Triggering on Heavy Flavors at Hadron Colliders

Luciano Ristori[1] and Giovanni Punzi[2]

[1]Istituto Nazionale di Fisica Nucleare, I-56127 Pisa, Italy; email: luciano.ristori@pi.infn.it
[2]Department of Physics, University of Pisa, I-56126 Pisa, Italy; email: giovanni.punzi@pi.infn.it

Annu. Rev. Nucl. Part. Sci. 2010. 60:595–614

First published online as a Review in Advance on July 20, 2010

The *Annual Review of Nuclear and Particle Science* is online at nucl.annualreviews.org

This article's doi: 10.1146/annurev.nucl.012809.104501

0163-8998/10/1123-0595$20.00

Key Words

trigger, electronics, particle physics, track reconstruction, pattern recognition, *b* physics

Abstract

For a number of interesting processes in the sector of heavy flavors, the quality of measurements made at hadron colliders is very similar to the quality achieved at e^+e^- colliders (known as *B* factories). The key to performing such measurements in a hadron environment is the ability to select rare processes from background in real time, that is, to trigger on them. Two distinctive features of heavy-flavor decays have been used for this purpose: the presence of leptons in the final state and secondary vertices produced by the relatively long lifetime. The selection of events based on long lifetime, although technically very challenging, is the most inclusive of all such techniques, allowing access to the widest range of channels. The focus of this review is on the innovative concepts that permitted the reconstruction of tracks produced in hadron collisions with sufficient speed and accuracy for use at trigger level to detect heavy-flavor decays.

Contents

1. INTRODUCTION

The study of hadrons that contain a b quark is very effective in testing our understanding of the fundamental interactions and the validity of the Standard Model (1–3) and, possibly, in detecting hints of new physics. Measurements of quantities such as cross sections, branching ratios, lifetimes, masses of new states, *CP* asymmetries, and neutral meson oscillation frequencies have improved our understanding of the Standard Model, constrained the elements of the CKM (Cabibbo-Kobayashi-Maskawa) quark-mixing matrix (4, 5), and provided guidance to the planning of future machines and experiments. Numerous measurements have been performed in experiments at the *B* factories (the BaBar and Belle Collaborations) and at the Tevatron (the CDF and DØ Collaborations) (6; also see **http://pdg.lbl.gov**).

1.1. *B* Production at Hadron Colliders

Due to large cross sections, hadronic collisions at high energy are the most effective way to produce large quantities of heavy-flavored hadrons. The largest rates are currently provided by the Tevatron, which produces of order 10^{11} b-flavored particles per year. However, large quantities of other particles are produced as well, and b production represents only a fraction (of order 10^{-3}) of the total. This large background was the reason for the design of dedicated e^+e^- colliders for the selective production of B particles through the decay of $b\bar{b}$ resonances [$\Upsilon(4s)$, $\Upsilon(5s)$]. These colliders were named B factories not because of the large production rate, which is many times lower than at hadron machines, but because of the specificity of the production of b-flavored particles with a much more favorable signal-to-background ratio.

Physics results obtained in recent years have shown that, for a number of interesting processes in the heavy-flavor sector, the quality of measurements performed at hadron colliders can be very similar to those performed at the B factories in terms of both precision and the number of events collected. An advantage of hadron colliders is that, due to their production mechanism, more types of heavy-flavored particles can be studied, including B_s, B_c, and B baryons. However, the key to performing such measurements is the ability to select such processes from the background in real time—that is, to trigger on those processes. Triggering on heavy-flavor events is the central topic of this review.

1.2. The Trigger

The collision rate at hadron colliders is much higher than the recording capability of experiments. The only way to exploit abundant heavy-flavor production is to select collisions in real time by recording only a small fraction for further processing and immediately discarding the data that contain only uninteresting, light-flavored processes.

The Tevatron, for example, produces approximately 2.5 million collisions per second at each interaction region. Two beams of protons and antiprotons are accelerated up to ~1,000 GeV in opposite directions; they then collide at two designated points along the circumference. Dozens of particles are created from vacuum in each collision. The detectors installed around the interaction regions record all the signals generated by all these particles. Approximately 500 Kbytes are needed to store all the information for each collision.

With current technology, the required storage space makes it impossible to permanently store the data from each and every collision. A typical large hard drive (~1 Tb) would fill up in less than 1 s, so it is necessary to be selective and keep only the events in which we are interested. If, for example, we wish to study collisions in which b quarks are produced, we would retain the data from approximately one in a thousand collisions.

All the signals from the different detector elements must be processed by complex computer algorithms before a clear picture of what occurred in each collision can emerge. This processing takes typically of order of 1 s for a modern central processing unit (CPU) per collision. It would take millions of CPUs working in parallel to process 2.5 million collisions per second and send only the data of interest to permanent storage. Present hadron collider experiments (CDF and DØ) use computer farms comprising ~1,000 CPUs to process events in real time; therefore, the event rate must be reduced by more than three orders of magnitude before full reconstruction can be afforded. Such selection is performed by a machine that can interpret the data from a collision, possibly with some approximation, in much less time than any commercial CPU needs. We term such a device a trigger.

The trigger is a very important component of any experiment at a hadron collider such as the Tevatron or the LHC. Its quality is primarily what limits the phenomena that can be observed and the measurements that can be made. Events that do not pass the trigger requirements are lost forever.

2. TRIGGERING ON HEAVY FLAVORS

Two features of heavy-flavor decays have been exploited to distinguish them from the background of light particles: (*a*) the presence of leptons and (*b*) the relatively long lifetime of their ground states, which can decay only through weak interaction. Although detection of leptons in the final state is a powerful tool in selecting event samples with a high heavy-flavor content, it limits the processes that can be studied to a small subset of the total. Conversely, the long lifetime of states

that decay by weak interaction creates so-called secondary vertices at the point in space where they decay, irrespective of the specific decay mode, and allows the collection of very inclusive samples of heavy flavors. For this reason, triggering on secondary vertices, although technically very difficult, has provided a large fraction of the flavor physics results from hadron colliders.

2.1. Leptons

The weak decay of b and c quarks proceeds through the production of a virtual W boson, which can decay into a charged lepton-neutrino pair. This decay leads to the production of either an electron or a muon, with a ~10% probability of each (production of tau leptons is less probable due to their high mass). These leptons can be used to tag the presence of heavy flavors because they stand out as a signature of their weak decay. Production of leptons from other sources is limited and easily recognizable (e.g., the Drell-Yan process), and those from weak decays of the lighter s quark are produced with significantly lower transverse momenta. However, the presence of a neutrino, which escapes detection, implies that it is not possible to completely reconstruct the decay with the momenta of all the particles produced in the final state. This limitation is more important in hadron collisions than at e^+e^- colliders, where knowledge of the initial total energy helps constrain the kinematics of the decay. Leptons are also used to recognize heavy-flavor decays through the $J/\psi \to e^+e^-$ and $J/\psi \to \mu^+\mu^-$ decays. Some interesting B hadron decays produce a J/ψ particle in the final state; thus, requiring the presence of a pair of leptons with the correct invariant mass is a good way to select them at trigger level. In hadron collisions at high energies, a sizable percentage (~20%) of all the produced J/ψ particles comes from B decays. Other important B decays have nonresonant pairs of leptons in the final state and can also be selected using this method.

2.2. Secondary Vertices

Detecting the presence of a secondary vertex in a particular event is not a trivial task: It requires one to reconstruct tracks and to recognize that some of these tracks do not originate from the collision point. When a long-lived particle decays after traveling some distance from the point where it was created, the trajectories of the decay products generally do not point back to the collision point. The distance of closest approach of the extrapolated trajectory to the collision point, known as the impact parameter, provides the key signal for the detection of secondary vertices. An impact parameter that is significantly different from zero is the primary clue that the track originates from a secondary vertex and that it is a good candidate for the product of the decay of a long-lived particle.

