
# Development of Workflow Planning Software and a Tracking study of the Decay $B^{\pm}$ to $J/\psi$ at the DØ Experiment

David Edward Evans *MPhys (Hons)*

Lancaster University, Department of Physics

A thesis submitted for the degree of

Doctor of Philosophy

September, 2003

# Development of Workflow Planning Software and a Tracking study of the Decay $B^{\pm}$ to $J/\psi$ at the DØ Experiment

David Edward Evans *MPhys (Hons)*

Lancaster University, Department of Physics.

A thesis submitted for the degree of

Doctor of Philosophy

September, 2003.

**Abstract**

A description of the development of the mc_runjob software package used to manage large scale computing tasks for the DØ Experiment at Fermilab is presented, along with a review of the Digital Front End Trigger electronics and the software used to control them. A tracking study is performed on detector data to determine that the DØ Experiment can detect charged B mesons, and that these results are in accordance with current results. B mesons are found by searching for the decay channel $B^{\pm} \rightarrow J/\psi K^{\pm}$.

# Contents

**Acknowledgements**

There are many people who helped me both during my PhD and before it. There are many people I would like to thank, colleagues, friends and family so here goes. Starting with the people I have worked with, Greg Graham, the author of the original mc_runjob package who it has been a pleasure to be able to work with again on the Shahkar project, and Guennadi Borissov for helping with the physics, without him, this thesis would be much more of a software manual than it already is. Other esteemed colleagues include Brian, Matt and Marilyn, fellow graduate students from Lancaster, and Roger Moore who kept me on a regular tea break schedule. It has been my pleasure to work and hoist ales with Rod "*Tommy Cooper*" Walker and Peter "*Paging Dr*" Love. I gained a lot of experience working with the DØ CTT group on the DFE project, especially Stefan, Jamieson and particularly Geoff Savage, who gave me great advice on using diagrams to complement text and had a direct impact on the style of this dissertation.

I must also single out my supervisor for some praise, Iain Bertam has been a great support and a great friend, and was especially sensitive about Australia's sporting dominance over Britain during the last few years. It has been a real pleasure to work with Iain, and I cant thank him enough for his assistance and advice through the last four years.

I have been particularly rewarded with great friends over the years, Jason and Sarah Copsey, Steve Clare, Peter Robson and the Ashill gang from Sixth

form, Lovejoy, Davey K and Cheese from the undergrad days. The guys from Iowa State, Andy, Graham, Dirty Pete and Lars.

Tim Blake. Now I have been involved with Lancaster nearly as long as you were, thanks for the appreciation of all things Cornish, especially the Blackbird song and Scrumpy.

Dr Chris and Dirty Newman. What can I say? Never a truer friend, never someone who could keep his pants on after a few beers. Ever since that first hung over week 1 undergraduate lab session, the path through university has been much greater for having staggered along it with you.

Matt Womack. Quite simply the best friend I have ever had. Quite strange to think that when we first got drunk it took a few pints, and last time we drank over 100 quid's worth of fine wine. Look at how far we have come!

My family have been a huge support to me throughout my life, and especially during university. Many weekends were spent by turning up at Rupe and Chris's house with a big bag of dirty laundry, which ended up clean and ironed as if by magic, Granny Evans bailed me out too many times to count, and Granny and Granddad always kept an eye out for me. The in-laws have also been a great help, notable mentions go to Angela (everyone should have a Pickle), Kate and Uncle Mark for introducing me to NASCAR.

Mum, Dad, Will and Rich, words cant express the love I feel for you all, so I'll simply say Thanks. ( I should probably also mention the cats, Marmalade and Mandy as well.)

And of course, my wife and love of my life, Sarah, who looks after me,

and keeps me sane. This is dedicated to you hunnybunny...

"Y'Know, its very hard to talk Quantum using a language originally designed to tell other monkeys where the ripe fruit is".

—Lu-Tze, History Monk. [1]

"On the first day, God created the Evans"

—Granny Evans.

*In Loving Memory,*

*Joyce Winifred Evans*

*$10^{th}$ April 1926 - $1^{st}$ February 2004*

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Elementary particle physics is concerned with discovering the nature of matter, its constituents and their interactions. Over the course of the last century, various objects have been discovered and studied, and theories to explain the nature of them formulated. As new results came to light, these theories were refined, altered or discarded until a consistent model was produced.

Over the last hundred years or so, particle physics has grown from the discovery of the electron, negotiated the quantum hedgerows in the theory of fields and been boosted by the laws of relativity, until taking its current form, sporting the unglamourous moniker of the *Standard Model*.

Whilst a great success, the Standard Model has to endure the daily strain of thousands of physicists attempting to find things wrong with it, and repeatedly pointing out its shortcomings. One of the largest of those shortcomings is the failure to include any mention of the force responsible for stopping those extremely discourteous physicists from floating off into space. As the-

orists invent improvements and explanations for all the unknowns, (and via the sheer untestable nature of some of them, guaranteeing them job security for life), experimentalists dream up exciting machines to probe these new ideas at the very edge of both technology and scientific funding.

One of the main requirements of these experiments is the ability to process vast amounts of data at high speed and apply very complex algorithms to find the particle physics equivalent of a needle in a needlestack.

Due to joining the DØ experiment in the middle of its upgrade cycle, the chance to do physics was somewhat limited, but there were plenty of opportunities to perform computational and electronics research, especially in the field of distributed computing, and so this thesis is a representation of the author's work at DØ for the past few years, a tasty combination of Workflow control software, a dash of detector systems management and integration and just a hint of physics.

## 1.1 The Standard Model

The Standard Model is the accepted description of present day particle physics, evolving from the quantum field theories of Glashow, Weinberg, and Salaam's Electroweak model describing electromagnetic and weak particle interactions and the theory of Quantum Chromodynamics that describes strong particle interactions.

### 1.1.1 The Elementary Particles

The elementary particles that we know today are summarised in Table 1.1 and are organised into three main groups:

- **Leptons**

  Leptons are spin-$\frac{1}{2}$ particles which experience weak interactions. There are three electrically charged leptons, the electron, muon and tau, and three uncharged leptons, the electron, muon and tau neutrino. For each lepton there is also a corresponding antiparticle with opposite quantum numbers.

- **Quarks**

  Quarks are spin-$\frac{1}{2}$ particles which experience strong, weak and electromagnetic interactions, and are not seen as isolated particles, instead only being found in bound states known as *Hadrons* due to the nature of the strong interaction. Quarks come in six different flavours named *up, down, charm, strange, top* and *bottom*, and each quark comes in three colours *red, green* and *blue*, which including antiparticles leads to 36 different quarks

- **Gauge Bosons**

  Interactions between quarks and leptons are mediated by the exchange of a gauge boson, between the interacting particles. Different particles mediate different interactions, with the photon being the mediator of electromagnetic interactions, the W and Z bosons mediating weak

interactions and the gluons mediating strong interactions.

Each of the three types of particle are, as far as we know, fundamental, that is they do not have any substructure that we can detect with current sensitivity.

### 1.1.2 The Fundamental Forces

As we understand it now, there are four fundamental forces, which govern interactions between matter: *Strong, Weak, Electromagnetic* and *Gravitational*. Each force is expressed as a physical quantum field theory, where the interactions are mediated by a boson exchange between the interacting particles.

- **Electromagnetic Force**

  Classically expressed as Maxwell's electrodynamics, the quantum field theory of the electrodynamic force is *Quantum Electrodynamics* or QED and explains the interactions of all particles interactions due to electric charge as the exchange of photons. Since the photon is massless, the range of the force can be very long.

- **Weak Force**

  The Weak force governs particle decays that change the type of particle, such as quark flavours or lepton types. Mediated by the massive W and Z bosons, the weak force has a short range, and is described by a field theory known as *Quantum Flavourdynamics*.

| **Leptons** | **Mass** | **Electric Charge** |
|---|---|---|
| Electron ($e^-$) | 0.5109 MeV | -1 |
| Muon ($\mu^-$) | 105.6583 MeV | -1 |
| Tau ($\tau^-$) | 1776.99 MeV | -1 |
| Electron Neutrino ($\nu_e$) | $< 3$ eV | 0 |
| Muon Neutrino ($\nu_\mu$) | $< 0.19$ MeV | 0 |
| Tau Neutrino ($\nu_\tau$) | $< 18.2$ MeV | 0 |
| **Quarks** | **Mass** | **Electric Charge** |
| Up ($u$) | 1.5 to 4.5 MeV | $+\frac{2}{3}$ |
| Down ($d$) | 5.0 to 8.5 MeV | $-\frac{1}{3}$ |
| Strange ($s$) | 80 to 155 MeV | $-\frac{1}{3}$ |
| Charm ($c$) | 1.0 to 1.4 GeV | $+\frac{2}{3}$ |
| Bottom ($b$) | 4.0 to 4.5 GeV | $-\frac{1}{3}$ |
| Top ($t$) | 174.3 GeV | $+\frac{2}{3}$ |
| **Gauge Bosons** | **Mass** | **Electric Charge** |
| Photon ($\gamma$) | $< 2 \times 10^{-16}$ eV | 0 |
| Gluon ($g$) | $0^*$ | 0 |
| W Boson ($W^\pm$) | 80.423 GeV | $\pm 1$ |
| Z Boson ($Z^0$) | 91.187 GeV | 0 |

Table 1.1: Table of fundamental particles giving current values for mass and electric charge. Antiparticles exist for all quarks and leptons and have opposite quantum numbers. All charges are in units of proton charge. Each quark flavour exists in three distinct colour charges. All values are from the Particle Data Group.[2] (* The Gluon mass is usually taken to be zero based on theory.)

- **Strong Force**

  The Strong force is responsible for interactions between particles carrying colour charge, which means it is responsible for binding quarks into hadrons. The strong force is very short range, but very strong over that short range, and hence the name. Strong interactions are mediated by the exchange of gluons and is described in the quantum field theory of *Quantum Chromodynamics* or QCD.

- **Gravitational Force**

  Gravity, which is classically explained by Einstein's General Relativity is responsible for interactions between massive bodies. To date there is no correct formulation of a quantum theory of gravity, explaining gravitational interactions as the exchange of a graviton boson. Gravity is generally disregarded in particle physics due to the low mass of the elementary particles, although its effects may still have some undiscovered effects.

Of these four forces, the Strong, Weak and Electromagnetic force quantum field theories combine to make up the Standard Model of particle physics, with gravity being left out in the cold.

### 1.1.3 Lagrangians and Gauge Invariance

The Lagrangian formalism of classical mechanics allowed the equations of motion for a system to be expressed as the difference of kinetic and potential energy. Quantum field theory also uses the Lagrangian to express the

equation of motion for the quantum fields representing the elementary particles. When formulated in this way, transformations of the particle fields that do not change the structure of the Lagrangian correspond to symmetries, these symmetries correspond to conserved quantities. Symmetries can be expressed in terms of mathematical groups, of which there are many types, so transformations of the fields can be based on a particular symmetry group. One particular type of transformation is called a local gauge transformation, which is a function of space-time position and based on a symmetry group. When the Lagrangian is required to be invariant under local gauge transformations of a particular group, it requires that extra gauge fields be added to the Lagrangian, which correspond to the gauge boson fields that mediate the interaction. There are three groups used in the Standard Model, the one dimensional group U(1) gives rise to the photon, the two dimensional group SU(2) gives rise to the W and Z bosons of the weak force, and the three dimensional group SU(3) produces the terms corresponding to the gluon fields. The SU(2) and U(1) theories are combined together in the Glashow-Weinberg-Salaam model to produce the electroweak model which treats electromagnetism and the weak force as a single theory. Thus the Standard Model is a gauge theory of the strong and electroweak field theories, and the elementary particles are grouped into multiplets in three generations and their interactions explained by the quantum field theories.

### 1.1.4 The Future of the Standard Model

While the Standard Model does fit with all experimental evidence to date, it is not complete, it contains the Higgs mechanism to give particles physical mass, and requires the predicted, but as yet undiscovered Higgs Boson. The Standard Model also fails to predict the values of the quark masses and offer any reason why the top quark should be so much heavier that the other quarks. It fails to address why there should be only three generations of particles, and it doesn't include gravity at all. It is anticipated that the Standard Model will begin to show signs of breaking down at energies larger than the electroweak scale, which should begin to appear at future accelerators, such as the Large Hadron Collider in Geneva. So there are many open questions left by the Standard Model, and this is what keeps people interested, as it clearly isn't the final theory that describes what makes the Universe tick.

## 1.2 Particle Physics Experiments

To study the nature of particles one first needs to produce them, and to look as deeply inside the objects as possible, requires the highest available energy. Following this idea most modern machines are very large and involve accelerating colliding beams to the highest possible energy to create the most energetic environment to produce particles that ordinarily cannot exist at everyday energies. These particles then decay and reveal insights into their properties. At Fermilab the Tevatron accelerates beams of protons and antiprotons to 1 TeV each, and collisions are produced in two experimen-

tal detectors named DØ and CDF. The Tevatron beam collisions currently create the highest energy particle environment on the planet on a very small scale, allowing a glimpse of the high energy behaviour of particles that is not normally observed in everyday environments. The DØ experiment is designed to catch the debris from the collisions and use it to infer what happened in the high energy collision. There are many different fields of study available at DØ , ranging from searches for new undiscovered physics, including the prophesied Higgs particle, to tests of QCD and detailed physics studies of the nature of the bottom quark which is produced in abundance at the experiment.

## 1.3 Workflow Planning Software

With DØ reading out over 2 million channels of digital signal every time the beams collide, which is in the time range of 400 to 100 nanoseconds, a lot of information is read out from the experiment. To convert those masses of binary data into a detailed physics analysis takes a lot of very intense computing power. In addition to the detector data, many simulation studies are required to make sure that we know what to expect from the detector, this all adds up to storing millions and millions of data events and simulated events, and processing them. There is a staggering level of detail involved in coordinating the processing of this data, indeed, the task is so large that it must be performed at many different computer installations around the world, so not only must the data be stored, catalogued and analysed, it

must be distributed around the globe. This leads to the concept of workflow planning and metadata cataloging to automate the complex processing tasks across the globe whilst recording all the details of what happens to the data so that it is well understood when it becomes part of the physics results. The major part of the authors work during the period of work described in this thesis has been to enable data processing tasks to be expressed as workflows and distributed to the various processing resources around the world.

## 1.4 Scope of this thesis

This thesis provides a record of the work performed by the author over the last four years, on the commissioning and software development of the DØ experiment for its Run II period of operations. This includes the development of control and monitoring software for the DØ Triggering electronics, and the development of the mc_runjob Workflow management software. While there was not a great opportunity to do physics during the time period, a study of the track finding algorithms was performed on data to examine the abilities of the DØ detector to find bottom quark hadrons and verify the results from the new detector systems.

# Chapter 2

# Theory

## 2.1 Introduction

This section provides an overview of the theoretical aspects involved in bottom quark (b) production in $p\overline{p}$ collisions at DØ . A brief overview of the theory of the strong force is given, along with a discussion of the main production processes of B mesons. Particular features of the strong force such as fragmentation and asymptotic freedom are introduced and the decay of the charged B mesons via the $J/\psi K^{\pm}$ channel is also discussed.

### 2.1.1 Conventions and Definitions

This section briefly describes some of the quantities and conventions that are commonly used by DØ . Particle physics in general uses the terms Luminosity and Cross Section, very frequently, so a brief description of these terms is

provided here:

- **Luminosity** Luminosity ($\mathcal{L}$)is the quantity that describes the flux of particles incident on a target in a collision, for a colliding beam experiment, this is proportional to the squared number of particles passing through a unit area per unit time. Typical units for $\mathcal{L}$ are $cm^{-2}s^{-1}$. When integrated over a period of time the luminosity is referred to as the integrated luminosity. Luminosity is affected by factors such as beam focus, the number of particles contained in the beams and the energy of those particles. The Tevatron Luminosity is measured for DØ by the Luminosity Monitor detector as described in §3.4.

- **Cross Section** Cross Section ($\sigma$) is the reaction dependent part of the rate of reaction for a specific process. The reaction dependent part contains the effect of the forces and particles that are involved in the interaction along with the interaction kinematics. The rate for a specific process can be written as $R = \mathcal{L}\sigma$. $\sigma$ has units of area, and are usually expressed in *barns*, where $1\,\text{barn} = 10^{-24}\,cm^2$. The Luminosity is independent of the process, all the process dependent information is contained within the Cross Section. The number of events is determined by multiplying the integrated luminosity with cross section.

The coordinate system in use at DØ is a cylindrical $(r, \phi, z)$ system, with the origin taken to be the nominal interaction point at the centre of the detector, $(x, y, z)$=(0,0,0). The polar z-axis is defined by the direction of the proton beam, with the direction of the y-axis defined to be straight upwards.

Spherical $(r, \theta, \phi)$ coordinates are also sometimes used, but the polar angle theta is usually expressed as *pseudo-rapidity* $(\eta)$. True rapidity $(y)$ is a Lorentz Invariant quantity defined as:

$$y = \frac{1}{2} ln \left( \frac{E + P_z}{E - P_z} \right) \tag{2.1}$$

Pseudo-rapidity is equal to the true rapidity in the case where the mass of a particle is very much less than its energy. Pseudo-rapidity can be expressed in terms of $\theta$:

$$\eta = -ln \tan \left( \frac{\theta}{2} \right) \tag{2.2}$$

Also commonly used at DØ is the transverse momentum $(p_t)$ and transverse energy $(E_T)$ relative to the z-axis. These are defined as:

$$p_t = p \sin \theta \quad E_T = E \sin \theta \tag{2.3}$$

Energy is expressed in units of electron volts (eV), and maybe expressed in terms of Mega Electron volts ($10^6 eV$ or MeV), Giga Electron Volts ($10^9 eV$ or GeV) and Tera Electron Volts ($10^{12}$ or TeV).

## 2.2 Quantum Chromodynamics

Quantum Chromodynamics (QCD) is a Gauge field theory based on the three dimensional special unitary symmetry group SU(3) and governs the interaction of particles carrying QCD colour charge, the quarks and gluons of the standard model. Requiring the quark field lagrangian be invariant

under local gauge transformations based on the SU(3) group leads to the introduction of eight gauge fields which correspond to eight gauge bosons, the gluons. The Lagrangian for QCD is given in Eq. 2.4 [3].

$$\mathcal{L}_{QCD} = -\frac{1}{4}F^{(a)}_{\mu\nu}F^{(a)\mu\nu} + i\sum_q \overline{\psi}^i_q \gamma^\mu \left(D_\mu\right)_{ij} \psi^j_q - \sum_q m_q \overline{\psi}^i_q \psi_{qi} \tag{2.4}$$

$$F^{(a)}_{\mu\nu} = \partial_\mu A^a_\nu - \partial_\mu A^a_\mu - g_s f_{abc} A^b_\mu A^c_\nu \tag{2.5}$$

$$\left(D_\mu\right)_{ij} = \delta_{ij}\partial_\mu + i g_s \sum_a \frac{\lambda^a_{i,j}}{2} A^a_\mu \tag{2.6}$$

where $g_s$ is the strong coupling factor and $f_{abc}$ are the structure constants of the SU(3) group algebra. The $\psi^i_\mu(x)$ are 4 component Dirac spinors associated with the quark fields of flavour q, and colour i (conventionally labeled red, green and blue). The $\lambda^a_{i,j}$ are the eight three dimensional matrices representing the SU(3) Group. The $A^a_\mu$ are the eight Yang-Mills fields associated with the eight gluons, which mediate interactions between particles with colour charge.

The nature of the SU(3) group is such that its components do not commute (the group is said to be *non-abelian*) leading to the $f_{abc}$ term in the QCD field tensor. (Eq. 2.5) This term gives rise to trilinear and quadrilinear terms in the QCD lagrangian corresponding to the three and four point gluon interactions illustrated in Fig. 2.1 [4]. This also means that the gluons themselves actually carry colour charge, and as such these gluon self interaction terms give rise to some interesting features of QCD, these being the apparent strength of the strong force at large distances or *Confinement*, and its apparent weakness at very short distances or *Asymptotic Freedom*.

One of the major impacts of the gluon self interactions, is on the strong force coupling strength, $\alpha_s$ , which becomes large at low energies.

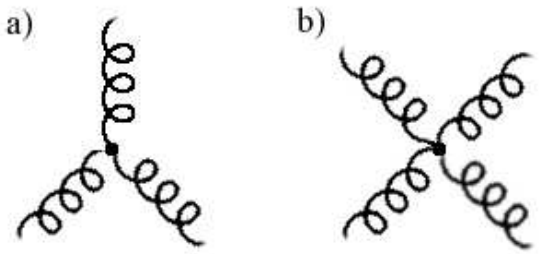


Figure 2.1: Feynman diagrams representing the trilinear (a) and quadrilinear (b) gluon self interaction terms of the QCD lagrangian, which arise from the non abelian nature of the underlying SU(3) Gauge group.

QCD Quantities are usually calculated as a series of increasingly complex Feynman diagrams starting with the lowest order in the strong coupling factor $\alpha_s$ and working upwards. For example a typical QCD cross section, given by:

$$\sigma = A_2\alpha_s^2 + A_3\alpha_s^3 + A_4\alpha_s^4 + ... \tag{2.7}$$

The $A_i$ factors are calculated from evaluating Feynman diagrams which contain the same number of vertices (i.e. same order in $\alpha_s$ ). Some example Feynman diagrams illustrating the Leading Order (LO) $gg \rightarrow qq$ processes can be seen in Fig. 2.2 and some of the Next-to-Leading Order (NLO) diagrams can be seen in Fig. 2.3. The total cross section is a sum over all

orders of $\alpha_s$ . If $\alpha_s$ is small then there is only a small contribution to the cross section from the higher order terms, with the lower level processes carrying the bulk of the interactions. However, if $\alpha_s$ is large, then the higher order corrections have more and more of an impact on the final cross section. At low energies, $\alpha_s$ becomes large enough that perturbation theory cannot be applied, since it requires that $\alpha_s$ be much less than 1. This means that many low energy QCD phenomenon are very difficult to calculate exactly without using some very complicated theories. So while jets produced from high energy collisions are a good probe of the details of QCD, due to the high energy nature of the collision and, hence small values of $\alpha_s$ , the process of hadronisation within the jets occurs at a much lower energy and hence, high values of $\alpha_s$ , making exact theoretical predictions of the hadronisation process very difficult. Indeed, many of these situations use numerical approximations that are tuned to experimental measurements, so low energy situations such as hadronisation and modeling of the internal structure of the proton are done using models such as the Pythia string hadronisation model and the CTEQ [5] Proton Structure functions.

### 2.2.1 Asymptotic Freedom

A colour charge induces an affect on the space surrounding it, by stimulating vacuum QCD processes in effect creating a "sea" of virtual particles around the charge which have an effect on how it is seen. However the virtual gluons themselves carry colour, and therefore do not leave the colour charge

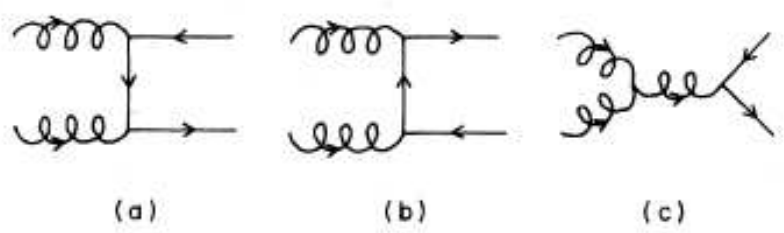


Figure 2.2: Lowest order diagrams for $gg \rightarrow qq$ processes, note the contribution from the trilinear gluon coupling (c). The (a) and (b) diagrams have an analogue in QED, the (c) term is a feature of QCD

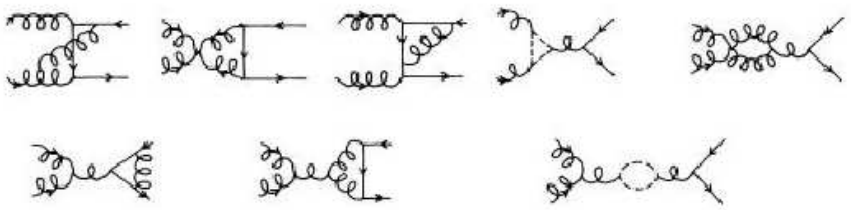

Figure 2.3: A subset of the Next to Leading Order diagrams contributing to $gg \rightarrow qq$ processes. Many extra terms arise from the gluon self interactions compared to the analogous QED process. Dotted lines represent virtual quark loops.

unchanged by these interactions, instead, charge is radiated away into the virtual sea surrounding the charge, which means that at short distances the charge itself cannot be seen in its entirety due to the screening effect of the virtual gluon field, but at larger distances the true charge of the colour charge is seen as the sum total of the surrounding cloud. In effect, this means that as the distance between charges decreases, the effective coupling strength of the force weakens, allowing particles in close proximity to move as if not bound by the force. Thus within bound quark states, there are effectively free valence quarks surrounded by a sea of virtual gluons and quarks, which account for a large percentage of the energy of the hadron.

### 2.2.2 Confinement

Comparing the virtual clouding of charge in QED (abelian U(1) group, with a charge neutral boson, and hence no gauge field self interactions) and QCD one sees that the screening effect of the cloud at long distance reduces the effective charge in the QED case, but amplifies the charge in the QCD case due to the self interactions of the gauge field (as illustrated earlier in Fig. 2.1). This implies that the strong force actually increases with distance. Hence, quarks and gluons exist only in colour neutral hadron bound states and we do not see isolated quarks and gluons. As quarks are moved apart, more and more energy is required to separate them, and eventually when enough energy is applied, pair production occurs creating two separate colourless bound states, this process is known as fragmentation, and is responsible

for the evolution of hadronic jets. A cartoon illustrating the fragmentation process is shown in Fig. 2.4.

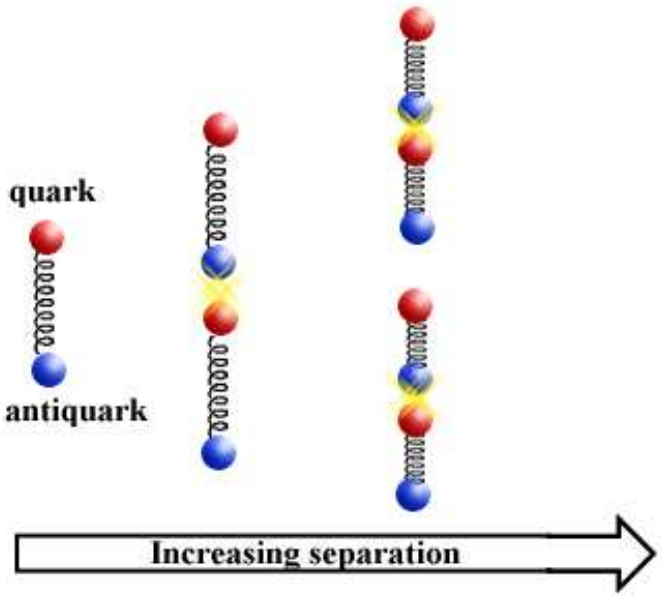


Figure 2.4: Illustration of the fragmentation of a quark-antiquark pair. As the original pair are separated, the strong coupling between them increases requiring more and more energy. Eventually enough energy is applied to pair produce an new quark-antiquark pair. Thus, the quarks remain in bound colourless states. If the pair are produced at high enough energy, moving apart, thus process will be repeated many times over leading to the evolution of a hadron jet.

### 2.2.3 Structure of the Proton

At the Tevatron, the protons (and antiprotons) in the collision possess a large amount of energy and this means that they have a complicated structure that must be taken into account. The proton (antiproton) consists of three valence quarks (antiquarks), those being two up quarks and a down quark. However, these three quarks themselves do not make up the proton by themselves, as they interact with each other via gluon exchange within the proton. In addition, the gluons themselves can pair produce quark-antiquark pairs, which since they are not on-shell, can be of any type. Thus the proton is a boiling sea of QCD processes that, if probed deeply enough, there is a chance that they may interact. QCD confinement keeps all of this within the proton, but at very small distances within the proton, asymptotic freedom between quarks and the anti-screening effects of QCD charge described in §2.2.1 occur, and provide a contribution to the constituent mass of the proton. As energy increases the virtual processes within the proton become more and more prevalent, and more and more of the proton energy is contained by gluons within the proton, as illustrated in Fig. 2.5. The proton structure plays an important part in the theory of interactions, since one must attempt to understand which partons inside the colliding hadrons interact, essentially boiling down to a probability distribution for an interaction to occur between certain constituents. In addition, interactions between other hadron constituents may also occur, which have an affect. Exact calculation of these effects is extremely difficult, so various models for the structure of

the proton are used, which express the proton structure as a set of structure functions which are determined using deep inelastic scattering [16] experimental results combined with theory to produce tuned models of hadron structure such as those provided by the CTEQ Collaboration [5]. The overall effect of this at DØ is that the colliding hadrons have enough energy that interactions between virtual partons within the proton have a large effect on the cross section for a process. The cross section for a process occurring from a $p\overline{p}$ collision can be expressed in terms of the Parton Model, for a collision between two hadrons with momenta $P_1$ and $P_2$ can be expressed as:

$$\sigma\left(P_1, P_2\right) = \sum_{i,j} \int dx_i dx_j f_i\left(x_1, \mu\right) f_j\left(x_2, \mu\right) \hat{\sigma}_{ij}\left(p_1, p_2, \alpha_s\left(\mu\right), Q\right) \qquad (2.8)$$

where the momentum of the hadronic constituents, or partons which actually interact, $p_1$ and $p_2$ can be expressed as a fraction of the momentum of the parent hadron: $p_{1,2} = x_{1,2}P_{1,2}$ where $x_{1,2}$ is the fraction of the parent hadron's momentum carried by the interacting parton in the interaction. Q is the characteristic energy scale of the process, for example the mass of the b quark and the $\hat{\sigma}_{ij}$ is the cross section for the partonic interaction between partons $i$ and $j$. The functions $f_i$ represent the structure of the colliding hadrons, and the momentum distribution among the constituent partons, and thus, underline the importance of understanding the internal structure of the hadrons, since it has a direct impact on the cross section.

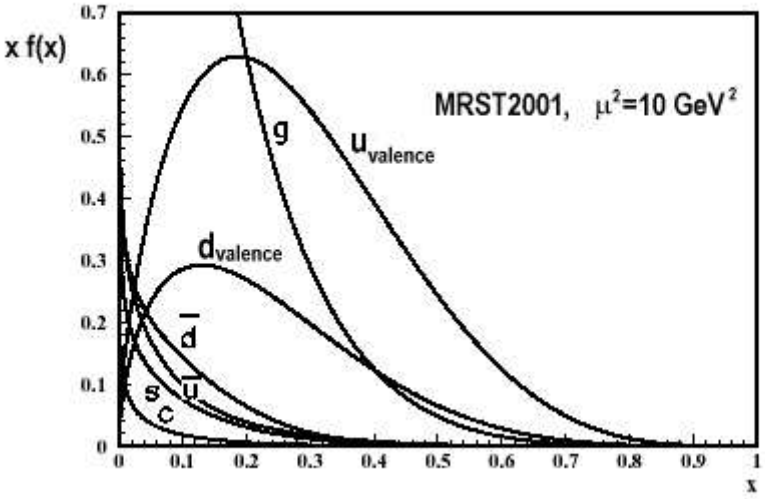


Figure 2.5: Plot showing the distributions of the partons within the proton $f(x)$, where $f = u_{valence}, d_{valence}, \overline{u}, \overline{d}, s, c, g$ calculated using the MRST2001 parametrization. This plot illustrates the fraction of momentum of the proton carried by its parton constituents, including the three valence quarks, the virtual quarks, and the virtual gluons. [3]

## 2.3 Bottom Production at DØ

There are three important processes contributing to the production of bottom quarks at the Tevatron [6][7].

- **Creation** b-Quarks are created by the interaction between the proton and antiproton as they interact. At Tevatron energies the primary source of this process is gluon-gluon fusion between sea gluons contained within the hadrons, $gg \rightarrow b\overline{b}$. See illustration in Fig. 2.6a.

- **Excitation** b-Quarks already existing within the sea of virtual partons inside the proton or antiproton are excited and emitted by the $p\overline{p}$ interaction, $bg \rightarrow gb$ . See illustration in Fig. 2.6b.

- **Fragmentation/Showering** As hadronisation occurs from $p\overline{p}$ interactions, b-Quarks are produced in the hadronic showers from the products of a $p\overline{p}$ interaction. See illustration in Fig. 2.6c.

The relative contributions of these three mechanisms to the integrated b-quark total cross section are shown in Fig. 2.7

Given the high energy of the $p\overline{p}$ collisions at DØ , the partons within the colliding hadrons can be treated as being almost free, allowing the interaction $p\overline{p}$ interaction to be reduced to a parton-parton interaction. It is then possible to factorize out this part of the reaction from the non-perturbative parts, like fragmentation, allowing perturbative QCD to be used to calculate the bottom quark production cross-section as an expansion in terms of $\alpha_s$ . The leading

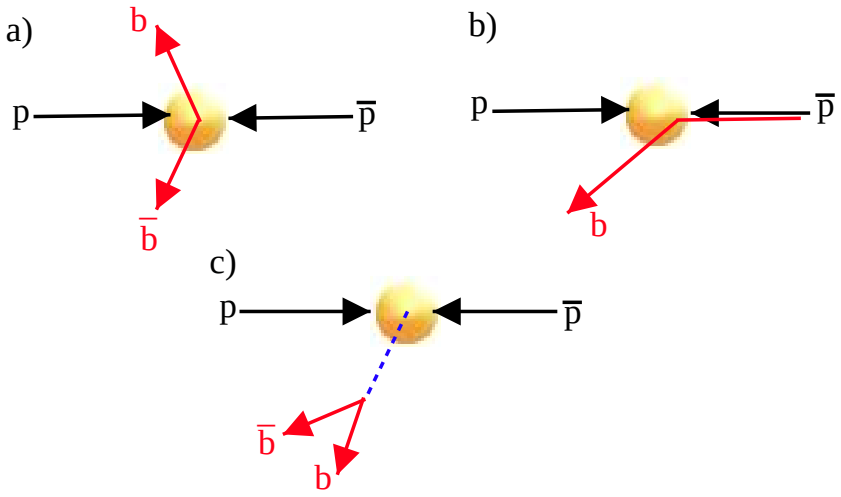


Figure 2.6: Illustration of the three main production processes for b-quarks at the Tevatron. Creation of b-quarks during the $p\overline{p}$ interaction (a), Excitation of a b-quark within the proton or antiproton (b) and creation of a b-quark during the hadronic decay of some other product of the initial $p\overline{p}$ interaction.

