

# The GigaFitter Upgrade

S.Amerio <sup>a,b</sup>, A.Annovi <sup>c</sup>, M.Bettini <sup>b</sup>, M.Bucciantonio <sup>d,e</sup>, P.Catastini <sup>f,e</sup>,  
F.Crescioli <sup>d,e</sup>, M.Dell’Orso <sup>d,e</sup>, B.Di Ruzza <sup>g,e</sup>, P.Giannetti <sup>e</sup>, D.Lucchesi <sup>a,b</sup>,  
M.Nicoletto <sup>b</sup>, M.Piendibene <sup>d,e</sup> and G Volpi <sup>d,e</sup>

<sup>a</sup> *University of Padova* <sup>b</sup> *INFN Padova* <sup>c</sup> *INFN Laboratori Nazionali di Frascati* <sup>d</sup>  
*University of Pisa* <sup>e</sup> *INFN Pisa* <sup>f</sup> *University of Siena* <sup>g</sup> *University of Cassino*

## Abstract

The Gigafitter (GF) is a next generation track fitter designed to upgrade the Silicon Vertex Tracker (SVT) during the final period of CDF data taking. With respect to the current system, the GF is much more compact (1 board instead of 15) and will allow to improve SVT track reconstruction efficiency. In this note we will first describe in detail the GF architecture and then illustrate its performance against the current system in terms of track parameter calculation and resolution, track reconstruction efficiency and purity, beam position measurement and timing.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Overview of SVT</b>	<b>3</b>
<b>3</b>	<b>Motivations for the upgrade</b>	<b>3</b>
<b>4</b>	<b>The Gigafitter: Hardware Structure</b>	<b>5</b>
4.1	Input and output . . . . .	7
4.1.1	Input data stream . . . . .	7
4.1.2	Output data stream . . . . .	8
4.2	Internal structure and algorithm . . . . .	8
4.2.1	The track processing module . . . . .	10
4.2.2	Merger module . . . . .	13
4.2.3	Debug features . . . . .	14
<b>5</b>	<b>Configuration for parasitic tests</b>	<b>15</b>

<b>6</b>	<b>Monitoring tools</b>	<b>15</b>
6.1	Online monitoring tools . . . . .	15
6.2	Offline monitoring tools and simulation . . . . .	17
<b>7</b>	<b>The Gigafitter vs TF++</b>	<b>17</b>
7.1	Full precision fits . . . . .	17
7.2	Track parameter study . . . . .	21
7.2.1	Tracks reconstructed by both GF and TF . . . . .	21
7.2.2	GF only tracks . . . . .	22
7.3	Resolution on parameters . . . . .	22
7.4	Efficiency and Purity . . . . .	27
7.5	Beam Fit using GF tracks . . . . .	31
7.6	Timing . . . . .	35
<b>8</b>	<b>Summary</b>	<b>35</b>

## 1 Introduction

Real time event reconstruction plays a fundamental role in High Energy Physics experiments. Reducing the rate of events to be saved on tape from millions to hundreds per second is critical. In order to increase the purity of the collected samples, rate reduction has to be coupled with the capability to simultaneously perform a first selection of the most interesting events.

CDF can perform a high-resolution track reconstruction at the trigger level due to the Silicon Vertex Tracker (SVT). This tracking processor was originally built for B-physics event reconstruction, and had an extremely significant impact on the CDF physics program as a whole. The SVT allows the selection at trigger level of events with displaced tracks; it increased by several orders of magnitude the efficiency for the identification of hadronic decay of B mesons.

The GigaFitter (GF) is a next generation track fitter designed to upgrade SVT track fitting system, in order to enhance SVT performances in a very high luminosity environment. The GF is based on a modern Xilinx Virtex-5 FPGA chip, rich in powerful DSP arrays and features high speed operation, modularity, flexibility and reduced size with respect to the current system.

In this note, after reviewing the architecture of SVT, we will illustrate the reasons for this upgrade. Then the Gigafitter architecture will be described in detail, followed by the description of the configuration used for the parasitic tests and the monitoring tools. Finally the Gigafitter performance in terms of resolution on parameters, track reconstruction efficiency, purity and timing will be shown and discussed compared to the current system.

## 2 Overview of SVT

SVT [1] [2] is the L2 trigger processor dedicated to the reconstruction of charged particle trajectories in the plane transverse to the beam line. Its inputs are the  $p_T$  and the azimuthal angle  $\phi$  of the tracks found by XFT and the data from SVXII. SVT proceeds through three main steps: hit finding, pattern recognition and track fitting and its data flow is schematized in fig 1.

Raw data from SVXII are first sent to Hit Finder boards (HF) boards [3] which find pulse height clusters and compute the coordinate of the centroid of each cluster. A Merger board [4] merges the hits found by the HFs with the  $p_T$  and  $\phi$  of the XFT track and sends them to the AM Sequencer and Road Warrior (AMSRW) board and to the Associative Memory (AM++) board. A copy of the hits and the XFT track is also stored in the Hit Buffer (HB++) board [9]. The AM++ [5] [6] performs pattern recognition: it searches low resolution tracks (*roads*) among the list of SVXII hits and XFT tracks; a road is a coincidence between hits on four of the silicon layers and a XFT track and it is precalculated and stored in large memories. Upon receiving a list of hits and tracks, each AM++ chip checks if all the components of one of its roads are present in the list of hits and XFT tracks. When it has determined that a road might contain a track, the road's hits are retrieved from the HB++ and passed to the Track Fitter (TF++) board. The AMSRW board [7] [8] implements both the sequencer for the AM++ and the road warrior function to eliminate redundant track candidates before track fitting. The TF++ [10] calculates the track parameters  $\vec{p}$  by a simple scalar product, using a linear approximation in each SVXII wedge ( $30^\circ$ ):

$$\vec{p} = \vec{f}_i \cdot \vec{x}_i + q_i \quad (1)$$

where  $\vec{f}_i$  and  $q_i$  are fit constants that are precalculated and stored in TF++ memory and the vector  $\vec{x}$  is formed by the positions of 4 SVXII hits and  $p_T$  and  $\phi$  from the XFT track. The TF++ provides precise measurement of track impact parameter  $d_0$ , curvature and azimuthal angle for all tracks with  $p_T > 2$  GeV, as well as the  $\chi^2$  of the fit. The tracks fitted by the TF++ are then sent to the Ghost Buster (GB) board [11] which is the final SVT board; the GB board deletes duplicate tracks which have the same parent XFT track using the track  $\chi^2$ , corrects the track impact parameter for beam position offsets and the azimuthal angle for bias due to the linearized functions used in the fit. It also provides diagnostic readout of SVT cable data and timing.

## 3 Motivations for the upgrade

The Gigafitter has been designed to upgrade the track fitting system of SVT in order to overcome its limits during the final period of CDF data taking and increase SVT track reconstruction efficiency, which is currently around 80% for tracks having at least 4 hits in the silicon detector. The efficiency depends on the number of roads stored in AM++ memory and their size as well as on the number of constant sets available

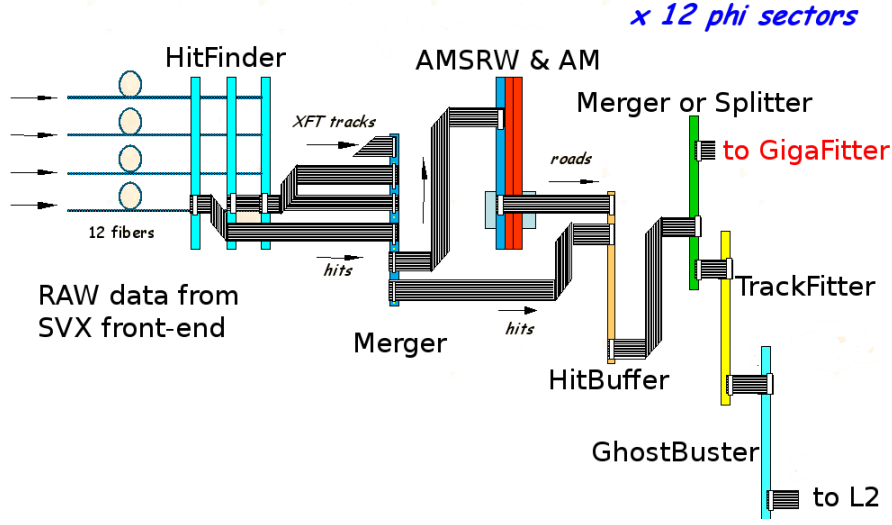


Figure 1: SVT data flux for a single SVXII wedge. Also shown the position of the GigaFitter board during the parasitic tests.

for the fit and stored in the TF++ memory. The current system uses 512k roads per SVX wedge, even if the AM++ memory could store up to 655k, and 30 sets of constants (6 barrels  $\times$  5 sets each, differing for a missing layer): these values are the best compromise between the limits imposed by the TF++ hardware and the track reconstruction efficiency. The TF++ board, infact, is provided with 8x8 bit multipliers while the track coordinates are expressed by 18 bits. Due to these feature, the scalar product performed by the TF++ has to be split into two terms: one is a function of the road boundary and can be precalculated; the other, function of the distance between the road boundary and the hit, is calculated online (more details in Section 7.1). The precalculated term (one for each AM++ pattern) is 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 feature is the actual limit to the bank size and to the number of constant sets we can use inside SVT. As an example, the current bank doesn't account for all the possible tracks which can be left by charged particles traversing the detector: tracks with  $p_T < 2$  GeV are not considered, as well as track crossing the mechanical barrels. In fig. 2 the track reconstruction efficiency as a function of  $\cot\theta$  and  $z$  is shown: the efficiency loss in correspondance of mechanical barrels crossing is clearly visible.

