CDF/PUB/ONLINE/PUBLIC/7188
Version 1.0
August 13, 2004

Online Monitoring for the Silicon Vertex Tracker at CDF II

Marco Rescigno, Subir Sarkar, Lucia Zanello

Abstract

The online monitoring system (SPYMON) for the CDF Silicon Vertex Tracker (SVT) exploits several diagnostic devices integrated in the SVT design, checks the integrity of dataflow and displays the SVT performances. It also calculates online, from the measured track parameters, the Tevatron beam position parameters. We describe the architecture and the various components of the SPYMON project. In addition we discuss the initial experience and the impact of SPYMON on data taking in CDF.

1 Introduction

The CDF experiment at the Tevatron proton anti-proton collider has been significantly upgraded for the Run II which began in April 2001 and is expected to continue until the end of the decade allowing to collect a factor 50 to 100 more integrated luminosity than in the previous Run I. The upgrade was designed not only to cope with the different operating parameters of the Tevatron but also to extend CDF physics reach with better detectors and trigger system [1]. In particular the new trigger is specifically geared towards Heavy Flavour physics with the substantially increased bandwidth at level 1 and the new Silicon Vertex Tracker (SVT) which allows to reconstruct online tracks using silicon detector information and to select events characterized by the presence of high impact parameter tracks, hence allowing for the first time to trigger on hadronic decays of heavy flavour hadrons.

In the following we shall briefly describe the trigger and DAQ system of CDF, then the SVT and in particular some feature of its design devoted to error detection and to monitoring functions. We shall then describe the specific functions that such a monitor can accomplish and the various components of the software package, SPYMON, that were written to this end.

2 CDF Detector

CDF is a general purpose detector designed to study the proton-antiproton interactions at the Tevatron collider at Fermilab. A large volume tracker is contained inside a superconducting solenoid providing a uniform 1.4 T field, while calorimeters and muon identification systems are placed outside the solenoid. The tracking system is composed of a large volume drift chamber providing excellent momentum resolution and of three different silicon detectors providing up to 8 high precision points for accurate measurement of track impact parameter: Layer00, SVXII and ISL [2].

Layer00 is a single-sided low-mass radiation-hard silicon strip detector mounted directly on the beryllium beam pipe at a radius of 1.6 cm. The SVXII is a five-layer double-sided detector. It is composed of 3 barrels each approximately 30 cm in length, and covers 90% of the interaction region. The Intermediate Silicon Layer (ISL) provides one additional double sided layer in the central region and two in the forward regions, to improve forward track reconstruction using silicon only information.

3 CDF Trigger and DAQ

The CDF trigger has been completely rebuilt for Run II. It consists of a three level trigger system that accomplishes the task of reducing the 5 MHz input rate to the 75 Hz maximum allowed output rate from Level 3 which is written to tape. The first trigger level is deadtimeless, with a 42 event deep analog pipeline implemented on all the front-end electronics that allows 5μs latency for the level 1 decision. At this level coarse resolution information from the central drift chamber is used in the eXtremely Fast Tracker processor (XFT) to reconstruct tracks in the plane transverse to the beam line [3]. The maximum output rate out of level 1 is currently 20 KHz and is expected to increase significantly in the near future. At the second level trigger the SVT receives the list of XFT tracks and the silicon detector raw data from SVXII. The output of SVT is a list of high resolution tracks that is sent to the level 2 processor. The latter selects events containing high impact parameter tracks which greatly enhance the purity for heavy flavor signals. The maximum output rate into the third level trigger is 300 Hz. At this stage a farm of PC performs full reconstruction and further filters the events writing up to 20 MB/s of data to tape.

4 SVT

The SVT [4, 5], has a very short time (about 20μs, including the time needed to receive the SVXII data which is currently about 10μs) to complete its task in order to keep up with the maximum design rate of 50 KHz level 1 accept rate. For this reason the SVT design has been concentrated on maximum speed. The processing is widely parallelized; each 30 degree sector in azimuth of the SVXII detector is analyzed in parallel in one of 12 identical SVT "wedges".

Each one receives the data only from the corresponding sector of SVXII. The tracking process requires several steps, each performed by custom VME boards:

1. Hit Finders (HF) receive the raw digital data from the SVX readout over optical links and look for clusters of strips above a certain threshold (hits) and compute the local coordinate of those in each silicon sensor. Three Hit Finder modules are needed in each SVT wedge.