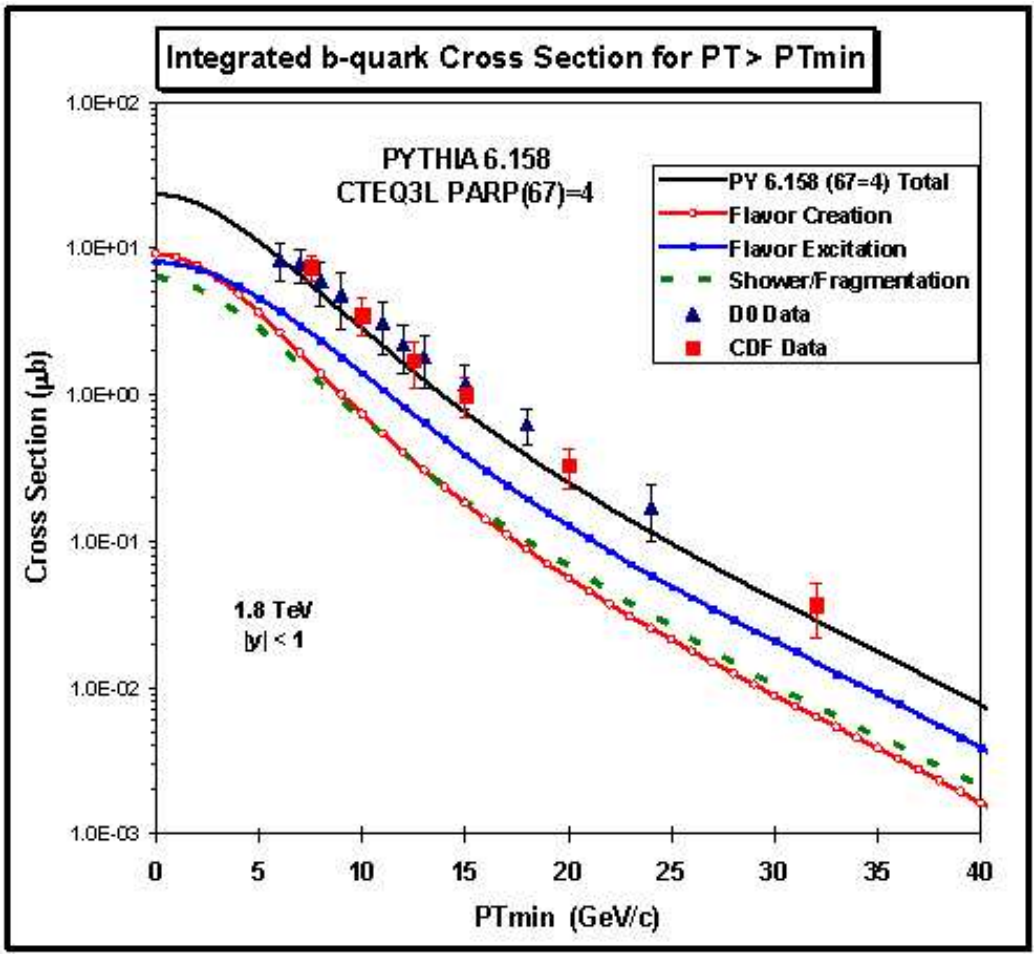


Figure 2.7: Plot of the Integrated b-quark production cross section vs. $p_T$ threshold showing the relative contributions from creation (red), excitation(blue) and fragmentation(green dotted) compared to the prediction of the Pythia MC program and the Run I data from the DØ and CDF Experiments. [7]

order terms that contribute to this are $q\overline{q} \rightarrow b\overline{b}$ and $gg \rightarrow b\overline{b}$, with next-to-leading order contributing terms like $qg \rightarrow gb\overline{b}$. The leading order processes are observed as two back-to-back jets, where the higher order processes result in three non-collinear jets. Measurements of the bottom quark production cross section [8]-[11] show that the measured cross section for the process

$$p + \overline{p} \longrightarrow b + \overline{b} + X$$

lies above the predictions made by $O(\alpha_s^3)$perturbation theory [12][13][14], by a factor of $2.5 \pm 0.4$ in the region $|\eta| < 0.8$ and a factor of $3.6 \pm 0.8$ in the region $2.4 < |\eta| < 3.2$ [15]. Which may indicate the need for better understanding of the theory or perhaps hint at some new physics. Current experiment and theory values are summarised in §2.5. After production of bottom quarks through the methods detailed above, they hadronise to form B mesons and baryons with further decay, leaving signatures such as displaced secondary vertices. The various branching ratios for bottom quarks to form the various hadrons have been measured [19] and the branching fraction to form $B^{\pm}$ mesons relative to other types of B hadrons are given as $B^{\pm} : B_d^0 : B_s^0 :$ B baryons

$$0.375 \pm 0.023 \ : \ 0.375 \pm 0.023 \ : \ 0.160 \pm 0.044 \ : \ 0.090 \pm 0.029 \qquad (2.9)$$

## 2.4 Decay of the Charged B Meson

The mode of decay studied in this thesis involves the decay of the $B^+$ ($B^-$) consisting of a $u$ and $\overline{b}$ ($\overline{u}$ and $b$) quark pair to the $J/\psi$ meson and $K^+$ ($K^-$) pair, as shown in Fig. 2.8. This decay occurs via a weak charged current interaction [4], as the bottom quark in the meson decays via a $W^{\pm}$ boson. This decay occurs at a relatively low energy, meaning that perturbative QCD is not the best tool to use, since the higher order contributions to these decays provide a large contribution, so a different approach is needed. One approach is that of Heavy Quark Effective Theory (HQET) [18], which expands the lagrangian into a series using powers of $1/m_b$, the inverse mass of the b-quark. Since the inverse mass (also called the Compton wavelength) is much smaller than the hadronic distance, the actual mass of the quark is not important for the low energy interactions within the state, allowing strong interactions between the heavy quark and light quarks and gluons to be described by an effective field theory. In the limit where $1/m_b \to 0$ the theory describes the leading order decay where the b-quark decays as a free quark. HQET can be applied relatively successfully to both b and c-quark hadron decays, and so can also be used to model the decay of the $J/\psi$ meson. However due to the complicated nature of higher order corrections, a variety of factorization schemes are used and combined in an attempt to produce an accurate model for the decays, however the theoretical uncertainties in calculating the matrix elements of these hadronic decays are much larger than the experimental errors.

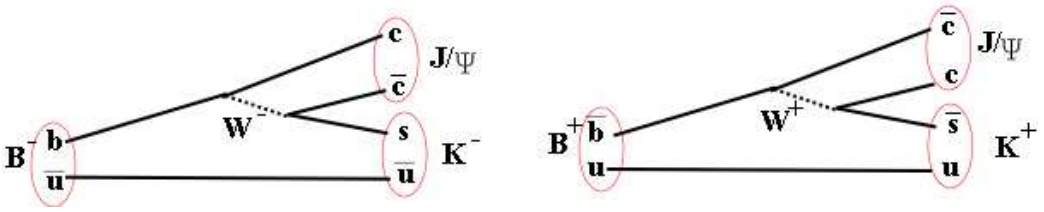


Figure 2.8: Feynman diagrams showing the decay of the charged B mesons to $J/\psi\, K$ via a charged weak current interaction.

## 2.5 Prospects for study of B Physics at DØ

The non-perturbative nature of QCD processes in low energy situations such as within the proton and fragmentation, makes analytical theoretical solutions unlikely in the near future, however the high energy nature of the collisions do allow us to test certain parts of the theory. Production of bottom quarks at DØ in Run II will be high, with a $b\overline{b}$ cross section estimated to be $\sigma\left(p\overline{p} \rightarrow b\overline{b}\right)$ expected to be approximately $150\mu b$ at 2 TeV [20]. This rate is significantly larger than $e^+e^-$ B Factory experiments and produces many species of bottom quark hadrons, allowing a full program of B physics studies to be pursued. In addition to the wide variety of B Physics, accurate identification of bottom quarks is vital for other physics such as top physics, new phenomena and Higgs searches which feature many models involving decay via bottom quarks. The current world average measurement for the mass of the $B^+$ meson is given by the Particle Data Group as $5279.1 \pm 0.5$ MeV [2].

# Chapter 3

# Experimental Apparatus

## 3.1 Introduction

In this section an overview of the Tevatron collider as used for the Run II period and the DØ experimental hardware, with an emphasis on the new features of the detector for Run II is presented.

## 3.2 The Tevatron

The Tevatron is a circular accelerator built at Fermilab, on the outskirts of Chicago, USA, and was first used as a proton anti-proton ($p\overline{p}$) collider in 1985 providing 900 GeV per beam to the CDF Collider Experiment. Later on in 1992, a second collider experiment was added, the DØ Experiment. During the Run I period of 1993 to 1995 the Tevatron Experiments made significant advances in the field of particle physics, culminating in the discovery of the

top quark particle. Recently the Tevatron has received a major upgrade to increase beam energy and luminosity. This upgrade included:

- **Main Injector** The Main Injector is a new machine used to supply protons and antiprotons to the Tevatron accelerator at an energy of 150 GeV.

- **Antiproton Recycler** Sharing the Main Injector tunnel, the Recycler is used to store large amounts of antiprotons and recover unused antiprotons from the Tevatron after a beam store, to increase the density of antiprotons, thus increasing the luminosity.

- **Antiproton Production Upgrade** Antiprotons are produced by firing 120 GeV protons at a nickel target which are extracted by the Debuncher, and the Accumulator which combine to provide a concentrated 8 GeV antiproton beam.

- **Tevatron Upgrade** The Tevatron itself has been upgraded, to accept the proton and antiproton beams from the Main Injector, for final acceleration within the 4 Tesla magnetic fields of the 4 mile circumference Tevatron, to produce beams of 1 TeV in energy.

The plan for Run II is to run in a few stages, steadily ramping up the number of interactions and Luminosity. The initial parameters for Run II operation are to run with 36 bunches of protons and antiprotons per beam, with a bunch spacing of 396 nano seconds. A typical Run II Luminosity is expected to be $0.86 \times 10^{32} cm^{-2} s^{-1}$.

## 3.3 Overview of the DØ Detector

The DØ detector was originally approved in 1983 as an experiment to study phenomenon arising from $p\overline{p}$ collisions in the Tevatron. The main focus of the detector was to examine high mass states and high $p_T$ interactions, including the search for the then undiscovered top quark, precision studies of the W and Z boson, mediators of the weak force, an examination of bottom quark production and QCD and the search for new phenomena. The detector was optimised for the detection of final states containing jets, electrons, muons and neutrinos (via inference of missing transverse energy) arising from a variety of processes. The original DØ detector did not contain a solenoid magnet for measuring the charge and momentum of particles, however it did possess a high precision calorimeter and very wide angle muon coverage [22]. With the upgrade of the Tevatron to provide higher luminosity and energies at very short intervals, the DØ detector was also upgraded to handle the increased collision rate and add new systems to take better advantage of this wealth of collision data. The upgrade to the DØ detector is intended to make use of the calorimetry and muon system from Run I and augment them with an improved tracking system including a 2T solenoid magnet and new silicon and scintillating fibre track detectors for measurement of charged tracks [23][24]. Additionally the triggering system has been upgraded to operate within the much shorter 396 ns time between interactions as well as exploiting the new tracking system to make fast trigger decisions. The trigger is designed to handle a bunch crossing interval of up to 132 ns. A

diagram showing a cutaway view of the entire detector for Run II can be seen in Fig. 3.1. The various subsystems of the detector are described in the rest of this chapter.

## 3.4 Luminosity Monitor

Measurement of the beam luminosity delivered by the Tevatron is performed using a pair of scintillator array detectors called the Luminosity Monitor. In addition to measuring the beam luminosity, this device also functions as a Level 0 trigger to flag beam crossings and multiple interactions. Located between the beam pipe and the Forward Preshower Detector (described in §3.5.4) the Luminosity Monitor is a series of 24 wedge shaped 1.6 cm thick scintillator detectors that covers the $2.7 \leq |\eta| \leq 4.4$ region around the beam pipe. Two of these arrays, illustrated in Fig. 3.2, are arranged at either end of the detector, at $\pm 135$ cm from the centre of the DØ detector as shown in Fig. 3.3, with the centre of the arrays being 3.3 cm from the beam pipe. High gain photo multiplier tubes are mounted on each wedge and are read out through a digitizing electronic system that calculates the luminosity coincidence, beam spot and halo and also provides a flag for multiple interactions during a single bunch crossing. The input of the Luminosity Monitor into the DØ trigger system is shown in Fig. 3.17. The measurement of luminosity is made by comparing the number of hits in the luminosity monitor to a threshold value calculated from simulation. When an interaction occurs, the remnants of the collision travel at high $\eta$, close to the beam pipe and impact the luminosity

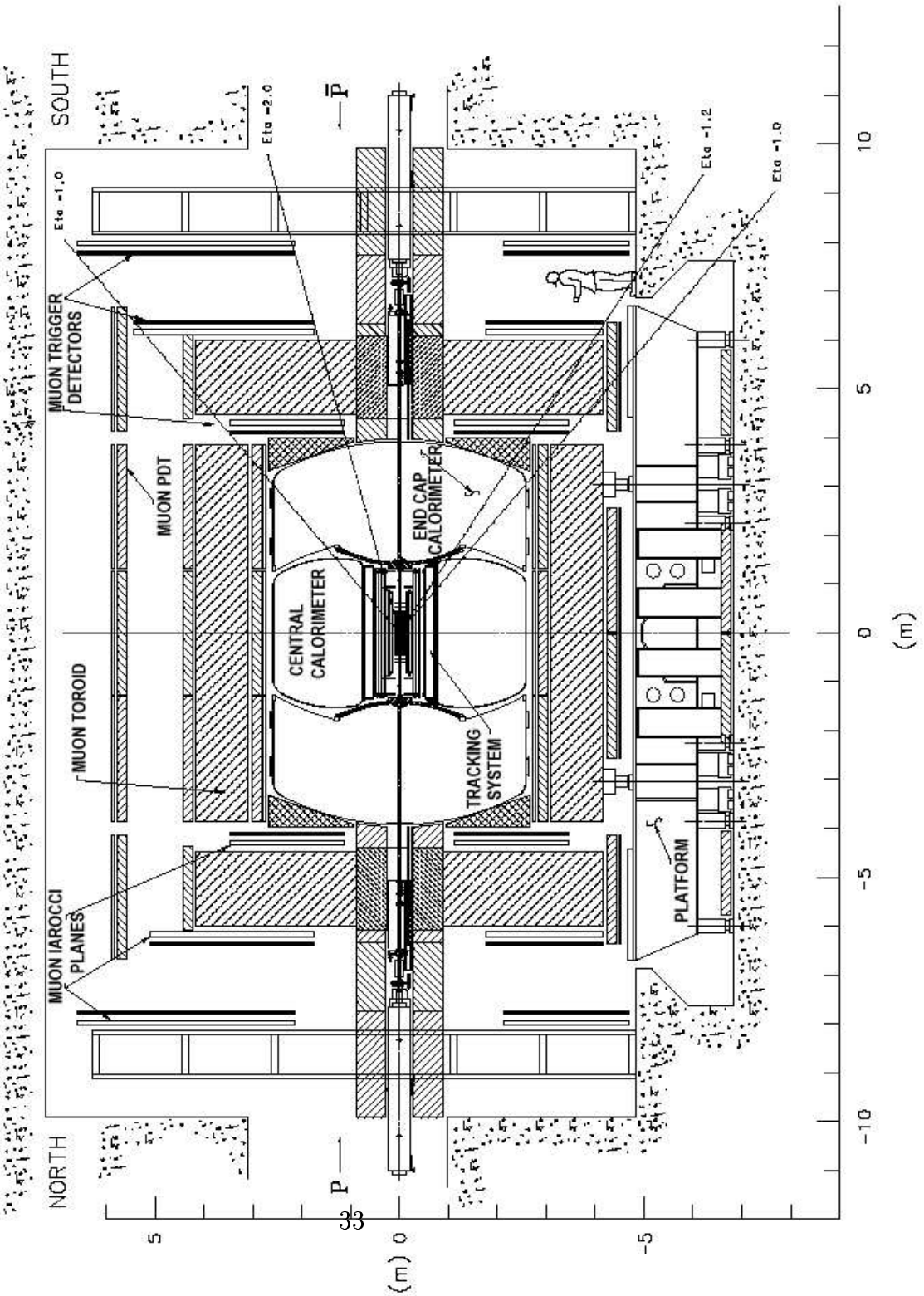


Figure 3.1: Cutaway Diagram of the DØ Detector.

monitor, thus causing many hits. Using simulations to calculate the response of the detector to elastic collisions (or near misses), the threshold for the number of hits can be determined. The counting rate from the luminosity detector can then be used with the world average cross section measurement for elastic $p\overline{p}$ events to determine the luminosity, via the optical theorem. The optical theorem relates the forward elastic scattering amplitude to the total cross section for the interaction by the relation:

$$Imf(\theta) = \frac{Q\sigma_{Total}}{4\pi} \tag{3.1}$$

where $f(\theta)$ is the elastic scattering amplitude at scattering angle $\theta$ in the centre of mass frame, $Q$ is the centre of mass momentum, and $\sigma_{Total}$ is the total cross section. At small scattering angles where $\theta \rightarrow 0$ one can use measurements of elastic differential cross-sections for the determination of the total cross-section.

## 3.5 The Inner Tracking System

The upgraded tracking system is a completely new system for Run II and consists of four distinct subsystems. The Silicon Microstrip Tracker (SMT) surrounds the Tevatron beam pipe. It is designed to reconstruct primary interaction vertices and secondary decay vertices. Radially outside the silicon detector is a scintillating Central Fibre Tracker (CFT) covering the central pseudorapidity region and, coupled with the SMT provides charged particle track reconstruction and momentum measurement. The SMT and

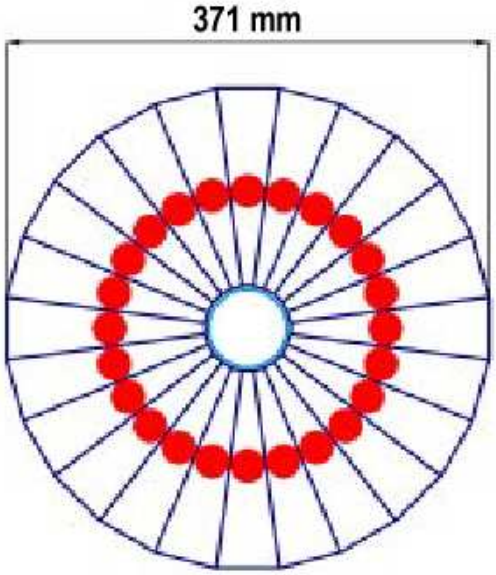


Figure 3.2: Diagram showing the radial arrangement of the scintillator wedges in the Luminosity Monitor detector. The spot on each wedge shows the position of the photo multiplier tube that provides readout.

CFT are contained within a cylindrical 2.8m long superconducting 2 Tesla Solenoid magnet [30]. In the gap between the solenoid and calorimeter is a preshower scintillator detector to aid in electron and photon ID and provide pre-calorimeter energy sampling. Fig. 3.3 Shows the layout of the detectors within the inner tracker.

The inner tracking system is designed to give momentum measurement and charged particle identification within the magnetic field, provide good electron identification and separation of electrons from pions. It provides tracking coverage over the $|\eta| \leq 3.0$ region and secondary vertex measurement for identification of flavour in jets. It must also provide this information in the short bunch crossing interval of 396 ns and provide the ability to trigger an event based on track candidates.

### 3.5.1 The Silicon Microstrip Tracker

Immediately surrounding the beam pipe is the Silicon Microstrip Tracker (SMT). Constructed in a barrel and disk arrangement to cover the extended interaction point of the colliding beams, which occur with a $\sigma_z$ of 25 cm about the centre of the detector, to provide three dimensional tracking and vertex reconstruction with a high resolution of approximately $30\mu m$ via 793,000 readout channels [27][28]. At the time of writing, the performance of the SMT detector is approaching this design accuracy as can be seen from the plot in Fig. 3.4 The tracking precision scales as the inverse of $p_T$ , meaning that the higher the transverse momentum, the better the resolution of the

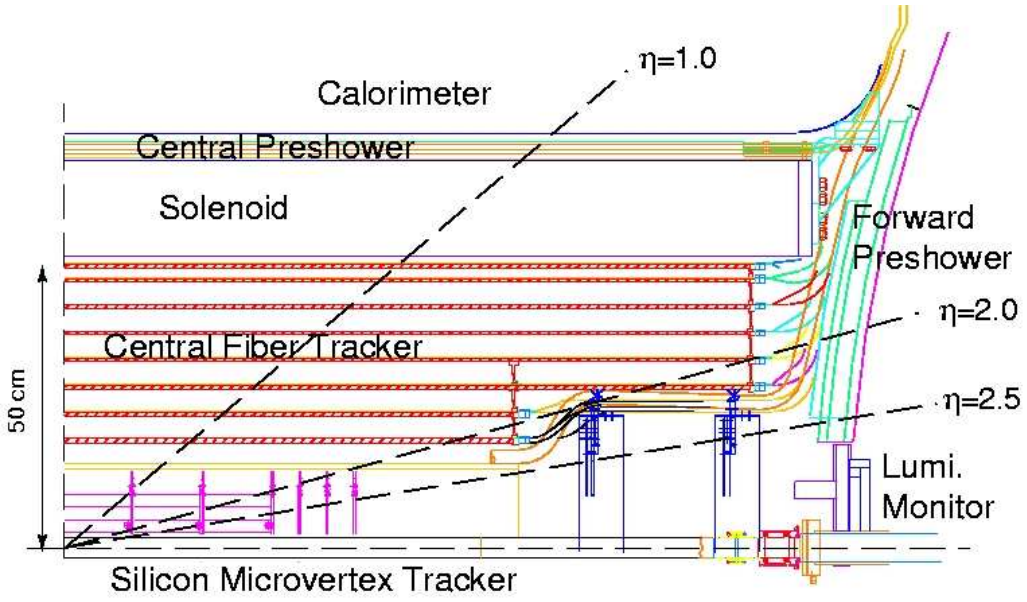


Figure 3.3: Diagram showing the layout of the inner tracking detectors. A quarter cross section of the detector is shown, with the SMT in purple, the CFT Layers in red and the preshower detectors in green. The Luminosity Monitor position is also shown around the beam pipe in the lower right hand corner of the diagram.

track measurements. The effect of the momentum on the tracking resolution is shown in Fig. 3.5. A table showing the physical parameters of the SMT detector is given in Table 3.1.

The disk and barrel construction, which can be seen as a 3D rendering in Fig. 3.6, is designed so that the barrel detectors provide r-$\phi$ coordinate measurement and the disk detectors provide r-z and r-$\phi$, allowing high $\eta$ particle vertices to be constructed by the barrels.

| DØ | Barrels | F-Disks | H-Disks |
|---|---|---|---|
| Layers/planes | 4 | 12 | 4 |
| Channels | 387120 | 258000 | 147456 |
| Readout Length | 12 cm | 7.5 cm | 14.6 cm |
| Inner Radius | 2.7 cm | 2.6 cm | 9.5 cm |
| Outer Radius | 9.4 cm | 10.5 cm | 26 cm |

Table 3.1: Table showing the physical attributes of each of the sections of the SMT detector.

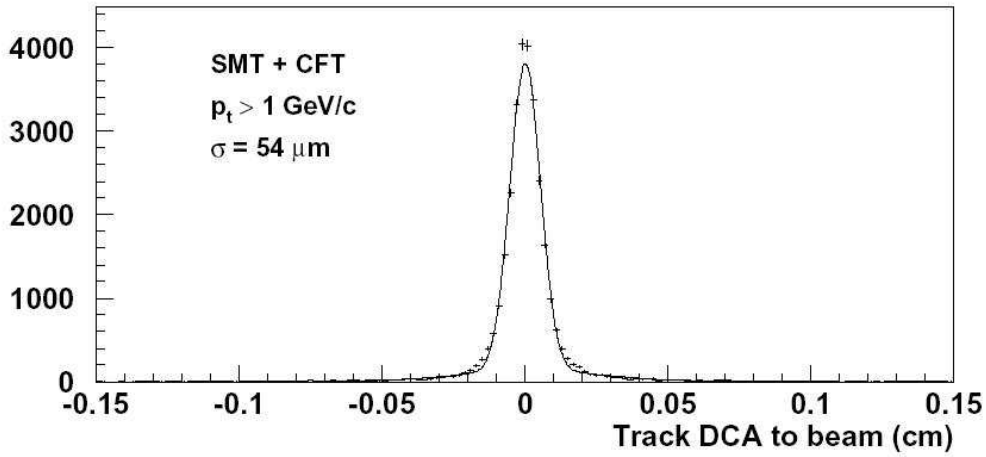


Figure 3.4: Plot illustrating the actual resolution of the SMT detector using the AATrack Tracking Algorithm for $p_T > 1$ GeV tracks using input from both the SMT and CFT detectors at time of writing, with a resolution of $54\mu$m.

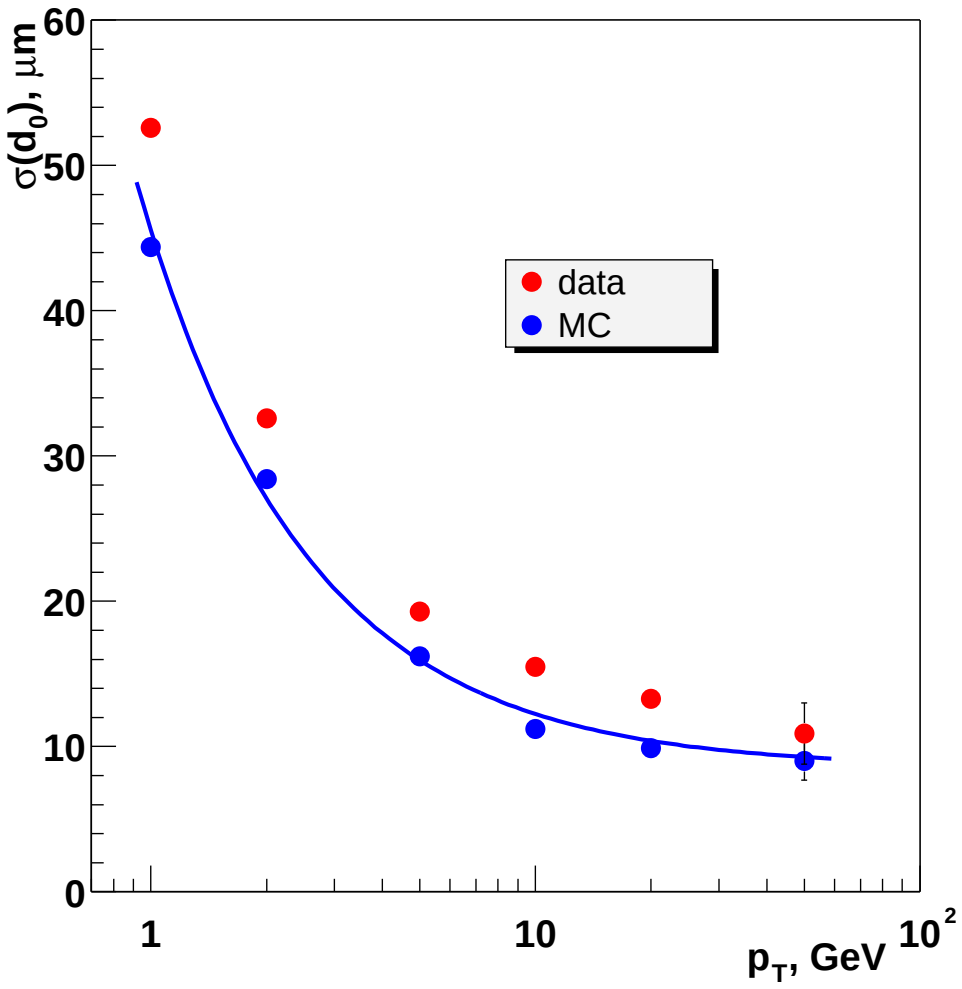


Figure 3.5: Plot showing the scaling of the standard deviation of impact parameter measurements versus transverse momentum. The plot shows the scaling of tracking resolution as a function of $p_T$ , illustrating the fall off in precision for low energy tracks.

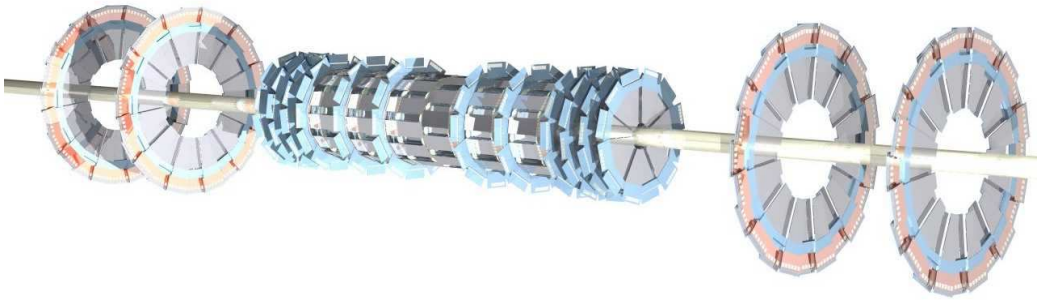

Figure 3.6: Rendering of the SMT detector showing the barrel and disk construction, and the four H-disks.

There are six 12 cm long barrels, each separated by an F-Disk. Each barrel consists of four radial layers. Layers 1 (closest to the beam) and 3 consist of double sided detectors comprised of an axial layer and a 90 degree offset z-layer for the four central barrels, while the two end barrels have only single sided axial strips for layers 1 and 3. Barrel layers 2 and 4 consist of double sided axial, and 2 degree stereo strips. A cross-sectional view of a barrel is provided in Fig. 3.8.

The F-Disks are small diameter double sided layered disks with a $\pm$ 15 degree stereo angle offset. Each F-disk is constructed from 12 overlapping r-$\phi$ wedges. The F-disks are approximately 8 mm thick. A cross-sectional (r-$\phi$) view of an f-disk is shown in Fig. 3.7.

At either end of the SMT are a pair of H-disk assemblies located at $|z| =$ 110cm and $|z| =$ 120cm, which are larger diameter disks with a $\pm 7.5^o$ stereo double strip layers that extend tracking coverage to $|\eta| = 3$.

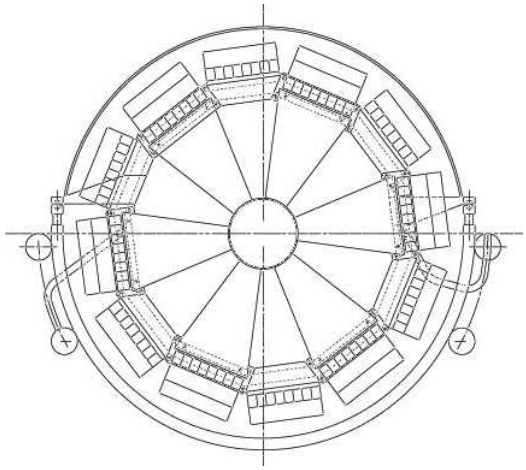

Figure 3.7: r-$\phi$ diagram of an F-disk in the SMT detector showing the arrangement of the wedge shaped silicon strips.

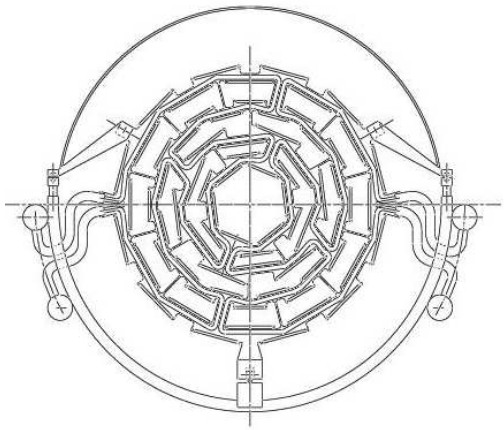

Figure 3.8: Cross sectional view of a barrel in the SMT detector in the r-$\phi$ plane showing the arrangement of silicon ladders around the beam pipe.

The silicon microstrip detectors themselves are 300$\mu$m thick and are single or double sided carbon fibre rails called ladders, which support the silicon strips and the SVXII digitizer chips that read out the signals. The SVXII chips are optimised for the Run II crossing interval and are read out via a high density circuit connection to the electronic signal processing crates on the platform below the detector.

The readout speed of the SVXII is too slow to provide fast information for a fast trigger pickoff, so the SMT readout is pipelined into the later stages of the trigger, with fast triggering in track based objects being left to the CFT.

### 3.5.2 The Central Fibre Tracker

Surrounding the SMT detector the Central Fibre Tracker (CFT) is a layered scintillating fibre detector that complements the SMT in track reconstruction and charged particle momentum measurements in the $|\eta| \leq 2.0$ region. The CFT also provides a fast readout for fast triggering on tracks within the $|\eta| \leq 1.6$ region for the Level 1 trigger [29].

The CFT is constructed from eight layers of scintillating fibres in closely packed 256 channel fibre ribbons. These ribbons are mounted on eight concentric carbon-fibre cylinders at radii ranging from 19.5 to 51.5cm from the beam pipe. The innermost cylinders are 1.7m in length to accommodate the SMT H-disks, while the outer layers are 2.5m in length. A diagram of the CFT construction is shown in Figure 3.9. A summary of the physical characteristics of each CFT layer is given in Table 3.2.

The Fibre ribbons are layered into interlocking doublet layers, arranged so that one layer of the doublet is offset by half the diameter of a fibre to remove dead spots between fibres, as illustrated in Fig. 3.9b. Two doublet layers are mounted per support cylinder with the first (extending radially from the beam pipe) doublet layer oriented axially to the beam pipe, and the second layer oriented at a $\pm 3^o$ stereo angle offset to provide full $\eta - \phi$-z information. The axial and stereo fibres are read out separately with the axial readout being used for fast track trigger input while the stereo information is used for later stages of the trigger system.