The track reconstruction efficiency also depends on the road size. The current value is  $200 \mu m$ , the best compromise achievable with the current hardware between track reconstruction efficiency and processing time. A larger value would provide a greater efficiency but also higher processing time, because many tracks could be associated to the same road. On the other hand, a smaller size would reduce the processing time, at the price of a lower efficiency, because the number of possible roads is limited. The



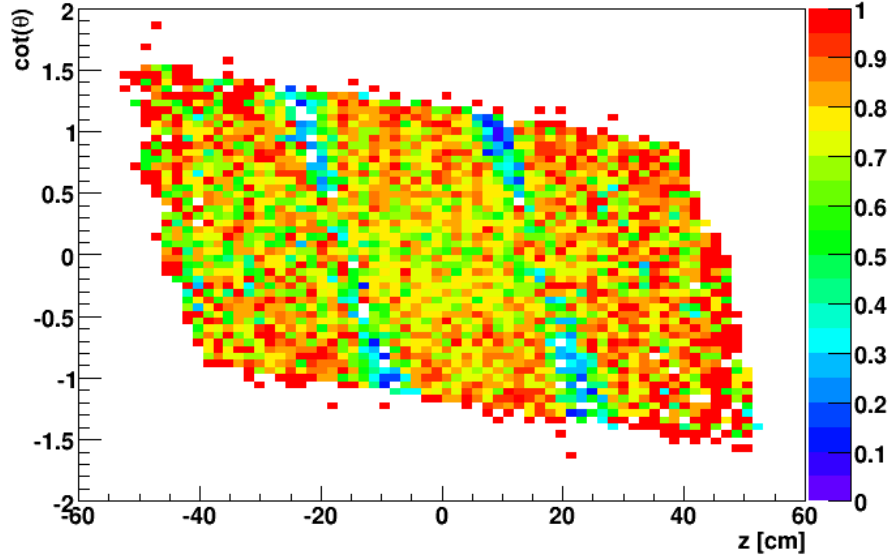


Figure 2: Track reconstruction efficiency as a function of  $\cot\theta$  and  $z$ :

maximum number of fits that can be performed by the TF++ board is limited, so increasing the road width is not feasible; as it is not possible to increase the number of the roads for a smaller road size. The Gigafitter is provided with 25x18 bit multipliers (see Section 4 for more details): the scalar product can be calculated with full hit resolution, without the need for precalculated terms and the memory on the GF board can be used to increase the number of constant sets. Moreover, without precalculated terms, the AM++ memory can be fully exploited adding more patterns.

Another limit of the current TF++ arises in the case of a track with hits in all 5 SVXII layers (*full of hits track*): the TF++ uses 4 out of 5 hits, discarding one hit on the basis of the layers used and the quality of the hits. In a high luminosity environment, with increasing probability of fake hits due to noise, this choice can reduce track reconstruction efficiency: if a real hit of low quality is discarded in favour of a fake hit, the corresponding track does not pass the  $\chi^2$  cut and is rejected. The Gigafitter has enough computing power to fit all possible combinations of 4 hits out of 5 and select only the best.

## 4 The Gigafitter: Hardware Structure

The GF has been designed to upgrade the current track fitting system: it is made by a single board to replace the 12 TF++ boards and the 3 final mergers. The GF is based on a motherboard called Pulsar [12] and three GigaFitter mezzanines.

The Pulsar board, shown in figure 3, is a 9U VME board based on three intercon-

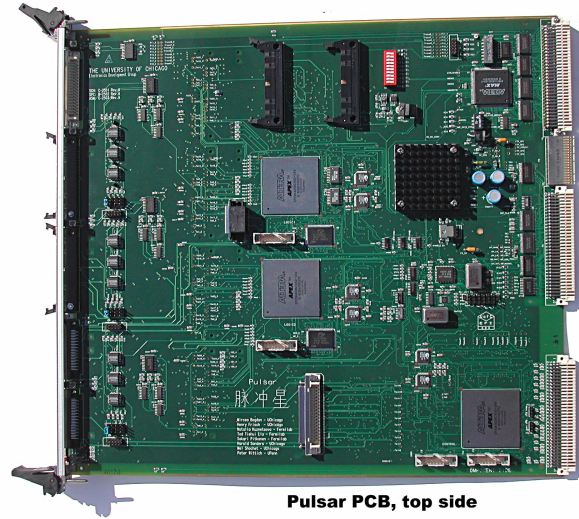


Figure 3: Front side of the Pulsar board. Mezzanine connectors used by the GigaFitter system are on the back side.

nected 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 [12], L2CAL upgrade [13], and SVT upgrade [14, 15]. It's also used for trigger and data acquisition systems of other experiments, as Magic [16] for example.

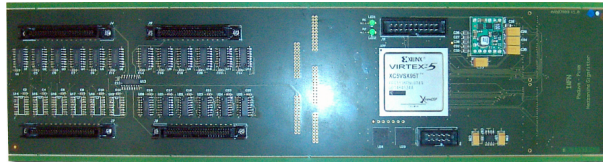


Figure 4: The GigaFitter mezzanine. All components on the front side.

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 4, 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 generates 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. The full GigaFitter system with all 12 inputs connected is shown in figure 5.



Figure 5: The GigaFitter system in the test crate of SVT. All 12 input are connected in parasitic mode to splitted HB++ outputs.

## 4.1 Input and output

The GigaFitter board receives hits and roads from the 12 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.

### 4.1.1 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 1. The 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 size. The road size commonly used in the past years gives a maximum of 25 words.

	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 1: HitBuffer++ to GigaFitter packet format

#### 4.1.2 Output data stream

Output data is composed by 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 ( $\chi^2$  and GF fit status) as described in table 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_	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_			$\chi^2$	x4	
1	HOLD_	DS_	EE_	EP_			GF status	track num	

Table 2: GigaFitter to GhostBuster packet format

## 4.2 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 6). 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.

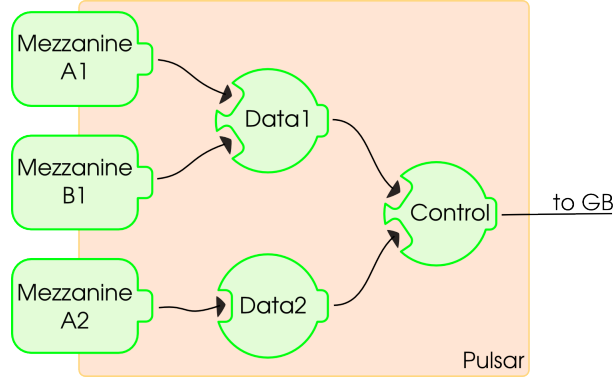


Figure 6: GF Pulsar Scheme: the tracks found in each mezzanine are merged inside the three Pulsar FPGA: Data1, Data2 and Control. The final stream is sent on one SVT cable downstream to GhostBuster board.

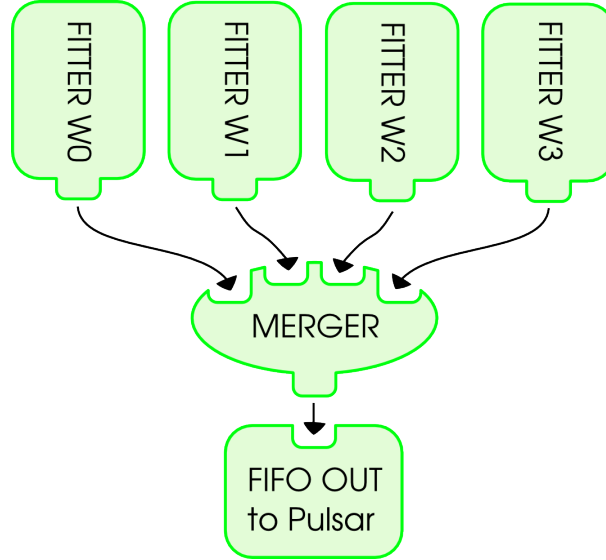


Figure 7: The internal structure of the GF mezzanine: four parallel fitter engines compute tracks from one wedge each, a final unit logic (Merger) merges the four data streams in a single output FIFO that communicates with the Pulsar motherboard.

Inside each mezzanine FPGA there are four track processing modules and a merger module (figure 7). 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 board and for debugging purposes.

### 4.2.1 The track processing module

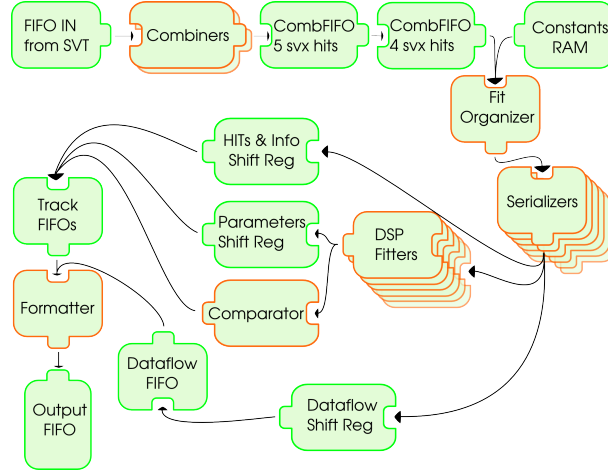


Figure 8: Schematics of the GigaFitter fitter module.

The structure of 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 shift registers have also been added to shift data downstream the pipeline that is fully synchronized. Figure 8 shows all these parts and their interconnections. It is more complete and has improved performances with respect to the first ideas presented in [17].

The Combiner provides combinations of input hits to be tested by the fit. The Fit Organizer coordinates the fetching of hit combinations 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.

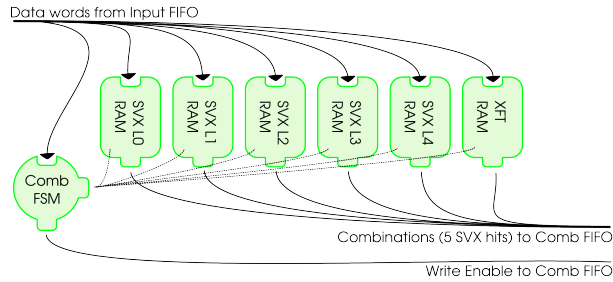


Figure 9: GF Combiner module: the Combiner is made by five RAMs and a finite state machine that controls both writing and combination pop.

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 9) 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 works generating combinations of hits with always 5 SVX layers: when an SVX layer is missing in the road hits (4/5 road) it has zeros in place and the missing information is stored in an hitmap field of the combination word. All the other stages in the GF Track Processing module works with 4 SVX layers and the hitmap information, like the TF++. For this reason a simple finite state machine and a multiplexer connect the two Combination FIFOs and converts all 4/5 combinations in a 4 SVX layer + hitmap format removing the missing layer from the combination word and all 5/5 combinations in five 4 SVX layer combinations computing the appropriate combination word.

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, input and output barrel  $z$ , missing layers and hitmap) and quality of the hits (long clusters are flagged by the Hit Finders as low precision points) require in principle different sets of constants. The right constants are fetched taking into account 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 bits for  $z$ , 3 bits for the hitmap and 4 bit for the long cluster map), but the physically relevant configurations are only 240 thus a two-RAM system is used: from the combination a first 8x8k bit RAM is accessed which is used to access the 756x256 constant RAM.

The whole set of hits and the associated constants are extracted in parallel in a single clock and the Fitter Organizer sends a start signal to a Serializer. The Serializer can accept one combination every 6 clock cycles so there are 6 parallel Serializers and the Fitter Organizer keeps track of which one has to handle the fetched hit combination.

Each Serializer register the hits and constants, 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.

