FERMILAB-Conf-03/059-E April 2003

Fermilab

Triggering and Data Acquisition
General Considerations

Joel N. Butler
Fermi National Accelerator Laboratory
Batavia, IL, US

Abstract. We provide a general introduction to trigger and data acquisition systems in High Energy Physics. We emphasize the new possibilities and new approaches that have been made possible by developments in computer technology and networking.

The goal of this series of two lectures is to introduce the topics of triggering and data acquisition in high-energy physics (HEP) to students who are interested in experimental particle physics and related fields. One theme of the lectures is that technology has made this an area of rapid change. I therefore emphasize the basic principles of trigger and data acquisition systems (DAQ), their general characteristics, and the key considerations that drive their design. I discuss how these systems have evolved over time to take advantage of the newest technologies available. I give concrete illustrations of the principles and how they are implemented using today's technology. I stress that tomorrow's technologies may reshape the way such systems are implemented and encourage people to think about how expected advances might affect these systems. Finally, I note that an "abstraction" of the typical modern HEP trigger and DAQ system might have applications to many other areas where massive amounts of data must be processed in quasi-real time.

The outline of the lectures is as follows. Section 1 presents some basic facts about high energy physics experiments that will be needed for the rest of the lectures; Section 2 introduces the general structure of a modern trigger and data acquisition system. It then reviews the main considerations in designing Trigger/DAQ systems, including the rationale for having triggers at all, the reasons for a "trigger hierarchy", and many key concepts and terms; section 3 discusses a selected issues in the implementation of modern data acquisition systems, especially event building; section 4 discusses many forgotten aspects of the triggering and DAQ "environment" – such topics as hardware management, run control, commissioning, fault tolerance, and the relation of online to offline analysis; section 5 presents an abstract view of a modern trigger system and connects it with offline systems in HEP and quasi-real-time applications outside of HEP; section 6 provides some brief concluding remarks on why this area is an exciting one for young physicists to work in.

1. Introduction

The data that are processed by trigger and DAQ systems come from high energy interactions. These are referred to as "events," usually the result of collisions between two elementary particles, or decays of an elementary particle.

1.1 Types of experiments

There are a few general types of HEP experiments:

- **Colliding beam experiments:** A high energy beam of particles coming from one direction collides and interacts with a second beam of particles coming from the opposite direction. This arrangement makes the most violent collisions. The properties of the interactions or the properties of the particles created, including their decays, may be the object of the experiment.
- **Fixed target experiments:** A high energy beam of particles impinges on a solid target and interacts with the nuclei of the target, which are essentially at rest. Again, the properties of the interactions or the properties of the particles created, including their decays, may be the object of the experiment.
- **Decay experiments:** A beam of unstable particles usually made in fixed target collisions, decays and the decays themselves are studied.
- There are also **non-accelerator experiments** that use similar techniques.

Each type of experimental detector has its distinctive characteristics and is basically a custom designed device to answer a particular physics question.

1.2 Types of events

Most high energy physics events are "ordinary" -- that is, well understood or, at least, familiar. "Ordinary" events are also referred to as "Minimum Bias" events, for reasons that will become clear below. Usually, one is looking for "rare events' that signify some new phenomena, perhaps a new particle.

1.3 Luminosity

Luminosity, $\mathscr{L}$, is a measure of the ability of a beam configuration to produce collisions. It captures the information about the number of incoming particles per second, the number of target particles per unit area, and various geometric factors such as the beam size. The resulting number has units of $1/\text{cm}^2$-s and is such that when multiplied by the cross section for a process, gives the number of interactions for that process occurring each second. Note that $\mathscr{L}$ does not depend on the specific particle interaction. As an example, a luminosity of 10^{33} $/\text{cm}^2$-s at the Tevatron gives about $7\text{x}10^6$ interactions per second. For colliding beams, the "target density" is from the second or target beam. For fixed target, it is given by the density of the solid, liquid, or gas in the target.

1.4 Beam Crossings

In colliders, the beams are often bunched – that is they come in intense clusters of particles followed by intervals empty of beam. The collisions take place periodically when a bunch from one beam crosses the other beam (or target) at a "bunch crossing". The experiment is usually located near the position of a bunch crossing. The "bunch spacing" is the time interval between bunches, usually measured in nanoseconds and the inverse of this time is the bunch crossing frequency. As an example, at the LHC the bunch spacing is 25 ns, corresponding to a bunch crossing frequency of 40MHz.

2. The General Structure of a Modern Trigger and Data Acquisition System

For one reason or another, which we discuss below, it is usually impossible to record every event in an experiment on archival media so that analysis can take place “offline”. In HEP, a “trigger system” is the collection of hardware and associated software used to select (YES/NO) which events are to be recorded to an archival storage medium by the DAQ, another collection of hardware and software, for subsequent offline physics analysis. Events that the trigger rejects are discarded -- that is, ***lost forever.*** The structure of a modern[1] trigger and DAQ system is shown in Fig. 1. The purpose of this section is to explain the figure.

2.1 Detector, Front End Electronics and Digitization

The experiment begins with a “detector,” which has devices that respond to the passage of particles produced in the collisions or decays of high-energy particles. The individual “sensors” or transducers produce electrical signals that characterize the properties of the particles. The electrical signals may be further amplified and shaped and are then digitized, and stored for subsequent readout. Typical examples of “sensors” are given in Table 1. Many of these are discussed in detail in other talks at this workshop.

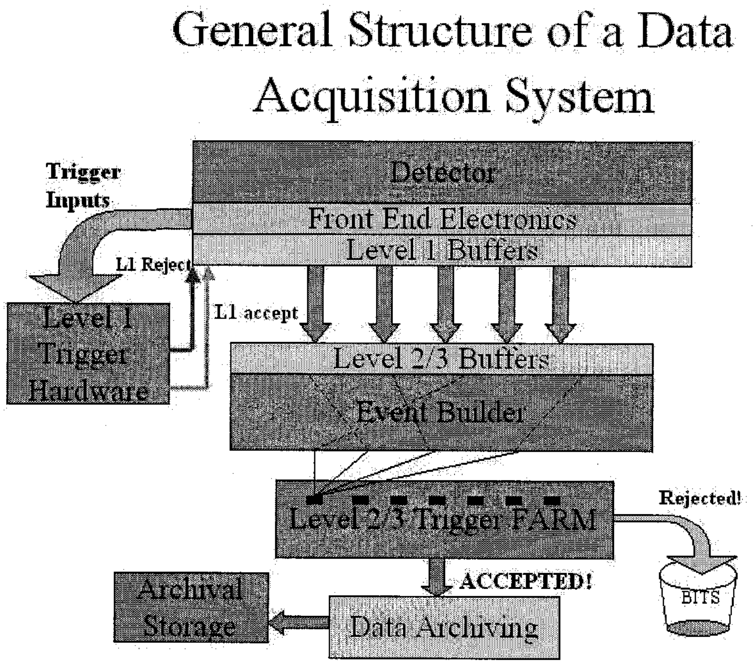


Figure 1: General structure of a Trigger and Data Acquisition system (DAQ)

Front end electronics amplifies, shapes, and conditions the raw electronic signals produced by sensors so that they can be subsequently digitized. The distinction between “front end” electronics and “data acquisition” electronics is somewhat arbitrary and may differ between experiments. The “digitization” of the data may be done in the the front end electronics. The front end may also “sparsify” the data. By sparsify we mean to eliminate signals that are below a threshold that defines the boundary between the lower range of a signal and electronic noise. It may also store the data in a “memory buffer”, “pipeline”, or “FIFO” to await a “trigger decision.” The “front end” usually has components on the detector or in electronics crates near the detector. It is usually "downstairs" or in the enclosure that houses the actual experimental apparatus. Signal drivers that send the signals to a "control room" or "electronics barracks" or the cables or optical links that carry the signals may mark the beginning of the DAQ. In other experiments, the end of the cables or receivers "upstairs" may define the beginning of the DAQ. The exact boundaries are a matter of convention for each experiment and are somewhat arbitrary.

Tracking Detectors (localize the particle track to a small region of space): • Silicon Microstrip Detectors, • Silicon Pixel Detectors; • Microstrip Gas Detectors; • Proportional Wire Chambers; • Drift Chambers; and • Scintillating Fiber Trackers	**Particle Identification (hadrons)** (Signal depends on type of particle): • Cerenkov detectors (ring imaging; liquid or solid (DIRC), threshold); • Transition Radiation detector; • DE/dX; and • Time of flight
Calorimetry (signal proportional to energy deposited): • Crystal calorimeters (CsI, PbO); • Shashlik calorimeter; • Plates with active media such as scintillating tiles, liquid argon, liquid krypton interspersed	**Muon Detection** (signal occurs only if particle penetrates a massive shield that absorbs hadrons, photons, and electrons): • Proportional Tubes; • Cathode strip chambers; • Resistive pad chambers; and • Scintillator hodoscopes.

Table 1: A list of typical sensors used in particle physics experiments

While there are many kinds of detectors and experiments, there are only a few general types of "digitized data". These include: **Latch bit** – a binary datum that indicates a HIT/NOT HIT or ON/OFF condition for a given sensor channel. The actual data may be the address of the "hit" channel or a map of all the channels indicating which had hits and which did not. To decide whether there is a hit, the electronics applies a threshold on the sensor's signal to suppress electronic noise; **ADC** (Analog-to-Digital Conversion) – A digitization of the pulse height or pulse area (after signal integration). Data consists of the channel address and the numeric value (characterized by some number of bits of range and of precision) of the digitized signal; **TDC** (Time-to-Digital Conversion) – The time of arrival of a signal relative to some time reference. Data consists of the channel address and the numeric value of the digitized time; and **Counter** (a scaler count) -- A channel address and the number of counts recorded. In some environments, there is a **"time stamp"** or **"bunch crossing number"** (BCO) that is used to associate various data fragments from the same event or collision with one another.