The fibres themselves are made from a doped polystyrene core clad with

| Layer | Radius (cm) | Fibres per layer | Active Length (m) |
|---|---|---|---|
| A | 19.99 | 1280 | 1.66 |
| AU | 20.15 | 1280 | 1.66 |
| B | 24.90 | 1600 | 1.66 |
| BV | 25.60 | 1600 | 1.66 |
| C | 29.80 | 1920 | 2.52 |
| CU | 29.97 | 1920 | 2.52 |
| D | 34.71 | 2240 | 2.52 |
| DV | 34.87 | 2240 | 2.52 |
| E | 39.62 | 2560 | 2.52 |
| EU | 39.78 | 2560 | 2.52 |
| F | 44.53 | 2880 | 2.52 |
| FV | 44.69 | 2880 | 2.52 |
| G | 49.43 | 3200 | 2.52 |
| GU | 49.59 | 3200 | 2.52 |
| H | 51.43 | 3520 | 2.52 |
| HV | 51.59 | 3520 | 2.52 |

Table 3.2: Physical Parameters of the CFT Layers, with A being the innermost layer, and U and V signifying the stereo offset layers. The values shown are the radius from the beam pipe, the number of fibres in the layer and the active length of fibre parallel to the beam pipe.

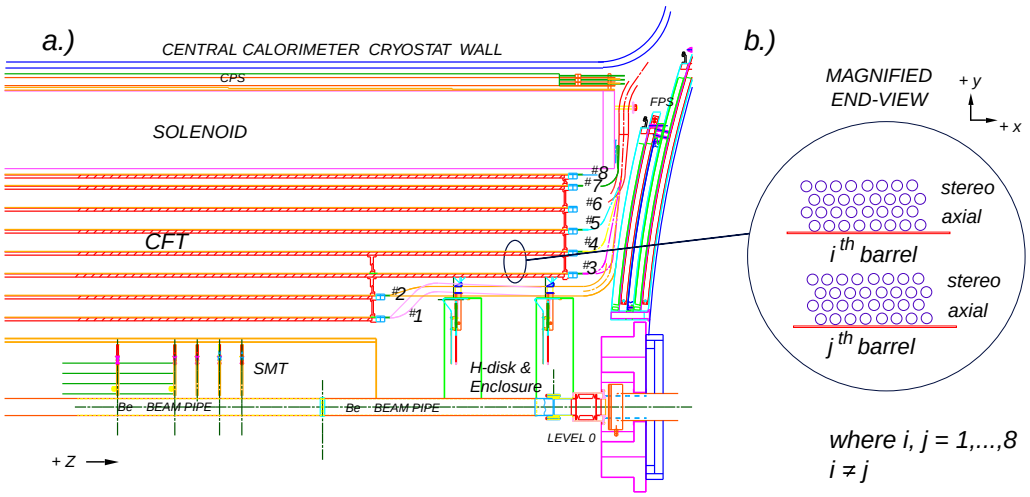


Figure 3.9: Diagram showing the construction of the CFT as a) Cross sectional diagram showing a quarter of the detector in r-z with the layers of fibres and b) showing the r-$\phi$ arrangement of the fibres within the ribbons and the doublet layers.

an acrylic, and further coated in a fluro-acrylic material to improve light trapping within the fibre. The fibres are 835$\mu$m thick with 15$\mu$m of cladding, and are mounted with an accuracy of 40$\mu$m or better to ensure high granularity and good resolution. When hit by a charged particle the fibres scintillate in the yellow-green part of the visible spectrum with a peak wavelength of 530nm, producing approximately 14 photoelectrons for a minimum ionising particle.

Each ribbon is optically connected to a clear fibre waveguide ranging from 8-11m in length, which conducts the light pulse to a Visible Light Photon Counter (VLPC). The VLPCs are solid state photo multipliers running at

cryogenic temperatures, capable of detecting single photons with high efficiency, and converting them into an electron shower with a gain factor of 20x-60x. Each VLPC has 8 x 1mm pixels, each of which connects to a fibre. The VLPCs are grouped into cassettes of 128 units providing 1024 channels per cassette. Each cassette is supported by two Analog Front End (AFE) circuit boards which convert the VLPC currents into digital signals using SVXII chips. Since the SVXII is not fast enough (90ns rise time) to provide a fast readout to the Level 1 trigger, and the VLPC signal charge is not in the SVXII acceptance range, a signal discriminator chip called a SIFT chip is inserted between the VLPC and SVXII chip. The SIFT chip has a higher bandwidth than the SVX and thus can provide the fast trigger pickoff while also shunting the VLPC signal into the SVX acceptance range.

The AFE then provides the digitised signals and raw analog signals to the trigger system using the same readout system as the SMT. The discriminated SIFT output is sorted and sent to the Digital Front End (DFE) system for fast Level 1 trigger examination. The DFE system is discussed in detail in Chapter 4.

### 3.5.3 The Central Preshower Detector

The Central Preshower (CPS) detector is a scintillator based system installed in the cylindrical 51mm gap between the outer edge of the solenoid and the inner face of the central calorimeter cryostat at a radius of 72 cm from the beam line, as illustrated in Fig. 3.10. The CPS covers the $|\eta| < 1.2$ region

and consists of three concentric layers of scintillator strips, the innermost of which is oriented axially to the beam pipe, the outer two being offset by a $\pm 22.5^o$ stereo angle. The CPS provides both extended tracking information outside of the solenoid and early energy sampling upstream of the calorimeter to correct for electromagnetic energy effects in the solenoid [31].

The scintillator strips are 7mm sided equilateral triangular cross section plastic scintillator with a 1mm diameter hole through the centre containing an 835$\mu$m thick wavelength shifting fibre. The fibres are split at z = 0, and are connected to clear fibre waveguides at either end to reduce occupancy. The triangular strips are arranged in an interlocking saw tooth manner as shown in the inset of Fig. 3.10 to improve position resolution for minimum ionising particles as a result of light sharing between two neighbouring strips.

A lead absorber plate, tapered in z, combines with the solenoid to present a uniform two radiation length thickness over the entire pseudorapidity range covered by the CPS. The combined thickness of solenoid and absorber initiates showers of electromagnetic particles just before the preshower scintillators. The scintillator fibres are read out via the same type of VLPC/AFE/DFE system as the CFT. The axial CPS layer is fed into the fast Level 1 trigger system to complement the CFT axial fibres for fast track and cluster triggering. The CPS readout system is discussed in detail in Chapter 4.

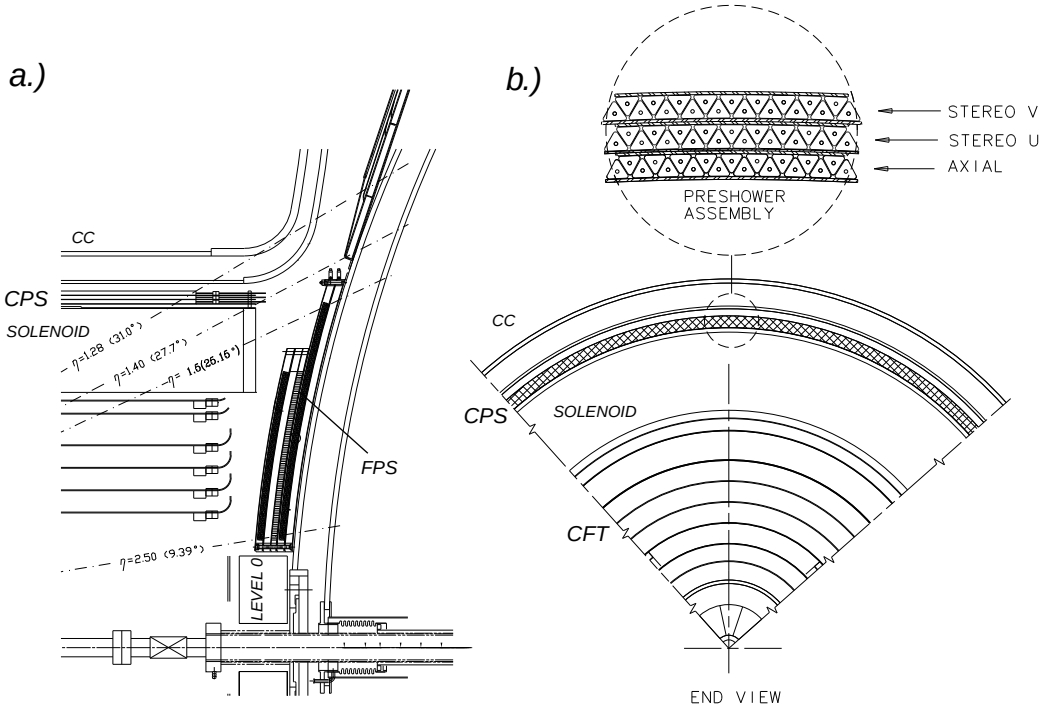


Figure 3.10: Diagram showing a cross sectional view of the CPS detector, and the arrangement of the interlocking scintillator strips (inset).

### 3.5.4 The Forward Preshower Detectors

The Forward Preshower Detectors (FPS) are two identical detectors mounted on the inner face of each end cap calorimeter cryostats. They use the same scintillator strips as the CPS, differing only in geometry and number of layers, the triangular strips with an embedded scintillating fibre read out via the VLPC/AFE/DFE chain. The FPS provides the same track clustering and pre-calorimeter energy sampling abilities as the CPS but covers the psuedo-rapidity region $1.4 < |\eta| < 2.5$ [32].

The FPS is composed of a two-radiation-layer-thick lead absorber sandwiched between two scintillator planes each consisting of two stereo sub layers as shown in Fig. 3.11.

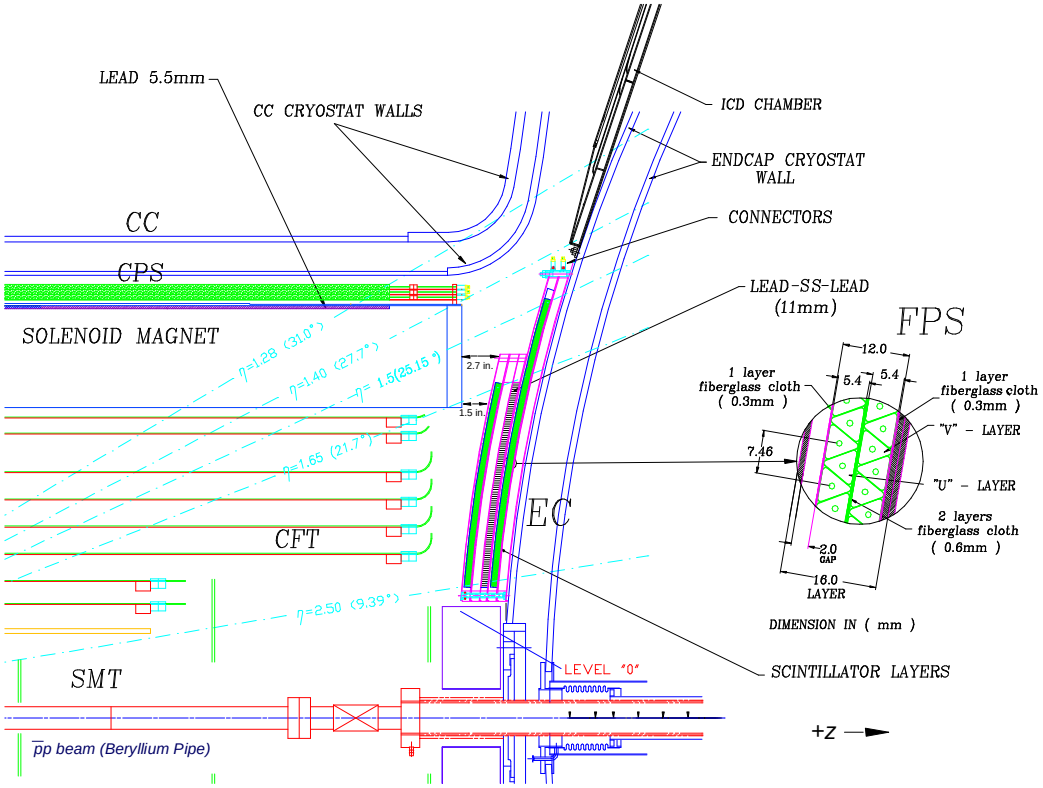


Figure 3.11: Diagram showing a cross sectional view of the FPS detector.

The FPS is segmented into eight azimuthal wedges that physically cover a $45^o$ angle in $\phi$, however only the central $22.5^o$ are active scintillator, with the remainder used to provide mechanical support and space for readout waveguides. The wedge positions are staggered in $\phi$ orientation in different layers to provide full azimuthal coverage.

## 3.6 The Calorimeter

The Calorimeter system is designed to provide measurement of the energy and direction of electrons, photons and jets over a large pseudorapidity region with good resolution. The calorimeter is also required to distinguish electromagnetic and hadronic showers and provide sufficient information to calculate missing transverse energy associated with non-interacting particles such as neutrinos. High Energy Particles traversing the calorimeter interact with large detector masses, which produce a shower of secondary, lower energy particles. These secondary particles also shower, producing a cascade of particles in the detector, which are primarily in the longitudinal direction of the original incident particle, although some transverse development occurs due to multiple scattering. Eventually, almost all of the original particle's energy is deposited in the calorimeter, and can be used to create a detector signal representing the energy of that particle.

The DØ Calorimeter is a finely segmented, hermetic sampling calorimeter composed of cell modules containing depleted Uranium absorber plates and a liquid Argon (LAr) sampling medium [33][34]. The high density of the Uranium allows the detector to be compact while still presenting a large amount of interacting mass to initiate showers, it also compensates for loss of visible hadronic energy in showers as it will emit radiation triggered by the showering particles. LAr was chosen as the sampling medium since it provides a uniform gain throughout the calorimeter allowing a channel-to-channel response dependent only on the gap and absorber plate thickness.

LAr can be easily segmented into cells, is easy to calibrate and is radiation hard due to its inert nature, it does however impose the restriction of running at LAr cryogenic temperatures, requiring the calorimeters to be contained in cryostats.

Construction and installation restrictions along with the required access to the Inner Tracking System forces the calorimeter to be built in three separate cryostat units, one central cylindrical unit and two symmetric end cap units. The Central Calorimeter (CC) extends radially from the beam pipe, enclosing the Inner Tracking System, and covers the $|\eta| \leq 1.2$ region. The two End Cap (EC) Calorimeters, one for the north and one for the south end of the detector cover the forward pseudorapidity region $1.3 \geq |\eta| \geq 4.2$. A cutaway schematic of the calorimeter and the arrangement of the three cryostats is shown in Figure 3.12.

Each of the cryostat sections houses three distinct calorimeter modules, an electromagnetic module, a fine hadronic module and a coarse hadronic module. These modules are composed of layers of cells, segmented into $\Delta\eta \times \Delta\phi$ cells which are arranged so that they form cell towers across the layers which point back to the nominal interaction point, as illustrated is Fig. 3.14. The cells consist of a Uranium absorber plate and a signal board separated by a LAr filled gap. The signal boards are printed circuit boards sandwiched in an insulator, which is coated in a resistive epoxy. These boards are charged to 2.0 - 2.5 kV with respect to the grounded absorber plate, inducing a drift field across the LAr gap. As particles interact in the absorber plate, they shower across the gap producing ionisation tracks, and a flow of charge towards the

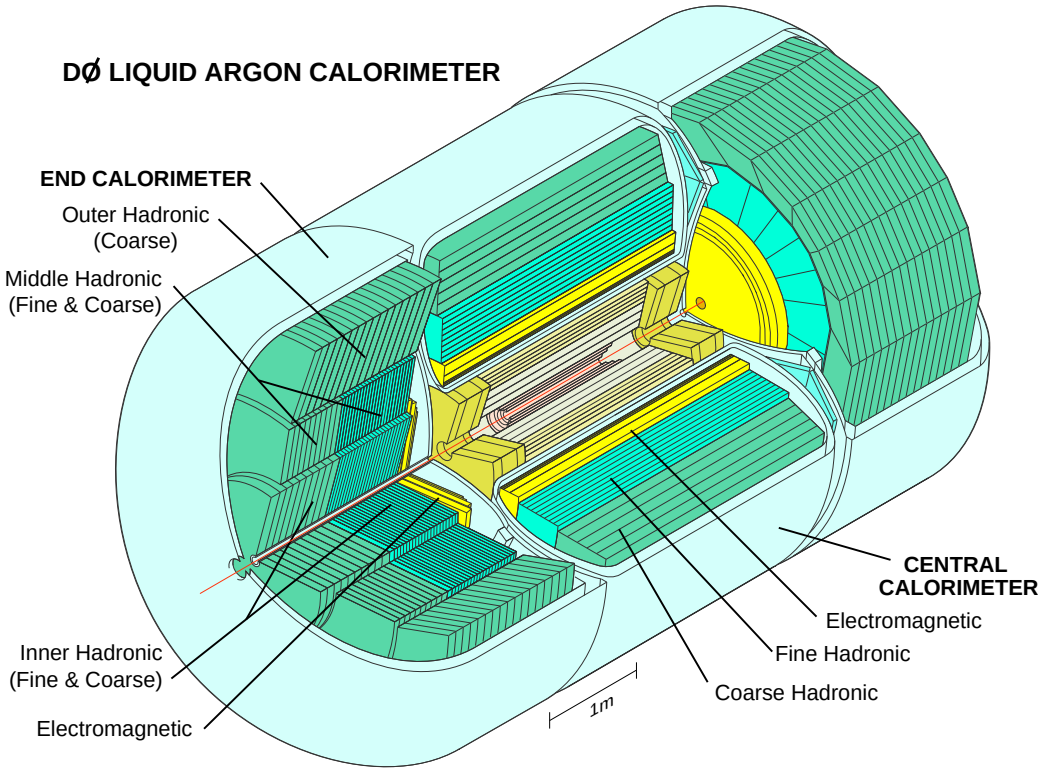


Figure 3.12: Cutaway diagram of the (Run I) DØ Calorimeter.

signal board, resulting in signal containing information about the amount of energy deposited. Readout pads that are consecutive in depth for the same $\eta - \phi$ regions are grouped into readout cells. A schematic showing the layout of a calorimeter cell is shown in Fig. 3.14. Since the average drift time of the ionisation charge across the LAr gap is approximately $450ns$ [22], readout of the Calorimeter has to be buffered to avoid deadtime. This is addressed by the upgraded Calorimeter electronics, which have been designed to reduce time spent shaping the cell readout, and buffers the analog calorimeter sig-

nals so that they can be digitized upon a Level 1 trigger decision, thus buying extra processing time and allowing multiple events to be handled simultaneously. The signal is produced by comparing the pre-event cell readout voltages to those after the event, thus reducing the impact of event pile-up from multiple events overlaying each other within the calorimeter. The new electronics improve the peak signal sampling time between the pre-event and post event readouts from $400ns$ in Run I to $2.2\mu s$.

- The Electromagnetic (EM) Calorimeter is the innermost layer of the calorimeter. It is the closest calorimeter to the interaction point and is optimised for measuring electrons and photons. The EM Calorimeter is 21 radiation lengths deep and consists of four layers of cells. The first and second layers extending radially from the beam are designed to measure the longitudinal development of a shower and is segmented into $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ cells which extend for 2(2.3) Radiation lengths in the CC (EC). Electromagnetic showers are expected to reach a maximum in the third layer of the EM Calorimeter, thus this layer is 7 radiation lengths thick and is more finely segmented into $\Delta\eta \times \Delta\phi = 0.05 \times 0.05$ cells to increase resolution for measurement of the transverse position of a shower. The fourth layer of the EM Calorimeter is 10 radiation lengths deep to contain the shower and has the same segmentation as the first two layers.

- The Hadronic Calorimeter exists in two parts, the Fine Hadronic (FH) section, and the Coarse Hadronic (CH) section. The FH calorimeter

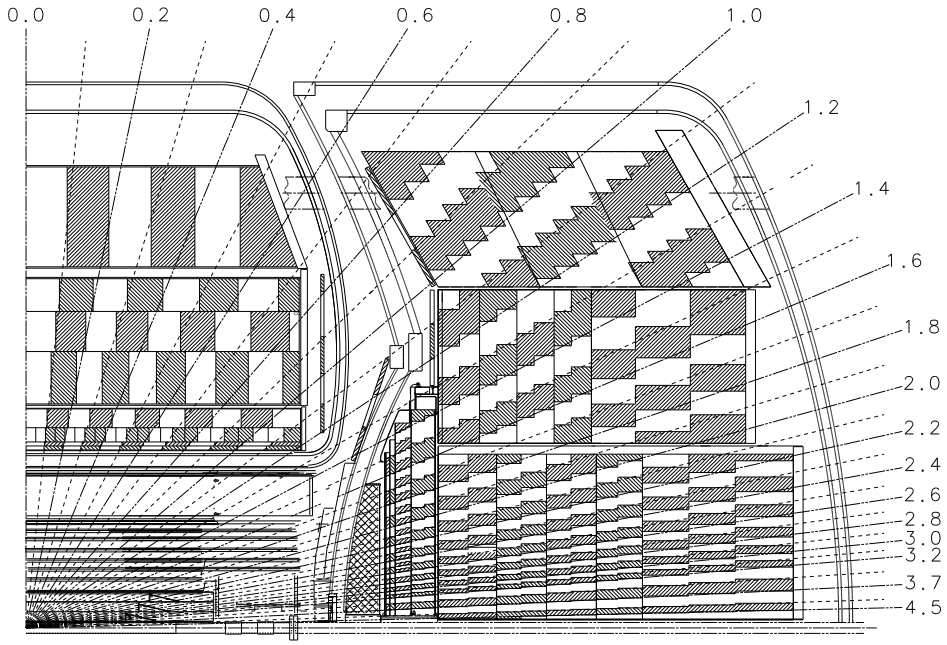


Figure 3.13: Cross-sectional drawing of one quarter of the calorimeter showing the cell and tower arrangement.

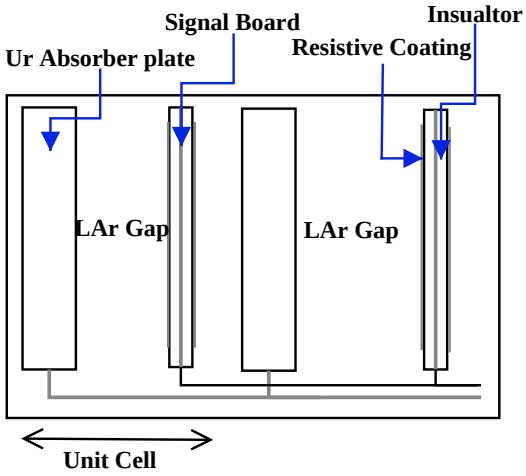


Figure 3.14: Schematic of a calorimeter cell, showing the Uranium absorber pad, LAr gap and the signal board.

surrounds the EM calorimeter and is four layers of cells containing 6mm Uranium alloy absorber plates to measure the shape of hadronic showers. The CH calorimeter surrounds the FH calorimeter and is a single layer of cells with 46.5mm copper absorber plates to contain the longitudinal shower within the calorimeter and measure its energy. The combined hadronic calorimeters present 7-9 nuclear interaction lengths of material in $\Delta\eta \times \Delta\phi = 0.1 \times 0.1$ segmented cells. The CC and EC are almost identical except for geometry and a slightly different cell ordering and an extra layer of hadronic calorimeters which use depleted Uranium absorbers to maximise the amount of material in the available space.

Although the Calorimeter exists largely in the same form as it was in Run I, the readout system needed to be upgraded to handle the higher event rate of Run II of the Tevatron. This reduces the time available for signal gathering and shaping between events and also requires event buffering until a Level 1 trigger decision has been made, without any signal degradation. Each cell signal is taken to a preamplifier, where the signal is converted to a voltage proportional to its charge. The signal voltage is then passed to a signal shaper which remove noise and any slow variations in the signal voltage, whilst also buffering events to avoid cross talk between crossings and avoiding dead time for signal processing. When a Level 1 trigger accept occurs, the signals are digitized and read out into buffers to await higher level trigger decisions. The Calorimeter has a total of 47,800 readout channels.

The Calorimeter performance is well known from Run I operations, and is expected to be comparable in RunII leading to a positional resolution of 0.8-1.2mm for 100 GeV electrons.

There is also an Intercryostat detector (ICD) to cover the $1.1 < |\eta| < 1.4$ to cover the region between the CC and EC calorimeters which contain various support structures and dead spots with no energy sampling. The ICD is mounted on the inner wall of the EC cryostats and consist of scintillating tiles arranged to sample the energy of particles which traverse the support structures and escape through the CC walls. The tiles are $22.5^o$ wedges in $\phi$ containing a wavelength shifting fibre, read out by photo multiplier tubes leading to preamps and signal shapers.

## 3.7 Muon Detectors

The DØ Muon detector is comprised of three detector systems, which are arranged into three layers throughout the detector, named A, B and C Layers. The three types of detectors are Proportional Drift Tubes (PDTs), Mini Drift Tubes (MDTs) and trigger scintillation counters [35][36]. The entire muon detector is divided into two subdetectors, the Wide Angle Muon System (WAMUS) which covers the pseudo-rapidity range $|\eta| \leq 1.0$, and two Forward Angle Muon Systems (FAMUS) which provide muon coverage $1.0 \leq |\eta| \leq 2.0$. A 2 Tesla muon Toroid magnet to deflect muons in the $r - z$ plane to allow muon trajectory and momentum measurements by comparing A Layer muon hits within the toroid to B and C layer hits outside the toroid. A diagram

showing the layout of the muon detectors is given in Fig. 3.15.

### 3.7.1 The Central Muon System

The Central Muon System consists of 94 PDTs, arranged in three layers, the A Layer in between the Central Calorimeter Cryostat and muon toroid, consisting of three or four layers of PDTs depending on the physical space available, and the B and C layers are positioned consecutively outside the toroid and are comprised of 3 PDT layers each. The A Layer also contains a layer of scintillators on the calorimeter side, which is segmented into $4.5^o$ wedges in $\phi$ for input into the trigger to allow decisions to be made using trigger information. A scintillator layer on the top and bottom of the detector is used to detect cosmic ray muons as they pass through the detector. The PDTs themselves are rectangular aluminium tubes containing cells $10 \times 5.7cm$ with a drift distance resolution of approximately 1 mm. When PDT hits are matched with tracks from the Inner Tracking system, the muon resolution for a 100 GeV $p_T$ Muon is approximately 15%.

### 3.7.2 The Forward Muon System

The Forward Angle Muon System is new for Run II, and consists of MDTs and pixel scintillators, with the A Layer in between the End Cap Calorimeter and toroid, with the B and C layers outside the toroid, similar to the central muon system. The MDTs are each comprised of eight $1cm \times 1cm$ cells and are arranged four layers deep in the A Layer and 3 deep in the B and C

Layers. The forward muon scintillators are arranged in $4.5^o$ $\phi$ sectors that match the segmentation of the CFT to allow easy matching with the CFT Axial tracking system and provide input to the trigger for forward muons. The Scintillators are divided into 1.0 $\eta$ sections and arranged as shown in Fig. 3.16.

## 3.8 The DØ Trigger System

With the short interval between beam crossings (396 ns), high luminosity provided by the Tevatron ($2 \times 10^{32} cm^{-2} s^{-1}$) in Run II and the millions of detector readout channels, it becomes impossible to record every event. Instead a fast selection system to find interactions containing well defined basic physics objects and store them is used. This system is called triggering and is implemented using multiple hardware and software steps. A schematic of the trigger system can be seen in Figure 3.17.

The Run II trigger is a significant upgrade over the Run I trigger, to account for the increase in event rates and to include information from the new tracking detectors. Event buffers between L1/L2 and L2/L3 de-randomize the Poisson distributed arrival times of the events to decrease event pileup dead time. The buffers also remove the L1 dead time while L2 is running that was observed in Run I [37].

Event rates start with the 132ns bunch separation in the detector, giving a rate of 7.5 MHz into L1. The maximum achievable rate between L1 and L2 is 10kHz due to the rate of SVX chip digitization used in the SMT and

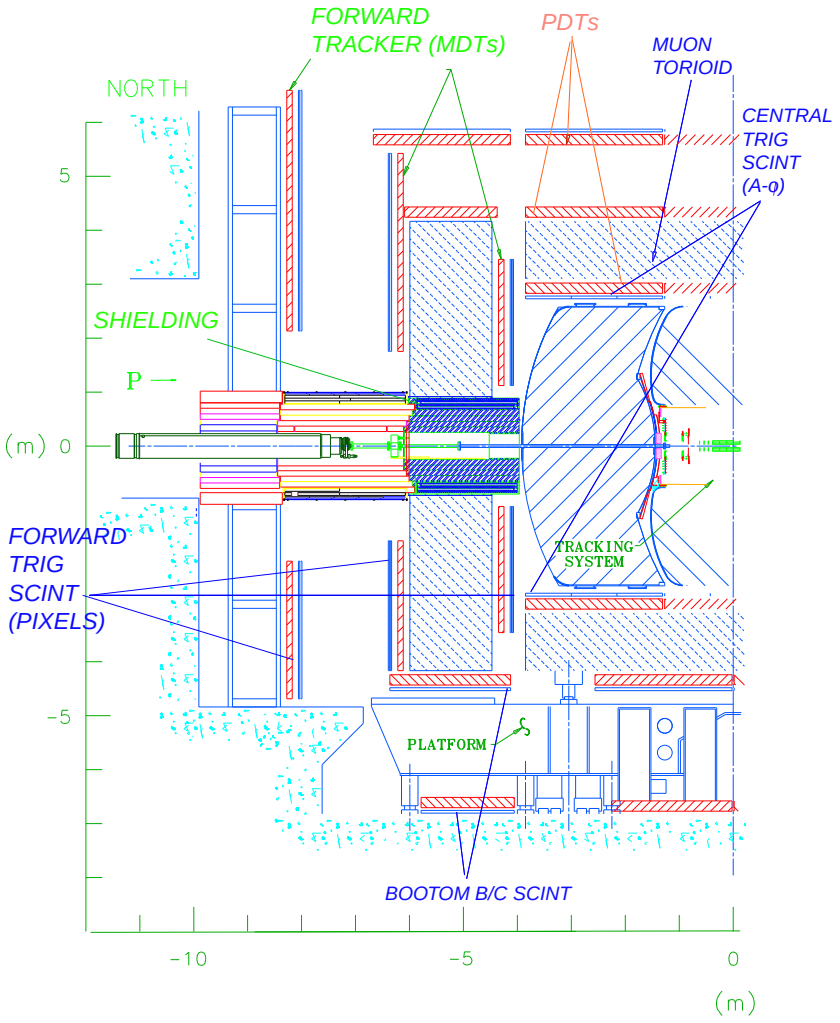


Figure 3.15: Layout of the DØ Muon System showing the central and forward muon systems, including the A,B and C layers and muon toroid magnet.

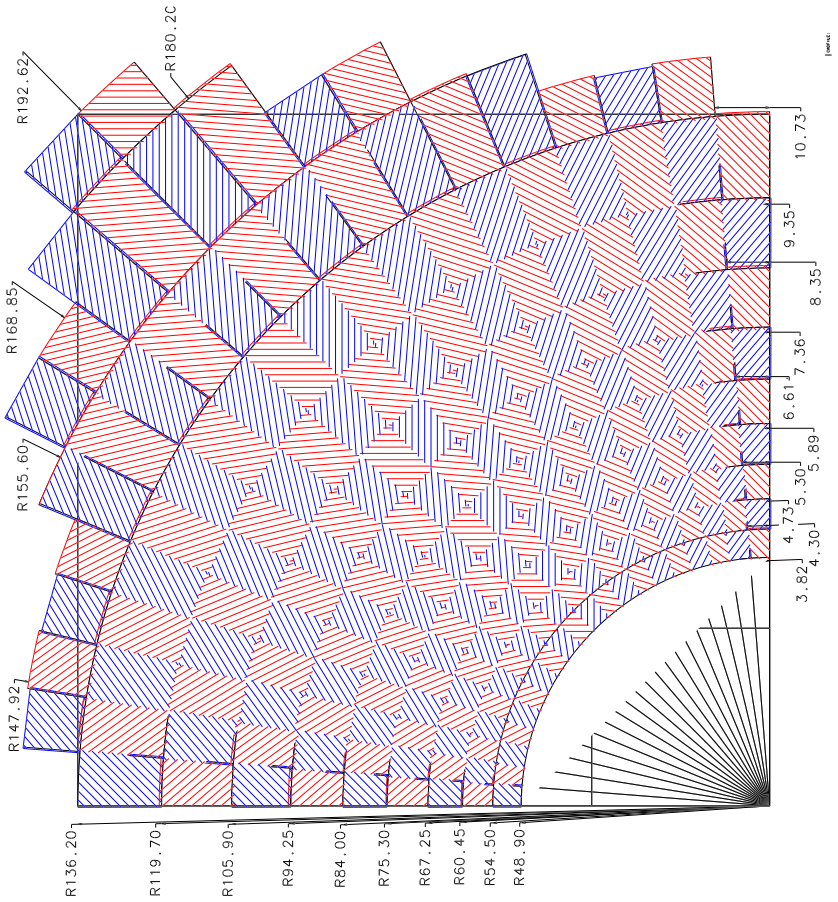


Figure 3.16: Diagram showing the arrangement of one quarter of the forward muon scintillator planes in the $r - \phi$ plane.

