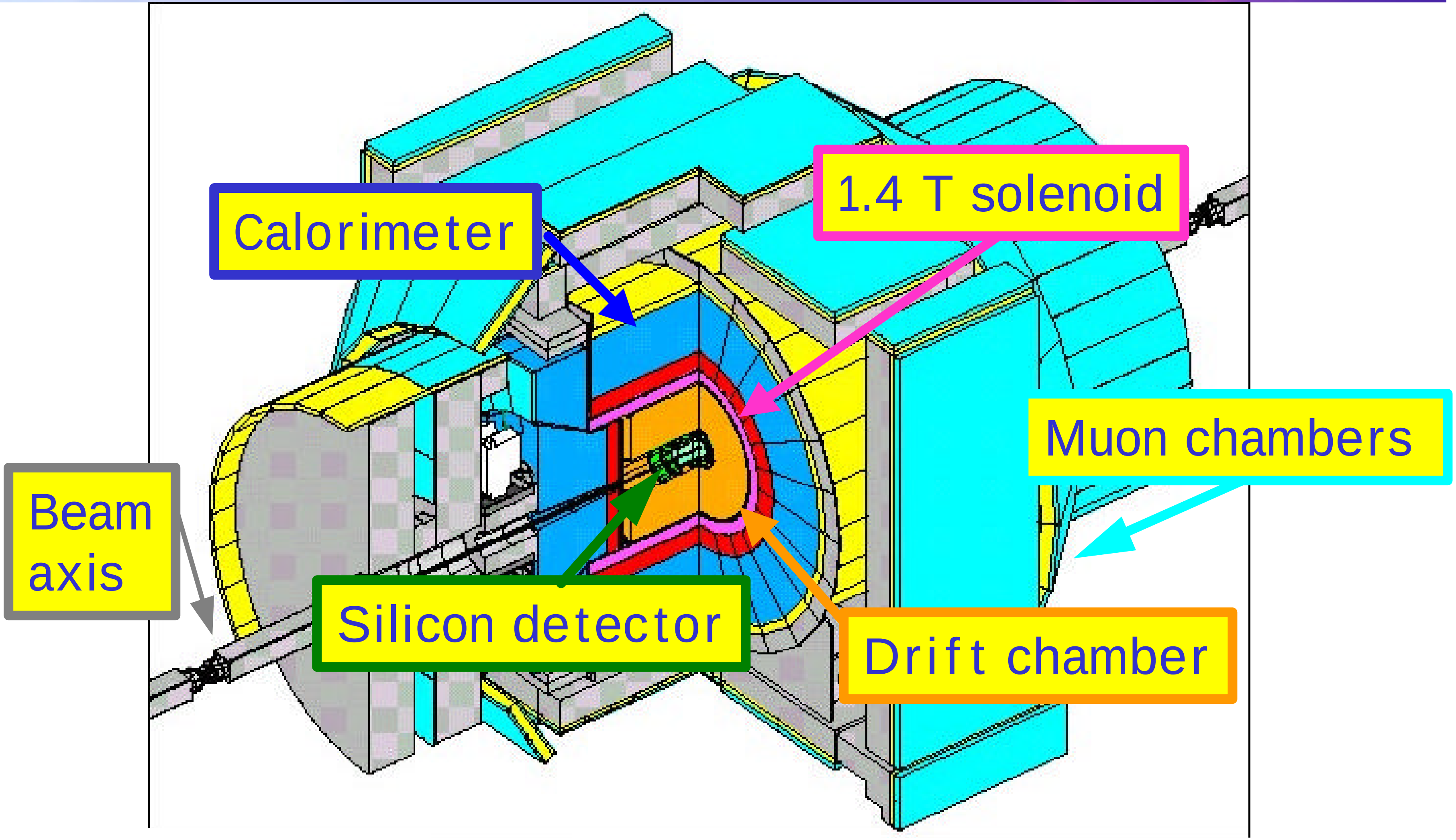
The CDF Silicon Vertex Trigger

Bill Ashmanskas
Argonne National Laboratory
(for the CDF-II Collaboration)
May 30, 2003

Outline

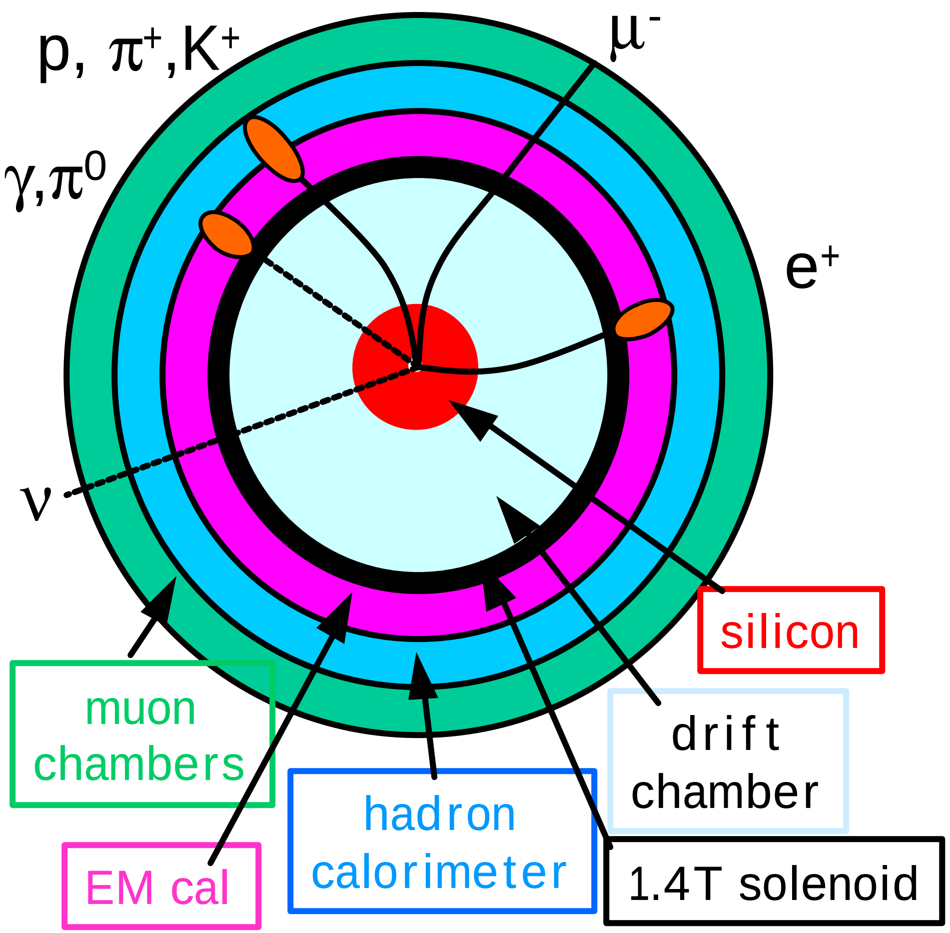
- CDF trigger overview $\rightarrow$ focus on SVT
- How SVT works (performance)
- Why SVT works (diagnostics/flexibility)

CDF Detector



- CDF: multi-purpose experiment, broad physics program
- Tevatron: E_{CM}=1.96 TeV, T_{bunch} = 396 (132) ns, L ~ $10^{32}cm^{-2}s^{-1}$

CDF trigger (detector) signatures



Trigger exploits a wide variety of signatures ...

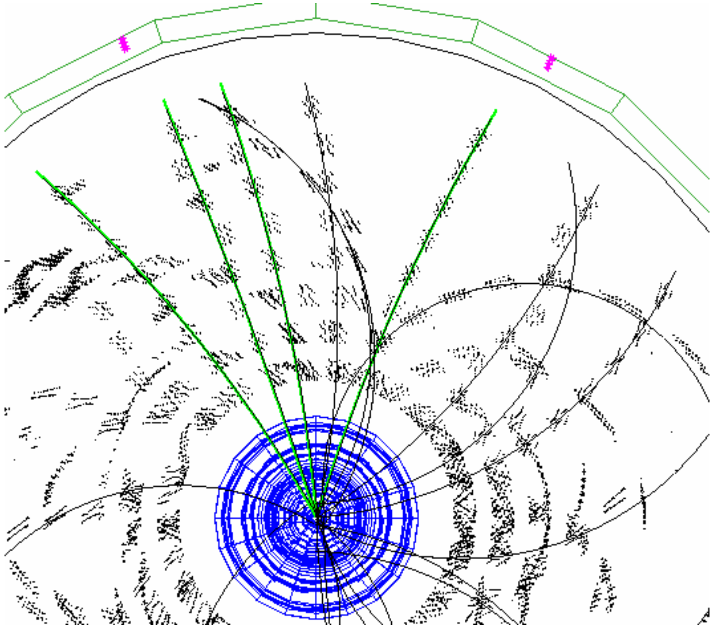
- e, μ from W, Z, SUSY, b
- γ, jet, ν (E imbalance), τ, b jet, ...
- "Displaced" hadrons from bottom and charm decay

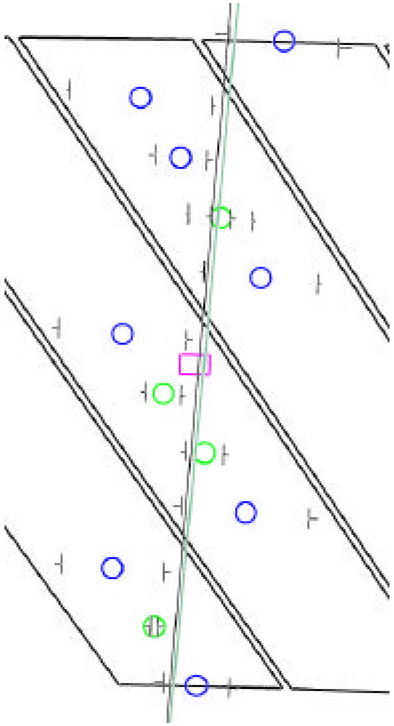
... to distinguish processes with a wide range of rates

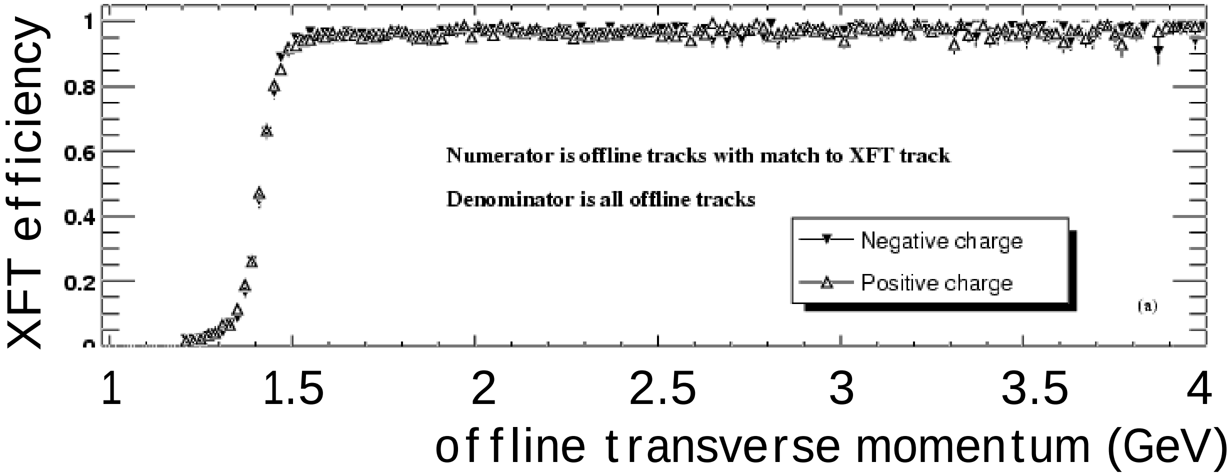
- ~ 50 mb total inelastic
- ~ 100 μb bottom, charm
- ~ 10 μb B, pT>6GeV, |y|<1
- ~ 2 nb W $\rightarrow$ eν
- ~ 5 pb top