2.2 Trigger Systems

A trigger is a FILTER – it selects some high energy data, in the form of records of individual collisions, to save to permanent storage for subsequent analysis and consigns all the rest of the events to oblivion. Stated simply, a trigger system should keep all the events you need to do your physics analysis – signal events plus any events needed for calibration or study of systematics, and throws out the rest. **More realistically, the trigger should have a high efficiency for keeping the events of interest and a very low efficiency for other, unwanted events.**

Trigger systems have the following features:

- They must run in quasi-real time – trigger decisions must keep up with the rate of data produced by interactions occurring in the experiment. The BTeV trigger, which is discussed below, must make a decision every 132 ns, while the CMS and ATLAS triggers, also discussed below, must make a decision every 25 ns on the average.
- Triggers are "mission critical". A defective trigger can throw out the "good events", as well as, or instead of the "uninteresting events."
- Trigger performance and behavior must be well-understood. A malfunctioning or poorly designed trigger can create "selection biases" that make it very hard to extract physics information from the recorded event sample.
- We always worry that the trigger can be "inherently physics biased." If you set up a trigger to be efficient at selecting a class of events that you are looking for, how can one ever find the unexpected.

You would not use a "trigger" if you could record and subsequently analyze every event that was produced. The need for a trigger is always due to the scarcity of some resource that prevents you from doing this. <u>A trigger is a necessary evil!</u>

The limitations that have forced one to use a trigger system have varied over time. In practice, any problem in keeping pace with the event rate results in "deadtime."Deadtime is a time period when data from the experiment is available, but data cannot actually be recorded because the trigger or DAQ system is unable to accept it because it is busy with a previous event. Sources of deadtime include:

Detector deadtime: In the past, detectors often needed to be put in a "sensitive" state and "fired" or "triggered" to record data when there was an interaction. While the detector was "recovering", more data could not be recorded. It was critical to fire only when there was an interaction of interest. Any good interactions that were "missed" while the system was recovering may as well never have happened. Modern detectors may still not be able to respond fast enough if two events occur very close together in time.

Trigger deadtime: In early trigger systems, the trigger itself could only handle one interaction at a time. While it was making a decision, it could not accept other events so these events were lost.

Readout deadtime: In early data acquisition systems, only one event could be read out at a time. During the long time interval required to record the event to an archival medium (usually much longer than it took to make a trigger decision), it was impossible to accept any more events for readout and it was customary to disable or block the trigger system from accepting new events during the readout period. Any events that occurred while one event was being read out were lost.

Storage limitations: Sometimes the writing/recording of data to the archiving system – conventionally magnetic tape – could not keep up with the event rate and, without affordable buffering, events were lost.

We shall see below how modern electronics techniques can be used defeat all these sources of deadtime.

In modern systems, an important reason for needing to employ a trigger is related to data storage and data access issues. A typical experiment now works very hard to reduce the data rate to archival storage to 200 Mbytes/sec. In a typical run of one year, accounting for scheduled data-taking time, accidental downtime, residual deadtime, etc., this results in 2 Petabytes (10^{15} bytes) of data. Typically, the output of computations done offline on the raw data produces additional “derived data” and this data must also be saved. The raw and derived data together may be as much as three times the size of the raw data alone, or about 6 Pbytes/year total. The cost of disk storage today is about $1/Gbyte or about $1M/Pbyte, but drives, robots, servers, operators, and networking can result in a large factor beyond this to support it. Therefore, affording the data storage, and more important the additional cost of being able to conveniently access the data, is a large expense which may limit how much data can be recorded.

For example, to get down to a level of 200 Mbyte/s of raw data, the trigger for the BTeV[2] experiment must select one out of every 2000 events to record; the triggers for the CMS[3] and ATLAS[4] experiments select about one event out of every 40,000.

2.3 General Considerations in Designing Trigger/DAQ Systems

2.3.1 Trigger algorithms

To make a trigger decision, there must be a physics property that distinguishes the “desired events” from the “unwanted events” that the trigger can exploit. The degree of complexity of the “differentiator” with respect to the raw signals generated by the front ends is a key determinant of how difficult it will be to implement a trigger.

Table 2 gives a list of properties that are significant in the design of a trigger and DAQ system. (The terms Level 1 and Level 2/3 trigger will be explained below).

One type of experiment is a so-called high P_t (transverse momentum) experiment. These experiments are interested in “hard” collisions that make energetic particles or particle clusters at large angles to the beam. The triggers in these experiments are looking for basic objects: high P_t electrons, muons, photons, jets, or events with large missing energy. The unwanted “soft” events have a much more steeply falling transverse momentum distribution than the hard events. The trigger must identify the basic objects and require them to exceed a “P_t threshold” to decide to take the event. Except for muons, these objects are found tby examining the information from the calorimetry, and the P_t can be determined by weighting the energy, given by the pulse height, from the contributing calorimeter towers by their angle. Sometimes tracking detectors are used to improve the “turn-on” threshold, especially for electrons. Various conspiracies can make a soft event, or several soft events occurring close together in time, look like a hard event and thus fool the trigger. Nevertheless, it is usually possible with a suitable choice of the P_t threshold to heavily suppress the background from soft events while retaining high efficiency for hard events. A schematic of the high P_t triggers that will be implemented for the CMS[5] experiment is shown in figure 2. Figure 3 shows the pattern recognition for finding the key high P_t objects in the calorimeters and compares this to the complexity that would be needed to use tracking in the triggers.

What is the total event rate of beam crossing rate?
Is the interval between events/crossings periodic or continuous?
If the interval is periodic, what is the frequency?
If the events are continuous, what is the average rate?
What is the response time of the various detectors?
How many "hits" occur in each detector per event or crossing?
How many bytes are produced per event/crossing?
What is the maximum bandwidth or rate of the Level 1 trigger?
What is the maximum bandwidth or rate of the Level 2 trigger?
What is the maximum bandwidth or rate of data to archival storage?
How much archival storage is available (or what is the cost of storage)?

Table 2: Typical questions pertinent to the design of a trigger and DAQ system. A three level trigger hierarchy is assumed.

A different trigger problem involves situations where charged track reconstruction is required to differentiate the desired events from the unwanted ones –so-called "tracking triggers". A good example, and an extreme one, of this kind of trigger occurs in the BTeV[6] experiment. This experiment will study the decays of particles containing "b-quarks." Only about 1/500 of the interactions at the 2 TeV center of mass energy of the Tevatron have a b-quarks. There are relatively small differences between these events and " minimum bias" events – higher kaon content, more tracks with P_t above about 1 GeV/c. **But the main difference is that the B EVENTS ALL HAVE DECAY VERTICES DOWNSTREAM OF THE MAIN PRODUCTION VERTEX, which are characteristic of a lifetime of about 1.5 x10^{-12} s for B decays. These decay vertices are displaced by about a millimeter from the production vertex (due to time dilation).** The trigger requires that tracks are reconstructed from the tracking detectors and then aggregated into vertices. The trigger is based on an indication of both a primary interaction vertex where the B is made, and some evidence for one or more vertices a little downstream of the primary vertex characteristic of the B decay. This is shown schematically in Fig. 4.

2.3.2 Dead Time

Minimizing the amount of dead time, or maximizing the live time, is an important function of the trigger system. The ultimate goal is a "deadtimeless system", both for the trigger and DAQ.

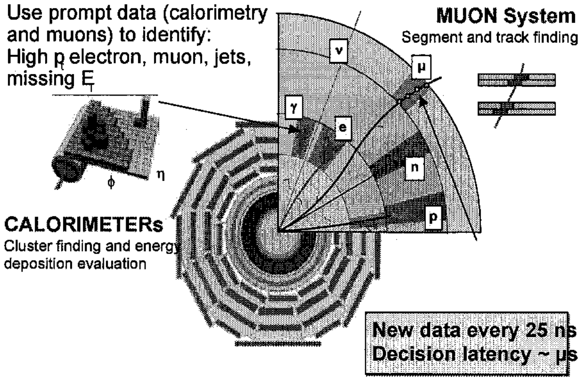


Figure 2: Representation of the physics basis for high p_t triggers based on calorimetry. The response of the various portions of the calorimeter to electrons (e), photons (γ), protons or pions (π), neutrons (n), and muons (μ) are indicated. The inner (pink) section is the tracking region and a solid line indicates a trajectory from a charged track. A dotted line or dashed line indicates the absence of signal from a neutral track. The green region is the electromagnetic calorimeter. The sequence of disjoint lines indicates a splash of energy deposited from a shower. A solid line indicates a "minimum ionizing" signal. A dashed line indicates no signal. The purple region is the hadron calorimeter, with the symbols having the same meaning as for the electromagnetic section. The outer pink region is the muon detector, which muons and neutrinos (ν) are penetrating enough to reach . Muons appear in the tracking chambers as minimum ionizing particles. Neutrinos don't interact at all in the detector but their presence is signified by "missing energy" or "missing transverse energy".

