FERMILAB-THESIS-2010-78

Università di Pisa

Facoltà di Scienze Fisiche, Matematiche e Naturali

Tesi di Dottorato in Fisica Applicata

GigaFitter at CDF: Offline-Quality Track Fitting in a Nanosecond for Hadron Collider Triggers

Candidato:
Francesco Crescioli

Supervisor:
Prof. Mauro Dell'Orso

SSD FIS/01

Ciclo XXII - 2010

To my wife

Contents

List of Figures **vii**

List of Tables **xi**

1 Overview **1**

2 Hadron Collider experiments and Trigger **7**

2.1 Online event selection 11

2.1.1 Multi-level trigger 13

2.1.2 Trigger at CDF 15

2.1.2.1 The CDF experiment 15

2.1.2.2 The CDF three-level trigger 19

2.2 Tracks in trigger: examples 20

2.2.1 Electron and muons in CDF 20

2.2.2 High p_T lepton isolation in ATLAS 22

2.2.3 Two-Track Trigger 25

2.2.4 High p_T triggers in CDF 29

3 Tracking in High Energy Physics **31**

3.1 The SVT Algorithm 33

3.1.1 Linear fit 34

3.1.2 Pattern and associative memory 39

4 Silicon Vertex Trigger 43

4.1 Design and performances 44

4.1.1 Hardware structure 46

4.1.2 Tracking resolution 51

4.1.3 Efficiency 51

4.1.4 Diagnostic features 58

5 GigaFitter 61

5.1 Design considerations and features 61

5.1.1 Full precision fits 62

5.1.2 Many set of constants for improved efficiency . 68

5.1.3 Handling of 5/5 tracks 69

5.2 Hardware structure 69

5.3 Input and output 73

5.3.1 Input data stream 73

5.3.2 Output data stream 73

5.4 Internal structure and algorithm 75

5.4.1 The merger module 76

5.4.2 The track processing module 77

5.4.3 Debug features 83

5.5 Parasitic mode for GF studies 85

6 GigaFitter performances 89

6.1 Timing 89

6.2 Efficiency studies 94

6.2.1 GigaFitter performances with current SVT data banks 96

6.2.2 GigaFitter performances with new SVT data banks 97

6.2.2.1 Recovering tracks that cross mechanical barrels 97

6.2.2.2 Ordered AM++ output 105

7 Conclusions 111

References 115

List of Figures

2.1 Fermilab accelerators 9
2.2 Tevatron peak luminosity and CDF Trigger upgrades 10
2.3 LHC Cross sections of various signals 12
2.4 The CDF Experiment 16
2.5 CDF Subdetectors 16
2.6 SVX Wedges 17
2.7 SVX Barrels 18
2.8 Three-level Trigger at CDF 19
2.9 CDF L1 Electrons 21
2.10 CDF L1 Muons 21
2.11 Muon trigger efficiency 23
2.12 Muon trigger efficiency with tracking 24
2.13 CDF online D^0 mass peak 25
2.14 $B^0 \to hh$ 26
2.15 Selections for $B^0 \to hh$ analysis 27
2.16 B_s oscillation in CDF and DZero 28
2.17 $Z \to b\bar{b}$ at CDF 29
2.18 B-jet trigger cross section 30

3.1 Unidimensional non-linear manifold in $\mathfrak{R}^2$ 1 35
3.2 Unidimensional non-linear manifold in $\mathfrak{R}^2$ 2 37
3.3 Unidimensional non-linear manifold in $\mathfrak{R}^2$ 3 39

4.1 SVT Scheme 43

4.2 The SVT racks 45
4.3 SVT Dataflow and Hardware scheme 47
4.4 SVT Impact parameter vs ϕ 50
4.5 CDF beam profile: SVT and offline 52
4.6 SVT impact parameter resolution 52
4.7 SVT efficiency in 2003 53
4.8 D^0 yield gain after 4/5 introduction 55
4.9 SVT Timing tails and upgrades 57

5.1 GF vs TF++ differences: χ^2 64
5.2 GF vs TF++ differences: $d0$ 65
5.3 GF vs TF++ differences: c 66
5.4 GF vs TF++ differences: ϕ 67
5.5 The Pulsar board 70
5.6 The GigaFitter mezzanine 70
5.7 The GigaFitter system installed 72
5.8 GF Pulsar Scheme 75
5.9 GF Mezzanine Scheme 76
5.10 GF Fitter module 77
5.11 GF Combiner module 78
5.12 GF DSP Fitter unit 81
5.13 GF Comparator unit 82
5.14 GF Spy Buffers 84

6.1 Global SVT Timing: GF and TF++ 91
6.2 SVT Timing vs Number of hit combinations 92
6.3 SVT Timing vs Number of fits 94
6.4 SVT (GF, TF++, standard data banks) Efficiency and fake rate at high luminosity 97
6.5 SVT (GF, TF++, standard data banks) Efficiency: impact parameter and p_T 98
6.6 SVT Efficiency vs cot(θ) with standard banks 99
6.7 SVT Efficiency vs cot(θ) comparison with new 433346 banks 100

6.8 SVT (GF, TF++, 544446 data banks) Efficiency and fake rate at high luminosity 101
6.9 SVT (GF, TF++, 544446 data banks) Efficiency: impact parameter and p_T 102
6.10 SVT Efficiency vs $\cot(\theta)$ comparison with new 544446 banks 103
6.11 Number of combinations per road with standard banks and 544446 banks 104
6.12 Maximum number of roads per wedge with standard banks and 544446 banks 107
6.13 Maximum number of roads per wedge with standard banks and 544446 banks and ordered AM++ 108

LIST OF FIGURES

List of Tables

5.1 HitBuffer++ to GigaFitter packet format 74
5.2 GigaFitter to GhostBuster packet format 74

6.1 SVT Average efficiency and fake rates with GF and TF++ 98

LIST OF TABLES

1

Overview

This thesis concerns the GigaFitter upgrade for the Silicon Vertex Trigger (SVT), the online tracking processor in the Collider Detector at Fermilab (CDF) experiment.

The GigaFitter is a track fitter of new generation, designed to replace the old SVT track fitters and to enhance the tracking processor capabilities. The reduction in fitting time by two orders of magnitude will amply enable CDF to continue to take data with high trigger efficiency for the reminder of Tevatron operations.

The GigaFitter is able to perform more than one fit per nanosecond, with a resolution nearly as good as that achievable offline.
It has been just commissioned in CDF and its computational power is available in order to provide:

1. A better SVT efficiency (i.e. larger signal yields) and more stable performances in response to the increasing instant luminosity, thanks to shorter execution times;
2. A better SVT acceptance, thanks to a greater capability to cover phase space regions;

3. An improved hardware reliability and easier maintenance, thanks to a considerable reduction in the number of boards (15 to 1) and board-to-board connections.

Better SVT efficiency The GigaFitter allows the reconstruction of those tracks formerly discarded by the fitters because of hardware limitations. This provides SVT with increased efficiency.

In particular, slightly different but alternative hit combinations are all fit at once, with no additional latency, in order to determine the best choice; while the former processors randomly picked one combination only.
This optimization becomes substantial at the highest Tevatron collider instant luminosity, because of the increased combinatorial noise, effectively opposing the SVT efficiency and impact-parameter resolution degradations due to high detector occupancy.

Better SVT acceptance The overcoming of previous hardware limitations allows three significant extensions of the tracking phase-space coverage, on both coordinates and momenta.

- Extending the SVT high-quality tracking to the forward-rapidity region will expand the lepton trigger coverage into that region.
- Extending the SVT acceptance lower-limit on transverse momenta from 2 GeV/c down to 1.5 GeV/c will significantly improve the online b-tagging capability.
- Extending the SVT acceptance upper-limit on impact parameters from 1.5 mm up to 3 mm will substantially improve the lifetime measurements.

As a final remark, the GigaFitter has been developed with a possible application to the Large Hadron Collider (LHC) experiments in mind. It is an essential ingredient in developing a hardware track trigger in that environment, where the luminosity will be two orders of magnitude higher than at Fermilab.
Tracking will be essential for virtually all triggers.
Separating b quarks or τ leptons from the enormous QCD background requires tracking: a secondary vertex identifying metastable b hadrons, one or three tracks in a very narrow cone from the hadronic decay of a τ.
Even the selection of high energy electrons and muons will rely much more heavily on tracking, since the calorimeter isolation is made ineffective by the energy deposition of overlapping collisions in the beam-crossing (pile-up).
The GigaFitter performances established in this thesis are requisite for LHC track triggers.

Chapter 2, **Hadron Collider experiments and Trigger**, highlights the problem of online event selection at hadron collider experiments and its importance for the experiment physics outreach. Section 2.1 describes the multi-level trigger approach and its implementation at the CDF experiment. Section 2.2 follows with examples of actual track-based triggers and their impact on physics.

In chapter 3, **Tracking in High Energy Physics**, the central argument is the problem of reconstructing particle trajectories in a tracking detector, in particular the challenge to perform this task with trigger timing performances. In section 3.1 I explain the SVT algorithm from a theoretical point of view.

I describe in chapter 4 the **Silicon Vertex Trigger** actual implementation, the complex hardware processor for the CDF experiment level 2 trigger. Section 4.1 shows the design ideas and the performances (reconstruction quality and timing) and their enhancement through a series of upgrades; the CDF upgrade program has been essential in order to allow the SVT processor to continue its online task, despite the event complexity increase due to the accelerator performances continuously improving. It also describes the diagnostic system and debug features used during development and commissioning of the processor and its upgrades.

The object of my thesis work, the GigaFitter, is described in chapter 5 (**GigaFitter**). It is the latest upgrade for the SVT processor, a new generation hardware processor for the SVT track fitting. I have worked on every phase of this project, from the design to the commissioning, coordinating a small group of three physicists and one engineer.
Section 5.1 outlines the design features and the improvements over the previous track fitters. In the sections 5.2 and 5.3 is the GigaFitter hardware structure: three powerful FPGAs on mezzanines mounted on a standard motherboard, the Pulsar, provided of other three older FPGAs. I have worked on the validation of the first prototype and all the hardware tests of the final system.
Section 5.4 shows the logical structure of the GigaFitter. The building blocks are distributed over the six interconnected FPGAs using three different firmwares (one for the mezzanine FPGA, one for the two Pulsar FPGAs connected to the mezzanines and one for the Pulsar FPGA connected to the output and VME backplane). I have developed all the firmware and all the tools necessary to validate and debug them.
Section 5.5 describes the careful test procedures used to validate the system from the prototype to the installation for commissioning. I have performed, coordinating other 3 young physicists, all of these tests: the stand alone tests in Pisa and Fermilab, the installation of the final system for parasitic data taking and the planning for commissioning and decommissioning of the old system. I have

also developed the software code for the GigaFitter simulation and integration with the existing SVT debug system, CDF online monitoring and Run Control system.

Finally chapter 6, **GigaFitter performances**, reports the first measurements of the performances performed with the final system in parasitic mode. In section 6.1 I show the timing measurements comparing the old system and the GigaFitter.
The GigaFitter is still underused in this initial installation. Prospects for future SVT performances, reachable only when the new processor capabilities will be exploited by CDF, have been studied using the simulation.
Section 6.2 shows the results of efficiency and fake studies I performed with the GigaFitter simulation, exploring new tuning possibilities for the system and showing how it's possible to gain in efficiency and acceptance thanks to the GigaFitter upgrade.

In chapter 7, **Conclusions**, there is a brief summary of what has been shown in this thesis, the main described topics and the obtained results.

1. OVERVIEW

2

Hadron Collider experiments and Trigger

Experiments in hadron collider high energy physics have grown to be very ambitious in recent years. The upgraded Tevatron at Fermilab and even more the upcoming Large Hadron Collider (LHC) at CERN are in an excellent position to give conclusive answers to many open questions of fundamental physics: for example the existence of supersymmetric particles, of the Higgs boson and the origin of the CP asymmetry. To reach these goals the new experiments deal with very high energy collisions, very high event rates and extremely precise and huge detectors.

Along with the development of new accelerators and detectors also the algorithms and processors to analyze the collected data and extract useful information need to evolve and become more powerful. The offline problem has required the birth of huge computing centers, the development of new world-wide computer networks and very advanced software. Web and GRID systems have been created for this challenging task. The online problem is even more complex and a particular effort must be put into it: the most interesting processes are very rare and hidden in extremely large levels of background, but only a small fraction of produced events can be recorded on tape for analysis. The online selection of events to be written on tape must be very clever and powerful to fully exploit the potential of new experiments.

The complex set of systems that analyze the data coming from the detector, extracts the useful information and makes the decision about whether or not to write that event on tape is called the "trigger". A very important part of the trigger is the one that reconstructs charged particles trajectories in the tracking detector: with this knowledge it is possible to make very sophisticated and powerful selections. The information from the tracking detector is often not used at its best at trigger level because of the big amount of data to process (the tracking detector is usually the one that produces most of the data) and the difficult task of trajectory reconstruction. Modern hardware along with clever algorithms allows us to fully exploit tracks in the trigger environment.

The SVT processor at CDF has been a pioneer in this field. The CDF experiment is located at the Tevatron accelerator (figure 2.1) at Fermilab (Batavia, IL, USA). The Tevatron is a proton-antiproton collider with a center of mass energy of 1.96 TeV and reached peak luminosity up to 3.5×10^{32} cm^{-2}s^{-1} as of 2009. SVT was installed in 2000 providing for the first time at an hadron collider experiment offline quality tracks to the trigger decision algorithm.

In general dedicated hardware is considered powerful but usually difficult to upgrade and not flexible. SVT has instead proven that a

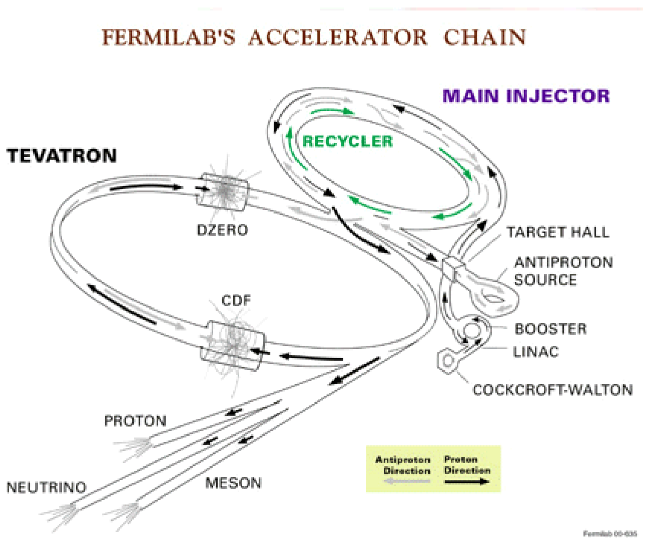


Figure 2.1: Fermilab accelerators - A simple scheme of the accelerator chain at Fermilab.

properly designed hardware can also be flexible enough to prospect upgrades to enhance further its capabilities and to cope with increasing detector occupancy. In 2003 when the Tevatron performances started to improve SVT showed the need of more powerful computation capabilities. No upgrade plan was in CDF. It could happen that the difficulties of an unpredicted upgrade could make void the effort of building SVT, that was quickly becoming obsolete. However the high degree of organization and standardization inside the CDF trigger and SVT system allowed a very quick upgrade, even if the function performed by SVT is very complex. SVT was upgraded into just 2 years, exploiting the experience of previous scheduled CDF upgrades: the Pulsar design (13) and the Global L2 upgrade. The Pulsar board, an FPGA based general purpose board was heavily used to implement all the SVT functions. The figure 2.2 shows the Tevatron instant luminosity grow and the correlated CDF actions to keep the trigger efficient. The SVT upgrade commissioning took place in the summer 2005 while the experiment was taking data. A phased installation was chosen: boards were replaced

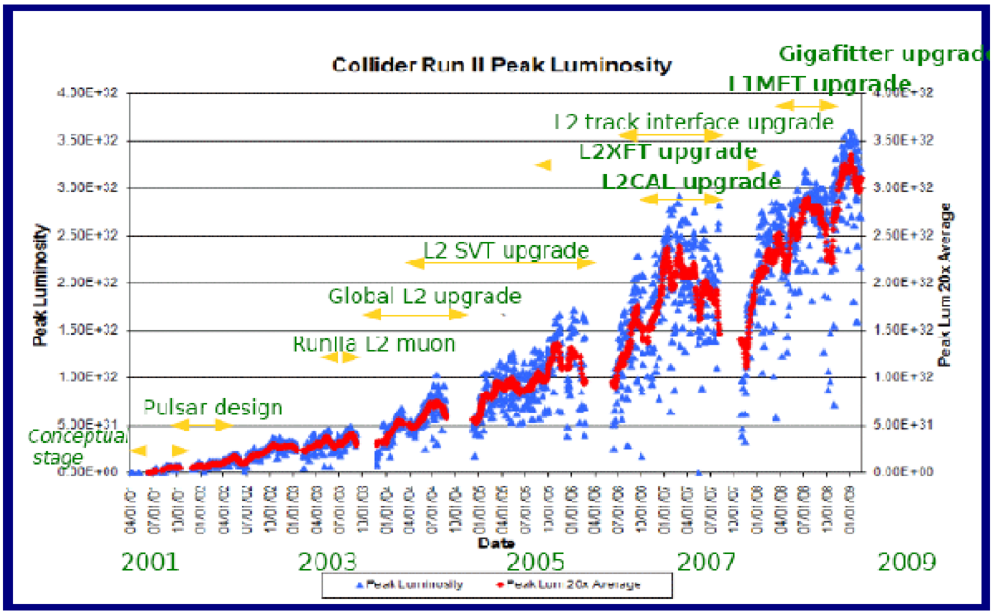


Figure 2.2: Tevatron peak luminosity and CDF Trigger upgrades - This figure shows the various upgrades programs that CDF made at the trigger system to adapt to increasing luminosity. The performance of the accelerator has steadily increased over time and it's foreseen to be able to beat the 3.5×10^32 cm^{-2}s^{-1} record before the end of operations in 2011/2012. Many of these upgrades were unpredicted and exploited the experience and method used during the successful SVT upgrade of 2006.

gradually, exploiting the short time between stores[1]. This phased procedure allowed for quick recovery if there were failures, since each small change was immediately checked before going ahead.
The power added to the experiment without any risk for the data taking convinced the collaboration to proceed with other important unpredicted trigger hardware upgrades, to fix problems caused to the trigger by the increasing Tevatron performances (shown in figure 2.2). The very last upgrade is again for SVT and it is the GigaFitter, the object of this thesis.
The GigaFitter will allow the SVT processor to deal with the increased luminosity of the Tevatron collider and also gives a prospective of what is possible at more challenging experiments such as at LHC.

2.1 Online event selection

Developing algorithms for online selection of events is a crucial step to fully exploit potential of new experiments.

At CDF the collision rate is about 2 MHz and at an instantaneous luminosity of 3×10^{32} cm^{-2}s^{-1} the average number of interactions per bunch crossing (pileup) is 6 (3) (396 ns bunch spacing). The rate at which events can be written on tape is about 100/s.

At the new LHC experiments the problem is even more harsh: the collision rate is 40 MHz and at an instantaneous luminosity of 10^{34} cm^{-2}s^{-1} there are 25 pileup interactions, while the rate at which events can be written on tape is still about 100/s (the storage

[1] A store is the period when the Tevatron accelerator is making collisions for High Energy Physics experiments. Its duration depends on initial luminosity of the store and accelerator status. During the SVT upgrade was about a day with few hours between stores, now it's typically 10-12 hours with 1-2 hours between stores.

technology is much faster at LHC but events are bigger).

The trigger system must perform a very stringent selection, reducing the rate of events of several orders of magnitude. This selection must be as sophisticate as possible in order to suppress background events while saving signal events.

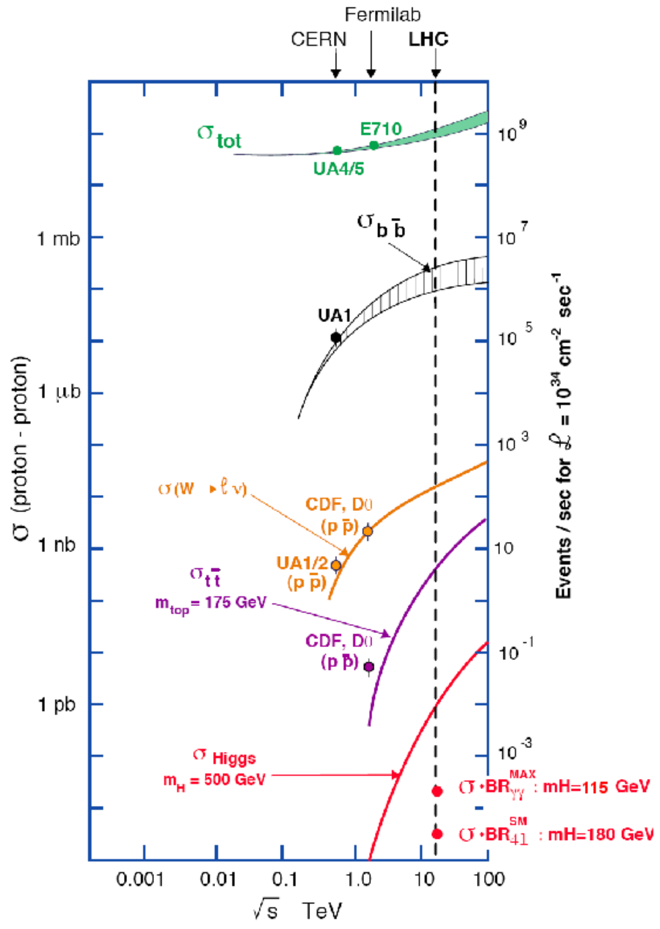


Figure 2.3: LHC Cross sections of various signals - It's shown the expected rate of events and the relative cross sections between various kind of signals and background events at the LHC baseline luminosity of 10^{34} $\mathrm{cm}^{-2}\mathrm{s}^{-1}$

To understand how critical is this task we can look at the figure

2.3: the total rate of produced events at LHC baseline luminosity is about 10^9 every second and only 100/s can be written on tape. However we must be sure to write among them a possible Standard Model (SM) Higgs of 115 GeV decaying in two photons that is produced roughly every hour, the even rarest SM Higgs of 180 GeV decaying in four leptons and so on. The trigger must be able to select a rich variety of interesting but extremely rare events, each one with its peculiar detector response, but be able to reject the overwhelming amount of uninteresting events.

The physics outreach of the experiment is determined by the trigger capabilities as much as by the accelerator and detector performances: producing an hypothetical 500 GeV SUSY Higgs every minute is useful only if the experiment is able to select and store that event with an high efficiency. If the trigger is inefficient, for example only 1% of such events are selected, the experiment is equivalent to another one that can exploit only a hundredth of the luminosity but with a better, full efficient trigger.

2.1.1 Multi-level trigger

Hadron collider experiments are made by many different subdetectors, the tracking detector being one of them, and each subdetector has its own data channels, it's own time response and readout bandwidth. Not all subdetectors can be read at every collision. At CDF, for example, the silicon tracker can be read at a maximum rate of 30 kHz without damaging the sensors and causing deadtime[1] to the experiment.

[1] "deadtime" is the technical jargon to call the period when experiment has data, but the data acquisition system is not ready and the data is lost.

Furthermore the algorithms to extract useful information from sampled data have a wide range of timing and complexity: finding global calorimetric parameters (sum of all transverse energies, missing transverse energy, for example) is very fast, finding jets (clusters of energy in calorimeter) is slower like finding tracks with offline quality. Also the trigger decision algorithm, that apply cuts on the parameters reconstructed by the various trigger processors, might be of a wide range of complexity and timing.

In this context it is not convenient to use all processors at one time on the same event, because it would be always necessary to wait for the slowest, and apply an efficient but complex and slow decision. This strategy would lead to a certain amount of time where collision would happen but the system would be busy and the data would be lost.

It is much more convenient to group processors based on their bandwidth and latency, then organize the trigger in a pipelined multi level scheme: at the first level the fastest algorithms are executed and a first decision is taken reducing the input rate that has to be analyzed by the slower processors at level 2. At the second level the second fastest algorithms are executed on data collected by the first level and a second decision is taken and so on. This scheme allows to employ complex algorithms that otherwise would generate deadtime at later levels characterized by lower input rates. The amount of data that needs to be buffered before the final decision is also minimized.

This strategy suggests to put slower processors at high levels of the trigger, but for the sake of collecting high purity data it's mandatory to be able to do sophisticate selections from the first levels of trigger. The solution is to employ powerful dedicated processors in order to make complex and precise algorithms fast enough to be put in the first levels of trigger. This is the strategy that CDF has followed for triggers based on reconstructed tracks, pushing tracking processors at the first two levels of trigger and allowing collection of high quality data.

2.1.2 Trigger at CDF

2.1.2.1 The CDF experiment

The CDF detector has the typical structure of a collider experiment: many sensors disposed in an "onion"-like structure starting from the interaction point as shown in figure 2.4. The inner detector is the tracker made by internal barrels equipped with silicon double-face microstrip sensors (it is subdivided in three subdetectors starting from interaction point: L00, SVX and ISL) followed by a multiwire drift chamber (COT). The tracking detector is inside a superconducting solenoid magnet. After the magnet there are the calorimeters: preshower, electromagnetic calorimeter and hadronic calorimeter. The outermost detectors are the muon detecting systems. A full description is found in (4).

Figure 2.5 shows one quadrant of the longitudinal section of the CDF tracking system. The outermost detector is the Central Outer Tracker (COT) drift chamber. The COT provides full coverage for $|\eta| < 1$, with an excellent curvature resolution of 0.15 p_T(GeV)%. The COT is the core of the integrated CDF tracking system. The COT provides 3-dimensional track reconstruction with 96 detector layers. The 96 layers are organized into 8 super-layers. Four of which are axial, while the others, called stereo, are at small angles.