Many trigger signatures use drift chamber tracks, e.g. coincidence of D.C. track with EM cal, muon stub, silicon hits

Level 1 drift chamber trigger (XFT)







Finds p_T>1.5 GeV tracks in 1.9 μs

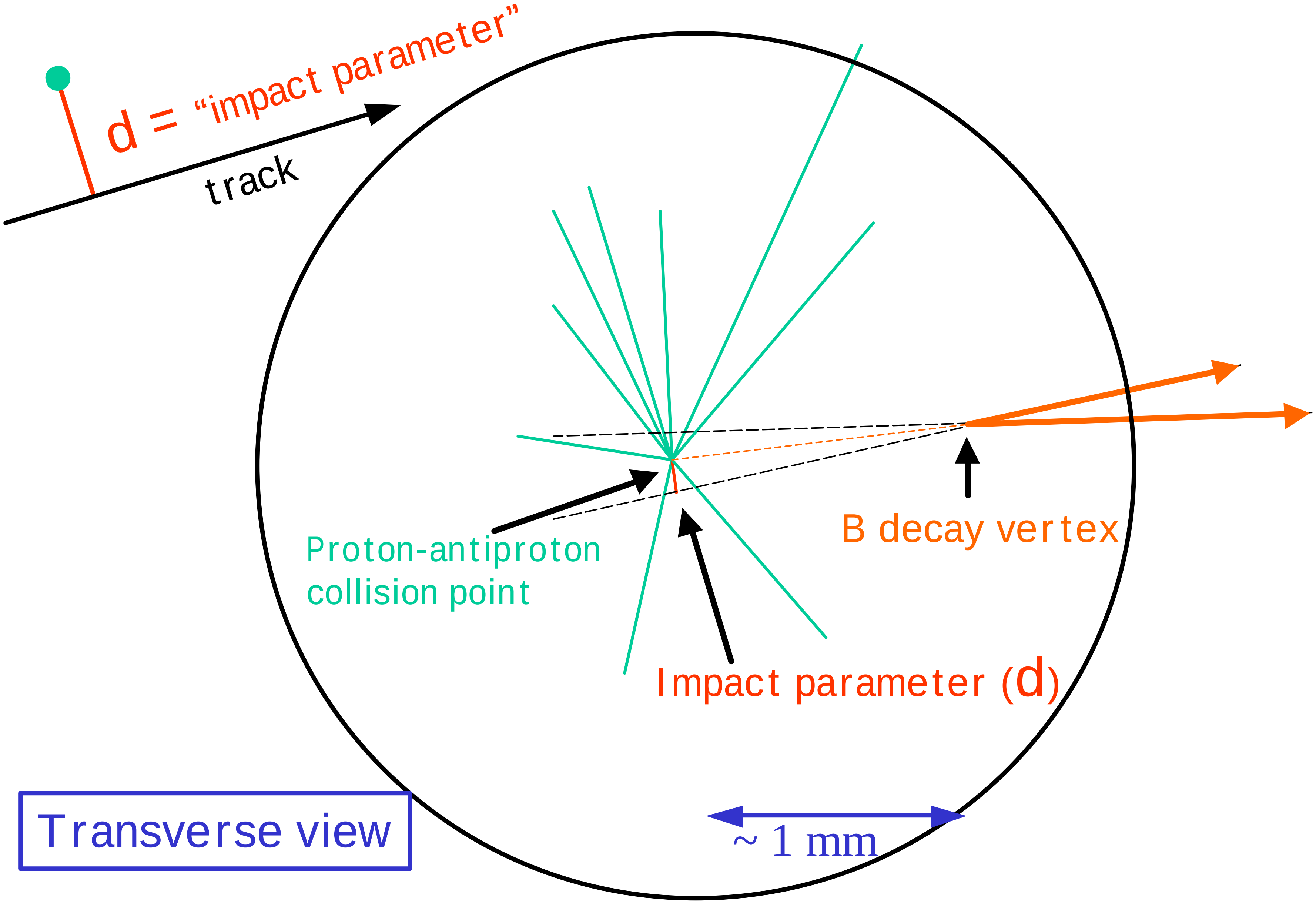
For every bunch crossing (132 ns)!

$\sigma(1/p_T)$ = 1.7%/GeV

$\sigma(\phi_0)$ = 5 mrad

96% efficiency

Exploit lifetime to select b,c decays



CDF three-level trigger

rates (approx) | track trigger b+c purity

FE digitizes (silicon S&H)

7 MHz — ~0.1%

Level 1: ~ 5.5 μsec (42-stage pipeline)

- drift chamber tracks (+ calorimeter, muon, ...)
- look for 2 tracks with momenta > 2 GeV

35 kHz — ~1%

FE boards store to buffer (silicon digitizes/sends)

Level 2: ~30 μsec

- fast silicon tracking (+ cal clustering, shower max, ...)
- look for 2 tracks with impact parameters > 120 μm

350 Hz — ~50%

FE crates transmit to event builder

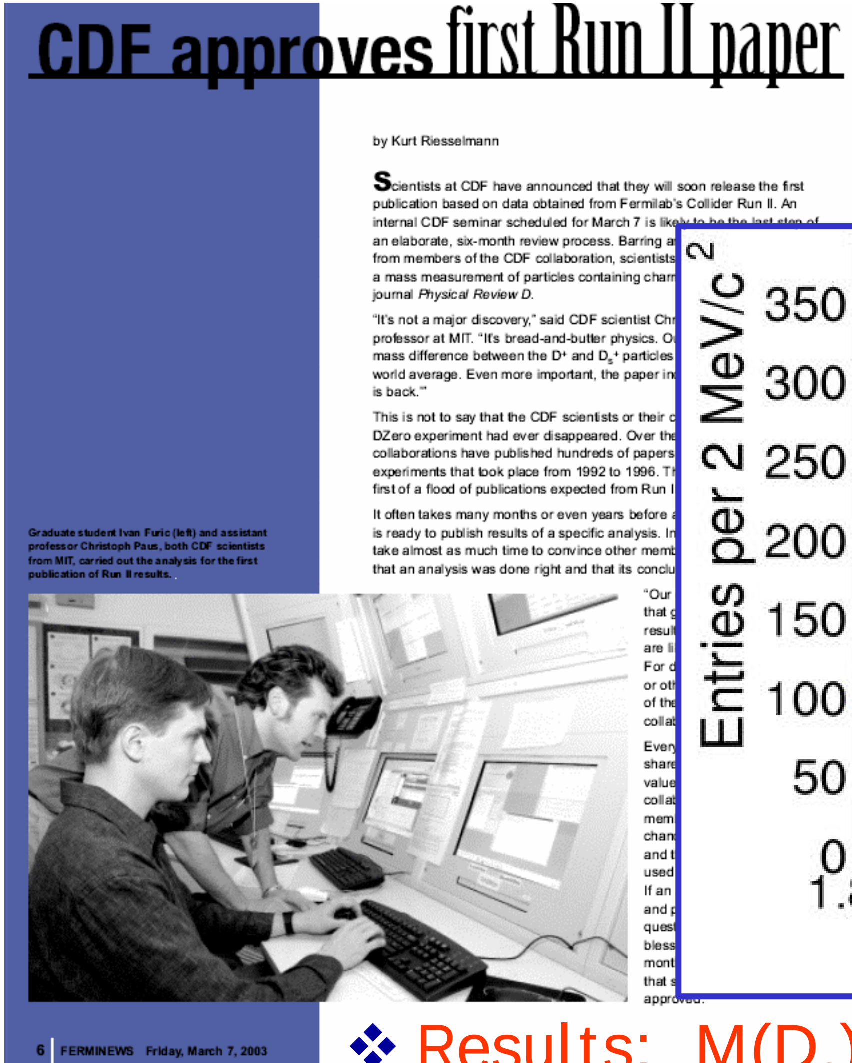
Level 3: ~ 1 second (using ~250 PCs in parallel)

- full event reconstruction; data compression
  - full-precision tracking (includes stereo)
- confirm hardware tracks; form invariant masses

70 Hz — ~90%

data logger writes event to disk

New triggers → a paper from first 12pb^{-1}


CDF approves first Run II paper

by Kurt Riesselmann

Scientists at CDF have announced that they will soon release the first publication based on data obtained from Fermilab's Collider Run II. An internal CDF seminar scheduled for March 7 is likely to be the last step of an elaborate, six-month review process. Barring a[...] from members of the CDF collaboration, scientists [...] a mass measurement of particles containing charm [...] journal *Physical Review D.*

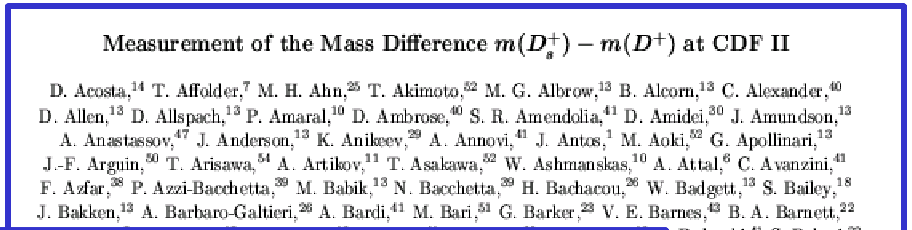
"It's not a major discovery," said CDF scientist Chr[...] professor at MIT. "It's bread-and-butter physics. O[...] mass difference between the D⁺ and D$_s$⁺ particles [...] world average. Even more important, the paper in[...] is back."

This is not to say that the CDF scientists or their c[...] DZero experiment had ever disappeared. Over the [...] collaborations have published hundreds of papers [...] experiments that took place from 1992 to 1996. Th[...] first of a flood of publications expected from Run I[...]

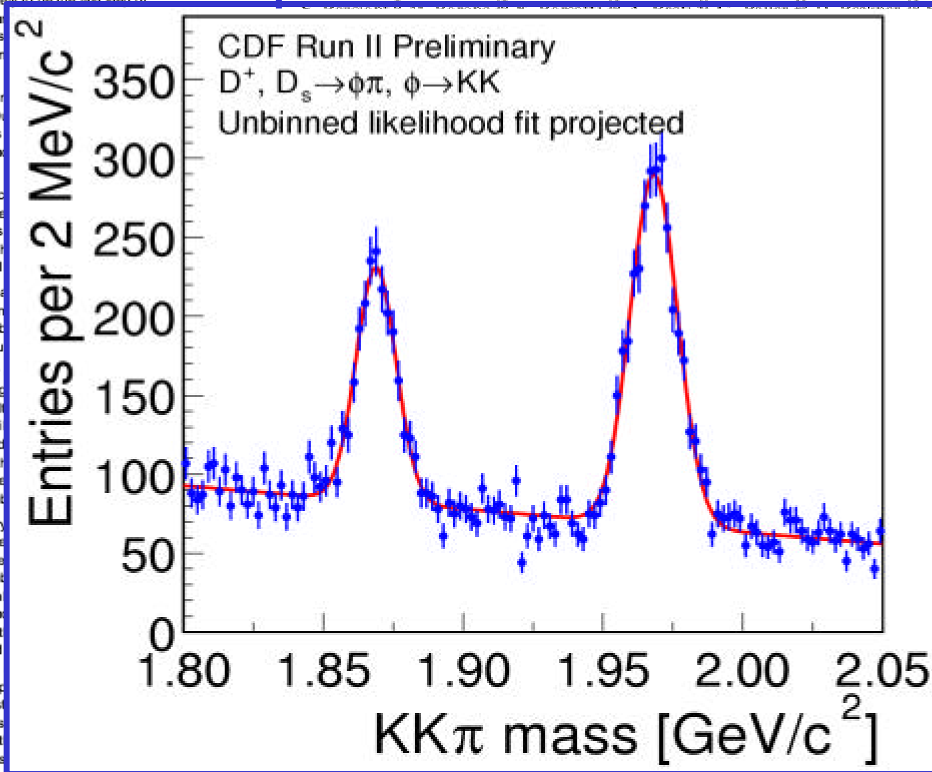
It often takes many months or even years before a[...] is ready to publish results of a specific analysis. In[...] take almost as much time to convince other memb[...] that an analysis was done right and that its conclu[...]