A typical situation is shown in **Figure 1**. The primary vertex is where the primary collision took place. Most of the particles produced in the collision originate from there. If a long-lived particle is created in the primary collision, it will travel some distance L (decay path) before decaying, and will generate secondary tracks originating from a secondary vertex. If we extrapolate back the trajectory of a secondary track, we find that, in general, it will miss the primary vertex by some distance d (impact parameter),

$$d = L \sin\theta,$$

where θ is the angle between the direction of the secondary track and the direction of the parent particle, and

$$L = \beta\gamma c t,$$

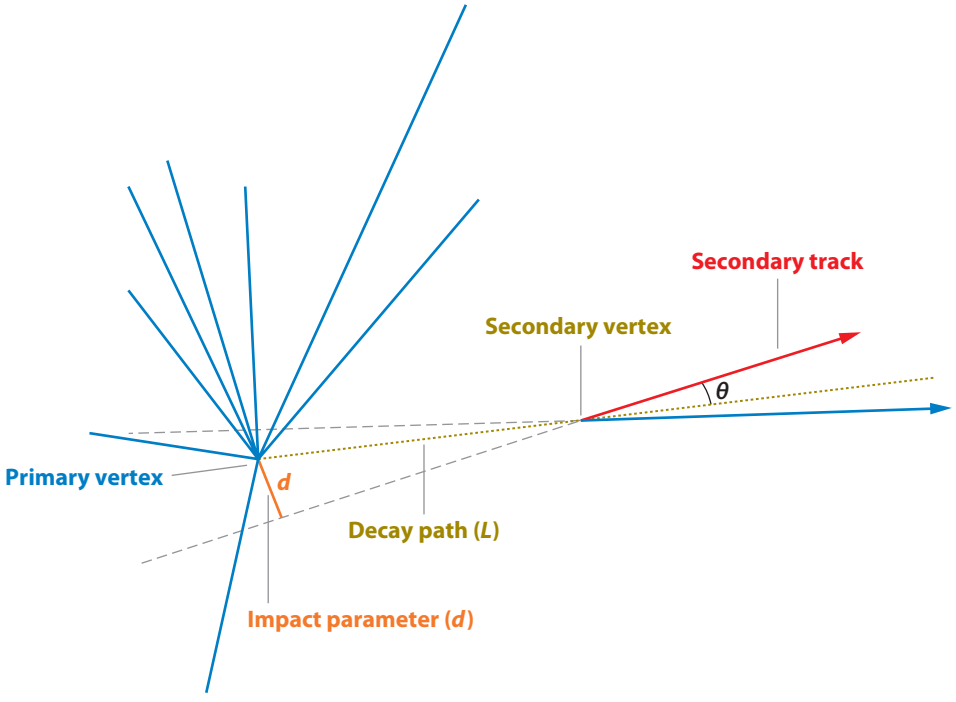


Figure 1

When a long-lived particle decays after traveling some distance, the trajectories of the decay products do not point back to the collision point. The distance of closest approach of the extrapolated trajectory to the collision point is known as the impact parameter.

where β and γ are the usual parameters of the Lorentz transformation from the rest frame of the parent particle to the laboratory, t is the proper decay time, and c is the speed of light.

If the particles involved in the process are all relativistic, then on average, θ is of the order of $1/\gamma$, and for $\gamma \gg 1$, the impact parameter is

$$\langle d \rangle \simeq \beta\gamma c\tau(1/\gamma) = \beta c\tau \simeq c\tau,$$

where τ is the lifetime of the parent particle. Note that, with these approximations, the average size of the impact parameter is independent from the Lorentz boost γ.

For particles containing the b quark, $c\tau$ is of the order of 450 μm, and for particles containing the c quark, $c\tau$ can be as small as 123 μm. These values set the scale for the precision of the impact-parameter measurement needed to detect secondary vertices from heavy-flavor decay to the order of a few tens of micrometers.

The first experiment that attempted to exploit the presence of secondary vertices to trigger on heavy-flavor decays for a real measurement was probably WE82 at CERN (7). This experiment searched for b and c production in pion-nucleon collisions at an energy of 350 GeV in the laboratory frame. WE82 used six planes of detectors to reconstruct tracks in real time, scanning for secondary vertices in a total time of 35 μs. It selected b and c quarks with a rejection factor of order 10 against lighter quarks by requiring three primary tracks accompanied by two nonprimary tracks. This device probably achieved the maximum possible performance with the technology available at that time; however, given the ~1-MHz interaction rate and the initial signal-to-background ratio

for heavy flavors of only $\mathcal{O}(10^{-6})$, the device was only usable in conjunction with other triggers, and it could only provide modest help.

A much more complex and ambitious impact-parameter trigger was designed for the BTeV experiment (8–10). The detector, which was never built, would have used an extensive telescope comprising 30 layers of silicon detectors and a complex tracking algorithm, supported by a large array of computers. The trigger was designed to work in three dimensions, with an input rate of 7.6 MHz. The trigger was intended to build tracks beginning from pairs of hits in adjacent detectors, then add a third hit and trigger on B events by using two detached tracks with large enough transverse momentum in the same arm of the detector.

The DØ experiment can perform track reconstruction and measure impact parameters in real time (11); however, the general organization of the DØ trigger and its timing prevent the implementation of a pure impact-parameter trigger. Impact-parameter information can be used in DØ only as a complement to other requirements, such as the presence of leptons or a high-P_T jet.

The LHCb experiment plans to trigger on secondary vertices by using an array of standard processors to reconstruct tracks in real time (12). Currently, CDF is the only experiment to have successfully operated a pure impact-parameter trigger that yielded numerous physics results. Our review is devoted to the description of the CDF trigger's design principles and some of the innovative underlying ideas.

3. SUPERFAST TRACK RECONSTRUCTION

The task of reconstructing the trajectories of charged particles (tracks) in the detector can be broken down into two steps. The first step is track finding; the second step is track fitting. Track finding involves choosing, from among all the hits present in the detector, those hits caused by the same particle. Doing so may be difficult if many particles are present in the same event and if noise hits are also present, as is normally the case. Track fitting involves extracting track parameters (e.g., the particle momentum and impact parameter) from the coordinates of the hits. These steps are usually executed multiple times to find and fit all the tracks in each event, and they require sophisticated algorithms that typically consume several seconds of CPU time per event. To use this information in the trigger, we must find a way to perform a similar task within approximately a few microseconds—that is, 100,000 to 1 million times faster than a commercial CPU.

A possible solution to this problem emerged toward the end of the 1980s, when very large scale integration (VLSI) technology, which brought us microprocessors and the home computer, became affordable not only for industry giants but also for universities and small research projects. Some universities taught classes in which students designed VLSI circuits and had them fabricated (15). At that time it was realized that VLSI technology could be used to build a machine that could reconstruct tracks in a very short time.