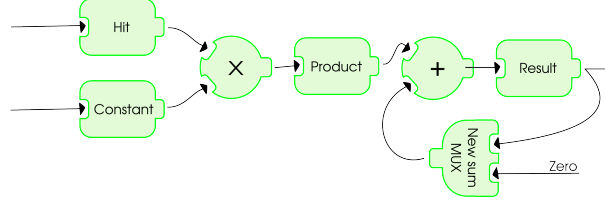


Figure 10: GF DSP Fitter unit: the scalar product unit is made inside the specialized DSP48 unit configured in MACC (multiply-and-accumulate) mode. Each unit can compute a scalar product in 6 clock cycles. There are 2 additional clock cycles of latency before the result appear in output, but the unit 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 ( $\chi^2$  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 10). In a DSP Fitter there are six DSPs, each one able to compute a fit parameter ( $c$ ,  $d$ ,  $\phi$ ) or one of the three  $\chi^2$  components. Thus with the six DSP Fitters, for a total of 36 DSPs, and the associated Serializer the GF is able to process one combination every clock cycle.

Once the results are ready, the  $\chi^2$  components are sent to the Comparator (figure 11) 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 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 chooses the best. The Comparator has to fine-tune the final decision using not only the  $\chi^2$ , but also the hit combination layout (used layers and quality of the hits).

The Comparator has the ability to choose the best track of an arbitrary sequence of tracks and the control bits going to the Comparator are thus set to consider the five 5/5 tracks as a 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 at road level. This feature may be implemented in a future revision of the firmware. The Comparator calculates the final  $\chi^2$  using a DSP in MACC configuration like the one used in DSP



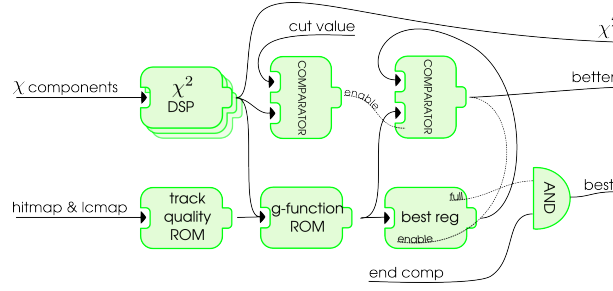


Figure 11: GF Comparator unit: the Comparator computes the  $\chi^2$  of the tracks from  $\chi$  components, applies the  $\chi^2$ -cut selection and compares the track with the previous one finding the best.

Fitter units (figure 10). 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. The Comparator compares the result with the threshold configurable via VME. If the track passes the  $\chi^2$ -cut its  $\chi^2$  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,  $\chi^2$  and additional informations). Once the sequence is finished if there was at least one track passing the  $\chi^2$  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  $\chi^2$  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.

#### 4.2.2 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.

#### 4.2.3 Debug features

Diagnostic and debug is a very important aspect for developing, commissioning and monitoring the status of the board during the normal operation. This aspect has been a key factor of the success of SVT and also in the GF board we implemented the standard debug feature: spy buffers. The GF board is unique in SVT: it has 12 inputs, one output and performs the task that was previously of 15 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 at each end of each “internal SVT cable” (figure 12): at the end of each track processing module inside a mezzanine, at the end of each mezzanine and at the end of each merge unit in the Pulsar board. This resulted in 30 spy buffers (12 input and 12 output, 3 mezzanines, 3 Pulsar FPGA), an unprecedented record for an SVT board, but the monitoring software was flexible enough to add all this spy buffers to the code without much effort.

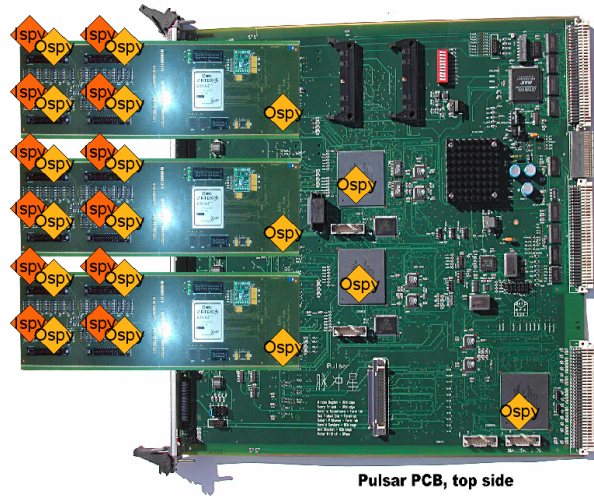


Figure 12: GF Spy Buffers: the GF board is the most complex board of SVT in terms of diagnostic features: it has 30 spy buffers, one for each input and output of track processing module and one for each output of merger units.

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 the status of every components the GF board online. There are several registers to configure the severity of each error and select if rise error bits on the EE word or rise 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 is it possible to insert a custom logic analyzer and pattern generator in-chip and 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), the capability of external logic analyzer and all the problems of routing signal copies to the debug pins. The ChipScope features are disabled in the stable version of the GF Mezzanine firmware.

## 5 Configuration for parasitic tests

The GF test phase has been carried out during the 2009 shutdown period (15 June - 15 September) and the first months of data taking with beam (September 2009 - January 2010). The GF has been installed in `tstsvt2` crate in trigger room. The outputs of the 12 HB++ have been splitted using three Splitter boards and 6 Merger boards, in order to provide to the GF an exact copy of the data in input to the TF++ boards. Three Splitters and three Merger boards have been installed in crate `b0svt09` and three additional Merger boards in crate `b0svt07`. The latency added to the SVT timing by the splitting boards has been measured to be  $\sim 100$  ns, negligible with respect to the overall SVT timing ( $\sim 25 \mu\text{s}$ ). An additional Merger board has been added in crate `b0svt09` to split the Bypass signal. This signal is necessary at high luminosity to reduce the overall SVT timing: SVT track reconstruction is bypassed for events not required to pass a SVT based selection. The configuration used for the test is shown in fig. 13. The GF output is connected to the Bypass board [18] through a Merger. The Bypass output goes to the GB board where GB functions are applied to the tracks reconstructed by the GF.

## 6 Monitoring tools

### 6.1 Online monitoring tools

To online monitor the performance of the GF we have modified the main SVT online monitoring tools, `TRIGMON` and `SPYMON`, to include the GF data.

- `SPYMON` [19] is a monitoring program running on the CPUs of the SVT crates. It collects spy buffer data, reads board error registers and publishes the following three kinds of SmartSocket messages:
  - SVT histograms filled with Spy buffer data quantities;

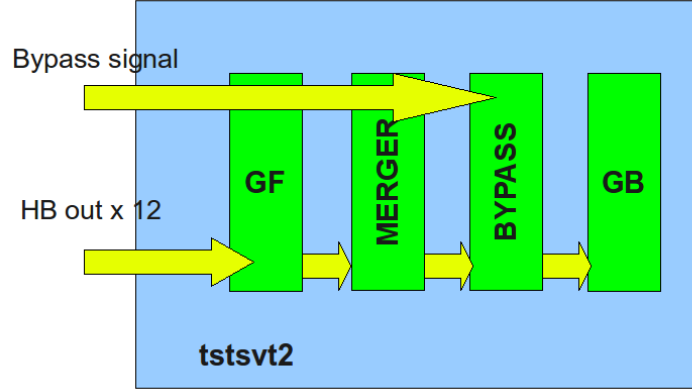


Figure 13: Scheme of GF configuration in crate `tstsvt2`.

- SVT status message containing error registers and status words for each of the boards in the crate and statistics about individual error register flags;
- Spy buffer dumps, i.e. a lengthy string containing the formatted dump of (part) of the Spy buffer data.

A custom version of `SPYMON` has been developed to run on `tstsvt2` crate to monitor errors, track multiplicity, SVT occupancy and track parameters in output from the GF and the GB board.

- `TRIGMON` [20] is a low level online trigger diagnostic and monitor. It is comprised of many modules, each monitoring a specific trigger bank. Official SVT data are monitored by `SVTDMonitor` module, which exploits the information contained in the `SVTD` bank written by the GB board. If the `tstsvt2` crate is included in the data taking, GF data can be saved in the `SVDD` bank, a SVT internal diagnostic bank which can be written by the GB board. This bank is organized in three cards, containing 1) the beam fit result for each SVXII barrel, 2) the result of the 3D beam fit and 3) a copy of the tracks stored in the `SVTD` bank. A custom version of `SVTDMonitor` has been developed to read the information stored in the `tstsvt2` `SVDD` bank and fill histograms with track parameters and beam fit results. `SVTDMonitor` output is a root file containing all the relevant SVT plots. A directory called *VSLICE*<sup>1</sup> has been added and is filled with GF data if `_useVSLICE` parameter is set true in the `TRIGMON` tcl file.

<sup>1</sup>In the past `tstsvt2` crate was used to reproduce a single wedge SVT sector (a so-called Vertical SLICE): we have maintained the same name.

## 6.2 Offline monitoring tools and simulation

If the `tstsvt2` crate is included in the data taking, the SVDD bank is saved on disk and it can be analyzed offline using the SVT offline monitoring tool, `SVTMon`, enabled to read the GF data<sup>2</sup>.

`SVTMon` can also run the SVT simulation `svtsim`, to be compared with the hardware response. The GF simulation has been developed following `svtsim` structure: GF or TF++ can be enabled by the correspondent flag to be set in the file `svtsim_lib.h`.

## 7 The Gigafitter vs TF++

At first the GigaFitter will be installed to exactly replace the TF++ system: the same constant and pattern sets will be used. The minimum requirement for this first step of GF installation is a performance equal or better than the current system. To check that this requirement is met we monitor

- the impact of GF new features (full precision fits, treatment of 5/5 tracks) on parameter calculations and track multiplicity;
- the resolution on parameters;
- the track reconstruction efficiency and purity;
- the impact on the beam position,
- the impact on SVT timing.

### 7.1 Full precision fits

The computation of track parameters and  $\chi^2$  components ( $p_n$ ) is done with a scalar product plus a constant term:

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

where  $c0_n$  and  $c_{ni}$  are known constants and  $x_i$  are the hit positions. The terms  $c_{ni}$  and  $x_i$  are 18 bit and 15 bit wide but in the TF++ board only 8x8 bits multipliers are implemented: as a consequence it is not possible to use exactly this 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

---

<sup>2</sup>`_useVSLICE` parameter set true in the `SVTMon` tcl file.

$$x_i = x_i^{ssborder} + x^{low8bit}$$

where the subscripts  $^{high8bit}$  and  $^{low8bit}$  indicate the 8 most significant or less significant bits respectively and  $x^{ssborder}$  is the position of the road border on each layer (usually called superstrip).