Graduate student Ivan Furic (left) and assistant professor Christoph Paus, both CDF scientists from MIT, carried out the analysis for the first publication of Run II results.

6 | FERMINEWS Friday, March 7, 2003


Measurement of the Mass Difference $m(D_s^+) - m(D^+)$ at CDF II

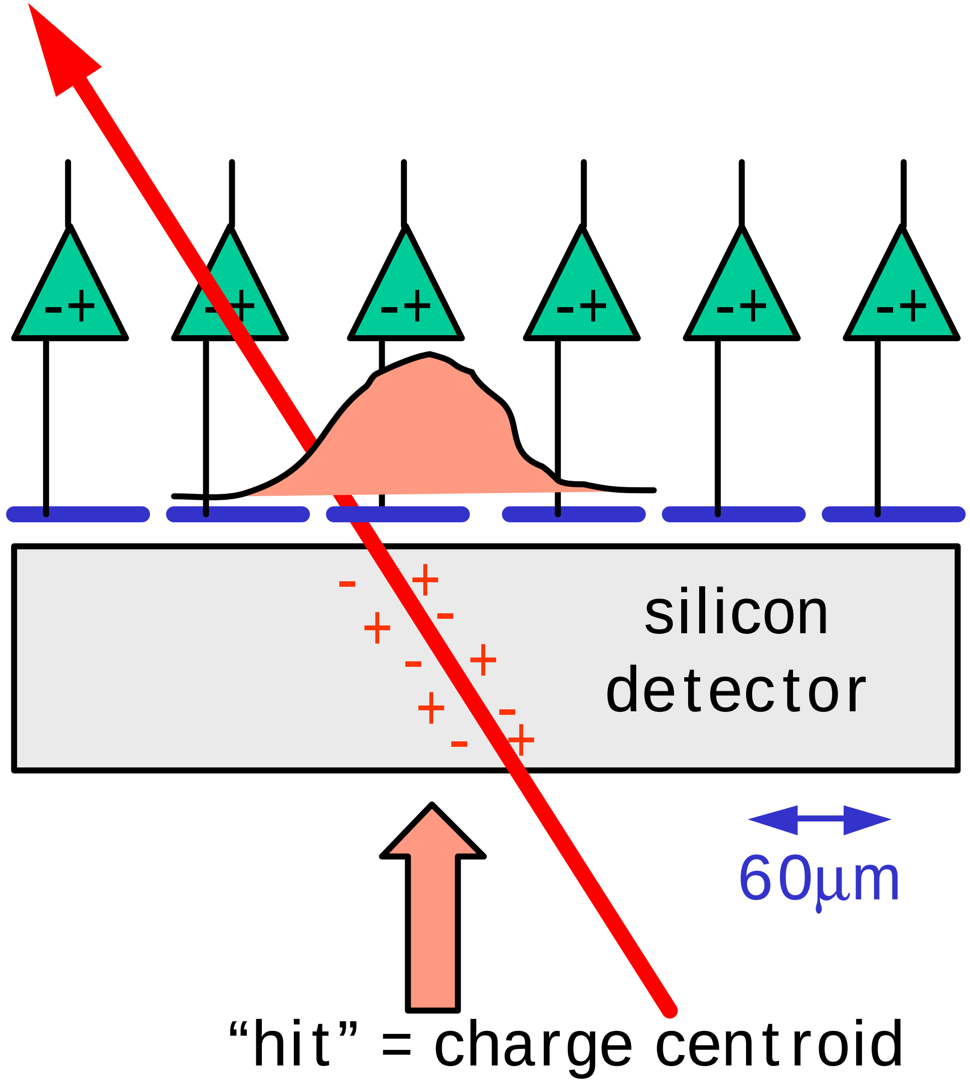
D. Acosta,[14] T. Affolder,[7] M. H. Ahn,[25] T. Akimoto,[52] M. G. Albrow,[13] B. Alcorn,[13] C. Alexander,[40] D. Allen,[13] D. Allspach,[13] P. Amaral,[10] D. Ambrose,[40] S. R. Amendolia,[41] D. Amidei,[30] J. Amundson,[13] A. Anastassov,[47] J. Anderson,[13] K. Anikeev,[29] A. Annovi,[41] J. Antos,[1] M. Aoki,[52] G. Apollinari,[13] J.-F. Arguin,[50] T. Arisawa,[54] A. Artikov,[11] T. Asakawa,[52] W. Ashmanskas,[10] A. Attal,[6] C. Avanzini,[41] F. Azfar,[38] P. Azzi-Bacchetta,[39] M. Babik,[13] N. Bacchetta,[39] H. Bachacou,[26] W. Badgett,[13] S. Bailey,[18] J. Bakken,[13] A. Barbaro-Galtieri,[26] A. Bardi,[41] M. Bari,[51] G. Barker,[23] V. E. Barnes,[43] B. A. Barnett,[22]

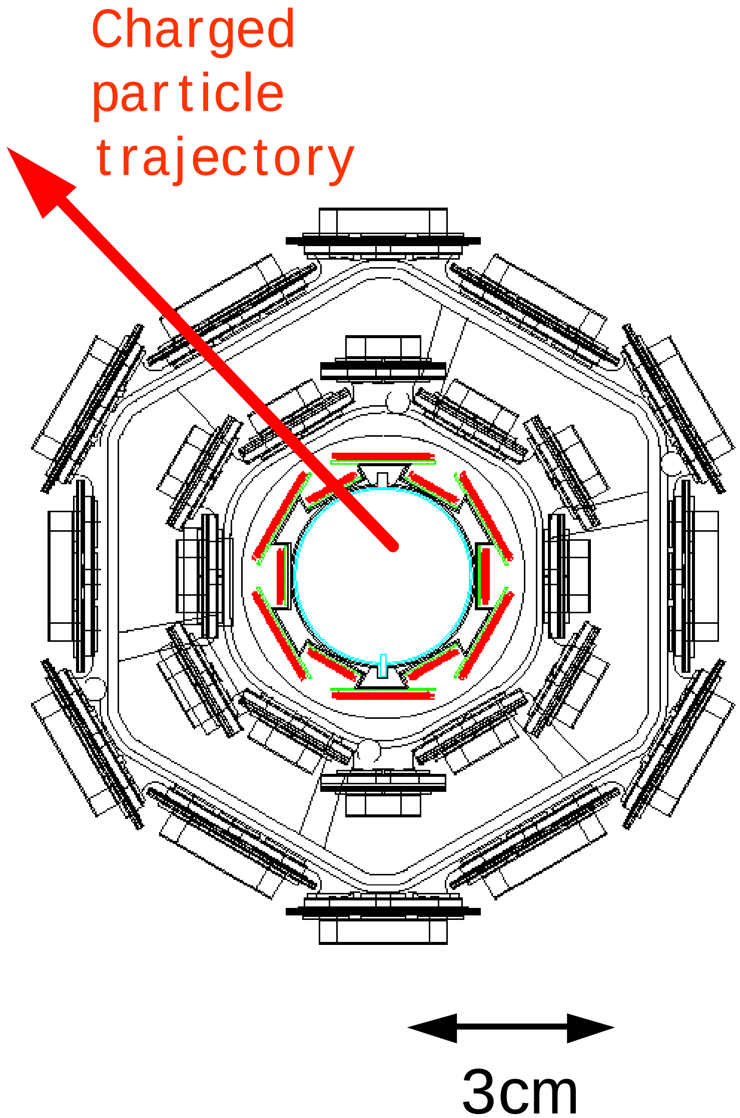


❖ Results: M(D$_s$) – M(D⁺)

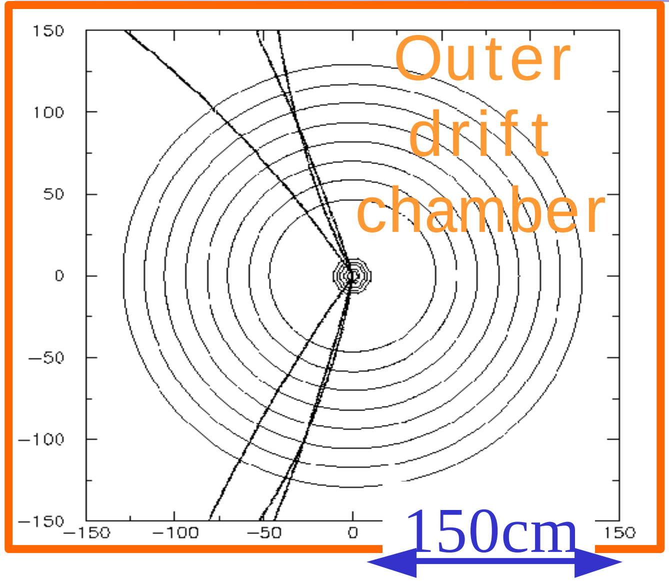
- 99.41 ± 0.38 ± 0.21 MeV/c²
- PDG: 99.2±0.5 MeV/c²

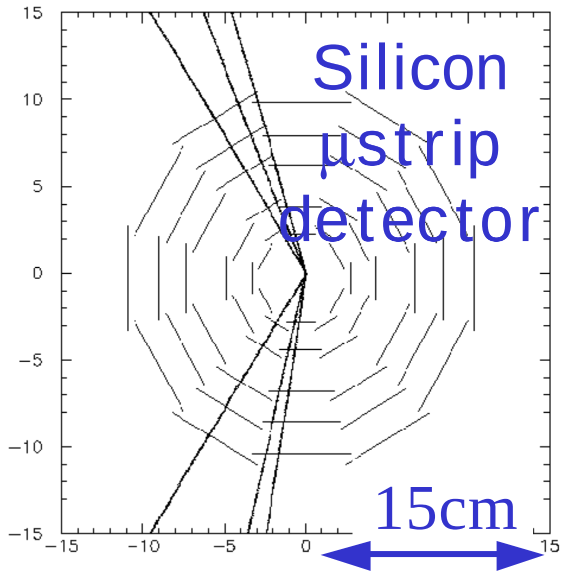
Position measurement ("hit") for charged particle

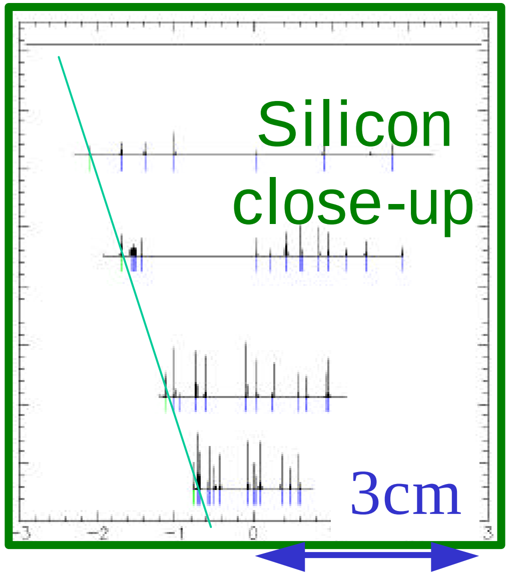




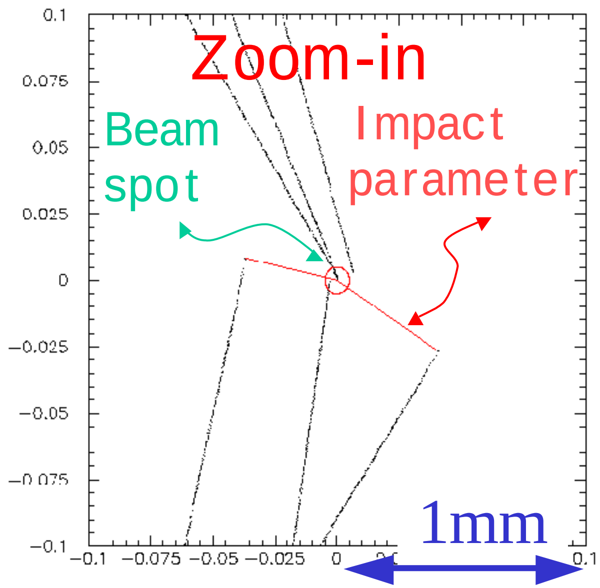
Silicon trigger problem synopsis







Input (every Level 1 accept):
- outer drift chamber trajectories
- silicon pulse height for each channel