Inside the COT there are the silicon detectors: SVX II, ISL and L00. They are complementary to the COT. They provide an excellent transverse impact parameter resolution of 27 μm.The silicon detectors provide 3-dimensional track reconstruction. The achieved longitudinal impact parameter resolution is 70 μm. Figure 2.6 shows a cross section of the SVX II. The SVX II is organized into 12 azimuthal wedges. For each wedge there are 5 detector layers each

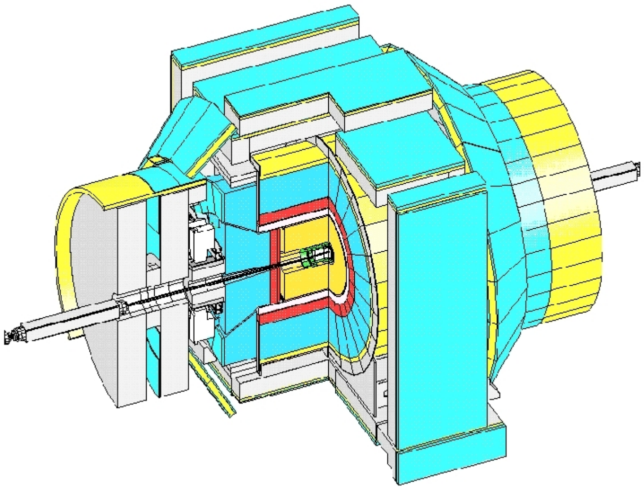


Figure 2.4: The CDF Experiment - An isometric drawing of the CDF experiment. Different colors highlights the various subdetector and components of the experiment. The innermost green subdetectors are the silicon microvertex detectors (ISL, SVX and L00).

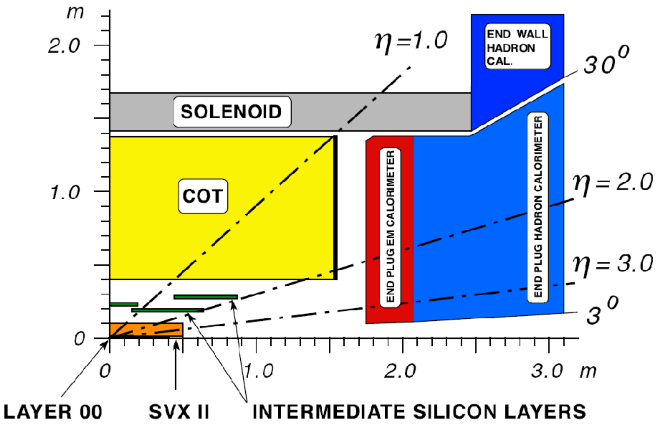


Figure 2.5: CDF Subdetectors - A schematic r-z view of one half of the CDF experiment. η coverage of the various silicon tracking subsystems is highlighted.

providing one axial measurement on one face of the silicon sensor and a 900 or small angle measurement on the other face. Figure 2.7 shows an isometric view of the SVX II. The SVX II is made of three mechanical barrels. Each mechanical barrel is made of two electrical barrels. In fact, within a mechanical barrel each detector element is built of two silicon sensors with independent readout paths. The two sensors are aligned longitudinally to achieve a total length of 29 cm, which is the length of each mechanical barrel. Hence, for each wedge and for each layer there are a total of 6 sensors belonging to 3 different mechanical barrels.

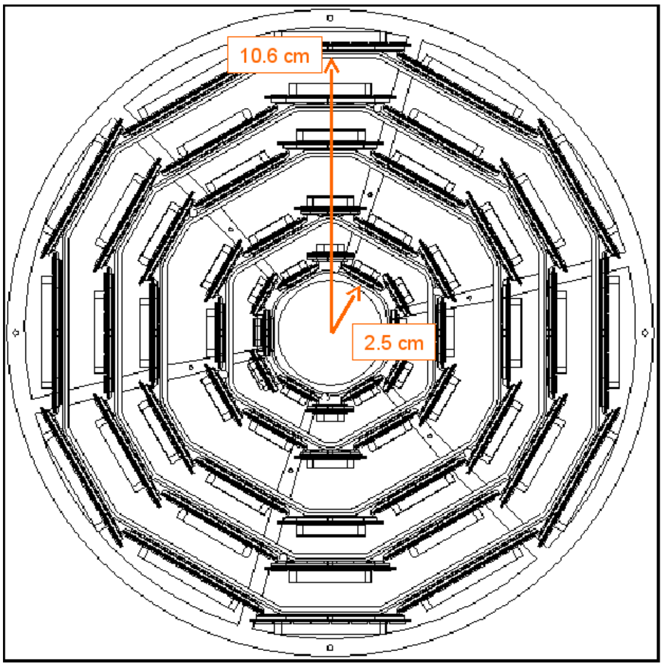


Figure 2.6: SVX Wedges - Each SVX barrel is made by five layers and on the r-ϕ plane is subdivided in 12 slices wide 30° ϕ called wedges

The L00 and ISL silicon detectors complete the silicon subsystems. The L00 detector, which is directly mounted on the beam pipe, provides best impact parameter resolution. The ISL detector provides up to two additional tracking layers, depending on track

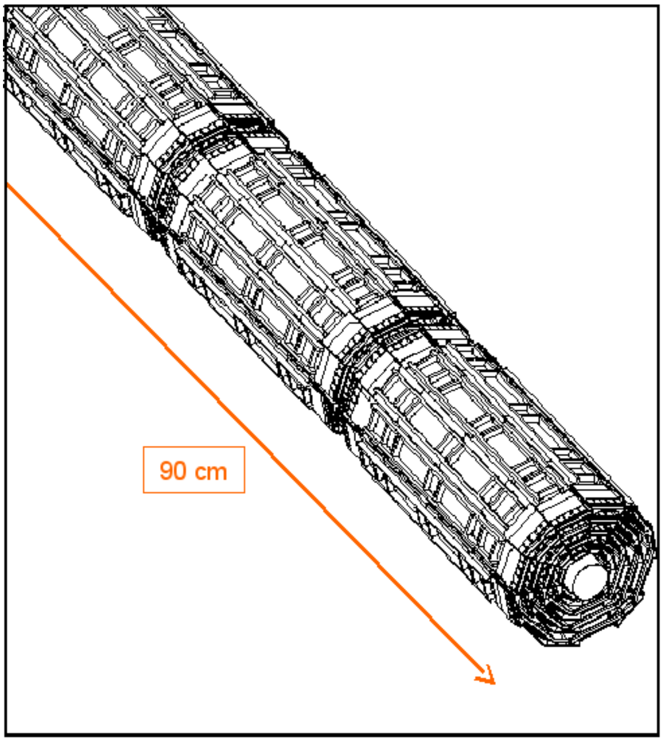


Figure 2.7: SVX Barrels - SVX is made by three separate barrels (called mechanical barrels) of five layers detector in the z direction. Each barrel is made by two bonded barrels (called electrical barrels).

pseudo-rapidity, that allow standalone silicon tracking. In particular, ISL allows to extend tracking beyond the COT limit ($|\eta| < 1$), and up to $|\eta| < 2$. The L00 and ISL detectors are not used by the SVT.

2.1.2.2 The CDF three-level trigger

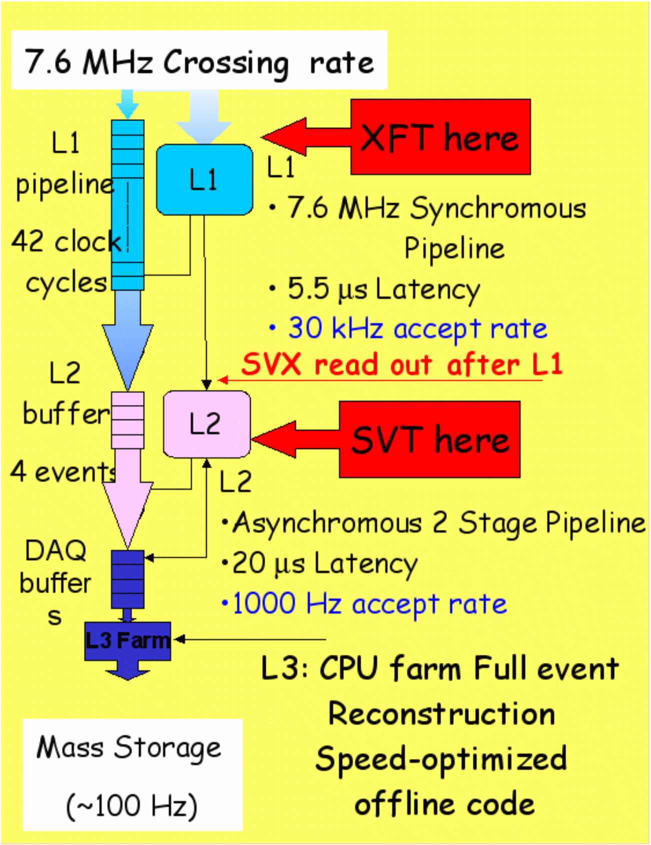


Figure 2.8: Three-level Trigger at CDF - The three levels of the CDF trigger and their bandwidth. In evidence the tracking processors: XFT (1st level) and SVT (2nd level).

In figure 2.8 is shown the three-level structure of the CDF trigger. The first level is a synchronous pipeline of dedicated hardware processors receiving data at the collision rate of 7.6 MHz and reducing the rate up to 30 kHz in 5.5 μs of latency. The second level is asynchronous, made by dedicated processors and a final decision commercial CPU. It process the data selected by the first level and makes a decision with an average latency of 20 μs reducing the rate up to 1 kHz. There are only four event buffers at level 2 (L2), so it's mandatory for all L2 processors to have not only the processing time with compatible average, but also short tails to avoid deadtime. The third level is a CPU farm that execute an optimized version of the offline reconstruction algorithms. The third level reduces the rate of events to 100 Hz for permanent storage.

2.2 Tracks in trigger: examples

It's worth to notice that in CDF a tracking processor is present starting from the first level of trigger. In fact at level one there is the XFT processor (3) for reconstruction of transverse trajectories segments in the COT chamber. Moreover at the second level there is the SVT processor for offline quality reconstruction using SVX and XFT tracks. XFT is also used at level 2 for 3D confirmation of previously 2D reconstructed segments.

2.2.1 Electron and muons in CDF

In CDF the use of tracks at level 1 has been extremely important for lepton identification.

Figure 2.9 shows that 8 GeV electrons can be selected at CDF occupying a very small part of the whole level 1 bandwidth (0.180 kHz of a total of 30 kHz). The rate is proportional to the instant luminosity, as happens for pure physics samples where the fakes are negligible. CDF rates are determined by the coincidence of the electromagnetic deposits (ECAL) with the XFT track. Using only the ECAL does not distinguish between electrons and photons. The large background of photons from the π^0 decay would produce large rates and would force the experiment to set much higher L1 thresholds. The L1 coincidence between the EM cluster and a track segment distinguishes the electrons from the large π^0 background and reduce significantly the L1 thresholds.

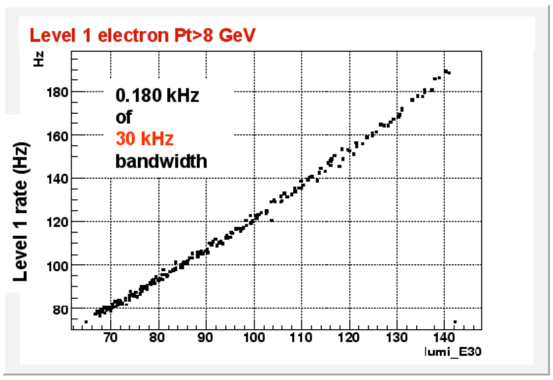


Figure 2.9: CDF L1 Electrons - Level 1 rate for trigger selection of electrons with $p_T > 8$ GeV. The occupied bandwidth is a very small fraction (0.180 kHz) of the total available bandwidth (30 kHz).

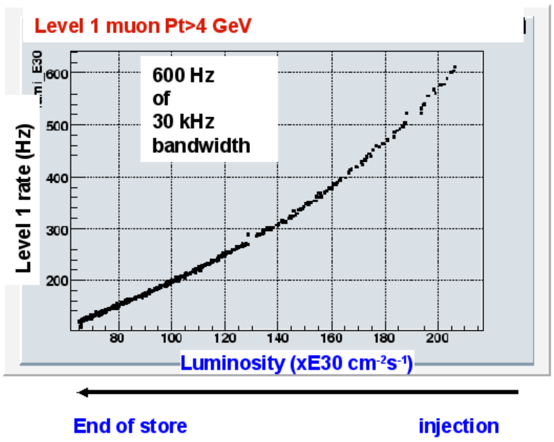


Figure 2.10: CDF L1 Muons - Level 1 rate for trigger selection of muons with $p_T > 4$ GeV. The occupied bandwidth is a very small fraction (0.180 kHz) of the total available bandwidth (30 kHz).

Figure 2.10 shows that also soft muons (4 GeV) can be selected occupying a small trigger bandwidth. The L1 muon trigger is first of all based on the measurement in the outer muon chambers. However the momentum resolution is lower compared to the capability of the tracking detector, being limited by multiple scattering, and the measurement of the impact parameter is very poor. Thus, a trigger mainly thought to select prompt muons from boson decays based only on muon chamber measurements suffers of many backgrounds: (a) promoted muons: real, low-momentum muons (mostly from b or c-decays), which are mis-measured and appear to have a p_T above threshold. (b) muons that are product decays of pions and kaons, (c) fake muons due to noise correlated to beam not screened because the muon detector is external with respect the calorimeter. Early combination of the muon trigger chambers with precision measurements in the inner tracker drastically reduces the background rate.

2.2.2 High p_T lepton isolation in ATLAS

Another important area where track abilities should be recovered is high Pt lepton isolation. We observe in CDF the decreasing efficiency of calorimeter-based isolation algorithms due to the pileup increase. The track ability to determine the high-PT primary vertex will help identify isolated electrons and muons. This would include the advantage at high luminosity of basing isolation on tracks above a PT threshold that point to the high-PT primary vertex. We have compared the calorimeter and track based isolations at LHC where the pile-up problem will be extremely more relevant.
Figure 2.11 shows that the isolation based on electromagnetic energy (red points) measured at Atlas around the lepton causes strong inefficiency on the 20 GeV muon already with a little number of pile-up events. The black points show the efficiency of a selection that

does not apply any isolation criteria.

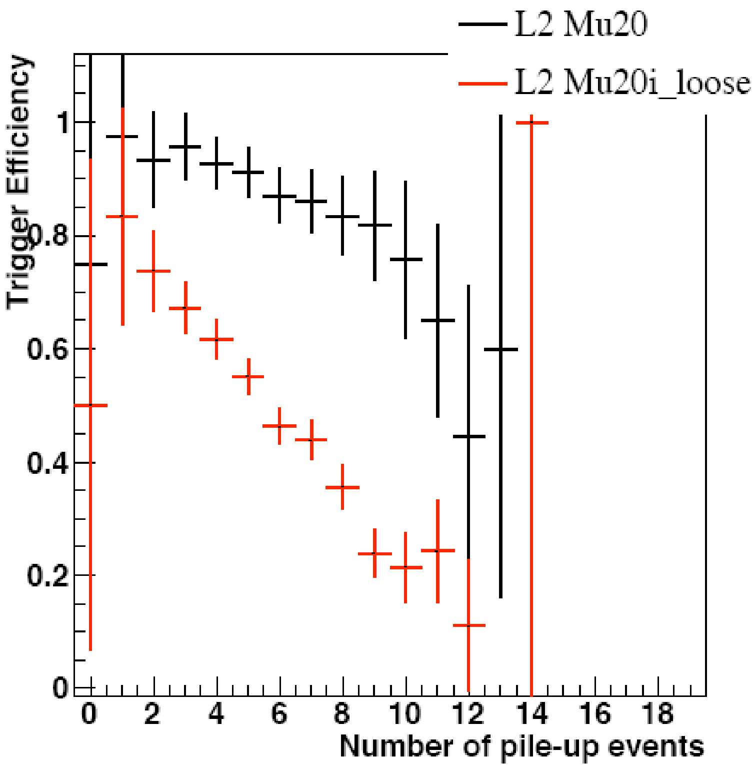


Figure 2.11: Muon trigger efficiency - The figure shows the efficiency versus the number of pile up events of two muon triggers, one with isolation (red) and one without (black), for the Atlas experiment at LHC. The trigger with isolation suffers from the increase of the number of pile-up events.

Figure 2.12 shows that applying a threshold over the Pt sum over all tracks above 1 GeV and inside a cone around the lepton (black points) produces a better result. Requiring the z0 of all tracks in the cone within 10mm of the muon track z0 (red points) produces a perfect result.

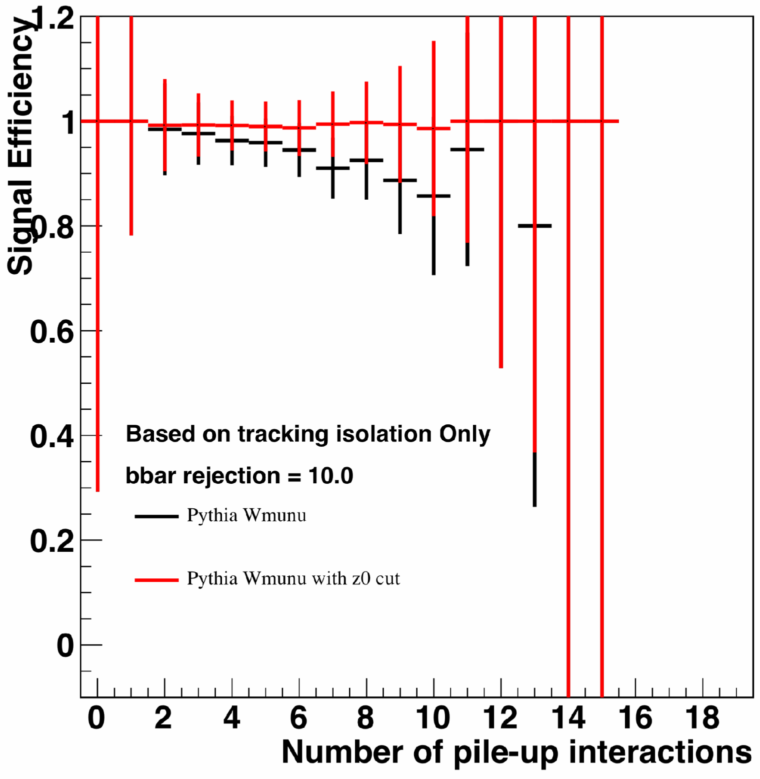


Figure 2.12: Muon trigger efficiency with tracking - The figure shows the efficiency versus the number of pile up events of two muon triggers plus isolation done with inner tracker for the Atlas experiment at LHC. The black points are a cone-based isolation, while the red points are obtained also requiring z0 match with muon track.

2.2.3 Two-Track Trigger

The most revolutionary use of tracks ever seen at an hadron collider, at both level 1 and level 2 is certainly the CDF Two-Track Trigger (TTT). It works on tracks above 1.5-2 GeV. Figure 2.13 shows the Two Track Trigger power.

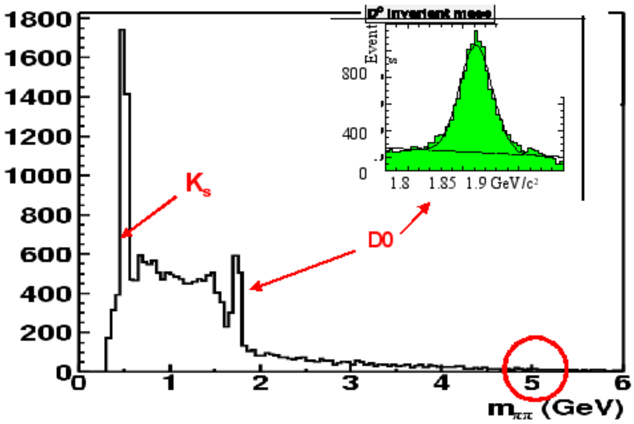


Figure 2.13: CDF online D^0 mass peak - Using data collected with Two Track Trigger (SVT) the D^0 mass peak can be used online to monitor the trigger.

It shows the online invariant mass distribution of track pairs with large impact parameter. We monitor the SVT efficiency run by run with the online reconstructed D^0 signal. The 5 GeV/c^2 region shows a very low background level.

With 1 fb^{-1} of data CDF has reconstructed a striking $B^0 \rightarrow hh$ signal (figure 2.14) (18): an excellent example of the concrete possibility of reconstructing rare and ”background-looking” signals, when a high-performance trigger and a sophisticated offline analysis are combined.

The plot in figure 2.15 is very interesting and shows how much background would cover the Ks, D0 and B peaks if the CDF tracking detectors were not used for the trigger selection. The plot shows the

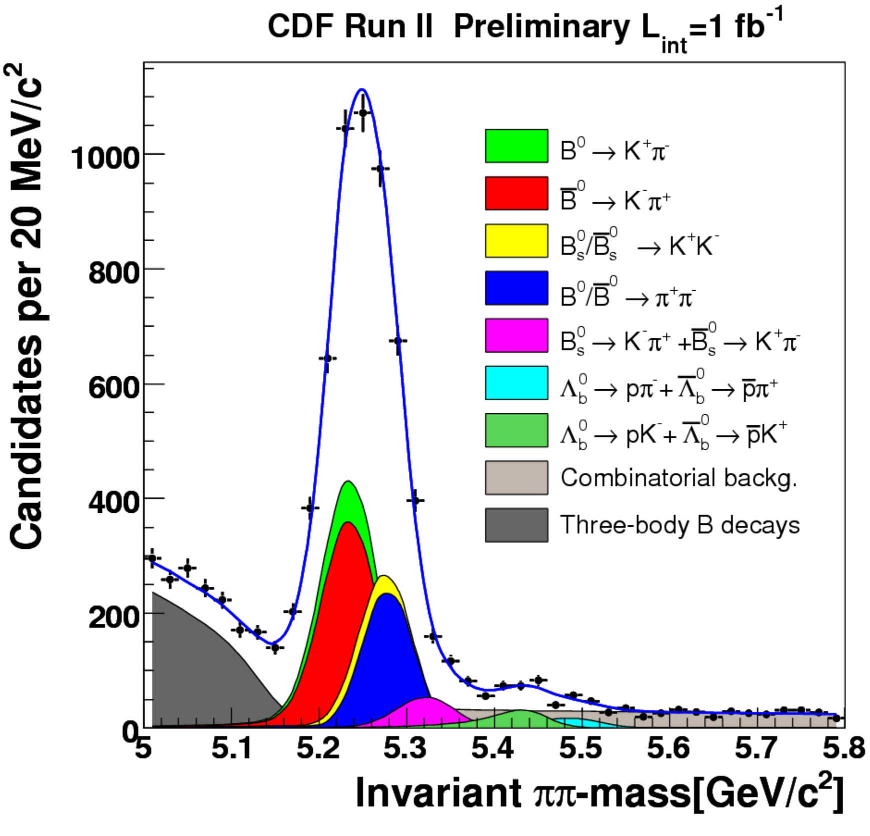


Figure 2.14: $B^0 \to hh$ - The figure shows the $\pi\pi$ invariant mass distribution with reconstructed $B^0 \to hh$ decays. Data was collected using SVT based trigger.

background (blue, measured on data) and the $B^0 \to hh$ signal (red) cross section as a function of the applied selection criteria (17). The request of two XFT tracks at L1 and of two SVT tracks with large impact parameter at L2 reduces the level of background of several orders of magnitude, while keeping the efficiency on the $B^0 \to hh$ signal at a few percent level. The purity of the selected sample is enormously increased. Since the B-physics has a limited rate budget, the better purity allows CDF to increase by several orders of magnitude its efficiency for the hadronic B decay modes.

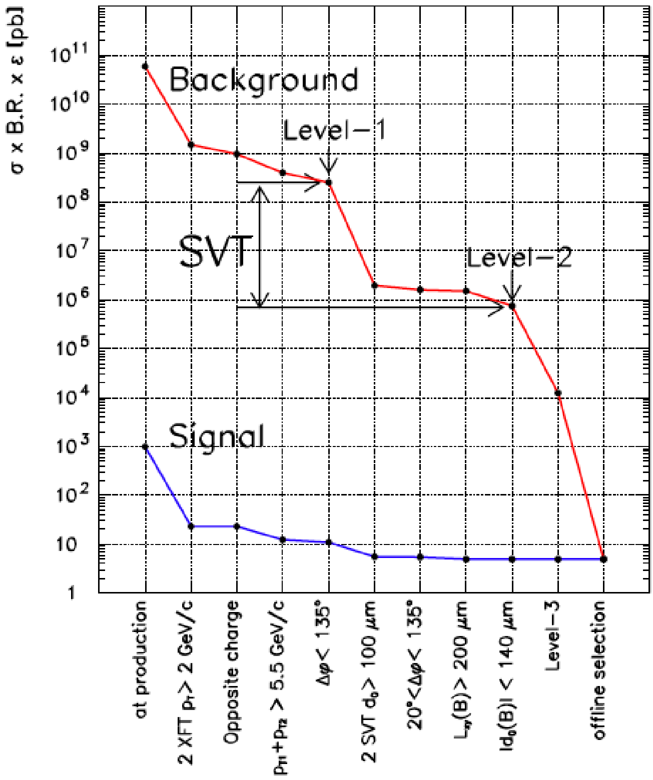


Figure 2.15: Selections for $B^0 \to hh$ analysis - The effect of the various selection cuts applied for the $B^0 \to hh$ analysis on both signal (red) and background (blue).

Historically, B-physics events have been selected at hadron colliders by triggers based on lepton identification. Trigger selections based on the reconstruction of secondary decay vertices increase the b-quark identification efficiency and allow collecting otherwise inaccessible hadronic decay modes. The availability of the hadronic decay modes at CDF determined the different quality of the CDF and D0 (15) Bs mixing measurements (see Figure 2.16).

