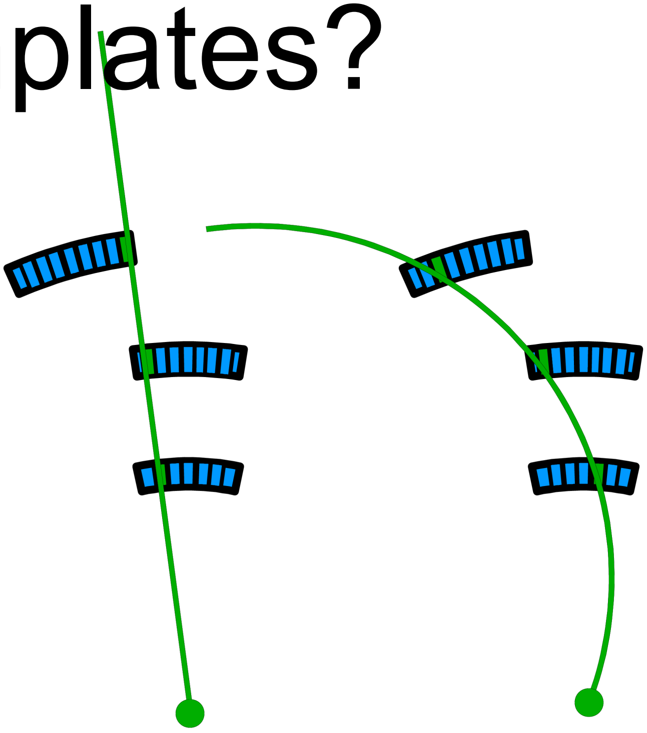
X	X	X	X	1
X	X	1	X	X
1	X			

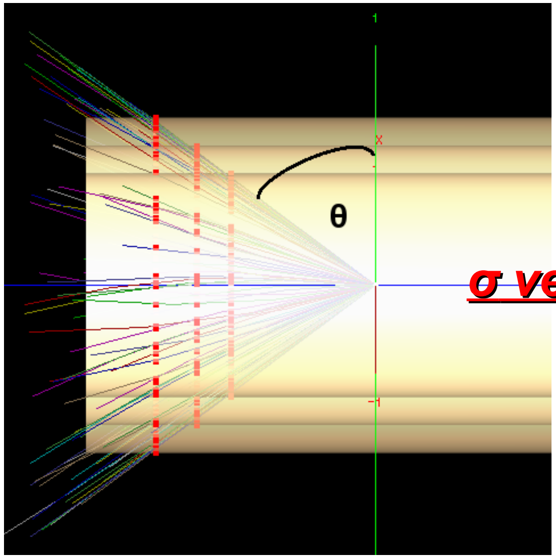
X = don't care

How many templates?

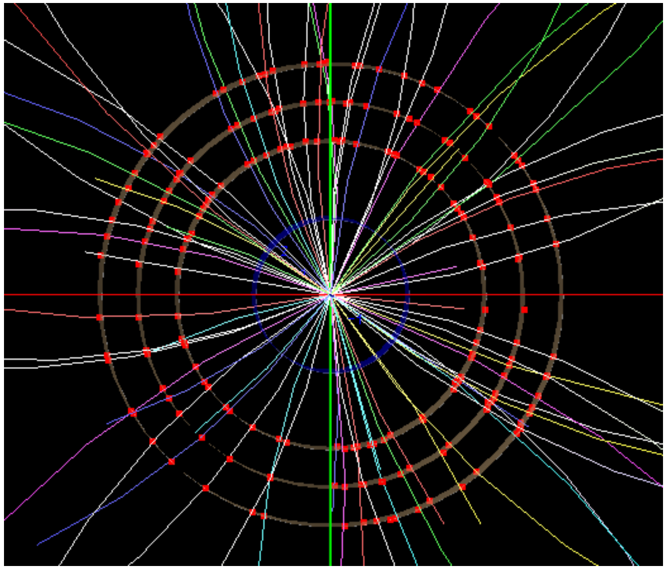
- **Toy MC and reconstruction tailored for templates studies**
- **Factorize transverse x-y and longitudinal r-z plane**
- **Total number of templates:**