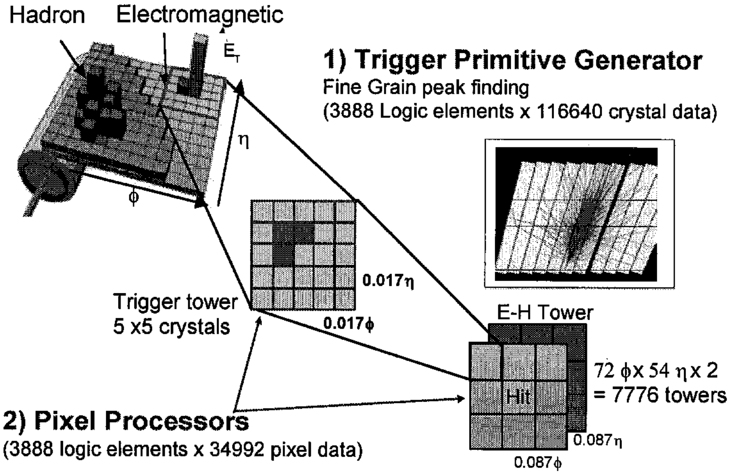


Figure 3: The upper left part shows a LEGO plot of the energy deposition in the CMS electromagnetic calorimeter. In the middle, we see an electromagnets shower in a so-called trigger tower, a 5 x 5 arrays of crystals. The lower right indicates a processor looking for a shower in a tower of the electromagnetic calorimeter and the corresponding tower elements of the hadron calorimeter.

2.3.3 Multiple Triggers and "Prescales"

Experiments always have "multiple" triggers, which typically include the main physics triggers, calibration triggers, triggers on samples that will be systematics limited at well below the capacity of the system to log them and lower priority, high rate triggers. The actual trigger (at any level) will be the OR of the individual triggers. The i-th(sub) trigger may be PRESCALED (I.e. only $1/N_i$ such triggers are taken). **However, the effective**

luminosity of a prescaled trigger is $\mathcal{L}/N_i$, so you try hard not to prescale the main physics triggers! This is a kind of Selective Deadtime based on physics priority.

2.3.4 Downtime

Downtime is when the experiment is supposed to be taking data but for some reason it is not. Sources of downtime include a failure in the accelerator or beam, the detector, the DAQ or trigger, and pilot (operator) error. Downtime is not Deadtime but the effect is the same – no useful data!

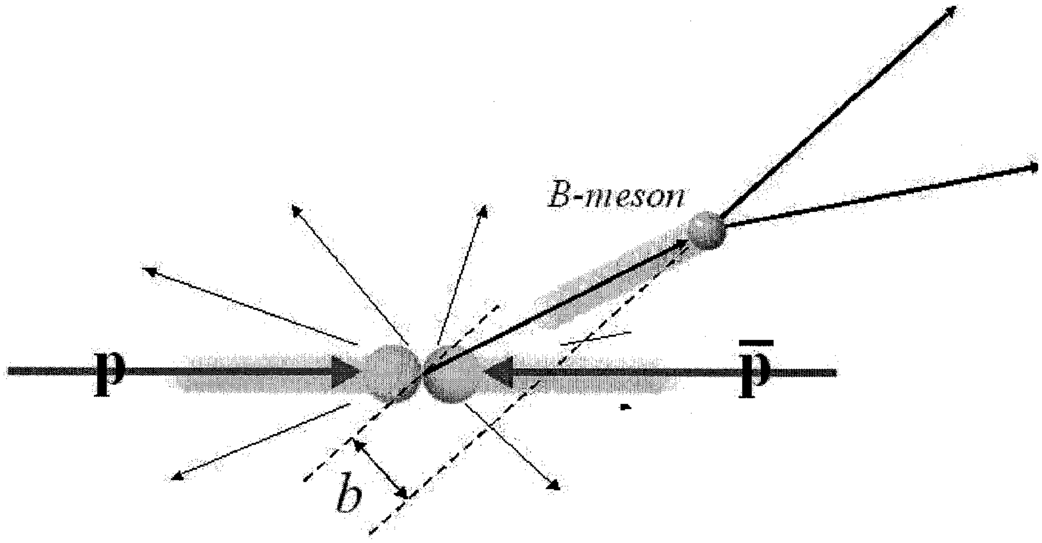


Figure 4: The physics basis for the "detached vertex trigger", a tracking trigger used in BTeV, is shown here. Tracks emerging from the collision are used to reconstruct the position of the "interaction vertex". Tracks from a B decay verticize at a different point downstream of the interaction and individual daughter tracks from the decay do not point to the interaction vertex.

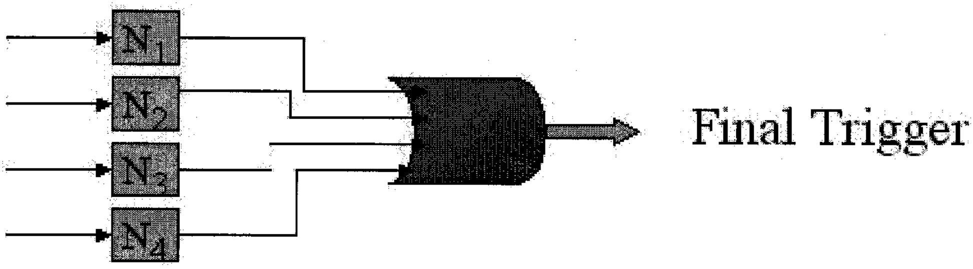


Figure 5: Schematic of a system with 4 "sub-triggers", each separately prescaled, and "orred" together to form the final trigger.

2.3.5 Trigger Bandwidth, Trigger Cross Sections

At any level in the trigger and DAQ system, if we call the maximum allowed rate the "bandwidth", then each trigger (after prescaling) can be said to consume a fraction of the total bandwidth which is the fraction of the total triggers satisfying it times the maximum rate. People describe adjusting trigger prescales as "managing the trigger bandwidth". At any level, the rate (after prescaling) of a particular trigger, can be described as

Rate of trigger i = Luminosity x cross section for trigger i

Inverting this gives: **cross section for trigger i = rate of trigger i/Luminosity**

2.4 "Pipelining" and "Hierarchical Triggering" as a Way to Avoid Deadtime

With modern electronics, there are obvious ways to eliminate or reduce deadtime. Consider that while the trigger is working on the calculations for one event, another occurs. **If you can save the data for the new event in buffer memory or in a "pipeline", then you can come back to it when you finish the first trigger decision and work on the second event. With enough buffer memory, the trigger can take much longer than the time between events to make a decision. However, it must still make decisions at the average event rate. So, this requires either very fast trigger hardware or parallel trigger hardware – several parallel systems handlings several events at once.** This technique is called "pipelining" because events are viewed as traveling through a "pipeline" of storage and processing elements until they reach the end when the trigger has decided whether to accept it. A FIFO is an example of a hardware implementation of a pipeline. The pipelined trigger is most correctly described as a quasi-real-time system.

Pipelining makes it possible to use "clusters" or "farms" of computers to perform very complex trigger algorithms and still keep up with the data. Below, we trace the history of trigger systems and show how increased sophistication and complexity in the algorithms led to hierarchical triggers, consisting of several "levels'" or "layers of complexity" based on pipelining and parallel computing.

2.4.1 Early Systems

Early systems used the simplest aspects of a collision for the trigger – hits or energy deposition in a few detectors. Later, they began to use more complicated quantities, such as the total transverse energy – ΣE_i x polar angle$_i$ – which could be done on specialized hardware boards with weighting schemes or ALU's (Arithmetic-Logic Units). Since this usually took longer and caused more deadtime, only events which passed the simple hardware trigger – now called "Level 1", were sent to this new, more complex and time consuming subsystem – "Level 2", thus giving a two level hierarchy.

2.4.2 Enter Computing and Level 3

Once microprocessors became available, it became the practice to add a "Level 3" to the "Trigger hierarchy." Level 3 generally uses FARMS or CLUSTERS of general purpose microprocessors to do much more sophisticated computations, but on the relatively small number of events passing Levels 1 and 2. In HEP, these were developed in parallel with the use of FARMS or CLUSTERS to exploit the "embarrassingly trivial parallelism" of OFFLINE analysis – I.e. each event is an (almost) independent analysis problem as far as event reconstruction goes. Similarly, each event is a separate trigger problem for a processor of the Level 3 cluster.

2.4.3 Computing Invades Level 2!

Within a very few years, general purpose computing was appearing also in subcomponents of the Level 2 triggers and even, in a few cases, for specialized purposes in Level 1. At the same time, FPGA's (Field Programmable Gate Arrays), PLA's (Programmable Logic Arrays), associative memories, etc were beginning to blur the distinctions between computers and combinatoric logic.

2.4.4 Rationale for a "Trigger Hierarchy"

Consider an experiment with a collision rate R1.

1. Level 1: R1 is the input rate to the Level 1 trigger and
    - The average decision time[7] must be at most 1/R1
    - If the Level 1 trigger accepts a fraction f1of all events, then
        - The Level 1 output rate is f1x R1
2. Level 2: The input rate to Level 2 is f1xR1
    - The average decision time must be at most 1/(f1XR1)
    - If the Level 2 trigger accepts a fraction f2 of all Level 1 triggers, then
        - The Level 2 output rate is f2xf1xR1
3. Level 3: The input rate to Level 3 is f2xf1xR1
    - The average decision time must be at most 1/(f2xf1xR1)
    - If the Level 3 trigger accepts a fraction f3 of all Level 2 triggers, then
        - The final trigger output rate is f3xf2xf1xR1

The trigger described above must output f3xf2xf1xR1 events to archival storage. If the average event size of "triggered events" (which may be different, often larger, than the average size of all events) is M bytes, then the total data rate to archival storage is Mxf3xf2xf1xR1. The requirement on the output trigger rate might well be set by ability of the DAQ to write data or the ability of the experiment to afford data storage.