3.1. Pattern Recognition by Brute Force

A time-consuming task in track reconstruction is pattern recognition: One needs both to try many different combinations of hits to find those that satisfy all geometrical constraints and to discard false hits arising from electronic noise or uncorrelated particles. All those different combinations are usually tested sequentially, and because there are so many of them, a great deal of time is needed to do so. But what if we could test these combinations in parallel, all at the same time? Use of VLSI technology makes it possible to integrate thousands of logic elements on a small silicon

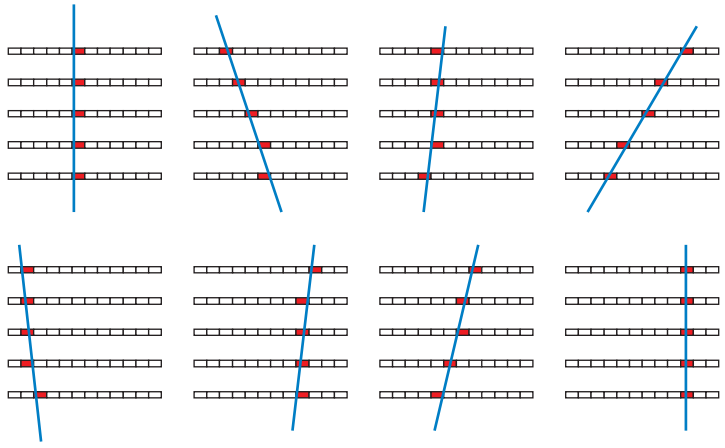


Figure 2

An ideal detector with five sensitive layers, each of which is divided into a number of segments. A charged track crossing the five layers fires one and only one segment per layer, producing a pattern of hits. If we let the track parameters span a certain volume of the phase space, a corresponding finite set of distinct patterns is generated.

chip and implement pattern recognition as a massively parallel algorithm. These ideas led to the invention of a special VLSI system: the associative memory device (16).

We use a simple example to illustrate the principle of operation of the associative memory device. **Figure 2** shows an ideal detector with five sensitive layers, each of which is divided into a number of segments. These segments could be, for example, five parallel layers of microstrip silicon detectors, as seen from along the strip direction. For the sake of simplicity, we assume that a charged track crossing these five layers fires one and only one strip per layer, thereby producing a pattern of hits. If we assume that each strip is uniquely identified by a coordinate, then each pattern consists of five coordinates (one per layer). If we change the track parameters enough, the resulting pattern changes. If we let the track span a certain volume of the parameter space, a corresponding, finite set of distinct patterns is generated. We term such sets pattern banks. **Figure 2** shows several tracks, each of which contributes a different pattern to the pattern bank.

Figure 3 shows a typical pattern-recognition problem consisting of so-called noise hits mixed together with hits from a real track. We need to implement an algorithm that is able to tell the difference between the scenarios shown in the two panels: In **Figure 3*a*** there is a combination of hits, one per detector layer, that corresponds to a pattern stored in the pattern bank and could therefore have been produced by a track; in **Figure 3*b*** no such combination exists. As discussed above, we usually solve this problem by trial and error by sifting through all possible combinations of hits, one per layer, until we find a combination that is compatible with having been generated by a single track. However, we can also proceed in the opposite way by scanning the pattern bank until we find a pattern whose coordinates correspond to a hit present in the detector. The advantage of the second approach is that, as we discuss in the following section, it lends itself easily to a massively parallel implementation.

Imagine that we have generated a pattern bank as described above. Now imagine that instead of storing all the patterns in normal random-access memory (RAM) (17–19), we store them in a device that is similar to a RAM but for which each storage element has additional intelligence built in: In addition to storing information, it can ascertain whether all of its coordinates (one per layer) correspond to a detector segment that was actually fired. If this method is used, the process

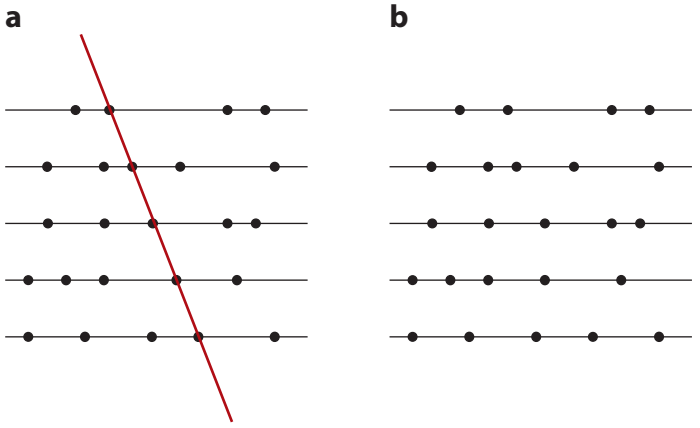


Figure 3

A typical pattern-recognition problem, consisting of noise hits superimposed on a track. The problem is to implement an algorithm that can tell the difference between the two cases shown in panels *a* and *b*, namely that in panel *a* we can find a combination of hits, one per detector layer, that can be produced by a single track, whereas in panel *b* no such combination exists.

can go much faster: We simply feed all the coordinates of all the fired segments into this device, and each pattern can determine whether there is a match. The time needed to perform pattern recognition is therefore equal to the time needed to extract the coordinates of all the hits from the detector and is independent of the size of the pattern bank.

3.2. The Associative Memory Device

The associative memory is based on the general concept of content-addressable memory (CAM) (21). In a CAM device, data are stored in the same way as in a RAM device (17–19) but are retrieved differently. To retrieve data from a RAM, one provides an address and obtains the data stored at that address. To retrieve data from a CAM, one provides the data and obtains the address at which that data is stored. CAM's typical applications are, for example, in cell phone and computer network address routing, where it is important to perform table lookups in the shortest possible time.

The schematics of a single CAM bit implemented as a CMOS (complementary metal-oxide semiconductor) device is shown in **Figure 4**. For more information on CAM architecture, see References 21 and 22.

Figure 5 represents the internal structure of the associative memory device that was designed for the CDF experiment and successfully used for many years (20). In the following discussion, we refer specifically to this implementation.

The associative memory device for the CDF experiment was designed for six detector layers. Each pattern is composed of six coordinates (one per layer), and each coordinate is encoded with 12 bits. Each coordinate can be accessed for reading or writing using the address bus to select one specific pattern and the layer bus to select one specific detector layer. Data are transmitted over the data bus and routed to or from the appropriate detector layer by a multiplexer driven by the layer bus.

The same data bus and layer bus used for writing are also used to feed hits from the detector to the associative memory device. One hit is encoded as a three-bit layer number (0–5) and as a 12-bit coordinate (0–4,095). The coordinate of each hit is transmitted to the correct layer and compared

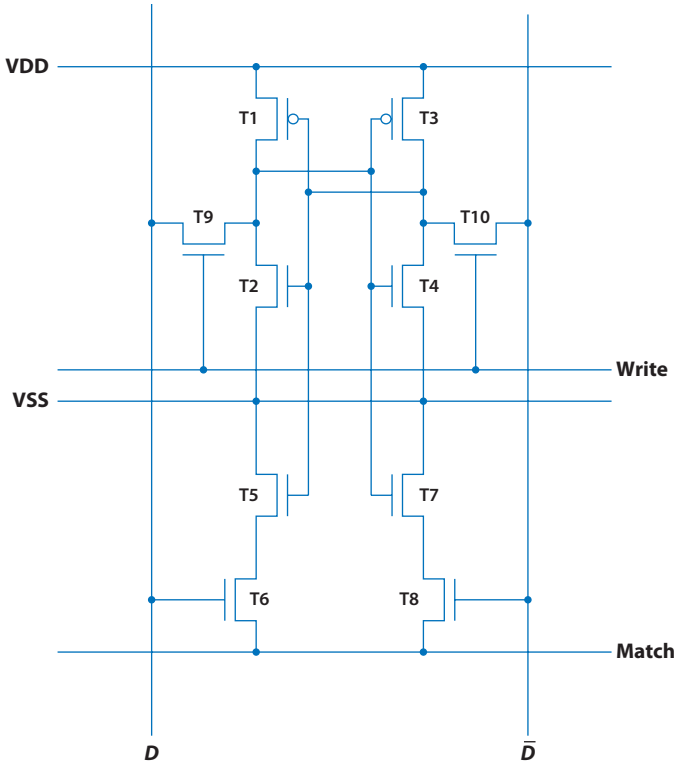


Figure 4

The schematics of a complementary metal-oxide semiconductor implementation of one bit of content-addressable memory (CAM). VSS and VDD refer to the source and drain polarization voltages, respectively. Transistors T1–T4 form the positive feedback loop that creates the two-state system that holds one bit of information. Transistors T9 and T10 are activated by the Write line to write the memory with the content of the data lines (D and $\bar{D}$). Transistors T5–T8 form a comparator that pulls the Match line down if the content of the memory does not match the data on the data lines. By stringing together a number of CAM bits, one can build CAM words of any length.

to the corresponding memory content for all patterns in parallel. Every stored coordinate that matches the data on the bus causes the hit bit to be set. This process is repeated for all the hit coordinates in one event.