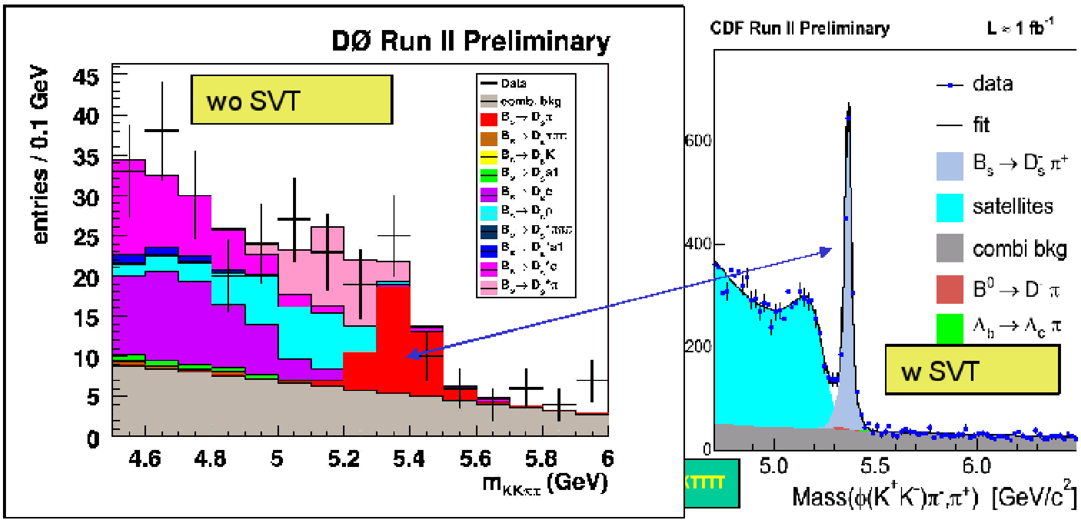


Figure 2.16: B_s oscillation in CDF and DZero - DZero and CDF, both Tevatron experiments, have published an analysis on B_s oscillation. CDF had the advantage of much more events collected by its trigger.

SVT had an extremely significant impact on the CDF physics program. It has been essential for the B_s mixing frequency measurement (1), and the first observation of the rare charmless decay modes of the B_s (2) and Λ_b which complement the existing "Beauty Factories" information on B_d charmless decays (10). These extremely challenging measurements and first observations would have been completely out of the CDF reach without the SVT. Severe constraints on several extension of the SM already arise from the new CDF measurements and will become even more stringent when the precision of the theoretical calculations will reach the current level of experimental accuracy.

2.2.4 High p_T triggers in CDF

The SVT IP trigger has been used to collect a sample reach of b-jets allowing the observation of an excess of $Z \to b\bar{b}$ events (figure 2.17).

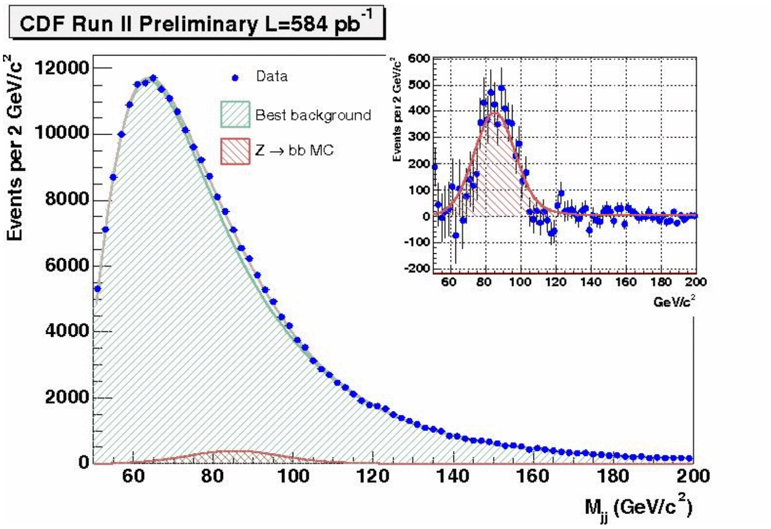


Figure 2.17: $Z \to b\bar{b}$ **at CDF** - Preliminary analysis of $Z \to b\bar{b}$ signal at CDF, data collected with SVT trigger.

The time made available to the final L2 CPU by the SVT upgrade allowed to reinforce the b-jet selection that initially was based, like for B-physics events, simply on the large IP requirement of one or two tracks. The b-jet trigger has been recently structured requiring:

- two towers with $E_T > 5$ GeV and two XFT tracks with $p_T > 1.5$ GeV at level1
- 2 central jets with $E_T > 15$ GeV, 2 tracks (seen by both SVT and XFT) matched to one jet, with $R_b > 0.1$ cm, $d_0 > 90$ μm for both tracks at level 2.

Figure 2.18 shows the trigger cross section before the improvement (green curve) and after (red-blue points) as a function of the instant luminosity. The green curve is stopped early because the fakes increase too much with the detector occupancy. The new more complex algorithm increases the sample purity and allows the use of the trigger at all Tevatron instant luminosities, occupying only 70 Hz (see the light blue lines indicating the points at constant L2 rate in the plot) at the maximum of $3 \cdot 10^{32}$ instant luminosity.

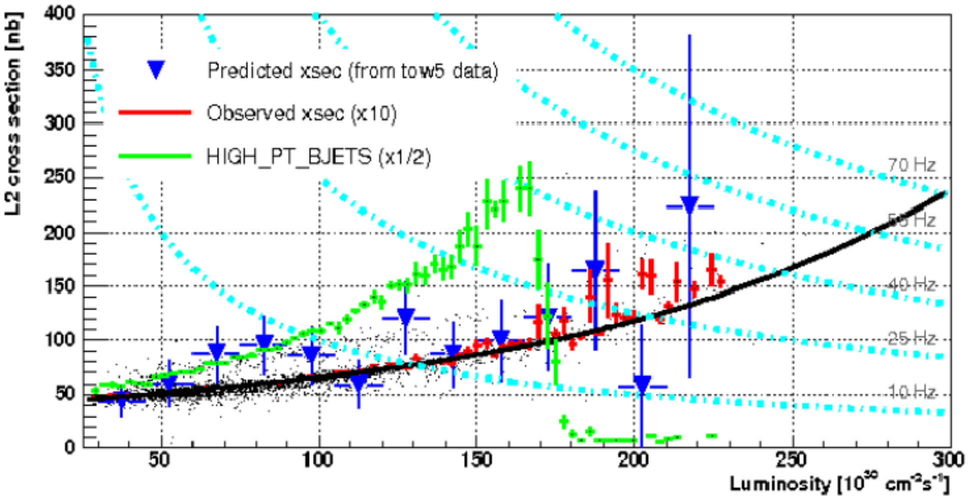


Figure 2.18: B-jet trigger cross section - The trigger cross section for the new SVT based b-jet trigger at CDF is shown (red and blue) with respect to the old trigger (green).

3

Tracking in High Energy Physics

Tracking algorithms reconstruct the trajectory of a charged particle going through a sub-detector of the experiment called tracker.

From the the trajectory it is possible to extract very useful informations for event selection. If the tracker is within a magnetic field, for example a solenoidal field oriented along the beam axis, it is possible to reconstruct the transverse momentum of the particle from the curvature of its trajectory. Reconstructing with great precision the track allows the primary vertex identification and also to find secondary decay vertices and thus identify events with particles with long life (b-quark, taus), extremely useful to select physics of interest.

Charged particle tracking is a very rich source of information, in fact it's a major technique in offline analysis where most sophisticated algorithms were developed.

To be able to develop an online tracking algorithm with timing performance suitable for trigger decision, but quality similar to offline tracking it's a very challenging task. To analyze the problem we must start from how the information is recorded in the tracker sub-detector.

Usually the tracker is composed by several detecting layers with known geometry where the crossing of the charged particles is observed in one or two coordinates on the layer. SVX, the inner silicon tracker of CDF, it's made of five cylinder layers of different radiuses, concentric with the beam axis. Each layer is equipped with double face microstrip detectors.

When a charged particle cross a tracking detector, for example the five-layer SVX detector, its passage is observed as five strip hits, one in each layer. The tracking algorithm must reconstruct the particle trajectory from the position of this five strips. This is not the only task of the tracking algorithm: if, for example, two particles cross the detector the signal will be ten strip hits, two in each layer, but there's no additional information that suggests which hit was produced by which particle. The tracking algorithm must also solve the combinatorial problem associating hits to candidate tracks, selecting the ones that are real tracks and rejecting the fakes.

The problem of associating hits to track candidates is as important as reconstructing trajectory parameters from the candidate itself. This is a very time consuming task for the tracking algorithm, especially in modern experiments where hundreds if not thousands of charged particles are present at the same time, together with the remnant signal of the previous collisions (pile-up) and a certain amount of random noise.

3.1 The SVT Algorithm

The SVT (Silicon Vertex Trigger) algorithm was developed to reconstruct the parameters curvature (c), impact parameter (d) and azimuthal angle (ϕ)[1] of the charged particles in the silicon detector of CDF, SVX. SVT is designed to provide reconstructed information to the Level 2 trigger decision algorithm.

The algorithm exploits the information coming from five layers of the SVX detector (using only one face of the double face layers) plus the parameters c and ϕ reconstructed by the XFT processors. XFT is a tracking processor for the Level 1 trigger of CDF which reconstruct trajectories of particles crossing the multiwire drift chamber (COT) that surrounds SVX and described here (25).

The SVT algorithm is however a very general approach to the tracking problem. I will use the real SVT in CDF to describe the algorithm details, however the arguments described here are easily extended to any tracking problem.

The algorithm is subdivided in two distinct phases: the first phase finds trajectories using low spatial resolution information, associating higher resolution strip hits to track candidates, the second phase does the combinatorics and finds high resolution track parameters fitting all the combinations inside each low resolution track candidate.

[1] Curvature is defined as $\frac{q}{2R}$ where q is the particle charge and R is the helix radius of the trajectory. The impact parameter is the minimum distance of the trajectory from the z axis (oriented as the magnetic field), defined as $d = |q| \cdot (\sqrt{x_0^2 + y_0^2} - R)$ where (x_0, y_0) is the nearest point of the trajectory to the origin of axis on the transverse plane. The angle ϕ is the direction of the particle on the transverse plane in the point (x_0, y_0).

3.1.1 Linear fit

The association of hits to a track candidate is usually a huge combinatorial problem, very time consuming, so an algorithm that want to solve it in time for trigger decision must find a clever and effective solution.

We start to look at this problem: how the combinations of hits coming from real charged particles differ with respect to the generic random combination?

If we look at the n-tuple of hits as a point in a n-dimensional space, where n is the number of detecting layers, and we eliminate every possible error of measurement or statistical physical effect we'll see that the points in n-dimension that are coming from real particles belong to a well defined m-dimensional manifold where m is the number of free parameters in the trajectory equation.

This means we can write $n - m$ equations of the hit coordinates $\vec{x}$, called constraint equations:

$$f_i(\vec{x}) = 0$$

For example in SVT at CDF we have a 6 dimensional space (4 coordinates from SVX, 2 from XFT) while the trajectory equation being constrained in the transverse plane has 3 free parameters (c, d, ϕ). The resulting manifold is 3-dimensional and we have three constraint equations. The simpler case of a unidimensional manifold in $\Re^2$ easy to be represented as a plot is shown in figure 3.1 (a) as a graphical example.

The trajectory parameters can also be a set of local coordinates on the manifold. We have to find a general method to identify n-tuples of coordinates produced by from real particles - they satisfy

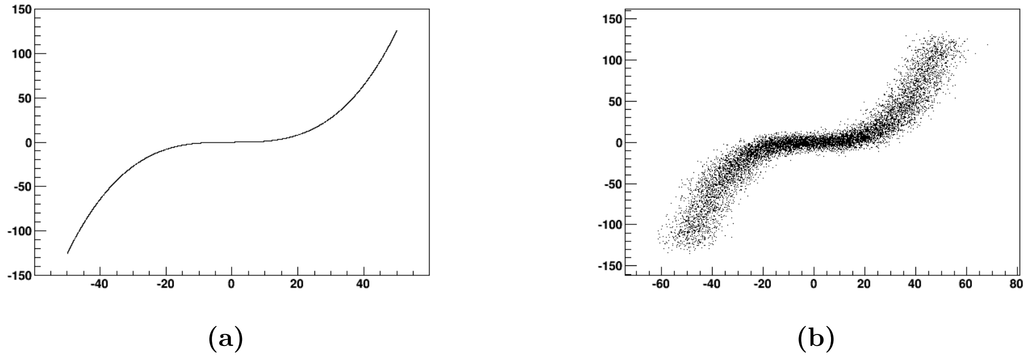


Figure 3.1: Unidimensional non-linear manifold in $\Re^2$: (a) without measurement errors and (b) with Gaussian noise on the two coordinates.

$f_i(\vec{x}) = 0$ - and a method to find track parameters - measuring the position of the n-tuple in the local coordinates of the manifold.

Up to now measurement errors or other physical effects of statistical nature like Bremsstrahlung, multiple scattering or delta rays were disregarded. If such effects are taken into account we'll have not a simple m-dimensional manifold but a probability volume around it. An example is shown in 3.1 (b).

Computing the covariance matrix F_{ij} of f_i is a way to quantify the characteristics of this volume. At first order is:

$$F_{ij} \simeq \frac{\partial f_i}{\partial x_k}\frac{\partial f_j}{\partial x_l}M_{kl}$$

Where M_{kl} is the covariance matrix of $\vec{x}$.

With F_{ij} it's possible to build a χ^2 function:

$$\chi^2 = \sum_{ij} f_i \cdot F_{ij}^{-1} \cdot f_j$$

This expression can be simplified writing new constraint equations $\widetilde{f}_i$ such as:

$$\widetilde{f}_i = \frac{S_{ij} f_j}{\sigma_i}$$

S_{ij} is found diagonalizing F_{ij}^{-1}:

$$F_{kl}^{-1} = S_{ik} \frac{\delta_{ij}}{\sigma_i} S_{il}$$

From which we can rewrite the above in a more compact form:

$$\chi^2 = \sum_i \widetilde{f}_i^2$$

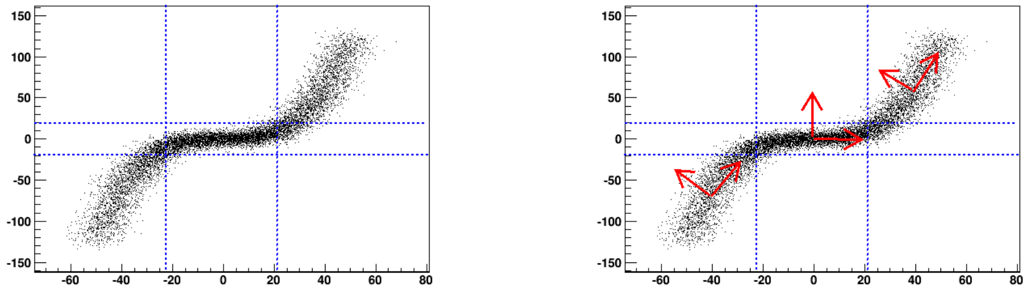
The χ^2 function is used as a quality function: a decision based on its value discriminates with the desired efficiency between $\vec{x}$ that corresponds to a real trajectory and the ones who doesn't.

In general it's not easy to compute the $\widetilde{f}_i$ and the charts for the fit, but we can exploit the fact that a differentiable manifold admit locally to define a tangent hyperplane and thus linearize the problem. We can find a series of n-dimensional hypercubes where apply the linear approximation and obtain an atlas of linear charts for all the manifold. In CDF we'll see that the geometry of the detector itself suggests how to find those regions. In figure 3.2 (a) is shown the procedure on the unidimensional manifold of figure 3.1.

This way we'll find for each one of those regions a set of constants $\vec{v}_i$, c_i, $\vec{w}_i$ and q_j such as the constraint equations became:

$$\tilde{f}_i(\vec{x}) = \vec{v_i} \cdot \vec{x} + c_i$$

$$p_j(\vec{x}) = \vec{w_j} \cdot \vec{x} + q_j$$



(a)

(b)

Figure 3.2: Unidimensional non-linear manifold in $\mathfrak{R}^2$: (a) areas where linear approximation is reasonable are defined and (b) principal axis are found using PCA. The parameter of interest is measured on the principal axis, while the orthogonal axis measure the probability that the point belong to the manifold.

From the knowledge of the equation of motion of the charged particle, of the detector geometry and of the statistical effects on measurements it is possible to find analytically an expression for such constants, but it's more practical to use the principal component analysis on a data sample (simulated or real) and find directly the constants.

The principal component analysis (PCA) is a linear transformation of the variables that define a new set of coordinates such as on the new first axis will lie the coordinate with the majority of variance, on the second axis the coordinate with the second majority of variance and so on. The application on the example of figure 3.1 is shown in 3.2 (b).

This transformation is found computing eigenvectors and eigenvalues of the covariance matrix M of the data sample $\vec{x}$: the eigenvalues quantify the variance of each axis found by the corresponding eigenvector.

It's possible to demonstrate that the variance in a generic linear function $f(\vec{x}) = \vec{x} \cdot \vec{v} + c$ at the first order is:

$$\sigma \simeq \vec{v} \cdot M \cdot \vec{v}$$

In the ideal case without measurement errors a basis of the kernel of matrix M define the $\vec{v_i}$ of constraint equations $\widetilde{f_i}$. In the real case the $n - m$ eigenvectors corresponding to the $n - m$ smallest eigenvalues will define the basis of the "kernel", normalized such as $v_i^2 = \frac{1}{\sigma_i}$ in order to verify the definition of $\widetilde{f_i}$. The c_i are easily found imposing $\left\langle \widetilde{f_i} \right\rangle = 0$:

$$c_i = \frac{\sum \vec{v_i} \cdot \vec{x}}{N}$$

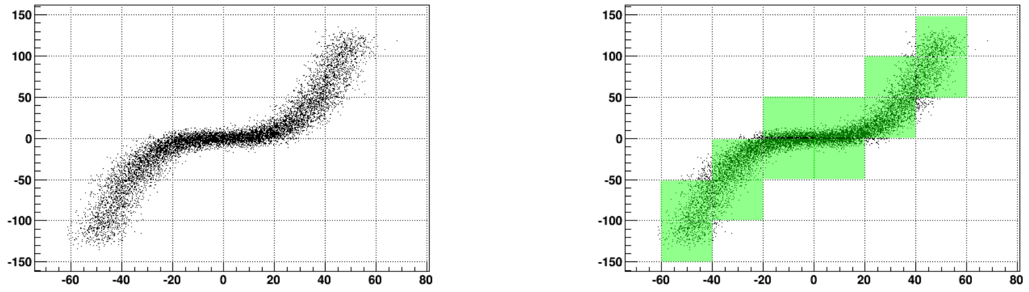
It's worth to notice that this method is based only on analysis of a data sample of the response of the detector to a single charged particle: it's possible to use real data and all the necessary information on detector geometry is automatically inserted in the constants of the constraint equations without need explicitly describe it and apply alignment corrections.

To find the constants $\vec{w_j}$ and q_j of parameters equation $p_i(\vec{x})$ we need again to use the matrix M coming from the data sample, but we also need to know the real parameters of the particle trajectory $\tilde{p_i}$. It's then mandatory to use a Montecarlo sample or to fit the trajectory parameters on the real data by other means.

Minimizing the variance $\sigma^2(\widetilde{p} - p)$ we find that $\vec{w_j}$ and q_j are:

$$\vec{w_j} = M^{-1} \cdot \vec{\gamma_j}$$

$$q_j = \langle \tilde{p_j} \rangle - \langle \vec{w_j} \cdot \vec{x} \rangle$$



(a) **(b)**

Figure 3.3: Unidimensional non-linear manifold in $\mathfrak{R}^2$: (a) coordinate axis are segmented and (b) a certain number of patters covering the manifold are found (in green).

Where $\vec{\gamma_j} = \langle \tilde{p_j}\vec{x}\rangle - \langle \tilde{p_j}\rangle \langle \vec{x}\rangle$.

This method is described in detail in (24).

3.1.2 Pattern and associative memory

In 3.1.1 we have seen that the space has been subdivided in hypercubes covering the manifold to apply the linear approximation and to calculate the track parameters and the fit quality. This subdivision suggests us also a mean to solve the combinatorial problems on real events.

The hypercubes in the manifold identify the combinations that may be a particle trajectory. So it's convenient to find a way to evaluate only those points in the hypercubes and discard the others.

We define the concept of pattern: we segment the coordinate axis in a certain number of steps (in the real case of SVT the microstrip of each detector layer are grouped in superstrip) and we call pattern a n-tuple of segments, one for each coordinate. The hypercubes of the previous description are patterns.

Instead of computing the combinations of full resolution hits on each coordinate we'll restrict the combinations only to the segments on layers. This way we'll reduce greatly the number of combinations to try. In the current SVT each superstrip contains at maximum two non adjacent strips, so at maximum two hits, and the mean multiplicity of hits inside a superstrip is 1.2: the number of combinations is much less than combining all the hits in the detector.

If we have a collection of patterns (pattern bank) that cover at best the volume of all possible trajectories the tracking problem is to find which patterns are present in a given event, compute the combinations inside each pattern and perform the fit. In figure 3.3 is shown how the concept of pattern apply to the previous example of unidimensional manifold (figure 3.1).

How to perform the fit and compute parameters and fit quality for a given combination has been described, the remaining problem is how to compare a large number of patterns to the event. To solve this problem we use an associative memory.

A common random access memory, a RAM, when an address is supplied returns the data corresponding to the address. An associative memory (in the computer science language is more often used the term content addressable memory or CAM) instead receives the content, look up if it's present inside and returns eventually the address of the location where it is. Typical applications of associative memories outside of high energy physics are CPU caches or routing tables in switch and routers.

A kind of associative memory, able to receive the flux of hits coming from the detector and find all patterns for a given event, has been completely developed by the CDF collaboration for SVT (16).

How an associative memory works can be explained with the following example: each element of the associative memory is a pattern and is like a bingo player with his own scorecard, incoming data flux is distributed to each pattern like the numbers in bingo are read out loud. At each given number each player checks if it's present in his own scorecard and when it makes bingo - when all superstrip in the pattern are present in the hits of a given event - it announces the win. All winning players, all pattern addresses, are collected and sent in output.

An important characteristic of the pattern bank is how well it cover the volume of all possible trajectories. As it's usually computed from a finite data sample (real or Montecarlo) and the number of storable patterns is finite as well, it's not always 100% of the volume.

We define the coverage of the bank as the ratio of covered trajectories with respect to all possible ones. Given a fixed amount of storable pattern the biggest the volume of the pattern - in terms of the step size of the segmentation of each coordinate - the highest the coverage, but also the number of combinations of fits belonging to the pattern will be higher.

Optimization of the bank coverage is a difficult problem, the solution is strongly dependent from the detector characteristics. It's anyhow true that regardless of the kind of optimization the largest the bank the highest the coverage and so the efficiency of the tracking algorithm. In fact during the evolution of the associative memory technology ((8, 19)) a great effort was put into pushing the density of the chips and the degree of parallelization in order to be able to store a lot of patterns and process them with very low latency. With this enhanced technology is possible also to think at applications for level 1 tracking (11, 20, 21).

4

Silicon Vertex Trigger

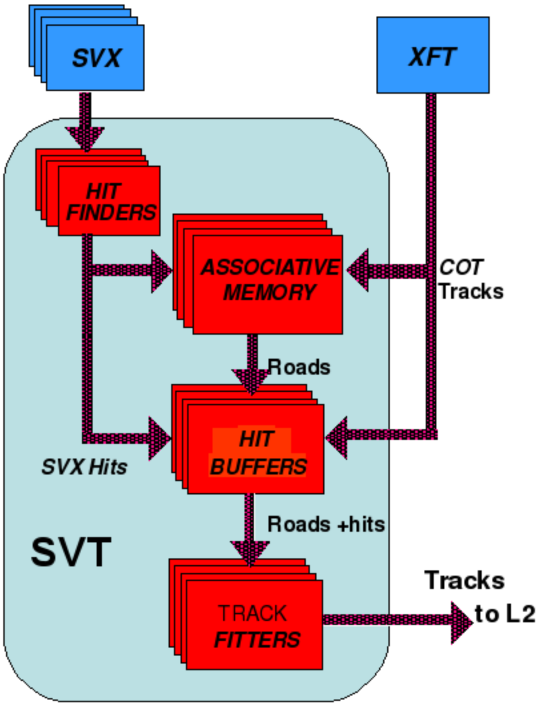


Figure 4.1: SVT Scheme - Input and major algorithm steps are drawn in this schematic view: input data come from SVX and XFT, SVX hits are clusterized, patterns are found in the associative memory and tracks are fitted with the high quality linear fit. The found tracks are sent to L2.

The Silicon Vertex Trigger (SVT) is the dedicated hardware pro-

cessor that implements a tracking algorithm based on the ideas described in detail in 3.1.
It receives SVX hits and XFT tracks for every event accepted by the level 1 trigger, performs the track reconstruction algorithm and sends the output tracks to level 2 decision processor in an average time of 20 μs (figure 4.1).
The system is made by over a hundred of 9U VME cards organized in ten 21-card crates (figure 4.2). The key elements used in building SVT were: (a) extensive use of FPGA, (b) flexible common motherboard for most of the functions, (c) common cabling and data format for all the cards and (d) for the most intensive computational part of the algorithm (the pattern recognition in associative memory) a custom ASIC chip was developed. Another key element of design is the parallelization of all tasks and the highly symmetric structure of the whole processor.

4.1 Design and performances

The algorithm performs the following steps:

- SVX hits and XFT tracks (c and phi parameters) are received. Both are treated the same way so I'll call them generically hits.
- For each hit the corresponding superstrip is computed, and hits are stored in a smart database ordered by superstrip
- Superstrip are sent to the associative memory bank
- Patterns found in the event (roads) are received from the associative memory and patterns containing the same information (hits) are deleted ("ghost roads" removal)

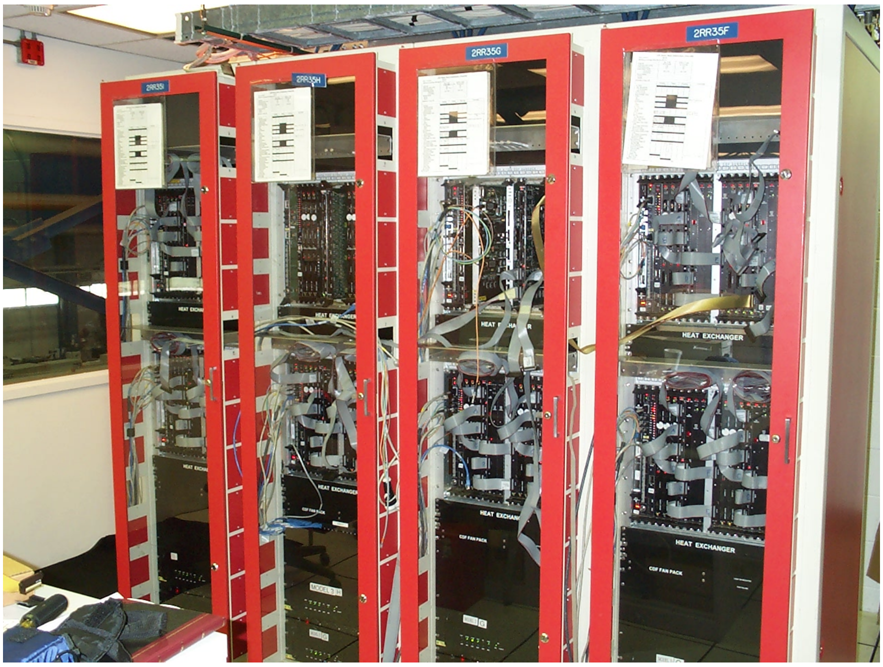


Figure 4.2: The SVT racks - SVT is made by 104 9U VME boards in ten crates. The picture shows the four main racks containing all the system up to the Track Fitting and Final Merge and Corrections that are in other two crates in another rack.