The Level 1 trigger has to operate, on average, very quickly and has time to carry out only a relatively simple computation based on relatively simple "trigger primitives" – pulse heights, latch bits, etc. The computations must allow for the maximum time to process an event so it is best, or at least simplest, to use algorithms that do not have large time variations and allow easy and natural parallelism. Pattern recognition on calorimetry has this characteristic as can be seen from Fig. 3 above. Similarly, asking for "hits" in a muon hodoscope is relatively simple. Correlating hits between several tracking stations takes much longer and has much greater event-to-event variation. It is also is vulnerable to electronic noise and beam background, and is, therefore, not often used at Level 1. The average decision time available at Level 3 is often long enough to permit many computers to each work on one event in parallel. Since tracking is often used at this level, the variation in times for different events is great.

2.5 Examples of Modern Trigger Systems

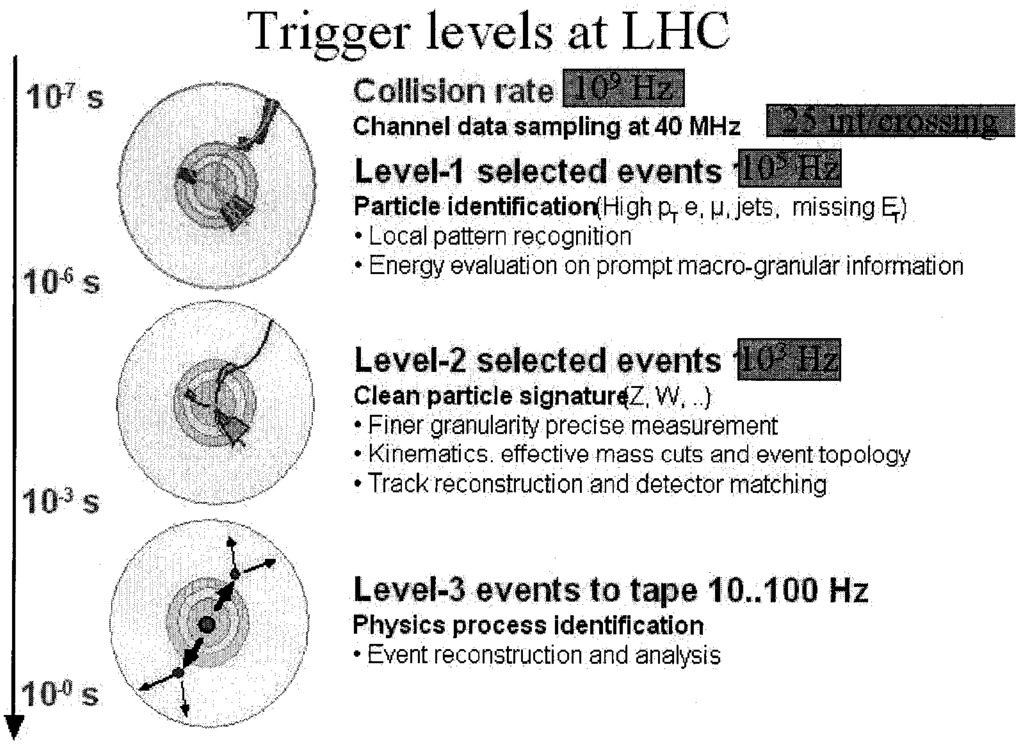
2.5.1 LHC experiments

ATLAS and CMS are examples of experiments that have Fixed Latency Trigger systems at Level 1. This means that a fixed time is allowed for the trigger to process each event. At the end of this time, the event is either accepted for further consideration by Level 2, or is discarded. To accomplish this, data are buffered in fixed length pipelines. With 25 ns between crossings, they buffer about 64 events, for about 2 microseconds. This is referred to as the "Level 1 latency": the time delay between when the data is generated and when the Level 1 trigger decision is completed. The trigger must make a decision every 2 microseconds, which includes all the time to acquire the data (over cables, etc) and all the time to send it back to front ends where the data are stored. This is a demanding task and limits the complexity of the trigger calculations. Speed is critical. Even today's fast processors may only be able to do 1000 instructions in the time allowed, so custom hardware may be necessary. To implement this kind of system, there may need to be all kinds of "regional parallelism" – I.e. processing of layers or quadrants (or octants, etc) of the detector in parallel – to meet the speed requirements. In an ATLAS-type trigger, much of the data will be buffered in the detector to await trigger decisions. The length of the "pipeline" adds complexity, power, cost, and even scattering material to the detector. If the buffering is done close to the detector, the electronics may need to be radiation hard, which increases the cost significantly.

The Level 2 trigger is now almost always implemented as a "cluster" of PCs or similar computers. ATLAS[8] retrieves information from regions of the detector, so-called REGIONS OF INTEREST, and sends them to a processor for a Level 2 decision. This is called "partial event readout". Typically, for an event surviving Level 2, all the event data is transferred to an available processor and the processor performs a calculation that is very similar to a full offline analysis and serves as a very high level physics filter. If it determines that the event is of interest, then it sends it to the DAQ for output. Otherwise, it discards the event. Decision times will vary greatly and the decision will be in force as soon as it is known. Thus, Level 3 latency is variable. If this were not so, every event surviving Level 2 would have to be buffered for the maximum time it takes to perform the Level 3 calculation rather than for the average decision time.

Currently, the Level 2 trigger is hardest to characterize and shows the most variation from experiment to experiment. Some experiments still use specialized hardware at Level 2, some use a mixture of specialized hardware and conventional computing, and others merge the Level 2 and 3 triggers into the same computing farm. The CMS trigger merges Level 2 and 3 into the so-called "High Level Trigger " (HLT). In such cases, we refer to a Level 2/3 trigger. Level 2 triggers also tend to have variable latency.

Figure 6 shows the projected rates at each trigger level for the LHC experiments ATLAS and CMS. Figure 7 shows the full readout structure. Both experiments use specialized hardware at Level 1 and have a fixed latency of about 2 microseconds. Note the differences at Level 2, described above.



Trigger levels at LHC

Collision rate 10^9 Hz

Channel data sampling at 40 MHz 25 int/crossing

Level-1 selected events 10^5 Hz

Particle identification(High p_T e, μ, jets, missing E_T)
- Local pattern recognition
- Energy evaluation on prompt macro-granular information

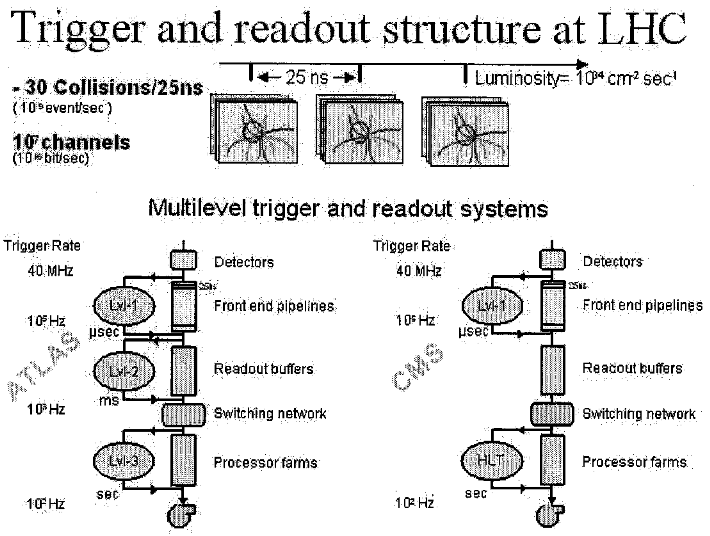
Level-2 selected events 10^3 Hz

Clean particle signature(Z, W, ..)
- Finer granularity precise measurement
- Kinematics. effective mass cuts and event topology
- Track reconstruction and detector matching

Level-3 events to tape 10..100 Hz

Physics process identification
- Event reconstruction and analysis

Figure 6: A description of trigger levels for the CMS and ATLAS experiment at the LHC.



Trigger and readout structure at LHC

- 30 Collisions/25ns
(10^9 event/sec)

10^7 channels
(10^{16} bit/sec)

Multilevel trigger and readout systems

Figure 7: The full trigger and readout structure for the ATLAS detector (left) and the CMS detector (right) at the LHC. Note the difference at Level 2 between the two experiments.

2.5.2 The BTeV Experiment

BTeV is an experiment at the Fermilab Tevatron Collider to study the matter-antimatter asymmetry of decays of particles containing the b-quark, a quark which is about 5 times the mass of the proton and decays away with a mean lifetime of 1.5×10^{-12} s. When produced at high energies, Einstein's time dilation allows these particles to go a few millimeters from their production point before they decay. BTeV has enough tracking precision to reconstruct the interaction vertex and the decay vertex and can therefore isolate and study the decays of b-particles. The goal is to study the asymmetry (difference in rate) between the decay of b-particles and b-antiparticles. Without these kinds of asymmetries, in the early universe all matter would have found an antimatter particle to annihilate with into pure energy (photons) and there would be no matter excess to form the universe we have today!

While this is an important physics problem in its own right, the main reason for discussing it here, is that the experiment will employ a very complex tracking trigger that requires the development of new state of the art hardware and software. It will constitute a new advance in trigger and data acquisition systems that take the trends discussed above to their logical endpoint. A schematic of BTeV is shown in Figure 8.