2. Mergers (MRG) combine the input from the 3 Hit Finders in each wedge with the list of tracks from the Level 1 track trigger processor (XFT) to form one single stream of data.

3. Associative Memory Boards (AMB) perform the core pattern recognition function of the SVT, by comparing all the incoming hits and tracks with a preloaded set of 32K roads. This function is done in a completely parallel way thanks to the use of custom Associative Memory chips [6]. The task is basically complete as soon as the last piece of data is input. Then, all the roads with 5 matches found (hits in 4 silicon layers and the XFT track) are passed in output.

4. Hit Buffers (HB) receive a copy of the same hits and XFT tracks in addition to the list of roads found in the AMB. They then combine the information in one single packet containing all the hits and tracks together with the road number and ship this information to the next stage.

5. Track Fitters (TF) then perform a full resolution fit of the hit coordinates found within each road using a linearized algorithm.

6. The data from the 12 Track Fitters are then merged together by a series of Merger boards. The last step is then performed by the Ghost Buster (GB) board that suppresses duplicate tracks and corrects track parameters for the beam position offset.

The whole system consists of over 100 VME boards housed in 8 crates. The data links between the boards are realized in a uniform way by using the same components and protocol everywhere. 21 bits of data and one strobe are used to send data to a FIFO on the receiving board. The latter can send back a HOLD signal in case its FIFO is getting full. The SVT has been operational since the early engineering runs of the Tevatron in Fall 2000, and has been fully integrated in the CDF trigger since February 2002.

5 Spy Buffers and Error Protocol

Much care must be taken to ensure the integrity as well as the synchronization of all the data paths inside SVT. Many potential problems can indeed be detected by the hardware itself. Parity is calculated for each data stream and matched at every stage. In addition, whenever two data streams are merged together (typically in a Merger board) event numbers are compared to check against potential loss of synchronization. Other anomalous conditions detected by the SVT boards occur when a FIFO gets full, or data format inconsistency with respect to the expected one is found.

An error protocol was specified at the design stage. All the errors that are detected are recorded in error registers and generate, upon enabling via VME an appropriate mask, special signals on the VME backplane (SVT_ERROR,CDF_ERROR), that propagate the error condition to the rest of the SVT system or to the CDF DAQ for appropriate action (like e.g. initiating a recovery procedure in which data acquisition is halted, all pipelines are emptied, and DAQ restarted again).

Of particular interest here is another feature of the SVT design, the Spy Buffer. Every data word output or input in any of the SVT boards is copied also in large (typically 128K word deep) circular memories, the Spy Buffer. These buffers typically contain the data corresponding to the last few ms. The Spy Buffers are used either as a data source for testing in situ the boards, or even the entire SVT, or as a powerful tool for debugging and monitoring. In fact it is possible to readout via VME the content of these buffers even during normal data taking mode. In this way it is possible to get a snapshot of what is going on in any stage of the SVT data processing at any moment. To readout the memory content it is only required to suspend ("freeze") the writing on the buffer. The latter effect can be obtained by using a dedicated backplane signal (SVT_FREEZE). This signal is driven by the Spy Control board (SC) [7] either by an appropriate VME controlled register in the Spy Control board, or by the detection, by the same board, of the SVT_ERROR signal issued by the board that generated the error condition. The SC board can even control the delay after which the SVT_FREEZE signal is issued in order to accumulate the needed amount of data in the buffers. There is one Spy Control board in each of the 8 SVT crates, and they are all connected in a daisy chain to allow exchange of different signals. One of the SC must be selected to operate as a Master. The SC board can control whether to propagate the error condition to the other crates via the G_ERROR signal on the daisy chain, and in turn can receive a G_FREEZE signal from the Master Spy Control. This feature allows the synchronization of the SVT_FREEZE on all the crates and is useful to check the data flow among boards sitting in different crates.

6 SPYMON Project

We have developed a software package (SPYMON) specifically devoted to exploit the built-in monitoring and debugging feature of the SVT design. The functions that such a monitoring program can perform are:

1. Collect statistics of the errors detected by SVT boards
2. Interact with the Spy Control boards
3. Collect samples of Spy Buffer data for high rate monitoring of SVT data quality by producing histograms of key quantities
4. Collect samples of Spy Buffer data for monitoring data integrity by using software emulation