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++. 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 accounts 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  $\chi^2$  is shown in figures 14, 15, 16 and 17, obtained running the simulation of the TF++ and the GF on real data. No  $\chi^2$  cut is applied. The  $\chi^2$  difference shown in 14 is proportional to the  $\chi^2$  itself because the  $c_{ni}^{low} * x^{low8bit}$  term for each component is squared and summed. A small amount of tracks 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 7.4 that the GF is more efficient of about the same percentage without reducing the purity of the sample, so overall the  $\chi^2$  computed by the GF is a more accurate quality parameter.

The differences on the other parameters are within the resolution (see Sec. 7.3).

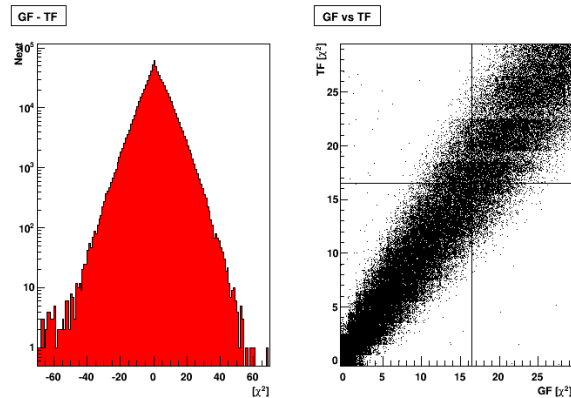


Figure 14: Differences in  $\chi^2$  computation between GF and TF++ due to  $c_{ni}^{low} * x^{low8bit}$  term not computed by TF++. Current cut values are shown with the solid lines.

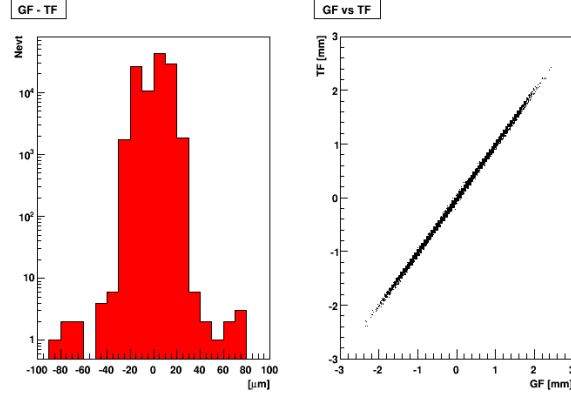


Figure 15: Differences in impact parameter ( $d0$ ) computation between GF and TF++ due to  $c_{ni}^{low} * x^{low8bit}$  term not computed by TF++.

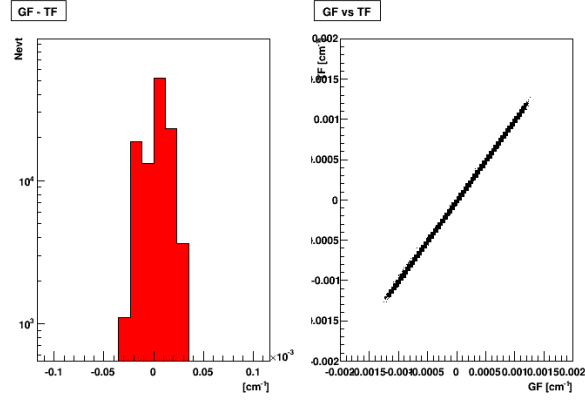


Figure 16: Differences in curvature ( $c$ ) computation between GF and TF++ due to  $c_{ni}^{low} * x^{low8bit}$  term not computed by TF++.

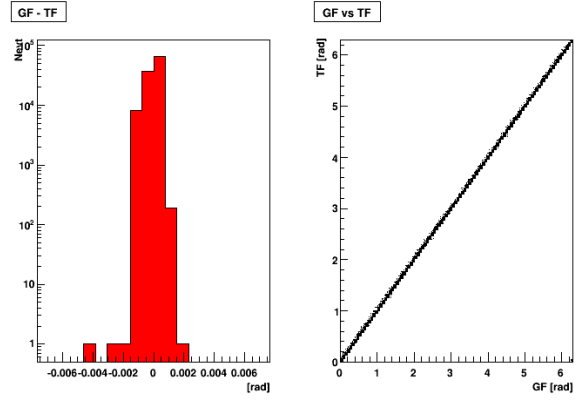


Figure 17: GF vs TF++ differences:  $\phi$   
Differences in  $\phi$  computation between GF and TF++ due to  $c_{ni}^{low} * x^{low8bit}$  term not computed by TF++.



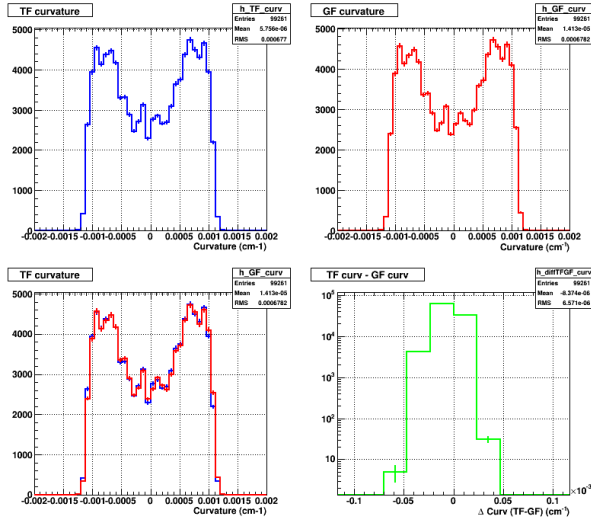


Figure 18: Curvature distribution for tracks reconstructed by both GF and TF++: the distribution of the track by track differences is shown in the lower right plot.

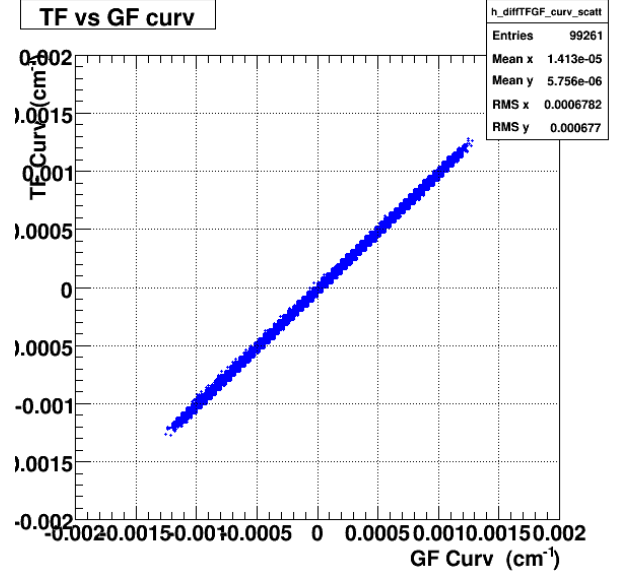


Figure 19: Track curvature calculated by TF++ as a function of the value obtained by GF.

## 7.2 Track parameter study

The GF performs a different treatment of 5/5 tracks and a full resolution calculation with respect to the current system. For these reasons the GF overall reconstructs more tracks than TF++ ( $\sim 3\%$ ). For tracks reconstructed by both systems, we can make a detailed comparison of the track parameters event by event, while for those tracks selected by GF only we can check that the parameter distributions are consistent with the correspondent L3 track distributions.

### 7.2.1 Tracks reconstructed by both GF and TF

We consider 4/5 tracks reconstructed and selected by both GF and TF++. The tracks are matched if they have the same XFT parent and belong to the same road. In fig. 18 we show the curvature distribution for TF++ (upper left), GF (upper right) and both superimposed (lower left). The  $RMS$  of the difference distribution (lower right) is  $RMS_{\Delta c} = 7 \cdot 10^{-6}$  rad, well below the resolution on this parameter (see Sec. 7.3). As a further check, fig. 19 shows for each track the curvature calculated by the TF++ as a function of the correspondent value obtained by the GF.

The same plots are shown in figg. 20-23 for the azimuthal angle and the impact parameter. In these cases  $RMS_{\Delta \phi} = 4 \cdot 10^{-4}$  rad and  $RMS_{\Delta d_0} = 7 \mu\text{m}$ , well below the resolution.

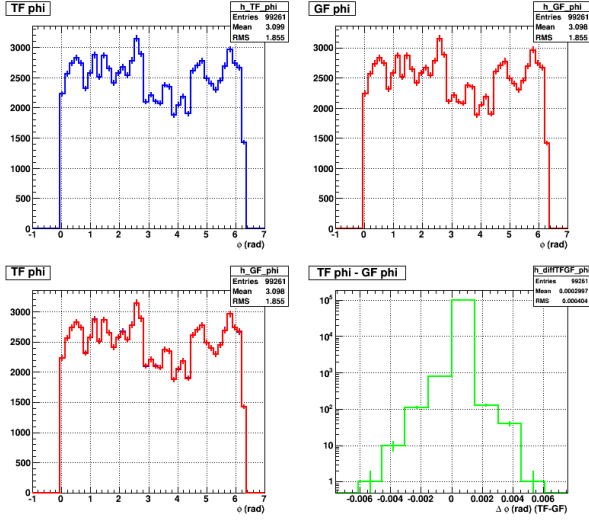


Figure 20: Azimuthal angle distribution for tracks reconstructed by both GF and TF++: the distribution of the track by track differences is shown in the lower right plot.

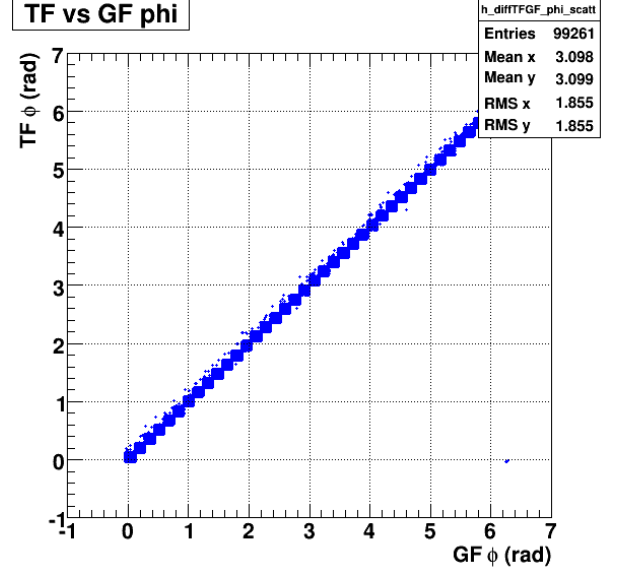


Figure 21: Track azimuthal angle calculated by TF++ as a function of the value obtained by GF.