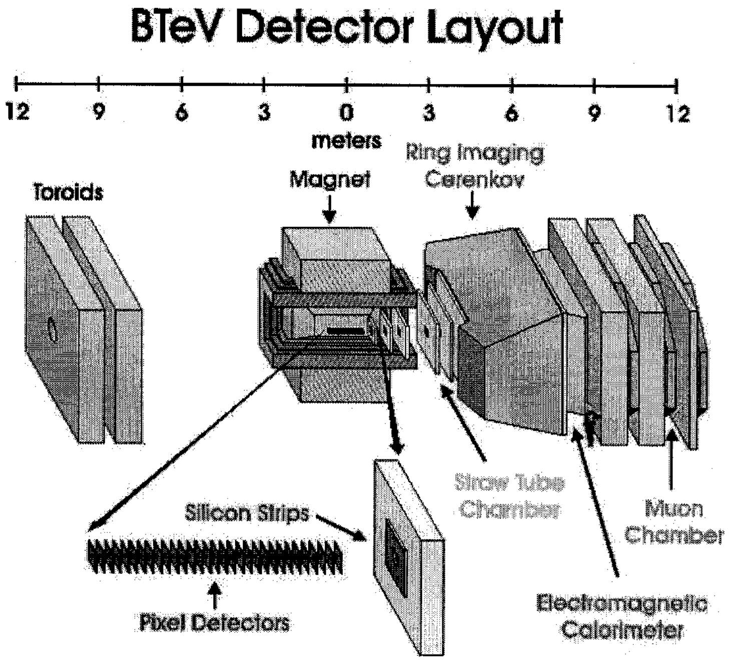


Figure 8: A schematic of the BTeV detector. Vertex reconstruction is provided by a 23 million channel silicon pixel detector, specifically designed for use in the Level 1 trigger.

Key features of BTeV are: 1)A **dipole located ON the Interaction Region (IR), gives BTeV a forward spectrometer** -- covering the forward anti-proton rapidity region; 2) A precision vertex detector based on **planar pixel arrays. The readout is designed to be fast so the data can be used in the Level 1 trigger; 3) A vertex trigger at Level 1 makes BTeV especially efficient for states that have only hadrons. The design of the tracking system has to be tied closely to the trigger design to achieve this;** 4) Strong particle identification based on a **Ring Imaging Cerenkov counter.** Many states emerge from background only if this capability exists. It enables use of charged kaon tagging; **5) A lead tungstate electromagnetic calorimeter for photon and π^0 reconstruction; And 6)a very high capacity data acquisition system which frees us from making excessively restrictive choices at the trigger level.**

The goal here is to make a trigger that can reconstruct tracks coming from the interaction vertex and to find evidence for the presence of tracks coming from a detached downstream decay vertex of a B particle. To do this, we need a superb tracking detector. In BTeV, we use a silicon pixel detector. The pixel detectors are arrayed in 30 stations along the beam in Z, as shown in figure 9. There must be a hole in the center of the detectors to allow the colliding beams to pass through them without encountering any material. The pixel dimensions are 50μm X 400μm. At each station, there are two pixel planes – one which has the short – 50 μm – dimension along X and one which has the short dimension along Y. Thus, each station gives highly accurate 3-dimensional measurements of the track intersections with the planes.

The reasons for using pixel detectors, instead of the more conventional silicon strips, are **superior signal to noise; Excellent spatial resolution -- 5-10 microns depending on angle; Very low occupancy (10^{-4}); Very fast – delivers a signal in well under the time between beam crossings at the Tevatron; and they are radiation hard and can survive operation very near the beam.**

Pixel detectors are now being used or developed for several experiments. BTeV's pixel detector[9] is similar and some respects but has the following unique features: **it is used directly in the Level 1 trigger; Pulse height is measured on every channel with a 3 bit FADC to permit one to improve the spatial resolution by exploiting beam sharing; and it is inside a dipole and gives a crude standalone momentum.**

Figure 10 shows what a "normal" event looks like in the silicon detector. There is one of these every 132 ns (7.6 MHz), although many crossings have a few of these superimposed. Events have vastly different numbers of produced particles, and so a big variation in the number of struck channels. The detector response is faster than the 132 ns interval between consecutive events. For this detector, there are 25 million channels but only a few thousand have signals from track passing through in any given event. Total event – all detectors – have 200Kbytes/event, for a raw, sparsified rate of 1.5 Terabyte/second. Runs are last for about 8 hours each and go on 24 x 7 for about 40 weeks a year.

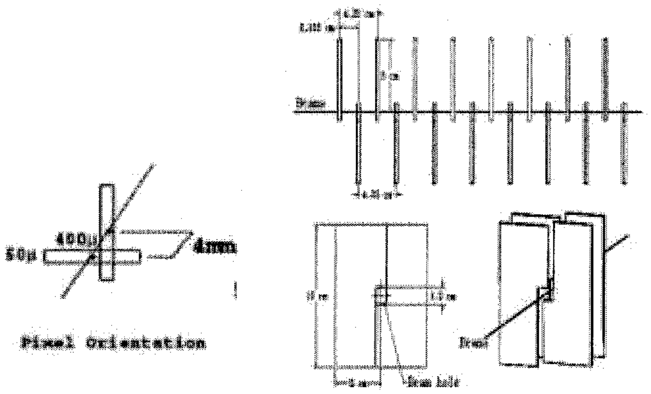


Figure 9: Configuration of the BTeV pixel detector. The figure on the left shows the pixel orientation described above. The figure on the upper right shows the position of the stations along the beam direction (Z). Each station has two planes with the two orientations shown on the left. The figure on the lower left shows the way each station looks to the beam. The hole that allows the beams to circulate is 12mm x 12mm. The planes are on a drive system that allows them to be moved farther out during the filling of the Tevatron and acceleration.

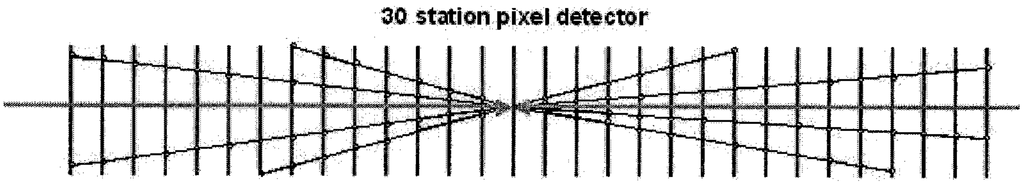


Figure 10: A collection of pixel hits in one view in the BTeV pixel detector

BTeV implements this vertex trigger at the lowest level – Level 1—using massive amounts of computing and **extreme pipelining** techniques. This is the final step in the evolution of triggers in that it extends computing to all aspects and levels of the trigger. The different levels of the trigger system are now mainly distinguished by the complexity of the algorithm; what part of the total event data they have access to; and what fraction of the events they run on. At present, there are also minor differences in the computer hardware at various levels, strictly due to cost considerations. **The trigger will reconstruct every beam crossing and look for TOPOLOGICAL evidence of a B decaying downstream of the primary vertex. It runs at 7.6 MHz!**

This is made possible by 1) a vertex detector with excellent spatial resolution, fast readout, low occupancy, and 3-d space points. The detector is in a magnet and the pixel resolution is good enough to that get a useful momentum measurement; 2) A heavily pipelined and parallel processing architecture using inexpensive processing

nodes optimized for specific tasks ~ 2500 processors (DSPs); and 3) sufficient memory (~1 Terabyte) to buffer the event data while calculations are carried out.

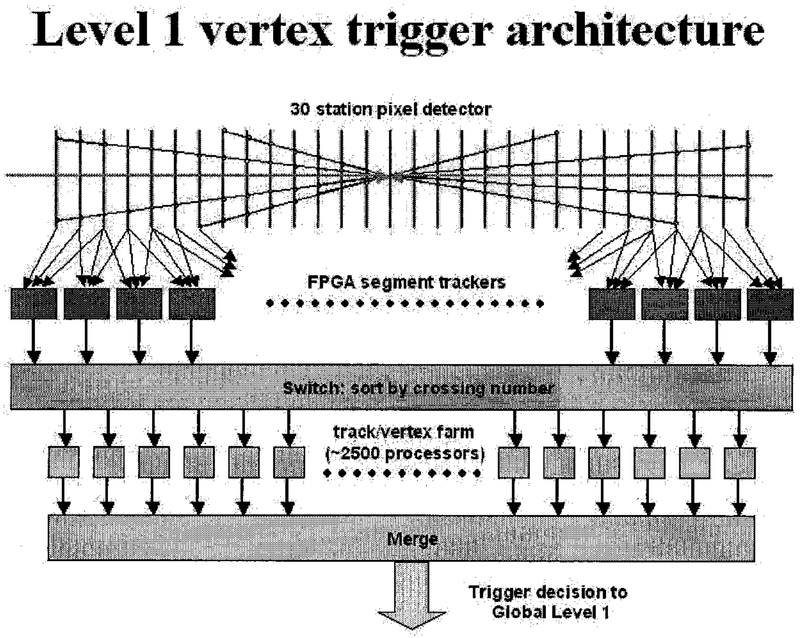


Figure 11: The Level 1 trigger organization. Track segment finding is done with FPGA's; segment are linked into tracks and vertices are found in 2500 processors, probably DSP's. The output goes to "global level 1", which makes the final Level 1 trigger decision.