5. Collect on disk samples of Spy Buffer data for debugging

6. Use the SVT track parameters from Spy Buffer data for a fast beam position monitor

While some of the above tasks are possible only by using Spy Buffers because otherwise the corresponding data is unavailable, there is some overlap in scope with the monitoring tools that are available also for SVT in the CDF online monitoring framework. The latter consists of programs based on the standard CDF offline code that analyze a fraction, of the order of 1%, of the data that is written out of the Level 3 filter [8]. Such monitoring programs are generally slow in collecting enough data for real-time tasks like the beam position monitor or even for simpler task like checking the impact parameter resolution and occupancies for each of the 72 single detector elements that send data to SVT. Moreover using the Spy Buffer we have access to a representative sample of events accepted by level 1 trigger. Given the large rejection of the order of 1000 inherent in the SVT trigger path, it is of particular importance to continuously monitor also events that would be later rejected.

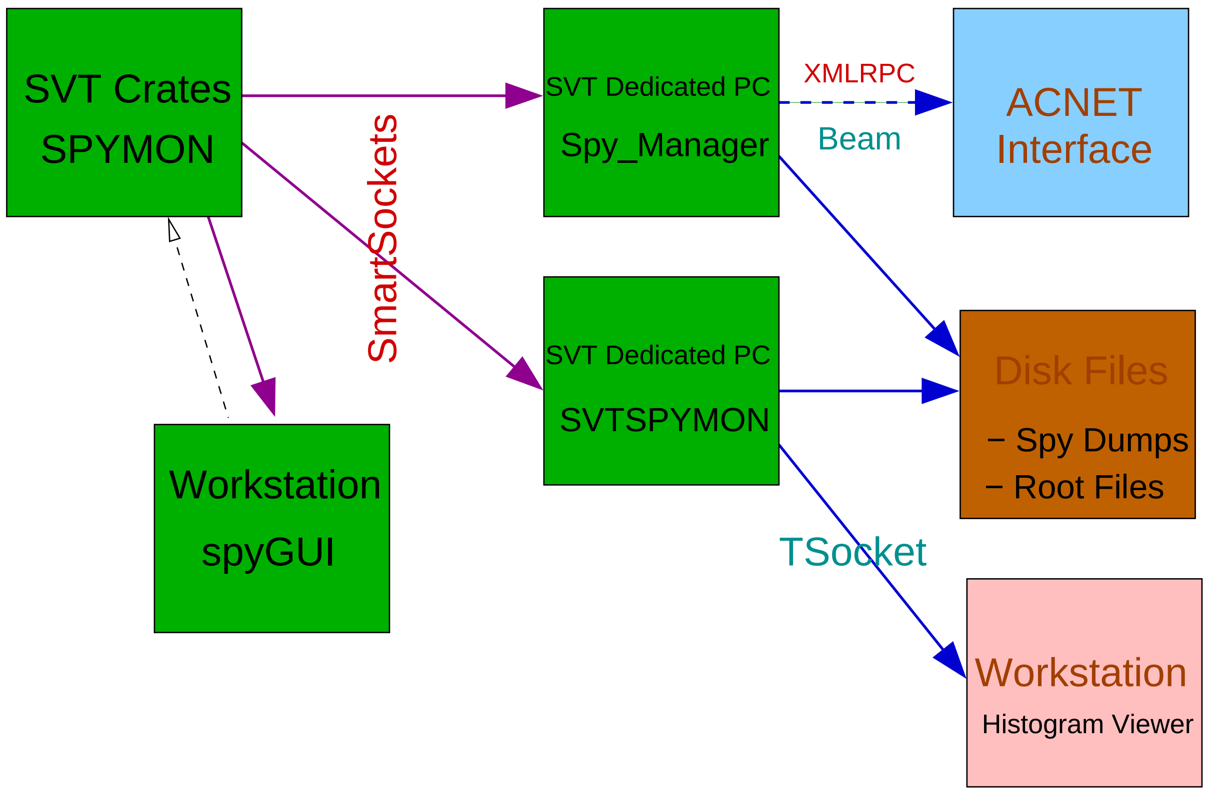

Figure 1: Diagram showing the various programs in the SPYMON package and their connections via either SmartSockets or Root socket (TSocket).

7 SPYMON Architecture

SPYMON has been developed as a collection of several programs running in parallel and communicating through messages built around the SmartSockets product from Talarian [9]. The latter is extensively used in the highly distributed CDF online environment for message passing and for shipping commands from the online computers running the data acquisition to the VME front-end processor. SmartSockets can be used to pack complex objects in messages while the decoding of the messages is made transparent across the different platforms and different programming languages that are used in the CDF online environment. SmartSockets uses a publish-subscribe mechanism, where all the messages are directed towards a server that relays them only to the clients that have subscribed to that particular class of messages.