$$N_{tot} = N_{trans} * N_{long}$$





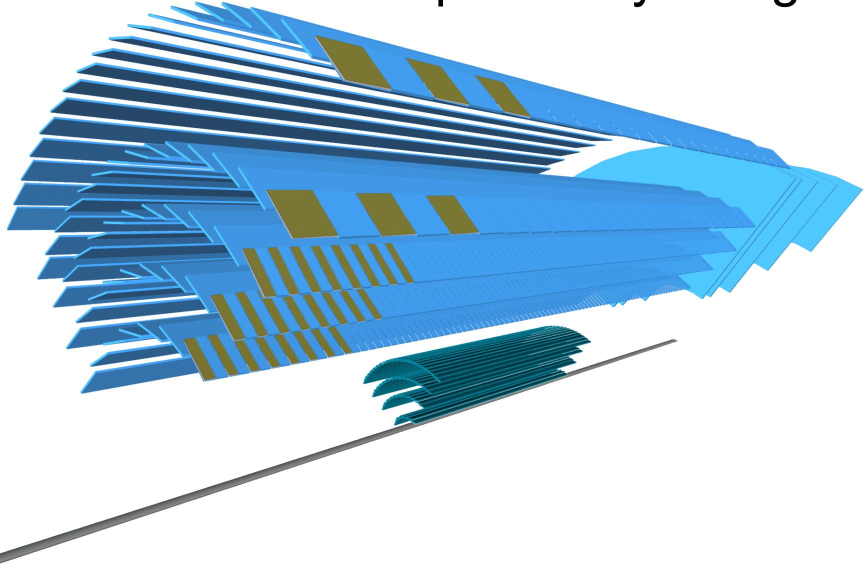
σ vertex = 3.5 cm



Studied Layouts

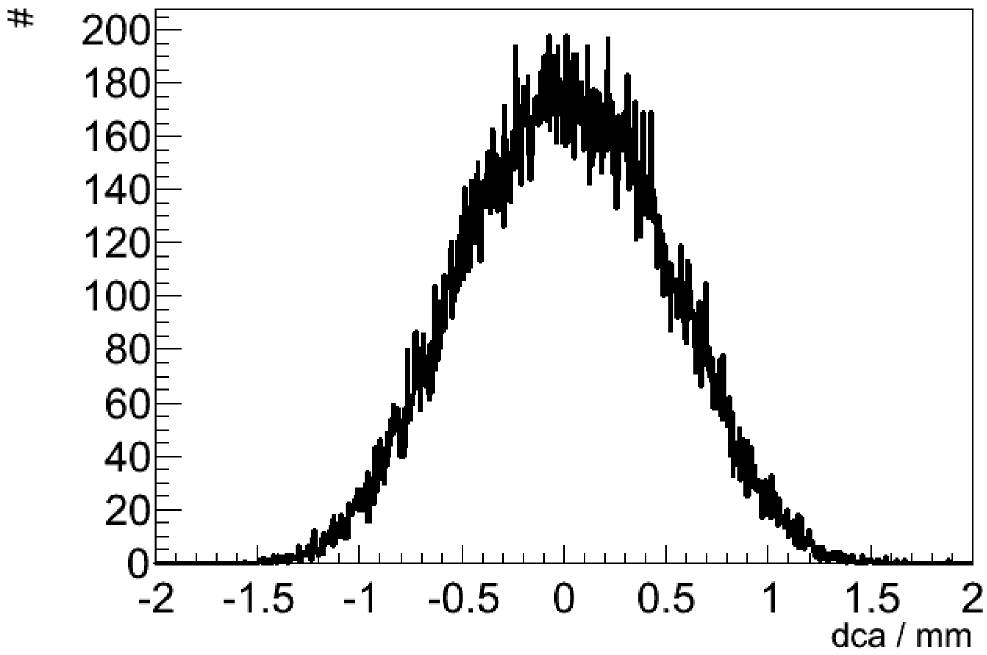
- **Reference Design 3 - 5 layers**
- **Either hits in all layers or one layer allowed to be inefficient**
- **Consider barrel only**
- **Trade-off:**
  - Fake rate vs. #templates
  - Track Trigger only fake rate < 100 tracks considered to be useful (for matches with calo and muon system)
  - 100 billion templates as limit

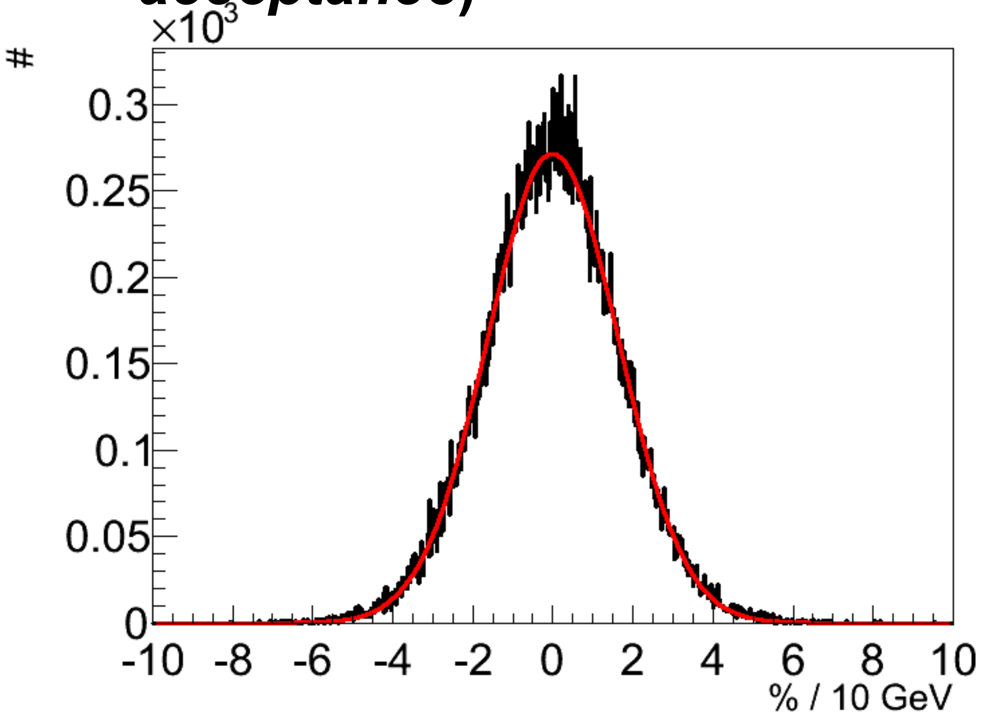
- **Strips:**
  - 3-5 strip (double) layers (75mu)
  - Short and long (2.4 cm, 10 cm)
  - All short
- **Pixel (additional)**
  - Outer pixel layers (50 mu)
  - Pixel merged in z (coarsening)
  - No pixel only design

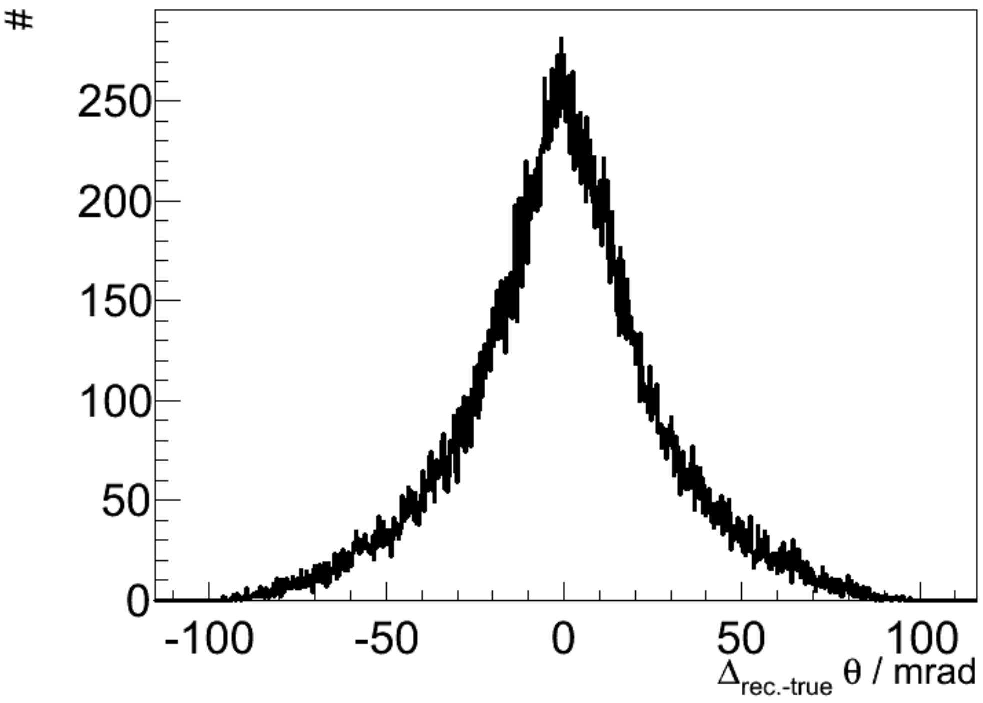


3 short layers:

- **2 % / 10 GeV p_T resolution, (sharp threshold)**
- **30 mrad polar angle resolution (good directional information)**
- **0.5 mm distance of closest approach resolution (vertex *acceptance*)**







Results

layout	number of templates (min. p_T10 GeV)
3 short (3/3)	~ 3 billion
3 short 2 long (5/5)	~ 100 billion
1 pixel* 3 short (4/4)	~ 20 billion
2 pixel* 3 short (5/5)	~ 80 billion
pixel*: coarsened 250 µm -> 2.4 cm	

- **Number of patterns highly dependent on layout**
- **3 short 2 long setup suffers from last layer at 100 cm (large envelope and multiple scatter)**