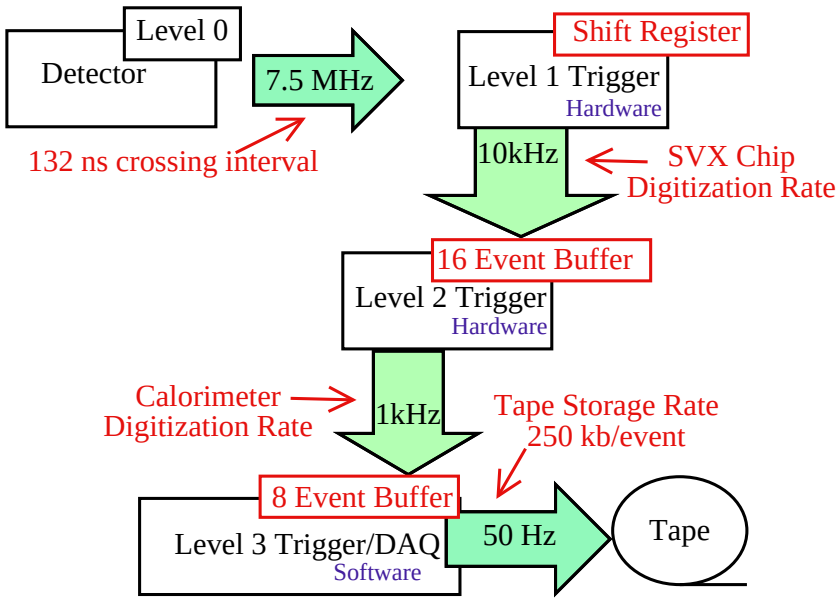


Figure 3.17: Schematic of the DØ Trigger System showing the event rates between the different trigger levels and the buffers between them. The Level 0 trigger provides luminosity information from the Luminosity Monitor (see §3.4) and sends a single input containing luminosity information to the Level 1 trigger. The Level 1 Shift Register is a 3.5 $\mu s$ memory that contains all the Level 1 data, allowing a 3.5 $\mu s$ decision time.

scintillator detectors, which require a minimum of 90 ns to produce fully digitised signals. The L2 to L3 rate is 1kHz, due to the calorimeter digitisation rate. L3 outputs events at 50Hz with an event size of approx 250kB to tape, for later offline reconstruction.

### 3.8.1 Level 1 Trigger

The Level 1 trigger is a hardware trigger using a pipelined architecture. The L1 trigger includes information from the CFT and CPS axial detectors, combined into the L1 Central Track Trigger (CTT) the FPS, Calorimeter and Muon detectors. Each subsystem's data is clocked into a 3.5 $\mu s$ long memory during which time L1 has to decide to trigger [38]. The L1 trigger decision is a two stage process:

1. Each subsystem sends trigger terms (bits) to the L1 framework every 132ns

2. The framework ANDS/ORS these subsystem bits into 128 L1 trigger bits. If any of these trigger bits are set, it generates an L1 trigger accept.

Each L1 subsystem processor examines the raw data from the detector and does some simple logic operations to produce trigger bits corresponding to objects with physics potential. For example one may require a Calorimeter trigger tower with $E_T$ above a threshold, Hit patterns that form tracks with $p_T$ above a threshold in the CFT and so forth. Each L1 subsystem is described

briefly below, a block diagram showing the components of the L1 trigger can be found in Fig. 3.18.

- The L1 framework is a pre-programmed hardware system based on Field Programmable Gate Array (FPGA) logic, which is used to collect the 128 trigger terms from the L1 sub-systems. The FPGAs are based on VME cards and determine whether the required combination of L1 trigger bits are set according to the programmed requirements and if so generates a L1 trigger accept in $4.2\mu s$, and passes its decision to the Level 2 trigger.

- The L1 Central Track Trigger (L1CTT) is based on axial fiber readout from the CFT and CPS detectors, which are integrated to form a single L1 trigger using programmable logic devices to perform tracking in four $p_T$ bins and CPS cluster matching. The L1CTT sends 64 terms to the L1 framework. This system is discussed in detail in chapter 4.

- The Level 1 Calorimeter (L1CAL) trigger groups calorimeter cells into "trigger towers". There are 1280 trigger towers in total, each of which has an Electromagnetic (EM) energy and a Hadronic energy. Each trigger tower has pre-defined energy threshold reference sets; 4 EM reference sets, and 4 EM+Hadronic reference sets. Each of those sets in turn has 4 count thresholds. It sets a trigger bit if there are more than the count threshold of cells above the reference. It is also possible to implement other triggers, for example: total EM and EM+Hadronic

reference sets that look for one or two or more towers in a given quadrant of the calorimeter, this information can also be used with the Quadrant information from the L1CTT for charged particle identification by matching showers to tracks. The L1CAL sends 32 L1 trigger bits to the L1 framework.

- The Level 1 Muon (L1MU) trigger makes fast trigger decisions using the muon detector which is divided into 3 regions, Central ($|\eta| \leq 1.0$), North Forward and South Forward ($1.0 \geq |\eta| \geq 2.0$). Each region has 3 layers A, B and C which are comprised of drift tubes and scintillators as described earlier in this chapter. For each layer, the L1MU trigger creates centroids, which are combined with scintillator and CFT tracks to produce muons. L1MU has 256 possible terms based on various criteria such as $p_T$ thresholds, eta region, muon quality etc. 32 of these terms are selected at the start of the run (and do not change during the run) and are sent to the L1 framework. Non-L1 triggered muon data is buffered for output to L2 and L3.

- The Level 1 Forward Preshower (L1FPS) provides 32 terms to the L1 framework based on FPS clusters as a function of high/low shower thresholds, minimum ionizing particle (MIP) layer hit matching using programmable logic. This subsystem readout is discussed in detail in chapter 4.

### 3.8.2 Level 2 Trigger

The Level 2 trigger system correlates the L1 trigger information from the different L1 subsystems, and examines input from some of the detector systems not used in L1 [39][40]. A block diagram showing the L1 and L2 trigger components can be seen in Fig. 3.18

The L2 trigger is a two stage design:

1. Pre-processors format the L1 data in parallel, in some cases augmenting it with other data

2. A global processor correlates the information across the whole detector

L2 runs with an input rate (L1 accept rate) of 10kHz and an output rate (L2 accept rate) of 1kHz and is required to respond to events in the same order they are received, using single Compaq Alpha processors, grouped into crates for each pre-processor and L2 global. Each L2 crate has a single administrator processor which controls L1 and L2 preprocessor input, L3 output, and I/O to the trigger control computer. Each crate also has one or more worker processors which process the data. Data is broadcast to all the alphas in a crate by a custom bus, and a VME Buffer Driver is used to send data to L3.

The L2 preprocessors are organised as follows:

- Muon Preprocessor
  - The Muon system is the only system to have full detector readout at L2

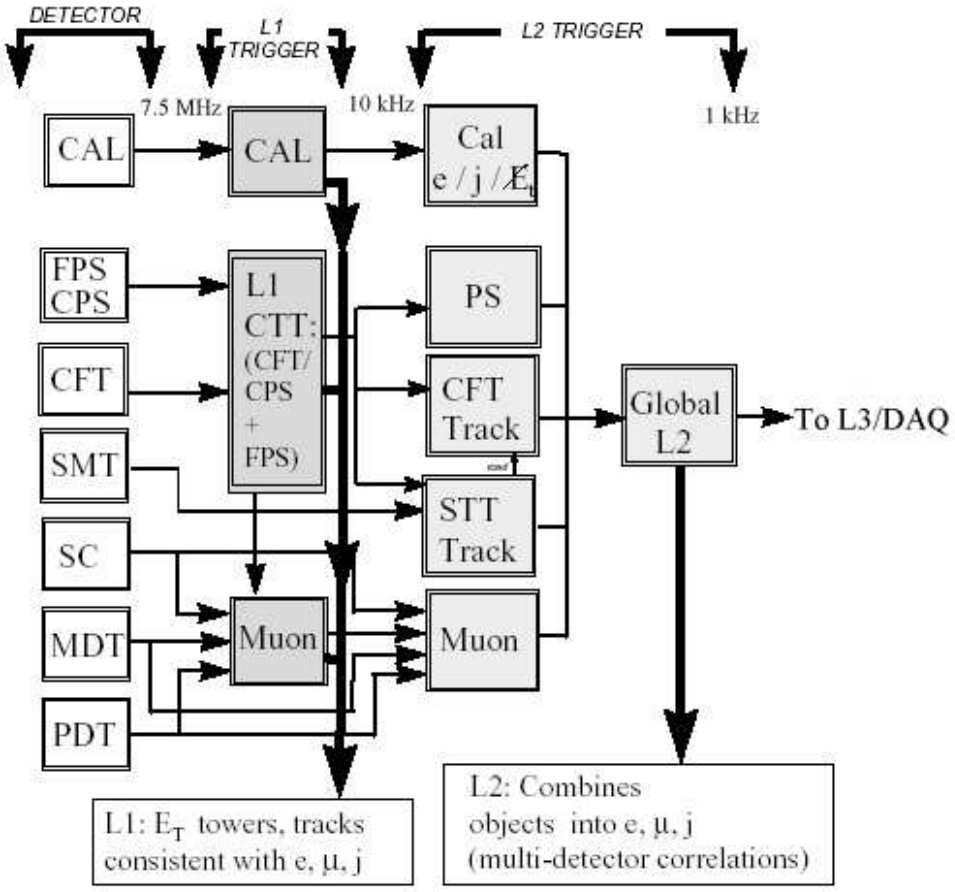


Figure 3.18: Block diagram illustrating the components of the L1 and L2 triggers and the connections between them. [39]

  - Full muon readout achieved using Second Layer Input Computers (SLICs) which reconstruct muons using the full muon detector readout.
  - Combines full readout and input from L1 Muon trigger.
  - Sends results to L2 global processor.

- Calorimeter Preprocessor
  - Has three types of worker; EM, Jet and missing Et.
  - Takes input from L1CAL trigger towers, EM+Hadronic tower seed masks.
  - Runs a clustering algorithm to make EM and Jet clusters and missing Et and sends the results to the L2 global processor.

- Preshower Preprocessors
  - Receives cluster information from axial and stereo EPS and stereo FPS and fiber channels.
  - Converts fiber channels to $\eta - \phi$ co-ordinates for matching with calorimeter clusters and orders them in $\eta - \phi$.

- Silicon Track Trigger Preprocessor
  - Includes SMT data in trigger to improve tracking resolution and provides vertex information.
  - Data enters through the CFT preprocessor.

- CFT Preprocessor
  - Receives tracks from L1CTT.
  - Sorts tracks by $p_T$ and impact parameter.
  - Takes data from L2 STT preprocessor.
  - Outputs all track info to the L2 global processor.

The level 2 global processor assembles preprocessor data into physics objects using software tools to create EM objects, jets, muons, masses and taus. The output of these tools is used by L2 filters to select objects meeting certain criteria depending on which L1 trigger bits were set. Objects that pass the filters set a L2 trigger bit and hence generate a L2 accept.

### 3.8.3 Level 3 Trigger

The Level 3 (L3) Trigger is a software based trigger running on a farm of high performance commercial rack mount x86 architecture based PC nodes using the Linux operating system to collect the entire detectors readout and run software algorithms on it within a 100 ms per event time window, and acting as a Data Acquisition (DAQ) system to write events to tape at a rate of 50 Hz. The Level 3 farm is currently comprised of 82 nodes, with an upgrade planned that will approximately double the capacity. The Level 3 farm is designed so that it can be greatly expanded, limited ultimately by physical space and available networking connections .

The detector data is read in through many parallel VME Buffer Drivers

(VBD) from digitizing electronics crates. The data from the L1 and L2 trigger systems is piped in across another set of VBDs and examined by the L3 Event Tag Generator, which uses the lower level trigger information to decide on the event topology and tags it as a particular type of interaction. The event is then pipelined to a L3 processor node which reconstructs the event using software physics tools to search for and reconstruct physics objects such as jets, electrons etc. Since the physics tools are software [41] it is possible to run various sets of algorithms depending on the structure of the event in the L1/L2 systems, events are filtered using these tools and can be written into specific event streams as required. From the processor node the event information is passed via VBD to the monitoring and data logging system where the event is dispatched to the tape store across a standard Ethernet connection. A schematic showing the L3 Data flow can be found in Fig. 3.19

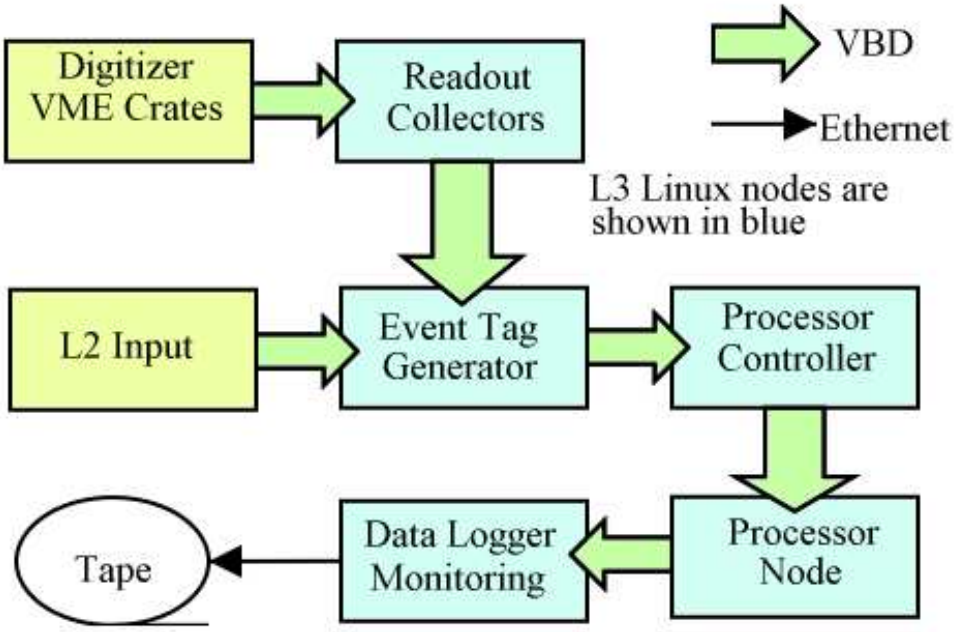


Figure 3.19: Schematic showing the flow of data through the Level 3 software trigger and Data Acquisition system

# Chapter 4

# Level 1 Central Track and Preshower Trigger Digital Front End System

## 4.1 Introduction

With the incredibly high data rate for Run II the Level 1 (L1) trigger has to handle around 1.5 Terabytes of information per second with minimal dead time. Since tracking is a major part of the DØ detector, the trigger must be able to make fast decisions based on information from the tracking detector readout. All the Inner Tracker detector systems use the SVXIIe microchip (SVX) to digitize the readout. A consequence of this is that the full output speed of the Silicon Microstrip Tracker (SMT), Central Fibre Tracker (CFT) and Central and Forward Preshower (CPS and FPS respectively) detectors

are restricted by the 90ns processing time taken by the SVX chip as it converts the analogue detector output into digital signals. This delay is too long to provide a fast trigger based on track objects using only the SVX readout information. An alternate form of fast readout for the trigger system is therefore used to provide fast digital triggering information, while the full SVX readout is sent straight to the final stage of the trigger. The scintillator detectors, the CFT and PS have a very fast readout from the scintillating fibres, and so the output of these detectors is used for a fast trigger pickoff before the SVX digitization stage to produce quick digital signals that can be processed in a large signal processing system to make trigger decisions for the L1 and L2 triggers. The computing system that handles trigger decisions for the scintillator tracking detectors is comprised of digital signal processing devices constructed from programmable logic microchips. These microchips are configured by loading firmware files into them containing triggering algorithms for various segments of the detectors. Since these devices are all constrained to be close to the detector, they are inaccessible during data taking runs, and must be completely configurable via a remote connections. The author's first project on arriving at DØ was to design and implement a command and control system for remotely managing these devices, providing both expert level and shifter level monitoring and diagnostics, combined with a rigorous version control system to make certain that the required trigger algorithms are being used for data taking. This chapter provides an overview of the scintillator detector trigger system components, and a review of the author's design and development work to address the needs of this system.

A review of the author's resulting DFEWare software package is presented including the expansions to the EPICS software and DFEC firmware made to execute the necessary tasks. This software plays a major part in the day-to-day data taking operations of the DØ detector by ensuring that the scintillator trigger system is running smoothly and as expected.

## 4.2 The Digital Front End System

Photon pulses received from the scintillator detectors and converted to an amplified electric charge in the Visible Light Photon Counter (VLPC) cassettes are collected by Analog Front End (AFE) circuit boards for digitization. Before reaching the SVXIIe Analog-Digital Converter chips on the AFE, the signals are passed through a custom microchip which picks off discriminated signals and sends digital information to the Digital Front End (DFE) system. The DFE system is essentially a massively parallel custom supercomputer programmed with several layers of pattern recognition filtering software. The computer devices used to implement this are Field Programmable Gate Arrays (FPGA) running track and cluster finding algorithms at speeds of 200 - 300 MHz for the L1 and L2 trigger systems [43][44]. An overview of the entire DFE system including AFE inputs and L1/L2/L3 output pathways is shown in Fig. 4.1.

The DFE system can be viewed as four distinct processing layers:

- The first layer is the Mixer layer which regroups the signals into $4.5^o\phi$ trigger sectors for axial readout from the CFT and CPS detectors.

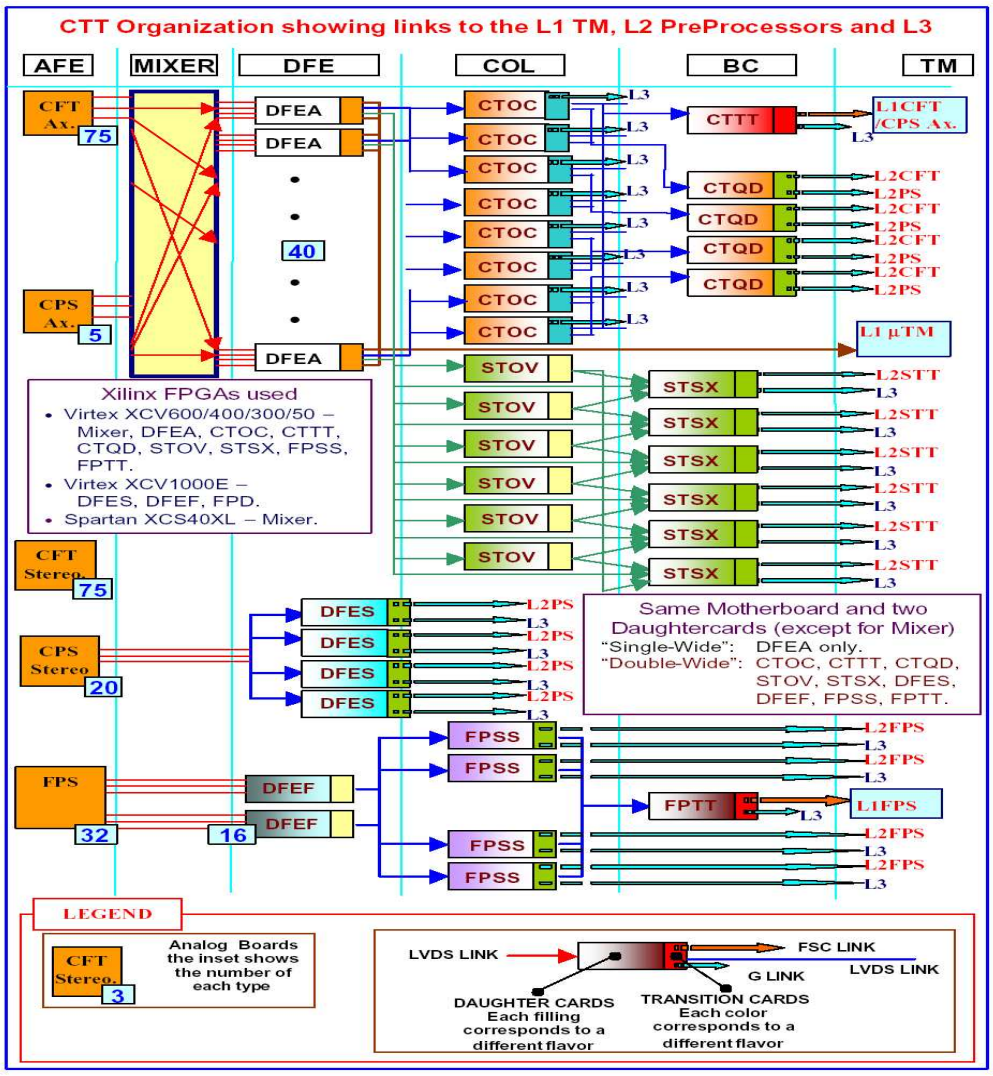


Figure 4.1: Schematic showing the Digital Front End Trigger system. The AFE system is shown in the left column, the four middle columns show the DFE system, and the far right column shows the readout L1 Trigger Managers, L2 preprocessors and L3 system. [42]

The readout at the AFE stage is grouped according to the physical fibre bundles in the detectors, which are arranged in five super sector segments in $\phi$. The Mixers reorganise the individual fibres into the trigger sections. Stereo Fibres are not included in the Mixer system, being read straight into the second layer of the DFE system.

- The second layer is the first digital processing stage which take the raw digital signals and construct coherent information about tracks or clusters in localised areas of the detectors. These are used as the basis for the rest of the DFE system.

- The third layer comprises digital processors that collect the data from the initial DFE system and build information about larger sections of the detector for making full scale trigger decisions.

- The fourth layer collects the input from the large detector sectors and broadcasts the information to the L1 trigger framework and the L2 pre-processors to allow trigger decisions to be made on tracks and clusters.

The readout from the three scintillator detectors (Central Fibre Tracker and Preshower and the the Forward Preshower) is used in three different pathways for trigger decisions. The axial fibre output from the CFT and CPS detectors is combined to form the fast L1 Central Track Trigger (L1CTT), and is used to make fast L1 track trigger decisions, and also feed the L2 track trigger preprocessors for the CFT and SMT. The Stereo output from the CFT and CPS is readout through a different path and is not used for

the L1 trigger, but passed up to the L2 system. The final data pathway uses the output from the FPS detector to form a L1 Forward Preshower (FPS) trigger and provide input to the L2 Preshower preprocessor. Each of these data pathways is described in detail below.

## 4.3 CFT and CPS Axial Digital Front End System

The CFT and CPS Axial DFE system which forms the basis of the L1CTT is shown in Fig. 4.2. The axial fibres are readout by AFE boards which digitize the full signal and send it to the L3 system, while the discriminated SIFT (ScIntillating Fibre Tracker) Chip signals are sent into the DFE system. The first DFE stage in this change is the Mixer stage which repackages the data into 80 azimuthal trigger sectors instead of the fibre bundle format received from the detector as illustrated in Fig. 4.3. Each Trigger sector is a $4.5^o$ slice in $\phi$. The output from the mixers is sent to the Axial DFE signal processors, conventionally named as DFEA boards. Each DFEA contains two sets of five FPGAs arranged on daugherboards, each of which runs track finding algorithms for a single axial trigger sector. The DFEA functions include:

- Finding tracks in predefined track roads in the eight layers of CFT Fibre doublets.
- Organise any tracks found into four major $p_T$ bins.

- Find CPS clusters with scintillator hits above a preset threshold in the CPS Axial strips within the trigger sector, and in the adjacent halves of the neighbouring sectors.
- Provide a total hit count of CFT doublet hits in the trigger sector.
- Provide counts of tracks, track/cluster matches, cluster counts and CFT azimuthal occupancy in terms of doublet hits for the L1 trigger.
- Send the six highest $p_T$ tracks to the L1 Muon trigger.
- On receiving a L2 accept signal, send detailed lists of up to 24 tracks and 8 CPS axial clusters to the next DFE stage for further processing.

The DFEA output is then used as input by several other stages of the DFE system, the first of these being the L1CTT Octant collector (CTOC) boards. Each of the eight CTOC boards handles one eighth of the axial readout as shown in Fig. 4.3 and combines input from 10 DFEA daughter boards (i.e., a wedge of 10 adjacent axial trigger sectors) and performs the following tasks:

- Sum the DFEA count information for the octant.
- Sum and rescale the CFT doublet hit occupancy.
- Send up to five isolated tracks per octant to the next DFE stage for the L1 CTT trigger.

- On receiving a L1 Accept signal, send the DFE quadrant processors detailed lists of up to 24 tracks and up to 24 CPS axial clusters.

- On receiving a L1 Accept, readout all L1 and L2 inputs to the L3 system, if an accept is not generated, then the data is flushed from the trigger system.

The output of the CTOC boards is sent to the L1CTT trigger manager DFE board, which is a single DFE board called the CTTT and is used to form the L1 trigger terms for the L1CTT and sends them to the L1 framework to be used in the L1 trigger decision. The CTTT receives the sum information from the eight CTOC boards and sums the information to create up to 96 trigger terms for the L1 trigger. It also reads out all of its L1 inputs to the L3 system.

The CTOC output is also gathered by the four quadrant DFE boards (named CTQD) which collects information from two neighbouring octants and forms quadrant information for the L2CFT and L2CPS preprocessors. The CTQD has no L1 functionality, being designed purely to provide formatted track information to the L2 system, in the form of lists of tracks and clusters from pairs of CTOC boards. Up to 48 tracks and clusters are sent to the L2 track and CPS preprocessors, and the entire input is read out to L3.

Also being fed from the DFEA boards is the L2 Silicon Track Trigger (L2STT) which provides track triggering combining output from the CFT and SMT detectors. The STT system is comprised of two types of DFE

board, the STOV and STSX boards which reformat the CTT trigger sector information into the SMT sextant geometry. The STOV boards are used to associate CFT tracks with SMT sextants, and the STSX boards correlate tracks which overlap SMT sextants when they are extrapolated into the CFT. Coverage of the STT DFE boards is illustrated in Fig. 4.3.

The STOV provides the following functionality:

- Merge track lists from 6 CTT trigger sectors.
- Pass $p_T$ ordered track lists to the appropriate STSX overlap boards.
- Send Duplicate/Ignore information to STSX boards to indicate if the track was sent to multiple STSXs.

The STSX provides the following functionality:

- Merge track lists from 7 or 8 CTT trigger sectors and from 2 STOV boards.
- Send 48 highest $p_T$ tracks to the L2STT preprocessor.
- Read out all inputs to L3

## 4.4 CFT and CPS Stereo Digital Front End System

The stereo output from the Central Fibre Tracker (CFT) and Central Preshower (CPS) are not used for L1 trigger decisions, in fact the CFT stereo output is

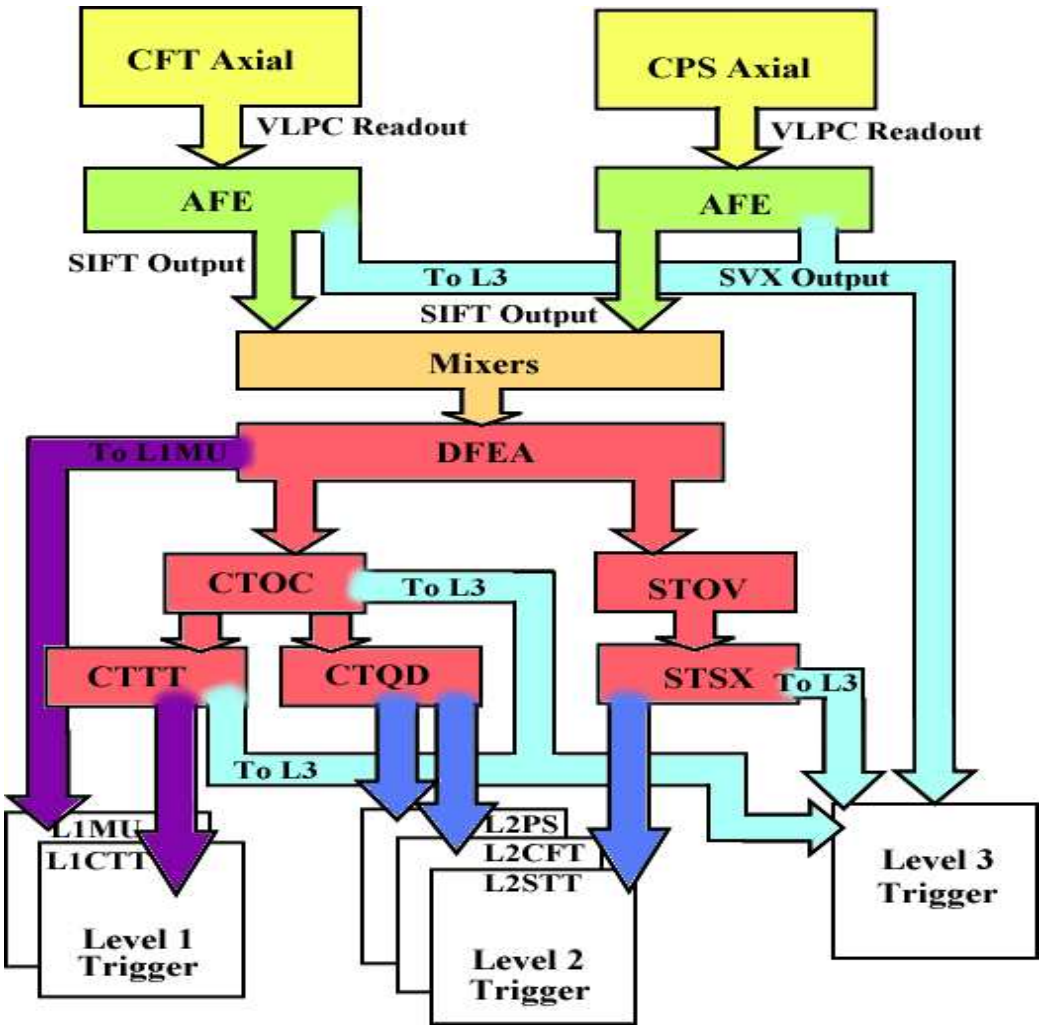


Figure 4.2: Flow diagram for the central axial system, showing the data flow from the detectors through the AFE boards which send discriminated SIFT output to the DFE system for the Central Track Trigger system. Output to the various trigger subsystems is flown. Output to L3 is shown in light blue, L2 in dark blue and L1 in purple. The DFE signal processors are shown in red. For detailed description of the subsystems, the reader is referred to the text.

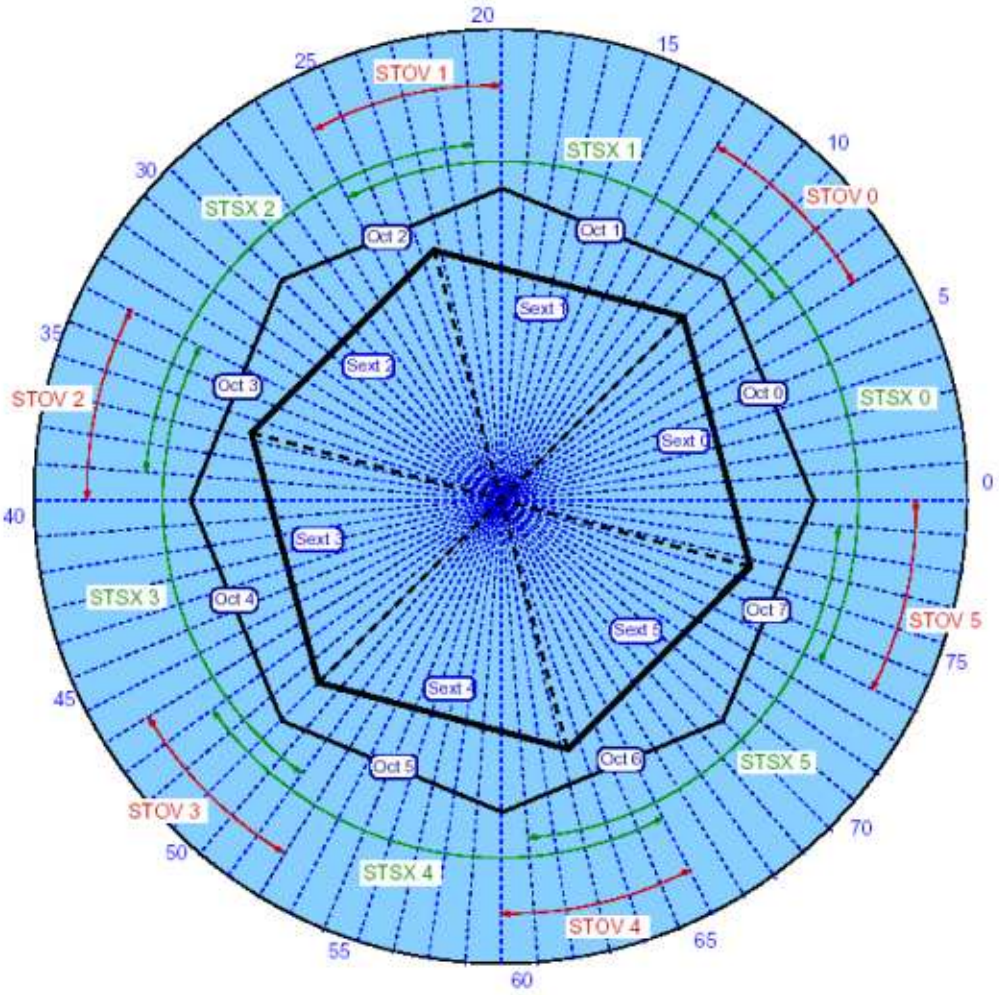


Figure 4.3: Diagram showing the arrangement of the azimuthal trigger sectors used in the CTT system, each $4.5^o$ sector is shown along with the CFT Octants, SMT Sextants and the coverage of the STOV and STSX DFE boards which map the CFT sectors to SMT sextants for the L2STT trigger. [42]

not passed through the DFE system at all but is only sent to L3, as shown in Fig .4.4. The CPS stereo output is routed through a DFE system, for processing and delivery to the L2CPS preprocessor. There are four DFE boards which receive signals, called DFES boards, which are used to find clusters in the CPS stereo U and V offset layers. These clusters are stored in a L1 pipeline buffer, until a L1 Accept/Reject decision is made. Upon a L1 Accept, clusters are formed in contiguous layers in both U and V orientations, up to 24 of each U and V clusters are then reported to the L2CPS preprocessor and the DFES input channels are readout to L3.

## 4.5 FPS Digital Front End System

The Forward Preshower (FPS) DFE system consists of three stages, the DFEF boards which find clusters, the FPSS which collects data from halves if the north and south FPS units, and the FPTT which forms the L1 trigger terms for the FPS trigger manager. The data flow for the FPS DFE system is shown in Fig. 4.5.

