Lars A. Gundersen

Development of Read Out Electronics for the ATLAS Silicon Detector System

Lars A. Gundersen

Development of Read Out Electronics for the ATLAS Silicon Detector System

Thesis for the Cand. Scient. degree
at the Physics Department,
the University of Oslo

The Universe may
be as great as they say.
But it wouldn't be missed
if it didn't exist

Piet Hein

1905–1996

Hardware: Apple Macintosh PowerBook 165

Software: Microsoft Word 5.1a
Design Science Equation Editor 1.0b
Deneba Software Canvas 3.0.1
Persistence of Vision POV-Ray 3.0
DeltaPoint DeltaGraph Pro 3.5
Adobe PhotoShop 2.5.1

Fonts: Times for the body and in the figures
Bookplate for the headings, headers and footers
Geneva for the figure text, footnotes and the code

All figures by Lars A. Gundersen, except 4.4, parts of 2.1 and 2.2.
Layout by Lars A. Gundersen.

Preface

This thesis is part of the master degree at the University of Oslo (UiO), the Department of Physics. The master degree at UiO has an estimated time-scale of one and a half year of project-work and theoretical studies. The work this thesis describes and discusses took part mainly from March 1994 to March 1995.

It is fair to say that my thesis-project did not turn out to be quite what I had expected. Fortunately, because I had imagined sitting in a half-lit laboratory at UiO for most the time. There, I thought, I would conduct some experiments and possibly build a piece of equipment or two to 'solve' my thesis assignment.

Instead, I found myself travelling abroad, at one time all the way to Japan! I found myself hurled into a huge, ongoing project with participants from all over Europe and beyond. I bit bewildered at first, I soon found it much more interesting and in all respects better than the kind of thesis I had imagined.

At CERN, Switzerland, I learned about a broad range of topics concerning the project this thesis is about. I learned of course about the technical details, requirements, limits and methods to overcome the limits. But I also learned about the administrative side, about the never-ending stream of suggestions, decisions, meetings and comparisons necessary to keep the project in movement along the right path. This, together with what I learned about the infra-structure of this project and of CERN is knowledge that I value as much as the technical insight gained through this thesis-project.

For all this I first and foremost have to thank my supervisor, Dr Steinar Stapnes. He dragged me into everyday work at CERN. He sent me to Japan. And he helped me a lot both at CERN and with the writing of this thesis. I already have expressed my gratitude towards him for making my most important discovery possible by sending me to CERN as a summerstudent in 1994. That discovery was Ans, a summerstudent from Belgium. Although it may not be relevant to my thesis, and consequently does not appear elsewhere in this text, I have even learned a lot about Belgium during the time of my thesis-project!

I would also like to thank Shaun Roe at CERN for having the patience to explain and help a freshman. Sometimes a great deal of patience was necessary, and Shaun possessed that. Other members of the CERN team deserve thanks, too, for making me feel like a part of that team: Jan Kaplon, Richard Brenner, Alan Rudge and Peter Weilhammer, the group leader. Also thanks to Frants for helpful input on the layout.

I am glad this turned out to be something different than what I had imagined. I had never thought I would meet so many new people, get so many new friends, eat so many new dishes and see so many new places during my thesis project. And I had never thought I would acquire knowledge of such a broad range of topics as I did.

Oslo, May 1996
Lars A. Gundersen

Table of Contents

1: Introduction

In this thesis I will discuss topics closely connected to an ongoing project at CERN, Switzerland. The project is the planning, design and building of the next, big particle accelerator and its detectors. This thesis does not attempt to cover the project as a whole, that would be a bit too ambitious, to say the least. Instead, I concentrate of the part of the project that the group at the University of Oslo has worked on.

Chapter 1 is an introduction to the thesis-project. I will in this chapter present CERN and the project in question. Chapter 1 is also meant to put the thesis into its proper context by showing where in the project it belongs.

The aim of the work undertaken during the master degree was to gain a thorough understanding of the electronic read-out system for a specific part of the ATLAS-detector, and more generally to gain an understanding of the defining parameters involved in designing and building a high-speed data readout system for silicon detectors.

1.1: CERN

CERN, the European Organisation for Nuclear Research is an organisation and a laboratory for research mainly targeted at sub-nuclear physics. It is situated on the border between Switzerland and France, approximately 10 km outside the city of Geneva. CERN celebrated its 40th anniversary in the autumn of 1994. From the very beginning, it has had a leading edge in the field of experimental particle-physics, and many industrial and technical offsprings from this primary activity have been seen over the years. The World Wide Web is one of the more recent, well known examples.

The core of the CERN activities is, and always has been, the particle accelerators and its detectors. One of the first experimental machine built at CERN was the *Proton Synchrotron* (PS). It accelerated and provided protons for fixed target experiments at a beam-energy of 26 GeV.

Over the years several accelerators and detectors have been built. As old accelerators have become outdated, they have frequently been transformed into pre-stages of newer, more powerful accelerators. This has provided massive cost-savings.

The *Super Proton Synchrotron* (SPS) succeeded the PS in 1981. With a centre of mass energy of 900 GeV, it was capable of achieving an equivalent resolution of 10^{-16} cm, as given by the DeBroglie wavelength $l = \hbar / p$. The W and Z bosons of the weak interaction were discovered at the SPS[5], thus giving important confirmation of the electroweak theory. The SPS is still in operation.

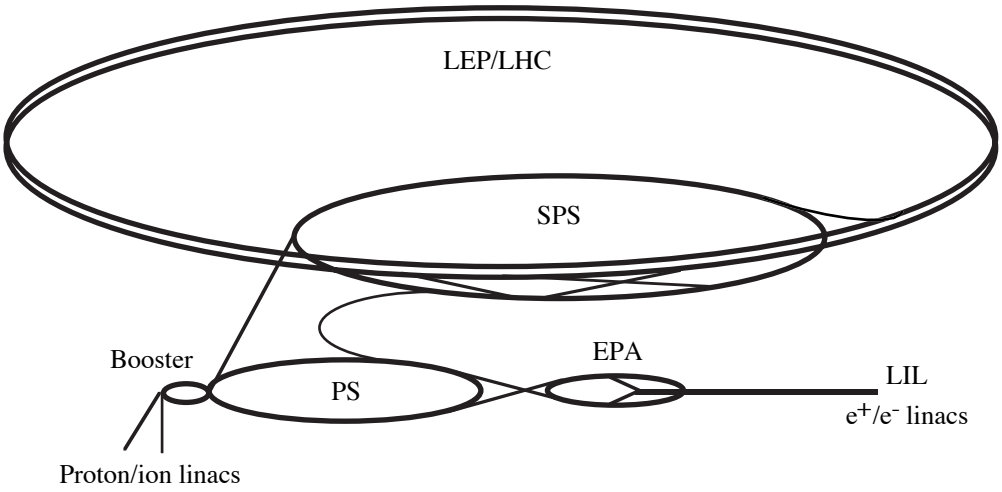


Figure 1.2: Schematic diagram of the CERN accelerator complex

The planning of a another new machine started as early as 1977, i.e. well before the SPS was in operation. A charged particle radiates energy in a circular orbit, with energy-loss

$$\Delta E \sim \frac{1}{r \cdot m^4}, \qquad \text{Eq. 1.1}$$

where r is the radius of the orbit and m is the particle mass. Thus the need for high beam-energy dictated a large-radius accelerator, especially since electrons and positrons were chosen as the accelerator particles. With their low mass, electrons loose a large fraction of their energy per turn. The reason for choosing $e^{\pm}$ as the accelerator particles was mainly to be able to study the production of the Z boson (with a mass of approximately 90 GeV) from electron-positron annihilation.

A circular tunnel with a radius of 4.3 km was drilled underground to house *LEP*, the *Large Electron Positron* accelerator. The machine collides electrons with positrons at energies up to 100 GeV per beam. Following the philosophy of re-use and cost-efficiency, space was provided for *two* accelerators inside the LEP ring when it was built: LEP itself and a future accelerator, not yet planned at the time.

LEP has proved an invaluable source of experimental data to test the Standard Model to very high precision. It has successfully searched for particles in the energy range up to approximately 190 GeV. While 190 GeV represented a substantial leap in beam-energy and thus 'exploration power', it is still not enough to exhaust the Standard Model. Yet higher energies are needed to search for heavier particles. This includes the search for the Higgs bosons, which remains undetected, yet predicted by the Standard Model. Other topics of interests in the energy-regime of 1 TeV would be the search for particles predicted by Grand Unification theories and Supersymmetry. *LHC*, the *Large Hadron Collider*, is the machine designated to undertake these jobs.

1.2: LHC

Experiments at higher energies are a constant demand in particle physics. In this century, development in physics has followed a path were theorists and experimentalists have taken turns pushing our understanding of the processes we study into deeper levels. One of the main reasons for this is probably that experiments are built to test the predictions of theory, and that theory often is constructed to explain some unexpected result from an experiment. LHC is the most powerful experimental machine yet to emerge from this 'race' between theory and experiment.

LHC will occupy the same tunnel as LEP presently occupies, but with its own beam pipes and detectors. As I said in the previous section, the idea originally was to put LHC ontop of the LEP beam-pipe. However, this has turned out to be to expensive. Since it is believed that LEP will have exhausted its discovering-possibilities by then, the solution will be to remove the LEP beam-pipe installation in the year 2000, before the LHC accelerator is installed in the tunnel. This will remove the constraint that LHC must follow the exact LEP geometry. However, it will also remove the option of e-p collisions, which originally was planned. Instead, enough room is kept free over the LHC installation that a lepton ring can be installed there in the future, possibly using LEP components.

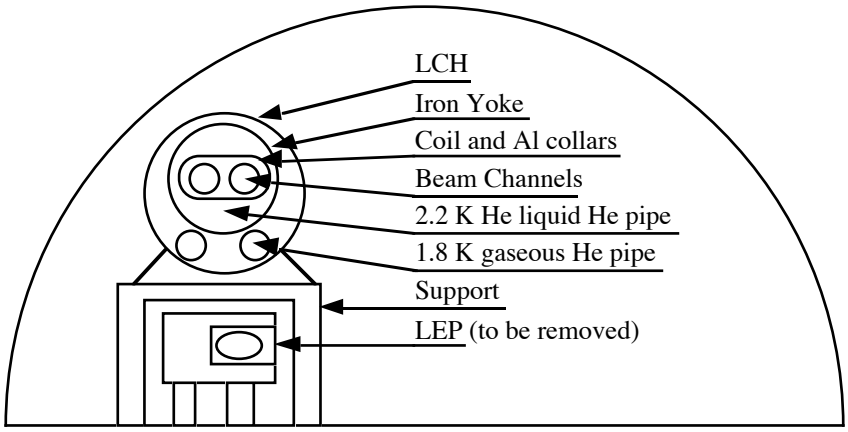


Figure 1.3: The LEP/LHC tunnel as it was planned. LEP will, however, be removed before LHC is installed.

Theoretical physicists have high hopes for interesting new physics at energies around 1 TeV. Therefore, a centre-of-mass energy sufficiently high to produce particles in that range is planned for LHC. Furthermore, very high luminosity is needed. When looking for new particles or new particle interactions, it is important to be able to extract very subtle effects from the data, effects that may only occur seldom and/or produce a small signal over a large noise-background. The higher the luminosity, the more data can be gathered in a given time, and the bigger amount of statistics is available.

The LHC accelerator will mainly collide protons with protons, at a centre-of-mass (CM) energy of 14 TeV and a luminosity of $10^{34} cm^{-2} s^{-1}$[10]. Alternative modes are also possible: LHC can collide heavy ions, more specifically Ph-Ph, produced in the existing CERN accelerator-complex. The CM energy will in that case be a formidable 1150 TeV, and the luminosity $10^{27} cm^{-2} s^{-1}$ [10].