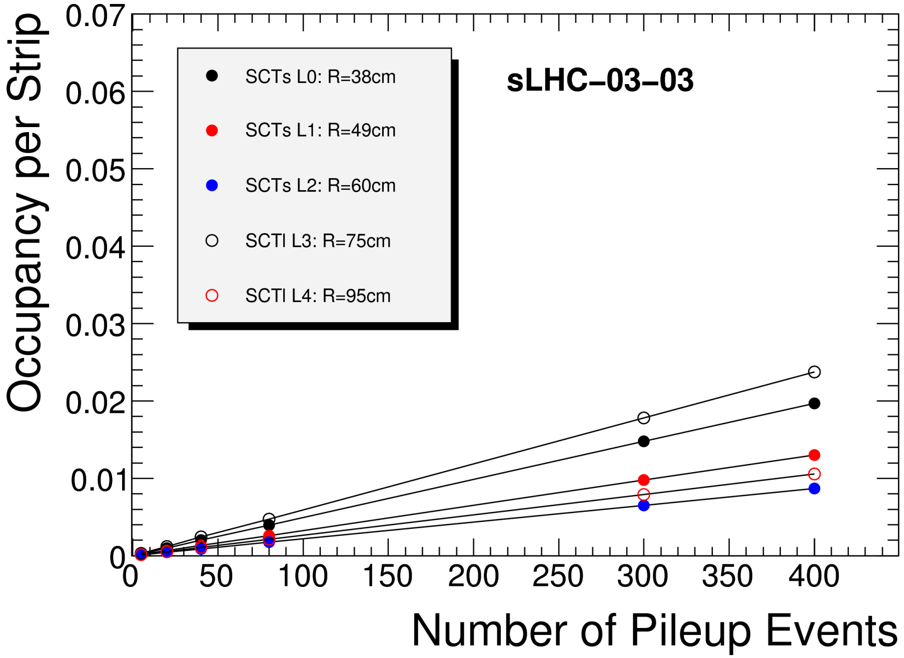
Results

layout	number of templates (min. p_T10 GeV)	fake rate (400 pile-up)
3 short (3/3)	~ 3 billion	~5000
3 short 2 long (5/5)	~ 100 billion	~100
1 pixel* 3 short (4/4)	~ 20 billion	~2000
2 pixel* 3 short (5/5)	~ 80 billion	~1000

pixel*: coarsened 250 µm -> 2.4 cm

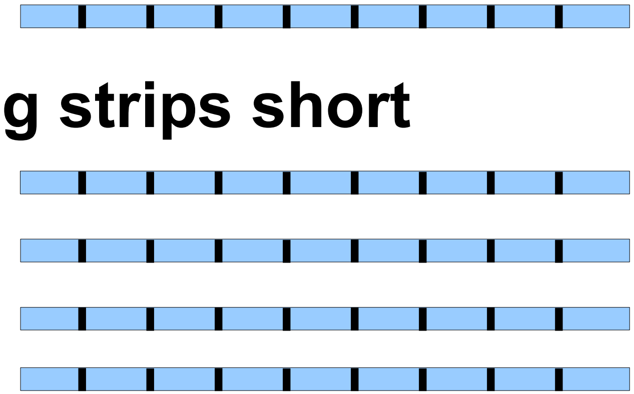
to be compared to
one signal track p_T>10 GeV

- **Number of patterns highly dependent on layout**
- **3 short 2 long setup suffers from last layer at 100 cm (large envelope and multiple scatter)**
- **Fake rate is governed by occupancy**

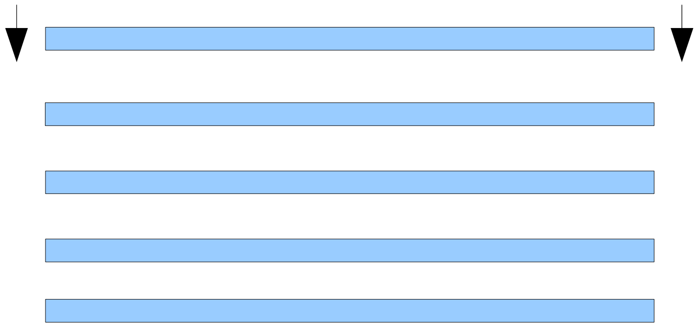


How to reduce the fake rate?

3 short + 2 long strip design:



- **Reducing occupancy by making the long strips short**
  - From 100 billion to 230 billion templates
  - From 100 to 5 fake tracks

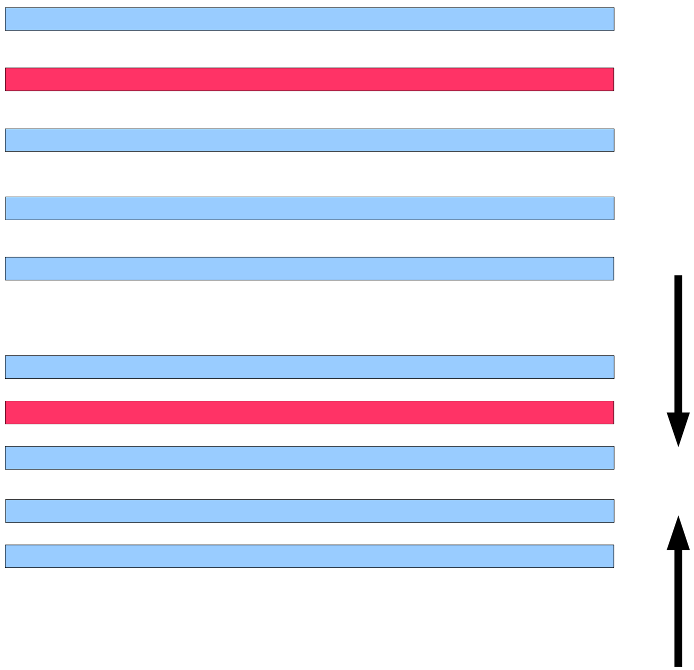


- **Keeping all strips short but moving the last layer from 100 cm to 86 cm**
  - From 230 billion to 100 billion templates
  - From 5 fake tracks to 20
    (higher occupancy in last layer)

  Best trade-off between fake rate and #templates so far

Robust design must allow for one inefficient layer

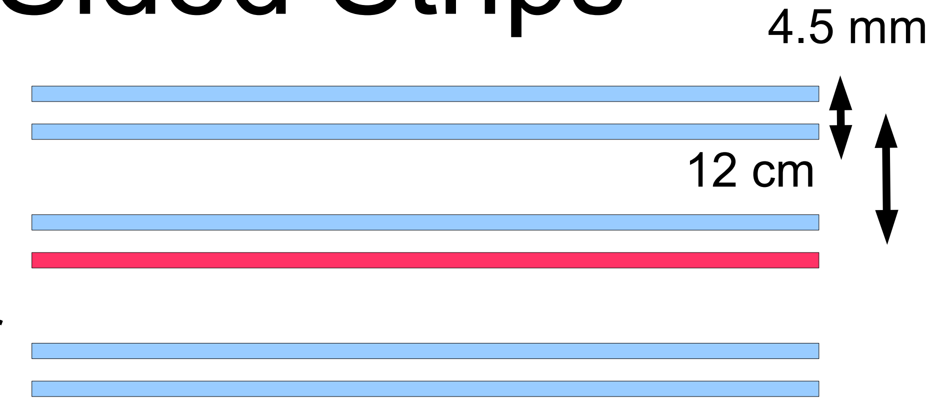
- **4/5 short strips 12 cm spacing (last layer 86 cm)**
  - #templates: 160 billion
  - Fake rate: 1000
- **4/5 short strips 8 cm spacing (last layer 86 cm)**
  - #templates: 60 billion
  - Fake rate: 1000