The FPS detectors are divided into 32 azimuthal trigger sectors, each DFEF board handles the output from two sectors, the DFEF performs the following tasks:

- Find clusters of contiguous strips above preset thresholds in both U and V stereo orientations.
- Check for possible energy depositions in the FPS MIP layers in both

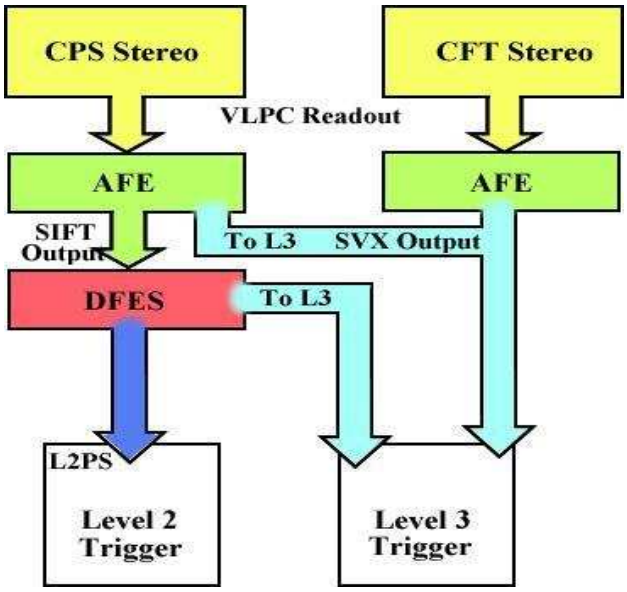


Figure 4.4: Data flow diagram for the central stereo system. Output to the various trigger subsystems is flown. Output to L3 is shown in light blue, L2 in dark blue, there is no L1 input from this subsystem. The DFE signal processors are shown in red. For detailed description of the subsystems, the reader is referred to the text.

U and V stereo orientations.

- Send cluster counts for charged lepton or electromagnetic shower candidates to the FPSS.
- Report detailed list of up to 12 U and 12 V clusters to the FPSS for L2.

The four FPSS boards each receive data from one half of each north and south detector from 4 DFEF boards. The FPSS then forms sums of the various cluster counts from the DFEFs and sends the total counts to the FPTT. The FPSS also merges the lists of clusters and sends up to 24 U and 24 V clusters to the L2FPS preprocessor and reads out all its inputs to L3.

The single FPTT board correlates information from all four FPSS to form a list of up to 96 trigger terms to send to the L1 FPS trigger manager to make L1 decisions. The FPTT is also read out to L3.

## 4.6 DFE Hardware

The DFE system is essentially seven electronics crates holding custom designed circuit boards to implement FPGA programmable logic. Each crate of DFE boards is managed by a controller board and all control and communication is done via a serial data bus using a communication protocol known as 1553.

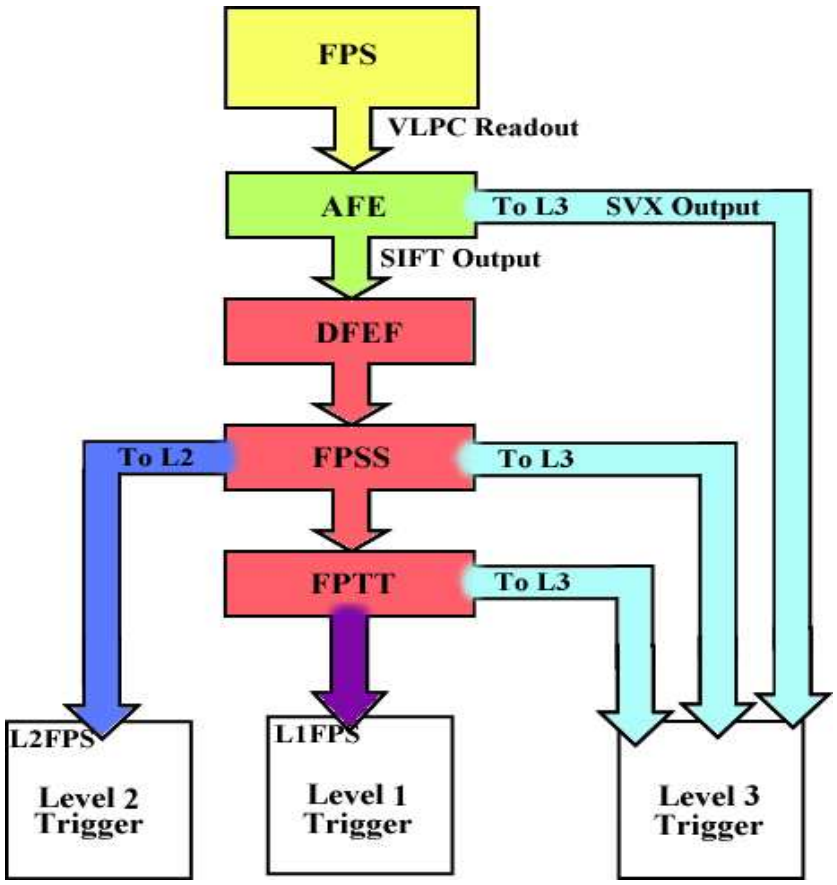


Figure 4.5: Data flow diagram for the Forward Preshower (FPS) system. Output to the various trigger subsystems is shown. Output to L3 is shown in light blue, L2 in dark blue and L1 in purple. The DFE signal processors are shown in red. For detailed description of the subsystems, the reader is referred to the text.

### 4.6.1 FPGA logic devices

The FPGAs used in the DFE system are all manufactured by Xilinx [50], and share a common chip footprint, allowing different FPGA chips to share a common socket. This allows the use of a common design for most of the DFE boards. The FPGAs themselves vary in processing speed and power, allowing the DFE boards to achieve the performance required in the L1 trigger system. The FPGA chips are loaded with a compiled firmware file, which contains the code to allow the FPGA to run its algorithm along with any threshold values required. The compiled configuration files (hereafter referred to as firmware files) are created by compiling Very High Definition Language (VHDL) format code with a Xilinx compiler to produce a hexadecimal file which can be loaded and understood in the FPGA device. The programmable nature of the chips allows flexibility for implementation of the actual triggering algorithms used, and easy maintenance due to the commonality between devices, something not possible with a custom built hardware electronics system.

### 4.6.2 DFE Motherboard Construction

The DFE Motherboards for each type are essentially the same design to allow a variable number of FPGA devices to be installed on the board and communicate with each other as required. All I/O traffic is performed via common connectors and configuration for the on board FPGA chips is loaded via a crate back plane interface from a Controller board. This means that

DFE boards performing very different functions within the trigger can all exist in the same electronics crate with a common control interface. Taking advantage of the common socket footprint of the FPGA chips used allows using higher powered FPGAs where needed and lower power FPGAs where appropriate to provide the specialisation needed for the various trigger processing tasks, but with an identical control and physical environment, thanks to the common DFE Motherboard design.

### 4.6.3 DFE Crate Controllers

Each crate of DFE boards is controlled by a DFE Controller (DFEC) board in the first slot of the crate. The DFEC is responsible for managing the DFE boards in the crate to set them up for data processing and provide continuous real time control and monitoring of the DFE boards.

There are three main goals of the DFEC:

- Fast Configuration of the FPGA devices on all the DFE boards in the crate in under a minute.
- Control of high speed data processing bus via 1553 control network.
- Continuous slow monitoring information readout for all DFE boards in the crate.

Each crate is capable of holding up to 20 DFE boards, and each board can contain up to sixteen FPGAs, the DFEC is responsible for initialising each

FPGA with the appropriate firmware configuration, requiring it to manage firmware files for hundreds of devices in a fast, efficient manner.

The DFEC contains several elements to enable it to perform its task.

- **Non-volatile memory** The DFEC uses a Compact Flash [51] memory card for storage of FPGA firmware configuration files in the crate. Compact Flash memory is a robust memory device that supports features such as bad sector marking, data integrity checking and self test capabilities, and conforms to the standard ATA/IDE interface, making the device appear like a small hard disk drive on the DFEC. The Flash memory is formatted using an MSDOS FAT 16 filesystem to organise file storage allowing files to be added via the 1553 remote interface and from a standard PC with a commercial Compact Flash memory reader. Owing to the envisioned size and number of files used by the DFE boards, Compact Flash memory cards ranging from 64 to 160 MB in capacity were used. Files stored on the DFEC Flash memory are named with a sixteen bit hexadecimal name so that they can be easily referenced using the 1553 remote control protocol.

- **Microcontroller** The DFEC is built around a small micro processor chip running at 6 MHz to execute instructions. The microcontroller is supported by an FPGA which handles all the logic required to connect the microcontroller to the flash, back plane, 1553 bus and so forth. The FPGA also contains a dual port RAM memory, which is used by the microcontroller as system memory. The RAM is divided into various

sections which act as a command buffer for the microcontroller, a slow monitor readout and a memory cache for transferring files to and from the Compact Flash memory. A map of the physical memory address allocation within the DFEC RAM can be seen in Fig. 4.8 in section 4.7.2.

- **1553 Interface** All communication between the DFEC and the outside world occurs using the 1553 protocol bus. The 1553 bus uses the dual port memory to read and write commands and data and generates an interrupt on the microcontroller to execute commands written into the memory. The 1553 bus is a radiation hard, low emission technology that can be used in high radiation areas without interfering with complex electronics. The 1553 system is an older standard, and was originally selected for Run I operations. The interface was selected for Run II remote access tasks mainly due to the legacy of equiptment and experience from Run I.

- **Slow Monitoring Interface** Status information from each DFE board is collected from a dedicated slow monitoring serial bus and is continuously collected and written to a block of the dual port memory, where it can be accessed by the 1553 bus.

- **DFE Back plane Bus** The DFEC communicates with the DFE boards in the crate via a simple 16 bit address bus, which is used to load firmware files to the FPGAs and issue commands to the DFE boards.

The DFEC Firmware in the microcontroller defines a list of commands which can be issued to the DFEC by writing the appropriate command identifier and arguments as a set of hexadecimal values to the command section of the DFEC memory and creating an interrupt to cause the DFEC to read and process the commands. A list of up to 30 commands can be executed in a single sequence. Access to the DFEC memory is provided via a set of three sub-addresses on the 1553 bus terminal connection. The first sub-address, Sub-address 0x16 (Sub16) is used as an address pointer to point to an address in the DFEC memory. Data is then written to the next sub-address, Sub-address 0x17 (Sub17) and this is written into the DFEC memory at the location provided by Sub16. The last sub-address used is Sub-Address 0x18 (Sub18) which can be read from to retrieve the DFEC Status information or written to, in order to generate a DFEC command interrupt to make the DFEC execute the instructions in its command memory segment.

The commands implemented in the DFEC firmware are very basic low level commands and some high level commands built from multiple low level commands. A broad overview of the commands is given here, details can be found in Ref [46].

- **Low level commands**
  - Write a byte to the DFE back plane interface.
  - Read a byte from the DFE back plane interface.
  - Transfer a Compact Flash sector to the memory sector data buffer.

  - Transfer the contents of the memory sector data buffer to the Compact Flash.
  - Transfer a file from the Compact Flash to the DFE Back plane interface.

- **High level commands**
  - Delete a file from the Compact Flash
  - Append the data from the memory sector data buffer to a file on the Compact Flash
  - Generate a Checksum for a file on the Compact Flash
  - Reset a DFE motherboard
  - Set the Clock select bits for a DFE motherboard
  - Configure a DFE FPGA device with a file from the Compact Flash
  - Return the firmware version information from an FPGA device
  - No-operation or Null command
  - Return the size of a file on the Compact Flash
  - End of command list marker, to signify the end of a command list.

The Download of firmware files to the DFEC Compact Flash memory card has to be done in stages by writing 512 byte chunks of the file to the DFEC Sector Data Buffer section of the RAM memory and then moving the contents of the buffer to the flash. Due to the serial nature of the 1553 interface used to communicate with the DFEC, the information in each byte

of the firmware file needs to be "nibble-swapped" (swap the first four and last four bits) when sent to the DFEC so that it is unpacked into the DFEC memory in the correct order.

## 4.7 DFEC Communication

Since the electronics crates are all in the collision hall, remote access to the DFE controls is required. This implies that complete reconfiguration of all FPGAs must be able to be done remotely, including all command and control functions. To enable this each DFEC is connected to the 1553 network to allow direct access to the DFE electronics from the online computing system. A schematic of the Hardware and software connections used to control the DFE System is shown in Fig. 4.6 The various components of this diagram are described in the following sections.

### 4.7.1 The 1553 Protocol

All remote communication with the DFEC is done via the 1553 communication protocol. The MIL 1553 protocol is a serial data transfer bus protocol originally developed for military applications in aircraft avionics, which provides a uniform way of connecting multiple devices to the bus and allowing them to communicate at a speed of approximately 500 Mbps. The 1553 bus cards reside in VME crates along with an I/O control node (IOC) running the VxWorks Operating System, which provides 1553 I/O across the crate back plane. The controller can support multiple 1553 buses, and is installed

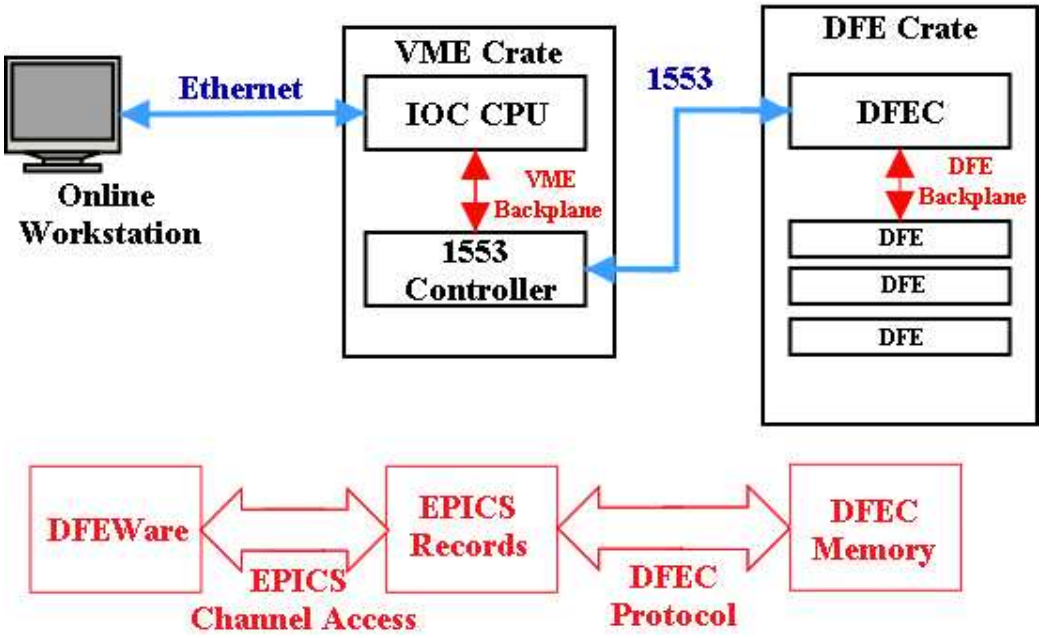


Figure 4.6: Diagram showing the hardware (top) and software (bottom) connections between the user and the DFEC system. The user works on a linux workstation and uses the DFEWare software package to communicate with the DFEC, via the IOC processor and 1553 bus controller. The IOC uses the EPICS system to provide interfaces to the DFEC command protocol over the 1553 bus. The details of each part are described in detail in the text. [45]

as a node in the DØ online computing system, allowing access to the 1553 system from the online nodes. Each 1553 bus has multiple addresses, corresponding to some device on the 1553 network, each DFEC appears on the 1553 network as a single remote terminal, that can be accessed by the online system via the IOC. Many subsystems from various detectors use the 1553 system to communicate with detector electronics, so the entire 1553 system is regulated using a database like control system, called EPICS.

### 4.7.2 EPICS

The Experimental Physics and Industrial Control System, or EPICS [52] software is a set of software tools that are designed for building distributed control systems for large complicated computer and electronic systems, such as particle physics experiments like DØ . EPICS is an architecture of front end controllers and operator workstations that communicate via a TCP/IP based protocol using a client/server and publish/subscribe model to provide direct data access between any two points in the system. EPICS Servers perform device level control and I/O tasks, and publish this information to clients using the Channel Access (CA) network protocol in real time. All the device level support is included in the server, so that the information is presented as a set of EPICS record fields, that can then be accessed by the client software. A diagram showing an example of how EPICS interfaces devices can be seen in Fig. 4.7

Each Device has an EPICS record database which contains fields that

provide access to parameters associated with the device, allowing I/O to the device via the EPICS Record.

For the DØ DFE System, all the 1553 bus devices are accessed via EPICS as an I/O Device, and a set of EPICS records are implemented to provide access to the DFEC Memory, thus allowing commands to be issued and information to be transmitted/received between the workstations of the online computer systems and the DFEC boards on the platform. All DFEC operations are therefore built from a combination of EPICS record transactions. Most of the DFEC transactions are straightforward reads or writes of values to areas of the DFEC memory, the exception to this is the download of firmware files which are handled by a C subroutine on the IOC processor to move the file sector-by-sector from a disk area on the online machines to the flash memory on the DFEC. This operation is triggered by a EPICS record which passes the name of the file to the subroutine. Since a download typically takes a few minutes to complete, EPICS records to monitor the download are provided by the IOC processor. The list of EPICS records provided for each DFEC is provided in Table 4.1. A logical map of the EPICS records to DFEC memory segments is shown in Fig .4.8.

## 4.8 DFEWare Software

The software package to provide control and monitoring facilities for the DFE System is called DFEWare, and is implemented in the Object Oriented

| EPICS Record | Purpose |
|---|---|
| STATUS | Read the DFEC Status from Sub18 |
| GO | Write to Sub18 to generate a DFEC interrupt |
| CB:W | Write to the command block of the DFEC memory |
| CB:R | Read the contents of the command block of the DFEC memory |
| BSTAT | Read the contents of the DFE monitoring block of the DFEC memory |
| GPM | Read the contents of the DFEC General Purpose memory block |
| FUNC | Trigger the download of a firmware file to the DFEC Flash memory |
| DLOAD | Flag for file download in progress |
| TSECT | Total Number of file sectors for download in progress |
| CSECT | Number of sectors currently downloaded successfully for download in progress |

Table 4.1: Table of EPICS records used for each DFEC, along with a brief description of the purpose of each record.

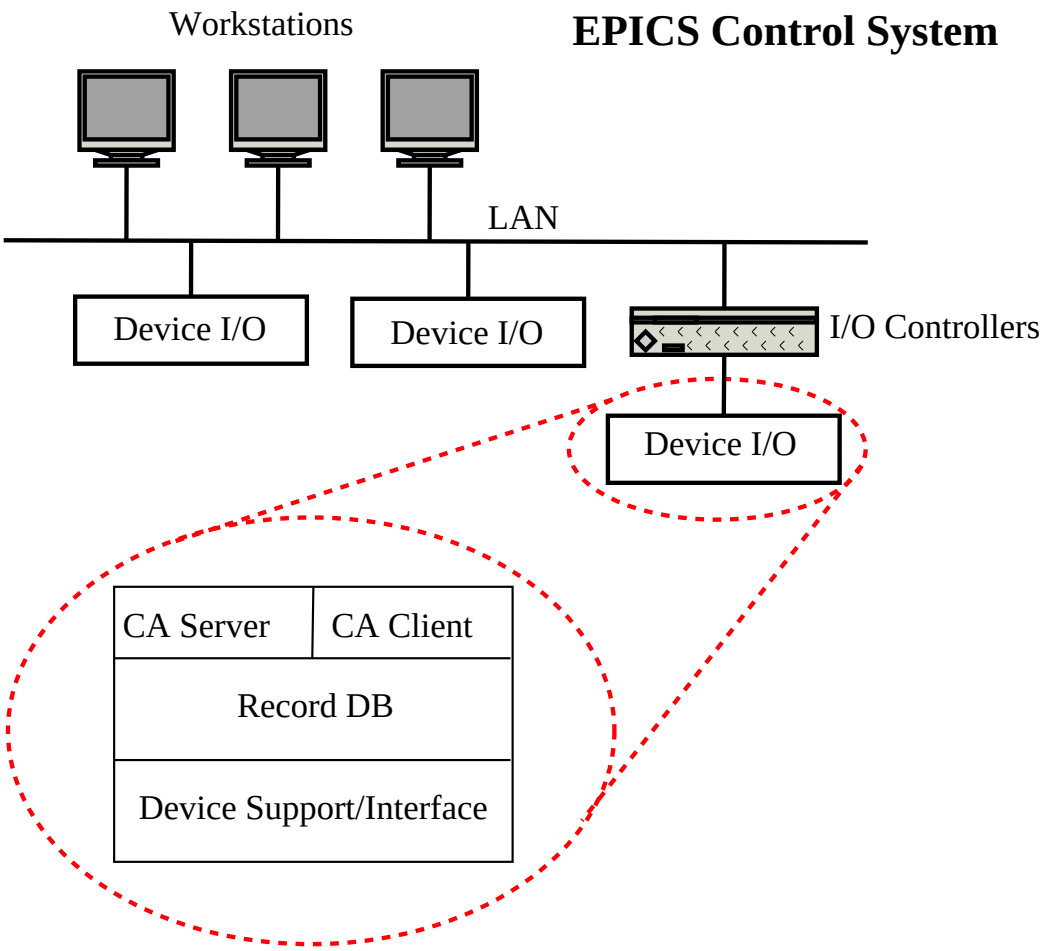


Figure 4.7: Diagram showing how the EPICS control system links workstations and I/O devices via a TCP/IP based network. Each I/O Device interface (shown as inset) contains device support for the particular hardware in use, and a set of EPICS record database fields that contain some information about the device. A Channel Access (CA) Server allows other CA Clients to access this information in real time, while a CA Client allows the device to access information from other CA Servers in the system. *Adapted from various diagrams from [52]*

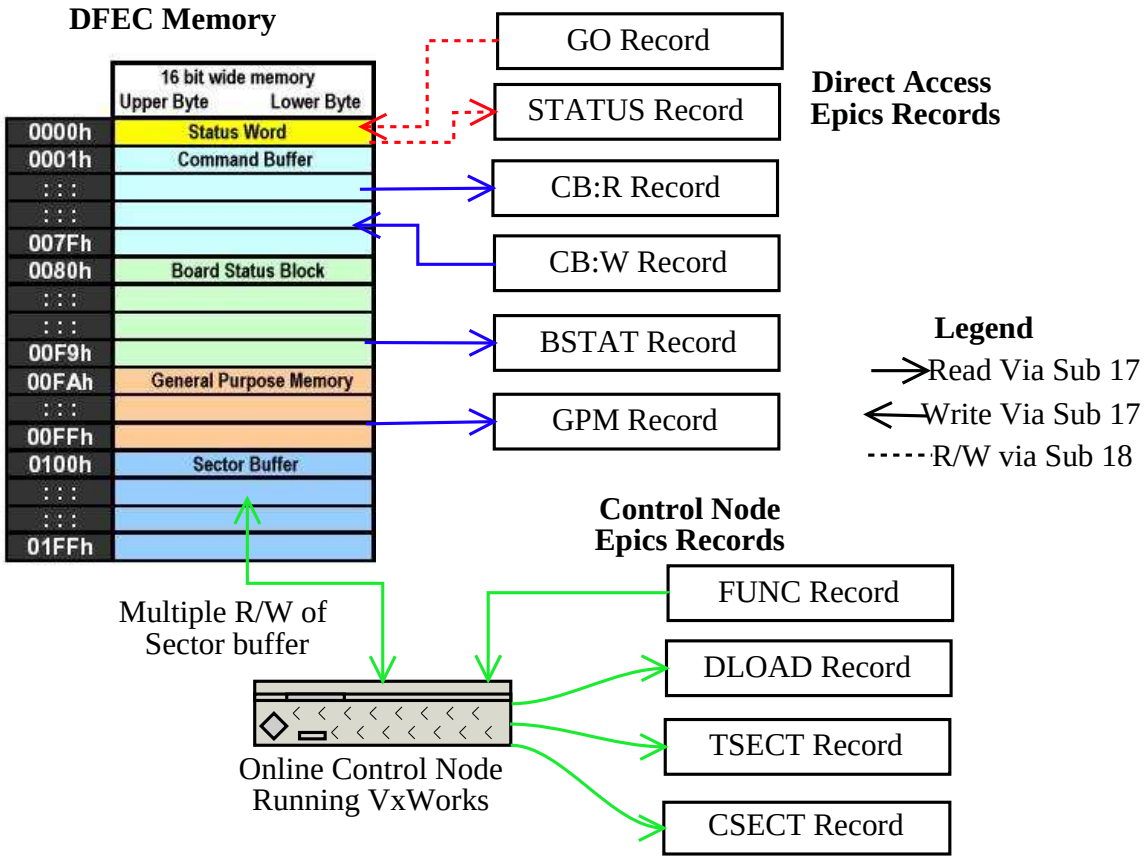


Figure 4.8: Diagram showing the DFEC dual port memory addressing scheme along with the EPICS records and the logical connections between the two. The large data transfers involved with downloading an FPGA Firmware file is handled by a C routine on the IOC processor that is triggered by an EPICS record. A description of each record is given in the text.

Python Programming language. The software provides a variety of tools to allow both expert and non-expert to perform DFE maintenance tasks when required, in the form of Graphical User Interfaces (GUI) and shell like command line functions. The basic functionality for the software is to allow execution of the DFEC Command set from the online computing system. These low level DFEC commands are then used to form high level interface commands that can be easily called by users of the software to perform monitoring and maintenance tasks. DFEWare also has to manage multiple connections to each DFEC from multiple clients and handle the requests from these clients in a robust, dependable manner.

There are a total of seven DFEC crates in use for the detector and several test crates which may or may not be in use at any given time. The cabling of the detectors determines the location of DFE boards within the DFE electronic crates so the allocation of the different types of DFE boards throughout the crate are predetermined. Additionally, the DFE boards also report their type to the DFEC monitors.

DFEWare uses the CaChannel[53] python package to access the EPICS record for each DFEC. This allows direct manipulation of each record via a CaChannel Python object.

The goals of DFEWare are fourfold:

1. Provide a common API interface to each DFEC controller that can be used by any software package to interface to the DFE system.

2. Provide expert level tools for any necessary maintenance that may arise

during detector operations when physical access to the DFE crates is not possible.

3. Provide a monitoring and simple control interface that can be used by non-expert detector shifters.

4. Provide accounting and version information the various run control databases to keep track of the firmware running for each data taking run.

An overview of the solutions to each of these points are now presented.

### 4.8.1 DFEWare DFEC API

The DFEC API is designed so that a DFEC object is instantiated for the required DFEC device to be used. This object connects to all the DFEC EPICS Records for the particular DFEC and maintains them for the duration of its existence. The Object provides access to all of the low level functionality of the DFEC and also provides high level interface methods that are more useful to external applications. The DFEC object maintains only its connection to the DFEC, it does not periodically update information from the EPICS records, this is done on demand by higher level software. The low level interface to the EPICS records is mostly done via hexadecimal values and bit patterns, so there are several features to represent the information as easily interpreted object attributes and methods to extract information from the hex values and present it as python data structures such as strings, lists

and tuples etc. The DFEC object is designed such that the DFEC status itself gives an overall glance at the operation of the crate as a whole, with information about the DFE boards themselves and the FPGA device on the boards being available through a tree-like structure, as shown in Fig. 4.9. Methods are provided to access information in the tree structure as needed, as well as inspection methods to determine the nature of the DFEC tree itself.

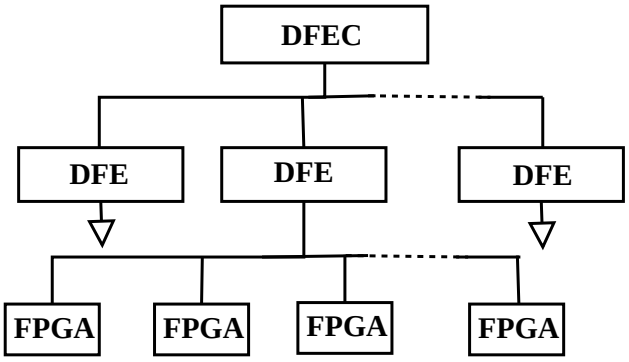


Figure 4.9: Figure showing the tree-like hierarchy of information available through the DFEC interface. The DFEC provides a broad status of all the DFE boards, which in turn provide an overview of their devices. Detailed information can be retrieved by traveling down the tree branches to the desired device.

Full documentation on the range of commands implemented in the DFEC interface is provided on the DFEWare webpage[54].

### 4.8.2 DFEWare Expert Level Tools

The expert level tools make use of the DFEC API described in the previous section to allow the electronics and trigger experts to access the details of individual devices and issue commands to them. There are several of these tools including a direct byte level read/write interface to the DFE boards over the back plane using low level DFEC commands to communicate directly to the motherboards and examine the byte level output. Several link test utilities also exist which allow the channels between various DFE boards to be tested by sending fake test signals that are well known and examining the reaction to these test signals throughout the DFE data processing chains described in §4.3, §4.4 and §4.5.

A text based macro interface is also supplied to allow experts to easily run large scale operations on many DFECs, such as downloading many firmware files or running system wide diagnostics. Such operations would be tedious and time consuming to do from GUI tools, so the macro processor allows commands to be expressed as a list of tasks for a particular DFEC, with logical connections to any DFEC, DFE board or FPGA device available, and an interface to the DFEC API commands for the connected device. This also has the advantage of keeping sets of macro files available to easily run large command sets, and simply running a macro is a task that can be started easily by a non expert if necessary.

### 4.8.3 DFEWare Shifter Level Tools

For monitoring of the DFE system during data taking detector runs, an easy to understand Graphical User Interface (GUI) is needed that supplies information on the state of the DFE boards and their devices in a way that clearly indicates error conditions in an obvious way for the detector shifters who are usually not experts on the DFE system. The main DFEWare GUI is essentially a set of sub-GUIs, one for each DFEC in use, that provide a simple interface showing the overall state of the DFEC using a series of LED type display entries, an example of which can be seen in Fig. 4.10. The information is kept to a fairly low level, but clicking on the DFEC pane brings up a menu of options for inspecting the DFE boards and devices in the crate and launching tasks. These include showing a fully detailed board-by-board account of the crate, including detailed information from the motherboards, an example of which is shown in Fig. 4.11. A simple at-a-glance display of all the boards is also available, showing a snapshot of the overall status of the FPGAs via a colour coded button panel. Maintenance tasks are performed using query/response wizard style GUI panes that guide the user through the steps required to perform tasks like downloading a new configuration file or initialising an FPGA. The colour scheme was designed to be visually interesting to the user and also provide very high contrast warning signals that are easy to notice. Thus, normal working states are indicated by steady, low level coloured LED displays, whereas error conditions are represented using a high visibility flashing LED display. Numerous extensions to the

basic DFEWare display are available to the user to display various features of the DFE systems, such as a sector based display showing the states of all 80 DFEA boards, as shown in Fig. 4.12.

A periodically updating histogram style display showing the state of all platform DFE crates is also provided for shifters to provide a fast, up to the minute percentage level display for each crate, which clearly indicates if there is a problem in the crate, alerting the shifter to a need for a more involved inspection for any crate which is not fully initialised for data processing.


Figure 4.10: Snapshot of the a DFEC GUI display for a single DFEC, showing, from left to right, the busy, idle and error indicator LEDs, the name of the DFEC, five error diagnostic LED displays and the download-in-progress display. Error conditions are signified by high visibility blinking red LED displays. [54]

### 4.8.4 DFEWare Accounting Tools

Due to the highly configurable nature of the DFE system, it is necessary to keep track of what firmware is actually in use, since this can have a direct impact on the physics results taken in a given run. This involves managing firmware files at several levels:


Figure 4.11: Snapshot of a DFE GUI display for an individual DFE Motherboard, showing, from left to right, the DFE board type, crate slot number, clock input and a customisable debug byte value, along with a Ready LED that indicates all of the boards FPGAs are initialised and capable of processing data, and a voltage regulator indicator. From this display a more detailed view of each devices configuration details can be launched. [54]

- Firmware loaded into the FPGAs
- Firmware downloaded onto the DFEC Compact Flash
- Firmware stored on disk on the online computing system

With 582 FPGAs being used in the DFE System that sets the minimum number of files to be managed. Additionally, two or more firmware configurations for each device are usually stored on the Compact Flash memory so that fast firmware changes can be performed. Many more versions of each firmware file are stored on disk as active development is performed to improve the algorithms used, as well as test and debug firmware leads to potentially thousands of file versions to be tracked in such a way that the firmware used for data taking for any given event can be found and inspected. Thus, firmware version control is a major issue that needs to be addressed and handled by the DFEWare software.