The possibility of colliding electrons with protons has been mentioned. The luminosity for this mode of operation is expected to be $10^{32} cm^{-2} s^{-1}$ [10].

Both the accelerator and LHC's detectors put great demands on available technology. It is fair to say that the technology to realise LHC was not available at the time its parameters were decided. The parameters were instead based on what was needed, and it was foreseen that the necessary technology would become available during the time the machine was planned and built. To propel the advancement of the needed technology, several *Research and Development* (RD) collaborations where set up with participants from Universities in many of the CERN memberstates. The collaborations were assigned specific tasks, typically proposing solutions to various technical challenges, and once a solution was agreed upon, to develop it into working equipment. The High Energy Particle Physics Group at the University of Oslo joined one such collaboration. This collaboration, RD20, was responsible for a certain part of the project, the silicon tracker, which is the basis for this thesis.

2: The ATLAS Detector

At the new LHC machine, scheduled to come into operation in 2004, two proton-proton detectors are planned. One of them is the *ATLAS* detector. ATLAS is the not-too-imaginative acronym for A Torodial LHC ApparatuS. The LHC particle accelerator itself and the other detectors do not enter in detail into this thesis, which concentrates on part of the electronic read out system of the ATLAS detector.

Before I start discussing that electronics, however, it is useful to present a more general discussion of the ATLAS detector, in order to put the following chapters into the right context.

2.1: ATLAS Overview

The proposed ATLAS particle detector will be a formidable 26 m long and have a diameter of 20 m. It should be quite obvious that it is a challenging task to plan and build such a huge, yet intricate machine.

A cut-through of ATLAS can be seen below. The detector can be divided into a *barrel* part and the two *endcaps* at $|\theta|<45$ and $135<\theta<225$. These again can be divided into layers of different sub-detectors. In the barrel, the sub-detector layers lay like concentric cylinders, outwards from the beam line. The LHC particle beam, or rather beams, carry *primary* particles in opposite directions. The beams are brought to collide in the very centre of the ATLAS detector. The particle collisions will generate new particles which will travel outwards, into the detector. The collection of signals generated in the detector layers from these particles is referred to as an *event*. The different sub-detectors will have the tasks of detecting vertex, tracks, momentum and energy of the outcoming particles.

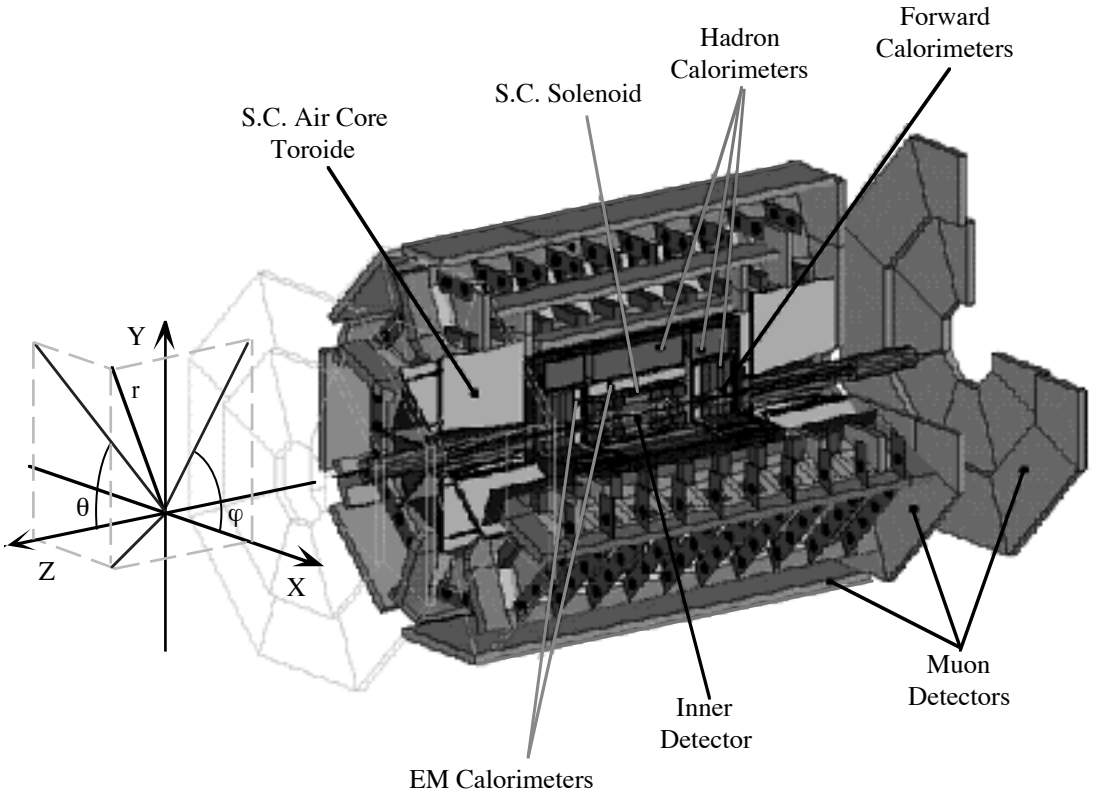


Figure 2.1: Cut-through of the planned ATLAS detector, showing the different layers of particle detection and the appropriate coordinate system[1].

The coordinate system's zero point is put at the centre of the detector, the *primary vertex*, where the particles from the beams collide. Keep the scale in mind when looking at the figure: The diameter of the complete detector is 20 m!

2.2: The Inner Detector

The innermost detector of the barrel is the *Inner detector*, on which this thesis will concentrate. The Inner detector is sub-divided into several layers, to which I will return in more detail in section 2.7. The Inner detector plays an important role in reconstructing the tracks of charged, outcoming particles. The track reconstruction is performed by interpolating hits in the different layers. The Inner Detector does not stop the particles, and it is designed to have minor effect on their momenta, for reasons that soon will be explained.

1 The coordinate-system is added by me, the rest of the figure is from [11].

2.3: The EM Calorimeter

Enclosing the Inner Detector is the *Electromagnetic Calorimeter*. It is designed to stop outcoming electrons, positrons and photons. Either of the three particles will deposit all its energy in the EM Calorimeter. The deposition is strongly dominated by two 'alternating' processes: pair production, in which a photon is converted to a $e^{\pm}$ pair, and bremsstrahlung, in which $e^{\pm}$ particles radiate photons in the presence of the field from an atomic nucleus. The two processes cause an electromagnetic shower to develop in the calorimeter, and the original energy of the particle is inferred from measuring the total ionization of the shower.

The EM Calorimeter which will be used in ATLAS is of the so-called LAr 'accordion' sampling type[9], the name stems from the geometry of the active/passive layers. The active material, i.e. the material in which the signal is measured, will be Liquid Argon. The shower develops in the passive material, and the signal can be measured in the Liquid Argon cells when the shower particles ionize the Argon atoms.

2.4: The Hadron Calorimeter

The next layer is the *Hadron Calorimeter*, designed to stop hadrons and measure their energy in a similar way, although the shower is a hadronic one. The development of the shower is far more complex in the hadronic case, though, since many more processes contribute than is the case for an EM shower. Since the development of such showers obeys statistical laws, it follows that the energy resolution of a Hadron Calorimeter usually is worse than for an EM Calorimeter. Whereas $\Delta E / E \approx 0.10 / \sqrt{E}$ is a typical value for the resolution of an EM Calorimeter, a factor 5 worse is common for Hadron Calorimeters. E is here measured in GeV.

The Hadron Calorimeter used in ATLAS will be of the traditional sampling type, with sandwiched layers of iron and scintillator. The iron is the passive (absorber) material, in which the shower develops, and the scintillators are the active sampling layers measuring the ionization.

2.5: The Muon Detector

The outermost layer of the ATLAS detector is the *Muon Detector*, which takes care of the measurement of the heavy muon leptons. The detector technology will be some form of drift cells, in which a passing particle ionize gas atoms. The electrons thus freed will drift towards the anode (biased sense-wires) in an electric field and create an electric pulse on the wires. However, the field is so strong close to the wires that the incoming electrons

will acquire enough momentum to ionize new gas-atoms near the wires. It is usually the positive ions drifting towards the electrode from this ionization-process that forms the signal which is used.

The Muon detector is the outermost layer because muons have a very high penetrating power, and little signals are seen from muons in the other layers. All the other particle-types are stopped in the calorimeters, thus a particle inducing a signal in the muon detector has to be a muon.

Neutrinos escape detection, as the only ones of the known elementary particles that are believed to be present in the detector layers. Their existence and properties are inferred using the appropriate lepton number conservation laws and transverse momentum conservation on the directly observed particles.

2.6: The Magnets

Two magnet-systems are also part of ATLAS. The magnets are not detectors in themselves, but serve to enhance the particle identification power of the detector.
Enclosing the Inner Detector is a strong solenoid superconducting electromagnet. It produces a magnetic field in the range of 2 Tesla which is parallel to the beam-line. The field will bend a charged outcoming particle more or less, according to $\vec{F} = q\vec{v} \times \vec{B}$ and also depending on the mass of the particle. The curvature is seen when reconstructing a particle track from signals in the layers of the Inner Detector, and is used to calculate the outcoming particle's momentum.

The muon detector is also enclosed in a magnetic field. The field is produced by a system of torodial magnets, sat up in a way that produces a magnetic field around the barrel so that $\vec{B} \perp \vec{u}_z$. This will bend a muon going through the Muon Detector, and the curvature will reveal its momentum and charge.

2.7: The Inner Detector Layers

This thesis concentrates on the innermost layers of the Inner Detector and the readout of the data from these. The Inner Detector is the first detector an outcoming particle will penetrate, as we have seen. Different solutions for this detector has been proposedwithin ATLAS collaboration, and several different detector technologies will be applied. The Inner detector must meet certain requirements, amongst which are:

- Radiation-hardness. The innermost layers are exposed to a massive particle flux, and must be able to withstand the flux with as little deterioration as possible.
- Space-resolution. An important aspect, as these layers act as the primary detectors for determining the point of impact by track reconstruction.

The layers of the Inner Detector will need to reconstruct tracks in a very high track-density environment.

- Time-resolution, dead time. The shorter the time between a signal and the next signal the detector can handle, the more data per time-unit can be gathered. The time resolution is defined by the frequency of the primary particle collisions to be better than 25 ns.
- Radiation length. If the material used were to significantly affect the outcoming particles, the job of reconstructing tracks would be much harder. The radiation length of the material must therefore be small.
- Cost. As with all components in this and other experiments, the costs of material, design and construction are major factors.

The 4th point is maybe the most striking difference between the Inner Detector and the enclosing calorimeters. The Calorimeters *must* stop the particle to do a satisfactory job, the Inner detector should affect it as little as possible.