The core tasks of SPYMON (readout of Spy Buffer data, interaction with the Spy Control boards, readout of error register as well as the analysis of the data) are performed in the VME crate controller CPU. In this way we avoid large data transfer over network and make the least possible impact on the main data acquisition task running on the VME CPU. In fact, accessing the VME bus with a network request has a very low priority when data acquisition is running and is not a viable option at the nominal Level 2 rate.

Each SVT crate has been equipped with one Motorola MVME 2304 board, based on a 333 MHz ppc processor with 128 MB of RAM. We run VxWorks 5.3.1 [10] as in all other front-end VME processor in CDF. The larger memory with respect to the ones used elsewhere in the CDF DAQ is justified by the large amount of configuration data needed by SVT. In fact the patterns and constants are cached in the processor memory and are thus readily available for software emulation of the boards. The SPYMON package includes the following programs that will be described in the following sections (Figure 1):

A. SPYMON is the core task running on the CPU crates

B. Spy Manager works as a centralized place where all messages are collected and processed

C. SVTSPYMON is used to convert messages representing histograms in ROOT [11] format for display

D. spyGUI is a Java Graphical User Interface that displays all the message types and allows the user to communicate with the core monitoring tasks running on the crate CPU by sending commands and configuration data.

7.1 SPYMON

In the SVT crate CPU we run the standard data acquisition task and the SVT monitor (SPYMON), unlike other CDF systems. Careful tuning of the VxWorks priority levels and of the VME software (the Fermilab package FISION) has ensured that the two tasks can run concurrently without spoiling the respective real-time performances. SPYMON detects the boards present in the crates and configures itself accordingly. The SmartSockets messages defined are:

1. Crate Status message, that has a hierarchical substructure:
    - Board Status message, contains counters on the errors detected in each board and few other board level status bits
    - Spy Buffer Status message, contain the status of the analysis of each Spy Buffer data
2. Histograms messages, both 1D and 2D, are defined as large vectors containing histograms contents, together with an identification string and other strings that specify graphical attributes
3. Spy Dump, contains an hexadecimal representation of the data in each spy buffer, and published either on request or when an SVT error is detected
4. Beam Fit message contains information on the results of the beam fit.

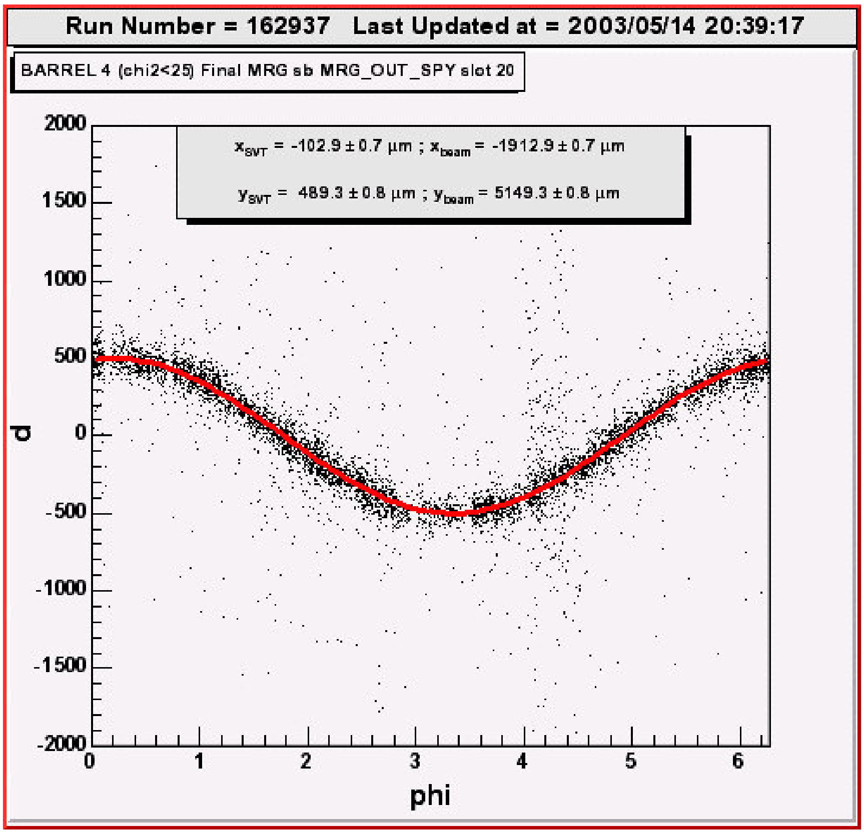

Figure 2: SVT impact parameter vs ϕ distribution with superimposed the online beam position fit. Plot like this are used to monitor the quality of the beam fit.