After all the coordinates have been fed, the number of hit bits is counted in parallel for all the patterns, and the match bit is set only if the count reaches six. This means that all the coordinates in each of the six detector layers have been detected and that they therefore correspond to a possible trajectory of a single track as stored in the pattern bank.

Thereafter, all the match bits are fed into a priority encoder (23), which drives the address bus in output mode. The address of the first match bit is thereby extracted from the associative memory device. The same match bit is then cleared, causing the address of the second match bit to be encoded on the address bus. The process is repeated until all the match bits have been cleared, effectively causing the list of the addresses of all matching patterns to be output sequentially. Interestingly, this process differs from RAMs in which the address bus is always used as an input: Here, the address bus is used as an input bus when the memory is written and as an output bus when the memory is read.

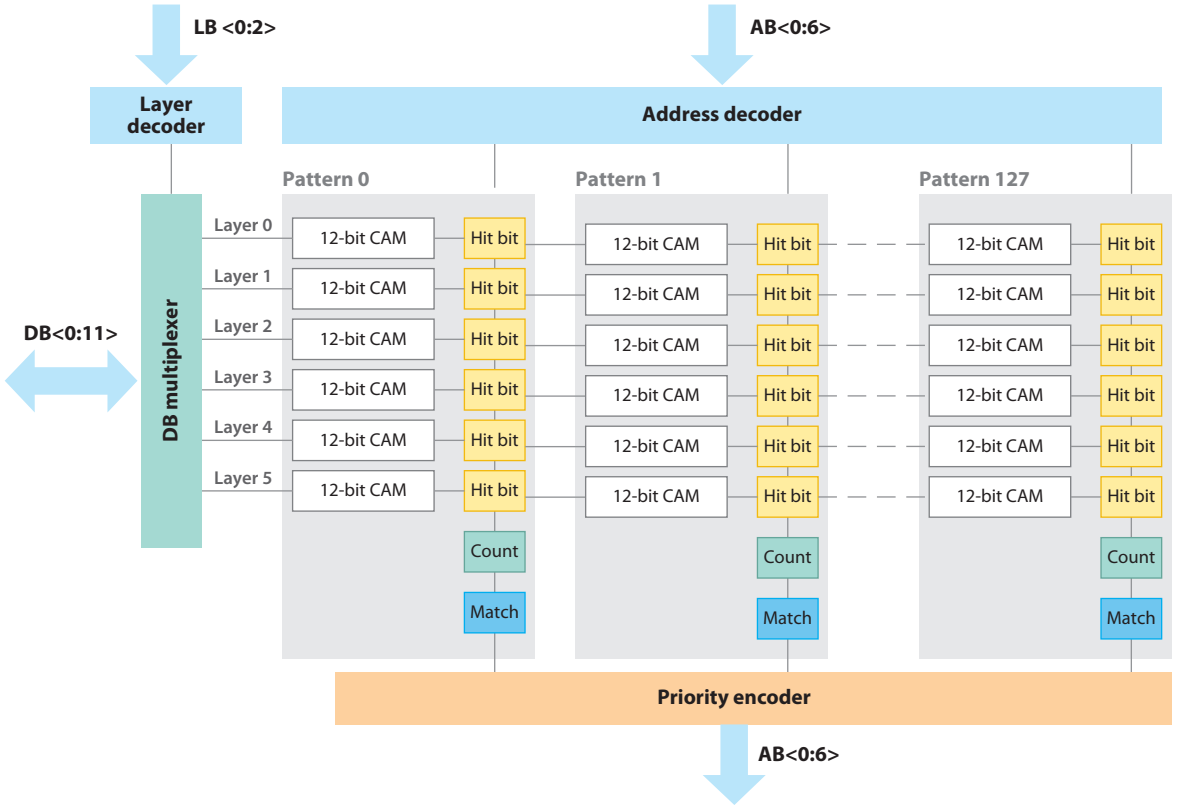


Figure 5

The internal structure of the associative memory device. Abbreviations: AB, address bus; DB, data bus; LB, layer bus.

It is often desirable to recognize patterns that have only a partial match with data so that we can find a track even when one or more hits have been lost due to detector inefficiencies. In most tracking detectors, allowing for some missing hits along a track still ensures sufficient rejection against random combinations and also increases the efficiency of finding all the tracks. For this reason, the associative memory device was designed with the option of having the match bit set when the hit count reaches a programmable threshold, which can be set to any number from one to six. If the threshold is set to six, only the patterns matching on all six detector layers are output; if the threshold is set to five, all the patterns matching on five or six detector layers are output. At CDF, requiring at least five hits out of six yielded sufficient rejection against background and a significantly larger track-finding efficiency than did requiring six out of six.

3.3. Generating the Pattern Bank

We now turn to the problem of constructing the pattern bank to be loaded in the associative memory device. The most direct and conceptually simple strategy is to use a Monte Carlo simulation method. Doing so involves generating tracks that are randomly extracted from a distribution in parameter space, similar to what we would expect in real experimental conditions, by simulating the trajectory of each track through the detector layers to determine the coordinate of the crossing point in each layer. All possible different hit patterns generated by simulated tracks are stored

and used to fill the pattern bank. Unfortunately, in practical cases, the size of the pattern bank thereby created may be unmanageably large. The size of the pattern bank grows with a power of the number of segments in each detector layer. The power is typically the number of degrees of freedom of the track-parameter space. In the simple case of straight tracks (**Figure 2**), the power is two, and in the case of nonnegligible curvature due to a magnetic field, it becomes three. For a silicon detector a few centimeters wide and with a spatial resolution of order 10 μm, the number of segments would be of order of a few thousand and would require millions or even billions of patterns to be stored in the associative memory. These numbers are entirely unrealistic, considering that the maximum density achieved in the VLSI implementation of the associative memory device for the CDF experiment was 128 patterns on a 1-cm^2 silicon chip (20). The solution to this problem is to reduce the number of segments into which we subdivide the detector by increasing the segment size. This is done by grouping a number of strips in a silicon detector into a single logic segment and treating them, as far as the associative memory device is concerned, as a single detector element.

Using segment sizes ranging from 125 to 250 μm, CDF created a pattern bank of the order of 10^5, which was implemented through the use of the custom VLSI technology available in the 1990s. The increase in segment size, however, caused a significant loss of resolution, which had to be recovered. In the following section, we explain how to use the output of the associative memory to recover full detector resolution and extract the parameters of each track with high precision.

4. LINEARIZED TRACK FINDING AND FITTING

To find secondary vertices, we must extract the parameters of each track given the position of the hits. In particular, we need to measure the impact parameter of each track with a precision of a few tens of micrometers. In off-line analyses, this task is performed by fitting the analytical expression of the trajectory of a particle to the position of the hits in the detector, using a minimization algorithm that typically works by successive approximations. For our purposes, such an approach is not an option, given the time constraints imposed by the trigger requirements.