### 7.2.2 GF only tracks

For tracks reconstructed by the GF only, we cannot make a direct comparison with TF++ tracks. Anyway, to check that the GF tracks not matched to TF++ ones are well reconstructed, we match them to L3 tracks ( $|\Delta\phi| < 0.02\text{rad}$  and  $|\Delta c| < 0.0002\text{ cm}^{-1}$ ) and compare the parameter distributions. The results are shown in figg. 24- 26.

## 7.3 Resolution on parameters

The resolution on fitted parameters is measured as the *RMS* of the distributions of the parameter differences between SVT and L3 reconstructed tracks. SVT tracks are matched to L3 ones in azimuthal angle and curvature ( $|\Delta\phi| < 0.02\text{rad}$  and  $|\Delta c| < 0.0002\text{ cm}^{-1}$ ).

The distributions are shown in figg. 27, 28 and 29.

The GF and TF++ have the same resolution on fitted parameters:  $\sim 30\text{ }\mu\text{s}$  on  $d_0$ ,  $\sim 1\text{ mrad}$  on  $\phi$  and  $\sim 3\cdot 10^{-4}\text{ cm}^{-1}$  on  $c$ .

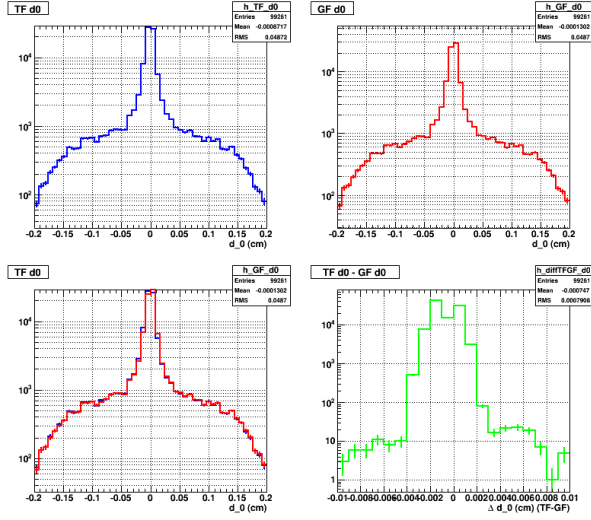


Figure 22: Impact parameter distribution for tracks reconstructed by both GF and TF++: the distribution of the track by track differences is shown in the lower right plot.

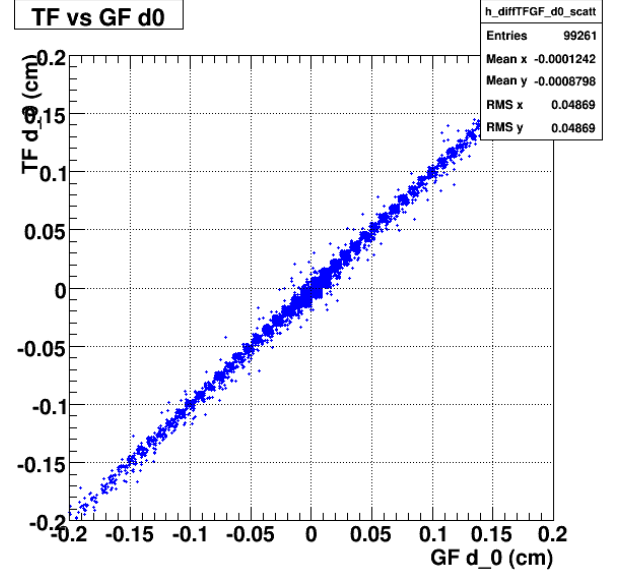


Figure 23: Track impact parameter calculated by TF++ as a function of the value obtained by GF.

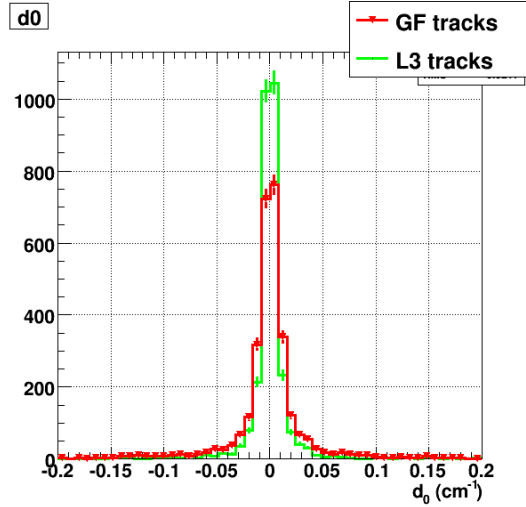


Figure 24:  $d_0$  distribution for tracks reconstructed by GF only and for matched L3 tracks.

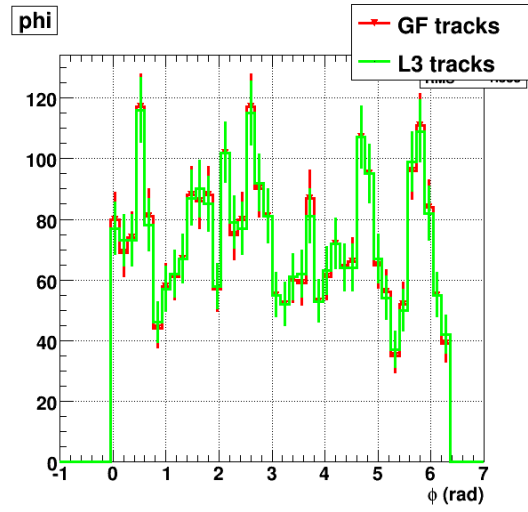


Figure 25:  $\phi$  distribution for tracks reconstructed by GF only and for matched L3 tracks.

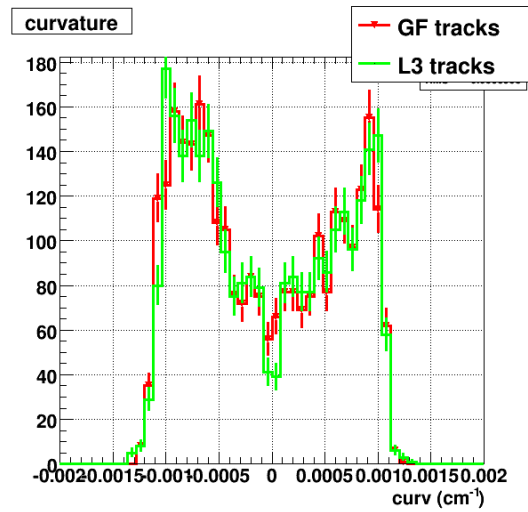


Figure 26: Curvature distribution for tracks reconstructed by GF only and for matched L3 tracks.

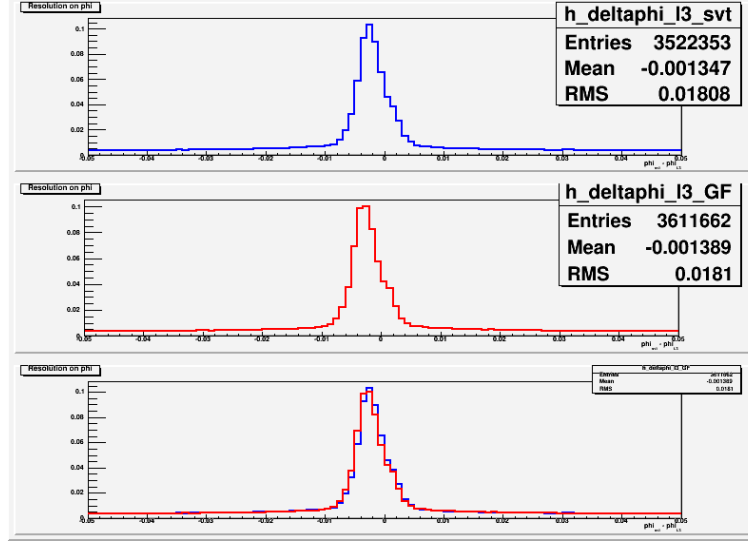


Figure 27: Distribution of the  $\phi$  differences between SVT and L3 tracks for TF++ (upper plot), GF (middle plot) and both (lower plot).

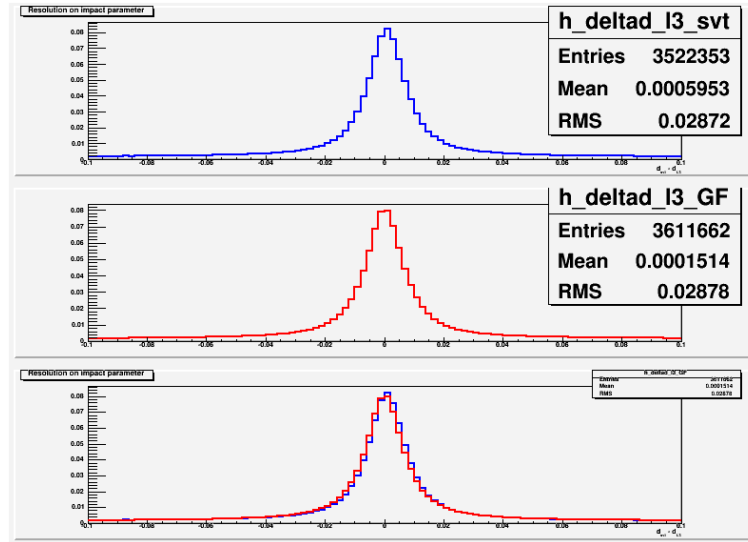


Figure 28: Distribution of the  $d_0$  difference between SVT and L3 tracks for TF++ (upper plot), GF (middle plot) and both (lower plot).

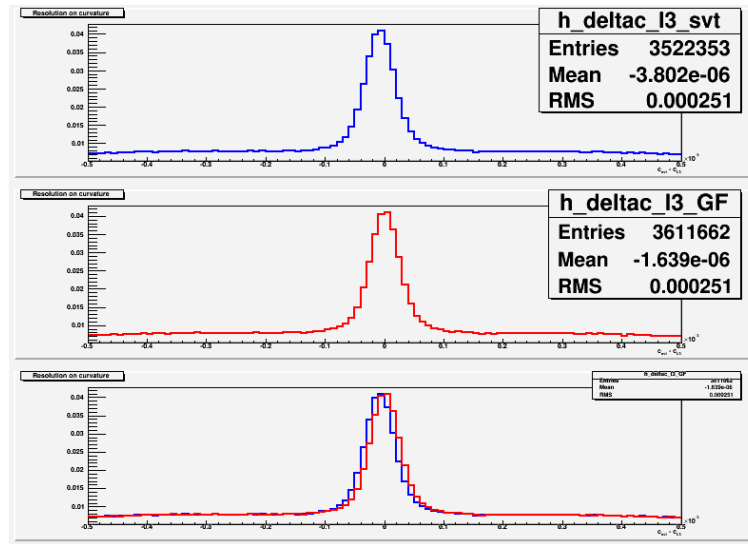


Figure 29: Distribution of the  $c$  differences between SVT and L3 tracks for TF++ (upper plot), GF (middle plot) and both (lower plot).