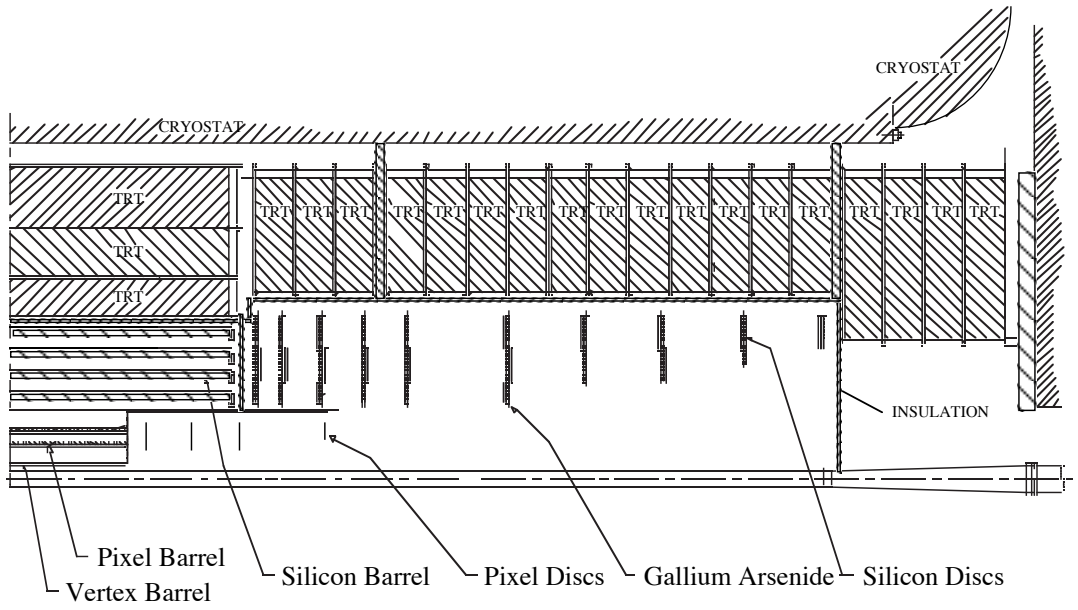


Figure 2.2: Schematic drawing of the Inner Detector layout[13][2].

Various technology solutions for the Inner Detector are possible. Some have been used before, and are therefore well tested. These frequently do no longer have the required resolution, either or both in time and space, to be of practical use. Others are still highly experimental, and not yet at a level of development that they can be seriously considered. For example, experiments using *diamonds* as transducers have been carried out. Although the bandgap in a diamond lattice is 5.5 eV, a factor of almost 5 higher than in silicon, a transducer-scheme similar to that used for silicon is possible. The results of the

2 The cryostat (cooling for the superconducting solenoid) has an inner radius of 115cm. Compare with fig. 2.1 and the 20m radius of the complete detector.

experiments so far are promising, but the technique falls in the latter category for the time being[1].

This thesis will focus on the use of *silicon microstrips* as transducers. I will discuss silicon microstrips and their transducer properties more thoroughly in the next chapter. Here I will focus on their application in the Inner Detector. In this thesis I differ purposely between *detector* and *transducer* where that is important, although both are usually referred to as detectors. With detector I refer more to the system as a whole, with cooling pipes, wires and mechanical support. Transducers refers to the material used to produce a signal from a passing particle, or the sub-unit which produces such a signal. Of course, the distinction is easy to make in the case of silicon microstrips, because as I will show, the transducer is the silicon material itself. Things look admittingly a little more complicated in the case of for instance a MWPC, a Multi Wire Proportional Chamber.

Originally, it was planned to use a *Micro-Strip Gas Chamber* (*MSGC*) in the endcaps of the Inner Detector. This was later abandon. Two different technology solutions remain in use for the Inner Detector: Silicon and Straw Chambers. The *Transition Radiation Tracker* (*TRT*) Straw Cambers occupies the outer layers of the Inner Detector, as figure 2.2 shows. The last layer of silicon will be at 52 cm and the TRT layers start there. The TRT detector plays a particular important role in pattern recognition due to its many layers, 40–60 depending on the track direction.

2.8: Silicon Microstrip Detector System

Silicon microstrips, along with silicon pixels are what will be used as transducers in the finished ATLAS silicon detector system. Silicon meets all of the requirements stated above. Silicon is a relatively low-cost material, can be made quite radiation-hard and silicon microstrips transducers have excellent resolution, both in time and space. The technology is well known through use at other experiments, including the LEP DELPHI detector system. What is planned for ATLAS is to build on that knowledge to refine the properties of the silicon microstrip detectors.

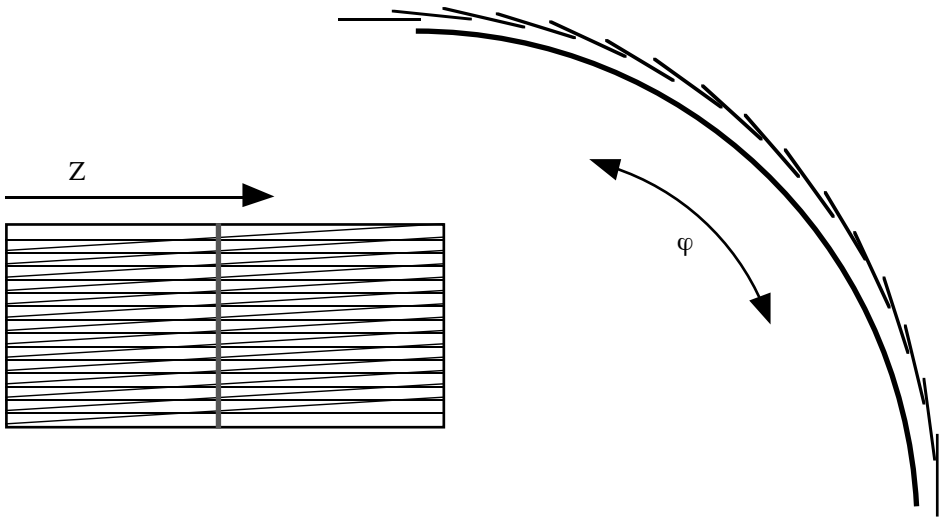


Figure 2.3: Silicon transducer wafer and Silicon Detector schematic cut-through (barrel).

Silicon microstrips transducers are manufactured from wafers of silicon. The size of the modules that will be used in the finished detectors are 6·12 cm. However, the manufacturing technology is restricted to 6·6 cm, thus two wafers are connected (*bonded*) together to form a module, as indicated in figure 2.3. The wafer has been manufactured as an n-type semiconductor. Strips of n^+-type are doped into this wafer, effectively forming junctions[1]. As I will discuss further in the next chapter, these junctions are the detecting elements.

Strips are doped into both sides of the wafer to obtain transducers which can yield φ / z (ref. figure 2.1) read-out data in the barrel. The obvious way of obtaining such readout capability would be to twist the strips on the one side 90° with respect to the strips on the other side. Calculations have however shown that only a small angle is necessary to obtain sufficiently precise z data, approximately 40 mrad is enough. This is indicated in figure 2.3. The figure only illustrates the concept, the different elements are not necessarily to scale with each other.

That the strips on both sides are almost parallel to the long edge of the wafer is an advantage. Having them twisted 90 degrees with respect to each other would require twice as many strips on one side. That would in turn have made it necessary with more electronics on one side, increasing both the cost and the power-consumption. The longer the wafer, the less electronics is needed per unit of transducer area. The length of the wafers is restricted by strip capacitance and strip resistance. Both factors will increase the noise and change the signal shape. Why the last is important I will show in chapter 4.

The wafers, with the necessary read-out electronics are simply referred to as *modules*. The modules will be mounted in the barrel of the Inner Detector in a way shown to the right in figure 2.3. The modules overlap slightly, and they have their long edges

parallel to the beam axis. As figure 2.3 reveals, there is a small area near the long edges of the wafers where precise φ / z data cannot be extracted. The overlapping is the remedy. And whereas the overlapping slightly increases the amount of material necessary for the detector, the effect is very small.

In the detector endcaps, similar silicon modules will be employed, mounted to form wheels or discs of the different radii necessary, as shown in figure 2.2. The modules will need to overlap in a similar way as they do in the barrel.

As of today, the placement of the on-module electronics is not yet fixed, although only a few alternative options are still being considered. I will not dwell on the many aspects of that decision, but apart from concerns about the mechanical support structure, the detector cooling system also enters as a parameter. The module read-out electronics, as well as the transducers themselves, need cooling. This will be supplied by a suitable liquid flowing through pipes over the read-out chips. Preferably, the chips should be placed in such a way as to make the pipes as short as possible, without too many bends.

Silicon microstrips have yet another advantage: They occupy a relatively small space per. unit of read-out channels, i.e. the *granularity* is fine. That enables all the layers enclosing the silicon detector system to be made with a smaller radius than would else be possible, reducing costs further. However, the reduced radius has to be weighed up against the radiation hardness. The particle flux through an area of the silicon transducer is higher the closer the silicon is to the primary vertex (point of collision). Careful simulations using available parameters for radiation hard silicon transducers suggests a radius of the first silicon strip layer of 30 cm. There are two silicon layers inside this, at 11.5 and 14.5 cm, but these are made of silicon pixel transducers[13]. Additionally, three more silicon microstrip layers are foreseen in the Inner Detector barrel, at radii 37, 45 and 52 cm. For the endcaps the current plan is to use 4 pixel wheels and 9 silicon microstrip wheels[13]. The pixels will occupy the smallest radii, for the same reason as in the barrel.

The pixel transducers, which the name indicates, obtain precise φ / z coordinate information from silicon pixels. Each pixel is the start of a read-out channel, and yields an φ / z coordinate directly. The pixels are intrinsically more radiation hard than strips, due to their small size[1]. They also provide good space resolution, but are more expensive to manufacture and use, the latter being caused by the need for more on-module electronics to read out the higher number of channels. There will be approximately 110 million channels in the full ATLAS system, to add even more is not seen as a very tempting option. The use of silicon pixel detector systems is restricted to areas with very high particle flux.

For the remainder of this thesis, transducers refers to silicon microstrips, unless explicitly stated otherwise.

3: Read-Out Electronics

The main subject of this thesis is the electronics used to detect particles in the ATLAS silicon detector system, and how to do it in the best possible manner. In the first chapter I provided the thesis background, presenting CERN, LHC and the ATLAS detector. I showed how the ATLAS detector is built of different sub-detector layers. Finally I discussed one of those sub-detectors, the Inner Detector, more carefully.

This chapter gives an overview of the *Read-Out Electronics* (*ROE*) of the silicon detector. The part of the ROE which is onboard the detector is called the *Front-end Electronics*, or FE for short. Here it will refer to the part of the electronics that handles the signals, from the transducers to the off-detector buffers. The FE will be discussed more thoroughly in chapter 4. Notice that I include the transducers in the FE, although it is more usual to separate them from the electronics. I do however find it natural to include them as the first link in the read-out *chain*, and since the Silicon Transducers are semiconducter devices with biased diode-junctions, I find it quite natural to include them in the FE, as well.

The signals from all the different FE subsystems are gathered outside the detector for processing. The implementation of that processing, and therefore the interactions between the different layers of the ATLAS detector do only enter this thesis through the consequences that interaction has on the FE.

I will also discuss the first component in the FE chain in this chapter: The silicon microstrip particle transducers. Introduced in the previous chapter, I will here deal with them in the context of being transducers.

In chapter 2 I mentioned that an event is the collection of all the signals resulting from one particle-burst. In the finished ATLAS-detector an event will be the result of two bunches of particles (protons) colliding with each other in the centre of the detector . The collision is referred to as a *Bunch-Crossing*, a BC, and will occur every 25 ns in LHC. In a *testbeam* situation an event might be a burst of primary particles traversing the detectors every 4 seconds, i.e. particles which are fetched more or less directly from the accelerator-beam. Chapter 5 will deal with a specific testbeam and the electronics used for it. In this chapter I will look more at the ROE of the ATLAS detector as it is planned to function in the finished machine.

3.1: Read-Out Electronics Building Blocks

The physical ATLAS detector, with all its layers, was described in chapter 1. For this thesis it is more interesting and appropriate to describe the machine in terms of the *read-out chain*, i.e. the path the signals follow from the detector transducer-material to they are stored for off-line analysis. In this thesis, the emphasis is on the on-detector read-out chain of the silicon detector, but a schematic overview will look quite the same for any detector system.