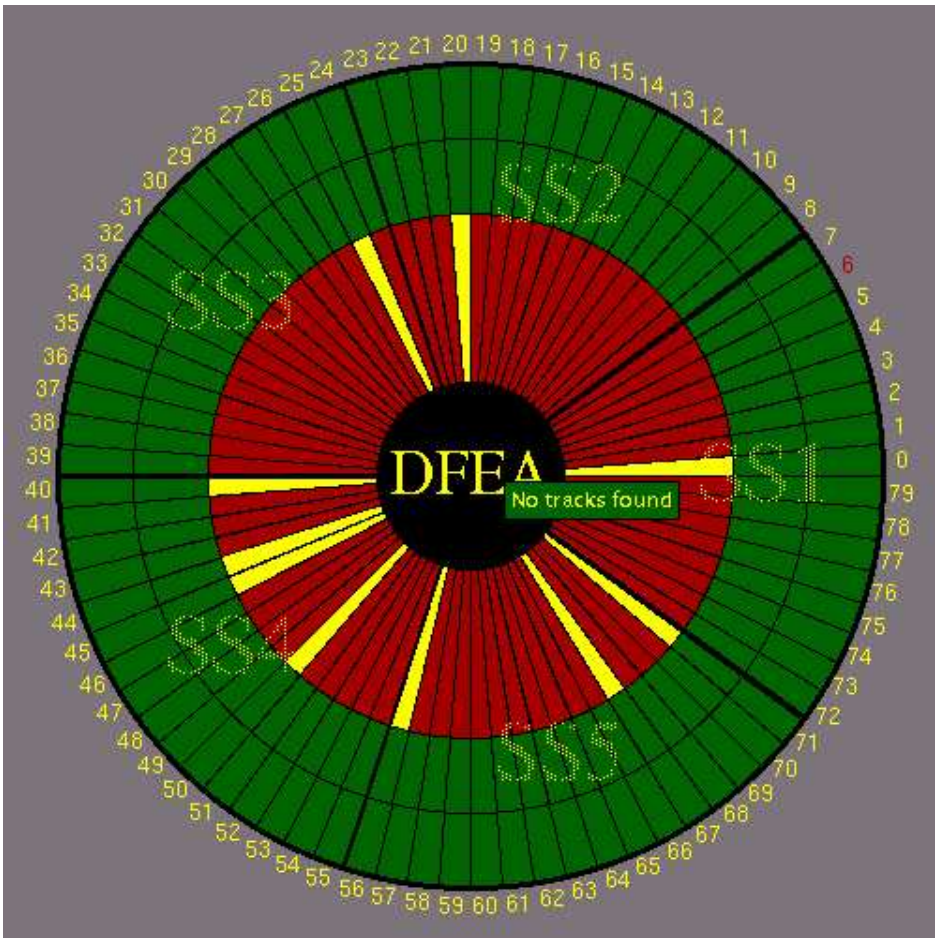


Figure 4.12: Snapshot of the DFEA GUI monitoring display, which shows the states of the DFEA boards in a manner that reflects their physical detector sectors. This is an example of using the DFEC API to extract information from the DFEA boards and display it in a manner which easily shows the actual impact on physics data taking. [54]

DFEWare contains several tools for managing file versions including a file naming system for files on disk, on the DFEC Compact Flash, and loaded into the FPGA devices, and tracks changes in these via a simple text based database system. Any file transaction is logged in this database system and only files within the database are available for use in production. The version control consists of three major parts:

- **Firmware Database Interface** This is a python interface object to the firmware database files. The DFEC API uses this to initially learn which files are stored where for a given DFEC, and updates the database as changes are made, such as file downloads, deleting files from the Compact Flash and loading files into the FPGAs. A catalogue of files on disk and available for download is also kept, with firmware developers required to perform a DFEWare file "check-in" procedure to make DFEWare aware of new files available for download, this procedure includes recording the file's checksum value for file consistency checks after a download. The Database interface strives to be independent of the underlying information storage mechanism, so that a real database, such as the official DØ Trigger Database can eventually be used instead of the text based system. The Trigger version used to take the data is stored in the events as they are recorded, and this can be used to look up what firmware was used from the Trigger Database.

- **Versioning Conventions** Strict versioning conventions are imposed on the firmware files, to make sure that files can be identified in the

various storage mediums. On disk, the files are named and looked up according to a convention that contains information in delimited fields. On the Compact Flash, file names that can be accessed via the 1553 interface are limited to a 16 bit integer value, so the 16 bit name must be traced back to the file name on disk. When a file is loaded into an FPGA device, only 7 bits are available for a version number, and this information is again, required to point back to the 16 but file identifier on the flash. The versioning schemes are given in Table 4.2.

- **Version Query Interface** The DFEC API provides methods that query the version numbers for files on disk, on the Compact Flash and loaded into the FPGAs directly. This can be used to verify firmware versions against the Firmware Database table. It is also envisioned to be used for the start-of-run health checks to make sure that the DFE system is ready to process data and that all the firmware loaded is the appropriate firmware for a given data run.

## 4.9 Summary

The DFE System has been fully installed and commissioned, and is now becoming an integral part of the DØ Trigger. Firmware is being refined and tested to improve the various fitting algorithms for track finding, and these versions are being managed with the DFEWare software package. The CTT was officially declared fully operational on April 15th 2003, and a sample

| Filename Type | Filename Structure |
|---|---|
| On Disk | <type>_<rack>_<crate>_<slot>_U<device>_... |
| | ...v<TriggerDB version>.<7 bit version> |
| On Flash | <slot><device><7 bit version> |
| (16 bit name) | |
| In Device | <test flag bit><7 bit version> |
| (8 bit register) | |

Table 4.2: Table showing the filename and versioning conventions and how they refer to each other. The On Disk filenames are very verbose, containing information on the crate and rack, as well as sub fields that are used to generate the lower detail names. The On Flash name is a 16 bit value, containing the 5 bit slot number (02-21), the 4 bit device number (0-15), and the 7 bit firmware revision version. The FPGA contains only the seven bit firmware revision number, plus a test flag for denoting debug firmware. [55]. Firmware versions are reset to zero whenever a new global trigger version is used, thus the seven bit version number is more than adequate to distingush between sub-versions during a run.

display from the online CTT monitoring software in Fig. 4.13 showing the DFE system in action. DFEWare has undergone steady expansion since it was originally written and has been in regular use in the DØ Control Room, providing shifters with easy access to DFE System information, and coupled with an ever expanding knowledge base allows easy maintenance of the DFE System. Experts use DFEWare for large scale system maintenance tasks and also inspecting the details of what is happening in the various stages of the DFE hardware chain. External interest in the DFE System has also attracted industrial interest, including an article in the Xilinx Journal [56]and a paper to be submitted to the IEEE [57]. The DFE System is making an important contribution to the DØ experiment and will be instrumental in helping to yield interesting physics results for the duration of Run II.

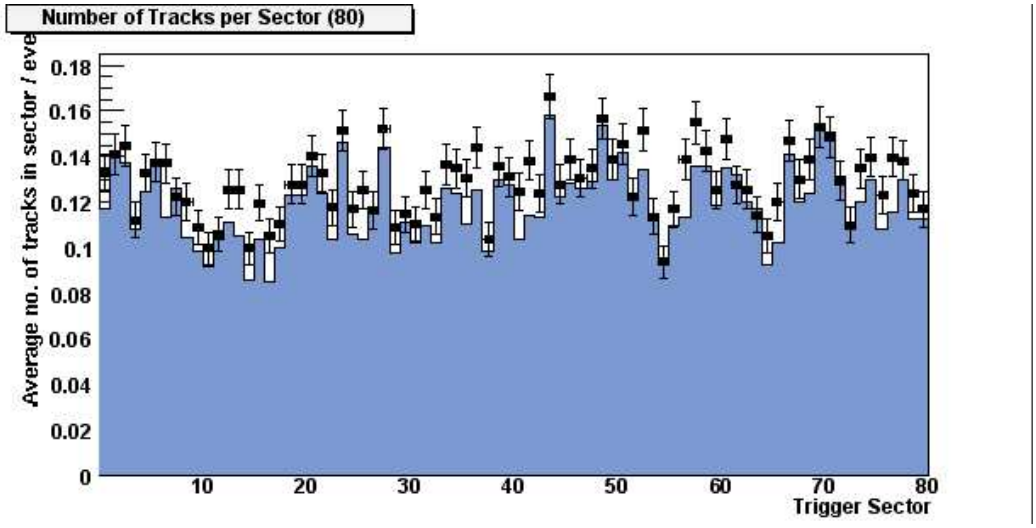


Figure 4.13: CTT Examine plot taken during a physics run showing the number of tracks per sector found in the CTT Axial DFE System.

# Chapter 5

# A Workflow Manager for Large Scale Monte Carlo Production

## 5.1 Introduction

This Chapter reviews the simulated data production and storage mechanisms at DØ. An overview of the DØ code offline software framework and data format is presented along with a discussion of the issues involved with mass production of Monte Carlo (MC) simulation events, whilst ensuring the simulations are accurate, reproducible and catalogued in a standard manner. The MC production software is reviewed and a description of the mc_runjob work flow planner software package used to manage it. The DØ data storage and cataloguing system (SAM) is described including its interface with the mc_runjob and mc_request job management software, and the tools required to run DØ software in a grid environment. The development, maintainence

and expansion of thr mc_runjob package formed the core contribution of the author to the experiment. Given the limited physics oppurtunities during the upgrade cycle and the need for effective distributed computing power, this work occupied most of the author's time. The result is a highly effective remote and local computing effort which has lead to improvements in most areas of HEP computing for the DØ collaboration, including the software and data handling systems. The mc_runjob package is now being propagated to other experiments which are benefitting from the development work pioneered by the author. Although many collaborators are unaware of the fact, most analyses that have large processing requirements will have been handled by the workflow software developed by the author.

## 5.2 The DØ Code

The DØ code is written in the C++ Programming Language, and provides an Object-Oriented (OO) software environment for data containment and processing in a standard manner across the experiment. Basic data storage objects are defined and implemented by a set of software packages called the Event Data Model (EDM) and processing of these data objects is handled by the DØ Framework [58].

### 5.2.1 The DØ Event Data Model

The DØ EDM is a set of objects based on the DØ Object Model (`DØom`) specification that contains generic data storage classes. The EDM is composed

of three distinct parts.

1. `Identifiers`

   The Identifiers package provides objects which identify the various components of, or related to, a physics event without introducing dependencies between the objects. It contains objects like unique collision identifiers and chunk identifiers to distinguish data containers within an event.

2. `EDM`

   The EDM package itself contains fundamental classes such as the Event class and the abstract data Chunk class information containers, and provide the objects for accessing and manipulating them. The Event class contains all the information corresponding to a single $p\overline{p}$ interaction as measured in the detector. This includes raw detector readout data, trigger data, MC data, GEANT simulation data and so forth. Each particular piece of information resides within a Chunk object inside the Event object. The Abs Chunk class in the EDM package is an abstract base class from which all specific data containing Chunks are derived, allowing a common interface for all data Chunks used as persistent containers for a related set of data within the Event.

3. `RCP`

   The Run Control Parameter (RCP) package [59] contains classes for RCP objects which configure the flow of events and processing in the

DØ framework. RCP files (RCPs) are formatted text files containing inputs to user packages. They are catalogued in an RCP database through an interface built into this package.

### 5.2.2 The DØ Framework

The DØ Framework is a set of classes that provides a common interface to Event data objects for processing and analysis. It is designed so that users can write application program modules that inherit from a generic framework object which provides access to the Events. The framework controls Event Input/Output (IO) and creates an event processing stream which the user controls via the RCP interface. Interface methods are provided throughout the processing stream by the framework allowing the user access to the events at any point in the stream. The framework provides basic IO tools to allow the reading, writing and creation of Events during processing. Multiple user processing packages can be run in any given order, as controlled by a master RCP file for the process. Each user package can have any number of control/input parameters which are read from other RCP files as defined within the individual package. A schematic diagram showing the framework processing flow for EDM Events is shown in Fig. 5.1.

## 5.3 Event Simulation

Simulated events form a key ingredient in all DØ analyses. These events are produced by using Monte Carlo (MC) stochastic modeling techniques to

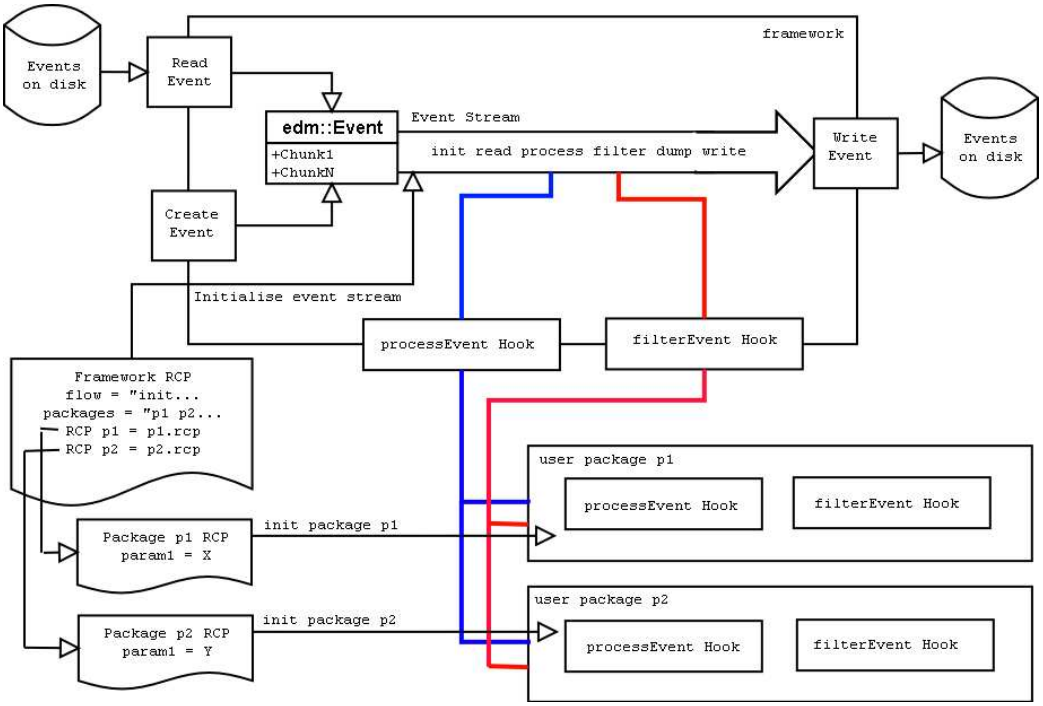


Figure 5.1: Schematic of Event stream in the framework. The framework RCP defines which hooks will be called and which packages will run and which RCP files the packages will use. Events are read from file or created and the EDM Event objects are passed through the event stream, where they are passed to each package for processing of some Event Chunk, which may add/modify/remove chunks before the output is written to disk.

simulate physics interactions as expected in the detector. These MC events are initially produced using MC generator software programs that generate a particular type of physics event based on theoretical predictions and a set of input parameters. The physics details of these generators are not presented here, for this chapter it is sufficient to note simply that they produce a set of four vectors corresponding to particles arising from a simulated $p\overline{p}$ interaction in empty space. The steps involved in producing MC Events are shown in Fig. 5.3, each step is described in detail in the following sections.

### 5.3.1 Event Generators

At DØ the event generators used are extremely complex independently produced programs, usually written in the FORTRAN programming language. They have a C++ interface to the framework which produces a DØ chunk in an Event via the `mcpp` interface which take the FORTRAN generator output and convert it into a set of linked objects which are stored in a chunk. Each generator, therefore, looks like a standard DØ framework executable that produces `DØom` format events. MC event information is stored in a chunk called the `MCKineChunk` which is primarily composed of `MCparticle` objects describing each particle and `MCvtx` objects describing the interaction vertices between the particles.

`MCparticle` objects are a persistent `DØom` class containing all the information for a single MC particle, including its Particle Data Group identification code (PDGID) as defined in Ref. [60], all the kinematic information about

the particle, pointers to it's production and decay vertices and a unique identifier within the `MCKineChunk` for the event.

`MCvtx` objects are a class containing descriptions of the vertices formed by interactions between the particles in the event. They contain a unique identifier within the `MCKineChunk` , the position of the vertex and lists of particles coming into and going out of the vertex.

By examining the `MCparticle` and `MCvtx` objects in an event, one can investigate the particle level nature of the event and compare it directly to the simulated effects of these particles in the detector.

Each MC generator executable is configured via the RCP system and a driver file for the FORTRAN process containing information governing the actual physics details to be simulated. The main generators used at DØ are PYTHIA [62] and HERWIG [63] with various customisations and plug-in tools for accurately simulating specific physics processes. These extra tools are specialised for certain tasks, for example Onetop [64] is used for top quark processes, and decays of hadrons containing bottom quarks are handled by the CLEO QQ or EvtGen decay tools [65][66]. All of these utilities are built into libraries and linked into the generator executables so that they can be switched on or off in the RCP or driver files at the user's convenience.

The events produced by the generators for a given process are produced across a broad kinematic range over which there is some control, but ultimately requires some post-generation selection of events to find specific signal events. To enable combined generation/selection for specific event channels within a strict kinematic range, a generic selection tool, controlled by RCP,

was written and is included in the generator executable.

### 5.3.2 The d0_mess Package

The d0_mess (standing for DØ MC Event Selection System) package [61] is a framework package which uses the filterEvent framework hook to examine the MCparticles in the `MCKineChunk` for each event and require that the event contain MCparticle objects which satisfy the user defined cuts. A schematic showing the main features of the d0_mess package can be seen in Fig. 5.2.

The Cuts are implemented as templated objects which act on some trait of the `MCparticle`, such as $p_T$ ,$|\eta|$ or the particles parents properties. These cuts can also be combined logically to produce complex cuts involving multiple traits, for example a combined $p_T$ and $|\eta|$ cut for all particles of a certain type. These cuts are exposed to the user via a string representation in the d0_mess RCP file. Thus the user can express cuts as strings in the RCP for example:

$$\text{string Cut1} = \text{“PdgId} == 13 \;\&\&\; \text{Pt} >= 1.5 \;\&\&\; \text{AbsEta} < 1.0\text{”}$$

Which produces a cut requiring a muon (PDGID 13) with $p_T$ of 1.5 GeV or greater and in the central $\eta$ region. Multiple cuts can also be specified, as well as flags which invert the result of a cut, thus allowing events to be selected that do not contain objects matching the cut. Initially, multiple cuts are all logically “anded” together, with a system to allow logical expressions to be grouped with AND, OR and parentheses to create complex cut arrays to filter events. Additional trait cuts are also in development including cuts that

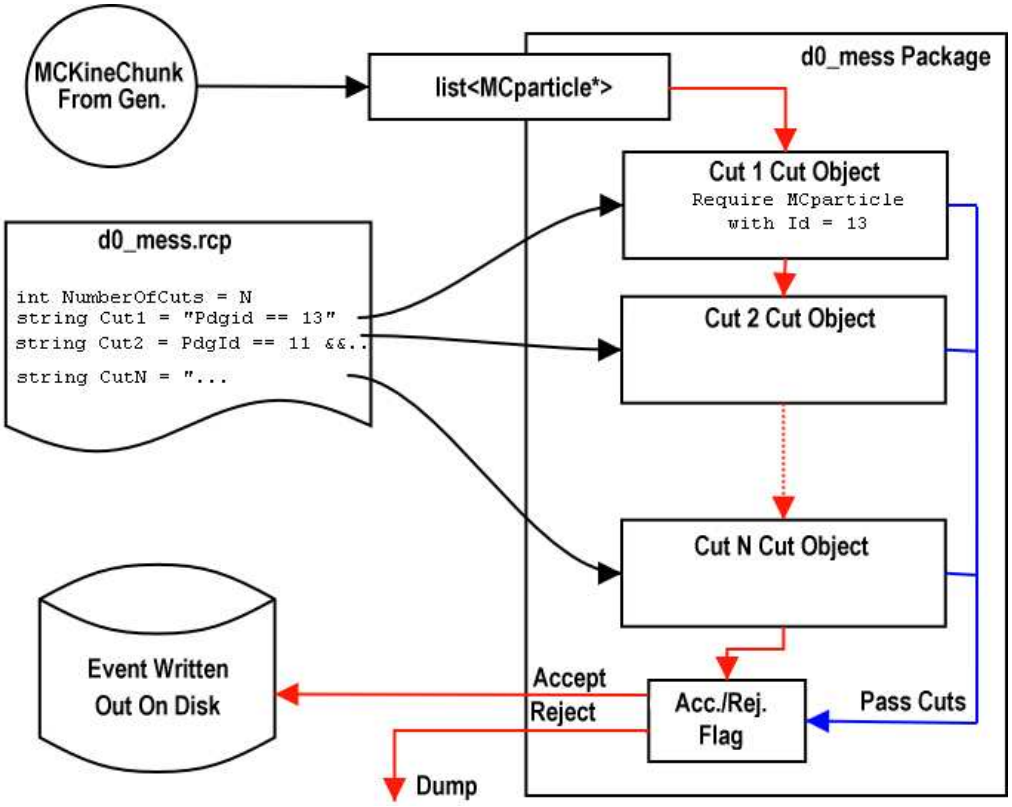


Figure 5.2: d0_mess Schematic. The d0_mess package is initialised from the RCP file which instantiates an array of cut objects based on the contents of the RCP. The list of `MCparticle` objects is extracted from the `MCKineChunk` created by the generator, and each cut is applied to each `MCparticle`. If all the cuts are passed then the Accept/Reject flag is set and the event is written to disk. If the cuts are not passed the event is discarded.

model the turn-on curves of the DØ trigger system to accurately generate MC events that mimic the data as closely as possible.

## 5.4 MC Event Processing

The MC Generator produced events are generated as if the event occurred in a vacuum, not within the confines of a detector with all the material present, thus to make the MC Events useful, they must be taken and manipulated to look as if they had occurred within the volume of the DØ detector. This includes scattering effects as they pass through material, charged particles must be modified to act as if they are in the detector's magnetic field and so forth. In addition, the response of the detector to these events must be simulated, for example, charged tracks passing through the Inner Tracker would create a certain hit pattern in the Silicon Microstrip Tracker and cause photoelectric showers in the Central Fibre Tracker, while muons would produce hits in the muon system etc. Consequently there has been considerable work to make MC events look like "real" events in the detector. This is done in two main steps, event simulation and digitization, each of which is performed by a framework executable. In addition a further processing step may be added, which simulates the response of the DØ trigger to the MC Event, and adds the trigger information to the event.

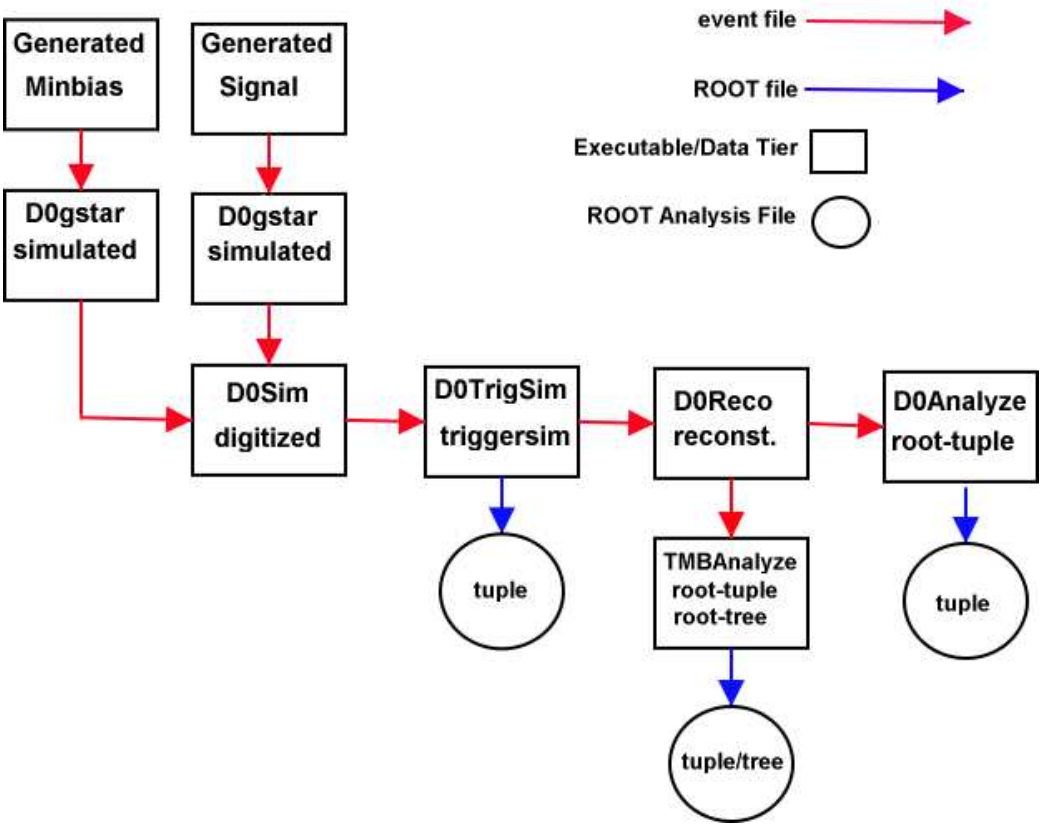


Figure 5.3: Flow Diagram showing the processing steps involved in producing Monte Carlo events for use in DØ analysis projects. Each step of the diagram is described in detail in the text.

### 5.4.1 Event Simulation

The Event Simulation stage involves the migration of the MC Event into the detector volume, and propagating its effects throughout the detector using the GEANT [67] software within a C++ wrapper interface to the DØ Framework. The executable responsible for this is called *d0gstar*[68], which stands for *DØ GEANT Simulation of the Total Apparatus Response*. The d0gstar executable takes the `MCKineChunk`input and transforms its components as if it had occurred within the detector, by propagating each particle through the various materials it would traverse according to the detector geometry and magnetic field. Calculation of hit positions within the various subdetector systems is performed and stored as simulator chunks in the event. In flight decays of particles such as pions and kaons are calculated taking into account the material they interact with, with the decay products being generated by calling Pythia subroutines. The Simulation process is extremely complex and can be very time consuming, with processing times per event of the order of 2-10 minutes on a 1GHz Intel Pentium III Processor.

### 5.4.2 Event Digitization

Event Digitization takes the simulator output and converts the hit patterns into digital and analogue signals that would indicate such hits within the detector. The end result is an event with the same structure as a real data event recorded by the online system stored in a `RawDataChunk`. The digitization process also adds in noise and inefficiencies such as resolution smearing and

dead readout channels for each subdetector system. Calorimeter pileup from previous events is added by overlaying additional inelastic $p\overline{p}$ collisions over the signal event. Minbias events are digitized soft QCD MC events tuned to Run I Tevatron data [69], which are merged with the signal event according to a randomly generated distribution to simulate multiple $p\overline{p}$ interactions per bunch crossings. The executable responsible for the digitization is called *d0sim*[70] and on average takes between 30-60 seconds to process an event on a 1 GHz Intel Pentium III Processor.

### 5.4.3 Trigger Simulation

Since the MC Events are not produced by the detector, there is no information in the event on which triggers were used to capture the event, as there would be stored in a real data event, thus the *d0trigsim* [71] executable provides a full simulation of the combined Level 1,2 and 3 trigger information and inserts it into the event. The trigger simulator provides a software simulation of the hardware Level 1 trigger system, and wraps the software algorithms used in the Level 2 and 3 trigger systems so that they can be run offline.

## 5.5 Event Reconstruction

After digitization and trigger simulation, the MC event is essentially the same as a data event, and as such can then be run through the offline detector reconstruction program *d0reco* [72]. The job of d0reco is to reinterpret

the various bit patterns and detector signals and convert them into physics objects for analysis processing, this is performed using a tiered set of algorithms which unpack the `RawDataChunk` and decode it. After this decoding, the electronics channels associated with each subdetector system are augmented with calibration and alignment information and used to build hit patterns and clusters through the subdetectors geometry. These basic hit and cluster objects are then used to build tracking information by combining output from the detector subsystems. Using the tracking information, primary and secondary interaction vertices are found and used to calculate basic kinematical quantities for the event. The final step in the reconstruction is to combined the basic objects found into physics-like objects and perform particle identification via many sophisticated algorithms. The first objects found are electrons, photons, muons, missing transverse energy and jet candidates, which are then use to build compound objects such as heavy quarks and tau decays. All the information found is added to the event in the form of chunks, to be written out and saved. The information created by the reconstruction process is stored in two formats:

- **Data Summary Tape (DST)**

  The DST format is designed to provide enough information to perform all required analysis tasks by storing the basic physics objects, from which the more complex, compound objects can easily be rebuilt during the analysis, and is stored on large scale magnetic tape storage, along with the original raw data if required. The DST files can be reprocessed

with a newer version of d0reco at a later date as the algorithms evolve.

- **Thumbnail (TMB)**

  The TMB format is a snapshot of the DST, almost a tenth of its size, which provides a summary of the event suitable for many useful standard analyses and sample selection which can be performed rapidly. The TMB files are intended to be stored on disk for data to promote easy access for analyses.

### 5.5.1 Event Analysis

The chosen analysis tool for DØ Run II analyses is the ROOT [73] analysis framework, which is a C++ based program with an interpreter language to allow interactive analysis tasks, and can also be extended with other C++ analysis objects. DØ uses a set of ROOT based analysis objects that can be stored as ROOT format files for easy analysis. ROOT Files are formatted as either an ntuple column style format or an object based tree format, and are produced by analysis packages for specific `DØom` Chunk formats. The most common analysis package is the *TMBAnalyze* which converts the TMB files into a ROOT Tree file for analysis tasks, other ROOT files that contain information about MC generator events, trigger simulator outputs and chunk contents.

## 5.6 Need for Large Scale Processing at DØ

Due to the large amount of raw data that will be collected by DØ during Run II, there are requirements on the computing ability of the experiment to handle the processing and analysis of this data. In addition large amounts of MC are required for analysis, at least twice the amount of MC as data. This means that large data samples that are multi-gigabytes to terabytes in size need to be moved, manipulated and examined for physics studies. This requires vast computer resources, and has resulted in DØ adopting a model of distributed computing farms [74] on which processing tasks such as MC production, reconstruction and re-reconstruction of data and existing MC and analysis processing of large data samples are performed at computing facilities around the world. Thus DØ has great need to efficiently handle and automate these processing tasks in a way that is as transparent as possible to the end physicist user.

## 5.7 Data Storage and Access

Data Storage and Access for DØ is handled by the SAM system [75][76]. SAM (standing for "Sequential Access to data via Metadata"), where sequential refers to the sequential nature of events stored in files, which are in turn stored on tapes. Metadata is a set of parameters that describe the data stored within a file in the SAM system, including how the file was produced.

SAM is used as a data catalogue, file delivery and storage manager for

all the data, both real and simulated. It also provides methods to search the metadata describing the files.

SAM is a Data Grid which implements distributed data storage and access to that data at processing facilities around the world. All DØ processing installations run a SAM Station controller which manages disk cache, file delivery to processing jobs and access to Mass Storage Systems (MSS) for file storage and retrieval. The delivery of a set of files is performed by intelligently gathering files first from local cache disk areas of the processing farm, then from other station's caches, and finally from MSS. The files are delivered to the processing nodes and then cached after they have been processed, and are eventually overwritten if not required by other jobs. SAM Stations also provide the ability to declare and store new files into the SAM catalogue and storage system, so that the output of processing tasks can also be accessed through SAM.

SAM Users access files using sets of metadata to define a group of files called a *dataset* that is built by creating a list of all files within the SAM catalogue that have the same set of metadata characteristics as defined by the user. Datasets are processed using *projects* that deliver the files in a dataset to executables running on processing nodes. Projects provide support for parallel processing, allowing multiple executable tasks to process pieces of a dataset, ensuring that all files are processed only once.

The DØ software executables have a SAM interface called *SAMManager* linked into them, which uses a C++ API to SAM to allow executables to have input files delivered to them from a SAM project, enabling execution

to pause for file delivery if necessary. This interface is controlled via an RCP file which contains the details of the SAM project that will be delivering files.

### 5.7.1 Metadata

Metadata is "Data describing the Data", a set of parameters that describe the contents of a data file in a concise way, and its origin and processing history. At DØ , SAM is used to catalogue data files using metadata to provide users with information on the type of data in the files, such as wether it is MC or raw detector data, details of the software used to process it and versioning information. In addition, for MC, details of the simulated process are stored as part of the metadata so that physicists can select datasets using physics characteristics, such as the event type, some kinematic information and simulation details. The metadata also contains enough information that a processed file can be reproduced at a later date if necessary, allowing interim processing steps to be recorded without the need to store the intermediate processing files. The use of metadata for MC production with SAM is shown in Fig. 5.4. In addition to information on content and processing history, the metadata can also provide information on the actual job details used to produce the files, allowing details of the workflow to be included in the metadata, this allows a complete job from beginning to end to be described in terms of metadata.

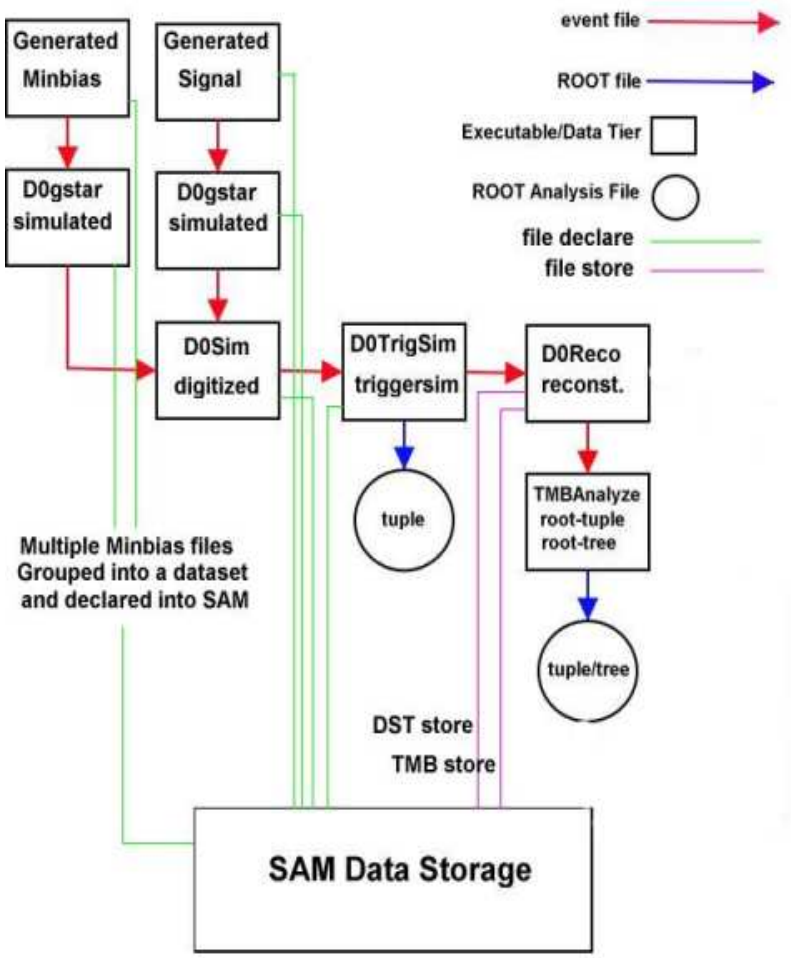


Figure 5.4: The MC processing chain showing how metadata is used to catalogue files in SAM. Declared Files (green lines) are not stored to the mass storage system, but their metadata description is, allowing the files that are stored to have a fully reproducible parentage. So for MC generation, only the d0reco output is stored for physics analysis, but the complete details of the simulated process is stored in the metadata so that the d0reco files are fully described in terms of generated process, kinematics and event simulation steps.