## 7.4 Efficiency and Purity

Track reconstruction efficiency is measured with respect to

- L3 tracks fiducial to SVXII wedges;
- L3 tracks fiducial to SVXII wedges and with at least 4 hits in the silicon detector.

The efficiency is calculated as

$$\frac{N_{SVT}^{matched}}{N_{L3}}$$

where  $N_{SVT}^{matched}$  are all the tracks reconstructed by SVT *and* matched to a L3 track, while  $N_{L3}$  are all L3 tracks having the input and output  $Z$  position within 50 cm from the center of the detector and fiducial to SVXII wedges.

In fig. 30 the efficiency of GF and TF++ is shown as a function of various track parameters ( $\cot(\theta)$ , curvature, impact parameter  $d_0$ , azimuthal angle  $\phi_0$ , transverse momentum  $p_T$  and input  $z$  position  $z_0$ ). GigaFitter efficiency is slightly higher than TF++ one, even if we are using exactly the same constant and pattern sets. This effect is also manifest in figg. 31 and 32, where the efficiency as a function of instantaneous luminosity is shown for all L3 tracks and for those having at least 4 SVXII hits.

The purity is calculated as

$$\frac{N_{SVT}^{matched}}{N_{SVT}}$$

where  $N_{SVT}$  are *all* tracks reconstructed by SVT. A low purity means that many tracks are incorrectly reconstructed, i.e. the *fake rate* is high.

In fig. 33 purity is shown as a function of track parameters while in fig. 34 we monitor it *vs* the barrel number, the wedge, the difference between the input and output barrels of the tracks and finally the instantaneous luminosity. The GigaFitter has the same purity levels of TF++, with a  $\sim 1.5\%$  gain in efficiency.

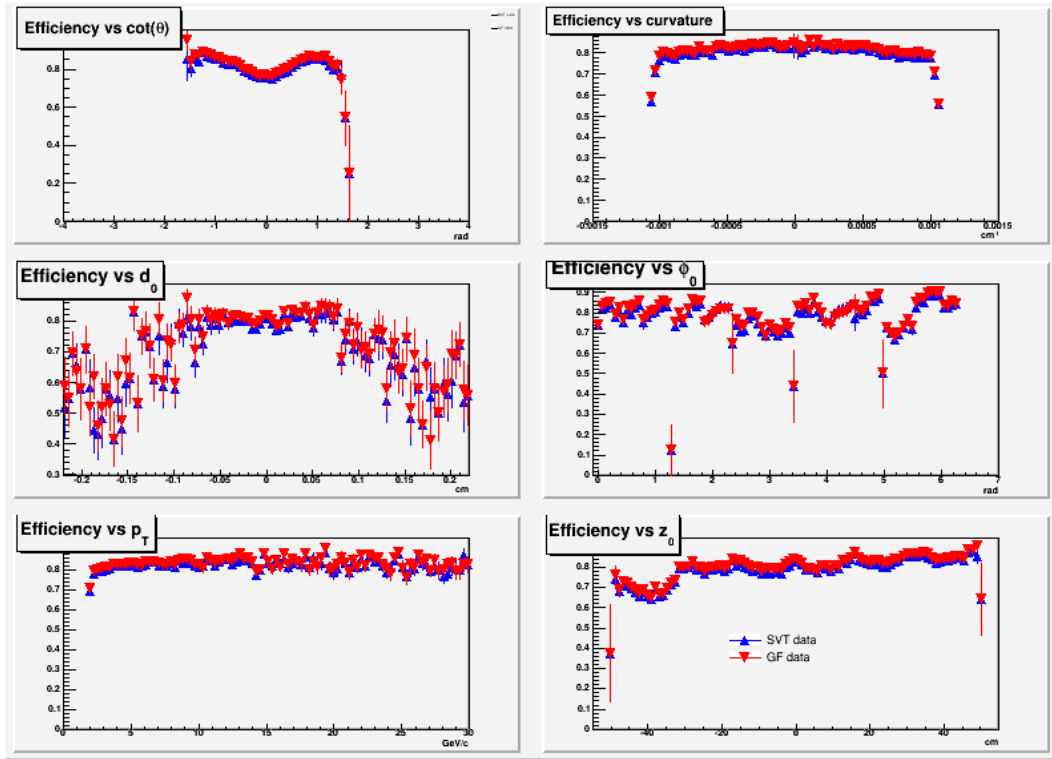


Figure 30: GF vs TF++ efficiency vs track parameters with respect to tracks having at least 4 hits in the silicon detector.

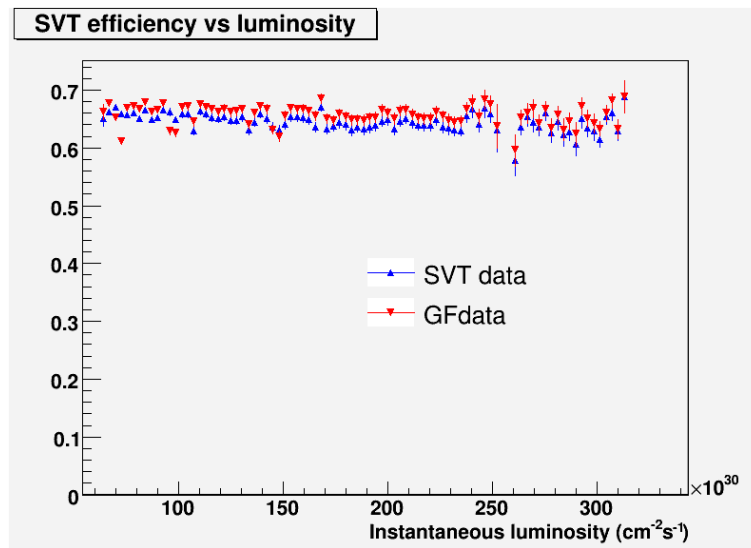


Figure 31: GF vs TF++ efficiency vs instantaneous luminosity



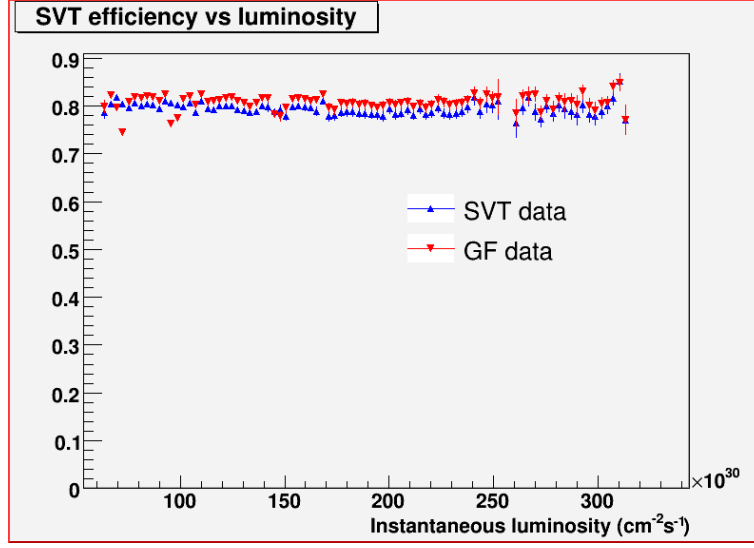


Figure 32: GF vs TF++ efficiency vs instantaneous luminosity for tracks having at least 4 hits in the silicon detector.

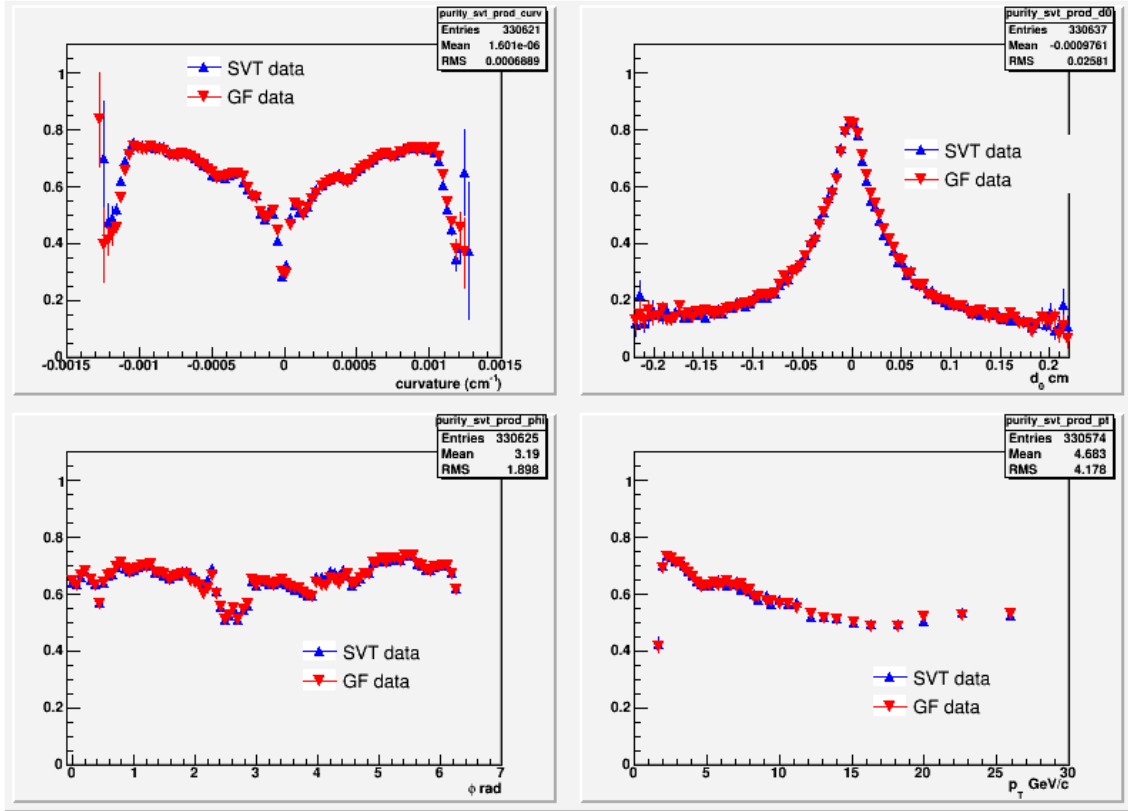


Figure 33: GF and TF++ purity as a function of curvature,  $d_0$ ,  $\phi$  and  $p_T$