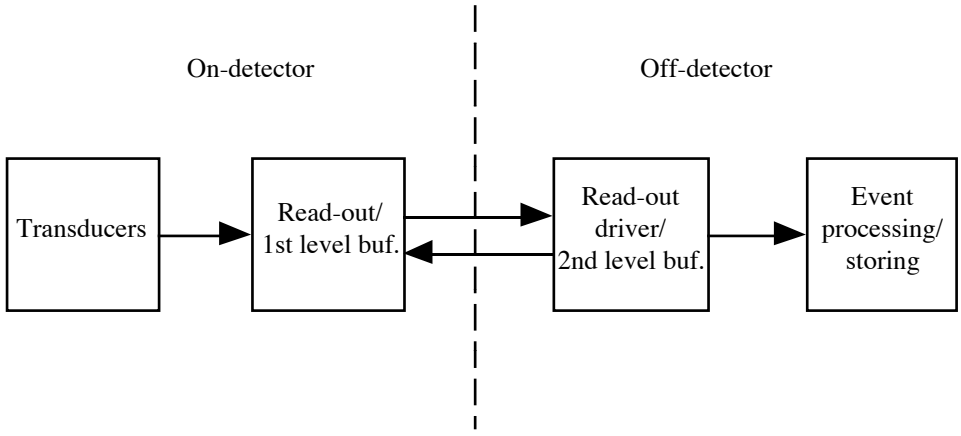


Figure 3.1: Read-out chain overview.

The chain is divided into a on-detector part and an off-detector part. The on-detector part includes the transducers for obvious reasons, as well as the FE. An alternative distinction between the FE and the rest of the ROE is that the FE handles *local* signals, i.e. signals from one wafer of silicon in the case of the silicon detector. The off-detector system handles and processes signals from several groups of transducers and from several detector layers. The 1st and 2nd level buffers refer to the ATLAS trigger system, which I will discuss in section 3.3.

3.2: Demands and Constraints

The overall demand on the read-out chain is that it must be able to retrieve the signal from a particle and send it to off-detector storage with as little signal degradation as possible. Another overall consideration would be to make the chain as efficient as possible, i.e. to specify every component of the readout chain to approximately the same level of quality. Overspecifying for example the speed of one component is pointless, as well as a waste of time, and most likely money. Likewise, underspecifying the speed of one of the components will at best create a bottleneck, lowering the capabilities of the whole readout chain.

The read out chain must obviously be fast enough to handle the high rate of particle signals. Since signals will be generated with as little as 25 ns between them, the ROE must be able to handle that without missing anything. To be able to handle small signals, the ROE must introduce as little noise as possible. And again, the less space the FE occupies, the better.

For the FE sub-system there are special considerations to be taken, given that the FE resides inside the detector. That the FE electronics is low power is a necessity. The options considered for designing the FE reflects that. For one thing, a choice have to be made as to which semiconductor technology to use. CMOS/FET has been the technology of choice for low power applications in the last decade, at least in the digital field of electronics. However, recent developments have been made with a mixture of CMOS and Bipolar technology produced on the same chip. An application of this technique in the silicon detector could be to make a transducer readout chip with a bipolar amplifier for the transducer signal and the rest of the chip in CMOS. CMOS is superb for digital electronics, but a bipolar amplifier has the ability to sink the input current generated in the silicon transducers. These days it can also be made low power. This has made it a interesting option for amplifying the transducer signal.

In addition, the signal-path technology has to be chosen. The options considered for ATLAS is either standard electrical, or a mixture of electrical and optical signal-paths. The optical paths would be used for transporting the signals off the detector and to feed the FE the necessary control signals. The advantage of optical paths includes that the signals travel with the speed of light, as opposed to approximately 60% of that in electrical paths. The higher speed could help reduce the problem of time-skew, i.e. that signals from different parts of the detector will take different amounts of time to reach the processing-units. To resolve which signals belong to which event is expected to be a significant problem. Even signals with the speed of light use 33 ns to travel 10 metres, well over the time between two ATLAS events. Furthermore, and even more important, is the fact that neither noise nor electromagnetic interference is introduced in an optical path. Disadvantages include the need for an extra stage, the electrical to optical converter, and the fact that the technology is not yet well explored in the speed-range and at the radiation level needed for the ATLAS system.

As before stated, it is important that the FE and the whole Inner Detector take up as little space as possible. When packing millions of silicon microstrips and chips so densely as is planned for the silicon detector, a problem arises: Heat. The electronics and the transducers dissipate power, i.e. heat, and that heat will have to be transported away by cooling-pipes containing a liquid, as mentioned in chapter 2. There is of course a limit to the effectiveness of the cooling system, and it would therefore help a lot if the electronics generated as little heat as possible in the first place. If the ATLAS FE is not designed with the necessary low power dissipation, the generated heat will become unmanageable. The

transducers generate heat because of their leakage current, and the leakage current is increasing with the operating temperature. This viscous circle may result in thermal breakdown if the heat is not transported away.

Detector-cooling is not a subject in this thesis. However, it seems the FE must be designed to meet two conflicting requirements: High speed and low power. Especially the transducer amplifiers introduce problems. The amplifier is the first stage after the microstrips, and amplifies the signal to a level suitable for further processing. If it were to have a response time in the same order as the signal from the microstrips, it would have to be high power. In CMOS, this problem is resolved using the technique of *Convolution/Deconvolution*. I will return in more detail to Convolution/Deconvolution in chapter 4, in which I will discuss the Felix chip. The Felix chip is a central part of the FE, and applies Convolution/Deconvolution.

It should be noted that faster bipolar amplifiers have been developed recently which can obtain the sufficient speed directly, without the need for a special technique like deconvolution. See section 4.2 for more information.

3.3: The ATLAS Trigger System

A central feature of the ATLAS detector is the three-level *trigger-system*. The importance of the trigger system alone makes it worth a discussion. It also directly affects most parts of the layout of the FE, so a knowledge of its main features is important in order to understand certain features of the FE.

The numbers presented in this sections must be regarded as preliminary and dates back to 1994–95. They might change slightly in the future, but the basic concepts and implications on the silicon FE are firmly established.

Bunches of particles will collide in the centre of ATLAS every 25 ns, generating a number of new particles. The particles thus produced will travel outwards in the detector and interact with the transducer material, producing signals in all the different layers of the detector. As much as 10^3 particles might be produced per event, and these will cause a burst of signals in the detector. Up to 7 Terabyte of raw data per second is foreseen. It is of course impossible to continuously store that amount of data for later analysis. The trigger-system works in real-time, and aims at filtering away as much as possible of the uninteresting events. The system is divided in three, with each stage working on a bigger sample of the data than did the previous. When an event arrives in the detector, a small sample of the data is extracted and processed in the first level trigger. The data is taken from evenly distributed locations in the Calorimeters and the Muon Detector. If the data looks interesting, the event is kept and a much bigger sample is sent to the slower second level trigger. If not, the event is thrashed.

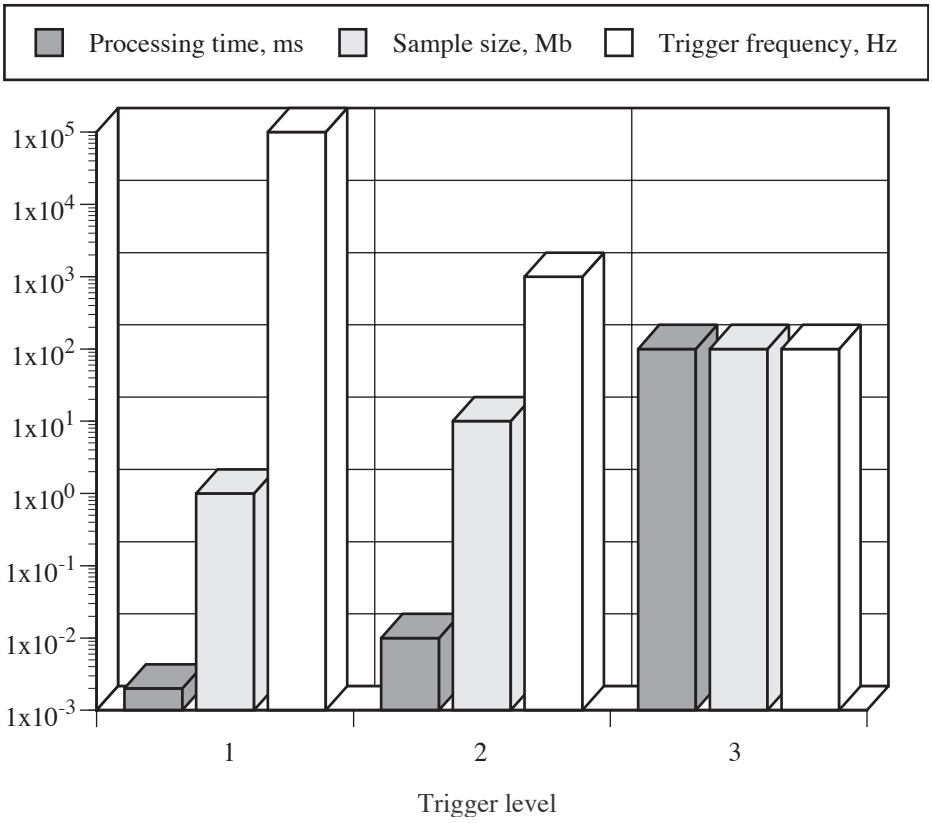


Figure 3.2: ATLAS trigger system principle.

The idea is that the 1st level trigger (LT) should be a fast, but rather inaccurate judge of whether the event is interesting or not. The decision algorithm should be simple, fast, and conservative: It should keep all interesting events, but will also keep many events that will turn out to be uninteresting. The 1st level trigger also tries to identify so-called *Regions of Interests* (RoIs) in the events it keeps[4]. This might for example be parts of the detector hit by a hadronic shower. The system transfers all the events that the 1st level trigger accepts to the 2nd level trigger, together with information about RoIs. The 2nd level trigger extracts a bigger sample of data from the event to work on, and takes the data mostly from the RoIs[4]. It is thus more accurate, but needs more processing time to arrive at a decision. The time it needs is available because the 1st level trigger has filtered away over 99% of the events. The same is repeated with the 3rd level trigger, except that the 3rd level trigger works on the full event, so it needs both much processing time and processing power.