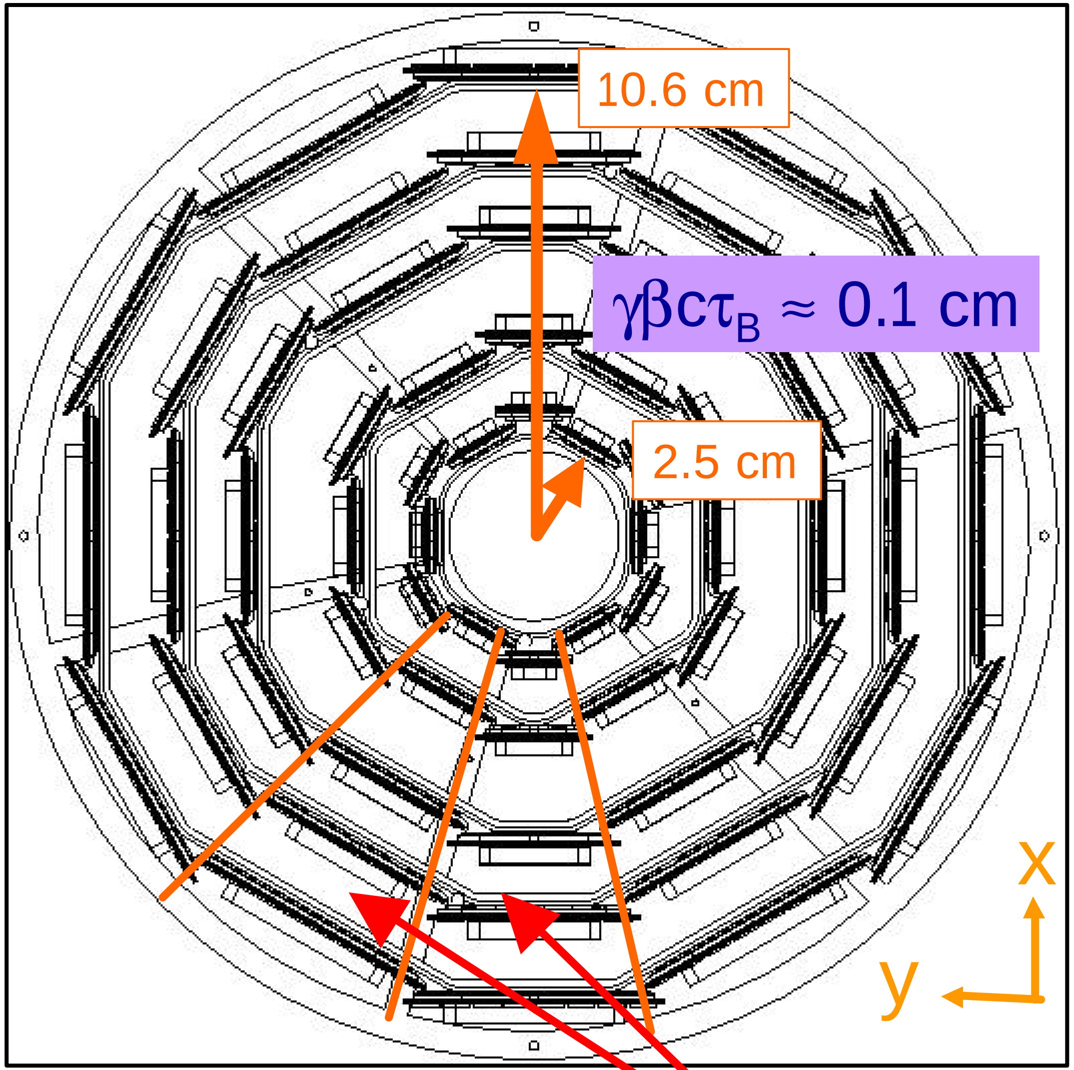
Output (about 20 microseconds later):
- trajectories that use silicon points
- impact parameter: $\sigma(d)$=35um

Doing silicon tracking quickly

Three key features of SVT allow us to do in tens of microseconds what typically takes software hundreds of milliseconds:

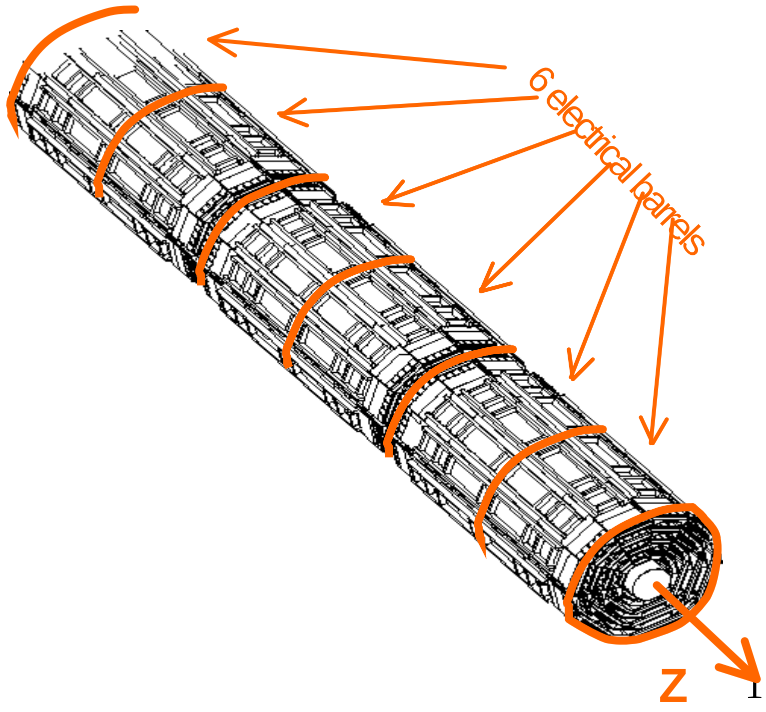
- Parallel/pipelined architecture
- Custom VLSI pattern recognition
- Linear track fit in fast FPGAs

Trick #1: symmetry allows parallelism



Note "wedge" symmetry

Symmetric, modular geometry of silicon vertex detector lends itself to parallel processing

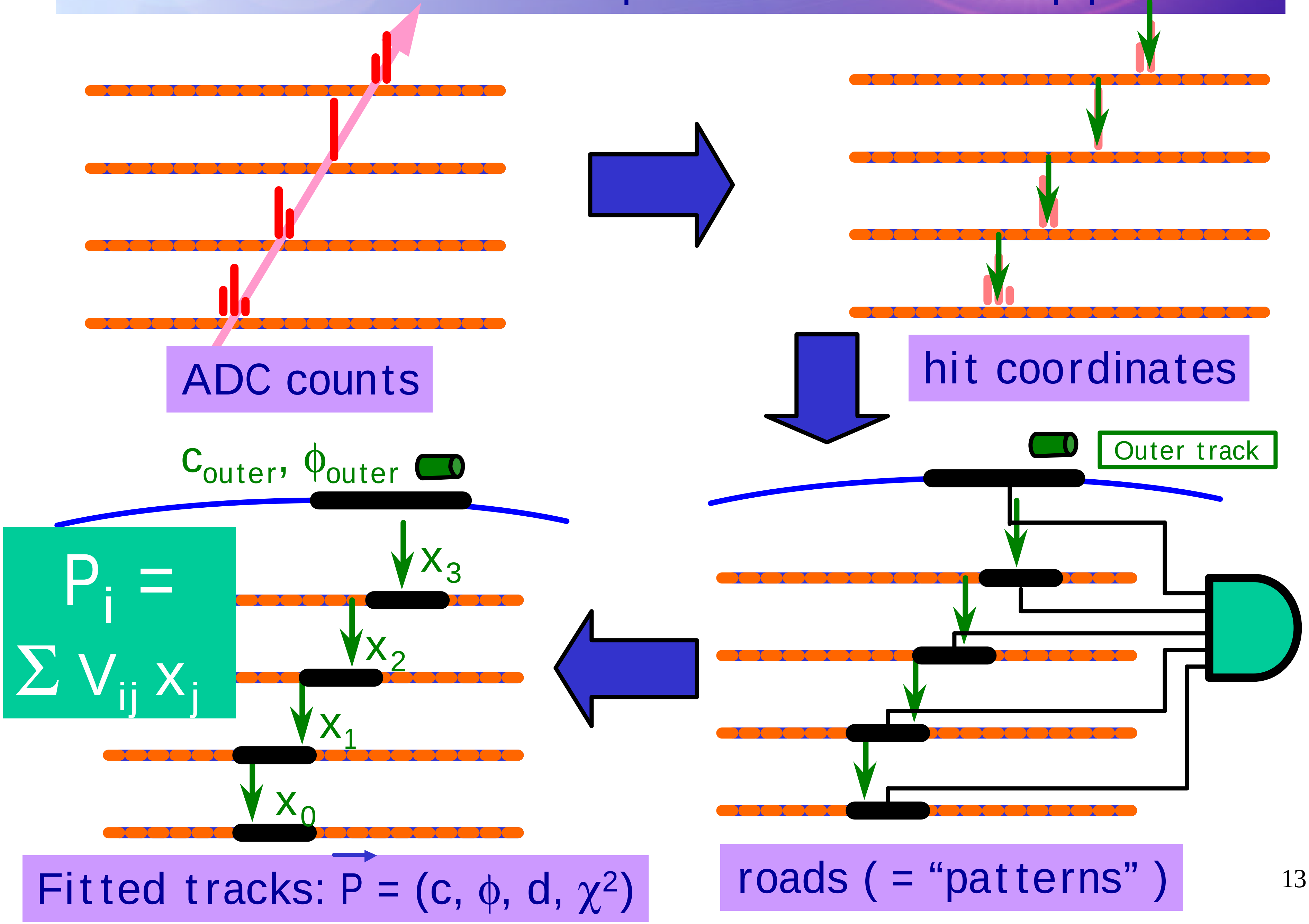


SVT data volume requires parallelism

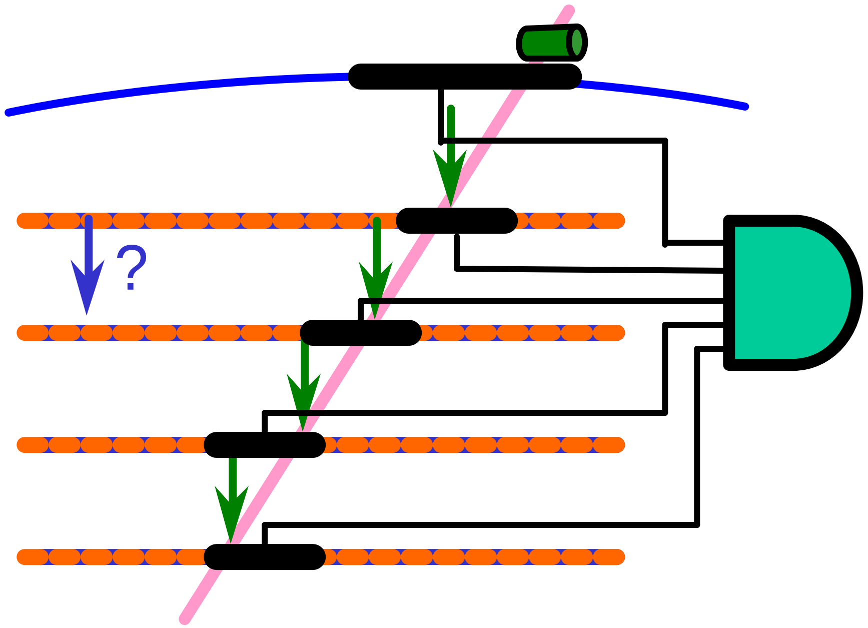
Reduces gigabytes/second to megabytes/second