The solution to this problem was to substitute all the functions involved in the calculations with linear approximations, thereby allowing a much simpler minimization that could be performed analytically. For each track parameter, this procedure yielded a simple scalar product of the hit coordinates with an array of coefficients that depended only on the geometry of the detector and which could be computed in advance once and for all. In fact, the idea of separating pattern recognition from the precision measurement of the track parameters is one of the key points that led to the success of CDF's silicon vertex trigger (SVT) because it allows both for easier pattern recognition with coarser resolution and easier determination of track parameters, which become limited to the much smaller space within the boundaries of a single pattern.

4.1. Linearizing the Constraints

Each candidate track is represented as a list of n hit coordinates, one per detector layer, and can be thought of as a point $\boldsymbol{x}$ within a set $\mathcal{C} \subset R^n$, limited in extension by the physical boundaries on the hit coordinates. Clearly, not every point in $\mathcal{C}$ is equally likely to represent a track: Only those with coordinates aligned along a track path are reasonable candidates. The set of all acceptable candidates is therefore some subset $\mathcal{T} \subset \mathcal{C}$, to be precisely defined by some criterion (such as a χ^2 cut) that determines the efficiency of the set, which is defined as the probability that a real track crossing the detector produces a candidate belonging to $\mathcal{T}$. The process of deciding whether a given candidate track belongs to the subset $\mathcal{T}$ is termed track finding.

Suppose each track can be labeled by a number $m < n$ of real parameters $\boldsymbol{p}$ (e.g., P_T, ϕ_0, and d). In the ideal case of perfect resolution, the hit coordinates are uniquely determined by the parameters $\boldsymbol{p}$, and the set $\mathcal{T}$ reduces to an m-dimensional surface contained in $\mathcal{C}$, described by n parametric equations,

$$\boldsymbol{x} = \boldsymbol{x}(p_1, \cdots p_m), \qquad 1.$$

which can also be cast in implicit form by eliminating the parameters. Doing so yields $n - m$ constraint equations,

$$f_i(\boldsymbol{x}) = 0; \quad i = 1, \cdots n - m. \qquad 2.$$

The constraint functions $\{f_i\}$ can be determined, either analytically or numerically, from our knowledge of geometry. These functions formally solve the problem of track finding: For each candidate track, one evaluates the $f_i(\boldsymbol{x})$ and accepts the track if and only if all f_is are zero.

This holds exactly only for infinite resolution. The effect of finite resolution is to make the f_is random variables that take values slightly different from zero, thereby adding some finite thickness to the m-dimensional surface $\mathcal{T}$. The measure of this effect is given by the covariance matrix F_{ij} of the f_is, which can be calculated at first order from the covariance matrix $\boldsymbol{S}$ of the coordinates $\boldsymbol{x}$:

$$F_{ij} \simeq \sum_{k\ell} \frac{\partial f_i}{\partial x_k} \frac{\partial f_j}{\partial x_\ell} S_{k\ell}. \qquad 3.$$

The covariance matrix, being symmetric, can be made diagonal through a rotation in the space of the f_is. Furthermore, the new coordinates $\tilde{f}_i$ can be scaled to unit variance so that a χ^2 forms as $\chi^2 = \sum_i \tilde{f}_i^2$, where $\langle \tilde{f}_i^2 \rangle = 1$.

The values of the constraint functions $\tilde{f}_i(\boldsymbol{x})$ must be computed for each candidate track. Generally, they are complicated functions, so we approximate them with linear functions by expanding about some convenient point $\boldsymbol{x_0}$:

$$\tilde{f}_i \simeq \frac{\partial \tilde{f}_i}{\partial \boldsymbol{x}} \cdot (\boldsymbol{x} - \boldsymbol{x_0}) = \boldsymbol{v_i} \cdot \boldsymbol{x} + c_i. \qquad 4.$$

Geometrically, this amounts to approximating $\mathcal{T}$ with its tangent hyperplane in $\boldsymbol{x_0}$, and the $\boldsymbol{v_i}$s are the vectors orthogonal to the hypersurface in $\boldsymbol{x_0}$. Clearly, the approximation works well when $\mathcal{T}$ is nearly flat. In general, to obtain a sufficient precision, it is necessary to segment $\mathcal{T}$ into several smaller regions and perform the expansion around the central point $\boldsymbol{x_0}$ of each of them.

All constants can be calculated analytically from our knowledge of the detector geometry, but there is a numerical method that is simpler and more intuitive and that also allows easy handling of misalignments. This technique is known as principal components analysis (24).

This method begins from a sample of vectors $\boldsymbol{x}$ belonging to $\mathcal{T}$ generated either by Monte Carlo or from real data. The covariance matrix $\boldsymbol{M}$ of $\boldsymbol{x}$ can be estimated as

$$M_{ij} \simeq \langle (x_i - \langle x_i \rangle)(x_j - \langle x_j \rangle) \rangle. \qquad 5.$$

If we diagonalize $\boldsymbol{M}$, we generally find (*a*) $n - m$ eigenvectors with small eigenvalues, which form a base in the linear subspace orthogonal to the surface, and (*b*) m eigenvectors with larger eigenvalues, which form a base in the linear subspace tangent to the surface. The small eigenvalues correspond to the thickness of the surface caused by finite detector resolution, whereas the large eigenvalues correspond to the variance of the coordinates caused by spanning the track-parameter space.

We can now take the coordinates with respect to the eigenvectors orthogonal to the surface, scale them to unit variance, and identify them with the $\tilde{f}_i$s. This procedure is much more convenient than any analytical calculation because it is simpler and can be performed directly on samples

of real data, thereby automatically incorporating corrections for possible detector misalignments that are already implicitly contained in $\boldsymbol{M}$.

4.2. Linearized Parameter Estimation

Now we come to the problem of estimating the parameters of the tracks we have found. We can invert (at least locally) Equation 1 to obtain the parameters as a function of the hit coordinates:

$$p_i = p_i(\boldsymbol{x}). \qquad 6.$$

By applying the same reasoning that we applied to the constraints in the previous section, we can use a linear approximation of Equation 6:

$$p_i \simeq \boldsymbol{w}_i \cdot (\boldsymbol{x} - \boldsymbol{x_0}) + p_i(\boldsymbol{x_0}) = \boldsymbol{w}_i \cdot \boldsymbol{x} + q_i. \qquad 7.$$

The optimal values of the constants $(\boldsymbol{w}_i, q_i)$ can be found simply by optimizing the resolution, that is, by minimizing the average quadratic deviation from the true values $\tilde{p}_i$:

$$\frac{\partial \langle [\tilde{p}_i - p_i(\boldsymbol{x})]^2 \rangle}{\partial \boldsymbol{w}_i} = 0. \qquad 8.$$

These constants can be readily obtained from a sample of tracks with known parameters. This is an important difference with respect to the case of the constraints discussed above, where we could obtain the constants from a sample of real tracks. In this case, we can use only Monte Carlo, and we are forced to rely on the accuracy of the detector geometry used in the simulation. We can easily determine the q_is from real data by using our knowledge of the average values $\langle \tilde{p}_i \rangle$ of each parameter for the track sample.

4.3. Indexing Parameters with Road Number

The strategy for track finding and fitting outlined above is based on the assumption that the linear approximations we are using are good enough to ensure that the performance in terms of background rejection and parameter resolution is not significantly degraded. Clearly, this strategy can be beautifully matched to the pattern recognition performed by the associative memory device (Section 3). Each track candidate found by the associative memory device is typically bound within a very small volume of the track-parameter space, and hit-coordinates range in a very small interval. Therefore, we can calculate in advance a different set of constants for the linear approximation for each road and index them with the road number, that is, the address in associative memory at which the corresponding pattern is stored. Given a track candidate, by using the road number we can immediately retrieve the constants relative to that road and calculate the χ^2 and, possibly, the track parameters. In practical cases, the associative memory device parameter-space segmentation may be much finer than needed for the linear approximations to be good, and the same set of constants may be used for multiple road numbers. For the CDF implementation, nearly 400,000 roads were needed in the associative memory device, but only 72 different sets of geometrical coefficients were sufficient to cover the whole detector.