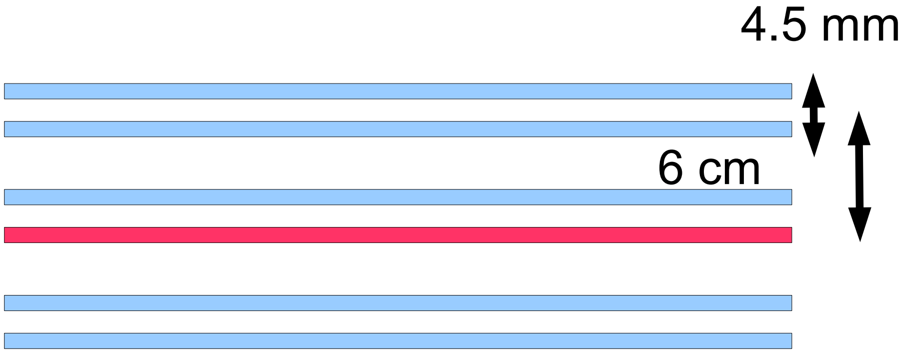


- **Reducing spacing reduces #templates**
- **4 / 5 not feasible, too many fakes**

Exploit Double Sided Strips

- **Strip layers are double sided with stereo angle to provide z resolution**
  - remove stereo angle -> double layer for triggering
- **3 shorts at default position but each two sided**
- **This leads to a 5/6 sensors layout, 3 double sided strip layers 12 cm apart**
  - #templates: 100 billion
  - Fake rate: 50
- **Manageable amount of patterns and fake rate**





- #templates and fake rate can be further reduced by decreasing layer distances: -> 6cm: 30 billion, 10 fake tracks

Matching 100 billion templates in < 1 µs

- **First step: coarse template search**
  - #templates ~ 1/pitch^2
  - Combining 32 strips gives a reduction of factor 1024:

    100e9 / 1024 ≈ 100 million

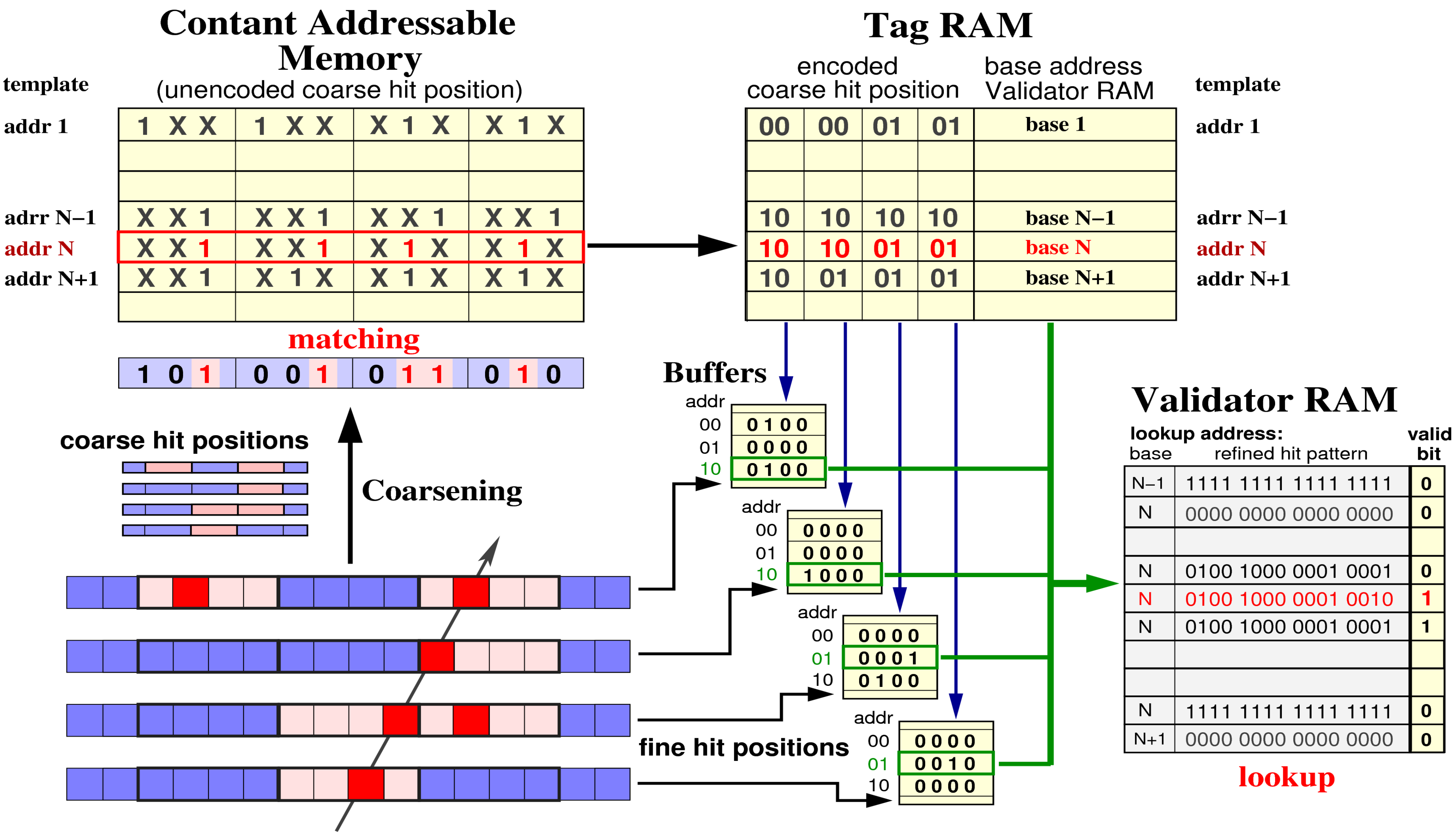
- **100 million templates can be distributed to 10000 parallel Track Finding Units (TFU) each storing 10000 templates**