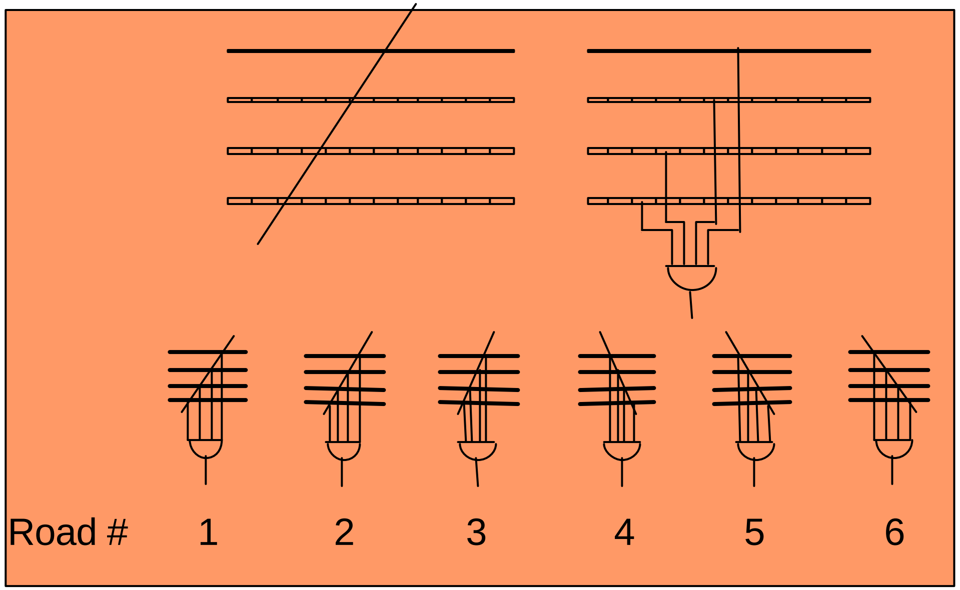
Peak (avg): 20 (0.5) GB/s ⟶ 100 (1.5) MB/s

Each 1/12 of detector is processed in its own pipeline



2nd trick: streamlined track finding





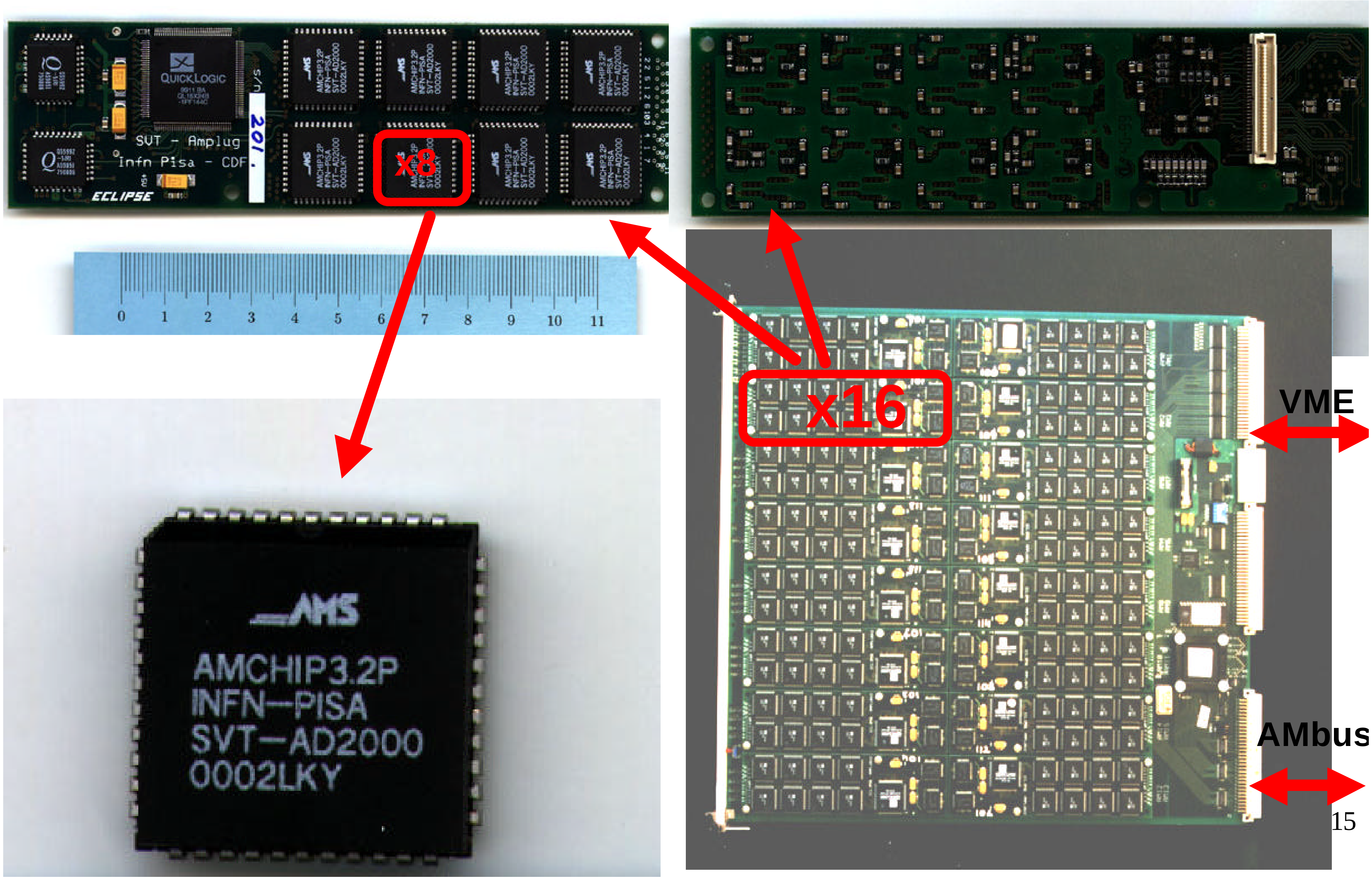
The way we find tracks is a cross between

- searching predefined roads
- playing BINGO

Time ~ $A*N_{hits} + B*N_{matchedroads}$

B	I	N	G	O
2	17	35	48	61
10	21	39	53	66
14	20	free	55	65
8	25	41	52	62
6	16	37	46	67

Custom VLSI AMchip (pattern recognition)

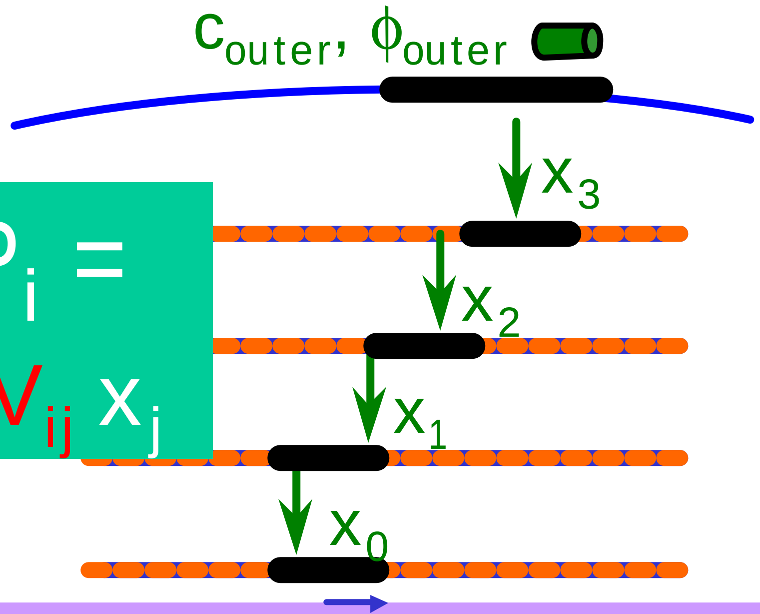


Trick #3: linear fit

Circle($\vec{P}$) Ç Planes at points $\vec{x}$

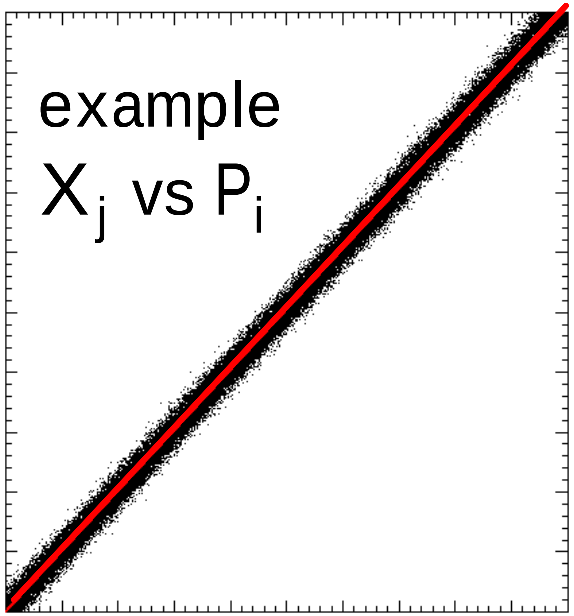
$\vec{x}$ not in general linear in $\vec{P}$

But for P > 2 GeV, d < 1 mm,
linear fit biases d ~ few %
⇒ no problem for trigger



$$P_i = \sum V_{ij}\, x_j$$

Fitted tracks: $\vec{P}$ = (c, φ, d, χ^2)



We derive V_{ij} by linear regression to Monte Carlo data

<u>Trick #3a</u>: use road as a hint

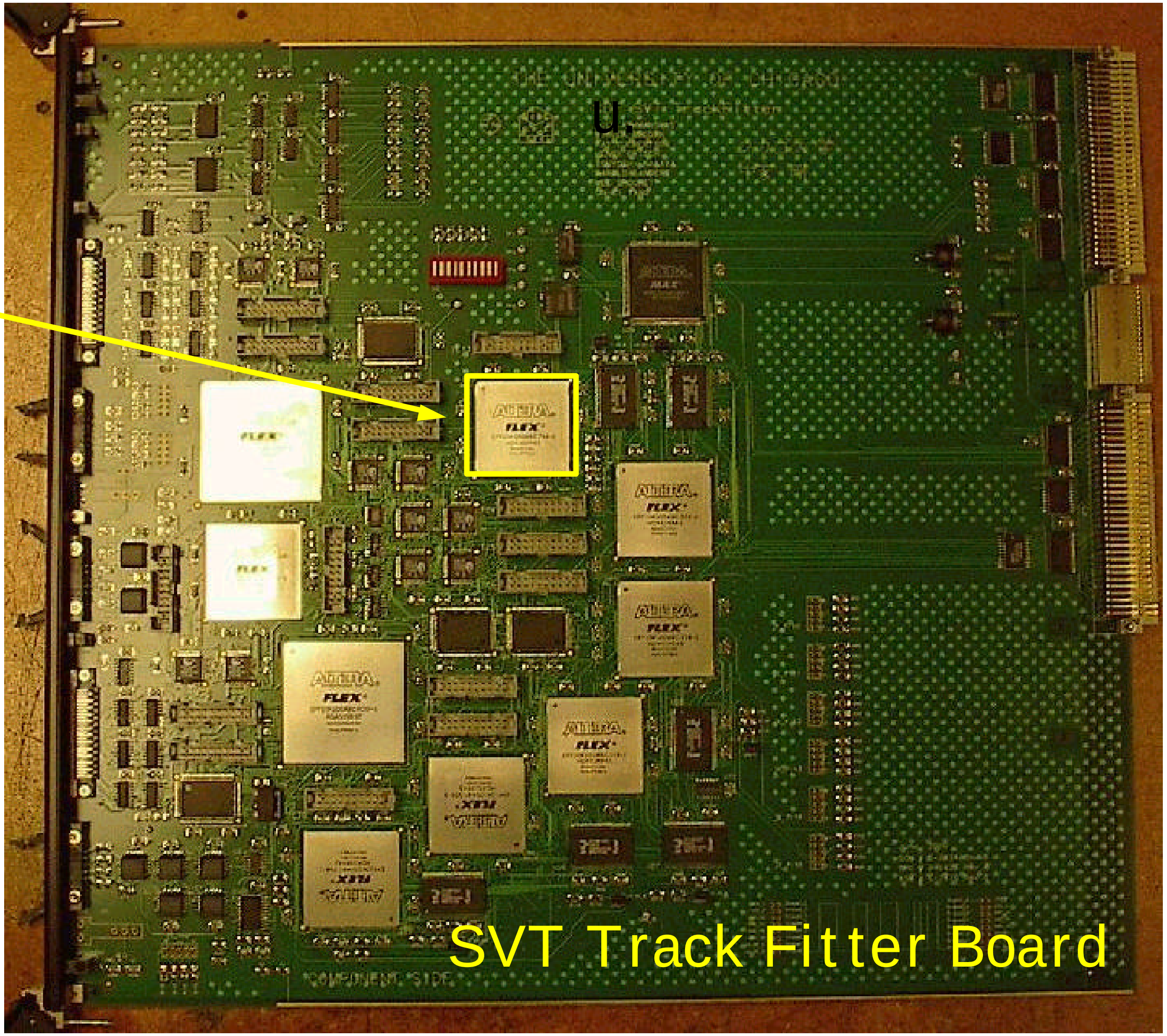
precompute $V_{ij}\, X_j^{road}$

⇒ 250 nsec per fit !