SPYMON is normally idle but can detect the status of the SVT_FREEZE signal on the VME backplane and, in case it is on, it will activate itself, poll from the SVT boards all the Spy Buffer data needed for further analysis, and reprogram the Spy Control for a new cycle. In case of no errors the SPYMON instance running on the crate hosting the Spy Control Master will issue the G_FREEZE signal at regular, user defined, time intervals, so that the monitoring process in each crate can collect data for histogramming and other analysis. Presently the full analysis is performed in less than 2 seconds, dominated by the time needed for publishing all the output messages over the network.

The beam fit is currently performed in the fanout SVT crate where the tracks from the 12 Track Fitters are merged in a single stream and sent to the Ghost Buster board for beam offset correction. In this case SPYMON collects data with a faster cycle, and finds the beam coordinate by fitting the two dimensional impact parameter vs ϕ distribution of SVT tracks. Details of the procedure are described elsewhere [12]. SPYMON can collect events at an average rate of approximately 2 KHz, ensuring a valid beam fit after just 2 min of data taking at the beginning of a new Tevatron store. After the initial fit the beam will be followed with updates of the beam position parameters every 30 sec. At each new valid fit SPYMON writes the new beam position values in appropriate registers of the Ghost Buster board for the automatic subtraction of the beam offset from the SVT track impact parameter.

Since the SPYMON task runs on a real-time CPU, critical to the data taking, the reliability and the robustness of the code is very important. SPYMON has reached a great degree of reliability, running continuously for very long period of time.

7.2 Spy Manager

This process receives all kind of messages and is responsible for writing log-files and Spy Dumps on disk, and for updating web pages with a summary of the SVT status. It will signal the lack of monitor data from any of the SVT crates if it does not receive messages for some predefined time interval.

It also acts as an interface to the ACNET system that the Fermilab beam division uses for storing all monitor data. The interface uses currently the XML-RPC protocol.

7.3 SVTSPYMON

SVTSPYMON receives histograms and other monitoring data (e.g. Beam Fit messages) from the SVT Online Monitoring (SPYMON) processes running on the SVT crates. The communication between the SPYMON processes and SVTSPYMON takes place through SmartSockets [9].

The monitoring data are then converted to Root objects (histograms, functions) and made available to clients like the standard CDF Root histogram browser, namely HistoDisplayMain. SVTSPYMON uses the DisplayServer classes developed within the CDF Consumer Framework [8]. In this way the operator in charge of the data quality check during data taking can access the results of SPYMON exactly in the same way as for the other monitor programs that have a completely different data source (i.e. the full event after Level 3).

The histograms that SVTSPYMON produces are divided into 3 categories,

1. Monitoring; contains error histograms detected in each individual SVT board and recorded in the Spy Buffers. A Timing sub-folder contains histograms to monitor the event processing time of SVT.

2. Tracking; stores histograms related to SVT tracking performance. This folder is divided further into sub-folders related to

   i. Track parameters (impact parameter, ϕ, χ^2 etc.)

   ii. Road matching, hot channel distribution etc.

   iii. Beam Fit (with 2D i.p. vs ϕ for each barrel)

   iv. Hit coordinates

   v. Results of Ghost-Busting (SVXII barrel vs wedge map, corrected impact parameter vs ϕ, etc.)

   vi. XTRP track parameters (ϕ, p_t etc.)