- **Second step: refinement relying on lookup only**

Detailed Hardware Implementation

Coarse Template Finding

Refinement Step



Contant Addressable Memory

(unencoded coarse hit position)

template				
addr 1	1 X X	1 X X	X 1 X	X 1 X
adrr N–1	X X 1	X X 1	X X 1	X X 1
addr N	X X 1	X X 1	X 1 X	X 1 X
addr N+1	X X 1	X 1 X	X 1 X	X 1 X

matching

1 0 1	0 0 1	0 1 1	0 1 0

Tag RAM

encoded coarse hit position				base address Validator RAM	template
00	00	01	01	base 1	addr 1
10	10	10	10	base N–1	adrr N–1
10	10	01	01	base N	addr N
10	01	01	01	base N+1	addr N+1

Buffers

addr	
00	0 1 0 0
01	0 0 0 0
10	0 1 0 0

addr	
00	0 0 0 0
01	0 0 0 0
10	1 0 0 0

addr	
00	0 0 0 0
01	0 0 0 1
10	0 1 0 0

addr	
00	0 0 0 0
01	0 0 1 0
10	0 0 0 0

Validator RAM

lookup address: base	refined hit pattern	valid bit
N–1	1111 1111 1111 1111	0
N	0000 0000 0000 0000	0
N	0100 1000 0001 0001	0
N	0100 1000 0001 0010	1
N	0100 1000 0001 0001	1
N	1111 1111 1111 1111	0
N+1	0000 0000 0000 0000	0

lookup

H1 FTT: A. Baird et al., IEEE Trans. Nucl. Sci. 48 (2001) 1276

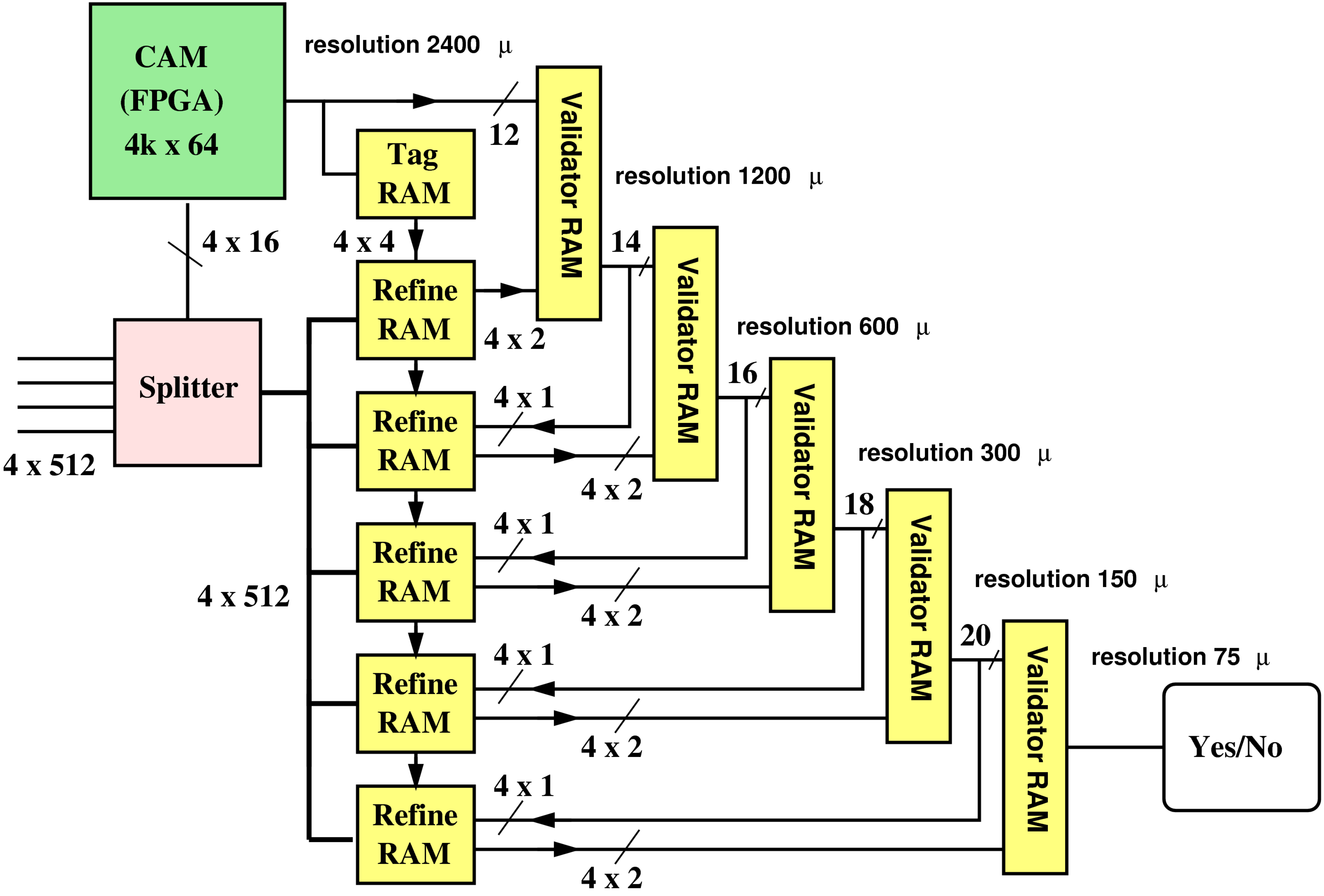
Conclusion

- **SLHC conditions very challenging for Level 1 Trigger**
  - Early Track Triggers are considered to improve lepton identification capabilities