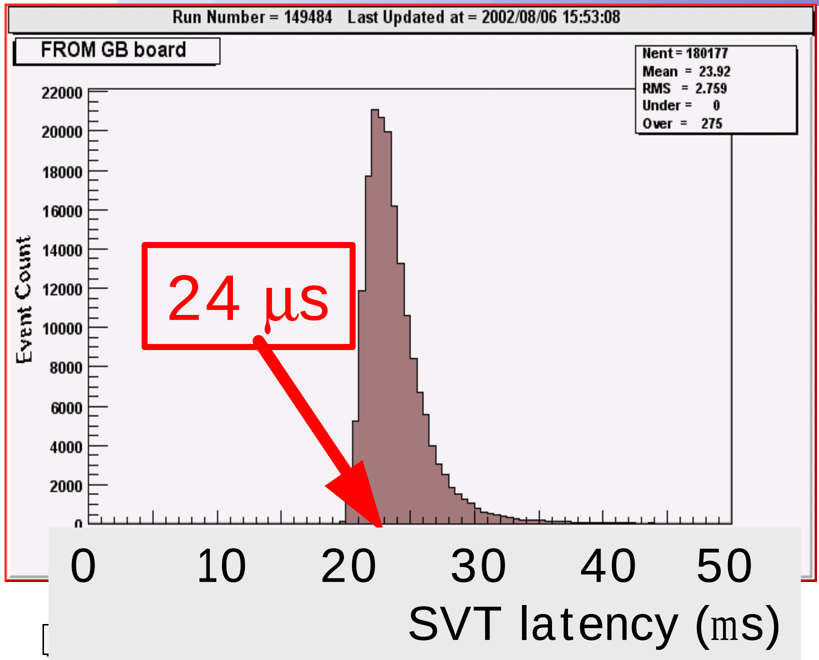
Least squares fit is performed in FPGA

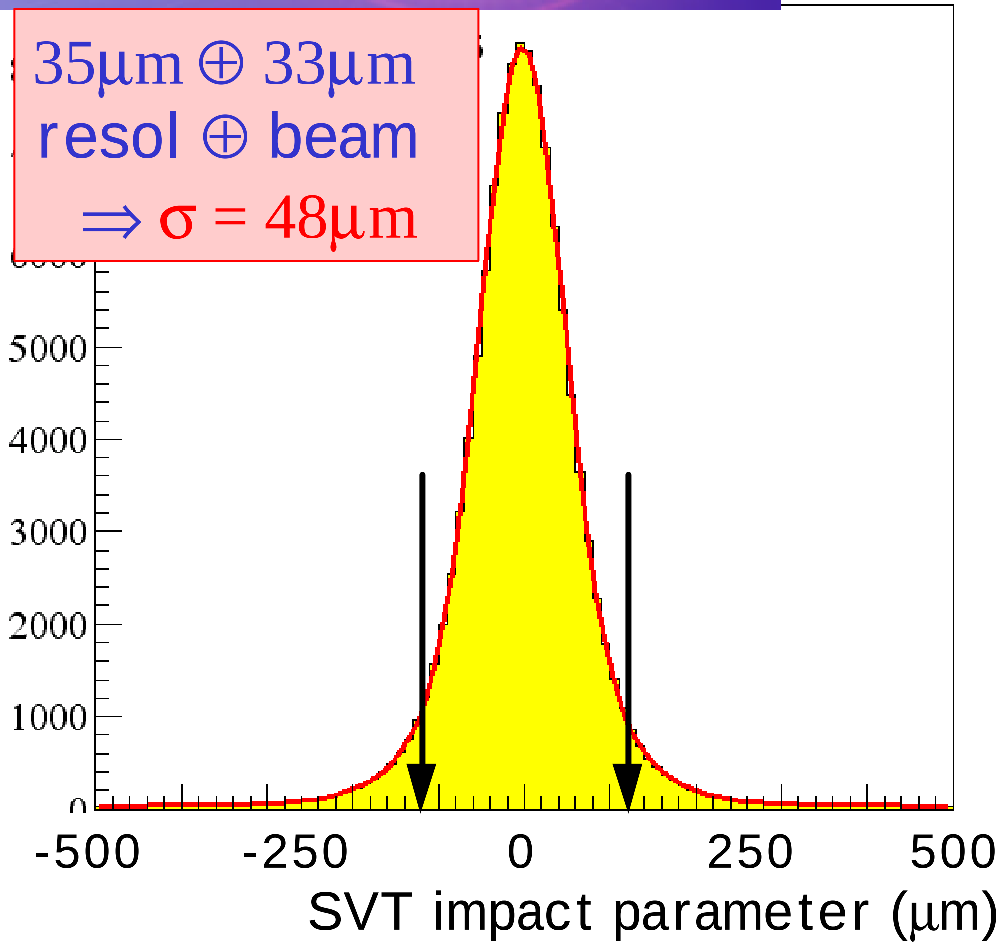
The 6 scalar products are computed in parallel

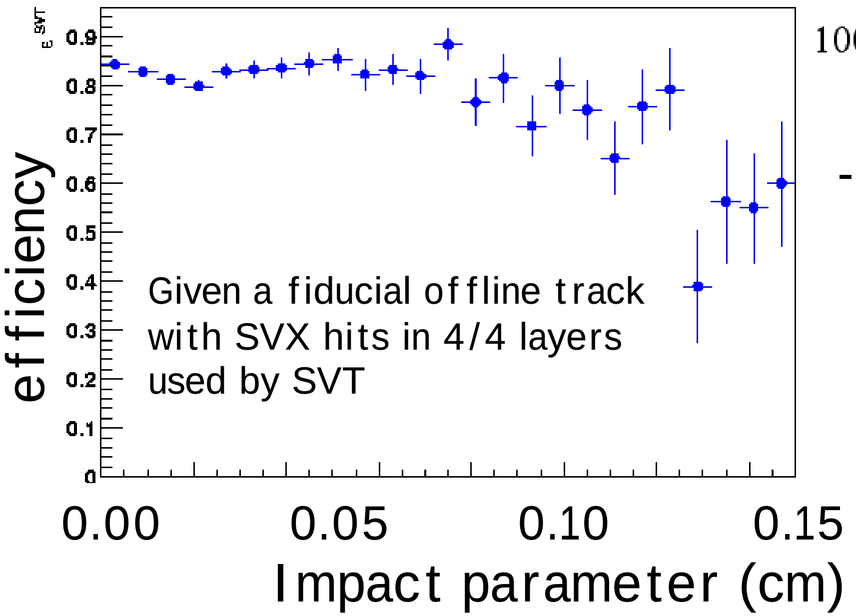
Each fit done in 250nsec



Performance





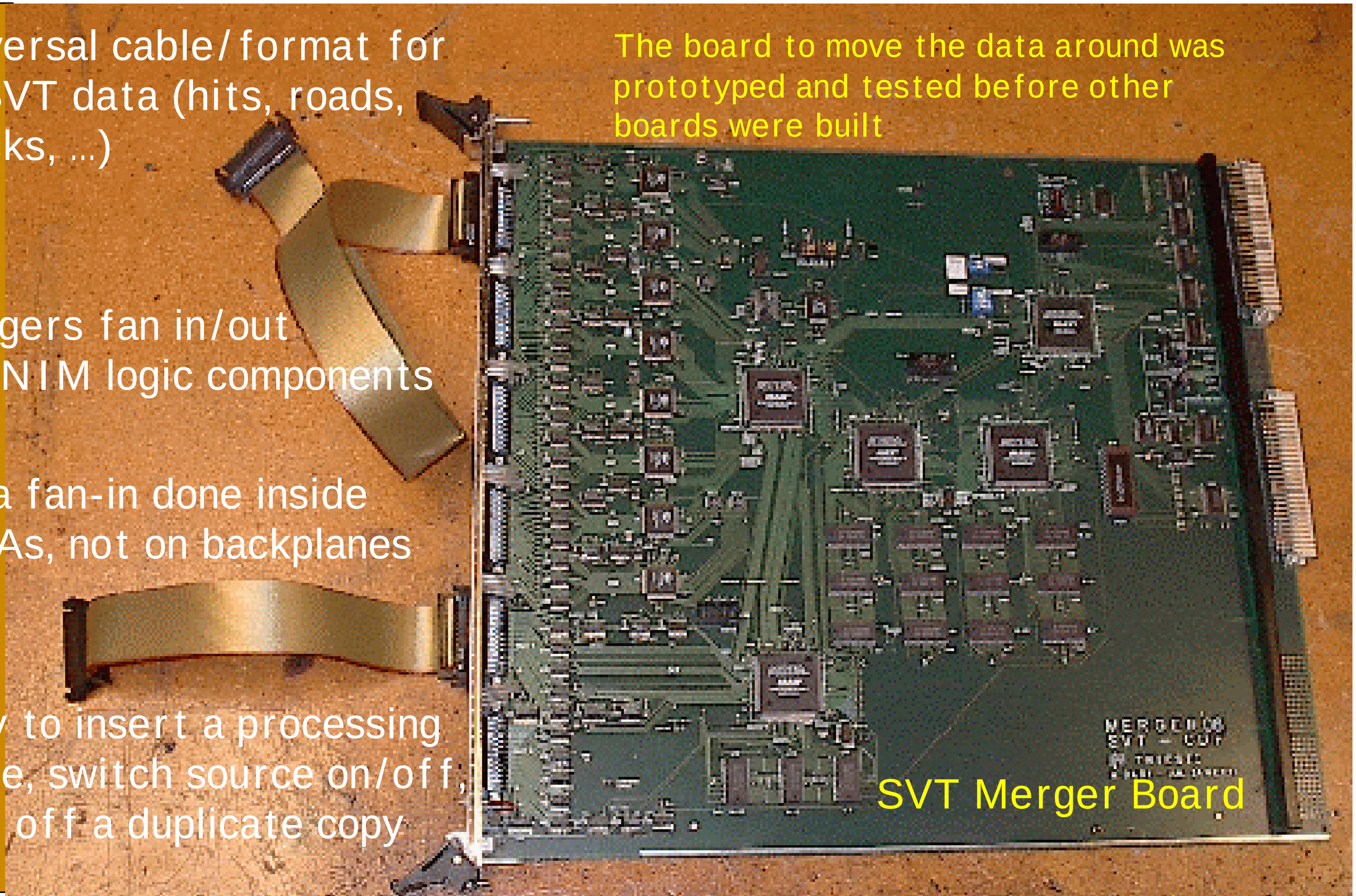


N.B.: lots of trigger & detector work to keep efficiency high
→ Lester Miller's talk on CDF silicon