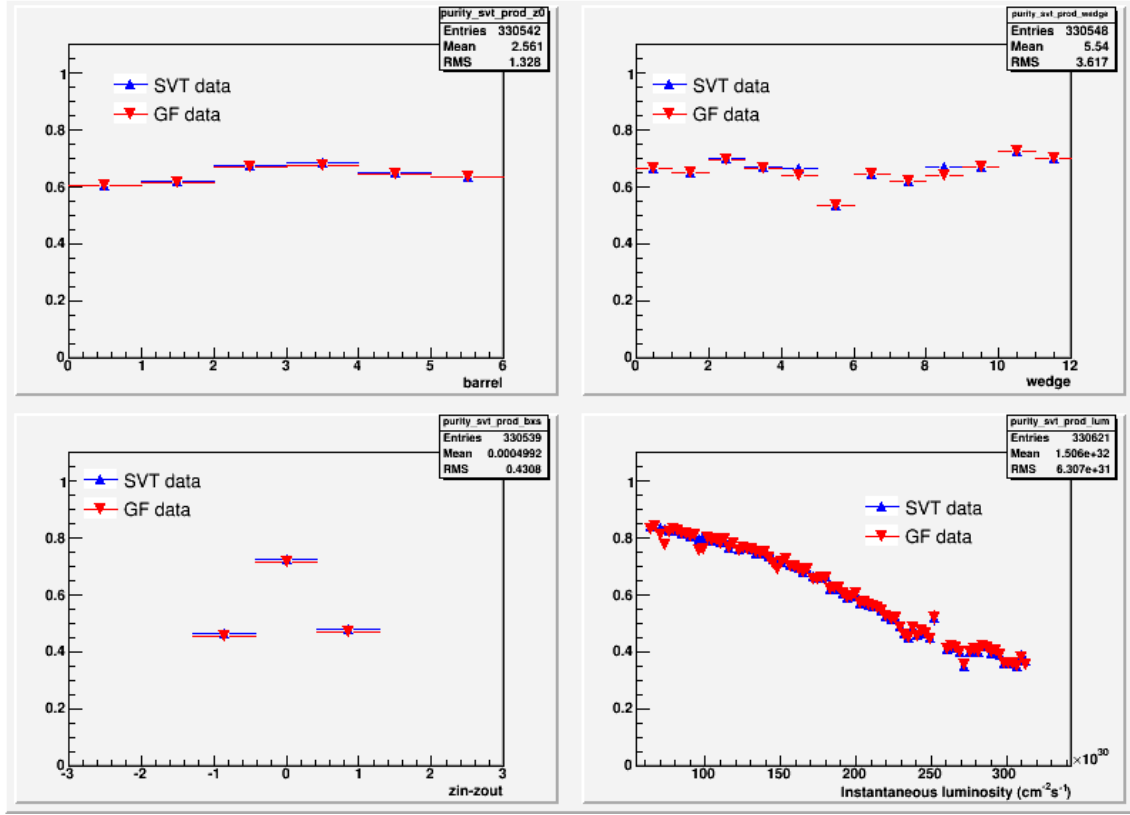


Figure 34: GF and TF++ purity as a function of barrel, wedge, difference between the input and output barrels of the tracks ( $Z_{IN} - Z_{OUT}$ ) and instantaneous luminosity.

## 7.5 Beam Fit using GF tracks

In SVT the beam position is measured in real time by a task running on the crate hosting the final Merger, the board merging all the data streams of the 12 TF++. Impact parameter  $d_0$  and azimuthal angle  $\phi$  of each track are measured with respect to the origin (0,0). If the beam spot is displaced with respect to the origin and has coordinates  $(x_0, y_0)$  in the transverse plane, there is the following relationship between  $d_0$  and  $\phi$ :

$$d_0 = x_0 * \sin(\phi) - y_0 * \cos(\phi).$$

and the  $d_0$  vs  $\phi$  distribution has a sinusoidal shape. The beam fit algorithm reads the track list from the spy buffer of the final Merger and fits the  $d_0$  vs  $\phi$  distribution to obtain the beam position in each of the six barrels. A linear fit to the six values obtained in the different barrels returns the beam line direction, expressed as  $\frac{dx}{dz}$  and  $\frac{dy}{dz}$ , and the mean beam position  $(x_0, y_0)$ .

The GF calculates track parameters with a different resolution with respect to the TF++, as seen in Sec. 7.3. Moreover, the GF reconstructs more tracks than the current system, due to the different treatment of the 5/5 tracks. These differences can have an impact on the beam fit calculation.

In fig. 35-40 the distributions of beam X and Y coordinates are shown.

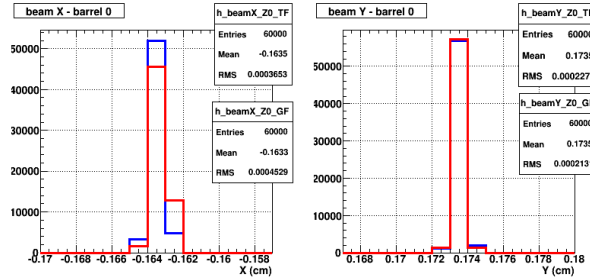


Figure 35: GF vs TF++: X and Y coordinates of beam position for barrel 0 (Blue TF++, red GF).

A summary of the beam spot positions in  $x$  and  $y$  coordinates for all barrels is shown for the GF (red crosses) and the TF++ (blue circles) in fig. 41. Barrel 1 shows a small displacement of the  $x$  coordinate with respect to the TF, but within resolution.

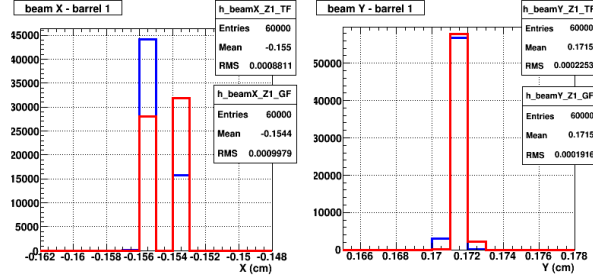


Figure 36: GF vs TF++: X and Y coordinates of beam position for barrel 1 (Blue TF++, red GF).

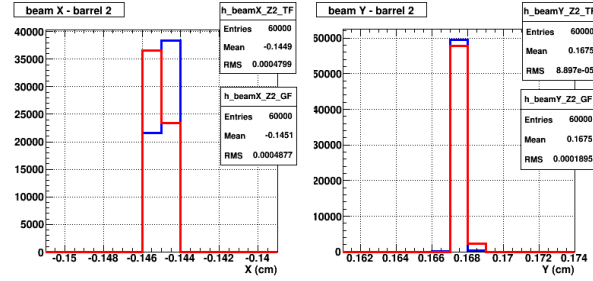


Figure 37: GF vs TF++: X and Y coordinates of beam position for barrel 2 (Blue TF++, red GF).

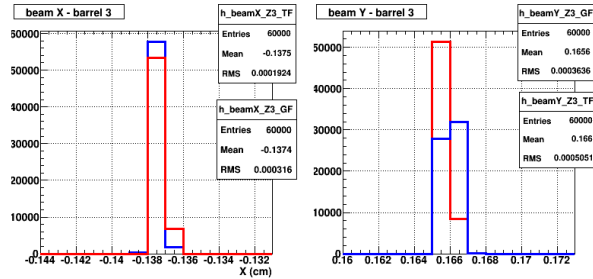


Figure 38: GF vs TF++: X and Y coordinates of beam position for barrel 3 (Blue TF++, red GF).

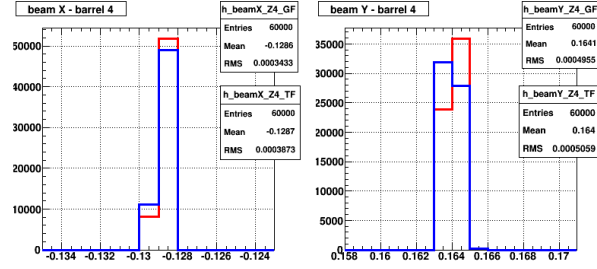


Figure 39: GF vs TF++: X and Y coordinates of beam position for barrel 4 (Blue TF++, red GF).

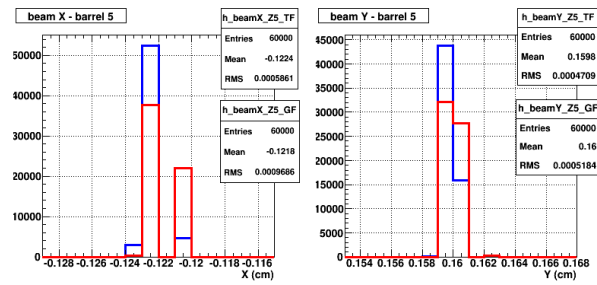


Figure 40: GF vs TF++: X and Y coordinates of beam position for barrel 5 (Blue TF++, red GF).

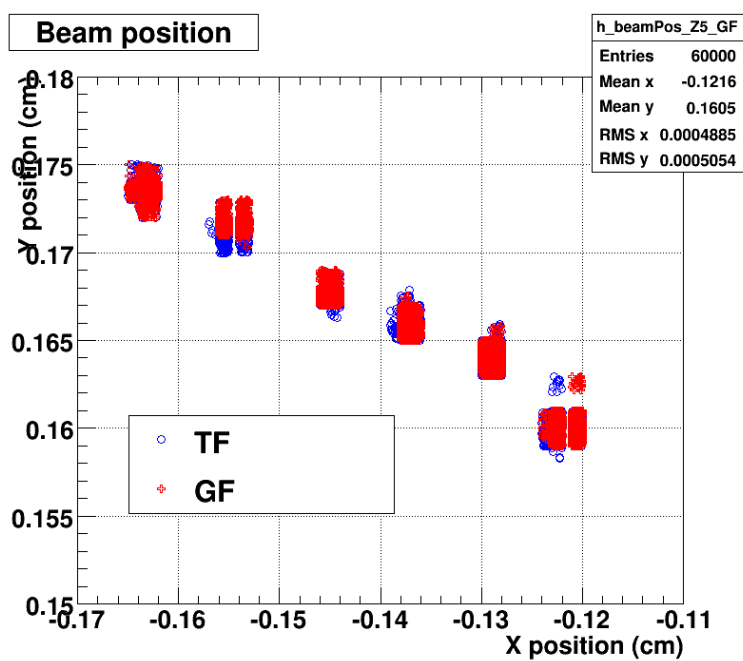


Figure 41: GF vs TF++: X and Y coordinates of beam position for all barrels.

The linear fit to the 6 values obtained in the different barrels result in the beam coordinates and slopes shown in fig. 42: we notice a difference of about 2 and 4  $\mu\text{rad}$  in  $\frac{dx}{dz}$  and  $\frac{dy}{dz}$ ; these displacements are within the resolution.