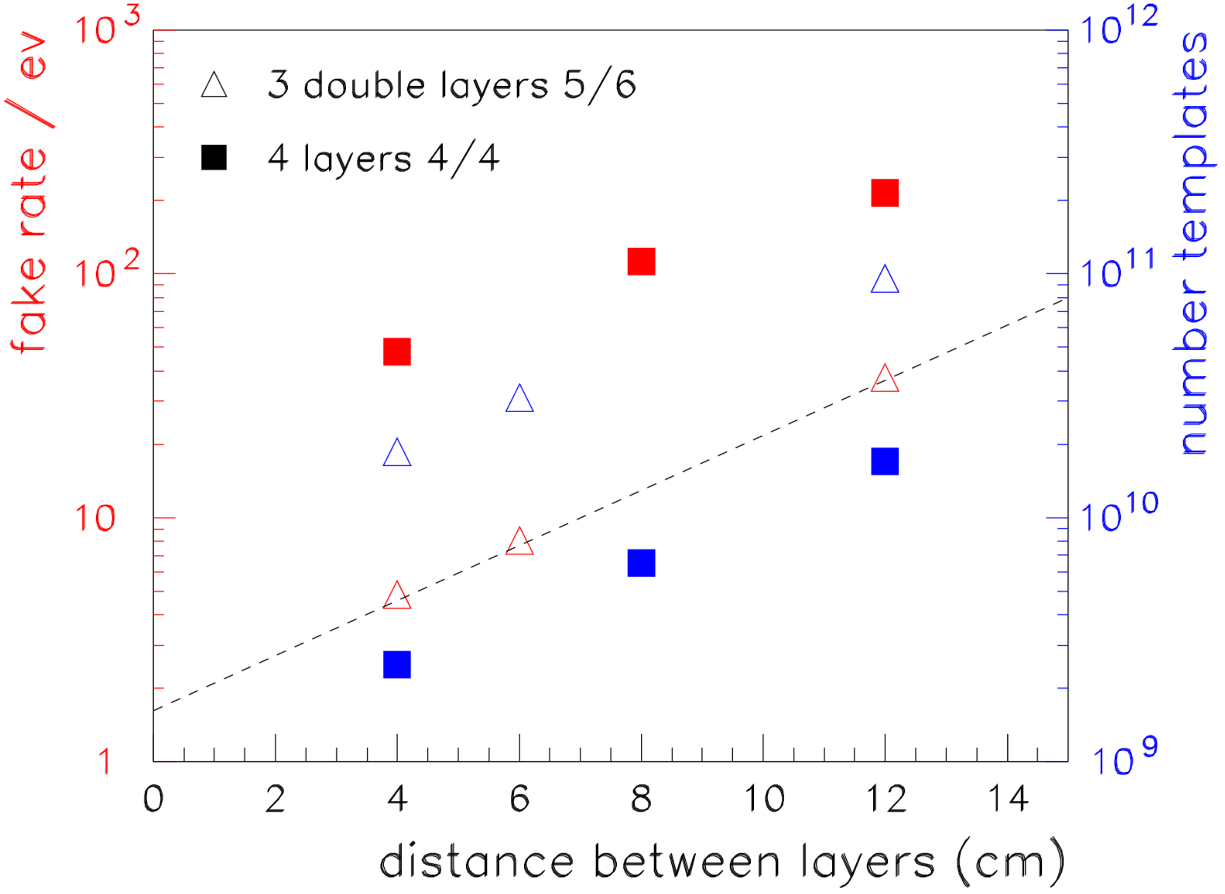
- **Minimum of 5 layers are needed to suppress fake rate**
  - Pixel layers not useful in L1 trigger
  - Double sided (parallel) strip layers reduce fake rate
- **Track Trigger Design using 10^{11} templates based on CAMs feasible in combination with fast lookups (refinement)**
  - progress in circuitry will further expand trigger capabilities

- **More detailed designs to be studies**
- **Ignored the severe bandwidth limitations so far (unsolved)**
- **Physics studies needed to specify maximum allowed fake rate**