Merger & cable capture essence of SVT architecture

- Universal cable/format for all SVT data (hits, roads, tracks, ...)
- Mergers fan in/out like NIM logic components
- Data fan-in done inside FPGAs, not on backplanes
- Easy to insert a processing stage, switch source on/off, split off a duplicate copy

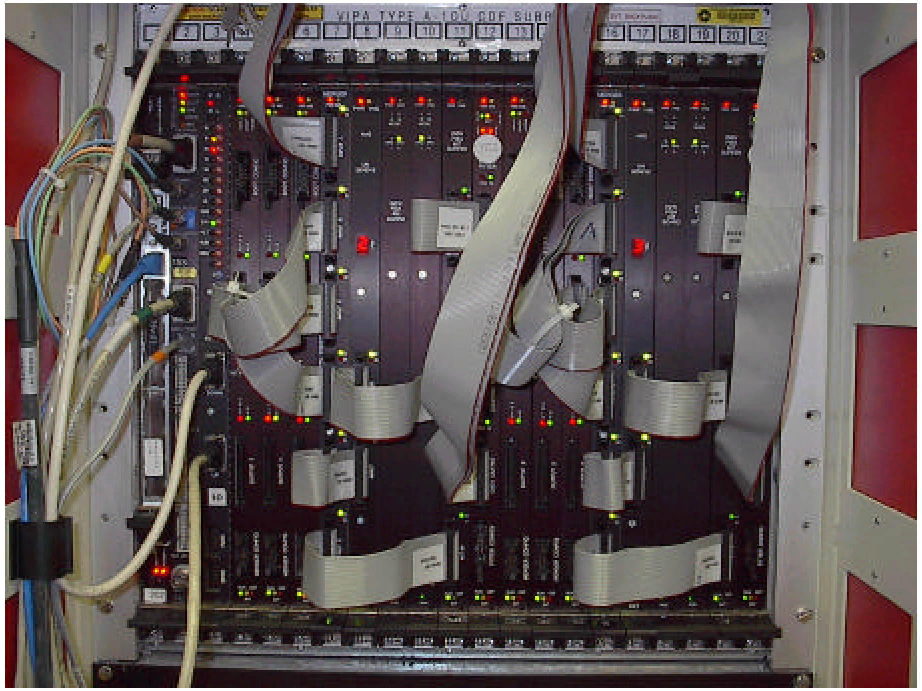
The board to move the data around was prototyped and tested before other boards were built

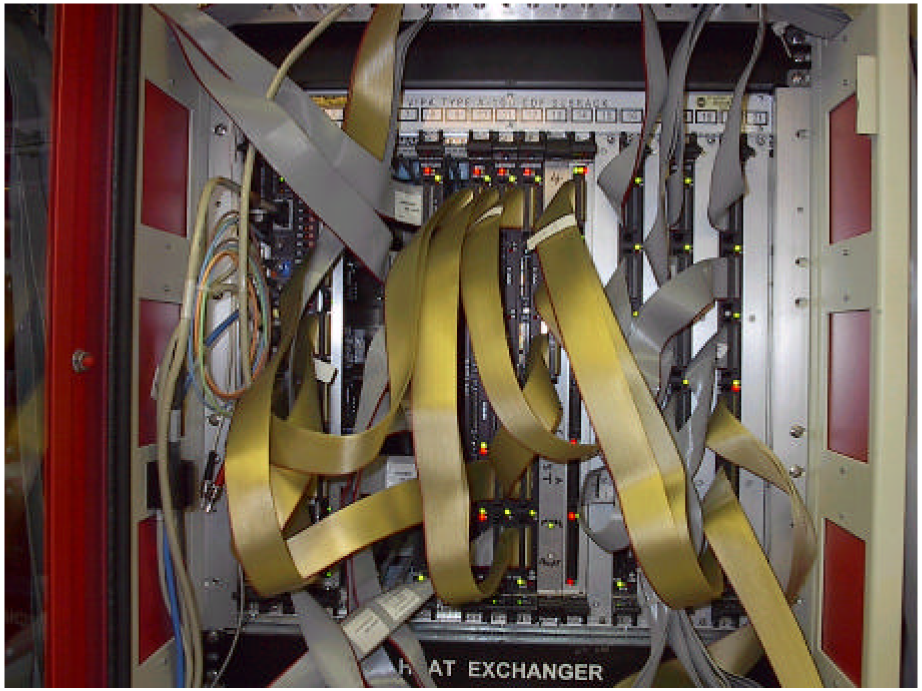


Universal cable -> can adapt quickly, as with NIM logic

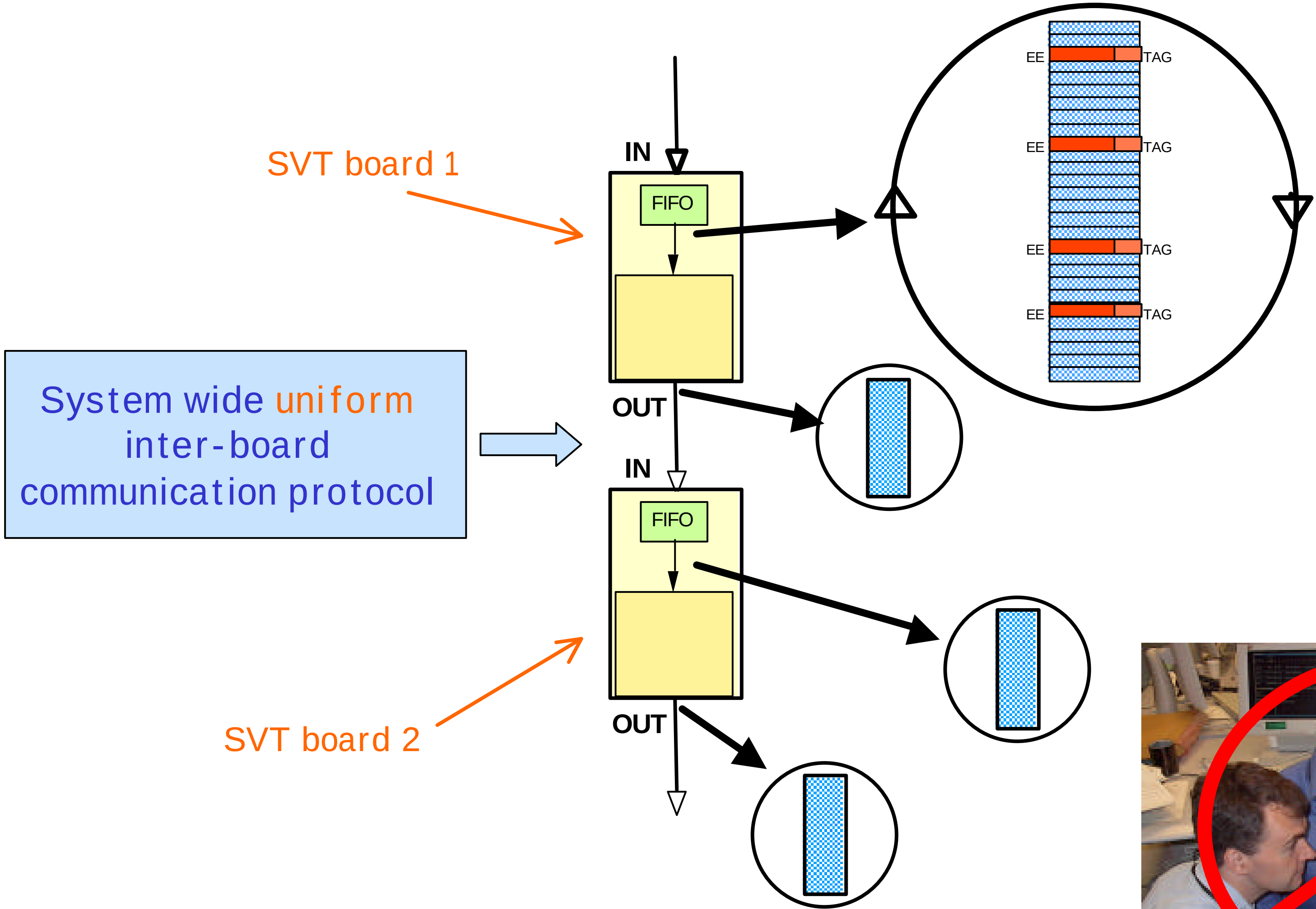
Most of SVT's cabling was carefully planned a priori

But the flexibility to adapt to the unforeseen was a big plus during commissioning

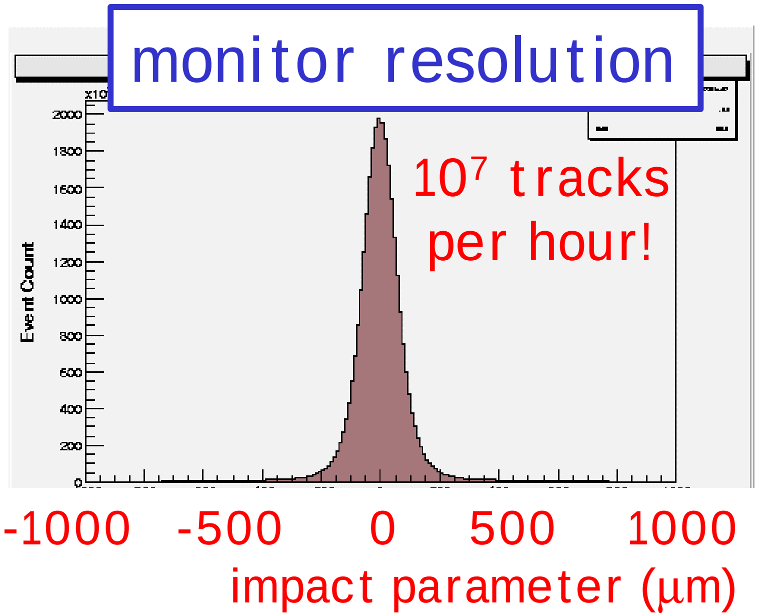


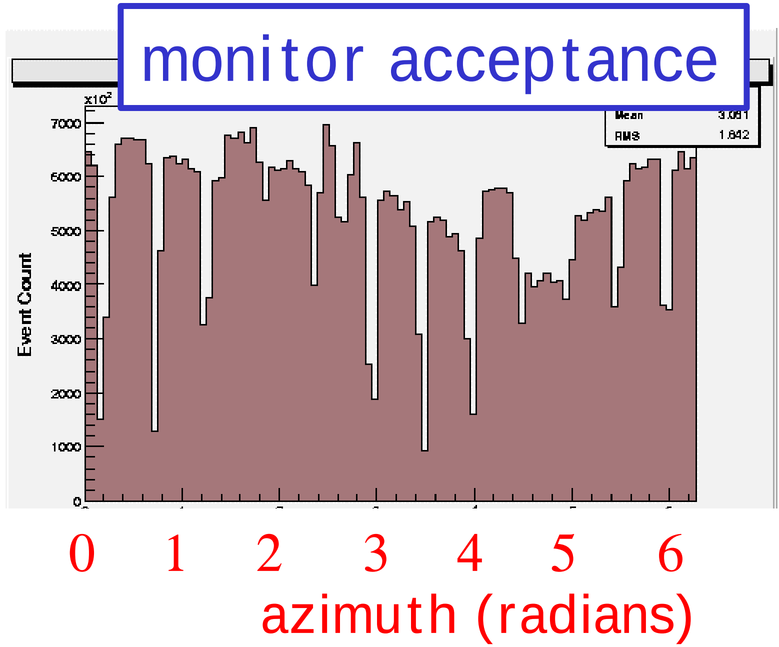


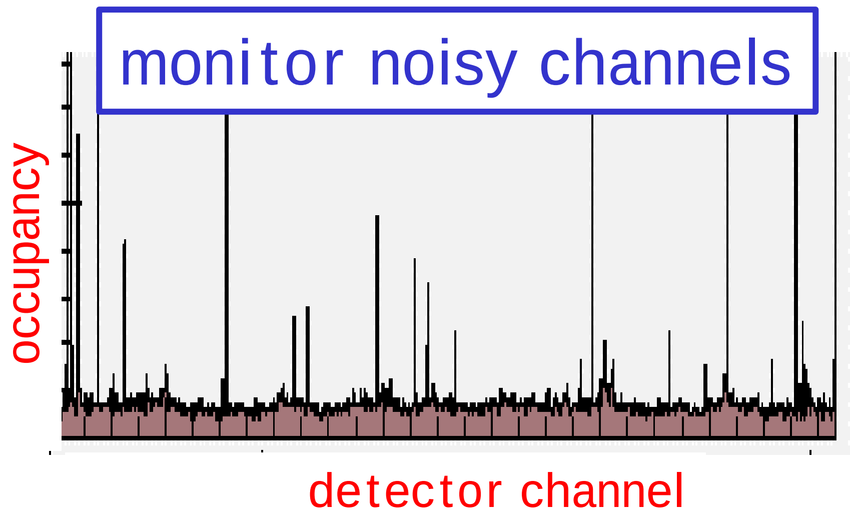
Circular buffers monitor every data link: like a built-in logic analyzer