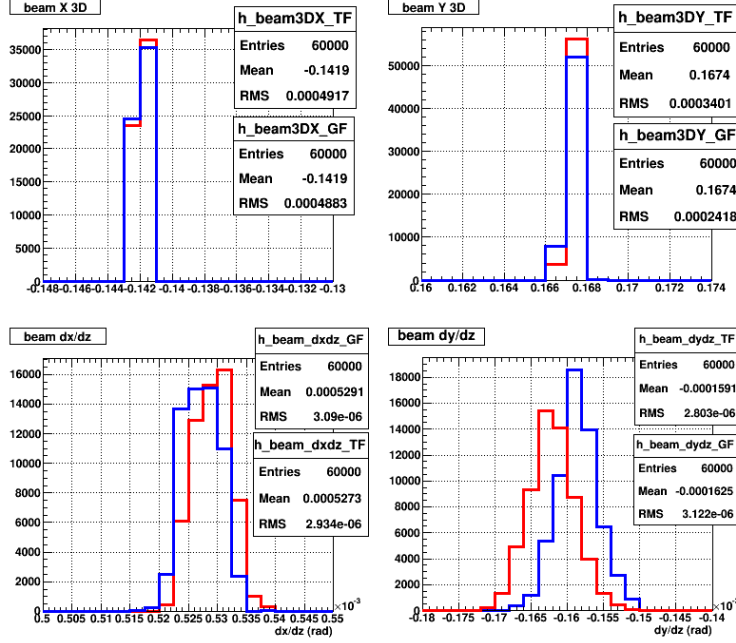


Figure 42:

The beam spot coordinates and beam line tilt are summarized for two different runs in fig. 43, showing an ACNET plot with  $x$ ,  $y$ ,  $\frac{dx}{dz}$  and  $\frac{dy}{dz}$ : The left part of the plot refers to a run with the standard SVT configuration, while on the right we can see the beam fit results for an EOS run with the GF used instead of the TF++.

## 7.6 Timing

The GB board, the last board of the SVT chain, measures for each event the overall SVT processing time as the difference between the L1 Accept time and the arrival time of the end event word. In fig. 44 the SVT timing is shown for the current system (blue) and for the system with the GF (red): the GF, even if performs more fits than the TF++, does not have any impact on the overall SVT timing.

## 8 Summary

We have presented the architecture and the performances of the Gigafitter, a new track fitter designed to upgrade the SVT system during the final period of CDF data taking.

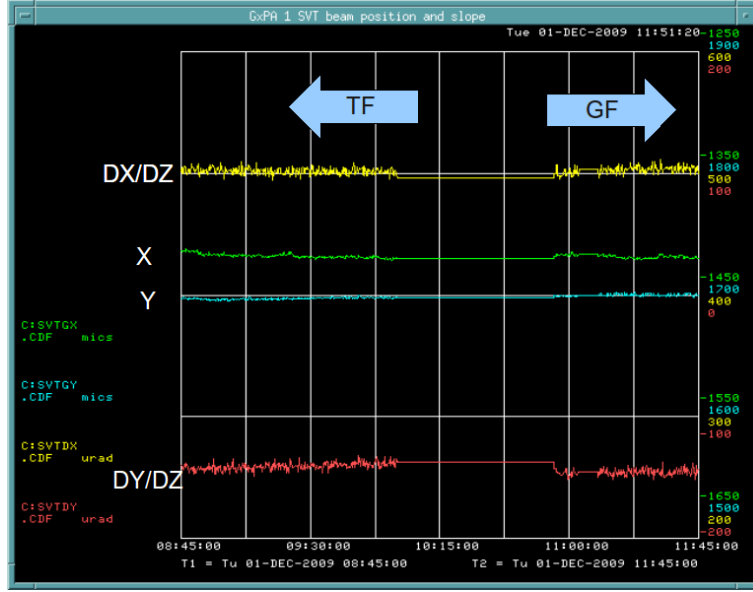


Figure 43: GF vs TF++ beam position: the plot shows on the left the beam coordinates and slopes measured with the TF++, on the right measured by the GF (EOS test).

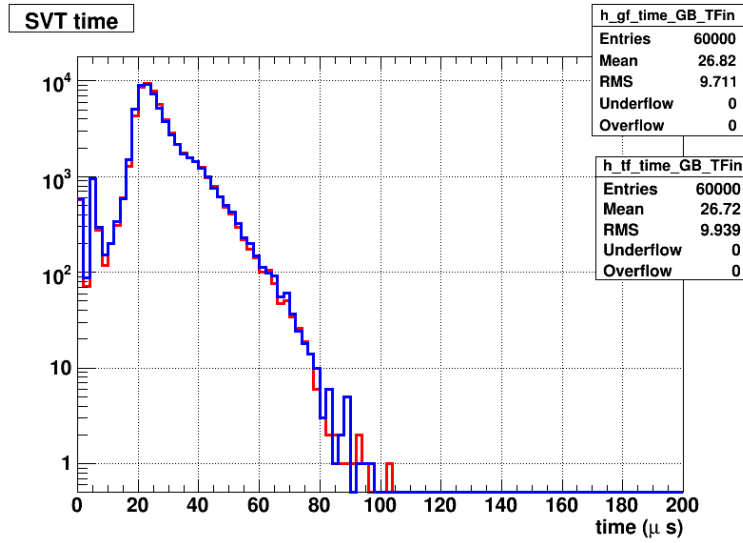


Figure 44: SVT overall timing measured by the Ghostbuster board: red line is GF, blue line TF++.

The GF is much more compact than the current system (1 board instead of 15) and allows to increase SVT performances in terms of track reconstruction efficiency.



At first it is going to be installed with exactly the same constant and pattern sets of the current system, so we have checked its performance against the TF++ in terms of track parameter quality, resolution on parameters, track reconstruction efficiency, purity and timing. The effect on the beam position calculation was also estimated. We found the GF assures the same resolution on tracks parameters as the current system and the same purity of the track sample, but with a gain of about 1.5% in track reconstruction efficiency, due to a different treatment of tracks having hits in all 5 SVXII layers. The differences in track parameter values are within the resolution as well as the differences in the beam position and slopes.

## References

- [1] S.Belforte et al., *SVT TDR (SILICON VERTEX TRACKER TECHNICAL DESIGN REPORT)*, CDF note 3108
- [2] A.Annovi et al, *Upgrade for the CDF SVT*, CDF note 6947
- [3] B.Ashmanskas et al., *Hardware Design and Specifications of the SVT Hit Finder*, CDF note 4849
- [4] M.Bari and A.M.Zanetti, *Merger Technical Specifications*, [http://www-cdf.fnal.gov/internal/upgrades/daq\\_trig/trigger/svt/BoardDocs/Merger/index.html](http://www-cdf.fnal.gov/internal/upgrades/daq_trig/trigger/svt/BoardDocs/Merger/index.html)
- [5] A.Annovi et al, *The AM++ board for the Silicon Vertex Tracker upgrade at CDF*, [http://www-cdf.fnal.gov/internal/upgrades/daq\\_trig/trigger/svt/BoardDocs/AMPP/specs/AMboard\\_Paper\\_IEEE\\_tiff.pdf](http://www-cdf.fnal.gov/internal/upgrades/daq_trig/trigger/svt/BoardDocs/AMPP/specs/AMboard_Paper_IEEE_tiff.pdf)
- [6] *The AM++ board*, [http://www-cdf.fnal.gov/internal/upgrades/daq\\_trig/trigger/svt/BoardDocs/AMPP/specs/am++\\_spec.ps](http://www-cdf.fnal.gov/internal/upgrades/daq_trig/trigger/svt/BoardDocs/AMPP/specs/am++_spec.ps)
- [7] J.Adelman et al., *The AMSRW board for the Silicon Vertex Tracker upgrade at CDF*, [http://www-cdf.fnal.gov/internal/upgrades/daq\\_trig/trigger/svt/BoardDocs/AMSRW/specs/AMSRW\\_paper\\_IEEE.pdf](http://www-cdf.fnal.gov/internal/upgrades/daq_trig/trigger/svt/BoardDocs/AMSRW/specs/AMSRW_paper_IEEE.pdf)
- [8] *The AMSRW board*, [http://www-cdf.fnal.gov/internal/upgrades/daq\\_trig/trigger/svt/BoardDocs/AMSRW/specs/AMSRW\\_doc/AMSRW.ps](http://www-cdf.fnal.gov/internal/upgrades/daq_trig/trigger/svt/BoardDocs/AMSRW/specs/AMSRW_doc/AMSRW.ps)
- [9] *HB++ Documentation*, [http://www-cdf.fnal.gov/internal/upgrades/daq\\_trig/trigger/svt/BoardDocs/HBPP/](http://www-cdf.fnal.gov/internal/upgrades/daq_trig/trigger/svt/BoardDocs/HBPP/)
- [10] J.Adelman et al, *SVT Track Fitter Upgrade*, CDF note 7872
- [11] *GhostBuster web page*, <http://fizzie.uchicago.edu/cdf/svt/gb/>

- [12] M.Bogdan et al., *CDF level 2 trigger upgrade - the Pulsar project*, Nuclear Science Symposium Conference Record, 2004 IEEE
- [13] A.Bhatti et al., *Level-2 Calorimeter Trigger Upgrade at CDF*, Real-Time Conference, 2007 15th IEEE-NPSS
- [14] A.Annovi et al., *The AM++ board for the silicon vertex tracker upgrade at CDF*, IEEE Trans. Nucl. Sci. Vol.53, 2006, pp. 1726-1731.
- [15] J.Adelman et al., *The 'Road Warrior' for the CDF online silicon vertex tracker*, IEEE Trans. Nucl. Sci. Vol.53, 2006, pp. 648-652.
- [16] R.Pegna et al., *A GHz sampling DAQ system for the MAGIC-II telescope*, Nucl. Instrum. Meth. A572, 2007, pp. 382-384
- [17] S.Amerio et al., "The GigaFitter for fast track fitting based on FPGA DSP arrays", Nuclear Science Symposium Conference Record 2007, NSS 2007, IEEE Vol.3, 2007, pp. 2115-2117.
- [18] *The BYPASS board*, [http://www-cdf.fnal.gov/internal/upgrades/daq\\_trig/trigger/svt/BoardDocs/Bypass/index.html](http://www-cdf.fnal.gov/internal/upgrades/daq_trig/trigger/svt/BoardDocs/Bypass/index.html)
- [19] *SPYMON web page*, <https://www-cdfonline.fnal.gov/internal/ops/svt/spymon/spymon.html>
- [20] *Trigmon Web Page*, <https://www-cdfonline.fnal.gov/internal/mon/consumer/trigmon/trigmon.html>