Iterative Refinement

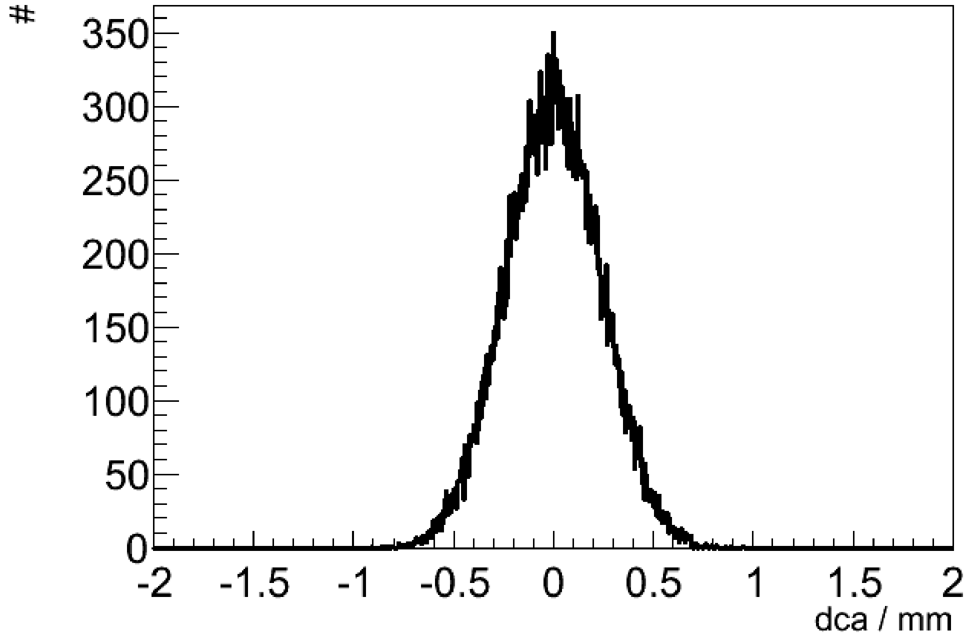


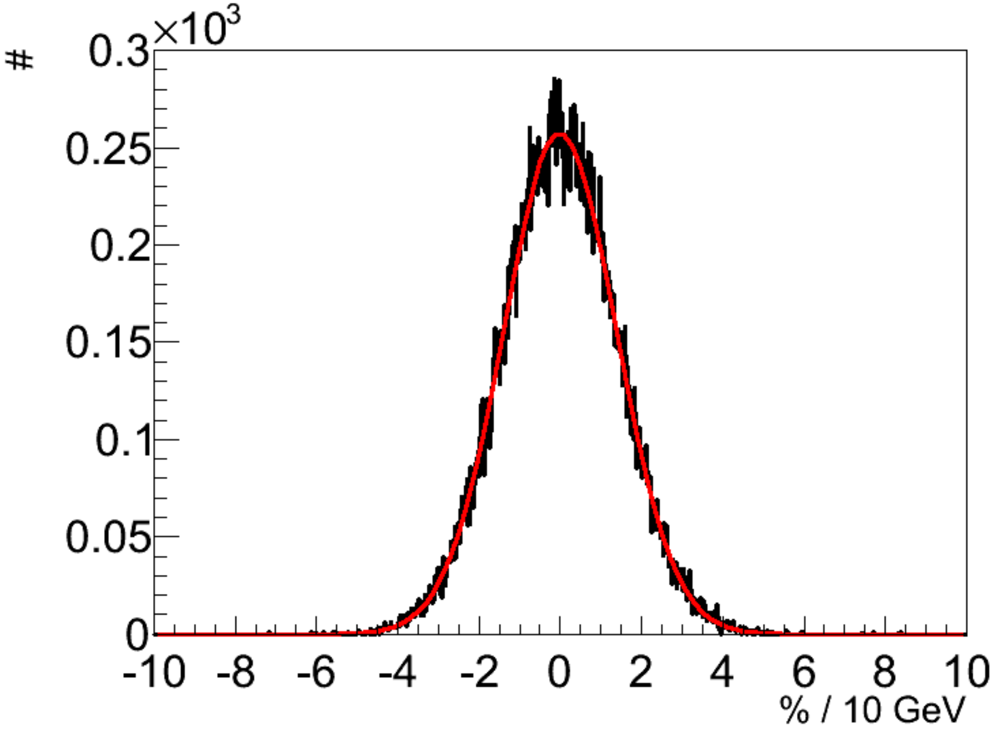
Fake rate and #templates vs. spacing

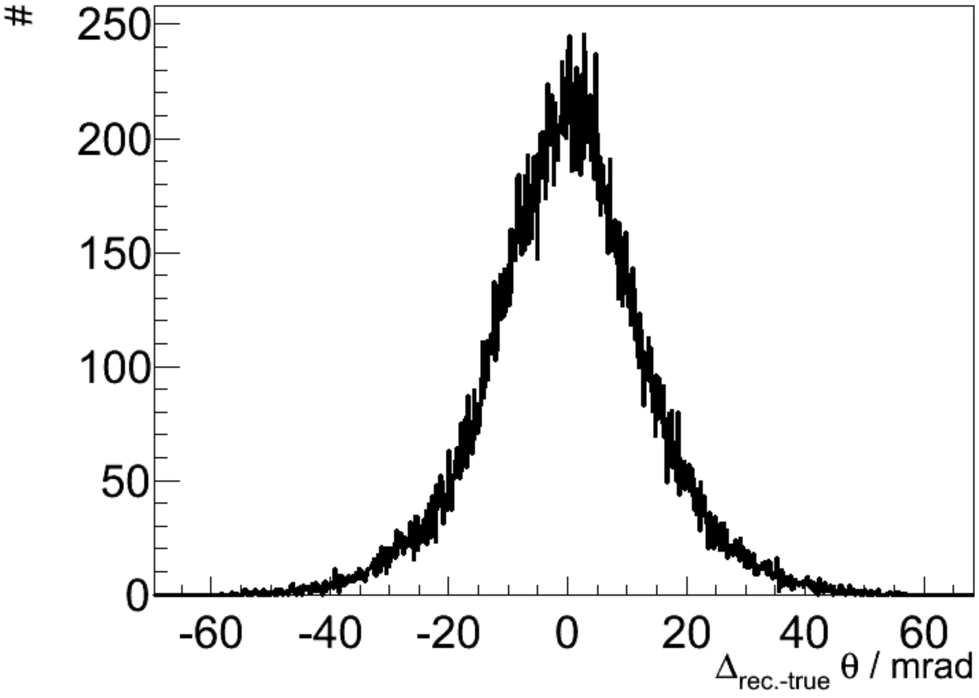


5 short layers, one inefficient (4 layers)

- **1.4 % / 10 GeV p_T resolution**
- **13 mrad polar angle resolution**
- **0.25 mm distance of closest approach resolution**