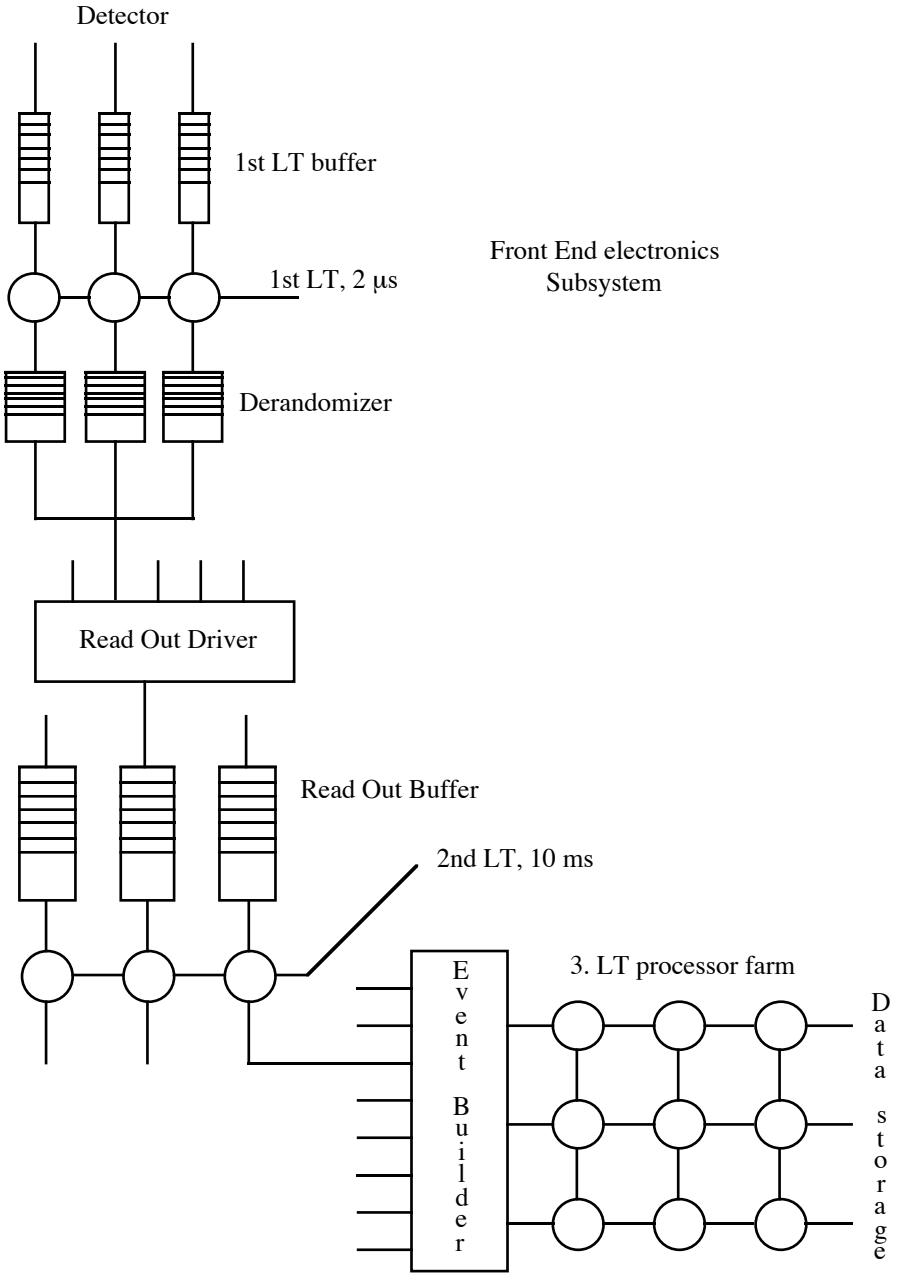


Figure 3.3: Schematic overview of the ATLAS trigger-system[4].

The Front End part of the system is indicated in the figure. The data flows downwards in the figure, and is built up into an event on its way down.

First data is temporarily stored in the 1st LT buffers while waiting for a trigger decision. Data from events that is not thrashed is passed on to the derandomizers, a buffer necessary to guarantee a maximum instantaneous 1st LT rate[4]. The value of the rate is fixed in ROE specifications to 100 kHz. The Read Out Driver collects the data from the derandomizers and puts together pieces of an event. These are sent to the Read Out buffers, where they are available to the 2nd LT and are being stored while waiting for an 2nd LT decision. The data, now assembled into event pieces, which passes the 2nd LT

are all passed into the Event Builder, which assembles the full event before making it available to the 3rd LT processor farm.

The following lists some of the main parameters for the trigger system:

- BCO: Every 25 ns
- BCO frequency: 40.08 MHz [4] (usually rounded to 40 MHz in this text)
- 1st level trigger delay: 2 μs
- 1st level trigger maximum average frequency: 100 kHz[4]
- 2nd level trigger average delay: 10 μs
- 2nd level trigger frequency: 1 kHz on average.
- 3rd level trigger frequency: $10–10^2$ Hz

There are several other parameters involved, but they are not relevant for this discussion.

Signals in many of the layers are used for the first level trigger, but data from the Inner Detector does not contribute because it is possible to make a sensible decision without it. Therefore it is sufficient for the FE of the Inner detector to simply store the data for 2 μs until a decision has been reached. If the event is trashed all the samples in all cells in all the buffers associated with the event are flushed.

When being filtered through the trigger system, the data stream from the detector is reduced by a factor of 40 k. The resulting ~100 Mb/sec on average is possible to store offline for later analysis.

Finally it must be mentioned that for this system to work, a few corners have to be cut. I said earlier that the ROE must be designed so as to not miss any events. The truth is a bit more complicated. Simulations of LHC events have provided a measure of the average rate of interesting events, and the trigger system has been designed to work well for that *average* rate, give or take a little. However, there is a limit to how many *consecutive* interesting events the system can handle. This is due to buffer-size limits and available processing power. For instance, the 1st LT is designed so that two consecutive triggered (accepted) events will be separated by at least two untriggered ones, and that no more than 16 triggers will occur in any given 16 μs period[4]. This ensures that the rest of the ROE can handle the data stream, but it also means that loss of a few interesting events will be unavoidable.

There is also a built-in limitation in the signal handling capabilities of the FE. This means a loss of a few hits in some silicon wafers. I will explain why this is so in the next chapter. While this may result in some ‘holes’ in a few single-track reconstructions, the effect is too small to have any significant impact on the quality or efficiency of the event reconstructions.

3.4: Silicon Microstrip Transducers

In chapter 2 I introduced the silicon microstrip transducers and explained how they would be applied in the ATLAS detector. Silicon is a semiconductor, and as explained below a well suited transducer for particle-detection.

3.4.1: Basic Silicon Microstrip Transducer

To make a particle detection transducer from silicon, one starts with a silicon wafer manufactured as a lightly doped n-type (electron majority carriers). Into this wafer is doped strips of p-type by diffusing for instance Boron into the surface of the wafer[1]. This makes up pn-junctions. As will be known, a depletion region forms over a pn-junction because of electrons diffusing over to the p-side and holes diffusing over to the n-side. This creates a region of net charges of opposite signs on both sides of the junction and the consequent build-up of an electric field. The diffusion current stops when in equilibrium with this electric field.

Applying an external voltage over the junction biases it. Forward biasing (V+ connected to p implant) makes the region of depletion narrower, whereas reverse biasing makes it wider. The higher the reverse biasing voltage, the wider the depletion region will be, up till the breakdown voltage limit. Breakdown occurs when e-h pairs generated in the depletion region by thermal excitation acquires so much momentum by the strong electric field that they ionize other e-h pairs, thus causing an avalanche. e-h pairs are of course always generated in the depletion region by thermal excitation, regardless of the applied external bias voltage. This will always cause a small amount of leakage current to flow in the junction, depending on the temperature.

The junctions of the silicon microstrip transducers are reverse-biased with a high voltage in the range of 50–150 V[1]. This is sufficient to completely deplete the entire n-type region. Breakdown does not occur because the n-type wafer is lightly doped, and therefore has high resistivity. The below figure shows the electric field that will result because of the bias voltage.

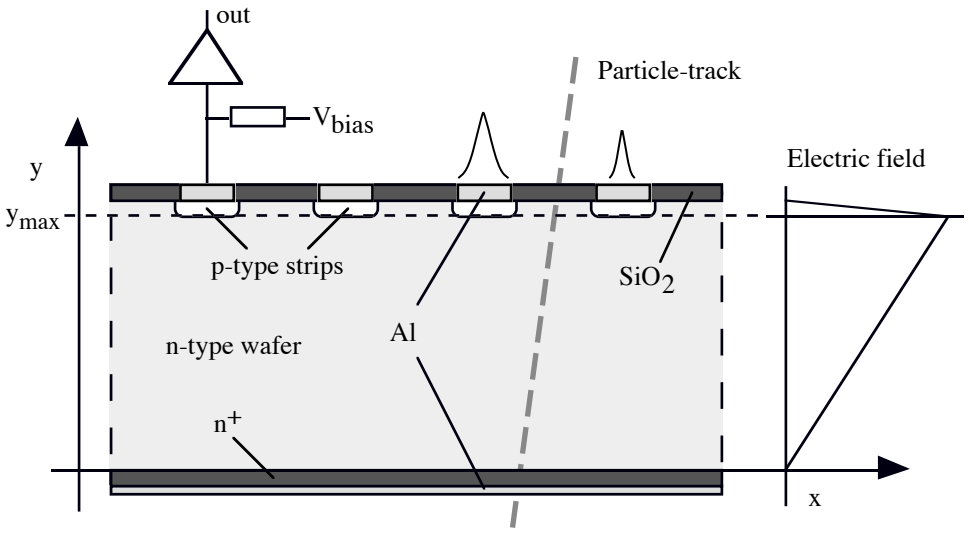


Figure 3.4: Schematic diagram illustrating the principle of particle-detection with a silicon micro strip transducer.

No attempt is made in the figure to visualise the drift and diffusion of charge. The waveforms shown over two of the strips are merely meant to indicate the resulting pulse and distribution. The other elements present are the Al contacts/protection over the strips and on the backplane, the n^+ layer functioning as an ohmic contact, and the protective SiO_2 oxide layer between the strips[1].

When the entire semiconductor is depleted, the resulting space-charge density function $\rho_v(y)$ will be constant in both the n-region (positive) and in the p-region (negative). This is neglecting fringe-effects. Since the electric field E(y) is[6]

$$E(y) = \int_{y_0}^{y} \frac{\rho_v(y')}{\varepsilon} dy', \qquad \text{Eq. 3.1}$$

it follows that E(y) is linearly rising with y in the n-region and linearly falling in the p-region, as figure 2.4 shows. Outside the depletion region $\rho_v(y)$ is zero everywhere, there is no voltage gradient and consequently no electric field. Furthermore, the constant value ρ_v in the n-type bulk must be equal to qn, where q is the charge of the donor ions and n is the donor concentration. The dielectric constant ε is 11.7 As/V/m for silicon. Thus the conclusion is that the electric field inside the n-type bulk can be expressed as

$$E(y) = \frac{qn}{11.3} y, \qquad \text{Eq. 3.2}$$

with $y = 0$ at the n^+ contact and y_{max} at the pn junction border, as shown in figure 2.4.

When a particle of energy at or above the minimum ionization energy (3.7 eV) traverses such a silicon transducer, electron-hole pairs will form along its trajectory. The band-gap in Si is only 1.1 eV, the additional energy is lost to mechanical deformations etc. in the silicon. Because of the electric field imposing a force $\vec{F} = q\vec{E}$ on the e-h pairs,

electrons will drift towards the n^+ contact and the holes will drift towards the p-n junctions, where they will create a very short current-pulse. This current-pulse is the transducer signal. A particle traversing such a 300 μm thick silicon transducer, will loose energy, with $\Delta E = 84 keV$ (given by the Bethe-Block formula)[1].We have seen that an average of 3.7 eV is lost for every e-h pair that is produced. The signal charge that will form is therefore $84000 / 3.7 \approx 22000$ electrons. 22000 electrons is usually called 1 MIP when discussing silicon transducers of 300 μm thickness, although MIP actually refers to the Bethe-Block distribution.

The electrons and holes will also *diffuse*, because the charge forms around the particle-track. This creates a non-uniform charge-distribution and as a result the electrons and holes will diffuse in the direction of less charge density, i.e. outwards, approximately parallel to the wafer-plane. Therefore the current pulse resulting from a particle passing through the transducer might spread out over several strips. This is what is indicated in the figure. Typical diffusions in a 300 μm thick silicon transducer is 10–30 μm.

3.4.2: Refinement Options

What has been described above is only one of the ways to make a silicon particle detector; the technique discussed in section 3.4.1 can be regarded as a baseline for producing such detectors. Several refinements are possible, and some are even necessary to make reliable transducers for the ATLAS silicon detector system.