- For each road associated hits are retrieved
- For each combination of hits inside each road the track fit is performed (full resolution tracks)
- All fits that under a certain χ^2 value are collected, duplicated tracks characterized by different silicon hits associated with the same XFT track are deleted ("ghost tracks" removal)
- Beam position is subtracted from impact parameter of each track and a second order correction is applied on ϕ parameter
- Finally the tracks are sent to the output

Each of these steps is handled by one or more cards that will be described in 4.1.1.

An important feature of SVT is that all of those steps, except the duplicate tracks removal and final corrections, can be executed independently in parallel on subregions of the detector. Since the SVX detector is subdivided in twelve 30° ϕ sectors called wedges (figure 2.6), 12 dedicated SVT pipelined processors process in parallel data for each wedge up to the last steps.
This makes SVT a highly segmented system. This was a crucial feature during the upgrade program: the new hardware was tested on a single wedge in parasitic mode and once stable was ready for all wedges. This configuration also helps with the maintenance of the system.

4.1.1 Hardware structure

The SVT hardware is made by several different VME boards each one handling a different step of the algorithm described in 4.1. The

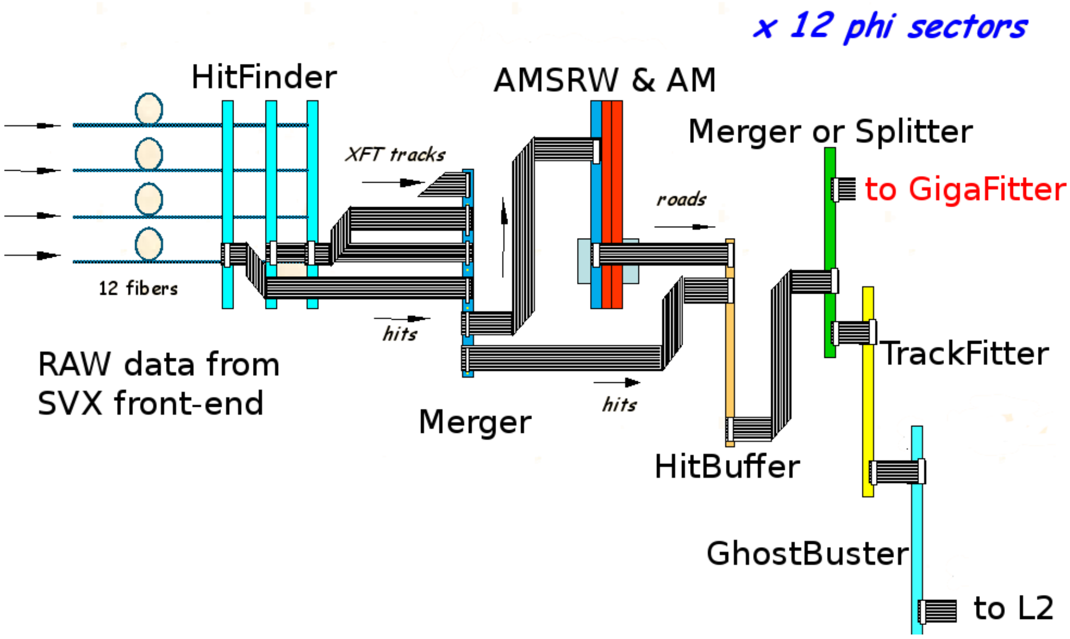


Figure 4.3: SVT Dataflow and Hardware scheme - The dataflow in one SVT wedge is shown up to the final stage of the Ghost Buster were all wedges are merged in a single data cable. The position of the GigaFitter in parasitic mode is highlighted.

dataflow for one SVT wedge with all hardware involved is shown in figure 4.3, the hardware is the actual one after the 2006 upgrade.

A uniform communication protocol is used for all data transfers throughout the SVT system. Data flow through unidirectional links connecting one source to one destination. The protocol is a simple pipeline transfer driven by an asynchronous $\overline{\text{Data Strobe}}$ (DS_ in the following text, it is an active-low signal). To maximize speed, no handshake is implemented on a word by word basis. An $\overline{\text{Hold}}$ signal is used instead as a loose handshake to prevent loss of data when the destination is busy (it is an active-low signal and will be called HOLD_ in the following text). Data words are sent on the cable by the source and are strobed in the destination at every positive going DS_ edge. The DS_ is driven asynchronously by the source. Correct DS_ timing must be guaranteed by the source. Input data are pushed into a FIFO buffer. The FIFO provides an $\overline{\text{Almost Full}}$ signal that is sent back to the source on the HOLD_ line. The source responds to the HOLD_ signal by suspending the data flow. Using $\overline{\text{Almost Full}}$ instead of $\overline{\text{Full}}$ gives the source plenty of time to stop. The source is not required to wait for an acknowledge from the destination device before sending the following data word, allowing the maximum data transfer rate compatible with the cable bandwidth even when transit times are long. Signals are sent over flat cable as differential TTL. The maximum DS_ frequency is roughly 40 MHz.

On each cable there are 21 data bits, $\overline{\text{End Packet}}$ (EP_), $\overline{\text{End Event}}$ (EE_), DS_ and HOLD_. Data are sent as packets of words, the EP_ bit marks the last word of each packet: the End Packet word. The EE_ bit is used to mark the end of the data stream for the current event. End Event words are one-word packets so EP_ is also 1, the data field is used for Event Tag (8 bit), Parity (1 bit, computed on all data words of the event) and Error Flags (12 bits).

A first important VME board to describe is the Merger: it has four input and two outputs. The purpose of the board is to merge the data coming from the four inputs or any subset of them into one output. The two outputs of the board are identical copies of the merged data, but with separate hold signal handling: it has the possibility to consider or ignore one or both holds on the two outputs. This board is used inside the SVT pipeline at various stages, but it's also extremely useful for planning the upgrade tests and commissioning because the two outputs provide easily a copy of the data made at any stage of the SVT pipeline to a new processor to be tested.
A similar board with only the output copy feature is the Splitter: it has two inputs and four outputs. Each input is copied into two outputs with separate handling of hold signals like the Merger. This board is not used in the normal SVT pipeline but it's used for the parasitic configuration in the GigaFitter upgrade (see 5.5).

The SVX hits coming from each wedge of each mechanical barrel (figure 2.7) of SVX are received by the Hit Finder (HF) boards from fiber optic links connected to the SVX readout hardware. For each wedge processor the data from three Hit Finder, one for each mechanical barrel, is merged in a Merger board along with the XFT tracks for that wedge.
Merged SVX hits and XFT tracks are sent to both the Associative Memory Sequencer Road Warrior (AMSRW) board and the Hit Buffer (HB++) board. The AMSRW provides the superstrips out of the hits and sends them to the Associative Memory (AM++) boards. The AM++ boards send back to the AMSRW the roads found and the AMSRW, after a duplicate road elimination (Road Warrior function), sends them to the HB++.
The HB++ associates the received SVX and XFT information to each road found by the associative memory and sends this information to the Track Fitter (TF++) board. The TF++ computes all

combinations of hits for each road and performs the track fits. All tracks from all 12 wedges satisfying a certain quality cut are merged (using 4 Merger boards) and sent to the Ghost Buster (GB) board. The GB performs the duplicate tracks suppression: SVT tracks that share the same XFT segment are considered redundant and only the one with the best χ^2 is chosen, the other are suppressed.

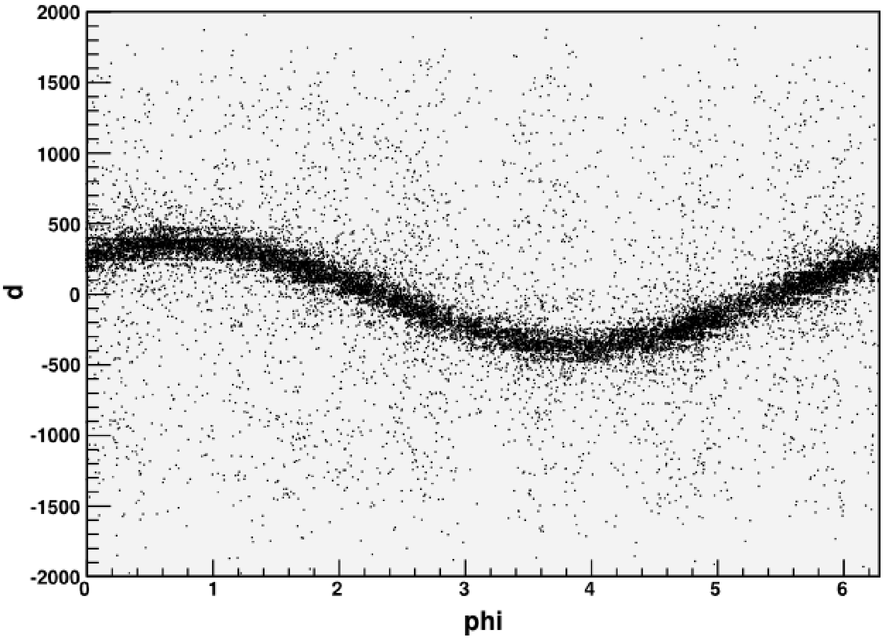


Figure 4.4: SVT Impact parameter vs ϕ - The sinusoidal shape of the impact parameter vs ϕ plot is given by the beam position relative to the center of axis. From this plot is computed online the position and the impact parameter is corrected in the GB with this information.

The SVT is supposed to work with the beam in his nominal position, for example parallel to the z-axis and with $x = y = 0$. In practice some misalignments and time variation of the beam position are possible thus corrections are needed. The beam position in the transverse plane can be calculated from the correlation between impact parameter (d) and ϕ angle shown in figure 4.4. If the beam position in the transverse plane is (x_0, y_0) the relationship between d and ϕ for primary vertex tracks is $d = -x_0 \sin\phi + y_0 \cos\phi$. The impact parameter with respect to the position of the beam is $d' = d + x_0 \sin\phi - y_0 \cos\phi$.

A beam-finding program monitors the tracks in input to the GB, fitting and reporting to the accelerator control network an updated Tevatron beamline fit every 30 seconds. The beam fit is also written to the DAQ event record and used by the GB board to correct in-situ every SVT track's impact parameter for the sinusoidal bias vs ϕ resulting from the beamline's offset from the detector origin, so that the trigger is immune to modest beam offsets. The GB also applies a tabulated second order correction to the ϕ parameter.
The GB then sends the SVT tracks to the Level 2 processor for trigger decision.

4.1.2 Tracking resolution

The most striking SVT performance is the impact parameter resolution. Figure 4.5 taken from the SVT TDR shows the comparison of SVT simulation on real RUN 1 data with the offline reconstruction.

Figure 4.6 shows the real SVT resolution measured on CDF RUN II data. The expectations declared in the TDR were confirmed by the real SVT.

4.1.3 Efficiency

The tracking efficiency (very sensitive to many factors in the experiment), needed adjustments causing variations during time, especially at the beginning of the data taking. Data taken by CDF

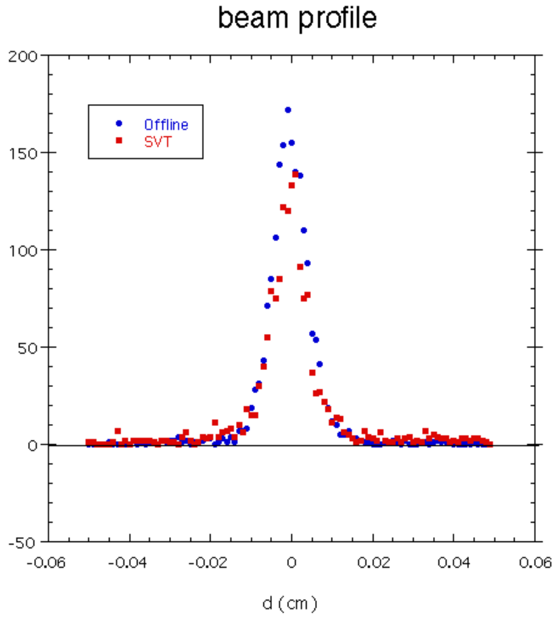


Figure 4.5: CDF beam profile: SVT and offline - The beam profile reconstructed with SVT tracks (red) and offline tracks (blue).

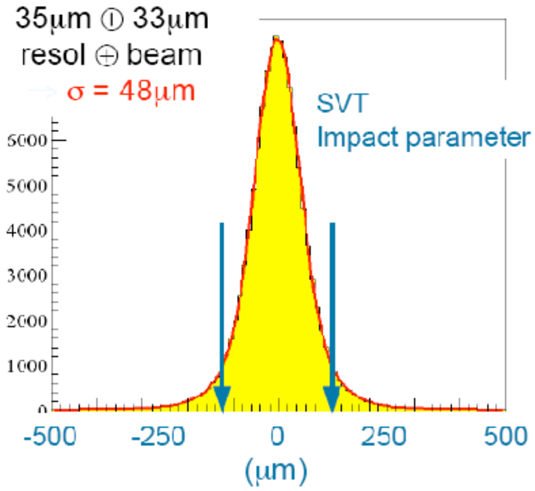


Figure 4.6: SVT impact parameter resolution - SVT resolution on impact parameter.

before June 2003 used only four silicon layers connected to the XFT segment for the pattern recognition, and requiring all of them to be fired (4/4). An important efficiency gain has been obtained implementing the use of the “majority logic” in the track match criteria. In fact SVT can require 4 fired layers among a total of 5 silicon layers (4/5). The gain is a “varying” number since it is a function of the detector status which, especially at the beginning, changed, even on short timescales.

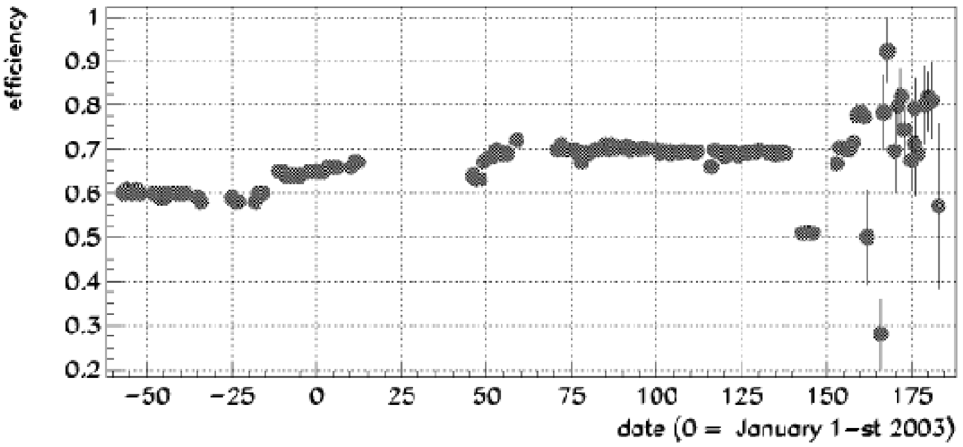


Figure 4.7: SVT efficiency in 2003 - At the beginning of 2003 the SVT efficiency was limited from the status of SVX layers. The implementation of majority logic helped to overcome the non uniformity of SVX efficiency on all layers.

Figure 4.7 shows the SVT track efficiency as a function of day in 2003, when the majority was implemented. The plot shows a long data taking period, where the 0 corresponds to the first of January 2003. The track efficiency has a slow increment from 60% to 70% due to the SVX detector improvement (larger number of active strips/ladders). This 70% efficiency is the product of different concurrent contributions:

1. the single hit efficiency (95%) contributes as $95^4 = 81\%$ to the track efficiency

2. the bank efficiency (95%)
3. the χ^2 cut efficiency after the track fitting (95%)
4. a geometrical acceptance due to the SVX ladder status (95%). This is the relevant part to explain the improvement from 60% to 70%

The 4/5 has been implemented in June 2003 and is shown in the plot as an additional track efficiency improvement up to 80%. Statistical errors are much larger in the last period since the track efficiency is calculated using a low-statistic sample. In fact the track p_T acceptance threshold has been increased in that period from 1.5 GeV to 2.0 GeV in order to reduce the L2 processing time, that increased a lot when the majority was implemented. Moreover, for the same goal, most of the events not used for the impact parameter selection have been forbidden to transit through SVT. Only few of them are allowed just to calculate the efficiency shown in figure 4.7.

Any increment on track efficiency has an amplified effect as increment on signal yields, since the searched particles are reconstructed combining multiple tracks, so the signal efficiency should be roughly a power law of the track efficiency. A simple example is shown in figure 4.8 where the yield of the D^0 is reported in the two cases of majority activated (4/5) or turned off (4/4). An increment of 10% in track efficiency corresponds to an increment of 30% on the size of the collected D^0 sample. An even larger effect is expected on samples that use more than 2 tracks to select the sample.

SVT, since the moment the majority was implemented, had impact on the level 2 dead time, because the processing time increased a lot causing large tails in the timing distribution. This effect, combined with a small number of level 2 buffers, caused dead time to the experiment. To reduce the dead time, the level 1 rate in input

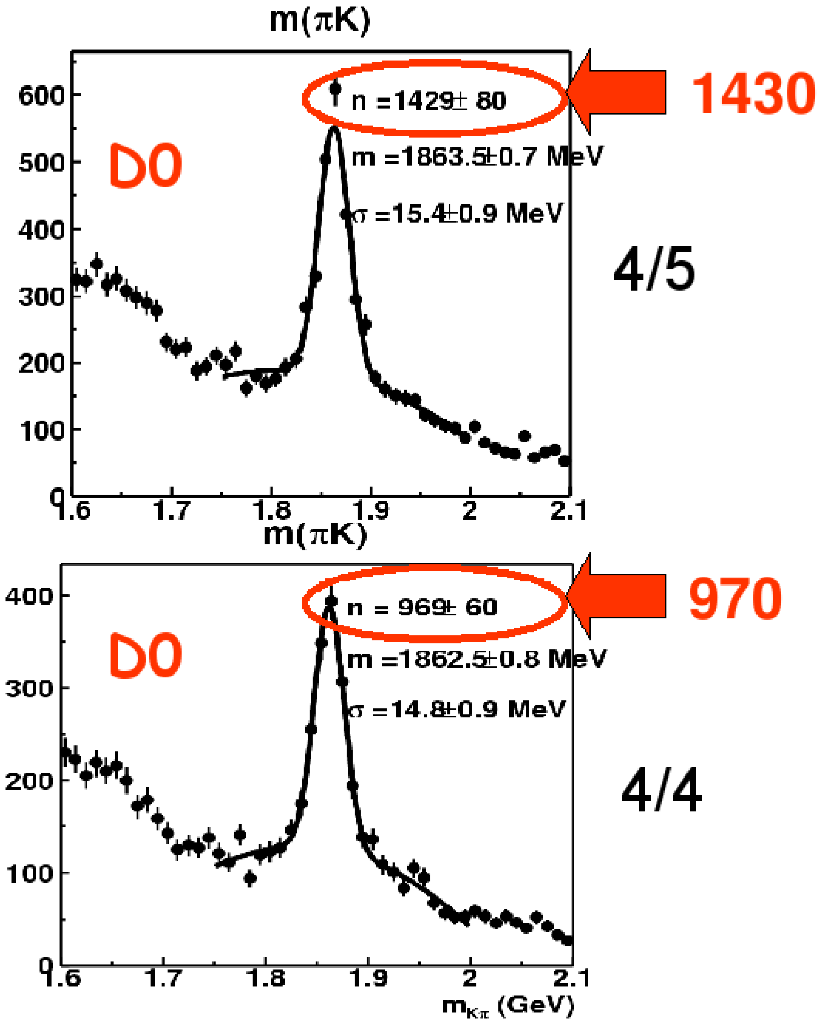


Figure 4.8: D^0 yield gain after 4/5 introduction - the number of D^0 mesons acquired increased by 30% after 4/5 majority was introduced.

to SVT was artificially reduced, reducing the CDF physics potentiality.
In 2003 the bandwidth was limited below 20 kHz, but it was clear that the Tevatron increasing instant luminosity would have reduced constantly the available bandwidth arriving to values below 10 kHz. The SVT upgrade solved the problem: the average SVT processing time was deacreased, but even more so the size of the tails, bringing back to CDF the full 30 kHz level1 bandwidth.
The effect of the upgrade can reliably be estimated by comparing SVT processing time before and after the upgrade at the same luminosity, with the same trigger path mixture. Figure 4.9 shows the improvement on the fraction of events with processing time above 50 μs as a function of luminosity, for different stages of the SVT upgrade. This fraction of events is interesting because over-threshold events directly contribute to trigger dead time.

The first improvement, installing the AMS/RW boards, reduced processing time by reducing the number of track fit candidates and reducing the pattern recognition time. The second upgrade, installing the upgraded Track Fitter (TF++) board, significantly reduces the fraction of over-threshold events by speeding up the track fitting process with faster clocks and a six-fold increase in the number of fitting engines on the new board. Next, the use of 128K patterns reduces the number of fit combinations per recognized pattern. The upgraded Hit Buffer (HB++) further increased the processing speed by virtue of the faster clock speed on the upgraded board. Finally, the full power of the upgrade is visible after enabling all 512K patterns. The fraction of events over threshold is well below 5% at the highest luminosities available for these tests. Data taking without an upgraded SVT system at these luminosities would clearly suffer huge rate penalties, as the corresponding fraction of events over threshold is roughly 25% at half the maximum tested luminosity, with a steeply rising tendency. The SVT upgrade has

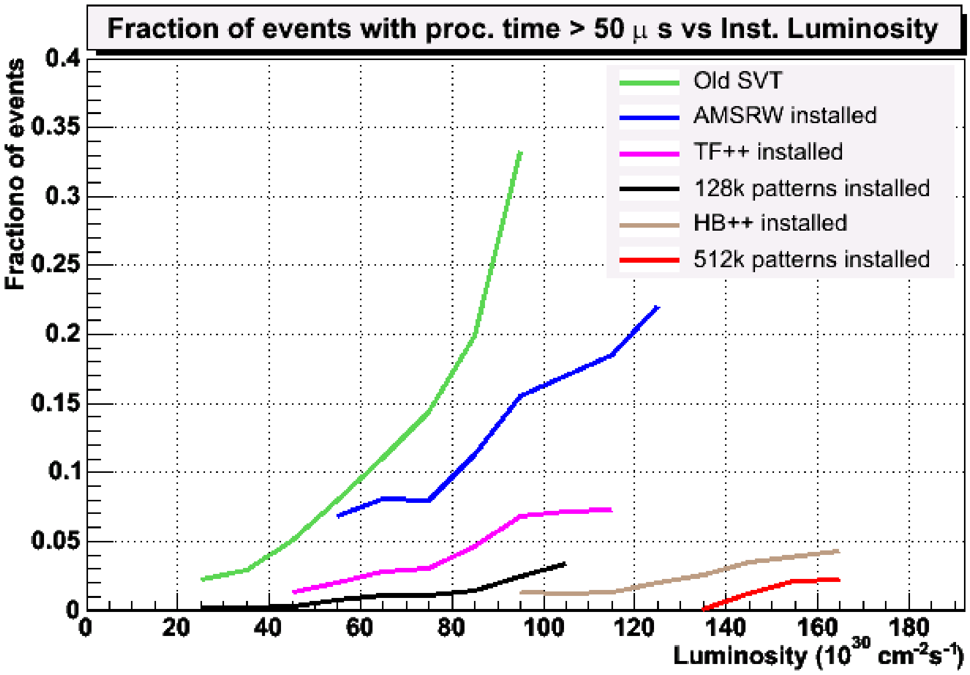


Figure 4.9: SVT Timing tails and upgrades - The phased installation of new SVT hardware allowed to contain the tails of the SVT processing time along with the Tevatron peak luminosity increase.

been a clear success in the tested range of luminosities. Since the fully upgraded system shows a very little dependence on the instant luminosity (the dependence is very near to be flat), the result is almost valid also for higher luminosity.

4.1.4 Diagnostic features

Several design features of SVT contributed to its rapid commissioning and reliable operation. The essence of SVT's component-based architecture is captured by the standard SVT cable and the SVT Merger board. As described in 4.1.1 nearly all SVT internal data travel as LVDS signals on common 26-conductor-pair cables. Data fan-in and fan-out are performed inside FPGAs, not on backplanes, by the Merger board. Every fan-in stage compares event IDs for its sources and can drive a backplane error line on mismatch. A parity bit for each cable-event provides a basic check of data integrity. It is illustrative of SVT's design strategy that the SVT cable and Merger board were prototyped and tested before the other boards.

On each end of every SVT cable is a circular memory buffer ("spy buffer") that records as a logic state analyzer the last words sent or received on that cable. Comparing a sender's output buffer with a receiver's input buffer checks data transmission. Comparing a board's input and out-put with simulation software checks data processing. The memories also serve as sources and sinks of test patterns for testing single boards, a small chain of boards, a slice of SVT, SVT as a standalone system, or the data paths to SVT's external sources and sink. The buffers can be frozen and read by monitoring software parasitically during data-taking, and all of SVT's buffers can be frozen together, via backplane signals, when any board detects an error condition, such as invalid data.