## 5.8 Workflow Planning

### 5.8.1 What is a workflow?

A workflow consists of a set of interdependent tasks that must be performed in a certain order. There can be many different, convoluted connections between steps in a workflow, including parallel processing steps, which need to be split or merged into processing streams. Workflows are described by metadata, which defines the relationships between the various processing steps. Many things can be used to add complexities to a workflow, such as the resources used by each step, where input data is situated, where the output is to be sent, management of intermediate files and so forth. For the DØ processing tasks, each executable is a different step in the workflow, and these tasks are very intensive in terms of the computer power required, and so requires parallelisation and optimisation of the processing to make sure that peak performance is attained from the resources. As an example, a typical MC processing task would contain around 50,000 events, whereas an optimal number for the typical execution node is around 500 to 1000 events, thus the MC task is split into smaller jobs that are run in parallel. Further complications to this process, could include using an existing input file containing all the input events, so that the processing has to be paralellised during the execution, or the output of the parallel streams may need to be merged together to allow an analysis step that examines all the output in a single place.

## 5.9 The mc_runjob Package

mc_runjob is a Python [77] based software package that acts as a workflow planner for the DØ software. It is used to manage large scale MC production for DØ and is also used by the CMS Experiment. It takes a metadata description of the workflow and converts it into a series of executable tasks, job descriptions and multi-path parallel workflows. mc_runjob is implemented via a very modular Object Oriented design making it easy to expand, and thus highly flexible. It defines a metadata macro language allowing non-expert users to easily express their requirements in a simple manner, and protects the user from the complicated mechanics involved in the execution of the workflow. At DØ mc_runjob is used for both large scale automated production and for individual user scale processing tasks. The main tasks of mc_runjob can be summarised as follows:

- Provides a uniform interface for varying software.
- Provides a simple interface for all common processing tasks.
- Simplifies complex production processing environments with hundreds of input parameters.
- Handles complex multi-application workflow topologies.
- Resolves Dependencies and relationships among the metadata easily.
- Produces a metadata description of workflow.
- Removes the users need to understand DØ software details.

### 5.9.1 Architecture

mc_runjob consists of three major parts:

- **Preprocessor** The mc_runjob preprocessor is used to adapt the metadata to the job execution environment, tailoring the job to run optimally on the available resources. It also starts any services that will be required by the jobs, such as SAM projects and splits large jobs into sets of parallel jobs if needed. The preprocessor model is shown in Fig. 5.5.

- **Configurator** The Configurators are objects which know how to perform a single atomic task, such as run an executable, interface to a batch system or move a file and other such tasks. The Configurator objects are configured via metadata keyword-value pairs which are handled according to the tasks the configurator is designed to perform. Configurators make heavy use of inheritance, allowing easy integration for common tasks such as file I/O for all DØ framework executables.

- **Linker** The Linker is a container for the configurator objects, chaining them together into a workflow. It acts as a communication bus, allowing the configurators to communicate between each other, while resolving dependencies between them and providing a user interface to the configurators in terms of the macro language. The Linker is also responsible for running the configurators to build and submit executable jobs. The Linker is illustrated in Fig. 5.6.

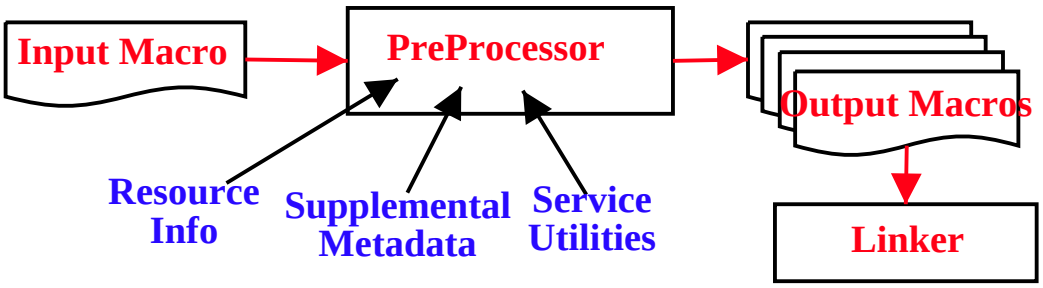


Figure 5.5: Diagram showing the mc_runjob Preprocessor. The initial set of metadata is expressed in the mc_runjob macro language and passed to the preprocessor which adapts the metadata to the local execution environment, tailoring the job to the local resources. Parallelisation of the job is performed by cloning the tailored macro and sending the tailored clones to the mc_runjob Linker.

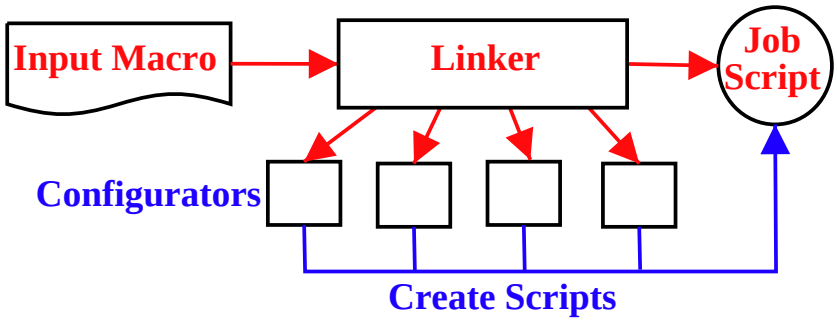


Figure 5.6: Diagram showing the mc_runjob Linker, which takes a metadata description of the workflow and interprets it in terms of configurators, which are then set up according to the metadata, to produce a set of executable tasks, which the Linker submits to an execution resource.

At DØ , mc_runjob is implemented so that each of the major executables is represented as a configurator, allowing users to configure the most often changed aspects for these executables without having to find the appropriate RCP files and editing them by hand. SAM integration is also provided by configurators to start projects and feed input into executables. Runtime input configuration files are also handled by mc_runjob , along with manipulation of all file IO tasks between executables, allowing the average user to express a complicated processing task as a single, simple text macro, that deals with physics properties and hides the user from the need to understand the complexities of the DØ Framework, mc_runjob operates as a "black box" system which the user feeds in a simple set of commands and (a few hours later) gets back useful output files. Very generic analysis configurators are also available, allowing the user to specify their own analysis executables using only a metadata description and transport the necessary components to the execution site. A system of runtime monitoring utilities are also included in mc_runjob , so that jobs are executed and monitored to make sure they are behaving as required, and feeding status updates to a variety of monitoring systems.

## 5.10 Remote Computing

As mc_runjob allows complex jobs to be expressed as metadata, it becomes straightforward to send processing tasks to various distributed locations to be converted into local executable jobs. mc_runjob is fully interfaced into

SAM via the SAM User API Python interface which allows mc_runjob to integrate SAM tasks into the various configurator objects as required, for example, starting projects from the preprocessor, or setting values for SAM-Manager to provide input to the executables. With SAM being a distributed resource, this allows MC processing to be handled within SAM, by storing requested MC details in the SAM catalogue as "potential" MC data, which is tagged with a unique request identifier. Then when a remote site needs to run a certain MC request, it is sent that request's identifier, via some queuing system, and it is passed to mc_runjob which extracts the request metadata from SAM to create and submit executable tasks suitable for the local resource as shown in Fig. 5.7.

Development of the request submission procedure is being incorporated into various remote computing grid projects such as the JIM SAM Grid project [78], to enable matching of execution resources based on having the software available to do the processing, the local availability of any input files needed and the free resources that can be devoted to the job. mc_runjob makes this task much easier as the entire job can be expressed as metadata, and the execution sites can advertise their capabilities in terms of this metadata, allowing the grid schedulers to match jobs to resources. In addition, since mc_runjob knows the details of the executables, it is envisioned that it will be used to distribute software as part of its execution and job building process, to enable the DØ software to be run on shared resources where it is not possible to install the DØ software.

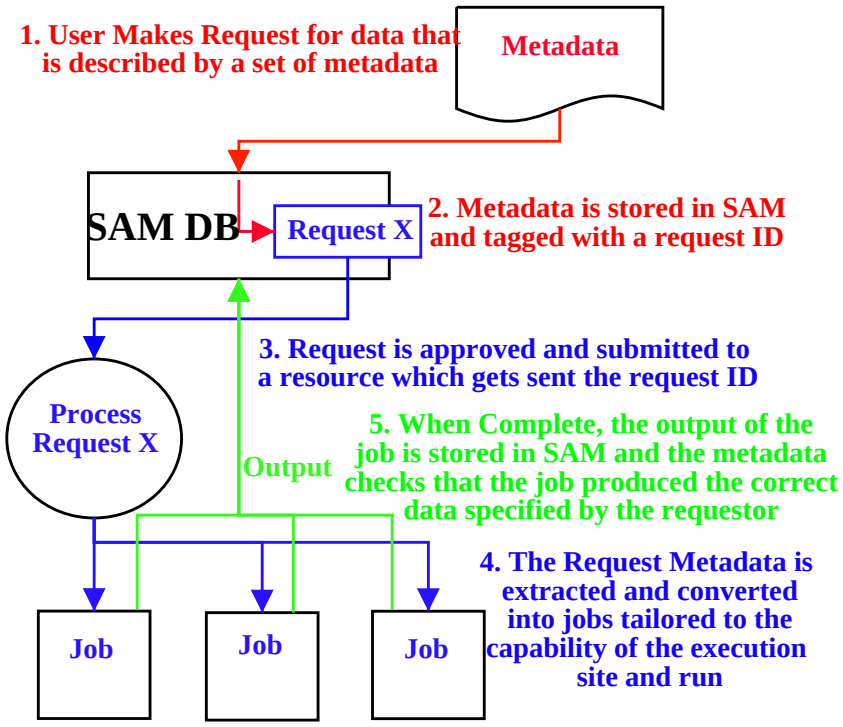


Figure 5.7: Diagram showing the request system as it uses SAM to distribute metadata to a remote resource, the preprocessor extracts this metadata and compiles it into a set of macros which are executed. The output is stored back into SAM and checked against the request metadata for consistency.

## 5.11 The Shahkar Project

mc_runjob is in extensive use as a large scale production tool at both DØ and the Large Hadron Collider's upcoming Compact Muon Solenoid (CMS) experiment [79]. However, the developments at CMS and DØ have not been synchronised, so there are two different incarnations of mc_runjob , one at each experiment. Since the basic architecture of the Linker and Configurator are extremely flexible, it makes sense to attempt to keep these classes in common between the two experiments as well as a common metadata language, so that jobs from different experiments can be expressed in the same semantics. This allows the experiments to potentially use each others resources with very little effort. In addition to the basic classes, many other features are common to both experiments, with interfaces to monitors, file transfer mechanism and security handling features being used by both experiments. The combined project is called *Shahkar* [80], being derived from the Urdu language for good worker, or great job, and seeks to combine the best of the developments from DØ and CMS into a centrally maintained project, so that both experiments can use it and benefit from an extended pool of development experience. Future tools include expansion of Shahkar to a web service based system that serves jobs to various compute grid projects such as SAMGrid and the LHC Computing Grid system.

## 5.12 Summary

mc_runjob has been in use at DØ for many years now and is maturing into a very powerful workflow tool, that will extend beyond the experiment it was originally developed for. It has been described in the CHEP conferences and proceedings [81][82] and is attracting interest from other experiments. At DØ mc_runjob is becoming the main tool for offsite processing, due to its flexible nature and highly configurable setup, it will be used for reprocessing detector data samples using a system similar to the request system to coordinate massive parallel distributed processing of detector runs, and eventually to run user analysis programs in an execution grid environment, that will run analysis jobs around the world via the various grid systems. The Shahkar project will build on the success of mc_runjob to evolve it into a more widely useful workflow planning tool, that will enable physicists to run their analyses in a distributed grid structure in a straightforward way which emphasises the physics aspects of the task while suppressing the need for detailed knowledge of the underlying software and computing systems.

# Chapter 6

# Finding Charged B mesons using the AA Tracking Algorithm

## 6.1 Introduction

With the introduction of the new Inner Tracker and Solenoid magnet for Run II at DØ , finding tracks became an important part of the event readout and reconstruction process. Various track finding algorithms were investigated to find the optimum track reconstruction algorithm for DØ , one of which was the `AATrack` (Alternative Algorithm Track) tracking algorithm, written by Lancaster Researcher Dr. Guennadi Borissov. This chapter involves an overview of the `AATrack` algorithm, and a review of a tracking study used to find charged B mesons with the tracks produced.

## 6.2 The AATrack Algorithm

The AATrack Algorithm [83] follows a hypothesis building and filtering structure which uses hit patterns to produce a large set of possible tracks which are then filtered to find the best fitting tracks in any given event. Initial track hypotheses are constructed from Silicon Microstrip Tracker (SMT) hits in two dimensions ($r - \phi$) from three hit patterns (triplets), with further hypotheses constructed using hits in the Central Fibre Tracker (CFT) to find tracks which do not form hits in the SMT. All track hypotheses are added to a central pool, and then ordered according to the quality of the hypothesis and then filtered to combine/reject/accept the hypotheses until the final best track candidates remain.

### 6.2.1 Finding SMT Track Hypotheses

The first step for finding tracks is to build tracks using three hits in $r - \phi$ starting within the SMT barrels or disks, commencing with hits in the innermost layer and working outwards as illustrated in Fig. 6.1. Starting the tracking search within the SMT has the benefits of being less complex due to fewer combinatorics from associating axial and stereo hits, and since particles interact within the detector, it is more likely that the original particles properties will be inspected, the closer to the interaction point one looks.

Hypotheses are constructed subject to a set of constraints on the kinematics of the track hypothesis. The first requirement is that the first hit occurs within the first 6 barrel layers or an F-Disk of the SMT detector.

This first hit is used to determine a window in $\phi$ about the beam spot and first hit, in which to search for the second hit as shown by the *Axial Angle Window* in Fig. 6.2. The third hit is found by matching a hit in any following layer, providing that the hypothesis has a track radius of less than 30 cm, corresponding to a minimum $p_T$ of 180 MeV and the axial impact parameter with respect to the beam spot is at most 2.5 cm, these parameters are shown in Fig. 6.2. Additionally the track hypothesis is also required to have a goodness-of-fit Chi-Squared test value of less than a cut-off maximum value.

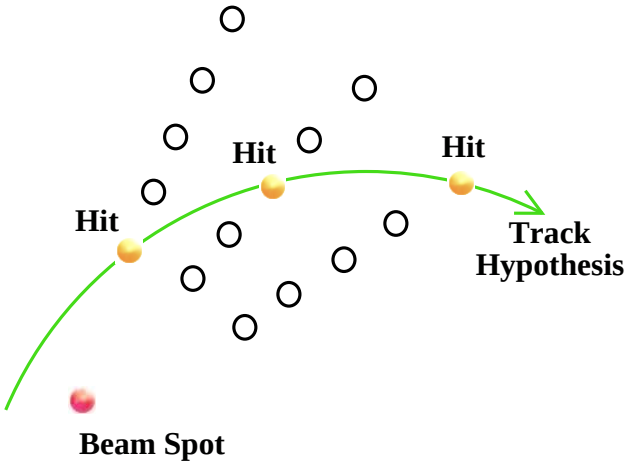


Figure 6.1: Diagram showing the construction of track hypotheses using three hits within the SMT detector.

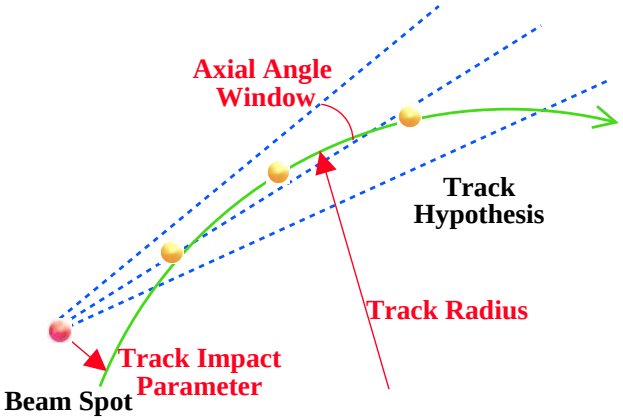


Figure 6.2: Diagram showing the variables used to constrain track hypotheses as they are constructed from an SMT hit triplet. The blue lines show the window in $\phi$ formed by the first hit and beam spot that restricts the position of the second hit. The radius of the track and impact parameter of the track with respect to the beam spot are used to restrict the track hypothesis for the third hit.

### 6.2.2 Finding CFT Track Hypotheses

Tracks with fewer than three SMT hits form a large part of the overall tracking signals, so building tracks in the SMT only would not include these, for this reason track hypotheses are constructed using hits in the CFT and worked back into the SMT. The main challenge to this is the large number of combinations rising from the axial and stereo hit association, which provides many possible track hypotheses, requiring large amounts of processing time to examine all the possibilities. By using the primary vertex constructed using the SMT track hypotheses, it is possible to remove many of the fake tracks by requiring that the tracks pass close to the primary vertex. CFT tracks are found using a two-dimensional hit triplet in the CFT in the same manner as for SMT tracks, but also requires that the track hypotheses are within 1.5 cm of a primary vertex candidate built from the SMT tracks in both axial and stereo dimensions. These tracks are then extrapolated back into the SMT to collect all possible SMT hits.

### 6.2.3 Expanding Track Hypotheses

Once the three-hit track hypotheses have been constructed, they are extrapolated to other layers of the tracking detectors by calculating a window where the hit should land to be associated with that hypothesis as illustrated in Fig. 6.3. The window is calculated by finding the local coordinates of the track at the new detector layer and restricting the window size to be three times the margin of error on the track extrapolation. Hits occurring within the win-

dow are associated with the track hypothesis, subject to a chi-squared test, with multiple associated hits in the same layer being split off into a separate hypothesis for each hit. For stereo hits, each hypothesis may be repeated many times, since there is no separation in the stereo (z-direction parallel to the beam line) due to the two dimensional nature of the track search. As more detector stereo layers are included, missing stereo hits can be used to exclude many of the hypotheses, although some will remain. Keeping extra stereo projections and using them to select the best projection in association with a vertex can be used to improve the quality of complex track objects such as vertices associated with objects like B hadrons.

During the search for tracks, misses are also found and recorded to measure track quality. Misses are layers of the detector where a hit is not found within the expected window, as shown in Fig. 6.4a. Misses are classified into three types in the `AATrack` Algorithm: inside misses, which occur between two hits along the track hypothesis, forward misses, which occur before all the tracks hits, and backwards misses which occur after all of the tracks hits. These types of misses are illustrated in Fig. 6.4b. Tracks are extrapolated until the end of the tracking detector is reached or there are three consecutive misses found at the end of the track.

### 6.2.4 Selecting and Filtering Track Candidates

Track hypotheses are saved for filtering if they match the following criteria:

- At least four layers in the SMT or CFT with both stereo and axial hits.

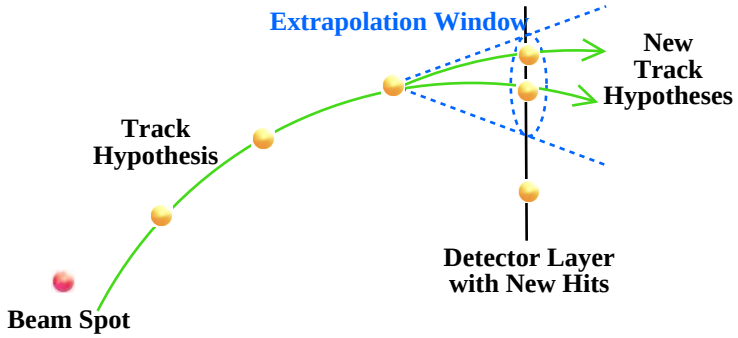


Figure 6.3: Diagram showing how the expectation window is used to search for hits in further layers of the detector, hits within the extrapolated window are added to the hypothesis, with multiple hits causing the creation of new hypotheses for each extra hit.

- Maximum of 3 inside misses.
- Maximum of 6 forward or backwards misses.
- Maximum of 2 inside misses within the SMT.
- The total number of hits divided by 5, must be greater than the total number of misses.
- If a hypothesis has an inside miss, then it must have less than four total inside and forward misses and no more than three total inside and backwards misses.

These criteria are a set of optimsed cuts that have been developed to improve efficiency and reduce fakes, by requirng that the overall track hit pattern is

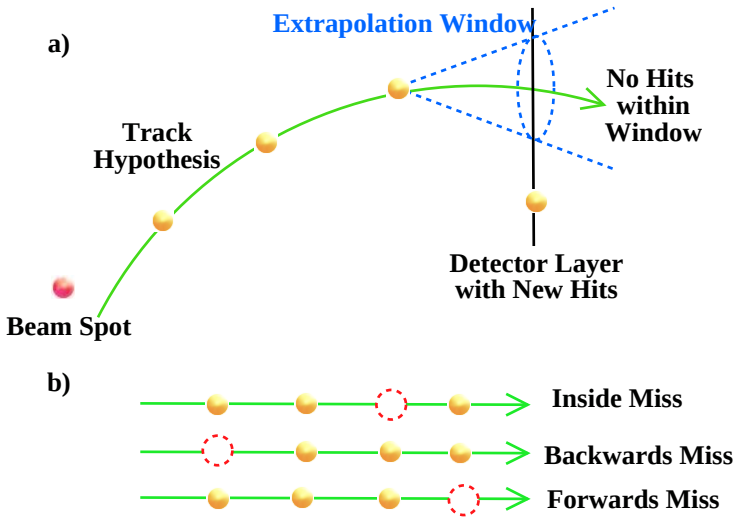


Figure 6.4: Diagram illustrating misses as defined by the `AATrack` Algorithm. a) Shows how misses occur in the extrapolation of the track, by finding no hits within the extrapolation window. b) shows the three types of misses, inside, forward and backwards hit misses along the extrapolated track.

well populated by hits. Tracks which pass these criteria are then sorted by number of hits, number of total misses and chi-squared value in that order. Different track hypotheses can share the same hits, so further criteria are applied to the number of shared axial hits, requiring that not more than two thirds of the total axial hits are shared hits, with more stringent restrictions applied to track hypotheses which contain misses and do not span the entire tracking detector.

### 6.2.5 Primary Vertex Finding

As mentioned earlier, the knowledge of the primary vertex position can be used to reduce fake track hypotheses due to stereo combinatorics, and can also be used to refine the filtering of hypotheses by comparing them to the primary vertex position as shown in Fig. 6.5. The position of the primary vertex is determined using the track hypotheses found from the SMT, and any track that is close to the primary vertex, is weighted with the addition of two extra hits to the hit total for that hypothesis. The track hypotheses are then re-ordered using the same criteria described in the previous section, which means that those tracks closest to the primary vertex occur higher in the order, but does not exclude tracks offset from the primary vertex if they have lots of hits unique to that hypothesis. The filtering process is again repeated to remove poor hypotheses and this results in the final set of `AATrack` Objects for the given event.

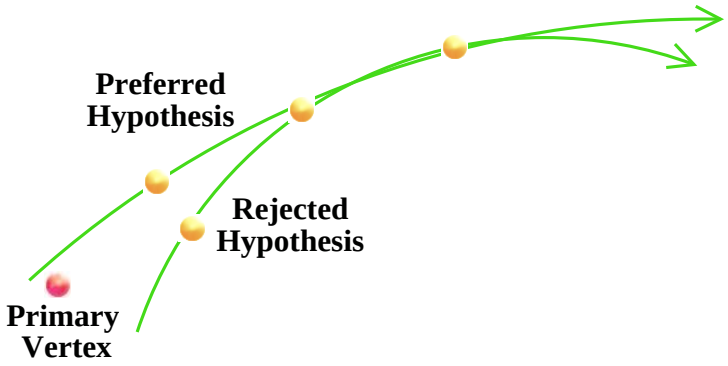


Figure 6.5: Diagram showing how the primary vertex position can be used to select track hypothesis by preferring tracks which approach the primary interaction vertex to those that don't.

## 6.3 Finding Charged B Mesons with tracking

To find the $B^+$ meson using tracks, the easiest way is to search for it via the decay $B^+ \rightarrow K^+ J/\psi \rightarrow \mu^+\mu^-$ as it will occur as a three track vertex formed by the two muons and $K^+$ charged tracks, as illustrated in Fig. 6.6. This occurs because the $J/\psi$ lifetime is very short and it barely moves away from the point of its production before decaying, hence the three track vertex. The $B^+$ is initially produced in the primary interaction with enough energy to displace it from the primary vertex before it decays, so the three track vertex should be a short distance away from the primary vertex.

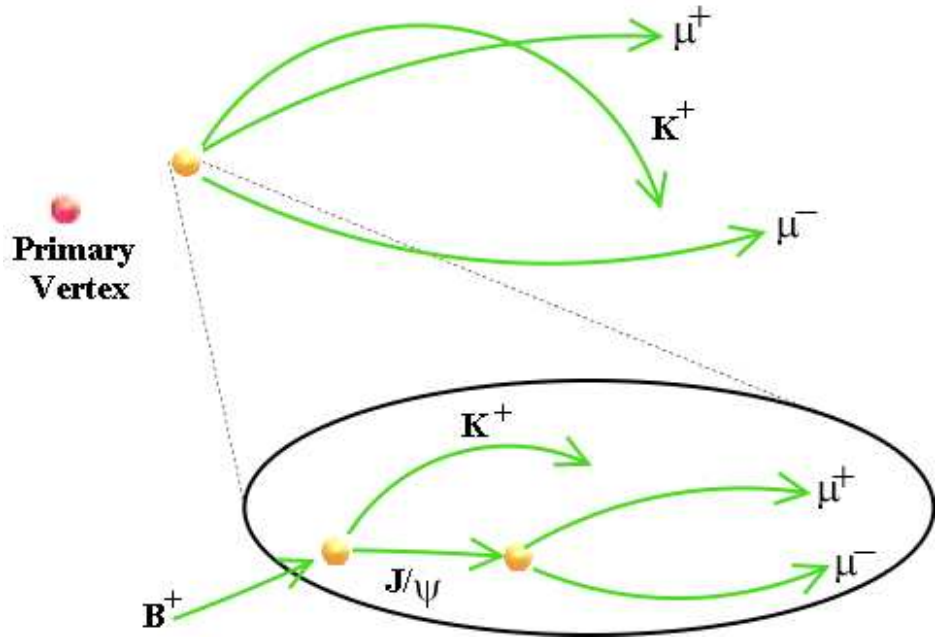


Figure 6.6: Illustration showing the signature of a $B^+$ via the $K^+$ $J/\psi$ $\mu\mu$ channel. The search is performed by searching for a three track vertex containing a dimuon pair from a $J/\psi$ and another charged track associated with the $K^+$. The inset shows the actual process "inside" the vertex as the $J/\psi$ decays almost instantaneously resulting in the three track signature.

### 6.3.1 Data Sample Selection

Using this mode allows easy selection of events in data, since the muon system is complete and fully working from a triggering point of view, by allowing examination of events which fire a simple dimuon trigger requiring two muons reconstructed in the trigger system with $|\eta|$ less than 2.4. Further event selection is done by examining the dimuons and selecting those that occur within the $J/\psi$ mass window. The data sample used is the official DØ dimuon event skim which corresponds to an integrated luminosity of $114pb^{-1}$. This data sample contains 290296 $J/\psi$ signal candidates, yielding a measurement of the $J/\psi$ mass to be $3072 \pm 2$ MeV, with the mass plot shown in Fig. 6.7.

### 6.3.2 Uncertainties on the $J/\psi$ measurement

The $J/\psi$ mass value of $3072 \pm 2$ MeV, differs from the Particle Data Group (PDG) value of 3096.8 MeV by approximately 25 MeV. The majority of this difference is due to the detector tracking momentum response, which is due to unknowns about the precise path tracks follow through the detector, including variations in the Magnetic Field, and incorrect description of the physical material through which the tracks pass. This leads to a momentum dependent deviation in the mass measurement as illustrated in Fig. 6.8.

### 6.3.3 Selection Of Signal Events

The search for signal events is performed in three main steps:

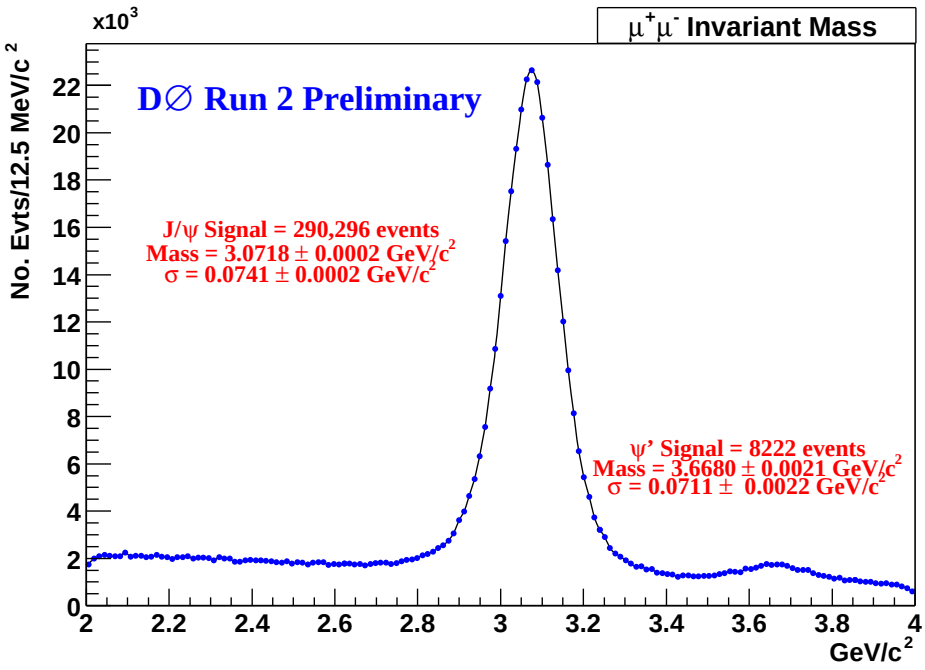


Figure 6.7: Official DØ Mass plot for the $J/\psi$ . This plot is made using $114pb^{-1}$ of data. The smaller peak on the right corresponds the $\Psi'$ particle, and excited resonance of the $c\overline{c}$ meson. [85]

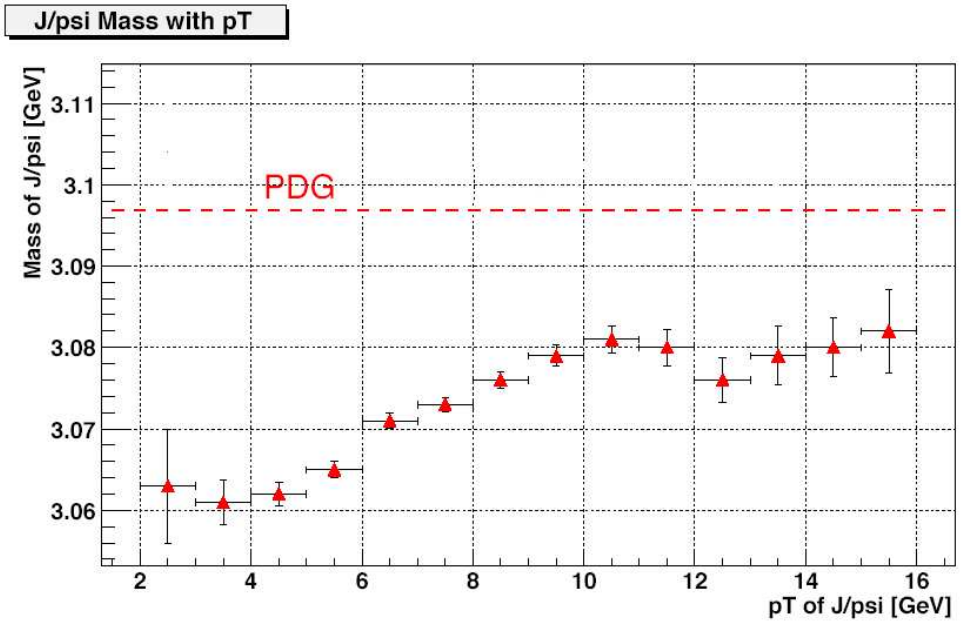


Figure 6.8: Plot of $J/\psi$ mass versus $p_T$ , showing the momentum dependence of the $J/\psi$ mass measurement due to momentum scale effects in the DØ tracking detectors.[86]

- Combine Track and Muon detector information to find pairs of opposite charged muons that combine to form a vertex.
- Examine the mass of the combined muon track pair to see if it falls within the $J/\psi$ mass window. To account for the distortion of the $J/\psi$ mass as discussed in §6.3.2, a mass constrained fitting technique is used to build the $J/\psi$ .
- Look for a third charged track and test its association the dimuon vertex.

The `AATrack` Ana package takes the raw output of the `AATrack` Algorithm and combines the track hypotheses into charged particle candidates and performs association with the raw muon information for any tracks that match muon hits. $J/\psi$ Candidates are selected by combining muon pairs and requiring that they come from a displaced vertex. The $J/\psi$ selection process is as follows:

- All Tracks associated with the primary interaction vertex are discarded.
- Selection criteria are applied to the two muon tracks, they must both be of opposite charge, have a $p_T$ greater than 1.5 GeV, have at least one hit in the CFT and be associated with a muon hit in the muon detector.
- The total momentum of the two muons must be greater than 5.0 GeV.
- The combined mass of the two muons is required to be within the range 2.80 to 3.35 GeV.

The $J/\psi$ is produced by combining the two muons with a *mass constrained fit* technique, which assumes that the combined mass of the two muons is the actual measured mass of the $J/\psi$ particle. This means that if the invariant mass of the two muons is within the mass range 2.8 to 3.35 GeV, then the track parameters are recalculated using the assumption that the mass should equal the $J/\psi$ mass. In cases when the $J/\psi$ candidate is used to reconstruct a heavier particle such as the $B^+$, the constrained fit will improve the momentum resolution of that heavier particle. This fitting technique reduces the effect of the momentum scale dependence discussed in §6.3.2, by correcting the $J/\psi$ momentum using the actual Particle Data Group value of the $J/\psi$ mass.

After applying this selection a resulting sample of 284253 $J/\psi$ candidates are found, with the mass and kinematic distributions shown in Fig. 6.9. The Selection criteria applied to the muon tracks are illustrated in the plots shown in Fig. 6.10.

To build $B^+$ candidates, each $J/\psi$ candidate is examined, to see if it can be used to build a vertex with another charged track. Each $J/\psi$ is examined with all other tracks that are not associated to the primary vertex.