As before stated, the Inner detector is exposed to a massive particle flux, i.e. radiation, under operation. This unavoidably causes radiation damage in the silicon bulk. The effects of radiation damage in silicon transducers include increased leakage current, increased transducer capacitance, increased full depletion voltage, and type inversion.

For the increased capacitance and depletion voltage there are no real remedies, although methods can be used to dampen the effects somewhat. The two phenomena will however inevitably slowly deteriorate the effectiveness of the detector and the quality of the signals. The ATLAS specification aims at a lifetime of at least 10 years.

The increased leakage current is due to the radiation causing irregularities to develop in the silicon lattice, such as displaced atoms. This may be described as ‘band-gap dirt’, and the result is that less energy is needed to create an e-h pair near such ‘dirt’. The leakage current may eventually become so high that it will saturate the read-out amplifier, leaving no room for a signal to be amplified. Getting completely rid of this problem is difficult, but using an amplifier which could sink a good deal of input current would certainly help. This is part of why bipolar amplifiers are currently being looked into as candidates for the microstrip read-out task.

Type inversion is a phenomena which is not yet well understood, but its effect is that the n type silicon in the bulk of the transducer transforms to p type under continued

radiation[1]. In our basic transducer, this would lead to p type strips in p type bulk, which obviously is rather useless. The remedy which is planned for ATLAS is to use n^+ type strips in the n type bulk, with the strips surrounded or separated by a layer of p implant[1].

Yet other schemes are possible. In chapter 1 I mentioned pixel transducer. I also showed that the silicon transducer to be used in the ATLAS detector will be double-sided. In that case strips are doped into the n^+ implant of the basic transducer in figure 3.4 also, forming additional junctions on that side as well. The drifting electrons that are created by a passing particle are responsible for inducing the current pulse at those junctions.

A drawback of such double-sided microstrip detectors is that ambiguities in determining the position of a passing particle are unavoidable in the case of a double hit in the same wafer. The effect can however be somewhat reduced by correlating pulseheights if the strips are read out using an analog scheme[1].

3.4.3: Analog vs Binary Readout

Analog readout means that pulseheight information from each strip is preserved through the readout chain. Another alternative is binary readout, in which only a single bit from each microstrip, 'hit yes' or 'hit no', is transferred over the readout chain. Each solution has its advantages and disadvantages. The position ambiguities mentioned in the previous section is obviously in favour of analog readout. Space resolution can also easily be made better with analog readout. This is because the pulseheight information can be used to calculate the inter-strip position at which the particle passed. The position x of a particle-hit can then be calculated as:

$$x = \frac{\sum_n x_{strip}(n) \cdot PH(n)}{\sum_n PH(n)}, \qquad \text{Eq. 3.3}$$

with PH(n) being the pulseheight of strip n. In real life, a more complicated algorithm using cluster-cutting is used to obtain an even better resolution[1].

In fact, with analog readout of the strips, it is not even necessary to connect each strip to a read-out channel. For each strip that are read out, there can be one or more in between that are floating. In such a *capacitive charge coupling* scheme, the floating strips are connected to the bias, but not to the read out electronics, and all the strips are connected to the p implant over a capacitor, called C_c in the below figure. The figure shows the basic model used for calculating properties of a charge coupling silicon transducer.

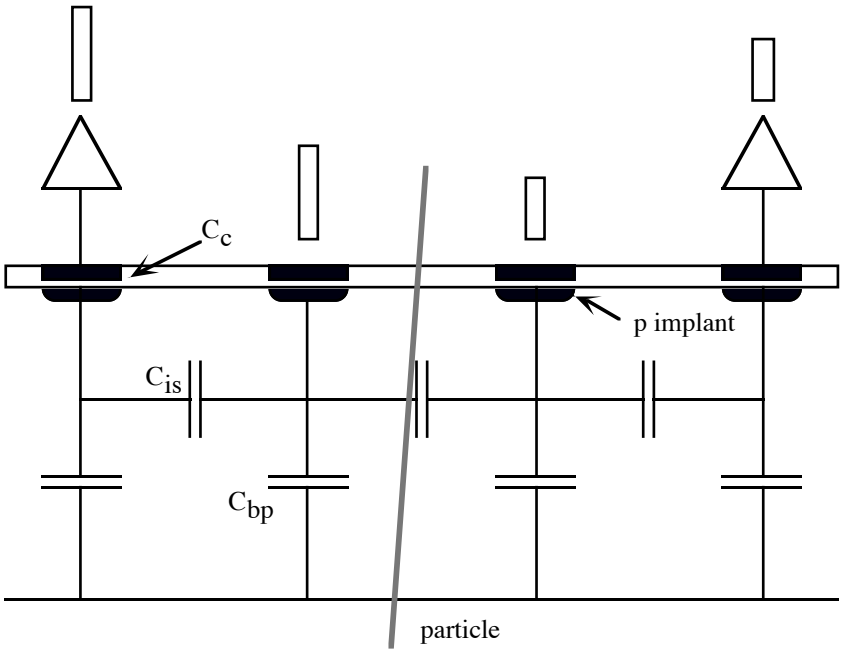


Figure 3.5: Principle of readout with capacitive charge coupling, indicating pulseheight distribution[1].

If a particle passes through the transducer in a way shown in the figure, the charge building up on the two nearest p implant strips will couple to the next strips because of the silicon interstrip capacitance C_{is}. These charges will again couple to the read out electronics through C_c. The model of the coupling is then used to calculate the position of the passing particle. Some charge is lost (<10%) due to the capacitive coupling to the backplane, C_{bp}, but very good resolution can still be achieved, in the order of $strip\_pitch / \sqrt{50}$ [1]. Another obvious advantage is the need for fewer read-out channels. Capacitive charge coupling is all in all a very attractive scheme.

The capacitive coupling read-out scheme is only possible if analog readout is used. Binary read-out and binary electronics, on the other hand, have a number of advantages, including less power consumption, ease of manufacturing and design. In a binary read-out scheme, a digital level 1 buffer storage system can be used, something which relaxes the space-requirements, as well. Furthermore, no potentially complicated capacitive model for the silicon transducers is needed, because in the binary scheme all the strips will have to be read out. This gives an resolution of $strip\_pitch / \sqrt{12}$ from the rms resolution-definition

$$\sigma^2 = \int f(x)(x - \bar{x})^2 dx, \qquad \text{Eq. 3.4}$$

since, for binary readout, $f(x) \equiv 1$ for $|strip\_pitch / 2| < 0.5$ and $f(x) \equiv 0$ elsewhere. This is clearly worse than for analog readout, but the other arguments do weigh heavily in favour for the binary read-out system because they all relax the demands on other

components of the detector system. And there is of course a question of what resolution is actually needed. Overspecifying the resolution is only a waste of time and money.

3.5: Read-Out Summary

I have in this chapter discussed some of the features of the detector read-out chain, with the emphasis on the Inner Detector and its silicon microstrip detectors. As can be seen, there are many considerations involved, several of which are conflicting with each other. Even so, there are still many more considerations to be taken, far more than can be mentioned here. Yet other schemes for silicon detectors are possible, both for readout and choice of semiconductor material, implant material and production technique.

I will for the rest of this thesis be content in discussing the analog scheme, used with silicon microstrip transducers. This does not necessarily reflect what will be used in the final detector, only what was used at the time I worked with it. At that time, the best resolution achieved was in the order of 5 μm, using capacitive charge coupled silicon strip transducers with a read-out pitch of 100 μm and a strip pitch of 50 μm (every other strip read out).

4: The Felix Chip

I ended the previous chapter with discussing the first element of the readout chain, the particle signal transducers. On-module electronics was mentioned, and this chapter discusses a central element of that electronics, which also is the next element in the readout chain: The Felix chip. Although Felix is only one of the chips considered for the read-out, I will for the sake of clarity write as if this chip is the final development of the Front End Electronics.

The Felix chip is an essential part of the FE. The chip has been developed and still is under development at the University of Oslo, where this master-study is based. Thus the testing of the Felix has been a central element (as viewed from my position) of the activities in the Oslo-group. This is also the reason why I discuss the Felix chip instead of another of the possible candidates for doing a similar job.

4.1: Felix Essentials

The Felix chip incorporates many of the functions discussed in previous chapters. Its analog inputs are connected directly to the silicon microstrips. It contains a pre-amplifier and electronics to sample and hold data.

Generally, any readout chip designed to be connected to the silicon microstrips of the Inner Detector must meet the following criteria:

- Time resolution of 25 ns or better to separate hits close in time in the same siliconstrip.
- Radiation-hard.
- High signal to noise.
- Low power.
- Must be able to store data and release them on demand.

Furthermore, such a readout chip has to be working extremely well from the moment the detector is put into operation. The modules will be densely packed in the centre of the ATLAS detector and connected to the mechanical support structure as well as to the cooling system. To exchange the modules and chips once the detector is assembled is going to be practically impossible.

4.2: Convolution & Deconvolution

The detector specifications set a strict upper limit on the power dissipation of the different components. The maximum is 4mW per. readout channel in the Silicon Detector system. The limit is governed by the space available for cooling pipes and the effectiveness of the cooling system. This poses a problem for any readout chip: The first stage in such a chip will necessarily have to be some sort of charge sensitive amplifier, to convert the strip charge into a voltage suitable to be processed by the rest of the electronics. As will be known, the power dissipation of an amplifier increases with shorter response time. Response time are also referred to as *rising time* or *peaking time*, and *slew rate* refers to the response speed. To make a low-power amplifier it is made slow.

To use a slow silicon transducer amplifier is fine as long as possible signals are separated by sufficient time. In such a scheme, a very short signal is generated in the transducer from a passing particle. The amplifier responds slowly to the signal, and a sample from the amplifier is taken when its output is highest, giving a measure of the strength of the transducer signal. Such a sample is called a *peak sample*. The amplifier then settles again before the next possible particle hit. This gives a readout scheme as in figure 4.1, where a sample is taken from the output pulse when it is at its maximum. The sampling is typically triggered by a signal from the same particle in an external scintillator, and since the response time of the amplifier is know, the correct amplifier sampling-time can be calculated.

For example, a well-designed amplifier with a rise-time of 1 μs will perform well under the required maximum ATLAS Silicon Detector power dissipation per channel, and such readout chips are being successfully operated in a number of detectors today. The problem faced in LHC and therefore ATLAS is of course that events are only separated by 25 ns, as dictated by the need for high luminosity.

The technique Felix uses to achieve both low power and sufficient time-resolution is called convolution/deconvolution. Sufficient time resolution here means that the chip must be able to resolve hits separated by two events, i.e. separated by 50 ns or more, as demanded in the ATLAS specifications. Therefore, it is not necessary that the chip can resolve two hits being directly consecutive, i.e. separated with 25 ns. This is the FE limitation mentioned at the end of section 3.3, and the arguments are as follows: The probability that two hits occur in the same strip with 25 ns separation is very small. The probability increases fast with hits separated by 50 ns, 75 ns and longer. To resolve hits separated with 25 ns means doubling the resolution from 50 ns, and will mean higher demands on technology and higher power dissipation. Since the transducer pulse has peaking time of approximately 25 ns, two hits separated with 25 ns would produce two pulses that would be mixed into each other from the beginning, making a separation very hard. All in all, a lot of effort only to gain a small fraction in efficiency. It was therefore

decided to relax the demand on time resolution slightly. The demand is now that the system should be able to resolve two pulses that are separated by 50 ns, i.e. one bunch-crossing.

For the sake of simplicity, I will in the remainder of this section refer to hits in the same strip, separated with 50 ns, as consecutive.

The pulse coming from the transducer resembles a delta-pulse: Very sharply peaked and very short, with a peaking time of less than 25 ns (ref. figure 4.4). The pre-amp, however, is made quite slow to meet the requirement of low power, and will react in a way shown in the following figure to this pulse, typical of a slow amplifier.