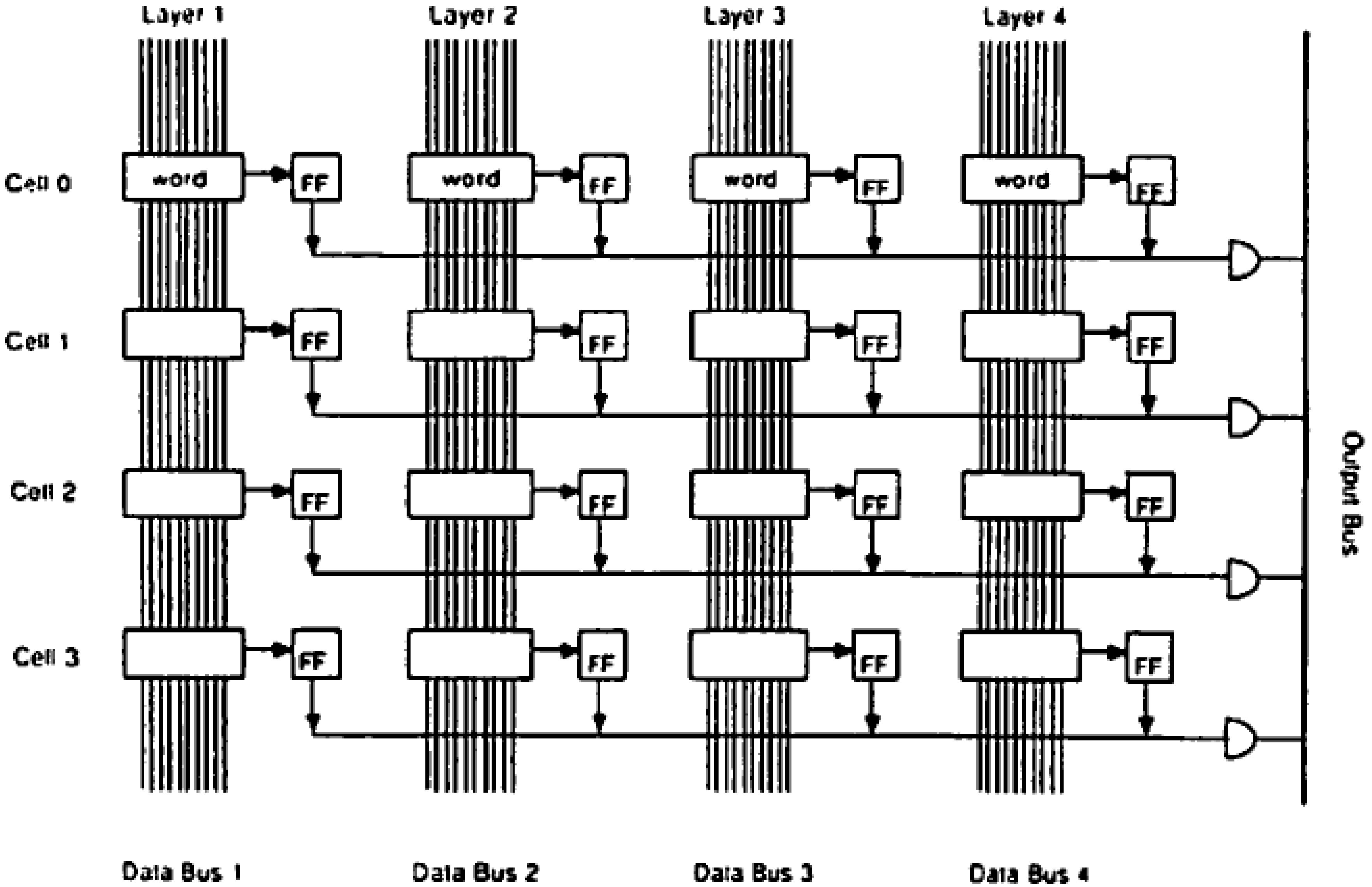




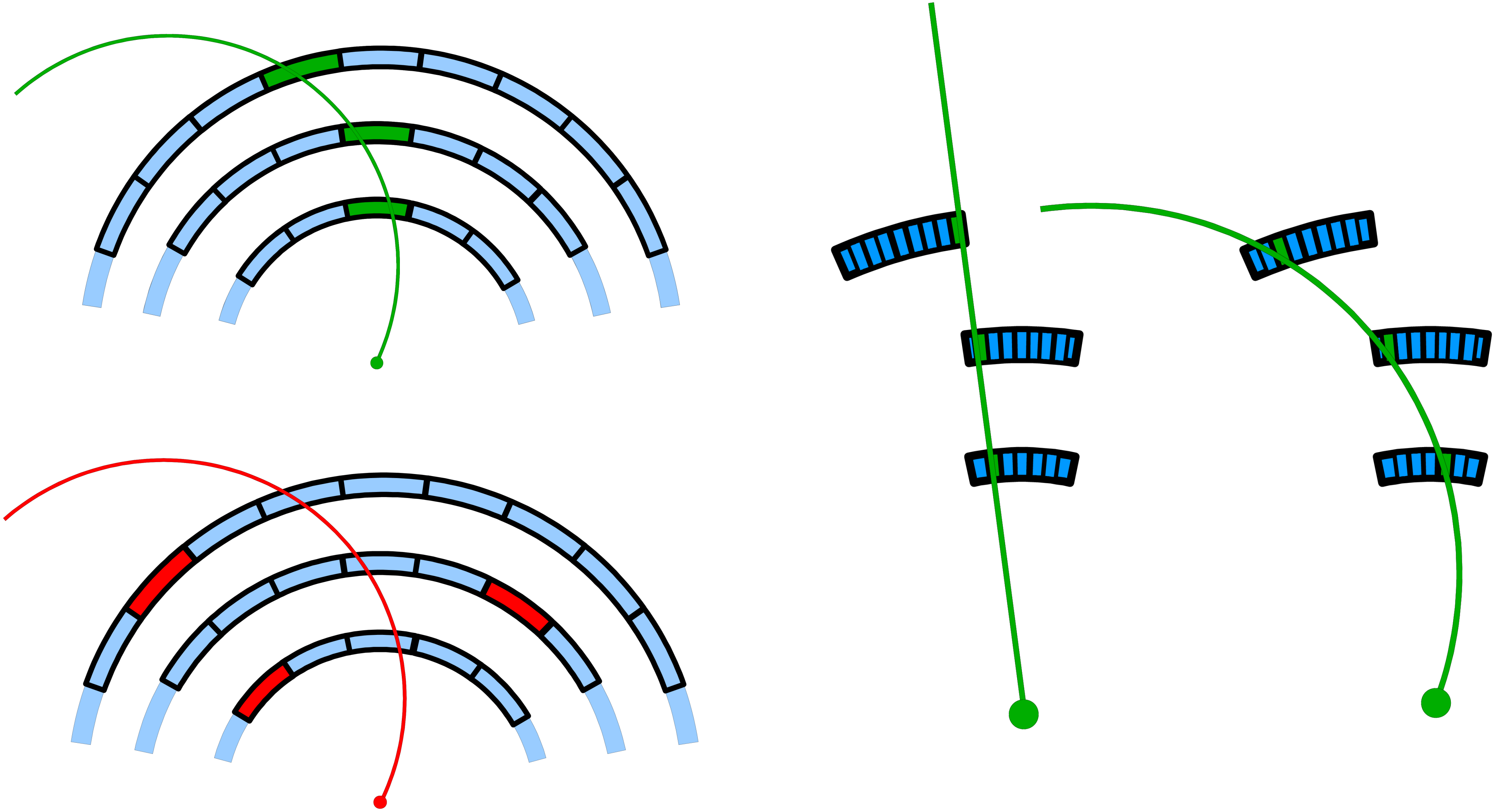


FTK AM chip

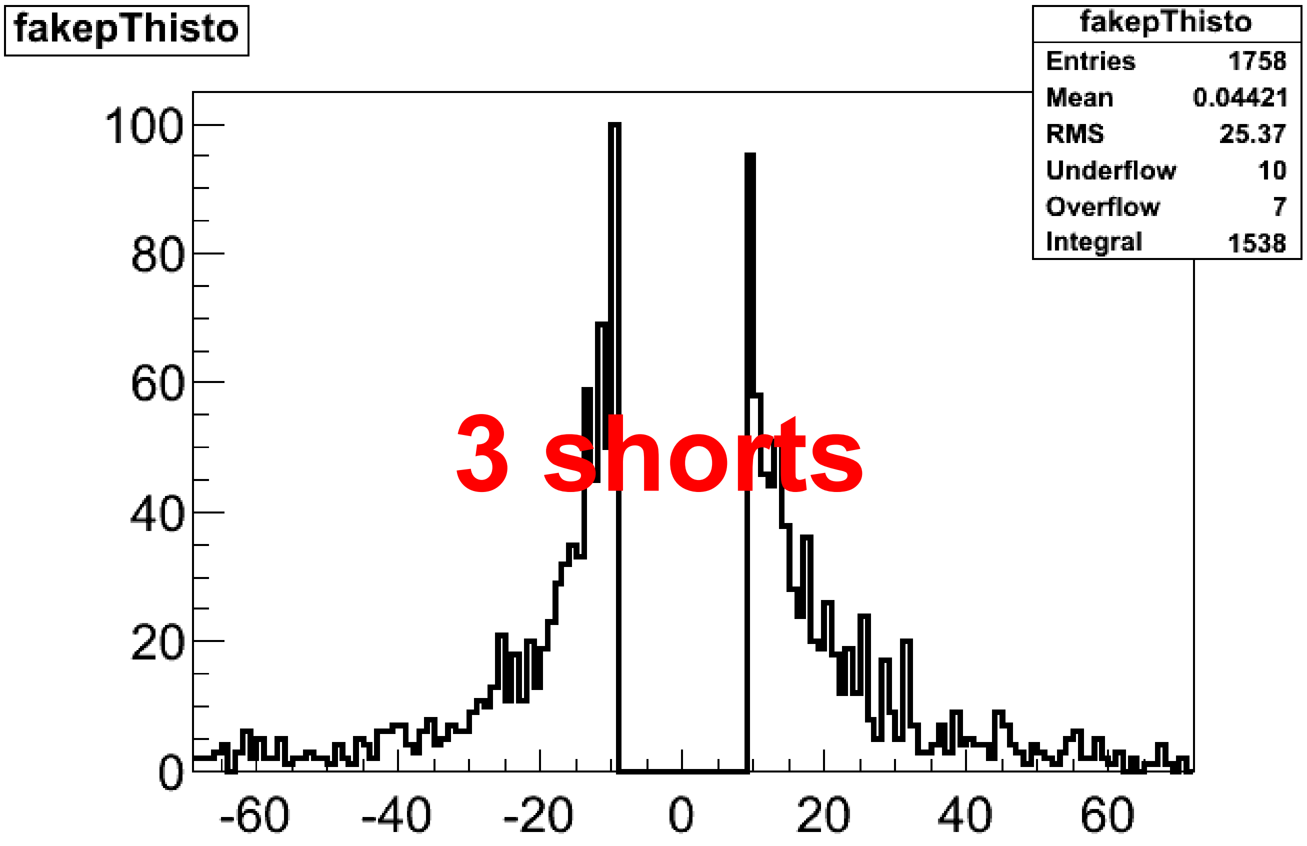
-


Layer 1
Layer 2
Layer 3
Layer 4
Cell 0
Cell 1
Cell 2
Cell 3
word
word
word
word
FF
Output Bus
Data Bus 1
Data Bus 2
Data Bus 3
Data Bus 4

Pattern Counting



Fake p_T distribution



- **fake tracks biased towards lower momenta**