Moreover, by polling SVT's circular memories during beam running, large samples of track and hit data, pattern IDs, etc. - unbiased by L2 or L3 trigger decisions - are sampled and statistically analyzed to monitor data quality. This kind of monitoring is independent of the CDF DAQ system and provides detailed informations of every board of the system regardless of whether SVT or part of it is in the DAQ or in parasitic mode.

Within the DAQ system there is also another monitor tool used for all Trigger and DAQ systems and applies also for SVT: a fraction of the collected data banks, which for SVT is the output of the GhostBuster board plus timing informations from AMSRW and TF++ boards, is sent to an online program during the data taking. This program analyze the data and monitor various quantities such as the efficiency with respect to level 3 tracks and the differences between SVT and his offline simulation. This kind of monitoring relate SVT with the rest of the DAQ system, providing essential informations regarding the health of SVT during data taking, but differs from the spybuffers as it treats SVT as a whole, with little informations on the internals of the system, and it's biased by L2 and L3 trigger decisions.

Both methods are used allowing efficient maintenance of the SVT system even if it's so complex and made by many different boards, furthermore these powerful monitoring tools have been essential for fast development of the past upgrades and the GF.

5

GigaFitter

The GigaFitter is an hardware processor to perform the track fitting task of the SVT algorithm.
It's designed to receive data from the 12 HitBuffers of SVT, to compute all possible hit combinations, to perform the linear fit of tracks finding high quality parameters, and to select the tracks to be sent out. The output format is such to be received by the GhostBuster board of SVT. It's a possible replacement of the current track fitting system of SVT, reproducing all functionalities of current system and enhancing it's capabilities.

5.1 Design considerations and features

The GigaFitter has been designed to replace a complex system made by 16 boards: 12 "processors" called TF++ and 4 boards for data stream merging into a single cable. The new system must be also faster than the old TF++, especially at high luminosity when events are complex and many candidate tracks must be fitted and evaluated. The GigaFitter should allow the use of SVT at higher luminosities with larger efficiencies.

To achieve this goal the design is based on a synchronous pipeline of simple and optimized logic modules; all modules with functions longer than one clock cycle are replicated and put in parallel to maximize the bandwidth. A set of FIFOs and buffers helps to keep high clock frequencies, cross different clock domains and compensate fluctuations in the input and output data streams.
The system is able to fit and evaluate one candidate track per clock cycle on each of the 12 inputs with an internal clock of 120 MHz, about 1.4 fit/ns.
However a timing faster than the TF++ system is not the only goal of the GigaFitter. It has been designed also to add some important features that were missing on the old system: full precision fits, larger variety of constants sets, better 5/5 track handling. These new features should produce better track efficiency and signal/background rejection.

5.1.1 Full precision fits

As seen in 3.1.1 the computation of track parameters and χ^2 components is done with a scalar product plus a constant term:

$$p_n = c0_n + \sum c_{ni} * x_i$$

The terms c_{ni} and c_i are 18 bit and 15 bit wide, in the GigaFitter there are DSPs with 18x25 bits dedicated multipliers so it's possible to compute exactly p_n with the equation above. Scalar products in the old low density FPGAs were implemented using discrete logic and for this reason they occupied a large chip area with timing problems. Finally only 8x8 bits multipliers were implemented, even if words to be multiplied were wider. As a consequence it's not possible to use exactly that equation in the TF++. A very clever approximation is adopted to compute the $c_{ni} * x_i$ terms.

Each c_{ni} and x_i is decomposed as $c_{ni} = c_{ni}^{high8bit} * 2^{shift_{ni}} + c_{ni}^{low}$ and $x_i = x_i^{ssborder} + x^{low8bit}$. This way the multiplications is written as:

$$c_{ni} * x_i = c_{ni} * x_i^{ssborder} + c_{ni}^{high8bit} * x_i^{low8bit} * 2^{shift_{ni}} + c_{ni}^{low} * x^{low8bit}$$

The terms $c_{ni} * x_i^{ssborder}$ and $shift_{ni}$ depends only on constants and patterns, so they can be computed offline and preloaded in a memory on the TF++. The information provided by the most significant bits is included in pre-calculated terms, one term for each AM pattern to be stored in dedicated memories of the TF++. This choice introduces a one by one correlation between the dimension of the AM and the TF++ memory that turns out to be very large. This is a disadvantage of the TF++ currently installed inside SVT: the constants used in fit's scalar products require a very large memory. This feature is the actual limit to the bank size we can use inside SVT.

The term $c_{ni}^{high8bit} * x_i^{low8bit}$ is a 8x8 bit multiplication and is calculated online. The term $c_{ni}^{low} * x^{low8bit}$ is negligible and is not computed.

The effect of not computing the last term account for a little smear of the resolution for the TF++ with respect to the full precision computation as done by the GigaFitter and the offline code. The difference for each parameter and χ^2 is show in figures 5.1, 5.2, 5.3 and 5.4. The χ^2 difference shown in 5.1 is proportional to the χ^2 itself because the $c_{ni}^{low} * x^{low8bit}$ term is 1-2 units for each component, then squared and summed. A small amount of track χ^2 found by the GF above the threshold were accepted by the TF++ and vice versa. Globally this effect is about 2% of the total number of tracks, but we'll see in 6.2.2 that the GF is more efficient of circa the same percentage without increasing the number of fakes, so the χ^2 computed by the GF is a more accurate quality parameter.

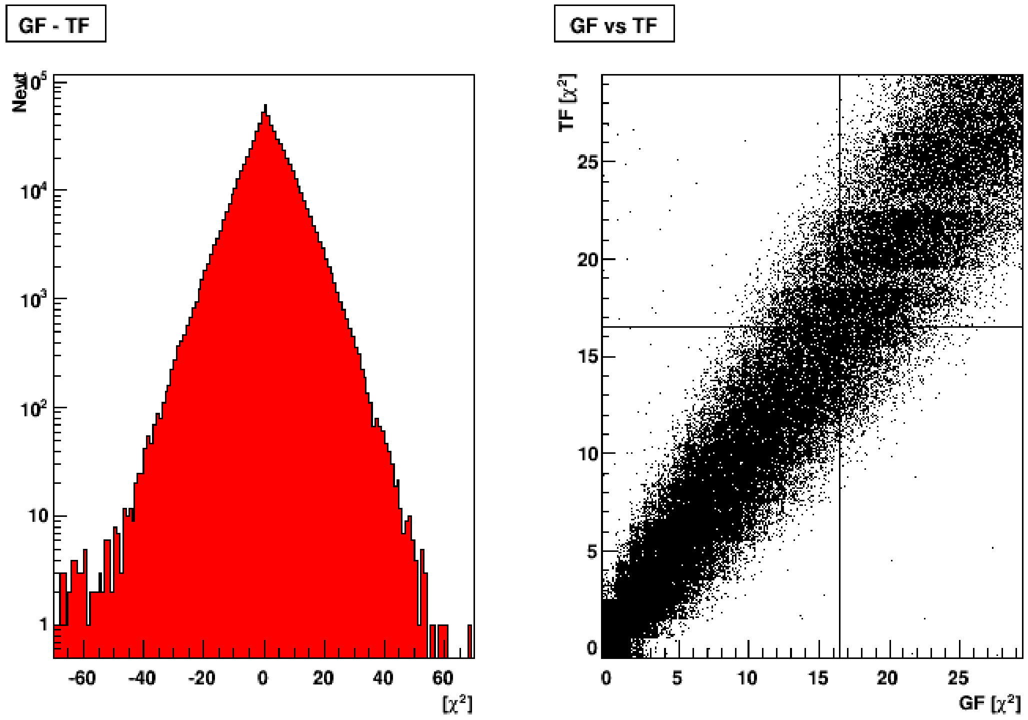


Figure 5.1: GF vs TF++ differences: χ^2 - Differences in χ^2 computation between GF and TF++ due to $c_{ni}^{low} * x^{low8bit}$ term not computed by TF++. Current cut values are shown with the solid lines.

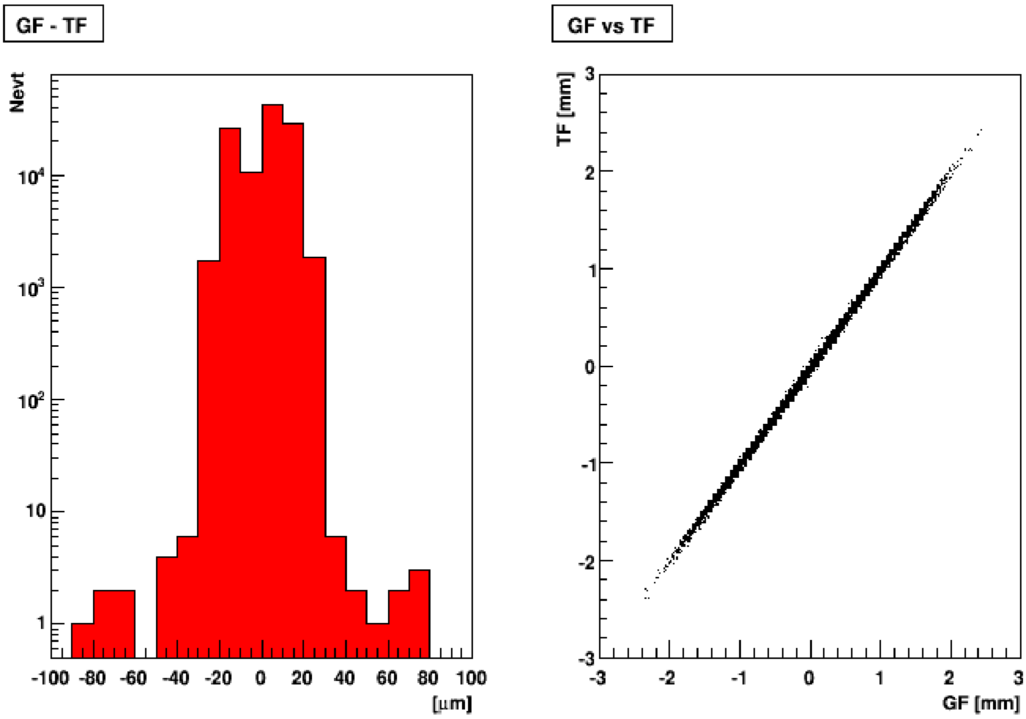


Figure 5.2: GF vs TF++ differences: $d0$ - Differences in impact parameter ($d0$) computation between GF and TF++ due to $c_{ni}^{low} * x^{low8bit}$ term not computed by TF++.

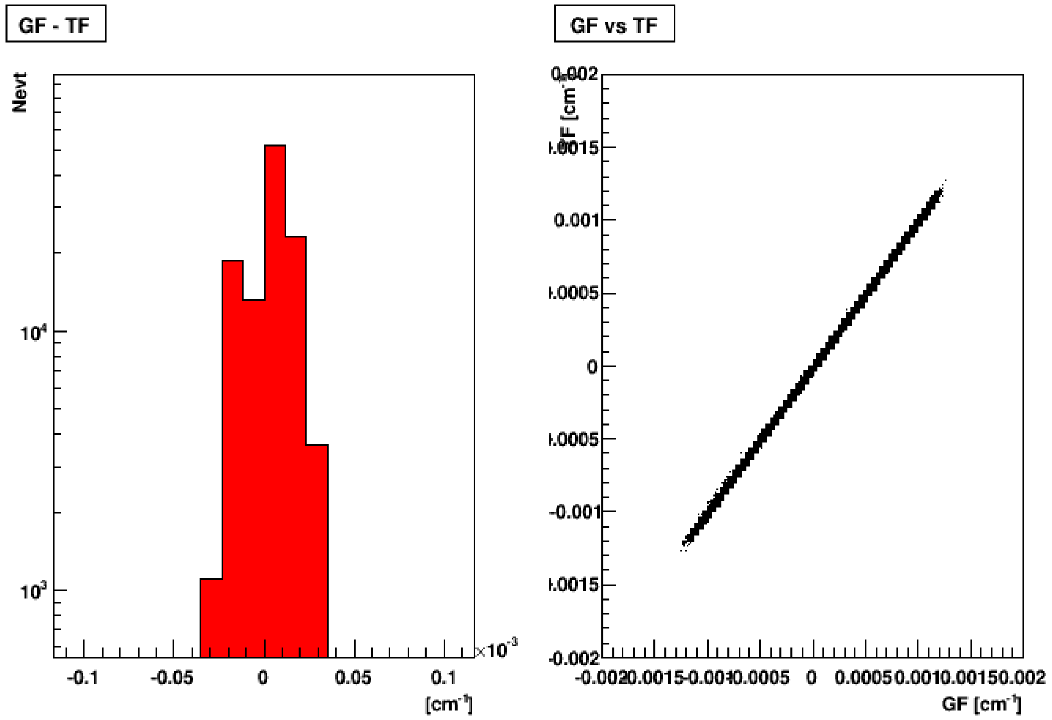


Figure 5.3: GF vs TF++ differences: c - Differences in curvature (c) computation between GF and TF++ due to $c_{ni}^{low} * x^{low8bit}$ term not computed by TF++.

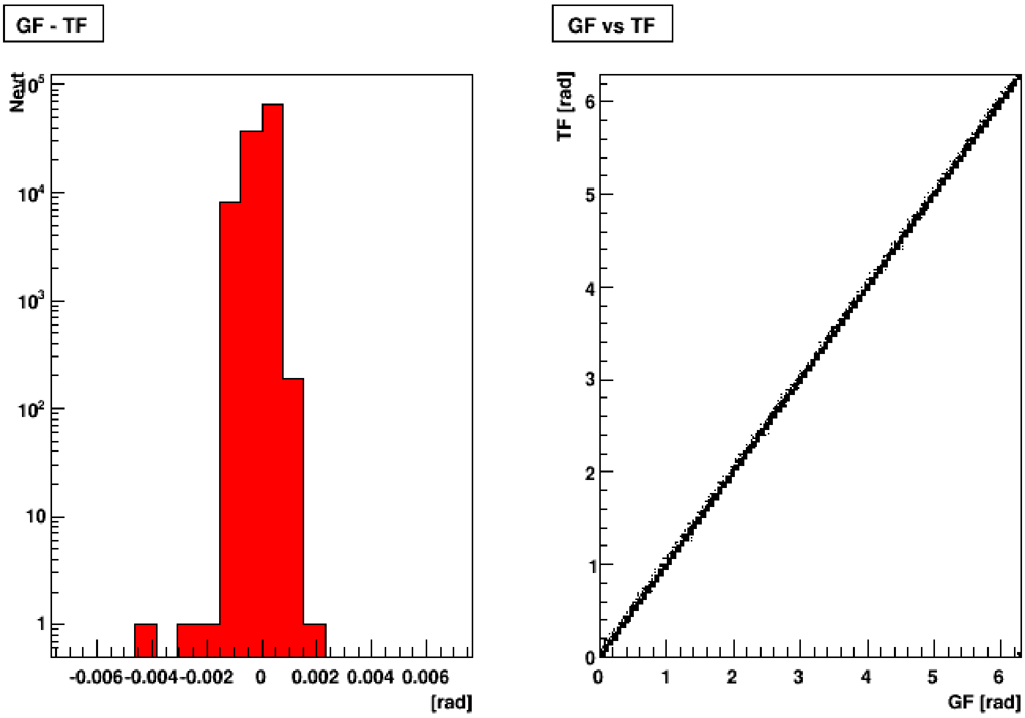


Figure 5.4: GF vs TF++ differences: ϕ - Differences in ϕ computation between GF and TF++ due to $c_{ni}^{low} * x^{low8bit}$ term not computed by TF++.

5.1.2 Many set of constants for improved efficiency

The TF++ has different constants sets for each of his TF++ boards, one per wedge, since the fit constants extends to the whole wedge. However inside a wedge each particular track configuration needs specific constants to be reconstructed precisely. For this reason each wedge requires various $c0_n$ and c_{ni} constants sets, each one computed for track fitting in particular conditions or regions of the detector characterized by a particular layer configuration.
The large size of each TF++ constant set (the $c_{ni} * x_i^{ssborder}$ and $shift_{ni}$ constants needs to be computed for each AM pattern) puts a limit to the number of specific cases that can be handled. The TF++ board is able to store only 30 different set of constants: one for each of the 6 SVX barrel and one for each of 4 out of 5 SVX layer combination (6x5 constant sets). For example this limitation result in poor quality of track crossing barrels, so poor that tracks crossing mechanical barrels are not included in the AM pattern bank and thus not reconstructed. If the patterns would be included without the addition of relative constants for their precise fitting we would probably increase track efficiency but also would increase the large impact parameter fake rate, resulting in a worst behavior of SVT for trigger decision.

In the GF board instead the 25x18 bit harder multipliers allows the use of full resolution hit position words without the storage of pre-calculated terms. The constants sets necessary to perform the scalar products are just the $c0_n$ and c_{ni} and occupy a small amount of memory. There can be a large number of different constant sets to allow the reconstruction of tracks characterized by hit configurations that up to now were discarded due to hardware limitations. This results in a potentially higher SVT efficiency and lower amount of fakes.

5.1.3 Handling of 5/5 tracks

Another clear advantage of having big DSP arrays is the capability to fit many times the same track deleting one particular layer in each different fit. We then choose the layer configuration producing the best track quality.
In the current TF++ a track that has hits in all the five layers is fitted using a fixed combination of four layers and no attempt to find a better combination is performed even if the resulting χ^2 is higher than the cut value and the track is rejected. As the Tevatron collider luminosity increases it is very important to have the capability to evaluate the track parameters under the assumption that the probability to have a noisy hit in the fitted combination is quite high. This discrimination capability allows to reduce the degradation of the SVT efficiency due to the high detector occupancy.

5.2 Hardware structure

The GigaFitter (GF) system is based on a motherboard called Pulsar (13) and three GigaFitter mezzanines.

The Pulsar board, shown in figure 5.5, is a 9U VME board based on three interconnected Altera APEX20K FPGA: two of them, called DataIO, handle two mezzanine connectors each, while the last, called Control, handles the various input and output connectors of the motherboard. VME communications are possible directly with each FPGA. This board has been widely used in CDF for upgrades of the Level2 trigger system: L2 Global Trigger upgrade (13), L2CAL upgrade (12), and SVT upgrade (5, 6, 9). It's also used for trigger and data acquisition systems of other experiments, as Magic (22) for example.

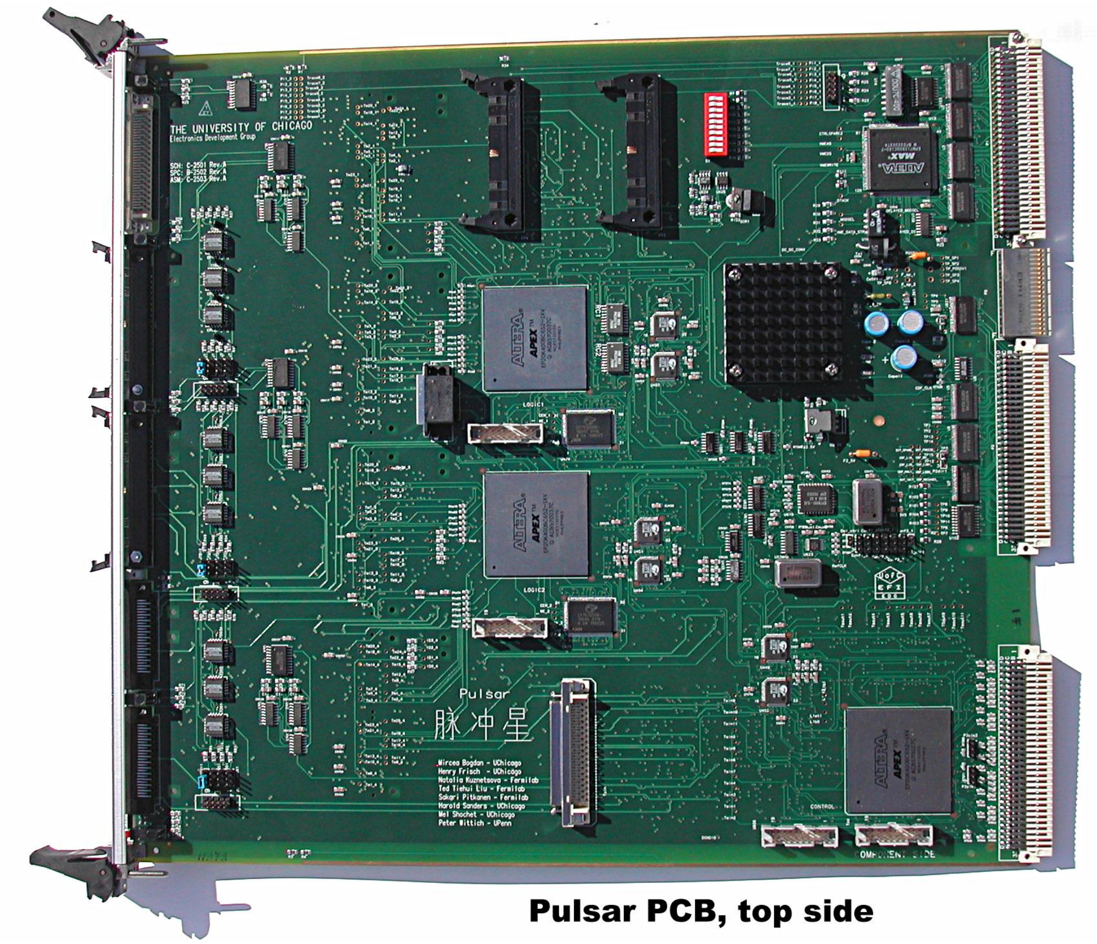


Figure 5.5: The Pulsar board - Front side of the Pulsar board. Mezzanine connectors used by the GigaFitter system are on the back side.

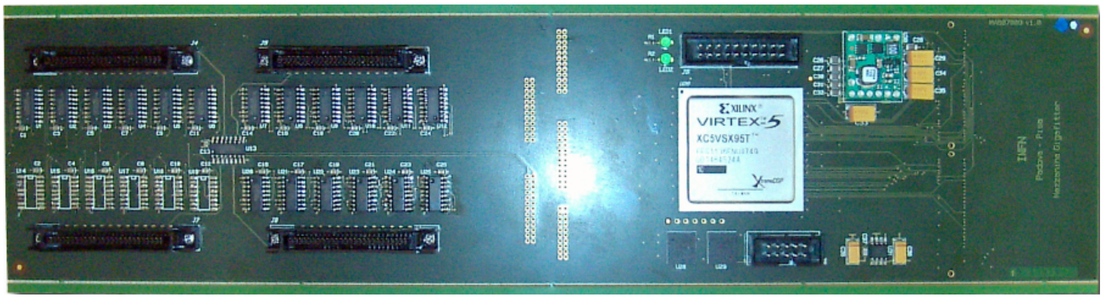


Figure 5.6: The GigaFitter mezzanine - GigaFitter custom mezzanine. All components on the front side.

The GF Pulsar board uses two clocks: 40 MHz to communicate to GF mezzanines (clock to mezzanines is sent by the motherboard) and a 66 MHz clock for all other functions.

The GigaFitter mezzanine, shown in figure 5.6, has been developed exclusively for the GigaFitter system by INFN Padova and INFN Pisa.

The core of the mezzanine is a Xilinx Virtex-5 XC5VSX95T FPGA. This model of FPGA is particularly suitable for the track fitting task because it has 640 DSPs units, each one provided with one 18x25 bit multiplier tied to a 48 bit adder, and BlockRAM of 8.6 Mb. With this components it has been possible to synthesize many parallel fitting units. They perform in parallel the scalar products for the track fitting and fully exploit the computing power of the device.

The mezzanine FPGA receives a 40 MHz clock from the motherboard and generate internally three clocks using Digital Clock Manager dedicated cells: a 40 MHz clock to communicate back to the motherboard, a 25 MHz clock to handle VME and a 120 MHz for all the other functions.

The mezzanine has four input SVT standard connectors to receive data from four wedges. All communications between SVT boards are made with standard LVDS cables. The communication signals and protocol is described in 4.1.1.
The full GigaFitter system with all 12 inputs connected is shown in figure 5.7.

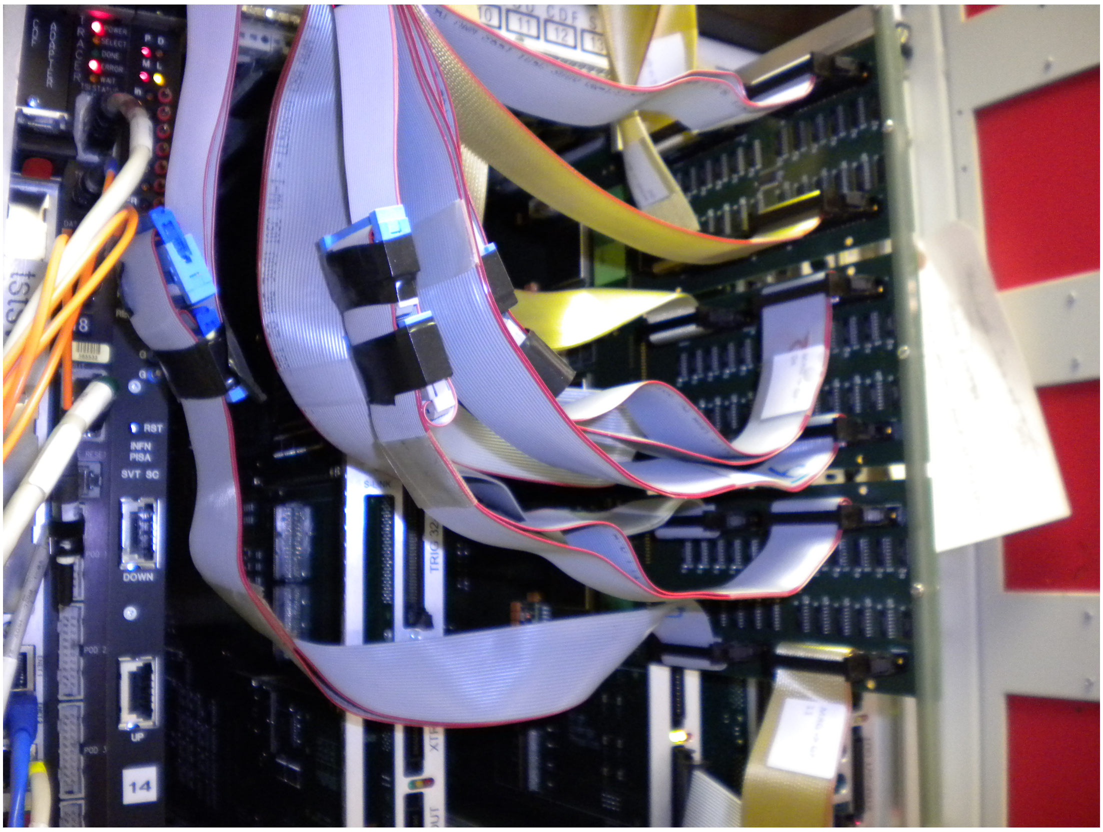


Figure 5.7: The GigaFitter system installed - The GigaFitter system in the test crate of SVT. All 12 input are connected in parasitic mode to splitted HB++ outputs.