These tracks must satisfy the following criteria:

- The Track must have $p_T$ greater than 0.5 GeV.
- The Track must have a total momentum greater than 0.7 GeV.
- The Track must contain at least two hits in the SMT.

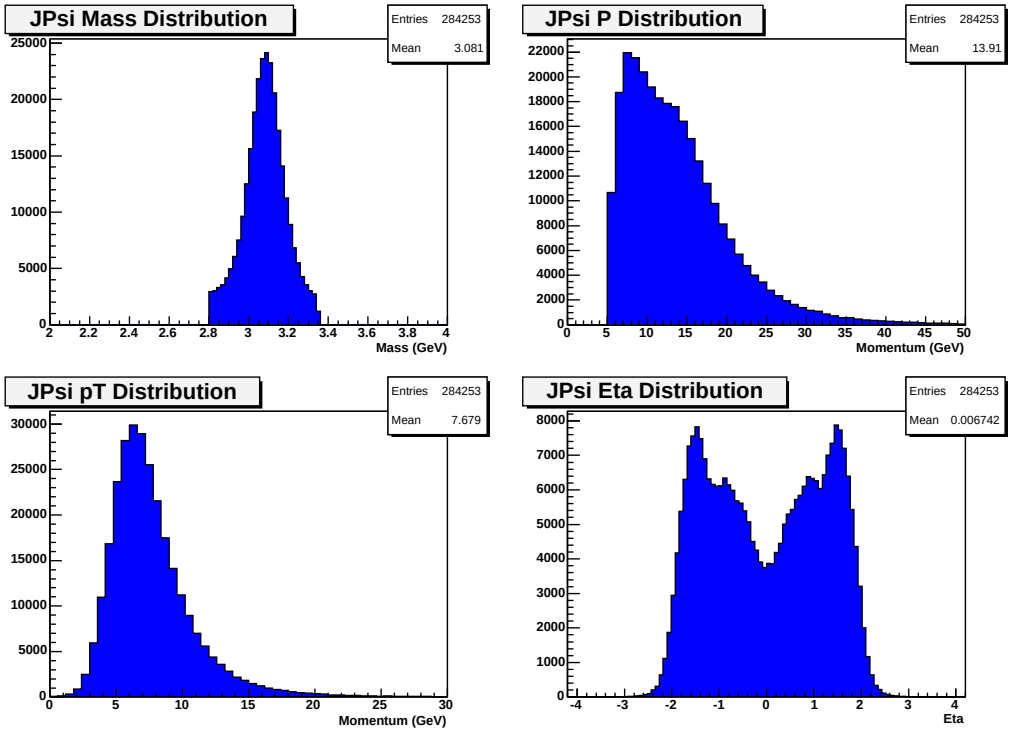


Figure 6.9: Kinematic distributions of the selected $J/\psi$ candidates. The Mass plot (top left) shows the mass selection range of 2.80 to 3.35 GeV constructed with the mass constrained fit. The Momentum distribution (top right) shows the 5 GeV momentum minimum imposed. The transverse momentum distribution is shown in the bottom left plot whilst the $\eta$ distribution is shown in the bottom right.

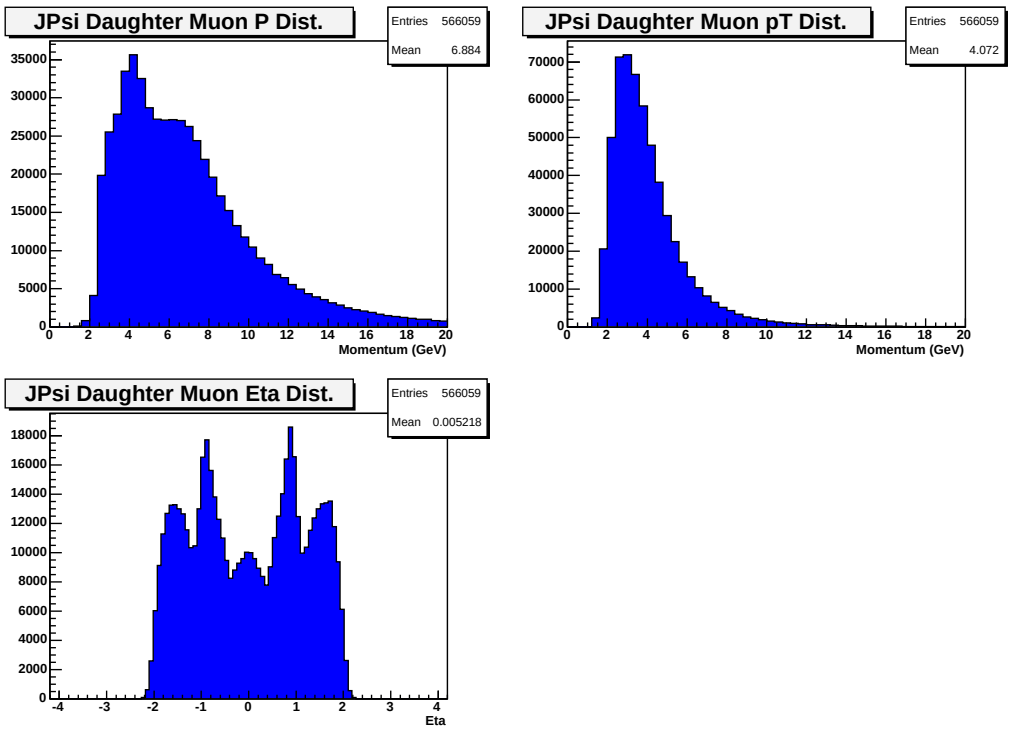


Figure 6.10: Kinematic distributions of the muon tracks used to construct $J/\psi$ candidates. The top left plot shows the momentum distribution of the muons used to construct the $J/\psi$ candidates, the top right shows the transverse momentum distribution, with the cut imposed at 1.5 GeV. The bottom plot shows the $\eta$ distribution of the muons.

- The Tracks Chi Squared value to be included in the primary vertex must be greater than 4.

- The Track must contain no more than two forward misses.

If the Track $p_T$ is between 0.5 GeV and 1.0 GeV further restrictions are applied to the track, requiring a higher value of Chi Squared result to the primary vertex of 9.

Tracks meeting these criteria are combined to form a vertex with the two muon tracks and the Chi Squared test is computed for the vertex. If the Chi Squared test results in a value greater than 16, then the candidate is discarded, since the tracks do not converge to form a good vertex. The distance from the primary vertex is computed and the vertex is required to be at least 4.5 cm away from the primary, or 5.5 cm away if the third track has a low $p_T$ value.

Finally the invariant mass of the $J/\psi$ candidate's corrected mass and the third track is computed and if the result is within the range 4.5 to 6.0 GeV, then the resulting vertex is taken as a $B^+$ candidate.

### 6.3.4 Results

Performing the steps above yields a set of 4938 three-track displaced vertices constructed from a $J/\psi$ candidate and a third charged track. The invariant mass distribution of these candidates is plotted in Fig. 6.11 and a fit performed using a sum of two gaussian functions, one for the background and one for the signal peak. Subtracting the Background fit function from the

data results in the distribution shown in Fig. 6.12. Subtracting the background function from the sample and defining the peak region to be plus or minus three standard deviations from the peak fit mean value results in a signal of 531 events and a Standard Error of $\pm 1.87 MeV$.

The Background function is fit over the entire range of the plot and is mainly due to the inability to distinguish the $K^+$ from the $\pi^+$ so that it is possible to construct vertices with charged pion tracks instead of a Kaon. QCD processes such as the underlying event give rise to many combinatoric contributions to the error, as there are many charged Kaon and pion tracks from the remnants of the hadronic collision that may cause false vertices.

Additionally, there is a contribution from other decay modes, for example from the decay $B^+ \rightarrow K^+ J/\psi \pi^+ \pi^-$.

The resulting mass value for the $B^+$ meson is deduced from the signal fit and background subtraction shown in Fig. 6.12 and is found to be:

$$M_{B^+} = 5281 \pm 1.870 \text{ MeV}(statisical) \tag{6.1}$$

The statisical uncertainty is the standard error on the peak fit, with the number of events in the peak being determined to be those events within $\pm 3\sigma$ of the peak mean value.

Variations of the background fit result in a variation of the number of signal events in the peak region, and this gives rise to a systematic uncertainty due to the type of fit used. To estimate this uncertainty, other types of backgound fit were used to compare and gave rise to differing numbers of

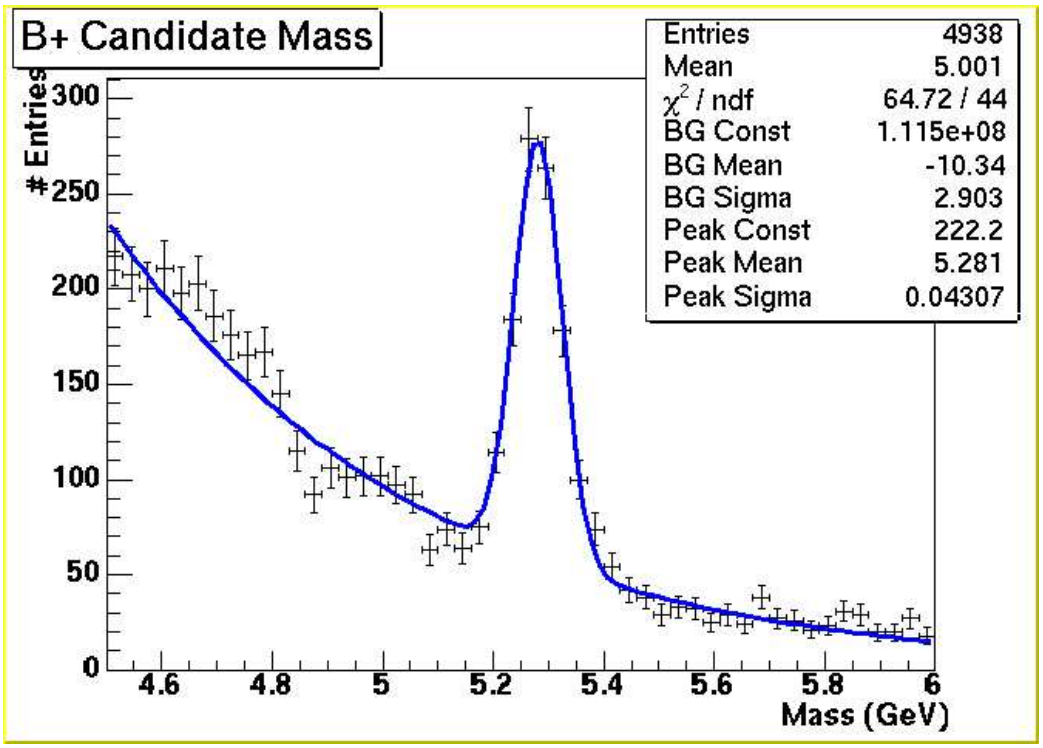


Figure 6.11: Plot of the invariant mass for $B^+$ candidates formed with the three track vertex technique. The result gives 4938 such candidates, and a summed gaussian fit is performed to model the background and signal distributions.

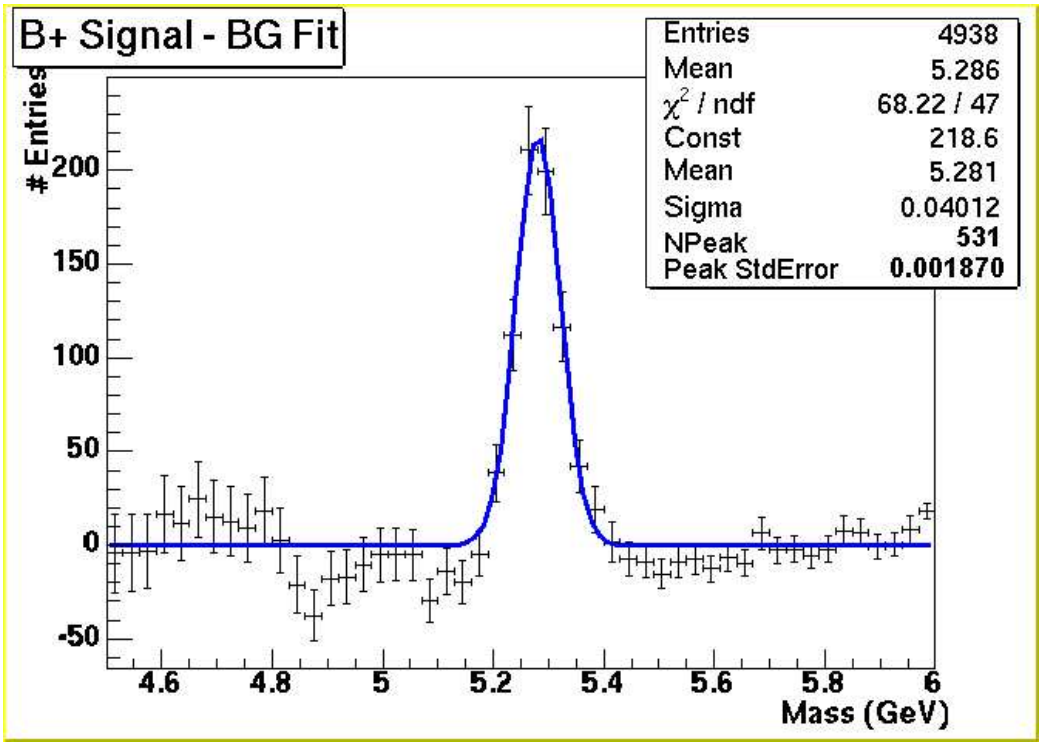


Figure 6.12: Plot of the invariant mass for $B^+$ candidates with the Background function subtracted from the data, resulting in 531 signal events in the peak region.

| Fit Type | Number of Signal Events | Standard Error/MeV |
|---|---|---|
| Gaussian | 531 | 1.870 |
| Linear | 602 | 1.527 |
| Exponential | 457 | 2.0228 |

Table 6.1: Table Showing the number of signal events and Standard Error for different types of background fits. The Gaussian fit used to determine the signal is shown in Fig. 6.12

signal events. A summary of these variations is presented in Table 6.1. From the values in the table a variation of $\pm 75$ signal events arises based on the choice of background fit. This change leads to a difference in standard error, based on the choice of fit and can be accounted for as a systematic error on the mass measurement of $\pm 0.3$ MeV.

Uncertainties on this measurement due to the preformance of the detector arise from several sources, including uncertainties in the beam spot position, uncertainties in the alignment of the tracking detector and the overall ability to construct accurate tracks. These contributions can be conservatively accounted for with a $\pm 1\%$ error on the mass due to momentum measurement uncertainty [88]. This results in an uncertainty of $\pm 52.81$ on the mass.

By far the largest contribution is from the momentum scale effects, similar to those shown for the $J/\psi$ measurement in §6.3.2. An estimate of this factor can be inferred by comparing a range of DØ mass measurments and calculating the difference between those measurements and the Particle Data

| Particle | DØ Mass/GeV | PDG Mass/GeV | Mass Difference/GeV |
|---|---|---|---|
| $\omega$ | $0.7747 \pm 0.0016$ | 0.7826 | 0.0079 |
| $\phi$ | $1.0093 \pm 0.0017$ | 1.0195 | 0.0102 |
| $J/\psi$ | $3.0718 \pm 0.0002$ | 3.0968 | 0.0250 |
| $\Upsilon(1S)$ | $9.423 \pm 0.008$ | 9.4603 | 0.0373 |

Table 6.2: Table showing comparison of masses as measured at DØ compared to the Particle Data Group values. These values are used to estimate the correction due to the momentum scale of the detector. [84][86][87]

Group values. A linear fit to these values is then performed and extrapolated to the $B^+$ mass, allowing an estimation of the correction that needs to be accounted for due to detector momentum response. A table of preliminary DØ mass measurements is given in Table 6.2 along with the Particle Data Group values and the differences between them. These quantities are plotted in Fig. 6.13 and yield an estimated correction for the $B^+$ mass of 32.24 MeV. Due to the preliminary nature of the measurements used to estimate this corrrection, there is an uncertainty associated with this correction that is estimated to be ten percent. Thus, the momentum correction is estimated to be $32.24 \pm 3.22$ MeV.

A Summary of error contributions is given in Table 6.3.

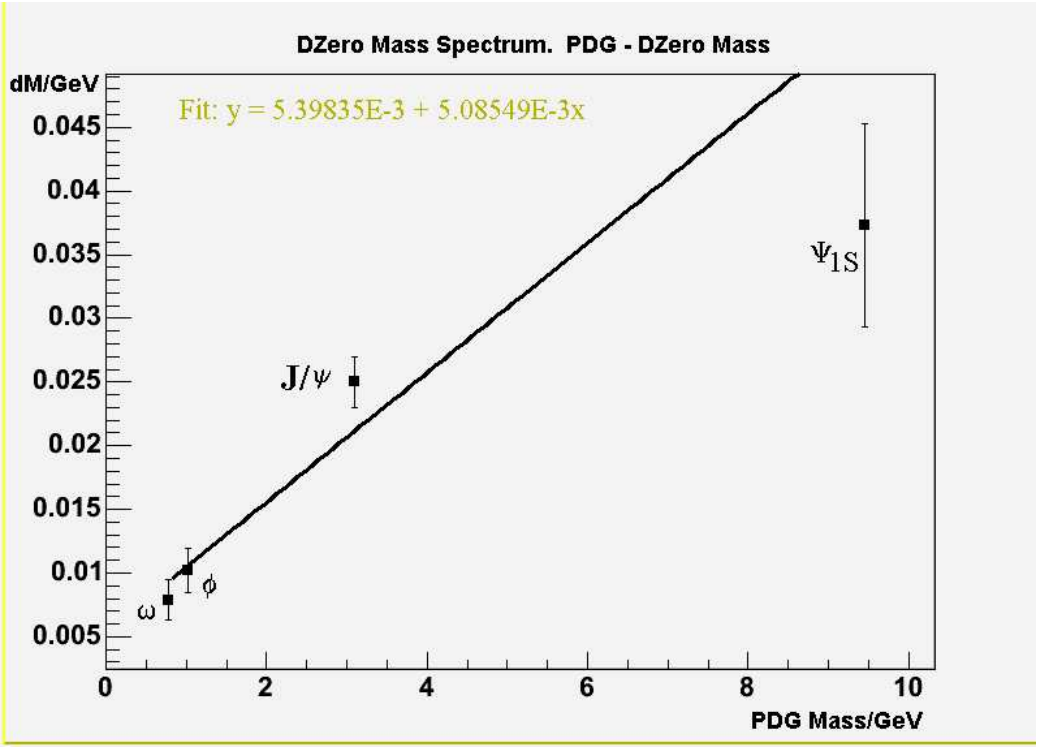


Figure 6.13: Plot of mass difference between PDG mass values an DØ mass measurements against the PDG mass. A Straight line fit is preformed and used to estimate a mass difference at the value of the $B^+$ mass. Using the fit shown in the graph, the mass difference at 5279.1 MeV is found to be 32.24 MeV, with an estimated uncertainty of ten percent.

| Uncertainty Cause | Value/MeV |
|---|---|
| Statistical Uncertainty | $\pm 1.870$ |
| Systematic Uncertainty due to background fit | $\pm 0.3$ |
| Uncertainty due to detector momentum measurements | $\pm 52.81$ |
| Momentum Scale Correction Factor | $+32.24 \pm 3.22$ |

Table 6.3: Table summarising the uncertainty contributions to the $B^+$ mass measurement in this analysis.

# Chapter 7

# Conclusions

## 7.1 Digital Front End Trigger Software

Since the completion of the Digital Front End (DFE) Trigger System and its offcial startup on April 15th 2003, it has become an integral part of the DØ experiment. The author's DFEWare software package is a fixture in the control room and has been remarked as being one of the more effective pieces of control software used by the shifters. Any problems with the DFE System are quickly diagnosed and fixed without requiring intervention of experts. Large scale system maintenance tasks and complete diagnostic tests have been built into the DFEWare package allowing experts to interact with the many components of the DFE hardware, either on a micro or macroscopic level as required.

## 7.2 Monte Carlo Event Selector

The author's d0_mess package is an indispensible part of Monte Carlo production at DØ allowing extremely fine control over the details of samples produced. The package is linked into most of the official generators and will form a part of many analyses. The flexible nature of the d0_mess software means that it is being expanded to allow further customisation of the Monte Carlo produced, including modelling the detectors trigger turn on curves.

## 7.3 Workflow Software

By far the greatest piece of work performed by the author has been the development of the workflow and data handling tools developed within and around the mc_runjob package. Using this package enables processing tasks to be expressed as a set of physics parameters in a language that is easily understandable by physicists. This has shortened the software learning time experienced by new members of the collaboration and enabled veteran collaborators to run processing tasks much more easily, thus allowing them to focus on improving their analysis work.

Remote processing has been made easier as mc_runjob makes it straightforward to create generic self contained jobs that can be run on distributed farms without requiring full installs of the DØ code distribution. Jobs are now produced and executed in a well defined framework that allows the job to dynamically react to conditions arising during execution and provide an

intelligent interface to the executing processes.

The developments in distributed computing made by mc_runjob have allowed DØ to take a leading role in the prototyping of Grid computing models. The advances made in the field of workflow management are being shared between multiple experiments in order to allow collaboration between different computing resources and is leading to interest from other experiments and external Grid organisations.

mc_runjob is emerging as the de facto standard for managing remote processing for all DØ processing tasks including remote raw data reprocessing, Monte Carlo production and distributed user analysis.

## 7.4 Measurement of the Mass of the Charged B Meson

The value of the $B^+$ mass of

$$M_B = 5281 \pm 1.870\,(\text{statistical.}) \tag{7.1}$$

$$\pm 0.3\,(\text{systematic due to Fit.}) \tag{7.2}$$

$$\pm 52.81\,(\text{systematic due to momentum measurements}) \tag{7.3}$$

$$+32.24 \pm 3.22\,(\text{Momentum Scale Correction}) \tag{7.4}$$

found in the tracking study presented, while not being a complete and rigorous study, does provide a result in agreement with the Particle Data Group

world average value of $5279.0 \pm 0.5$,. This result shows that the DØ Tracking system is working, enabling the experiment to pursue a full program of B Physics studies in Run II. The `AATrack` Algorithm produces results that are consistent with well known measurements and has been adopted by DØ as the official tracking algorithm in use for the Run II period. As more data is collected and systems such as the Central Tracking System become more refined the tracking capabilities of the experiment will increase as will the statistics available for study resulting in the ability to produce physics measurements that will contribute to out understanding of the fundamental nature of matter.

# Bibliography


[1] T. Pratchett, *Night Watch, A Discworld Novel*, p83.

[2] K. Hagiwara *et al.* (Particle Data Group), Phys. Rev. D **66**, 010001 (2002)
`http://pdg.lbl.gov`

[3] K. Hagiwara *et al. Particle Data Group, Review of Quantum Chromodynamics*, Phys. Rev. **D 66**, (2002) 010001.
`http://pdg.lbl.gov`

[4] C. Quigg, *Gauge Theories of the Strong, Weak and Electromagnetic Interactions* Pub. Addison-Wesley.

[5] CTEQ Webpage,
`http://www.phys.psu.edu/ cteq/`

[6] F. Olness, *Heavy Quark Production*, Proceedings of the 2002 CTEQ Summer School, Madison, Wisconsin.

[7] R. Field, *The Sources of b-quarks at the Tevatron and their correlations*, Phys. Rev. **D 65** (2002) 094009.

[8] DØ Collaboration, *Inclusive $\mu$ and b quark production cross-sections in $p\overline{p}$ collisions at $\sqrt{s} = 1.8/, TeV$*, Phys. Rev. Lett. **74** (1995) 3548.

[9] DØ Collaboration, *The $b\overline{b}$ cross-section and angular correlations in $p\overline{p}$ collisions at $\sqrt{s} = 1.8/, TeV$*, Phys. Lett. **B 487** (2000) 264.

[10] CDF Collaboration, *Measurement of the bottom quark production cross-section using semileptonic decay electrons in $p\overline{p}$ collisions at $\sqrt{s} = 1.8/, TeV$* , Phys. Rev. Lett. **71** (1993) 500.

[11] CDF Collaboration, *Measurement of bottom quark production in $1.8\,TeV$ $p\overline{p}$ collisions using semileptonic decay muons*, Phys. Rev. Lett. **71** (1993) 2396.

[12] P. Nason, S. Dawson, R.K. Ellis, *The total cross-section for the production of heavy quarks in hadronic collisions*, Nucl. Phys. **B 303** (1988) 607.

[13] W. Beenaker, W.L. van Neerven, R. Meng,G.A. Schuler and J. Smith, Nucl. Phys. **B 351**, 507 (1991).

[14] M.L. Mangano, P. Nason, G. Ridolfi, *Heavy quark correlations in hadron physics at next-to-leading order*, Nucl. Phys. **B 373** (1992) 295.

[15] D.K. Fein. *Tevatron Results on b-quark Cross Sections and Corrlations* FERMILAB-Conf-99/050-E.

[16] G. Sterman *Introduction to the Parton Model and Perturbative QCD*, Proceedings of the 2002 CTEQ Summer School, Madison, Wisconsin.

[17] D. Griffith, *Introduction to High Elementary Particles* Pub. Wiley.

[18] M. Neubert, *Heavy Quark Symmetry*, Phys. Rep. **245**, (1994) 259

[19] CDF Collaboration, *Measurement of b-quark fragmentation fractions in $p\overline{p}$ collisions at $\sqrt{s} = 1.8TeV$*, Phys. Rev. Lett. **74** (1995) 2632

[20] W. Taylor *Prospects for B Lifetimes, Oscillations and CP Violation at DØ* , Talk given at HCP 2002 Conference, October 2002.

[21] Fermilab Beams Division *Run II Handbook*, 2001

[22] DØ Collaboration, S. Abachi *et al The DØ Detector*, *Nucl. Instrum. Meth.* **A338** (1994) 185-253.

[23] DØ Collaboration, *The DØ Upgrade: The detector and it's physics.* Fermilab Pub-96/357-E, July, 1996.

[24] J. Ellison, *The DØ detector upgrade and physics program*, `arXiv:hep-ex/0101048`.

[25] *Run II Luminosity Monitor Home Page*
`http://www.hep.brown.edu/lm/`

[26] C. Miao, *The DØ Run II Luminosity Monitor*, Talk given at the 6th International Conference on Advanced Technology in Particle Physics, Como, Italy, October 1998.
`http://www.hep.brown.edu/lm/como_LM.pdf`

[27] *DØ Silicon Tracker Technical Design Report.* DØ Note 2169, July 1994.

[28] M. Roco, *The Silicon Microstrip Tracker for the DØ Upgrade*, DØ Note 3553, October 1998.

[29] *DØ Fibre Tracker, Technical Design Report*, `http://d0server1.fnal.gov/projects/SciFi/cft_home.html`

[30] J. Brzezniak *et al, Conceptual Design of a 2 Tesla Superconducting Solenoid for the Fermilab DØ Detector Upgrade*,DØ Note 2167, May 1994.
`http://www-d0.fnal.gov/solenoid`

[31] M.Adams *et al. Design Report of the Central Preshower Detector for the DØ Upgrade.* DØ Note 3104, January 1996.

[32] A. Gordeev *et al. Design Report of the Forward Preshower Detector for the DØ Upgrade.* DØ Note 3445, May 1998.

[33] L. Bagby *et al. Calorimeter Electronics Upgrade for Run II, Technical Design Report.*
`http://www-d0.fnal.gov/hardware/cal/intro/calupgrade.htm`

[34] Calorimeter Upgrade Overview Web Page,
`http://www-d0.fnal.gov/hardware/cal/intro/calupgrade.htm`

[35] J. Butler *et al. The DØ Muon System Upgrade*, DØ Note 2780, January 1996.

[36] Muon System Upgrade Home Page,
http://www-d0.fnal.gov/
hardware/upgrade/muon_upgrade/muon_upgrade.html

[37] G. C. Blazey, *The DØ Run II Trigger*, FERMILAB-CONF-97-395-E, Talk given at 10th IEEE Real-Time Computer Applications in Nuclear, Particle and Plasma Physics (RT 97), Beaune, France, 22-26 Sep 1997.

[38] M. Abolins *et al. The Level One Framework*, DØ Note 328, June 1998.
http://www.pa.msu.edu/
hep/d0/ftp/l1/framework/l1fw_tdr_05june98.txt

[39] L2 Trigger Home Page
http://www.pa.msu.edu/hep/d0/l2/index.html

[40] D. Edmunds *et al. Technical Design Report for the L2 Global Processor*, DØ Note 3402, February 1998.

[41] A. Boehnlein *et al. Description of the DØ L3 Software Components*, DØ Note 3630, April 1999.

[42] DØ CTT Group Webpage,
http://www-d0online.fnal.gov/
www/groups/cft/CTT/online/ctt_main.html

[43] L. Babukhadia, M. Martin, *Track and Preshower Digital Trigger in DØ* , DØ Note 3980, September 2002.

[44] F. Borcherding *et al, Central Track Trigger Technical Design Report*, DØ Note 3551.

[45] G. Savage, *Private communication on the nature of the hardware and software chains for EPICS and the DFE System.* July 20003.

[46] J. Olsen *Standard Register Interface for DFE and MIXER boards.*DØ Engineering Note 2000-01-28a

[47] J. Olsen *DFEC Commands and Examples*DØ Engineering Note 2001-04-16a

[48] J. Olsen *Mapping between DFEC device numbers and physical FPGAs on DFE Daughterboards*DØ Engineering Note 2001-09-25a

[49] DFE Hardware Webpage,
`http://www-d0online.fnal.gov/`
`www/groups/cft/CTT/online/docs/`

[50] Xilinx Inc. Webpage,
`http://www.xilinx.com`

[51] Compact Flash Organisation Webpage,
`http://www.compactflash.org`

[52] EPICS Webpage,
`http://www.aps.anl.gov/epics/`

[53] G. Savage, *CaChannel* August, 2000,

`http://d0server1.fnal.gov/www/online_computing/`

`projects/controls/epics_python/cachannel.pdf`

[54] DFEWare Webpage,

`http://www-clued0.fnal.gov/dfe_ware`

[55] DØ CTT Group,*Private Communication on firmware versioning issues*

September 2001.

[56] Xilinx Xcell Journal Article on the DØ DFE System
`http://www.xilinx.com/publications/`

`xcellonline/partners/xc_pdf/xc_higgs44.pdf`

[57] J. Olsen *et al. The DØ Central Track Trigger.*

To be submitted to IEEE Transactions in Nuclear Science.
`http://www-d0online.fnal.gov/www/groups/`

`cft/CTT/online/Overview/ieee_ctt.pdf`

[58] J.Kowalkowsky *et al. Framework Users Guide*

`http://www-d0.fnal.gov/d0dist/dist/`

`releases/test/framework/doc/user_guide.ps`

[59] RCP System Webpage:

`http://cdspecialproj.fnal.gov/d0/rcp/index.html`

[60] Particle Data Group, *Monte Carlo Particle Numbering Scheme*

`http://pdg.lbl.gov/`

[61] d0_mess Webpage:

`http://www-clued0.fnal.gov/d0_mess`

[62] Pythia Generator Webpage

http://www.thep.lu.se/tf2/staff/torbjorn/Pythia.html

[63] Herwig Generator Webpage

http://hepwww.rl.ac.uk/theory/seymour/herwig/

[64] Onetop Generator Webpage

http://www-clued0.fnal.gov/hit/top/onetop/

[65] QQ Generator Webpage

http://www.lns.cornell.edu/public/CLEO/soft/QQ/

[66] EvtGen Generator Webpage

http://www.slac.stanford.edu/ lange/EvtGen/

[67] GEANT Detector Description and Simulation Tool:

http://wwwasd.web.cern.ch/wwwasd/geant/

[68] D0gstar Webpage:

http://www-d0.fnal.gov/

computing/MonteCarlo/simulation/d0gstar.html

[69] R. Field *Min-Bias and the Underlying Event at the Tevatron and the LHC*,

http://www.phys.ufl.edu/ rfield/cdf/JetMET_fixed_5-6-03.pdf

[70] D0sim Webpage:

http://www-d0.fnal.gov/

computing/MonteCarlo/simulation/d0sim.html

[71] D0Trigsim Webpage:

`http://www-d0.fnal.gov/computing/trigsim/trigsim.html`

[72] D0Reco Webpage:

`http://www-d0.fnal.gov/computing/algorithms/`

[73] ROOT: An Object-Oriented Data Analysis Framework:

`http://root.cern.ch/`

[74] A. Boehnlein, *DØ Computing Model*, Talk given at the DØ Workshop, Beaune, June 2003.

[75] L. Lueking *et al. The Data Access Layer for DØ Run II: Design and Features of SAM*, CHEP March, 2000.

[76] SAM Webpage

`http://d0db.fnal.gov/sam/`

[77] Python Language Website:

`http://www.python.org`

[78] A. Baranovski *et al. Management of Grid Jobs and Information within SAMGrid.* CHEP03 Proceedings, March, 2003.

[79] The CMS Experiment.

`http://cmsinfo.cern.ch/Welcome.html/`

[80] Shahkar project web page.

`http://www.uscms.org/s&c/testbed/Tiger/SHAHKAR/Shahkar.htm`

[81] G. Graham, D. Evans *DZero Monte Carlo Production Tools* CHEP01 Proceedings, September 2001.

[82] G. Graham, D. Evans, I. Bertram *Mc_RunJob: A High Energy Physics Workflow Planner for Grid Production Processing* CHEP03 Proceedings, March, 2003.

[83] G. Borissov *Technical Details of AA Tracking*,
Talk given at the All DØ Meeting. Feb 28th, 2003.

[84] Particle Data Group: K. Hagiwara *et al., Phys. Rev. D 66, 010001* (2002)

[85] DØ B Physics Group,
Approved Conference Results Webpage:
`http://www-d0.fnal.gov/`
`Run2Physics/ckm/Moriond_2003/index2.html`

[86] R. Van Kooten *Mass Scale, Average momentum resolution corrections on dimuon resonances*
Talk given at the B Physics/Tracking Meeting. Dec 18th, 2003.

[87] D. Bauer *Upsilon Production at DØ*
Talk given at Pheno2004. Apr 26th, 2004.

[88] G. Borrisov,*Private Communication on the nature of systematic errors in the DØ tracking system*