5. THE SILICON VERTEX TRIGGER

The principles discussed in Section 4 are very general and could be applied to different situations and different experimental conditions. Below, we describe in some detail how these principles were applied to the design of the SVT for the CDF experiment (25–31).

5.1. The Tracking System of CDF

A more detailed description of the CDF upgraded detector can be found in References 32–34. The trigger system uses information from the silicon vertex detector and the central drift chamber, known respectively as SVXII (35) and COT (36). All these components are immersed in a longitudinal magnetic field produced by a superconducting solenoid.

Five layers of SVXII are used for triggering. Their distances from the beam range from 2.5 to 10.6 cm. Each layer of the silicon detector is subdivided into 12 ϕ sectors and six segments along the beam direction (i.e., the z coordinate). The ensemble of the five layers at the same ϕ and same z is termed a wedge. There are a total of $12 \times 6 = 72$ wedges in SVXII. For each charged track produced in the η interval between -1 and $+1$, the COT provides approximately 90 points, distributed more or less uniformly along the trajectory and ranging from 40 to 140 cm away from the beam. The signals from the COT are processed by the extremely fast tracker (XFT) trigger processor (37), which finds tracks and computes their ϕ and P_T in a time of approximately 5 μs. The precision of the XFT is far from sufficient to detect the impact parameters of displaced tracks from heavy-flavor decays, so the tracks found by the XFT are fed to the SVT, where they are combined with information from the silicon detector, first to measure impact parameters with a precision of a few tens of micrometers and second to make them available in time for a trigger decision (~20 μs).

5.2. Silicon Vertex Trigger Architecture

The SVT uses five concentric SVXII layers (0–4), with strips parallel to the beam (z coordinate), and two track parameters as calculated by the XFT (ϕ and P_T). We assign five layers of the associative memory to the five silicon layers. Hit coordinates are segmented into 250-μm bins. The two track parameters output by the XFT (ϕ and P_T) are mapped to one parameter ϕ_0 (angle ϕ at the origin of the track), segmented into 20-mrad bins, and assigned to the sixth layer in the associative memory device. Later, the two original parameters of each track are used in the fit, together with full-resolution hit coordinates from the five silicon layers.

For each event, the SVT reads out data from the SVXII, combines this information with information from the XFT, finds all tracks with $P_T > 2$ GeV/c, and reconstructs them with an impact-parameter accuracy of the order of 30 μm. All these steps must be performed within approximately 20 μs. **Figure 6** shows the SVT's main physical components and their interconnections. The SVT consists of more than 100 large electronic boards of 10 different types.

The SVT is implemented as a data-driven architecture. Data communication between functional blocks is obtained through point-to-point high-speed links. Various tasks are divided among functional blocks. Each functional block begins working on data as soon as it obtains them and outputs results as soon as they are ready: There is no attempt to synchronize the operation of different blocks by any other means than the flow of data. The communication protocol has been designed for maximum simplicity and speed, and it is uniform throughout the system.

Each SVXII wedge is read out through an independent data path. Sparsified[1] data from each readout line are fed into one hit finder, whose task is to find pulse-height clusters and compute the coordinate of the centroid of each cluster. There are a total of 72 hit finders, one per SVXII wedge. Hit coordinates from the six hit finders from the six wedges at the same ϕ and tracks from XFT (the COT track processor) are merged into a single data stream and fed to both the associative memory device and the hit buffer. The task of the associative memory device is to perform the

[1] Sparsification is the action of eliminating from the output data stream all the channels containing no relevant information (empty channels).

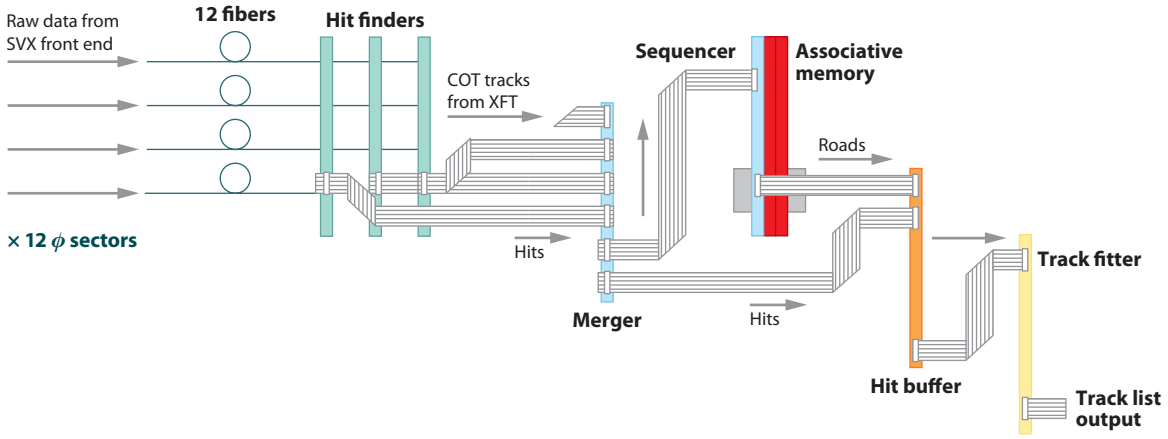


Figure 6

The physical components of the silicon vertex trigger and how they are interconnected. Abbreviations: COT, central drift chamber; SVX, silicon vertex detector; XFT, extremely fast tracker.

first stage of the pattern-recognition process using 250-μm bins on SVXII layers. These coarse-resolution track candidates are termed roads. Roads are transmitted from the associative memory device to the hit buffer, which associates each road with the proper hit coordinates and XFT track and transmits them to the track fitters. Each track fitter receives one or more roads, each with the corresponding full-resolution hits from the hit buffer, and reconstructs one or more tracks within each road by using linearized constraints and linearized parameter estimation (discussed above). Each processor outputs the parameters (P_T, ϕ, d, and z) of every track it finds. All the data from the 12 track fitters are merged into a single track list that is output by SVT and used by the overall trigger logic to build different physics triggers.

5.3. The Beam Spot

The SVT approach to tagging nonprimary tracks is to look at the impact parameter relative to the center of the beam spot. We abandon any attempt to find the position of the primary vertex of the event because (*a*) finding the primary collision vertex is difficult and (*b*) the results are often poor for *B* events, in which the track multiplicity is low. Moreover, at the typical luminosity for Run II, we have multiple primary vertices in a single event. The Tevatron beam spot has a transverse sigma of ∼30 μm, and for *B* physics we are interested in tracks with an impact parameter of at least ≈100 μm. Therefore, use of the beam spot center instead of the primary vertex results in only a modest degradation in performance.

Because SVT measures only transverse coordinates, it observes only the transverse projection of the luminous region of the Tevatron. For this reason, we need to ensure that the detector axis is parallel to the beam; otherwise, the beam spot will appear to SVT as an elongated region, impairing its ability to recognize secondary tracks. Also, the need for measuring impact parameters with such a high precision imposes very strict requirements on the mechanical tolerances of the detector. The entire mechanical structure of the SVXII detector was built so that, after installation, all detector elements would be at their nominal relative positions within 20 μm (35).