5.3 Input and output

The GigaFitter board receive hits and roads from the 12 HitBuffer++ (HB++) boards and sends all found tracks merged in a single output to the GhostBuster board for non-linear corrections, beam subtraction and duplicate tracks suppression (see 4.1.1 for details). All GhostBuster functions can be implemented in the GigaFitter motherboard. Future versions of the GigaFitter firmware will include them so that the GF output can be sent directly to the L2 decision node, skipping the GhostBuster board. Both input and output are handled with SVT cables described in the previous section.

5.3.1 Input data stream

The HB++ transmits for each event some hits+road packets, one for each road found by the AM, followed by an end event packet.
The hits+road packet contains all hits associated to a given road found by the associative memory plus the road identifier as described in table 5.1. Number of words in this kind of packet is not fixed, the minimum is 7 words while maximum is open and depends on the road super strip size. The road size commonly used in the past years gives a maximum of 25 words.

5.3.2 Output data stream

Output data is a packet for each track found in an event followed by the end event packet.
The track packet is always composed by 7 words and contains information about SVX hits associated to the track on each layer, the linked XFT track, AM road, fitted track parameters and fit quality (χ^2 and GF fit status) as described in table 5.2.

	24	23	22	21	20 19	18	17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
1	HOLD_	DS_	EE_	EP_	Layer (0..4)		SVX Hit
2	HOLD_	DS_	EE_	EP_	Layer (0..4)		SVX Hit
..					...		
x	HOLD_	DS_	EE_	EP_	Layer (0..4)		SVX Hit
x+1	HOLD_	DS_	EE_	EP_	Layer XFT		XFT 1st word
x+2	HOLD_	DS_	EE_	EP_	XFT 2nd word		
..					...		
x+2n+1	HOLD_	DS_	EE_	EP_	Layer XFT		XFT 1st word
x+2n+2	HOLD_	DS_	EE_	EP_	XFT 2nd word		
x+2n+3	HOLD_	DS_	EE_	EP_	Road ID		

Table 5.1: HitBuffer++ to GigaFitter packet format

	24	23	22	21	20	19	18	17 16	15 14 13 12 11 10	9	8 7 6 5 4 3 2 1	0
1	HOLD_	DS_	EE_	EP_	1	0	z out-in		phi			
1	HOLD_	DS_	EE_	EP_	road		sign	c		sign	phi	
1	HOLD_	DS_	EE_	EP_	phi sector			road				
1	HOLD_	DS_	EE_	EP_		x1				x0		
1	HOLD_	DS_	EE_	EP_		x3				x2		
1	HOLD_	DS_	EE_	EP_	χ^2					x4		
1	HOLD_	DS_	EE_	EP_	GF status						track num	

Table 5.2: GigaFitter to GhostBuster packet format

5.4 Internal structure and algorithm

The GigaFitter is structured with a modular design: in each mezzanine FPGA there is one independent processor for each GF input, for a total of four independent engines for four SVT wedges. The track output of four wedges is merged in the mezzanine FPGA and collected by the Pulsar FPGAs. Tracks from the 12 wedges are merged in the motherboard in the single output of the board (figure 5.8). The system is very flexible: an arbitrary number of inputs (wedges) can be activated, a feature that was extremely useful during the developing and commissioning phase.

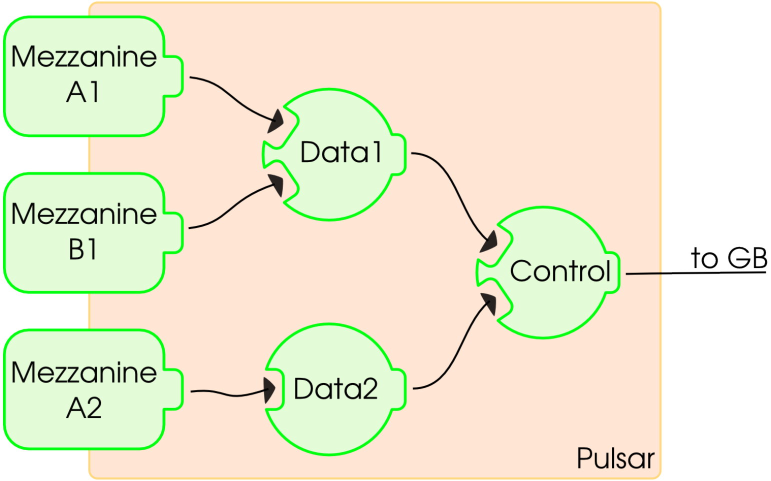


Figure 5.8: GF Pulsar Scheme - The tracks found in each mezzanine are merged inside the three Pulsar FPGA: Data1, Data2 and Control. The final stream is sent on one SVT cable downstream to GhostBuster board.

Inside each mezzanine FPGA there are four track processing modules and a merger module (figure 5.9). Inside each Pulsar FPGA there is an equivalent merger module. All FPGAs, of both Pulsar and mezzanine, have VME modules that communicate with the VME CPLD on the Pulsar board; they are used to set all the needed configurations (initializing functions), to monitor the status of the

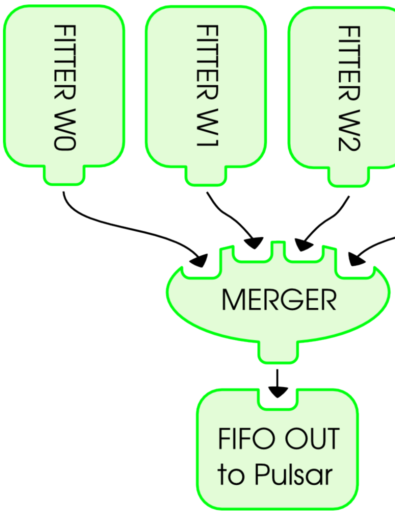


Figure 5.9: GF Mezzanine Scheme - The internal structure of the GF mezzanine: four parallel fitter engines compute tracks from one wedge each, a final unit logic (Merger) merges the four data streams in a single output FIFO that communicates with the Pulsar motherboard.

board and for debugging purposes.

5.4.1 The merger module

The same merger logic is used inside the mezzanine FPGA and Pulsar FPGAs to merge the various output data streams in a single stream. The merge is done in a simple and predictable way ("deterministic merge"): the inputs are ordered and the one with higher priority is read until an end event packet is found. Then the next is read and so on; after all inputs reach an end event packet a final end event packet is sent to the output. The event tags in the end event packets are used to check that all data streams are correctly synchronized. If the sequence of end events is not correct in a stream a severe error (Lost Sync) is set. The error bit fields of the various input end events are ORed in the final end event packet sent to the output.

The "deterministic merge" is not optimal if the data stream occupancies are very unbalanced. If data do not arrive roughly at the same time on different streams, reading them in a predetermined order can be inefficient. A first-in-first-out fashion would be more efficient saving time, but the track order in the output will be unpredictable for the simulation. Their order would depend on timing details that are not available in the simulation. However the extra latency has been measured to be a small effect since the GF is working at a much higher clock frequency than the final output. In conclusion the output is exactly predictable by the simulation.

5.4.2 The track processing module

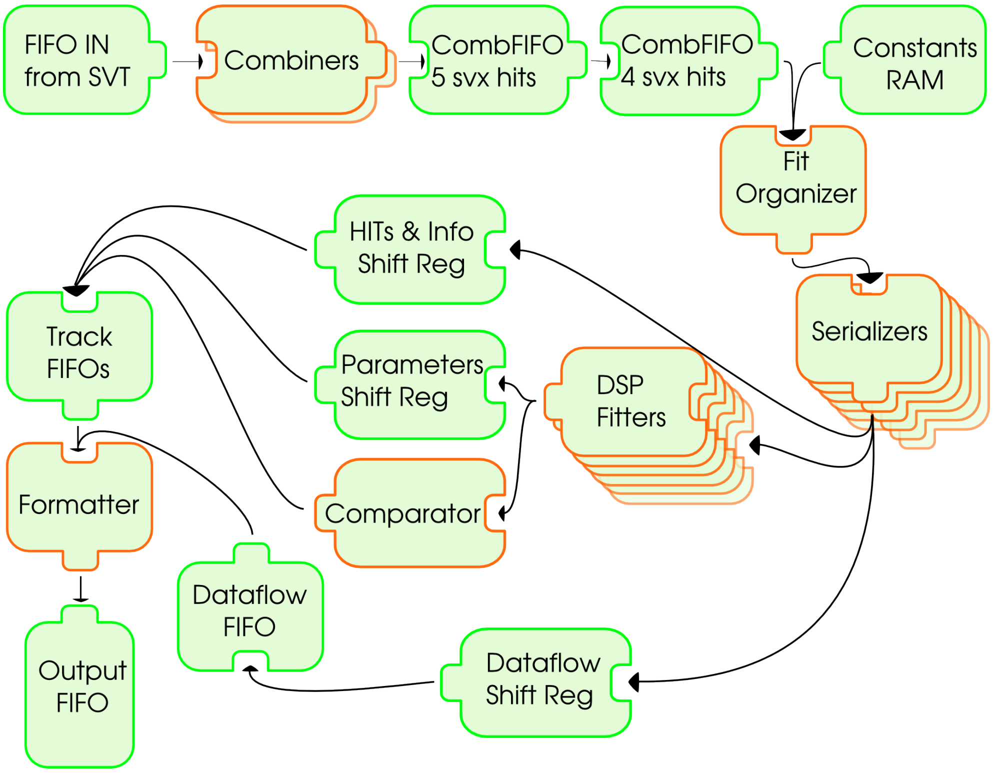


Figure 5.10: GF Fitter module - Schematics of the GigaFitter fitter module.

The track processing module is naturally divided in six different parts: the "Combiners", the "Fit Organizer", the "Serializers", the "DSP Fitters", the "Comparator" and the "Formatter". Large RAMs are used to store the fit constants, FIFOs are used to interconnect the various stages of the pipeline and certain number of shift registers have also been added to shift data downstream the pipeline that is fully synchronized. Figure 5.10 shows all these parts and their interconnections. It is more complete and has improved performances with respect to the first ideas presented in (7).
The Combiner provides combinations of input hits to be tested by the fit. The Fit Organizer coordinates the fetching of hit combinations and relative constants and starting of Serializers. The DSP Fitter performs the fit. The Comparator judges the fit results and selects the best choice in the case of multiple fits. The Formatter provides the right data format to the output, exactly the same of the TF++ output.

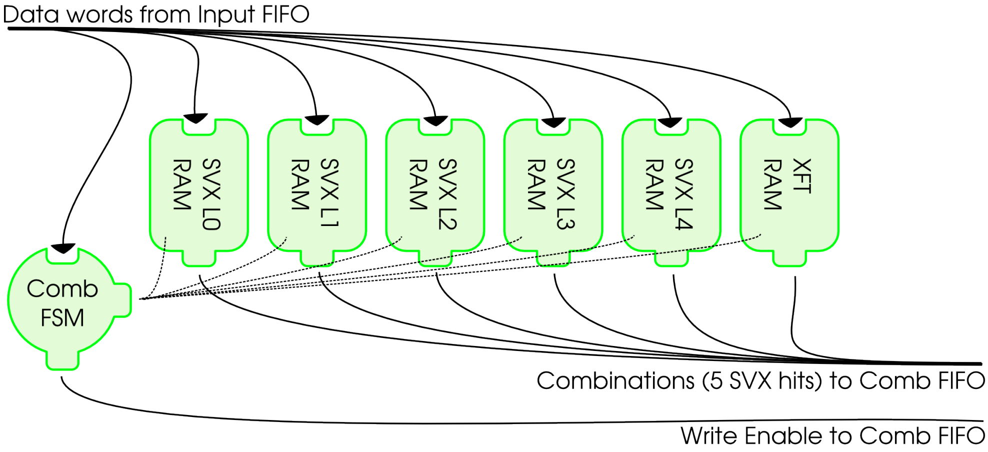

Data words from Input FIFO
SVX L0 RAM
SVX L1 RAM
SVX L2 RAM
SVX L3 RAM
SVX L4 RAM
XFT RAM
Comb FSM
Combinations (5 SVX hits) to Comb FIFO
Write Enable to Comb FIFO

Figure 5.11: GF Combiner module - The Combiner is made by five RAMs and a finite state machine that control both writing and combination pop.

The Combiner works in two subsequent steps: in the first step it pops road packets (road ID and list of relative hits) from the Input

FIFO (figure 5.11) and stores them inside small RAMs (32x19 bit each, implemented in the distributed memory of the FPGA), one for every layer. Counters keep track of how many hits are recorded in each layer.
After all hits have been loaded it starts the second step: processing the road. A road can have more than one hit per layer and every hit can belong to a track; for this reason the Combiner forms the candidate tracks by generating all the combinations that can be done with the road hit list. Using the counters information to generate RAM addresses it fetches hits from the RAMs (one per layer) in parallel to create one hit combination at each clock cycle until all combinations are fetched.
There are two independent "Combiners", each one provided of its set of RAMs, working in parallel. While one is processing one road, the other can pop and load hits of a following road from the Input FIFO. Both the Combiners work full time to provide a continuous flux of combinations that are stored in a large FIFO called Combination FIFO.

The Combiner generates combinations as vectors of seven coordinates: hits from the 5 SVX layers plus the XFT layer with two coordinates (curvature and ϕ). If in the road there wasn't hits in an SVX layer the corresponding place in the vector is zeroed and a bitmap is set accordingly to which layer is missing (called "hitmap"). This kind of roads and combinations are called 4/5, while the ones with hits in all SVX layers are called 5/5.
The fit constants are made for 4 SVX layers and thus all the other stages in the GF Track Processing module works instead with vectors of six coordinates, using always 4 out of 5 SVX layers. For this reason there are two pipelined Combination FIFOs: in the first the Combiner writes the combinations in seven coordinates format. A simple module reads from the first FIFO and convert the seven coordinates format to six coordinates format accordingly to the hitmap

and write them in the second FIFO.
As said in 5.1.3 the TF++ choose a fixed combination of layers when a 5/5 combination has to be fitted, the GF instead fits all possible five combinations of layers choice. This is done by converting every 5/5 combination in seven coordinates format to five combinations in six coordinates format.

The Fitter Organizer pops combinations out of the Combinations FIFO and completes them with the fit constants retrieved from a private large RAM. Each set of constants is a 756 bit word (7 18-bits terms in each scalar product to be multiplied by 6 scalar products). The RAM, implemented using the memory blocks embedded in the chip (BlockRAM), provides space for 256 sets (756x256 RAM). Different layer conditions (which are the involved barrels, zeta in and out, and missing layers, encoded in the "hitmap") and quality of the hits (long clusters are flagged by the Hit Finders as low precision points, encoded in the "lcmap") require in principle different sets of constants. The right constants are fetched taking into account of all this information: which layers the used hits belong to and their quality. Using this information directly to address the constants RAM would require 13 bit addresses (6 bit to encode zeta inner and outer barrels, 3 bit for hitmap and 4 bit for lcmap), but the physically relevant configurations are only 240 thus a two-RAM system is used: a first 8x8k bit RAM addressed by the 13 "condition" bits provides the address for the second one, the 756x256 constant RAM.
The whole set of hits and the associated constants are extracted in parallel in a single clock by the Fitter Organizer that also sends a start signal to the Serializers. Each Serializer can accept one combination every 6 clock cycles so there are 6 parallel Serializers and the Fitter Organizer keep track of which one has to handle the next fetched hit combination.

Each Serializer registers all the hits and constants in a single clock, then serializes them associating each hit to the corresponding term in the constants set and sending one hit-constant pair for clock cycle to its own associated DSP Fitter.

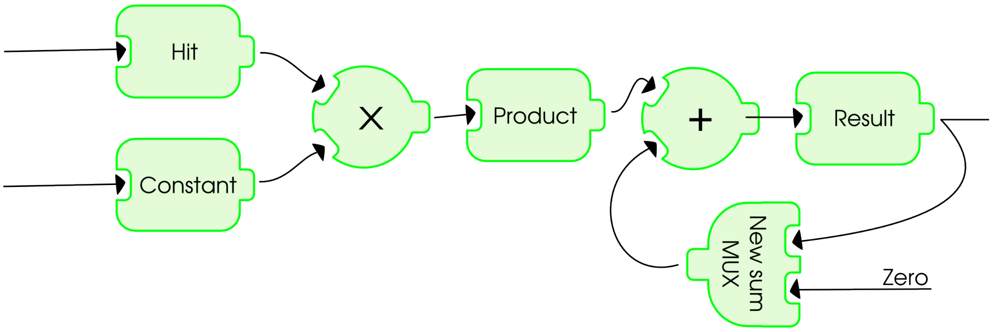


Figure 5.12: GF DSP Fitter unit - The scalar product unit is made inside the specialized DSP48 unit configured in MACC (multiply-and-accumulate) mode. Each unit can compute a scalar product in 6 clock cycles. There are 2 additional clock cycles of latency before the result appear in output, but the unit is already ready to compute the next scalar product.

The DSP Fitter receives the hits and constants data and calculates the track parameters and the fit quality parameters (χ^2 components). This function requires the computation of 6 scalar products, but it is executed in parallel by exploiting the large number of on-chip DSPs of the Virtex 5 device. The scalar products are performed configuring the DSP processor as MACC (multiply and accumulate) and serially processing the hits. The products of a 6 term scalar product are calculated and accumulated sequentially using 6 clock cycles (figure 5.12. In a DSP Fitter there are six DSPs, each one able to compute a fit parameter (c, d, ϕ) or one of the three χ^2 components. Thus with the six DSP Fitters, for a total of 36 DSPs, and the associated Serializers the GF is able to process one combination every clock cycle.

Once the results are ready, the χ^2 components are sent to the Comparator (figure 5.13) while the track parameters obtained by the fit and the used hits are stored in the shift registers waiting for the Comparator decision. The additional information provided by the Combiner at the very beginning but not used in the fit, has been maintained in shift registers to be provided to the Comparator at the right time. This is particularly important when different fits of the same track have to be judged to choose the best one. As already mentioned, in fact, the GF has the capability (7) to fit many times one track that has hits on all layers ("full of hits track" or 5/5 track) deleting one particular layer in each different fit and finally choose the best. The Comparator has to fine-tune the final decision using not only the χ^2, but also the hit combination layout (used layers and quality of the hits).

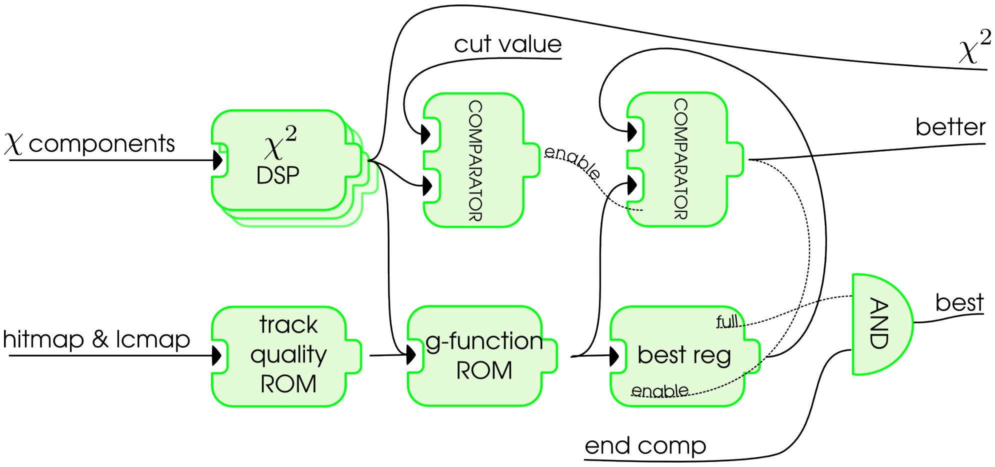


Figure 5.13: GF Comparator unit - The comparator compute the χ^2 of the tracks from χ components, apply the χ^2-cut selection and compare the track with the previous one finding the best.

The Comparator has the ability to choose the best track of an arbitrary sequence of tracks. The control bits from the Fitter Organizer going to the Comparator are thus set to consider the five 5/5 tracks as a such sequence, while the 4/5 tracks are considered as a one-track sequence. This system is flexible and we could use it to consider all combinations of the same XFT hit as a single sequence implementing a sort of Ghost Buster suppression inside a road. This feature is ready be implemented in future revisions of the firmware. The Comparator calculates the final χ^2 using a DSP in MACC configuration like the one used in DSP Fitter units (figure 5.12). Three clock cycles are necessary for each track and there are three of such units to sustain the output rate of one track candidate every clock cycle. It compares the result with the threshold configurable via VME. If the track pass the χ^2-cut its χ^2 and the track quality (a function of used layers and single hit quality) are used to compute the g function (as in goodness) which is compared with the g of the best track in the sequence. If it's better a signal is sent to update the registers that store the best track (parameters, χ^2 and additional informations). Once the sequence is finished if there was at least one track passing the χ^2 cut a best track found signal is used to store the best track in the Track FIFOs.

Finally the Formatter reads the parameters and the χ^2 of the accepted tracks from the Track FIFOs and merges all this information with the hits, the road identifier, and some status data, pushing them to the output in accordance to the SVT protocol.

5.4.3 Debug features

Diagnostic and debug is a very important aspect for developing, commissioning and monitoring the status of the board during the normal operation. As described in 4.1.4 this aspect has been a key factor of the success of SVT. Also in the GF board we implemented the standard SVT debug feature: the spy buffers.
The GF board is unique in SVT: it has 12 inputs, one output and perform the task that was previously done by 16 boards. For this

reason the standard spy buffers at the end of input and output cables were not enough for fully monitor and diagnostic the GF. It was necessary to attach spy buffers also at the end of each "internal SVT cable" (figure 5.14): at the output of each track processing module inside the mezzanines (logically corresponding to the output of each TF++), after the merging inside each mezzanine and after each merge unit in the Pulsar board. This resulted in 30 spy buffers (12 input and 12 output, 3 mezzanine outputs, 3 Pulsar FPGA outputs), an unprecedented record for an SVT board. The monitoring software was flexible enough to add all this spy buffers to the code without much effort.

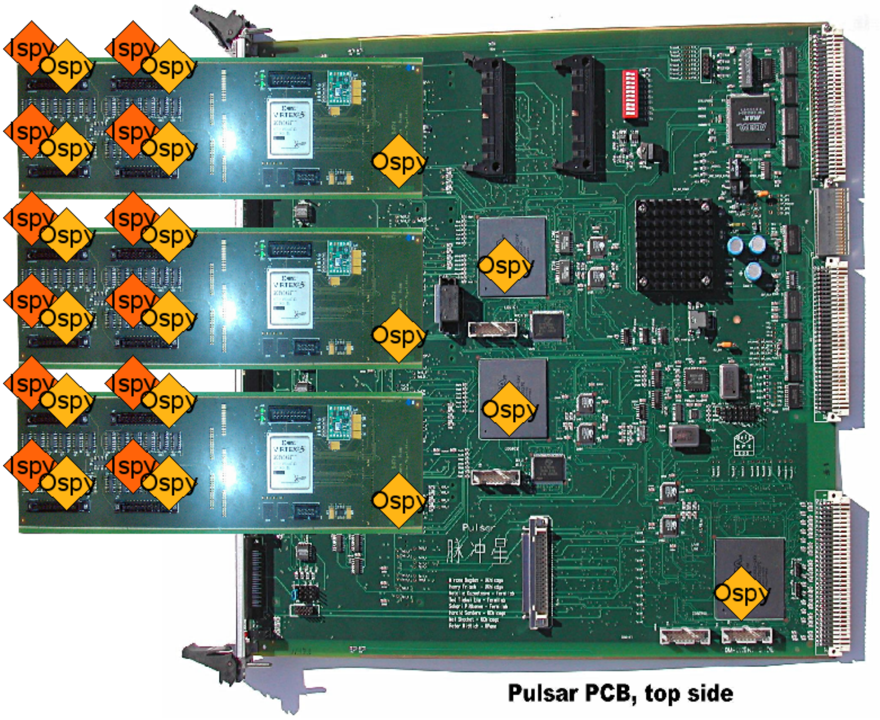


Figure 5.14: GF Spy Buffers - The GF board is the most complex board of SVT in terms of diagnostic features: it has 30 spy buffers, one for each input and output of track processing module and one for each output of merger units.

There are also error registers that keep track of various kind of errors (fit overflow, FIFO overflow, invalid data, etc.) for each track processing module and for each merger module. Those registers are readable via VME to investigate online the status of every GF component. There are several registers to configure the severity of each error. Normally errors are include in the EE word and registered in VME registers, locally where they happened. If an error is set as "severe" it also rises the standard SVT_ERROR and CDF_ERROR lines on VME backplane to either freeze spy buffers of all SVT or perform a reset of the DAQ system.
Another tool that was extremely useful for in depth debug of the GF Mezzanine firmware is the ChipScope tool from Xilinx. ChipScope is a suite of firmware modules (cores in the Xilinx jargon) and a standalone PC software. With the firmware cores it is possible to insert a custom in-chip logic analyzer and pattern generator fully controllable via the JTAG programmer cable. Using ChipScope it was possible to analyze lots of logic lines (all 756 bit constants, 15 bit hits and 48 bit partial fit results at once, for example) without be limited to the number of debug pins that were available on the PCB (a 20-pin connector for the Mezzanine) for external logic analyzer use. Bringing debug signals to output pins also may degrade timing performances of the firmware due to increased routing complexity. The ChipScope features are disabled in the stable version of the GF Mezzanine firmware.