On-crate monitoring of circular buffers





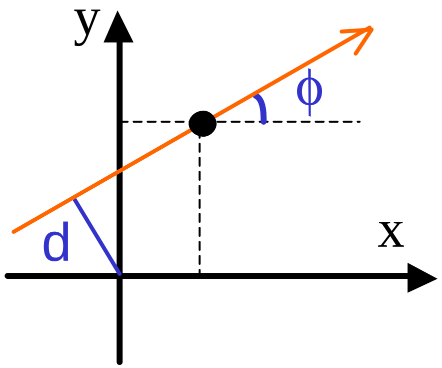


Sample hits, roads, tracks at high rate

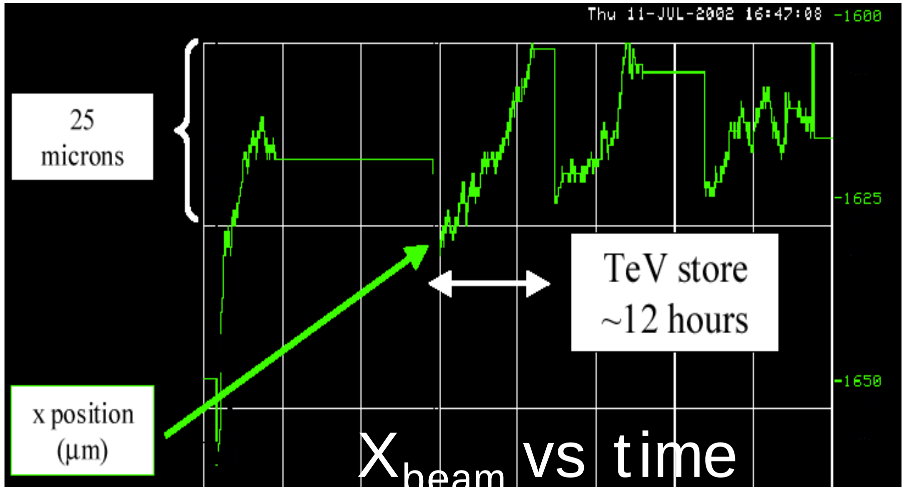
Check boards against emulation software

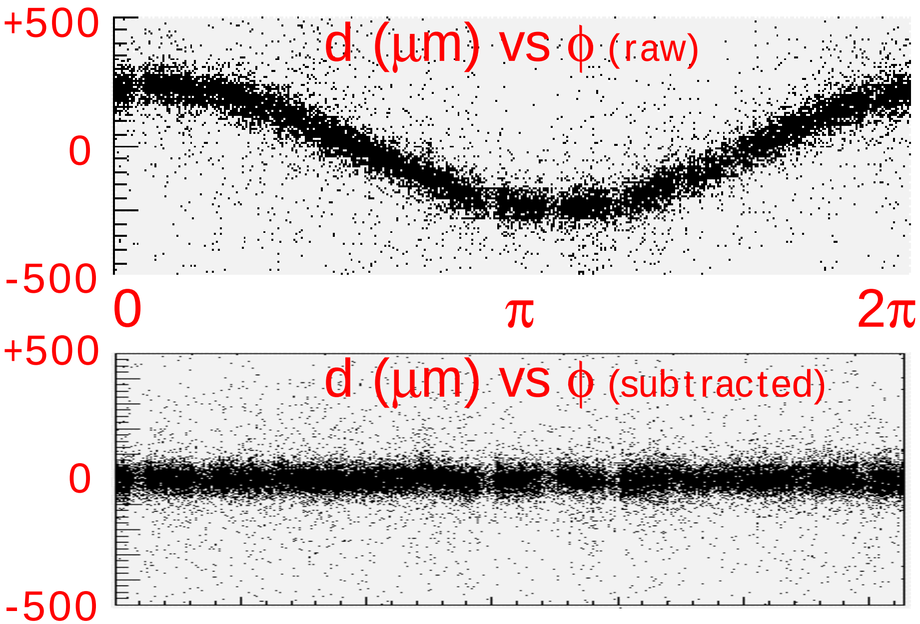
Fit for beam position ...

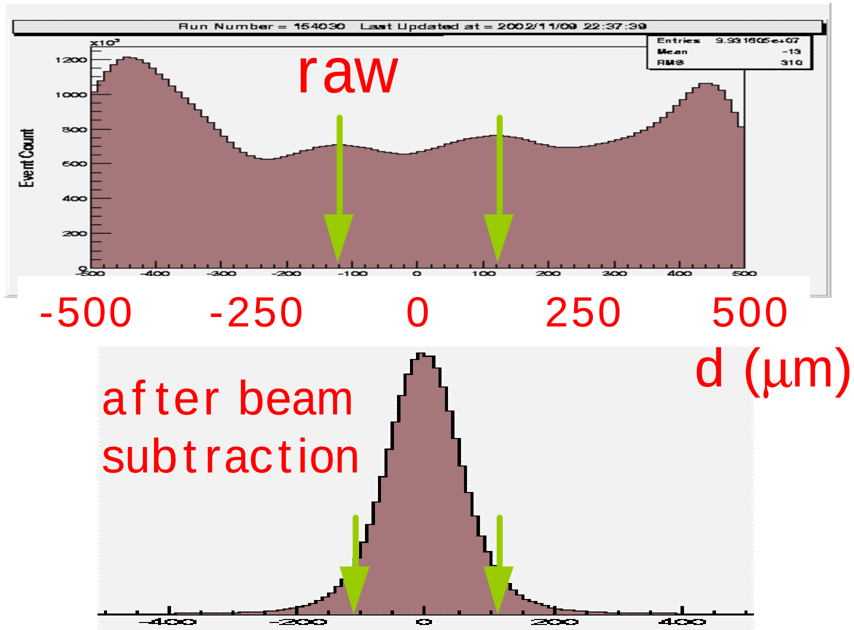
Online beamline fit / correction



$$\langle d \rangle = Y_{beam}\cos\phi - X_{beam}\sin\phi$$







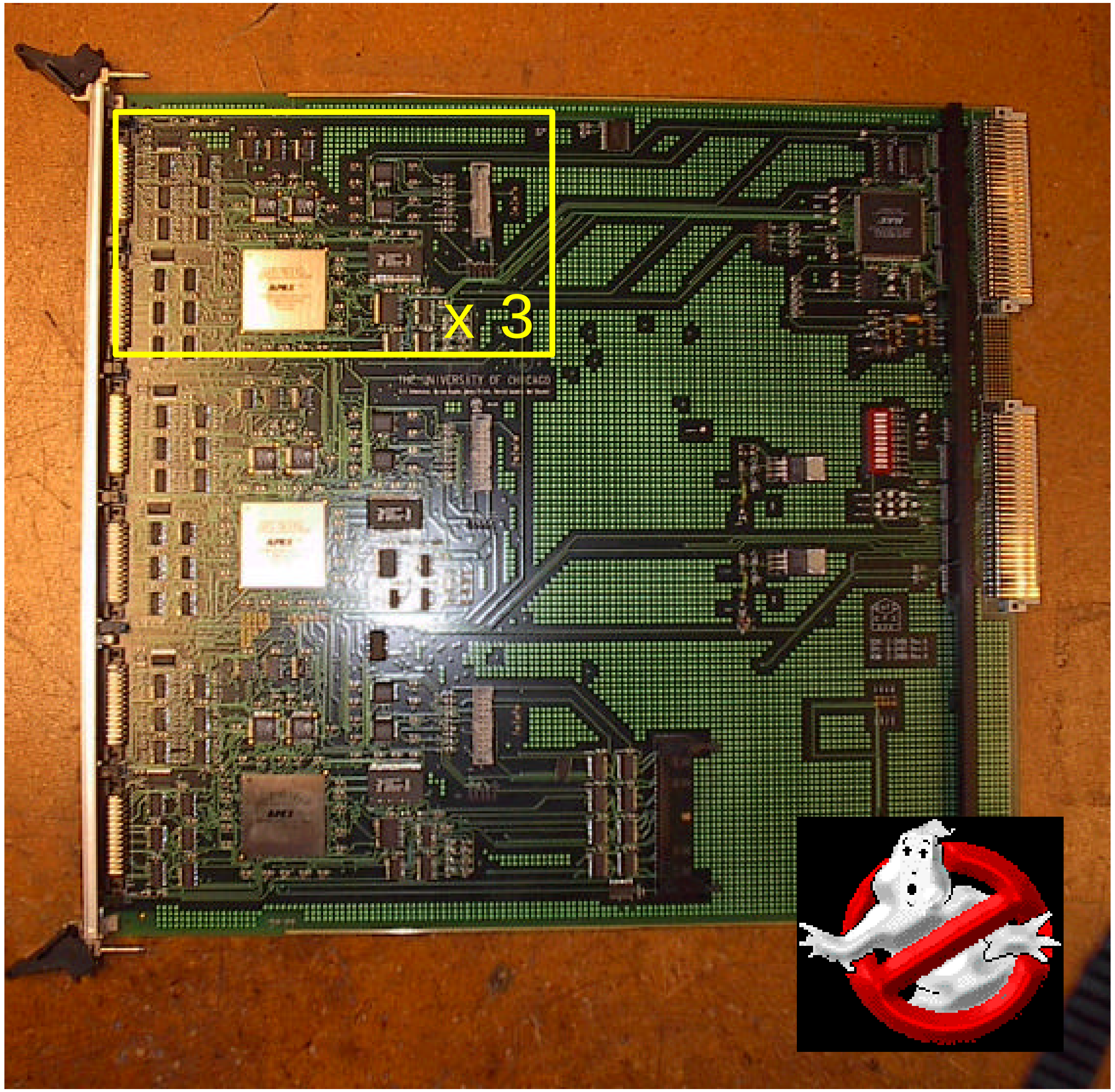
SVT board-on-a-chip

Input + output + one Altera APEX FPGA

- Modern FPGAs have impressive capabilities

Swiss army knife:

- Subtract beam offset
- Filter duplicate ("ghost") tracks
- Read diagnostic data into DAQ
- Measure timing



Why SVT succeeded

- Performance:
  - Parallel/pipelined architecture
  - Custom VLSI pattern recognition
  - Linear track fit in fast FPGAs
- Reliability:
  - Easy to sink/source test data (many boards can self-test)
  - Modular design; universal, well-tested data link & fan-in/out
  - Extensive on-crate monitoring during beam running
  - Detailed CAD simulation before prototyping
    - See poster by Mircea Bogdan
- Flexibility:
  - System can operate with some (or all) inputs disabled
  - Building-block design: can add/replace processing steps
  - Modern FPGAs permit unforeseen algorithm changes
- Key: design system for easy testing/commissioning