For trigger purposes, therefore, we need to measure the impact parameter of all tracks with respect to the center of the beam spot projected on the plane transverse to the beam. To do so,

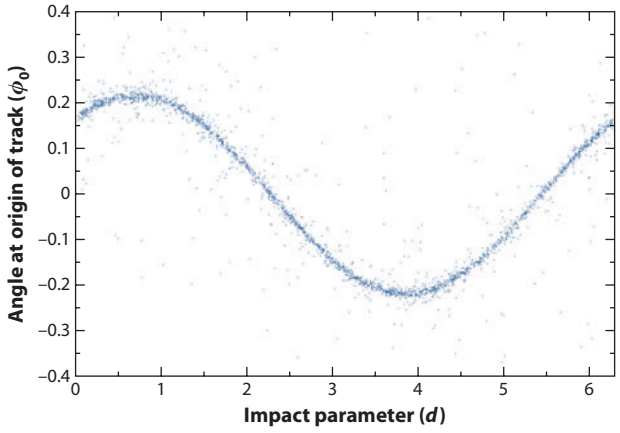


Figure 7

A typical d-ϕ_0 plot. The sine wave shape is created by tracks originating from a beam spot that is slightly displaced from the center of the coordinate system.

we need to know the exact position of the beam at the interaction point inside the CDF detector. This position is not constant in time; rather, it changes slightly whenever protons and antiprotons are injected into the machine. The position also drifts slowly during data taking. These changes can be of the order of hundreds of micrometers, and they cannot be neglected without affecting the performance of SVT. It follows that we need to measure the position of the beam spot in real time and use it to compute the impact parameters of all the tracks. The SVT strategy is to have two independent processes. One method finds all tracks in each event and computes their impact parameter with respect to the center of the SVX detector, which is assumed to be the origin of the coordinate system. This process is performed as described above. Another method takes all the tracks found by the SVT in all events and plots their impact parameter d versus ϕ_0. A typical d-ϕ_0 plot is shown in **Figure 7**. If the beam spot were located exactly in the center of the coordinate system, all points would lie on the horizontal axis because all the tracks would have zero impact parameter. If the beam spot were slightly displaced from the center, then a nonzero impact parameter would be measured with respect to the origin of the coordinate system, and a sine wave shape would be obtained as a function of ϕ_0:

$$d = x_0 \sin(\phi_0) - y_0 \cos(\phi_0),$$

where x_0 and y_0 are the coordinates of the beam spot that can be obtained by fitting the above expression to the d-ϕ_0 plot.

Given the very high rate at which the SVT processes events, statistics of several hundred thousand tracks can be collected in a matter of seconds, and a very precise determination of the position of the beam can be obtained and used to correct the impact parameter of all the tracks in real time. **Figure 8** shows the distribution of the impact parameters with respect to the actual position of the beam spot of a sample of tracks reconstructed by SVT.

5.4. Trigger Paths

The track list output by SVT is used by the CDF trigger processors to look for decays of long-lived particles. The selection criteria are typically based on the requirement of at least two tracks in the same event with a transverse momentum greater than 2 GeV/c and an impact parameter

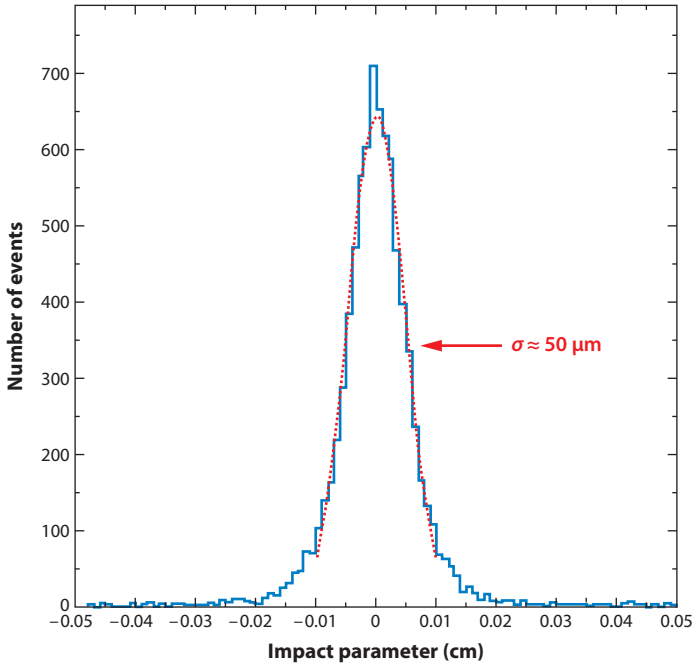


Figure 8

The distribution of the impact parameters of a sample of tracks as reconstructed by the silicon vertex trigger (SVT). A Gaussian with $\sigma \approx 50$ μm fits the core of the distribution very well. The width of the distribution is interpreted as the convolution of the actual transverse size of the beam spot (≈25 μm) with the impact-parameter resolution of the SVT (≈43 μm).

greater than 100 μm, but the exact value of these cuts is optimized as a function of instantaneous luminosity and can range up to 2.5 GeV/c and 120 μm, respectively. A number of distinct trigger paths, with slightly different cuts, are provided in parallel and are enabled, disabled, or prescaled[2] according to the evolution of the instantaneous luminosity in the duration of a Tevatron store. In this way, we strive to optimize the number of signal events that are permanently recorded and available for physics analyses (38).

6. PHYSICS RESULTS

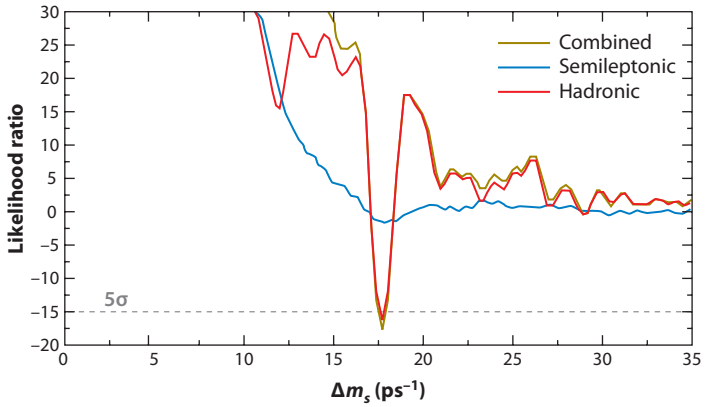
The SVT was designed and built by CDF between 1990 and 2000. It was installed in the CDF control room in the summer of 2000 and began operation in October of the same year. The first publication of the Tevatron Run II results (a measurement of the mass of the D_s) was based on data collected by the SVT (39).

The SVT has been operating without interruption since 2000, and it has allowed CDF to perform important measurements that would otherwise have been impossible. Examples of such measurements are those of the B_s oscillation frequency (40) and the direct CP asymmetry in the charmless decay of B_d, B_s, and Λ_b (41–43). The SVT has also provided CDF with the world's largest sample of D^0 mesons, paving the way to precision measurements of D^0 oscillations and

[2] Prescaling is the action of sampling only a fraction of the events selected by a specific trigger path.

Figure 9

The likelihood plot for the B_s oscillation frequency, as obtained by CDF (40). The contribution of hadronic decays, collected by the silicon vertex trigger, is shown in red, and the contribution from lepton triggers is shown in blue.