5.5 Parasitic mode for GF studies

A very careful test procedure was devised to reduce to a negligible level the impact of the commissioning on detector operation and functionality. The new board has to be deeply tested before being allowed to enter the experiment. The test steps are described below:

1. The first validation is performed on a stand-alone test stand using a Merger board to send and receive data from the GF board, where random or sampled from data events are sent through the GF and the output is compared to board level simulation.

2. A second level of validation is performed using real data spied from the experiment (parasitic mode). A test crate, placed near the real SVT, is configured to duplicate the track fitting function.
   The outputs of all HB++ boards are spied using 6 Mergers and 3 Splitters (see section 4.1.1) additional boards. The hold signal is set to be ignored for the second output in order to avoid back pressure on the SVT from the GF test crate. A GB board is in the crate to receive GF output and perform all the final correction exactly as the real SVT. In this parasitic configuration, highlighted in figure 4.3, it is possible to process the same data as the TF++ boards without interfering with normal data taking and thus to make a direct comparison between the current system and the upgraded.
   The output of the crate can be compared directly with that of the SVT final crate receiving identical input data. At this step all discrepancies between the two systems are completely understood. The same inputs are also fed to a board-level simulator. Comparison between the hardware and the simulation are used to validate both the board and the upgraded simulation itself.
   At this stage the efficiency and failure rates are monitored. The efficiency is defined as the fraction of SVT tracks matching more precise and complete Level 3 track reconstruction within SVT acceptance and is measured to be 80%. The failure rate is defined as the fraction of reconstructed tracks that do not match to simulated tracks. The failure rate must

be as low as possible for two reasons: (a) failures can be a symptom of hardware problems, and (b) the simulation must be able to reproduce the hardware in a very detailed way for purposes of data analysis, where the simulation is used to understand various efficiencies. Normal SVT allows for a failure rate of the order of 10^{-3} in the whole system, but for the GF board an exactly zero failure rate has been required in order to fully validate the new hardware.
From this step the standard SVT monitoring (spybuffers), which runs on crate processors, is also used for validation. It is monitored the impact parameter, the angle in the transverse plane and the track transverse momentum distributions.

3. After successfully passing step 2, the new GF board can replace the old TF++s for a short, low-luminosity test, and, after being successful, for data taking at any luminosity in a whole store. This final test is important because it checks that the control signals used by the DAQ system do not interfere with the board functionality and vice versa. Furthermore, it is an extra validation with higher statistics than the previous tests, allowing for detection and debugging of lower rate errors.

4. The GF had to work correctly in data taking both at low and high instantaneous luminosity before being considered ready to be installed and remove all the TF++: a commissioning period of one week of GF data taking in the final position (not in the test crate) is required before decommissioning and remove TF++.

5. The HB++ splitted outputs are not removed after the commissioning. The test crate will be used to validate the spare GF boards and also to quickly develop and test new improvements to the system.

The commissioning procedure was completed during February 2010 and at the beginning of March the old TF++ was phased out. The GF system is now the track fitter of SVT.

6

GigaFitter performances

6.1 Timing

SVT processing time is determined by multiple factors. Each wedge has its own pipeline of boards, the Hit Finder, the AMSRW/AM, the HB++ and the TF++, and each one of these boards adds its own latency proportional to the complexity of the event. A significant fraction of the processing time of each wedge is due to data transfer, that often is slower than data processing (I/O timing): all data goes from one board to another using the standard SVT cable at 32 MHz (certain boards can run up to 40 MHz, such as HB++ output). The transfer time depends on the number of present hits, the number of found roads and the number of tracks that pass the quality and χ^2 cuts. Furthermore the global timing is dependent on the merge time of the twelve wedges before the GB processing. Each time two streams are multiplexed the output frequency should double to avoid loosing time in the process. Instead in SVT the Merger output frequency is the same as its own four inputs and the final GB

output frequency is 20 MHz, slower than any other transfer rate.
In this context it is not easy to spot the real timing performances of the GF board: the GF receives and sends data at the same speed as the TF++ and the possible gain due to the much higher track fitting rate provided by the GF is hidden under the bottlenecks caused by the downstream and upstream boards. The GF will do quickly its job but will spend some time inactive waiting the inputs. Its output will lie around in the FIFO waiting to be transferred. For this reason the global event processing time across the GF path is almost the same as the timing across the TF++ path. If, for example, there is an event with few tracks in most of the wedges and a lot of tracks in one wedge the TF++ of the wedge with lots of tracks will be much slower than the GF Track Processing module for the same wedge but this difference is hidden due the the time necessary to deliver roads to the track fitters and to merge the few tracks of the other wedges.
The comparison of the global SVT timing with TF++s and with GF inserted in the pipeline is shown in figure 6.1. The time is measured between the level 1 accept (the start for level 2 processing and SVX readout) and the last event word to the GB output. The distribution looks very similar for the two configurations.

Unfortunately it's not possible to measure the timing between the GF/TF++ input of a specific road and its tracks output. We can measure only event global timing between the first road arrival at both tracking devices and the last track output at the very end of the GF Pulsar, when the data streams are all merged. It's not possible to highlight directly the timing difference between the single TF++ and the logically equivalent module inside the GF Mezzanine FPGA.
Anyhow a certain difference in the timing appears on some classes of events: the ones with many hit combinations to be checked. In figure 6.2 is shown on the left the mean processing time versus the

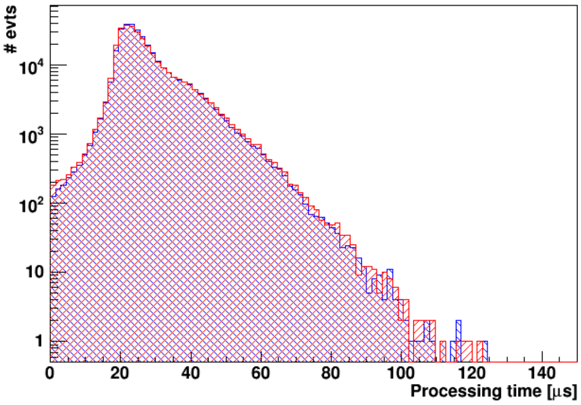


Figure 6.1: Global SVT Timing: GF and TF++ - The plot shows the processing time distribution as measured by the GB boards, the one of the normal SVT system in blue and the one of the test crate with the GF board in parasitic configuration in red. Data is taken for both systems at the same time on the same events.

total number of hit combinations summed over all the wedges. On the right there is the fraction of events with timing falling in the tail section of the global timing distribution (> 50 μs).

The growth in figure 6.2 for both the TF++ and GF timing depends on the increasing number of roads to be transfered to the track fitters and by the number of found tracks to merge. Both these quantities grow when the number of combinations or fits increase. In the low number of combinations range both systems (TF++ and GF) are limited by the input rate from the HB++ and the timing is absolutely similar, instead when the events have roads with a large number of combinations (approximately over 64, which is the limit of roads per wedge, meaning that on those events roads are carrying more than one combination) a difference appears even if both systems are still limited by the merge rate.
The GF is significantly faster when the event complexity goes up. The difference of the fraction of events with long processing time

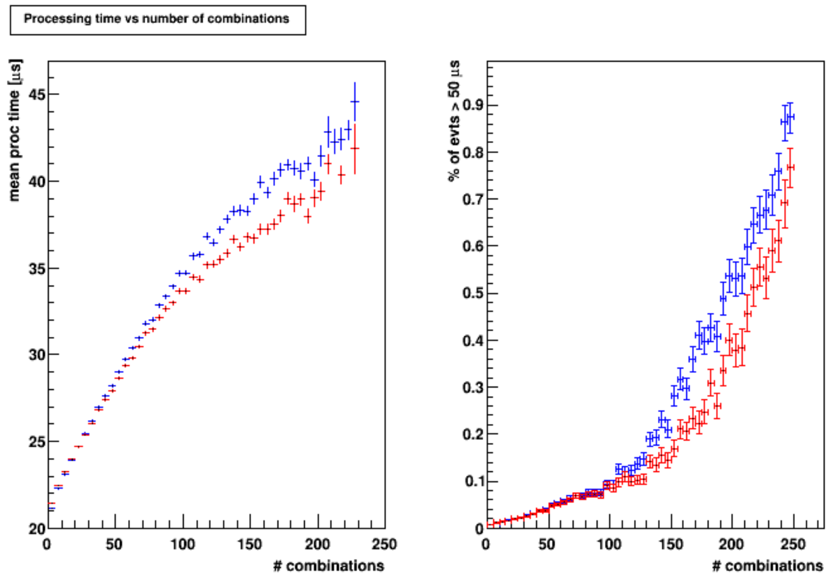


Figure 6.2: SVT Timing vs Number of hit combinations - The standard SVT system is in blue, the GF parasitic test crate is in red. On the left is shown the mean processing time with respect to the number of processed combinations. On the right the fraction of events with processing time > 50 μs. The processing time in these plots is computed subtracting I/O time to GB board and from GB board to L2, which is linearly dependent with the number of found tracks.

reaches about 20% in the 160-170 combinations range. This is an important feature, more than the lower mean processing time. The tails in fact have a direct effect on the deadtime of the DAQ system. All level 2 buffers can become full loosing next events.
If the HB++ bottleneck in input or the output bottleneck to GB could be removed the performance improvement will be much more visible.
We have proposed the collaboration to remove GB bottleneck incorporating all GB functions (duplicate tracks removal, beam position fit and subtraction, ϕ second order corrections) in the GF Pulsar. The GF Pulsar could then communicate directly to the L2 PC using fast S-LINK connections and fully exploit the higher internal Pulsar merge rate. Some of the GB functions for the GF Pulsar are ready to be implemented in future revisions of the firmware.
It was also prospected to the collaboration to double the HB++ output rate using a mezzanine similar to the GF Mezzanine, provided of outputs instead of of inputs. The HB++ internal processing rate is double with respect to the output rate and using such mezzanine in the free slot of the HB++ would allow to use a double rate output. This system would require also a GF system made by two Pulsars with three mezzanines each to double the inputs.

A figure of the GF system power is shown in figure 6.3 were is plotted the mean timing with respect to the number of fits performed. The GF is performing much more fits than the TF++ system even if the number of combinations to be processed is the same for the two processors, because the GF fits five times every full of hits (5/5) track and then choose the best fit, without adding latency to the system.

The GF is not expected to lower the SVT timing because of the said bottlenecks and also because of the current pattern banks and road size: the current system is tuned to be balanced with respect to

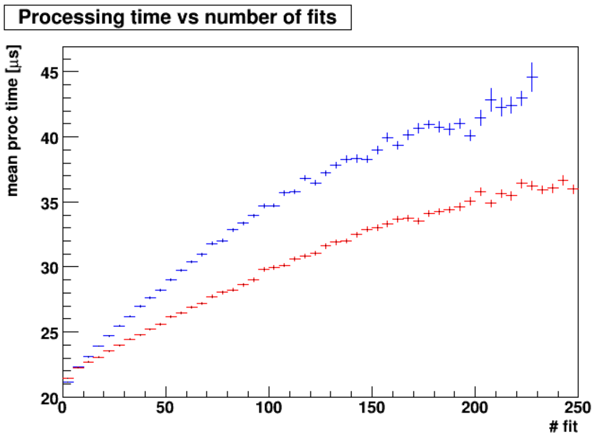


Figure 6.3: SVT Timing vs Number of fits - The standard SVT system is in blue, the GF parasitic test crate is in red. The mean processing time versus the number of fits done is show, the GF does much more fits than the TF++ system because it fits 5 times every 5/5 track.

the computing capabilities of each component of the pipeline. The processing time spent in the TF++ is limited by the small road size that limits the maximum number of hit combinations per road. The GF however allows now to change freely the road size because it's able to sustain a large number of fits without impacting much on the global timing. The GF is in fact able to do one fit every clock cycle at 120 MHz on each wedge once a road is loaded. Even if the number of combinations per road increases a lot the impact on timing is small.

6.2 Efficiency studies

The performances of the SVT algorithm largely depends on the used data banks: the pattern bank and the constants sets.
SVT can reconstruct only tracks that match a road in the pattern bank. A finite size pattern bank has a certain geometrical acceptance called coverage that limits the efficiency of SVT. The current

AM++ boards in SVT can store 640k patterns per wedge, but only 512k patterns are used due to hardware limitations in the TF++, the GF does not have this limitation.

A parameter to tune the coverage of a bank given a fixed number of patterns is the size of the road, that is the number of microstrip hits or XFT c and ϕ bins that makes a superstrip. The larger the road size the higher the coverage of the bank, but it also increases the number of combinations to process inside each road as more hits can be contained in each superstrip. The current road size is tuned to balance coverage and workload of the track fitting stage with TF++ and the configuration used is 4 strips per superstrip in layer 0 of SVX, 3 strips per superstrip in layers 1, 2, 3, then 4 strips per superstrip in layer 4 and 6 bins for XFT (this configuration is called "433346").

The pattern bank is created generating tracks (or reading them from data) with certain parameters distributions: the current bank is made from tracks with $p_T > 2$ GeV/c and a beam spot radius of 0.14 cm. Tracks crossing mechanical barrels are not generated, thus no patterns for those tracks are in the pattern bank. The larger is the parameter acceptance (for example lowering p_T threshold to 1.5 GeV/c or allowing mechanical barrels crossing) the lower will be the coverage at a fixed number of patterns and road size.

The constants sets are generated from tracks with the same parameter distributions as the patterns but with all hits in the same barrel. It's assumed that the tracks crossing electrical barrels, for whom the patterns exists in the current bank, are reconstructed well enough by the constants trained for no barrel crossing.

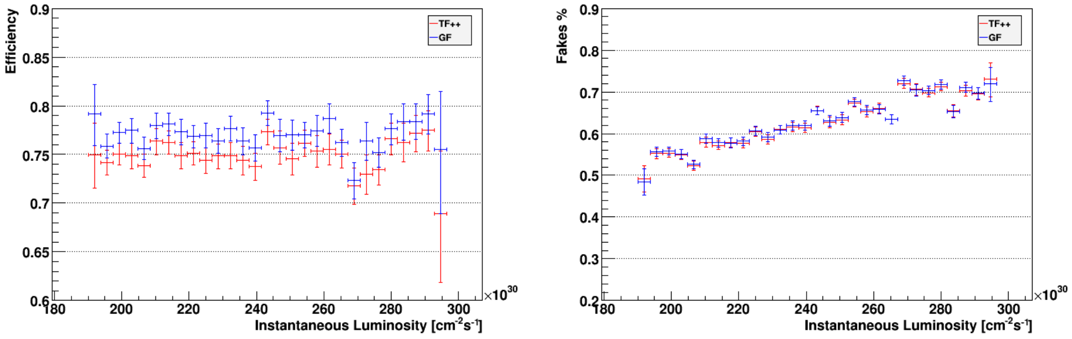
The used data banks have also an effect on the timing performances of the hardware (especially the pattern bank) as an increase in the number of found roads by the AM++ or in the number of tracks found by the track fitter leads to increased I/O transfer time that, as shown in 6.1, is a significant contribution to the total processing time of SVT.

6.2.1 GigaFitter performances with current SVT data banks

The used data banks have a different impact on the performances of SVT whether the GF or TF++ is used for the track fitting stage: the full precision fit done by the GF (described in 5.1.1) leads to a different χ^2 with respect to the same computation done by the TF++ (see figure 5.1) and thus a more accurate selection of good tracks. The handling of 5/5 tracks by the GF (described in 5.1.3) recover tracks that are discarded by the TF++ because of the wrong 4/5 hit combination to fit chosen a priori.

The use of the GF leads to improved performances even with the current data banks tuned for TF++ usage. In figure 6.4 (a) is shown the efficiency[1] for the TF++ and GF paths as a function of the instant luminosity ($290 \cdot 10^{32}$ to $190 \cdot 10^{32}$ cm^{-2}s^{-1} high luminosity range of a recent CDF store): the use of the GF leads to a significant increase in the efficiency of about 2% (see table 6.1). The fake rates are instead very similar for both systems as shown in figure 6.4 (b) as a function of the instantaneous luminosity.
The GF finds about 45% of 5/5 tracks more than the TF++ system for a total of 3% more tracks at the track fitting output. This is the main factor that contributes to the higher efficiency in this high luminosity range. The increase is shown in figure 6.5 with respect

[1] Efficiency is defined as the ratio between SVT tracks matching a L3 track and all L3 tracks with at least 4 SVX hits, $p_T > 2$ and a matching XFT track. Matching between SVT and L3 is defined as $|curv_{SVT} - curv_{L3}| < 0.0002$ and $|\phi_{SVT} - \phi_{L3}| < 0.02|$. This definition differs from the standard SVT efficiency as no geometrical acceptance cut is applied to L3 tracks. The global efficiency is used instead of the standard efficiency limited to the geometrical SVT acceptance region in order to highlight the improvements.



(a)

(b)

Figure 6.4: SVT (GF, TF++, standard data banks) Efficiency and fake rate at high luminosity - (a) Efficiency vs instantaneous luminosity: in red SVT with TF++ in blue SVT with GF. (b) Fake rate vs instantaneous luminosity with the same color code.

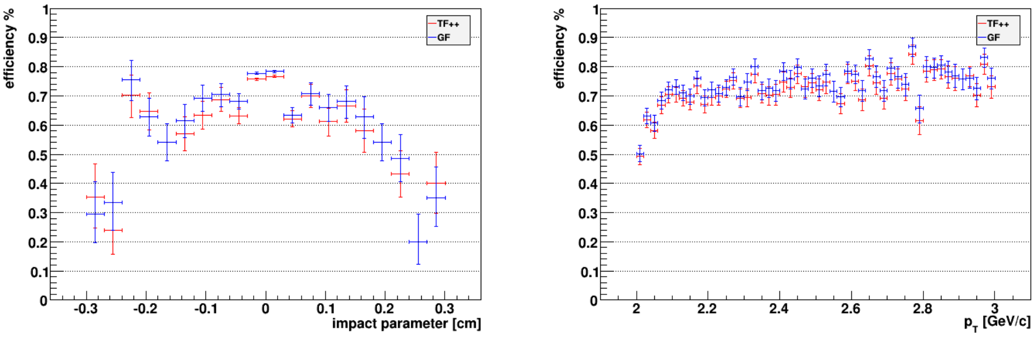
to impact parameter and p_T.

6.2.2 GigaFitter performances with new SVT data banks

The GF board is powerful enough to sustain an elevated number of combinations per road, it has a better 5/5 fit mechanism and has the ability to load a more comprehensive collection of constants to fit with better precision tracks that cross barrels. This facts open the way to develop new SVT data banks, patterns and constants, to enhance the SVT performances.

6.2.2.1 Recovering tracks that cross mechanical barrels

The figure 6.6 shows the efficiency as a function of the zeta (z) and $\cot\theta$ of the tracks. It's visible a significant loss of efficiency in four



(a)

(b)

Figure 6.5: SVT (GF, TF++, standard data banks) Efficiency vs impact parameter and p_T - (a) Efficiency vs impact parameter: in red SVT with TF++ in blue SVT with GF. (b) Efficiency vs transverse momentum with the same color code, zoom in the region $2 < p_T < 3$ to highlight the turn on at low p_T.

Data banks	TF++		GF	
	Efficiency	Purity	Efficiency	Purity
Standard	**75.0%**	37.9%	77.0%	37.6%
Standard ord	72.7%	38.1%	77.1%	37.3%
433346 BC	75.7%	37.2%	77.7%	37.4%
544446 BC (*)	77.4%	35.7%	80.0%	35.5%
544446 BC ord (*)	75.1%	36.0%	**80.2%**	35.1%

Table 6.1: SVT Average efficiency and fake rates with GF and TF++ on the instantaneous luminosity range from $290 \cdot 10^{32}$ to $190 \cdot 10^{32}$ $\mathrm{cm}^{-2}\mathrm{s}^{-1}$. “BC”: pattern bank contains patterns for barrel crossing tracks. “ord”: AM++ ordered output activated. (*) banks not suitable for TF++ use because of processing time issues. All banks have 512k patterns.

regions of the $(z, \cot\theta)$ plane. There's where tracks that from one mechanical barrel cross to the other: there are three barrels one after another along zeta, so there are two boundaries between mechanical barrels. A particle can cross that boundary in one direction or the opposite one, accounting for the four regions of efficiency loss.

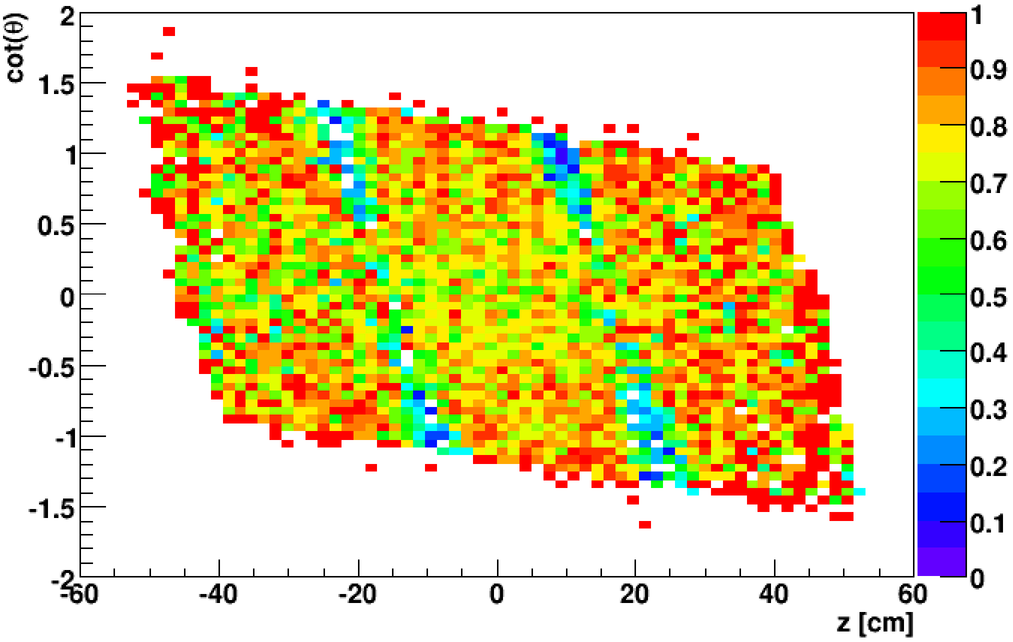


Figure 6.6: SVT Efficiency vs $\cot(\theta)$ **with standard banks** - The efficiency is computed with TF++, with GF it's identical. There are four zones where the efficiency suddenly drops. Those are where tracks cross two mechanical barrels.

Those regions are currently excluded in the pattern banks. In order to recover them it is necessary to build a new pattern bank including also the crossing tracks.

The result using a new bank using the same superstrip configuration as the standard bank ("433346"), but including patterns for those tracks, is shown in figure 6.7. The top plots compare the efficiency vs cot(θ) for current SVT banks and new 433346 banks with barrel crossing, the bottom plots shows the gain (only tracks recovered by the new banks) and loss (only tracks disappeared because of the new banks) in efficiency vs cot(θ) for the new banks with respect to the old ones. The gain in the previously missing regions is clearly visible, but the total average gain calculated in the whole acceptance region is small: +0.7%. There is also a small decrease in purity[1] (see table 6.1).

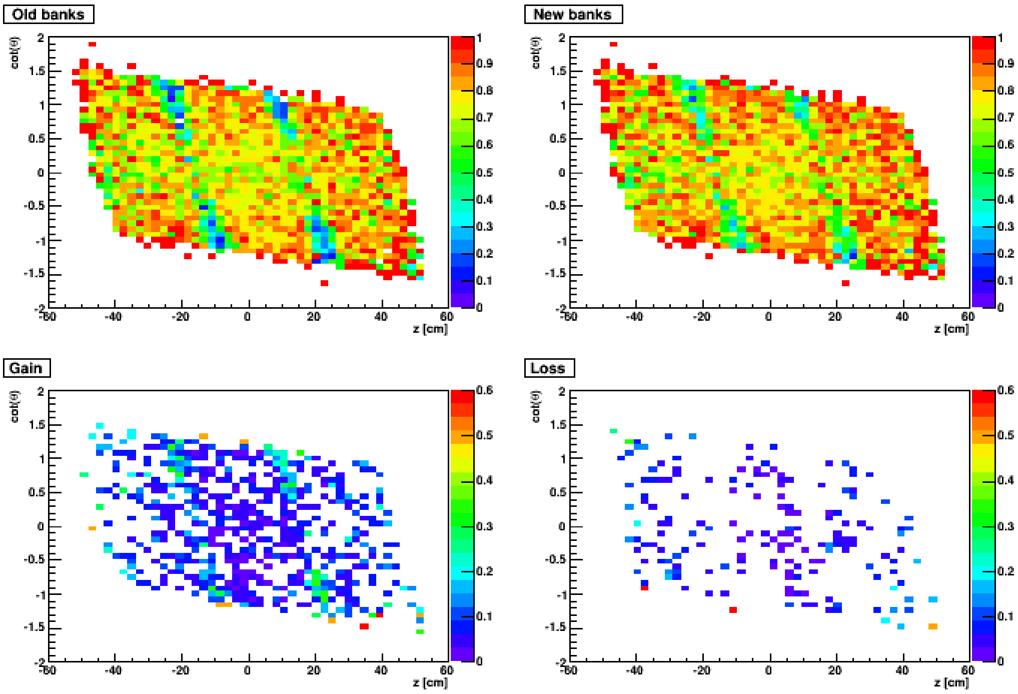


Figure 6.7: SVT Efficiency vs cot(θ) **comparison with new 433346 banks** - The gain in the barrel crossing zones is clearly visible, but the average net gain is very small: +0.7%.

The reason for the small gain in average efficiency lies in the fact that adding new patterns to the bank, the ones that match barrel crossing tracks, while keeping the size of the pattern bank constant,

[1] Purity is defined as the fraction of SVT tracks matching an L3 track.