The full Level 1 system is shown in Figure 11. The pixel data are sent to a very large system of several hundred FPGA's which gather all the hits from a single crossing and perform pattern recognition to find track segments of particles entering the inner edge of the detector from the Interaction Region and also to find segments that exit the detector from the outer edges or the upstream and downstream end of the pixel detector. The track segments are then sent to a "switch" which sorts them and sends all the segments associated with one crossing to one of 2500 DSP's (Digital Signal Processors) which match the inner and outer track segments to form single tracks. The vertical angle difference between the inner and outer segment of a track gives a measurement of the track momentum and transverse momentum. The low transverse momentum tracks (but above a certain momentum) are then formed into "primary" vertices. High transverse momentum tracks that do not point to the primary vertex, so-called "detached tracks", are evidence for B decays and form the basis of the actual trigger decision. Specifically, for each track not included in the primary vertex, the program computes the impact parameter, b, shown in Fig. 4 above. To be considered "detached" the impact parameter, b, divided by its computed measurement uncertainty, σ, must exceed a threshold cut M, typically set at 6. Thus, we require a track to miss the primary vertex by about M~6 standard deviations before we consider it detached. We also have a requirement on the number of "detached tracks", N. Typically, we require N>=2, so we finally require at least N tracks, each detached from the primary by at least M standard deviations in order to accept the event.

The algorithm has been coded and run on data simulated using GEANT3[10] and a very complete model of the detector and all its material. Figure 12 shows the performance of the trigger algorithm. The signal is taken from a study of the B decay $B_s \rightarrow D_sK$, which is important for measuring one of the key parameters of CP violation. The goal is to be very efficient on "beam crossings" with one of these events. The "background" is composed of "minimum bias" or "soft" events. We generate minimum bias events with a Poisson distribution with an average of two per beam crossing. The trigger should be inefficient on these background events. The plots show the trigger efficiency for various values of N and M as described above. It can be seen that if we set the trigger requirement at N>=2 and M>=6, then the trigger efficiency on the B decay is 74% and the efficiency on minimum bias events is only 0.8%, or a rejection 0f 99.2%! This is excellent performance. Figure 13 shows the efficiency using this set of cuts on a wide range of interesting B decays. It can be seen that the trigger is quite efficient on all topologies.

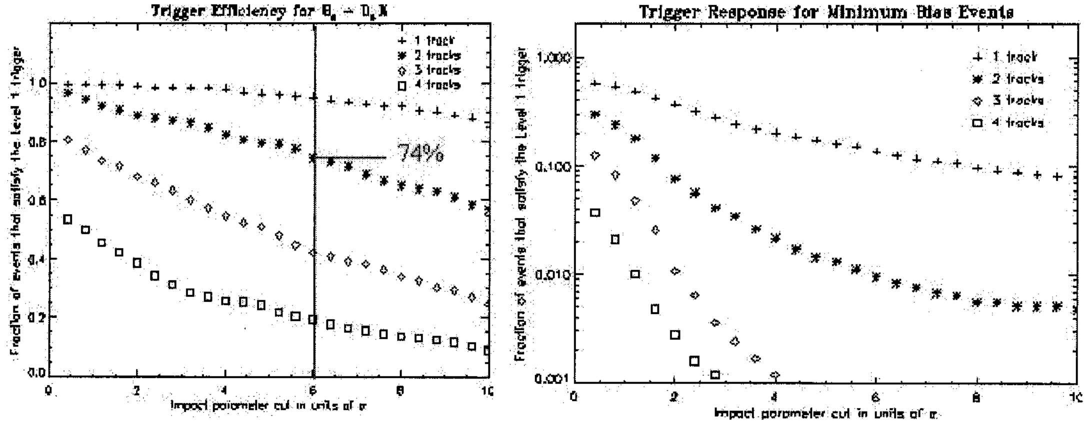


Figure 12: Left side: Level 1 trigger efficiency for signal events. Each curve is for a different value of N(defined in the text) from 1 to 4. The vertical axis is the efficiency. The horizontal axis is the cut M (defined in the text); right side: Level 1 trigger efficiency for minimum bias events.

Process	Eff. (%)	Monte Carlo
Minimum bias	1	BTeVGeant
$B_s \rightarrow D_s^+ K^-$	74	BTeVGeant
$B^0 \rightarrow D^{*+}\rho^-$	64	BTeVGeant
$B^0 \rightarrow \rho^0\pi^0$	56	BTeVGeant
$B^0 \rightarrow J/\psi K_s$	50	BTeVGeant
$B_s \rightarrow J/\psi K^{*o}$	68	MCFast
$B^- \rightarrow D^0 K^-$	70	MCFast
$B^- \rightarrow K_s\pi^-$	27	MCFast
$B^0 \rightarrow$ 2-body modes $(\pi^+\pi^-, K^+\pi^-, K^+K^-)$	63	MCFast

Figure 13: Level 1 trigger efficiency for a variety of important decays, using either GEANT3 or the fast simulation package MCFAST[11].

The Level 1 trigger is quite complex. The average time it takes to do all the calculations on a crossing is about 200 microseconds. Data from the other detector components must be saved while the calculations are being performed. This is done by storing it all in a large buffer memory, perhaps as big as a terabyte! As soon as the Level 1 trigger decides to keep an event, a signal is sent to the buffer memory to transfer the event to other memory space for access by Level 2. If the event is rejected, the memory occupied by the event data in the Level 1 buffer is freed. The massive parallelism, being able to work on up to 2500 hundred events simultaneously, is what makes all this possible. This parallelism in turn is made possible only by the availability of inexpensive processors – in this case, DSPs, which are used in cell phones and many other applications – and very cheap computer memory – $0.25/megabyte – used for the "pipeline". It is worth noting that the Level 1 latency in BTeV is variable!

The remaining levels of the BTeV trigger are shown in Figure14.

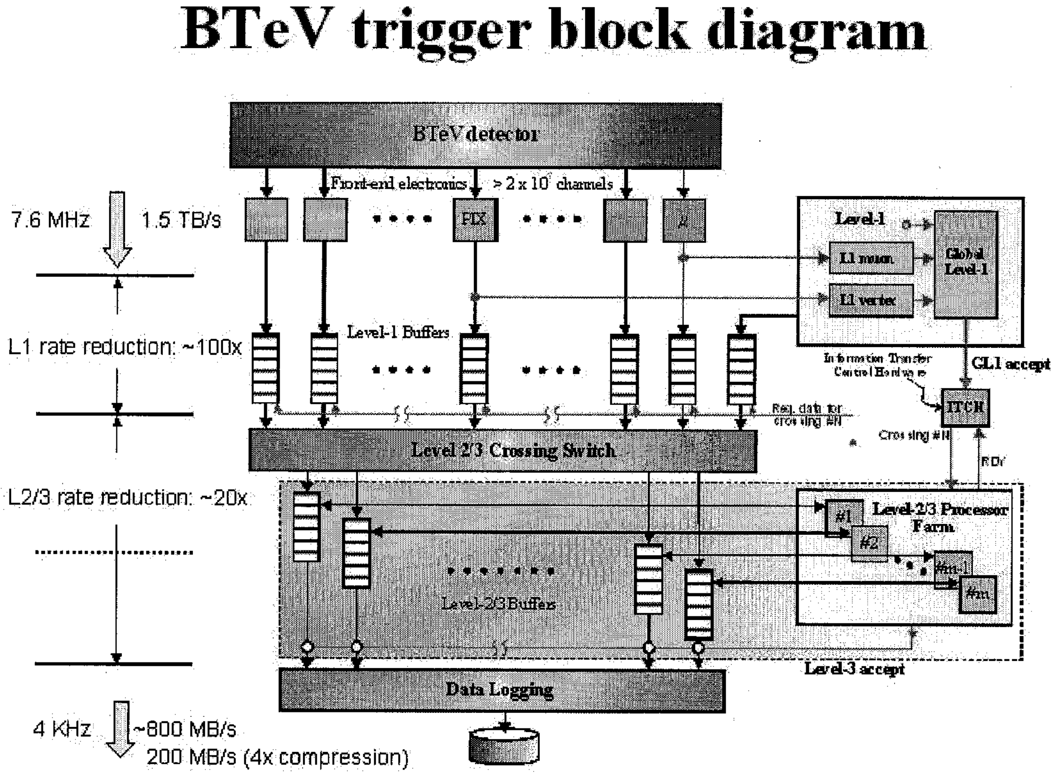


Figure 14: Schematic of all three levels of the BTeV trigger. On the left side, we show the (crossing) rates at each Level and the amount of data entering the system and being written out. The Level 1 and Level 2/3 farms are shown on the right. Buffer memories and the network switch fabric are shown in the center.

Figure 14 shows how the pixel data are routed both to the buffer memory and to the Level 1 trigger farm (in the upper right corner of the diagram). The farm, in all its complexity described above, is shown as the small box labeled "L1 Vertex". BTeV has another Level 1 trigger, a muon trigger, and the data is similarly intercepted and routed to a second (but smaller) Level 1 farm, called "L1 Muon". The subsystem called "Global Level 1," effectively combines the two triggers and generates the final Level 1 decision. The Level 1 trigger rejects 99% of the events, retaining nearly 75% of all useful b-quark events. When an event passes the Level 1 trigger, it is moved to the Level 2/3 buffers. The Level 2/3 trigger is carried out by a "cluster" of about 2500 commercial PC's, probably Pentium based, running the LINUX operating system. Events (actually beam crossings) passing Level 1 are assigned to an available Level 2/3 processor by the module called the "ITCH". Sophisticated physics algorithms use both the pixel detector and the forward tracker to do a more precise vertex reconstruction with better-defined tracks and more accurate momentum measurements. Some additional topological requirements are placed on the events and this results in a further reduction of a factor of about 20 in the total rate. Crossings surviving the Level 2/3 trigger are sent to the DAQ, a series of computers and disk buffers which eventually writes the events to archival mass storage where it is available for subsequent offline analysis which will produce the final physics results.