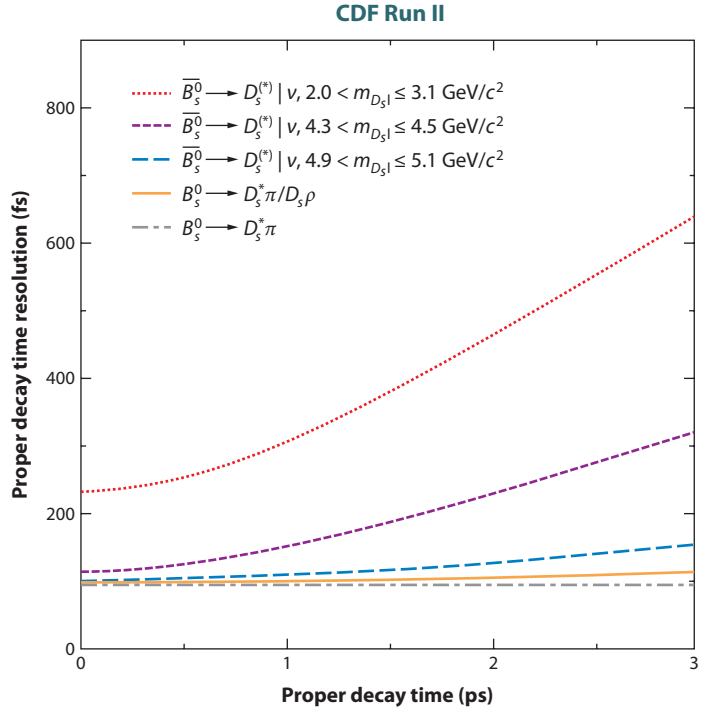


direct *CP* violation (44). The range of possible measurements using the SVT data sample is still being explored.

The measurement of B_s oscillations is a good example of the importance of triggering on displaced vertices rather than on the presence of leptons. The proper time of the B_s decay is obtained from the measurement of its momentum and the path length traveled before decay, but in all semileptonic decays a neutrino in the final state escapes detection and precludes a precise momentum measurement, which in turn severely affects the precision of the determination of the proper time. **Figure 9** shows the likelihood plot for the B_s oscillation frequency as obtained by CDF (40). **Figure 10** shows the proper time resolution for different decay modes of the B_s.

Figure 10

The proper time resolution as a function of proper time, as obtained by CDF for the measurement of the B_s oscillation frequency (40). Plots for different decay modes of the B_s are shown.



The advantage of hadronic decay modes over semileptonic decay modes is evident. Without the SVT, we would still be waiting for an observation of the B_s oscillations.

7. CONCLUSIONS

The performance of SVT at CDF shows that by using a sophisticated trigger to measure impact parameters in real time, hadron colliders can achieve results in the field of c and b physics that were difficult to imagine in the past and that are competitive with, or in some cases superior to, results obtained at dedicated e^+e^- colliders.

The experience of designing, building, running, and doing physics with the SVT at CDF has generated ideas about how to use the same principles for the next generation of hadron colliders, namely the LHC. Although nothing similar to the SVT has been included in the baseline design of current LHC detectors, ATLAS is now considering the addition of a hardware-based fast track finder (13, 14) as part of its upgrades.

The complexity of track-finding algorithms, which require a great deal of computer power, will be an important factor in determining the quality of the physics results from the LHC and other future high-energy physics experiments. Innovative techniques to efficiently solve pattern-recognition and track-fitting problems on the basis of the latest electronic technologies could significantly increase these experiments' physics reach, and they certainly represent an important and fascinating challenge for future generations of experimental particle physicists.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

LITERATURE CITED

1. Weinberg S. *Phys. Rev. Lett.* 19:1264 (1967)
2. Salam A. In *Elementary Particle Theory*, ed. N Svartholm, pp. 367–77. Stockholm: Almquist & Wiksells (1969)
3. Glashow SL, Iliopoulos J, Maiani L. *Phys. Rev. D* 2:1285 (1970)
4. Kobayashi M, Maskawa T. *Prog. Theor. Phys.* 49:652 (1973)
5. Cabibbo N. *Phys. Rev. Lett.* 10:531 (1963)
6. Amsler C, et al. (Part. Data Group) *Phys. Lett.* B667:1 (2008)
7. Hurst RB, et al. (WA82 Collab.) *Nucl. Phys. Proc. Suppl.* 16:302 (1990)
8. Kaplan DM. (BTeV Collab.) *Proc. APS/DPF/DPB Summer Study Future Part. Phys.* 1:E704 (2001)
9. Christian CD, et al. *Nucl. Instrum. Methods A* 473:152 (2001)
10. Zimmermann S, et al. *Nucl. Instrum. Methods A* 465:224 (2000)
11. Caron S. (DØ Collab.) *Nucl. Phys. Proc. Suppl.* 156:143 (2006)
12. Augusto Alves A, et al. (LHCb Collab.) *J. Instrum.* 3:S08005 (2008)
13. Annovi A, et al. *IEEE Trans. Nucl. Sci.* 48:575 (2001)
14. Brubaker E, et al. *IEEE Trans. Nucl. Sci.* 55:145 (2008)
15. Mead C, Conway L. In *Introduction to VLSI Systems*, p. v. New York: Addison-Wesley (1980)
16. Dell'Orso M, Ristori L. *Nucl. Instrum. Methods A* 278:436 (1989)
17. Weste NHE, Eshraghian K. In *Principles of CMOS VLSI Design: A System Perspective*, pp. 564–80. New York: Addison-Wesley. 2nd ed. (1993)
18. Taub H, Schilling D. In *Digital Integrated Electronics*, pp. 420–31. New York: McGraw-Hill (1977)
19. Millman J, Grabel A. In *Microelectronics*, pp. 352–66. New York: McGraw-Hill. 2nd ed. (1987)
20. Morsani F, et al. *Nucl. Instrum. Methods A* 315:446 (1992)

21. Weste NHE, Eshraghian K. In *Principles of CMOS VLSI Design: A System Perspective*, pp. 589–90. New York: Addison-Wesley. 2nd ed. (1993)
22. Pagiamtzis K, Sheikholeslami A. *IEEE J. Solid-State Circuits* 41:712 (2006)
23. Millman J, Grabel A. In *Microelectronics*, pp. 286–91. New York: McGraw-Hill. 2nd ed. (1987)
24. Pearson K. *Philos. Mag.* 2:559 (1901); Jolliffe IT. *Principal Component Analysis*. New York: Springer. 2nd ed. (2002)
25. Bardi A, et al. *Nucl. Instrum. Methods A* 409:658 (1998)
26. Ashmanskas W, et al. *Nucl. Instrum. Methods A* 447:218 (2000)
27. Ashmanskas W, et al. *Nucl. Instrum. Methods A* 477:451 (2002)
28. Bardi A, et al. *Nucl. Instrum. Methods A* 485:178 (2002)
29. Ashmanskas W, et al. *Nucl. Instrum. Methods A* 501:201 (2003)
30. Ashmanskas W, et al. *Nucl. Instrum. Methods A* 518:532 (2004)
31. Adelman J, et al. *Nucl. Instrum. Methods A* 569:111 (2006)
32. Acosta D, et al. *Phys. Rev. D* 71:032001 (2005)
33. Acosta D, et al. *Phys. Rev. D* 71:052003 (2005)
34. Abulencia A, et al. *J. Phys. G* 34:2457 (2007)
35. Sill A, et al. *Nucl. Instrum. Methods A* 447:1 (2000)
36. Affolder T, et al. *Nucl. Instrum. Methods A* 526:249 (2004)
37. Thomson EJ, et al. *IEEE Trans. Nucl. Sci.* 49:1063 (2002)
38. Anikeev K, et al. arXiv:hep-ph/0201071 (2002)
39. Acosta DE, et al. *Phys. Rev. D* 68:072004 (2003)
40. Abulencia A, et al. *Phys. Rev. Lett.* 97:242003 (2006)
41. Abulencia A, et al. *Phys. Rev. Lett.* 97:211802 (2006)
42. Aaltonen T, et al. *Phys. Rev. Lett.* 103:031801 (2009)
43. Morello M, et al. *Nucl. Phys. Proc. Suppl.* 170:39 (2007)
44. Aaltonen T, et al. (CDF Collab.) *Phys. Rev. Lett.* 100:121802 (2008)

Annual Review of Nuclear and Particle Science

Volume 60, 2010

Contents

Indexes

Errata

An online log of corrections to *Annual Review of Nuclear and Particle Science* articles may be found at http://nucl.annualreviews.org/errata.shtml