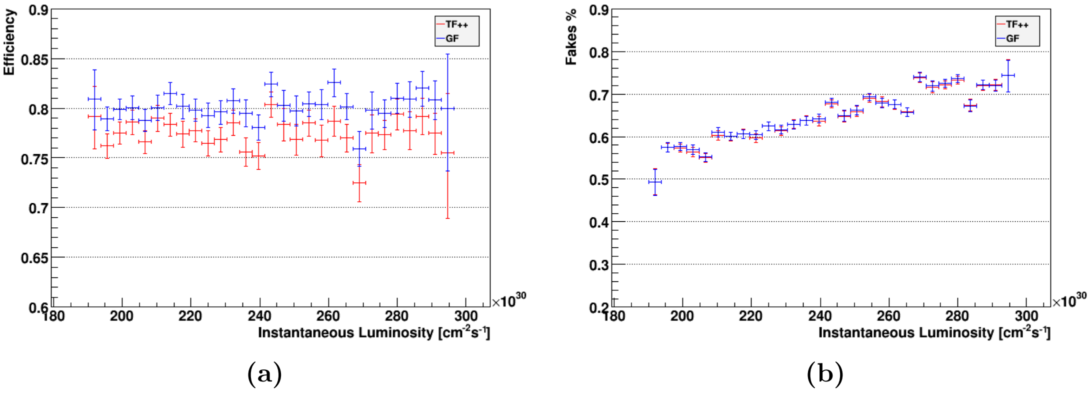
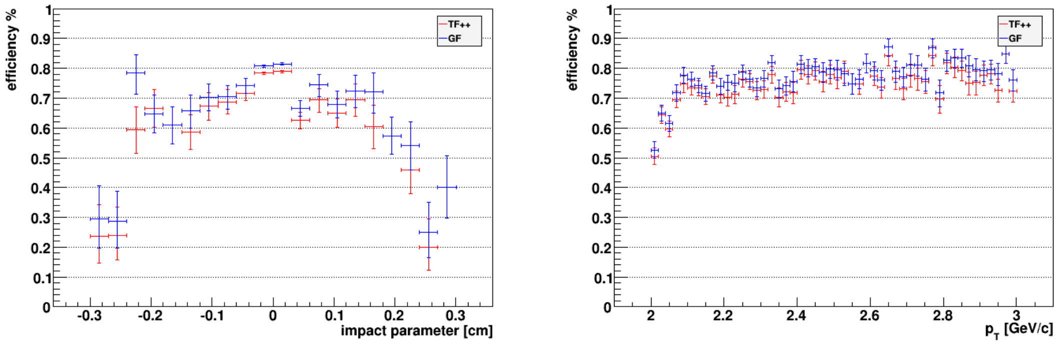


Figure 6.8: SVT (GF, TF++, 544446 data banks) Efficiency and fake rate at high luminosity - (a) Efficiency vs instantaneous luminosity: in red SVT with TF++ in blue SVT with GF. (b) Fake rate vs instantaneous luminosity with the same color code.

produces a minor coverage of the bank in other regions. In conclusion we just obtained a more uniform coverage, adding patterns to the crack regions excluding others from everywhere else. A way to achieve an higher coverage without increasing bank size is to configure larger superstrips: the number of combinations per road will increase and that would have been a problem with TF++ (increased processing time and a maximum of 32 fit per road allowed), but not with the GF.

In figures 6.8 and 6.9 is shown the SVT performance with TF++ and GF using a bank with 5 strips per superstrip for layer 0, 4 strips for layers 1, 2, 3, 4 and 6 c and ϕ bins for XFT, labeled "544446". This bank also includes barrel crossing patterns. The efficiency is shown as a function of impact parameter and p_T in figure 6.8, while figure 6.9 shows the efficiency and the fakes as a function of instant luminosity. Figure 6.10 has the same plots of figure 6.7 repeated using the larger road banks. In particular it is shown the gain in the $(z, \cot\theta)$ plane.



(a) (b)

Figure 6.9: SVT (GF, TF++, 544446 data banks) Efficiency vs impact parameter and p_T - (a) Efficiency vs impact parameter: in red SVT with TF++ in blue SVT with GF. (b) Efficiency vs transverse momentum with the same color code, zoom in the region $2 < p_T < 3$ to highlight the turn on at low p_T.

As it's clearly visible in the plots there is a much bigger efficiency gain of +3% for the GF. The price for this gain is a reduced purity (-2%), which is a tolerable amount also in the prospect that purity is luminosity dependent and at lower luminosity the purity will be the same as the current SVT, while the efficiency is constant for all luminosity regions and so the gain.
Also the TF++ would increase it's efficiency, but less (+2.4%), with the approximately the same decrease in signal purity (see table 6.1).

However, as already mentioned, the 544446 bank is not suitable for use with TF++. As shown in figure 6.11 the number of combinations per road is increased because the road size is larger, causing a too large load for the TF++ that would increase the processing time. The TF++ is also limited to 32 maximum combinations per road. Not all the combinations for some more complex roads will be processed. This partially explains the lesser efficiency gain of the TF++. With the GF there is no such problem: as described in

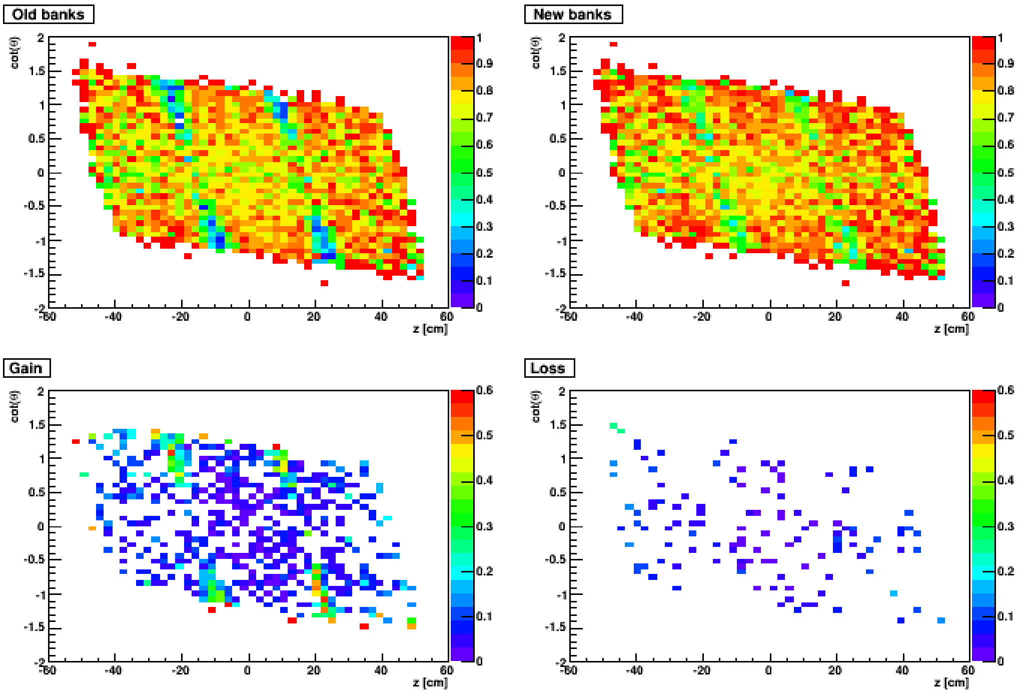


Figure 6.10: SVT Efficiency vs $\cot(\theta)$ **comparison with new 544446 banks** - There is a gain in the barrel crossing zones. The average net gain is significant: +3%.

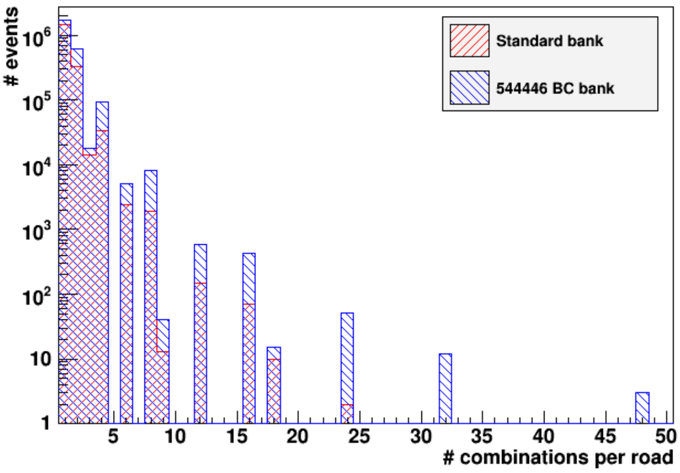


Figure 6.11: Number of combinations per road with standard banks and 544446 banks - The standard bank is tuned in order to keep the number of combinations per road low to contain processing time by the TF++. The maximum number of combinations is also kept under 32 as the TF++ can not fit more than 32 combinations. The new 544446 bank exceeds those limits, but the GF has the capabilities to sustain this number of combinations without impact on the timing and no limits on the maximum fits allowed.

5.4.2 the GF has the ability to fit all the combinations inside of a road, one each clock cycle, at 120 MHz. For example the processing time difference between a road with ten combinations with respect to one with one combination is about 75 ns in case of a 4/5 road and about 373 ns in case of a 5/5 road. The GF has no limits on the maximum number of combinations per road it's able to fit.

6.2.2.2 Ordered AM++ output

Part of the smaller efficiency gain in of the TF++ with the new 544446 banks is also explained by the following fact: due to increased coverage of the pattern bank (larger road size) there will be more real and fake found roads for each event. The AMSRW will eliminate duplicate 4/5 roads that match a 5/5 road already found in the event. If more roads are found more 5/5 are found that are able to delete all the matching 4/5 roads that come after. The events are in average richer in 5/5 roads than with the standard banks so the deletion capability increases.
However the TF++ each time that has to fit tracks of a 5/5 road makes an “a priori” choice of which 4/5 combination has to be fit. This choice is sometimes wrong and the track is rejected even if there was a good 4/5 combination, different to the one chosen, that would be accepted. This leads to a reduced efficiency gain.
The AM++ boards have the possibility to output the roads giving priority to 5/5. In this case the deletion operated by the AMSRW is maximized, reducing also the timing bottleneck of roads transfer from HB++ to track fitters that is linearly dependent on the number of found roads. This feature has never been implemented in the actual AM++ board because it causes inefficiency. It was observed a significant efficiency loss (2% with the standard banks) (23) due to the explained behavior of the TF++.
To moderate the effect of efficiency loss in case of many 5/5 tracks the current AM++ are configured to output roads in the arbitrary

order they are collected from the parallel AM++03 chips in each board. The 4/5 roads that are output by the AM before a matching 5/5 road in the same event are thus not eliminated by the AMSRW.

The GF is not affected by this problem: in case of 5/5 track it will fit always all 4/5 combinations and then choose the best “a posteriori”. Thus the GF allows to enable the ordered output mode of the SVT upgrade AM++ without efficiency loss.

This ability will have also a big effect on the timing performances of these new banks: as seen in 6.1 the I/O time from one board to another has a significant contribution to the total SVT processing time. Reducing the number of roads to be transfered from AMSRW to HB++ and from HB++ to the track fitting stage will reduce the total processing time. Since the HB++ output rate is a known bottleneck of the system its timing reduction is particularly important.

It will also have an effect on the efficiency as the HB++ limits the number of roads to be fitted to 63. Early reduction of useless 4/5 roads decreases the probability to reach the 63 roads limit or cut 5/5 useful roads in very crowded events.
Figure 6.12 compares the distribution of the number of roads per event to be processed in the most crowded wedge when the 544446 bank is used with the same distribution obtained with standard bank and using unordered AM++ output: the larger road size generating the pattern bank causes the increase of the mean number of roads to be processed from 15 to 21. Moreover the number of events with 63 or more roads cut by the HB++ is more than doubled.

Using the same 544446 bank as before, but enabling AM++ ordered output, results in almost the same efficiency and purity for the GF (a little +0.2% gain is observed) while there is a big drop for the TF++: from 77.4% to 75.1% (see table 6.1). As seen in figure

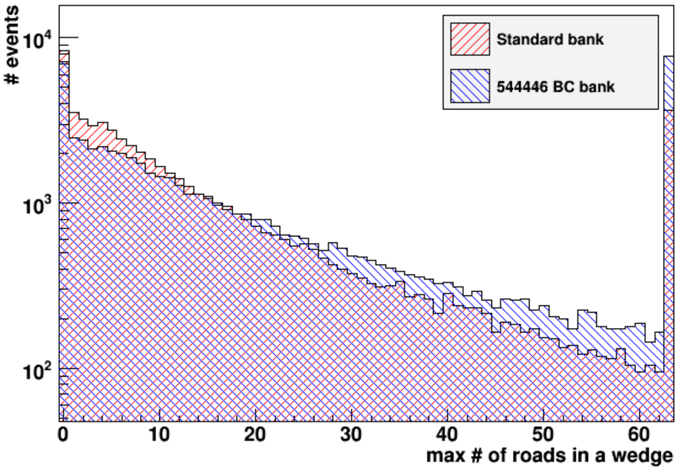


Figure 6.12: Maximum number of roads per wedge with standard banks and 544446 banks - The plot shows the number of found roads in the wedge where it is maximum. In red is the standard banks and in blue the new 544446 banks

6.13 the number of roads in the most crowded wedge is greatly reduced: the mean number of roads is lowered from 21 to 18 and even if while the number of events limited by the HB++ is still large with respect to the standard bank we have to underline that the discarded roads in the case of ordered output will be all 4/5 with smaller coverage.

Even with the current data banks ("433346" without crossing tracks) the AM++ ordered output can be enabled thanks to the GF: in table 6.1 is shown the efficiency and purity for the standard bank and ordered output. The GF have a very little gain in efficiency (+0.1%) due to less useful roads cut and the TF++ drops it's efficiency by 2.3% due to 5/5 bad fits, as already observed in (23).

In conclusion with the GF is possible to use a 544446 road con-

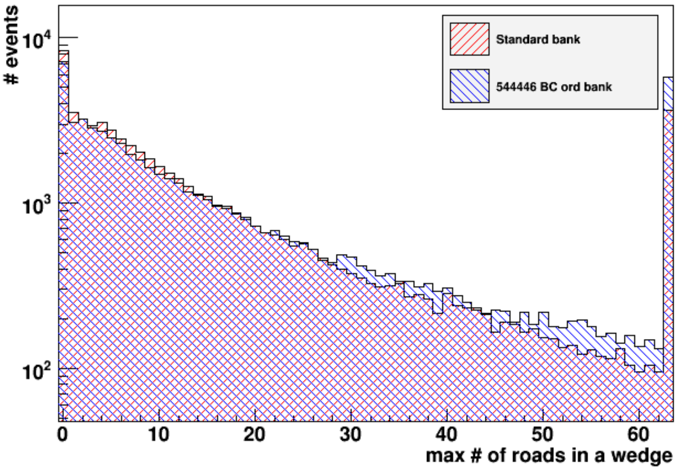


Figure 6.13: Maximum number of roads per wedge with standard banks and 544446 banks and ordered AM++ - The plot shows the number of found roads in the wedge where it is maximum. In red is the standard banks and in blue the new 544446 banks. In the 544446 banks case it was enabled the ordered AM++ output that maximize the duplicate roads suppression by the AMSRW.

figuration banks, with crossing tracks included, leading to a more uniform efficiency distribution in the $(z, \cot\theta)$ plane and a total average efficiency gain of 5.2% with respect to the current bank. Enabling the AM++ ordered output, thanks to the GF 5/5 track fit capabilities, would make the use of this data banks possible with little impact on timing. Also other kind of data banks tuning is possible thanks to the GF: the 5/5 fit mechanism and no limitations in the number of fits to perform allows to explore new configurations of SVT previously forbidden, adapting SVT to the new Tevatron capabilities and physics requests.

The GigaFitter is the first SVT upgrade that allows to improve track efficiency. These improvements will produces physics sample enrichments, as shown in paragraph 4.1.3 and figure 4.6 for the D^0 yield. Even a larger effect will be produced when the Ghost Buster logic will be transferred inside the GigaFitter and if the HB output will be made more powerful, since we will have larger margins to improve the system.

7

Conclusions

It has been described the problem of online selection of events (triggering) in hadron collider physics (2), in particular at the CDF experiment (2.1.2.1) at the Tevatron collider in Fermilab.
It has been shown how the reconstruction of the trajectory of charged particles (tracking) is a critical task for the trigger and various examples of its usage on actual physics problems were provided (2.2).

Tracking is considered one of the hardest task for online selection: the amount of data sampled by the tracking detector is huge, the number of tracks to find big, but hidden over a bigger combinatorial background. It has been described a sophisticated technique (SVT algorithm: 3.1) to perform the track reconstruction task with performances comparable to the best offline algorithms, but executable by a dedicated processor fast enough for usage in the trigger system.

It has been shown in detail the hardware implementation of such algorithm for the CDF experiment: the SVT processor (4). Design, current performance and upgrade history has been described. A particular attention has been put in describing the flexibility of the SVT processor and how it was possible and necessary to upgrade

the hardware in order to adapt to the ever increasing Tevatron luminosity.
The first SVT upgrade was also a pioneer in the field of unplanned trigger hardware upgrades: it has shown how even a complex hardware trigger can be upgraded and commissioned during data taking using a phased plan if it was designed to be flexible enough. Thanks to the SVT upgrade experience it was possible to upgrade other parts of the CDF trigger and fully exploit the increased luminosity for physics measures.

The last SVT upgrade was in 2006 and it was mainly used to reduce the SVT processing time. Without that upgrade the SVT would have been turned off. The GigaFitter upgrade is a second generation upgrade and its main goal is to improve SVT efficiency and acceptance without loosing the SVT timing performance. It has been described the GigaFitter processor: a new generation single board processor for the track fitting stage of the SVT algorithm. It has been designed to replace the current 16 board TF++ processors in SVT and to provide the SVT new and enhanced capabilities. Its architecture has been described in 5.
The GigaFitter board has been fully developed by a small group of physicists and engineers coordinated by me. I have designed most of the details of the architecture and written a very large part of the firmware that implements the actual GigaFitter.
I have actively coordinated all the phases developing of the board and all the steps in the phased upgrade. I have also written the maintenance and configuration code and part of the simulation code and used it to analyze new effects and performance of SVT with the GigaFitter board.

It has been shown how the SVT system can benefit from the new GF board: the timing performances has been studied (6.1) and it has been highlighted how the GF is able to deal with the most

complex events much better than the TF++. The overall timing is not improved much with the current SVT tuning, but is foreseen how the use of the GF opens new possibility of SVT tuning that were forbidden by the lesser TF++ computing power.
In 6.2.2 it has been shown how it's possible to use new banks that will enhance the current SVT efficiency from 75% to 80% by recovering previously unexploited kind of tracks and new, larger data banks. It is also shown how it's finally possible to fully exploit a previous part of the algorithm (the Road Warrior duplicate roads suppression algorithm) that could improve the timing especially with crowded events at high luminosity.
Those results were not possible with the current SVT: the GF board is an upgrade that effectively can enhance the aim of the SVT processor and enable its profitable usage at the new high luminosity of the Tevatron.

The GF board has been commissioned during February 2010 and the TF++ system has been decommissioned in March 2010. It is now an official part of SVT.

The GF board also shows how to design a new generation track fitter for this kind of algorithm, exploiting compact and powerful FPGAs with DSP processor. This experience will be essential with future application of SVT-like processors such as FTK at the Atlas experiment (14).

References

[1] A. Abulencia et al. **Measurement of the $B_s^0 - \bar{B}_s^0$ Oscillation Frequency**. *Phys. Rev. Lett.*, **97**:062003, 2006. 28

[2] A. Abulencia et al. **Observation of B^0 $(s) \to K^+K^-$ and Measurements of Branching Fractions of Charmless Two-body Decays of B^0 and B_s^0 Mesons in $\bar{p}p$ Collisions at $\sqrt{s} = 1.96$-TeV**. *Phys. Rev. Lett.*, **97**:211802, 2006. 28

[3] A. Abulencia et al. **The CDF II extremely fast tracker upgrade**. *Nucl. Instrum. Meth.*, **A572**:358–360, 2007. 11, 20

[4] Darin E. Acosta et al. **Measurement of the J/ψ meson and $b-$hadron production cross sections in $p\bar{p}$ collisions at $\sqrt{s} = 1960$ GeV**. *Phys. Rev.*, **D71**:032001, 2005. 15

[5] J. Adelman et al. **The Silicon Vertex Trigger upgrade at CDF**. *Nucl. Instrum. Meth.*, **A572**:361–364, 2007. 69

[6] Jahred A. Adelman et al. **The 'Road Warrior' for the CDF online silicon vertex tracker**. *IEEE Trans. Nucl. Sci.*, **53**:648–652, 2006. 69

[7] S. Amerio, M. Bettini, P. Catastini, M. A. Ciocci, G. Cortiana, F. Crescioli, M. Dell'Orso, J. Donini, P. Giannetti, V. Greco, D. Lucchesi, M. Nicoletto, S. Pagan Griso, M. Piendibene, L. Sartori, A. Scribano, P. Squillacioti, and G. Volpi. **The GigaFitter for fast track fitting based on FPGA DSP arrays**. In *Nuclear Science Symposium Conference Record, 2007. NSS '07. IEEE*, **3**, pages 2115–2117, October/November 2007. 78, 82

[8] A. Annovi et al. **A VLSI processor for fast track finding based on content addressable memories**. *IEEE Trans. Nucl. Sci.*, **53**:2428–2433, 2006. 41

[9] A. Annovi et al. **The AM++ board for the silicon vertex tracker upgrade at CDF**. *IEEE Trans. Nucl. Sci.*, **53**:1726–1731, 2006. 69

[10] B. Aubert et al. **Improved Measurements of the Branching Fractions for $B^0 \to \pi^+\pi^-$ and $B^0 \to K^+\pi^-$, and a Search for $B^0 \to K^+K^-$**. *Phys. Rev.*, **D75**:012008, 2007. 28

[11] G. Batignani, S. Bettarini, G. Calderini, R. Cenci, A. Cervelli, F. Crescioli, M. Dell'Orso, F. Forti, P. Giannetti, M. A. Giorgi, A. Lusiani, S. Gregucci, G. Marchiori, F. Morsani, N. Neri, E. Paoloni, M. Piendibene, G. Rizzo, L. Sartori, J. Walsh, E. Yurstev, C. Andreoli, L. Gaioni, E. Pozzati, L. Ratti, V. Speziali, M. Manghisoni, V. Re, G. Traversi, M. Bomben, L. Bosisio, G. Giacomini, L. Lanceri, I. Rachevskaia, L. Vitale, D. Gamba, M. Bruschi, R. Di Sipio, B. Giacobbe, A. Gabrielli, F. Giorgi, G. Pellegrini, C. Sbarra, N. Semprini, R. Spighi, S. Valentinetti, M. Villa, and A. Zoccoli. **The associative memory for the self-triggered SLIM5 silicon telescope**. In *Nuclear Science Symposium Conference Record, 2008. NSS '08. IEEE*, pages 2765–2769, October 2008. 42

[12] A. Bhatti, A. Canepa, M. Casarsa, M. Convery, G. Cortiana, M. Dell'Orso, S. Donati, G. Flanagan, H. Frisch, T. Fukun, P. Giannetti, V. Greco, M. Jones, D. Krop, T. Liu, D. Lucchesi, D. Pantano, M. Piendibene, L. Ristori, L. Rogondino, V. Rusu, L. Sartori, V. Veszpremi, M. Vidal, and L. Zhou. **Level-2 Calorimeter Trigger Upgrade at CDF**. In *Real-Time Conference, 2007 15th IEEE-NPSS*, 2007. 69

[13] M. Bogdan, R. Demaat, W. Fedorko, H. Frisch, K. Hahn, M. Hakala, P. Keener, Y. Kim, J. Kroll, S. Kwang, J. Lewis, C. Lin, T. Liu, F. Marjamaa, T. Mansikkala, C. Neu, M. Pitkanen, B. Reisert, V. Rusu, H. Sanders, S.H. Stabenau, R. Van Berg, P. Wilson, D. Whiteson, and P. Wittich. **CDF level 2 trigger upgrade - the Pulsar project**. In *Nuclear Science Symposium Conference Record, 2004 IEEE*, 2004. 9, 69

[14] E. Brubaker, C. Ciobanu, F. Crescioli, M. Dunford, P. Giannetti, Young-Kee Kim, T. Liss, M. Dell'Orso, G. Punzi, M. Shochet, G. Usai, I. Vivarelli, G. Volpi, and K. Yorita. **Performance of the Proposed Fast Track Processor for Rare Decays at the ATLAS Experiment**. *Nuclear Science, IEEE Transactions on*, **55**(1):145–150, Feb. 2008. 113

[15] DZero Collaboration. **Measurement of the Flavor Oscillation Frequency of B_s Mesons**. D0note 5474-conf, 2007. 28

[16] M. Dell'Orso and L. Ristori. **VLSI structures for track finding**. In *Trieste 1988, Proceedings, The impact of digital microelectronics and microprocessors on particle physics* 239-246. 41

[17] S. Donati. *Una strategia per la misura della asimmetria CP nel decadimento $B^0 \to hh$ a CDF*. PhD thesis, University of Pisa, 1997. 27

[18] S. Donati. **CP violation in the B_s^0 system**. 2007. 25

[19] F. Morsani et al. **The AMchip: A VLSI associative memory for track finding**. *Nucl. Instrum. Meth.*, **A315**:446–448, 1992. 41

[20] F. Palla. **Proposal for a first level trigger using pixel detector for CMS at super-LHC**. *JINST*, **2**:P02002, 2007. 42

[21] F. Palla, F. Crescioli, and P. Catastini. **The CDF Associative Memory for a level-1 tracking system at CMS**. Presented at 15th IEEE Real Time Conference 2007 (RT 07), Batavia, Illinois, 29 Apr - 4 May 2007. 42

[22] Raffaello Pegna et al. **A GHz sampling DAQ system for the MAGIC-II telescope**. *Nucl. Instrum. Meth.*, **A572**:382–384, 2007. 69

[23] B. Simoni, A. Annovi, R. Carosi, and S. Torre. **Patterns generation for the SVT upgrade**. Technical report, CDF Note CDF/DOC/TRIGGER/PUBLIC/8256, 2006. 105, 107

[24] The CDF Collaboration. **SVT Techinical Design Report**. Internal note CDF/DOC/TRIGGER/PUBLIC/3108, 22 November 1994. 39

[25] Evelyn J. Thomson et al. **Online track processor for the CDF upgrade**. *IEEE Trans. Nucl. Sci.*, **49**:1063–1070, 2002. 33