In BTeV, the distinction between the Level 2 and Level 3 trigger is really in the algorithms. When a processor is assigned a crossing, it will handle all remaining computations. First it does a relatively quick recalculation of the Level 1 algorithm but using information from the forward tracker. This runs relatively quickly and eliminates about 80% of the Level 1 triggers, mostly having false or poorly measured tracks. This is the Level 2 algorithm and crossings that fail it are rejected immediately. (It is possible to do the Level 2 algorithm with only the pixel and forward tracker data, but BTeV has not yet decided whether to just read these detectors in or read the whole event in for Level 2. If we only have "partial event readout" then this would be another distinction between Level 2 and Level 3 – the amount of data accessible to each.). In Level 3, all the data for the crossing is available and the algorithm applies more sophisticated topological selections than just the simple impact parameter. It rejects about 75% of the events passing Level 2. The total rejection from Level 2/3 is about a factor of 20 and its overall efficiency for signal events is over 90%.

Note that "event building" which will be discussed below, is accomplished when the Level 2/3 processor requests the data from the crossing it has been assigned from the buffer memory. The buffer memory controllers arrange to route the events fragments through the switching network to get them all to the requesting processor.

There are several unique features in the BTeV trigger system design: There are massive amounts of computing at every level, including Level 1; There is no fixed Level 1 latency; There is no on-board buffering or pipelining in the front ends. Data are "sparsified" on the fly and shipped off the detector every 132 ns; Every event fragment has a timestamp, which is the "crossing number". The crossing number enables the trigger and DAQ to correctly associate the various pieces of an event; The events do not emerge from any trigger level "ordered" the same way that they occurred in nature. This doesn't matter !; There is no fixed latency or ordering at Level 2 or Level 3 or to archival storage; Since archived data resides on disk for very long periods of time, additional analysis steps that can be thought of as Level 4...N triggers[12] , could in principle be carried out with a latency of days, weeks, or months.

3.0 Data Acquisition

The Main functions of the DAQ are

- To accept and buffer data from the front ends while the Level 1 trigger makes a decision. AT THIS POINT, THE EVENTS ARE USUALLY IN FRAGMENTS THAT CAN BE RELATED BY A TIME STAMP OR CROSSING NUMBER;
- For events satisfying the Level 1 Trigger, to BUILD COMPLETE EVENTS – that is, put the fragments together – into a single (logical) block of data – or in some cases, to provide groups of event segments to the Level 2 trigger (partial event readout) before providing the full event to Level 3;
- To move the “built” or “partially built” events to the appropriate trigger level;
- To archive – store on permanent or persistent storage (disk or tape) the data chosen by the final level of trigger to be kept for subsequent analysis; and
- To provide whatever controls, interfaces, services, etc needed to keep the whole trigger/DAQ system running correctly.

3.1 Sparsification

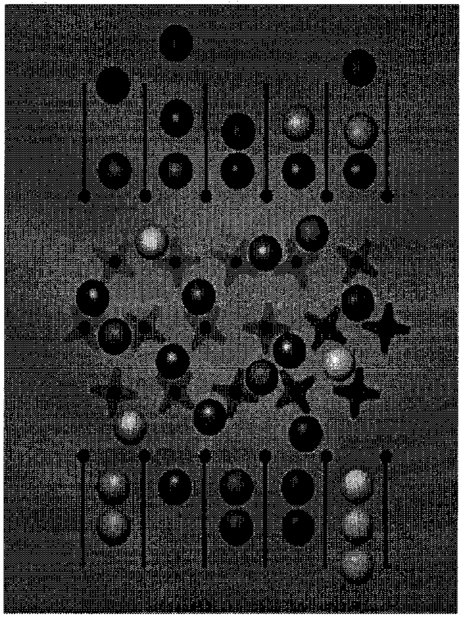
Sparsification may be considered a front end function or a DAQ function. It refers to the act of eliminating data which is consistent with being detector or electronic noise and not consistent with being from a signal. This relies on each detector having good separation between signal and background. If this is the case, then the amount of data that needs to be “buffered” or “pipelined” will be small. If it is not true, then threshold discrimination will either be set low and will accept a lot of “noise” which will need to be buffered or, if the threshold is set high, to control the amount of data, some signal will be lost.

3.2 Buffering

Buffering is a critical element of trigger/DAQ systems. We have seen that it is necessary to implement a “pipeline”, so that the trigger calculations can take much longer than the time between events as long the average decision rate keeps up with the average event rate. With the rapidly declining cost of computer memory, it is easy and economical to make large buffer memories out of commercial memory chips used in PCs. In fact, even with packaging and networking costs included, it is possible to make a 1 Terabyte memory, such as might be used by BTeV, for under $1M. For BTeV, such a memory can buffer as many as 10^8 “crossings” or 1 full second of data. This is far more than the average “latency” of the Level 1 trigger, which is about 200 microseconds.

3.3 Event Building

Event building refers to the act of collecting up all of the event fragments associated with a single event or bunch crossing, which are usually scattered in different buffer memories, and assembling them into a single, contiguous event record. The fragments may be stored in front ends or in buffer memory units. The fragments are associated to a single event /crossing by an event/crossing number encoded into the fragments. The event building task is depicted schematically in Figure 13. The hardware used in event building can consist of specialized devices such a special switches which can route fragments based on their crossing number to specific locations. Nowadays it is more common for networks and computers to perform this function.



Event fragments
Event data fragments are stored in separated physical memory systems

Event builder: Physical system interconnecting **data sources** with **data destinations**. It has to move each event data fragments into a same destination

Full events: Full event data are stored into one physical memory system associated to a processing unit

Figure 15: Schematic event builder. Different colors (shadings in black and white) represent data fragments belonging to one event. The goal is to sort them so that all balls (fragments) of a single color are together

Network interconnects are used to permit the data to move from the various memory buffers to the processors. With the development of high-speed commercial data networks, it has become common practice to construct these "switches" out of conventional, commercially available networking equipment, such as high speed, 1 GHz, Ethernet gear. The use of "networks" rather than specialized switching gear is illustrated in Figure 16, which shows the CMS Event Builder.

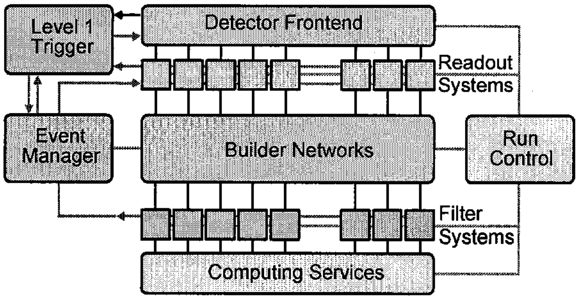


Figure 16: The CMS trigger and DAQ system showing the event builder "network"

Here, the data are buffered in the detector front ends while the "fixed latency" Level 1 trigger makes its decision. When an event is accepted, the Level 1 trigger communicates the decision to the "Event Manager", which then supervises the event building by communicating with the readout systems and the High Level Trigger system – a processing farm. The builder network in the middle routes the data to the correct processor.

In the BTeV system, a Level 2/3 processor that is available to handle a new crossing requests a crossing number from the ITCH, which maintains a list of crossings that have passed Level 1. Once it gets its crossing number, it sends a request for the data from the crossing to the buffer memory system. Each buffer memory manager with data for the crossing routes the data through the commercial network switch to the requesting computer. When the computer has received all expected fragments, then it has the full event.

One reason why it is possible to use commercial parts in the event building network is due to the rapid development of economical high speed switching fabrics for general purpose networking. Networking is now a key element of business computing and of our normal lives. The large market for networking has led to rapid development and lower prices. Table 3 shows the performance evolution (really "revolution") since 1970 of CPU, computer memory, and networking.

	Rate of increase
CPU Power	X 10 every 5 years
Memory density	X 4 every two years
Disk storage	X 2 every 18 months
Network speed	From 100 Mbit to 1 Gbit in about 5 years

Table 3: Rate of improvement in various computer technologies useful in constructing trigger and DAQ systems.

Trigger and DAQ systems take advantage of all this progress but the recent rapidly development of high speed networking now marks it as the most dynamic growth area in computing technology. It is worth noting that the density and costs of disks, not shown in Table 3, is another area of stunning improvement both in density and economy.

4.0 A Miscellany of Crucial Associated Issues in Trigger Systems

Hardware Management: Nearly every aspect of a modern experiment is computer controlled and has monitoring points to check that everything is working. These include High Voltage, Low voltage, Thresholds, Gains, Pulser systems for monitoring – light levels, gas and liquid flows, temperatures, and pressures.

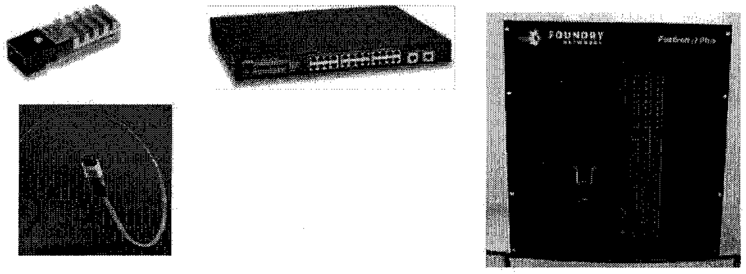


Figure 17: Four examples of commercial components used in trigger/DAQ systems: upper left, a parallel fiber optic driver (12 channels); lower left a parallel fiber optic cable; center: fanout switch with two 1Gbit and 24 Fast Ethernet ports; right: Highway cross connect and switch with 64 channel 1 Gbit Ethernet ports.