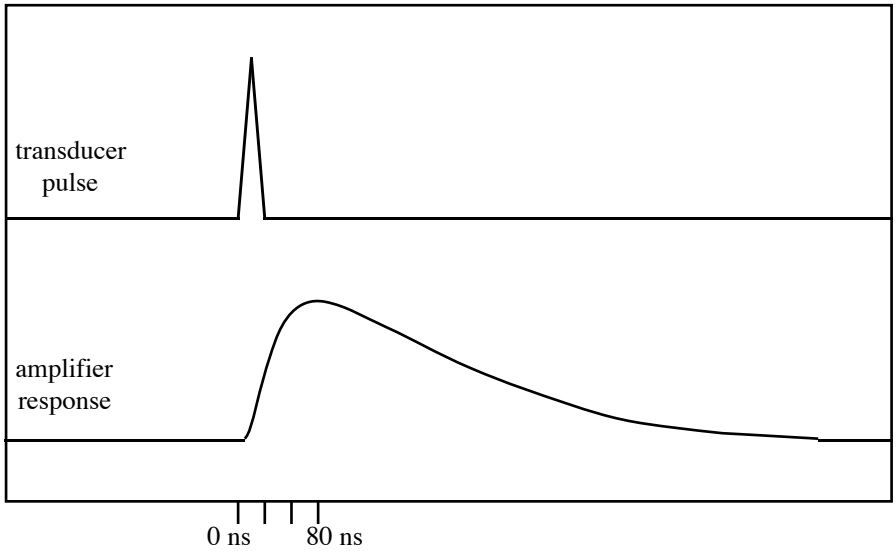


Figure 4.1: Transducer output-pulse and corresponding pre-amplifier output-pulse.

The amplifier has a rising-time in the order of 75 ns, and in effect convolutes the transducer pulse. The information about the input-pulse is not lost, though: The amplifier has a known transfer function, determined at design time, and will output one unique pulse for every unique input-pulse. The input-pulse is of course not really a delta pulse, but its shape is also known. The only unknown is then the amplitude of the input pulse. It can be shown that it is sufficient to extract three samples from the output-pulse to uniquely determine its shape, as long as the transfer function is of the special CR-RC type. When the shape of the output-pulse is know, the input-pulse can be recovered by applying the inverse transfer-function to the output. In particular, when it is known at exactly which time the three samples where taken, relative to the input-pulse, *and* the transfer-function is of the CR-RC type, as is the case for the Felix preamp, it is enough to apply a weighed sum to the three samples taken to recover the input-pulse with sufficient accuracy, thereby determining its amplitude. This procedure is the deconvolution.

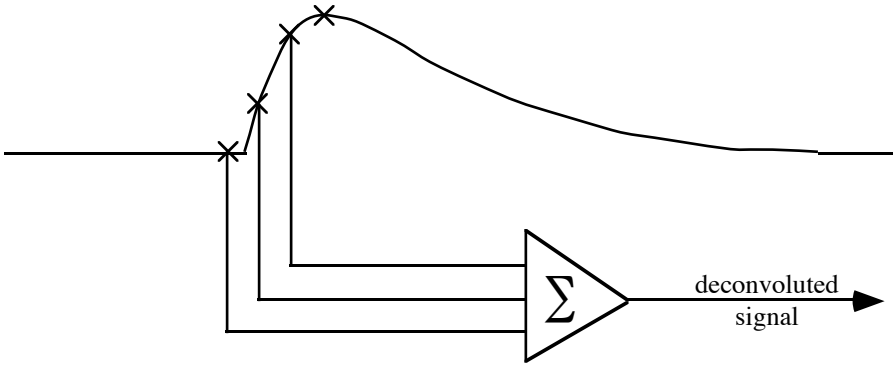


Figure 4.2: Schematically showing the technique of sampling the pre-amp output to recover the original transducer-pulse.

The figure indicates that four samples are taken, although only three of them are used in the weighed sum. The fourth sample is taken at the moment when the output is at its maximum, and is what I have referred to as a peak sample. It provides a good indication of the strength of the input-signal without requiring any processing, in a similar way as have been discussed for slow amplifying chips. The peak sample is frequently used in the various stages of testing because of this. The limitation of using the peak sample only is that the amplifier output-shape will look quite the same for one hit as for two consecutive hits in the same strip, and that the occupancy of the detector will become problematic.

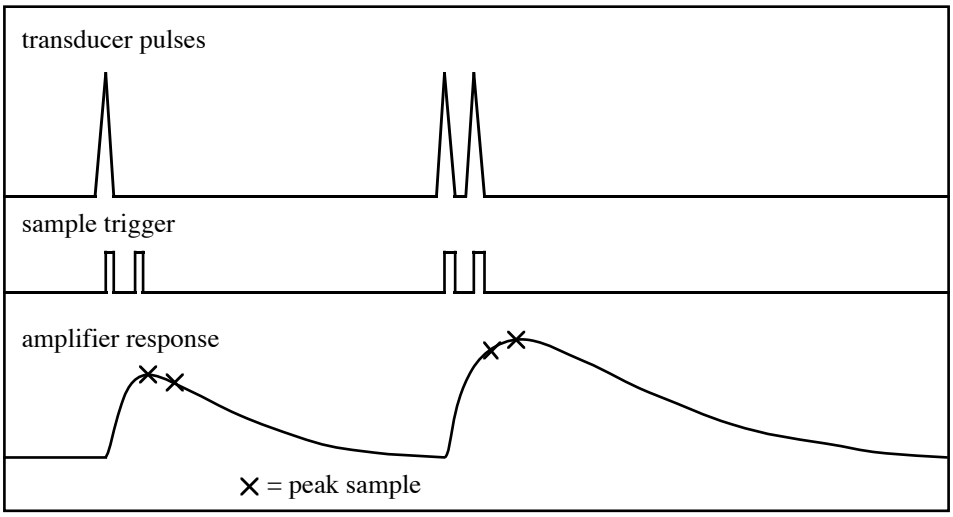


Figure 4.3: Amplifier response of one hit, and of two consecutive hits in the same microstrip.

The amplitude will be greater for two consecutive hits, but the peak samples cannot be used for resolving two such hits. This is seen clearly in figure 4.3. In the case where there are two consecutive hits, the two peak samples does not necessarily reflect the strength of the two signals, because the slow amplifier mixes the response into a single pulse, and both peak samples will be taken inside that pulse. The sampling is triggered

externally, so there is a good chance that peak samples will be taken consecutively now and then, mostly when there are no consecutive hits in a given microstrip. This is illustrated in the left situation of figure 4.3. Given the closeness of the two peak sample values, and the uncertainties that always exist due to noise, it is not possible to say whether the samples result from two consecutive transducer signals, or just from one strong transducer signal.

Enter deconvolution. The deconvolution algorithm, when done correctly, has exactly the ability to resolve whether an amplifier output pulse results from one or two transducer pulses. To sum it all up, the peak sample has an insufficient time resolution, whereas the Felix chip can achieve sufficient time resolution through deconvolution.

The deconvolution done in the Felix chip does not attempt to produce an exact replica of the transducer input pulse, since that is not necessary anyway. It sufficient that a) it is determined in which event the signal occurred, b) that the deconvolution is able to rebuild two or more consecutive hits, and c) that the deconvolution result contains precise information about the pulse amplitudes.

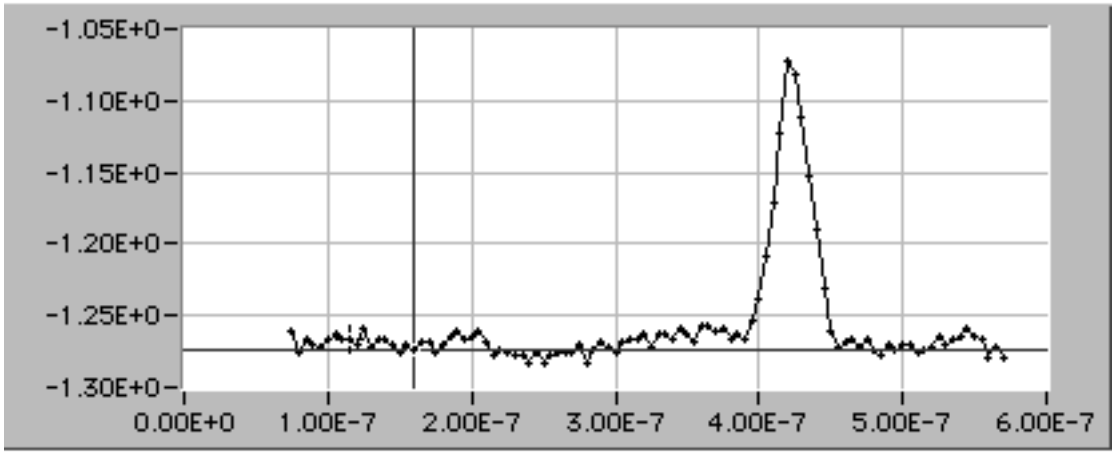


Figure 4.4: Felix deconvoluted output pulse[12].

As can be seen in the above figure, Felix at least meet the demand a). It is clocked at 40 MHz; it has a time resolution of 25 ns, the exact time between two bunch-crossings. The deconvoluting algorithm also works with the same time resolution, which means it is able to produce exactly one output voltage per 25 ns. A Felix output sequence with deconvolution performed on two consecutive hits will therefore look something like this:

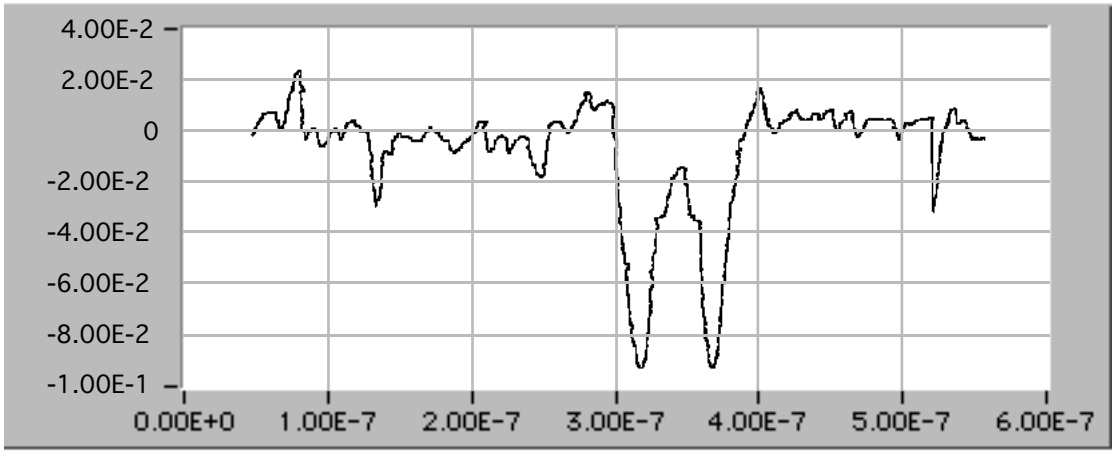


Figure 4.5: Felix deconvoluted output pulse for two consecutive hits[8][3].

This demonstrates that Felix also meets demand b). That it meets demand c) can not be seen from the figures, because the input pulses used are test-pulses from a pulse generator with equal amplitude. By using pulses with different amplitudes from such a pulse generator, it is however not difficult to demonstrate that the deconvolution algorithm does indeed produce an output pulse whose amplitude is proportional to the amplitude of the input pulse.