3. Expert; sub-folders are created under this folder for each SVT Crate which sends data.

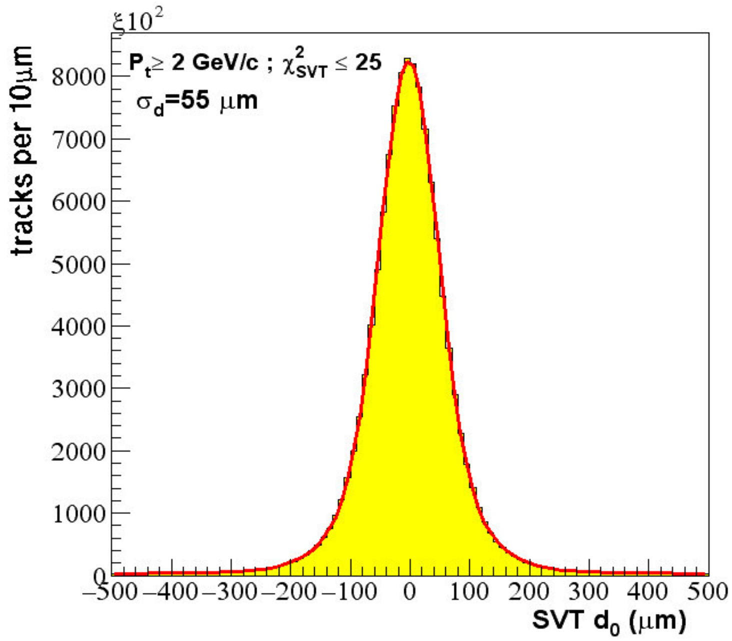

Figure 3: SVT impact parameter distribution as fitted online to monitor SVT resolution. The actual width of the distribution is a convolution of the beam spot width and the actual SVT resolution.

The currently available histograms are saved in a Root file every 60 seconds. The file gets overwritten during the next iteration. At the end of the run the final histograms are saved in a Root file for offline analysis. We also create two postscript files with the most important monitoring histograms, one of which contains histograms from the currently available data while the other contains the same set of histograms from the last good run. The postscript files are always available for WEB browsing.

A typical impact parameter resolution fit from SVTSPYMON is shown in Figure 3.

SVTSPYMON has proved to be a useful tool, not only to the SVT experts, but also to the Silicon experts. SVTSPYMON works as a beam position monitor also.

As discussed above, erroneous conditions of SVT, SVXII strip noise, movement of the beam can be detected immediately by looking at the SPYMON results.

7.4 spyGUI

spyGUI is a Java based graphical user interface to SPYMON. It collects informations from the SVT Crate processes and displays them in real time after suitable transformations. spyGUI tries to perform all the tasks from within a single application that are individually carried out by different components of the entire SVT Spy buffer analysis system. Among other things, spyGUI,

1. displays SVT hardware and analysis errors in detail,
2. decodes raw data into physics quantities; offers an interactive plotting widget based on Java Analysis Studio (JAS) [13],
3. works as an Online Beam Position Monitor,
4. works as a JAS based Histogram Producer and Viewer,
5. configures SPYMON analysis options. This was the initial goal with which spyGUI was created,
6. works as an SVT Database Browser,
7. performs simple error recovery.

spyGUI is maturing into a useful application. Individual components of spyGUI can be used as independent applications. This has vastly simplified the usage of spyGUI. This scheme is realistic too since much of what spyGUI does is alternative to already existing tools. A full-featured, matured version of spyGUI will be a powerful tool for SVT online monitor and debugging.

8 Conclusions

The SPYMON project is now almost completed. All the components are mature and stable and provide routinely important monitoring data both to expert and non expert users. It has been also an important tool for the successful commissioning of the SVT in the early Run II period.

9 Acknowledgments

This work has been partly supported by EEC contract: HPRN-CT-00292-2002.

References

[1] R. Blair *et al.*, [CDF Collaboration], FERMILAB-PUB-96-390-E 1

[2] A. Sill *et al.*, Nucl. Instrum. Meth. A 447, 1 (2000) 2

[3] E. J. Thomson *et al.*, IEEE Trans. Nucl. Sci. 49, 1063 (2002) 3

[4] S. Belforte *et al.*, IEEE Trans. Nucl. Sci. 46,933 (1999) 4

[5] W. Ashmanskas *et al.*, Nucl. Instr. Meth. A 447, 218 (2000) 4

[6] F. Morsani, S. R. Amendolia, S. Galeotii, D. Passuello, L. Ristori, N. Turini, Nucl. Instrum Meth A315:446(1992) 3

[7] F. Spinella *et al.*, IEEE Trans. Nucl. Sci. 48,1286 (2001) 5

[8] K. Maeshima *et al.*, to be published in CHEP 2003 proceedings 6, 7.3

[9] http://www.talarian.com 7, 7.3

[10] http://www.windriver.com/products/vxworks5/index.html 7

[11] R. Brun, F. Rademakers, http://root.cern.ch 7

[12] http://www-cdf.fnal.gov/upgrades/daq_trig/trigger/svt/Beam/Docs/BeamAlignment_LR.html 7.1

[13] http://jas.freehep.org 2