There has to be a system that can globally initialize (set it up from a random state to an operational one) all parts of the experiment or can initialize, reset, or modify any part of it. Keeping track of everything requires a huge database. It is an even greater challenge to control and monitor the entire experiment, keeping track of it behavior over several years. Monitoring is essential for detecting faults: temperature humidity, barometric pressure, and perhaps various radiation levels need to be monitored. There must be an excellent, informative, prioritized ALARMS AND LIMITS SUBSYSTEM. The system must allow the operator/user to have access to the hardware to make necessary changes and adjustments, while preventing him/her from doing anything too destructive. Automatic problem resolution has to be at least considered due to the system complexity. The analysis needs to know the condition of the detector at every point in time so the analysis probably needs to talk to these databases. This part of the system is often referred to as **"SLOW CONTROLS and MONITORING".**

Run Control, run tracking and characterization: Run control allows an operator/physicist to start, pause, resume, stop, or abort runs. This requires the system to stop triggering, to allow all events being processed to run through the system to completion, and to record the crossing numbers, times, etc of the crossings that generated the data. Histograms and statistics can be acquired on every channel, and distributions of key physics quantities can be saved and used to characterized how GOOD various runs are, to follow the trends of key variables and rates, to track changes due to changes in the machine operation, or the detector operations (dead channels, more background, some triggers happening at a higher rate than normal). These can help in assessing whether there is condition that may make the run not useful for analysis or that needs to be repaired or corrected.

Commissioning, problem detection and debugging: A problem that you overlook with PERIL in designing a trigger and DAQ is the need to be able to bring the system into being – to debug and commission it. For example, in BTeV, the pixel detector may be completed and installed later than many other detector components. You need to have other triggers! At the beginning, each group will be struggling just to understand its own device largely independent of others. The DAQ and Trigger must be "Partitionable" so that each subdetector will appear to have its own trigger, DAQ, and run control. Debugging and checkout might occur also with noise, cosmic rays, pulsers or simulated inputs. The system must be able to run even without beam (simulated clocks, etc)

Fault Tolerance and Fault Adaptation: The trigger is working on many beam crossings at once. To achieve high utilization of all processors, it makes decisions as quickly as possible. There is no fixed latency and events are not emerging in the same time ordered sequence with which they enter the system. Keeping the trigger system going and being sure it is making the right decisions is a very demanding problem -- 6000-12,000 processing elements: FPGAs, DSPs. Commercial LINUX processors. To make the system work, we require sophisticated fault tolerant, fault adaptive software. BTeV has been joined by a team of computer scientists who specialize in fault-tolerant computing under an award of $5M over 5 years from the US NSF. The project is called RTES, for "Real Time Embedded Systems." This topic is critical but too complex to go into here. Information about the BTeV project may be obtained from the RTES web site[13].

Relation to offline analysis – feed forward and feedback: Offline analysis needs "state information" from the trigger. Much of this can be about detailed detector performance since so much computation has been done. The trigger system can also provide initial alignments and calibrations. In the past, there was often a "rapid turnaround" offline system to provide the necessary feedback. For BTeV, and perhaps other experiments, this step may be unnecessary because it can be done at the trigger level. This information must also be stored in databases for trending, etc. Of course, if there are problems with the detector, the offline analysis is another opportunity to explore them and any conclusion about alignment, malfunctions, etc can be fed back into the trigger system or lead to a change in software, operational practice, etc. The first pass of offline analysis should be almost concurrent with data collection. Information from the offline must also be stored in databases for trending, etc.

Efficiency/System Evolution: Every trigger is always under pressure to perform. One needs to pay a huge amount of attention to efficiency. It is mission critical to run it fast enough. HOWEVER, one should not forget that these systems need to last for a long time. They should be designed, where possible, to conform to standards that are very likely to survive. Where compatible with efficiency, one should implement layers that hide parts that are likely to change behind interfaces that will permit you to bring in new programs, operating systems

5.0 Relation of Trigger/DAQ to more general problems

So far Trigger and DAQ systems in High Energy Physics look pretty specialized. But almost all the pieces are "commodity devices" and all are "programmable". In the BTeV trigger, for example, the only experiment "specific" part is where two data "substreams" – the pixel detector and the muon detector – are intercepted and routed to the Level 1 trigger and, of course, the algorithms themselves. However, lets abstract this by assuming that all data is first stored in the Level 1 buffer memory and we have instead " a data extractor" to get data into the processors of the various computer clusters at each level. In this generalized view, data may come from many sources, not just HEP experiments:

- From any kind of sensors, including those found in HEP, Nuclear Physics, Space Science, Astrophysics observations, earth science, or geology
- Communications streams, Data Mining on such data streams as EMAIL, Internet traffic, Written, graphic, verbal files, any large database
- Generated patterns, such a molecular sequences, which are to be matched
- and many others!

It obviously takes a tremendous amount of software to realize such a scheme: 1) A toolkit of parallelization software to split the computations among Levels and among many computers at each Level. THERE ARE SEVERAL TOOLKITS TO DO THIS; 2) A toolkit of fault tolerant, fault adaptive software to make sure all is working, data is not getting lost or miscalculated, there are no CPU or network problems. **THERE IS NOTHING AVAILABLE TO DO THIS AT THE SCALE OF THOUSANDS OF PROCESSORS!!!!; 3)** In real time cases, software to check that the apparatus is working, which is another class of fault. This must be handled at the application level but can use many of the elements of the toolkits above. **All this must be made to look "simple" to an operator or analyst**

6.0 Conclusions

In addition to telling you about recent trigger and data acquisition systems, I hope I have gotten you interested in actually working on them! Reasons why you should want to are: Triggers and DAQ are **absolutely crucial** to HEP experiments. These systems are **exciting** – they allow you to use some of today's most **cutting edge technologies** to do things that would have been science fiction only 10 years ago. The systems are now capable of doing almost offline analysis in quasi-realtime and **have lots of "physics content."** They require **new ideas and R&D** in databases, slow controls, fault tolerant computing, monitoring, and maybe self-diagnostic, fault-adaptive expert system type software. They may have, in their abstract view, **wide spread application. They are Way Cool!**

Acknowledgments

I would like to thank the BTeV Trigger Group for carrying out much of the work presented here. Erik Gottschalk, Michael Wang, and Harry Cheung provided material for the talk and I would also like to thank Erik for proofreading this paper. This work was in part supported by Fermi National Accelerator Laboratory, which is operated by Universities Research Association Inc. under Contract No. DE-AC02-76CH03000 with the United States Department of Energy.

References

[1] For a good description of trigger and data acquisition systems as they existed around 1990, see R.K. Bock, H. Grote, D. Notz, and M. Regler, *Data analysis techniques for high energy physics experiments*, chapter 1, Cambridge University Press, Cambridge, UK (1990)

[2] The BTeV home page is http://www-btev.fnal.gov and the proposal may be found at

---

http://www-btev.fnal.gov/public/hep/general/proposal/index.shtml under the link “BTeV Proposal Update”. The trigger and DAQ are described in chapter 4.

[3] CMS

[4] Information about the ATLAS experiment may be found at http://atlas.web.cern.ch/ATLAS/welcome.html

[5] Information on the CMS trigger and DAQ may be found at http://cmsdoc.cern.ch/cms/outreach/html/CMSdetectorInfo/CMStridas.html and the CMS trigger and DAQ Technical Design Report may be found at http://cmsdoc.cern.ch/cms/TDR/TRIGGER-public/trigger.html

[6] Key documents on the BTeV trigger may be

E.E. Gottschalk, BTeV Deteached vertex trigger, Nucl. Instrum. Meth. A 473 (2001) 167

R. Isik, Real-time pattern recognition for HEP, University of Pennsylvania, preprint UPR-233E, 1996

R. Isik, W. Selove, K. Sterner, Monte Carlo results for a secondary-vertex trigger with on-line tracking, University of Pennsylvania, preprint, UPR-234E, 1996

E.E. Gottschalk, The BTeV DAQ and trigger system-some throughput, usability, and fault tolerant aspects, Proc. CHEP 2001, Beijing, Sept. 3-7 2001, p628

[7] The average decision time must actually be a lot less than indicated, because otherwise local fluctuations would cause deadtime. This is also true at Levels 2 and 3.

[8] The ATLAS trigger and DAQ Technical design report may be found at http://atlas.web.cern.ch/ATLAS/GROUPS/DAQTRIG/SG/TP/tp_doc.html

[9] **Overview of the BTeV pixel detector.** By BTeV Collaboration (C.R. Newsom *for the collaboration*). 2000. Prepared for International Workshop on Seminconductor Pixel Detectors for Particles and X-Rays (PIXEL 2000), Genova, Italy, 5-8 Jun 2000, **Nucl.Instrum.Meth.A465:34-39,2001**

The BTeV pixel detector and trigger system. By BTeV Collaboration (M. Artuso *for the collaboration*). 2001. Prepared for Beauty-2000: 7th International Conference on B-Physics at Hadron Machines, Sea of Galilee, Kibbutz Maagan, Israel, 13-18 Sep 2000. Published in **Nucl.Instrum.Meth.A462:249-253,2001**

[10] GEANT: CERN Program Library, Long writeupW5013

[11] P. Avery et al., “ MCFAST: a Fast Simulation Package for Design Studies”, in the Proceedings of the International Conference on Computing in High Energy Physics, Berlin, 1997

[12] The trigger for the HERA-B experiment at DESY lab in Germany implements a four level trigger and is another excellent example of a modern trigger/DAQ system. A brief description may be found in “Architecture of the HERA-B Data Acquisition System”, submitted to IEEE Transactions on Nuclear Science.

[13] The RTES home page is http://www-btev.fnal.gov/public/hep/detector/rtes/index.shtml