As a final point about power savings it must be mentioned that recent developments have demonstrated that it is possible to produce radiation-hard bipolar amplifiers which have sufficient low power, low noise *and* a peaking time in the order of 25 ns. This allows for resolving consecutive hits without the need for extra processing, simply by using peak samples directly. This technology is still in its early stages, but the possible simplifications it would mean for the silicon microstrip read out electronics are so tempting that it is emerging as a strong candidate for being the solution which will end up in the final silicon detector system.

4.4: Felix Internal Structure

The Felix chip consists of several blocks that can be clearly separated. These blocks have typically been developed as separate chips by the different laboratories and university groups that participate in the collaboration. Then, when their characteristics met the basic requirements, the components were included in one chip, namely the Felix.

If we follow the signal through the Felix chip, blockwise and for each channel, we find the Slow-shaping Amplifier, an Analog Delay Buffer (ADB), an Analogue Pulse

3 That the wave-form appears to be upside-down is only a result of lab-software.

Shape Processor (APSP), and a Multiplexer to hold the data for all the channels, and multiplex the samples out on one output-line upon readout.

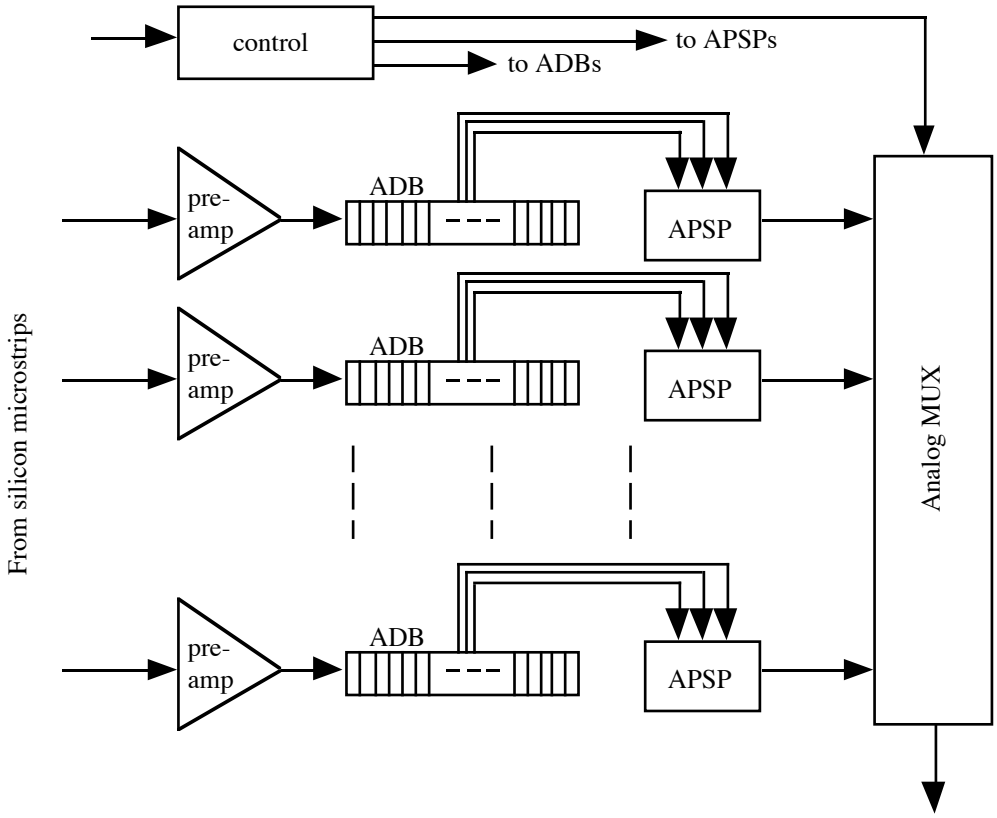


Figure 4.5: The different logic blocks of the Felix chip.

4.4.1: The Preamplifier

The preamp meets the specifications for speed as discussed under 4.2. The job of the pre-amp is to amplify the transducer-signal to a level suitable for further processing in the succeeding stages. The preamp consists of two stages: A fast charge sensitive amplifier and a CR-RC Shaper.

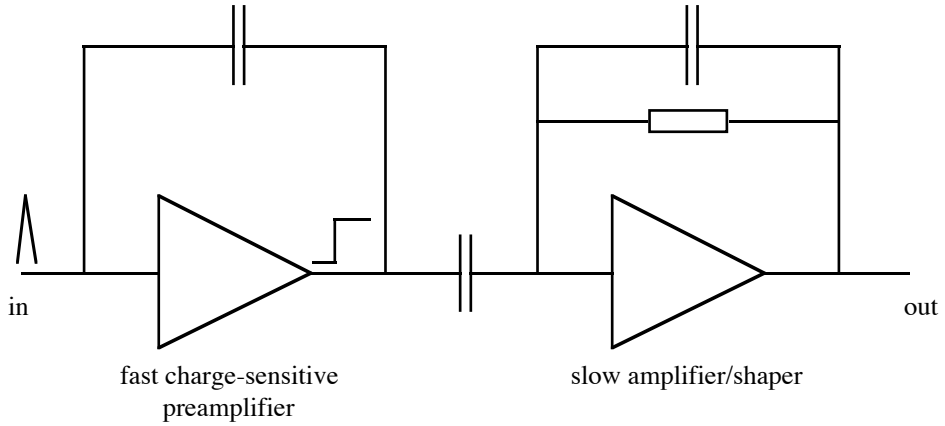


Figure 4.6: Simple diagram for the preamplifier.

The first stage is a fast charge-sensitive amplifier. It integrates the charge from the microstrips. It has a transconductance of 4.5 mA/V, equivalent to very little 'gain'. Even so, it needs a current of 700 μA (most recent, 128-channels version) for its operation. The shaper, on the other hand, has a gain of approximately 20, but consumes only 250 μA on average. This is the trick; the amplifier can be low power while still having the necessary gain due to a slow response-time. The original pulse is later recovered with deconvolution. The second stage, the shaper, is in effect a low-pass filter. This has the additional advantage of filtering away high-frequency noise.

4.4.2: The ADB

The Analogue Delay Buffer is a sample storage-unit. It makes it possible to hold the events the 2 μs needed for the level 1 trigger decision, while still sampling at 40 MHz. The trigger decision is based upon signals originating from other layers of the detector, and its logic implementation has been discussed in chapter 3. In a test-situation, however, the trigger will usually come from one or more simple scintillators placed in the beam line.

The ADB has 4 parallel channels, one for each of the samples that are taken when a signal-read is triggered. Each of the channels contain 84 quite simple storage-cells. 67 of the cells acts as preamp-buffer, and $4 \cdot 4$ cells are reserved for APSP input.

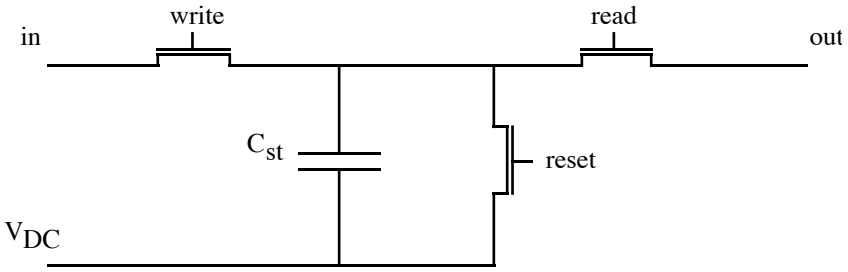


Figure 4.7: ADB storage-cell

Whenever a positive level 1 trigger is generated, three of the four samples belonging to the same event as the trigger are transmitted to the APSP for deconvoluting. The forth channel holds the peak samples. These four samples can be anywhere inside the channels. The ADB is being filled continuously at 40 MHz, and when the last four cells have been written to, the first are overwritten again. Since $25ns \cdot 67 = 1.675\mu s$, it follows that the ADB does not have enough space to wait for the 1. level trigger decision in its current version. The ADB buffer will in the final version of the Felix chip be extended to 100 cells per channel.

4.4.3: The APSP

The APSP is used to retrieve an amplified version of the transducer signal. This is done by performing a deconvolution on three of the four samples transmitted from the ADB, as discussed in 4.2 and illustrated in figure 4.2. The value of the weights used for the sum are hardwired into the APSP. In addition to recovering the transducer signal, the APSP algorithm amplifies the signal further, by a factor of 3.

Actually, the APSP performs a *Finite Impulse Response Filter* (*FIR*) algorithm, having an output defined by[7]:

$$Y_i \sum_{k=0}^{2} h_k x_{i+k}\,, \qquad \text{Eq. 4.1}$$

where h_k is the weights, x_i is the sample in the ith ADB cell ($0<i\leq 65$) and Y_i is an amplification factor.

4.4.4: The Multiplexer

The APSP outputs an analog signal, thus the multiplexer needs to be analog to preserve the pulseheight-information. The MUX has a gain of approximately 0.7.

Upon receiving a positive level 1 trigger, clocks and other control-signals are sent to the ADB, the APSP and the MUX. Three samples are fetched from each ADB and placed in the corresponding APSP. Each APSP does a deconvolution and places the result on the MUX inputs. The MUX has a sample&hold function, ensuring that none of the APSP output amplitudes are lost. Finally, the MUX clocks all its inputs to its output, completing the readout of all the Felix channels. The signals are thereafter transferred to delaying buffers (the derandomizers) while waiting for a level two trigger decision.

4.5: Felix Development

It has been mentioned that the Felix chip is only one of several chips under development for the same read-out task. Of these chips, the Felix is probably the one with the simplest design, at least of the analog chips. The idea behind the Felix-design was precisely to emphasize important characteristics like signal over noise while being conservative with 'luxury-features'.

Felix has undergone many test, and been upgraded many times. Of the more visible upgrades can be mentioned increases in the number of channels. The first version that incorporated all the blocks discussed in 3.4 on a single chip had 32 channels, the most recent has 128.

The increase in number of channels is certainly one of the easier upgrades. It is the optimisation along the signal-path that takes time and is difficult. The individual blocks (ADB, APSP...) have to have matching input- and output characteristics. And from section 4.2 it should be clear that matching the transducer signal, the shape of the preamp output-signal and the APSP weights are very important to get an efficient deconvolution.

Equally crucial are matching the FELIX internal characteristics to the outside world. While the internal blocks only connect to each other, seeing quite similar voltage levels, impedances and capacitances, the preamp connects to long silicon strips which will be biased to up to 200 V and the MUX will connect to external wires which might have a quite unfamiliar impedance. While RD20 itself is responsible for the silicon strip modules, the matching of the MUX to the outside world has to be done in collaboration with the RD group responsible for transporting the signals off the detector.

The most effort recently has however gone into matching the internal characteristics with the transducer signal. I will show what progress has been made in that area at the end of chapter 7.

5: ATLAS Testbeam

As with the Felix itself, all the other components of what will become the vertex-detector sub-system of the ATLAS detector need to be tested and tested again. Limited testing of small parts can be made in the laboratory. One such experiment will be described in chapter 6. For more extensive testing, however, a so-called *testbeam* is more appropriate. A testbeam facilitates testing the integration of a system on a scale not possible in the lab.

5.1: Overview

A testbeam differs from a laboratory experiment in many aspects. In a testbeam there is access to particles from an accelerator beam. This provides particles with a rate and momentum not accessible in the lab. Radiation danger dictates that the control-system be outside the beam-area. This means long wiring and thus long signal delays. Also, in a testbeam several groups that work on different parts of the detector can come together and integrate parts of their system. In all this respects, a testbeam resembles the situation in which the finished detector will operate in ATLAS much closer than does a laboratory experiment. This is what makes a testbeam so valuable.

Setting up and running a testbeam experiment is a major task. Success requires a vast number of components to be incorporated and integrated. This chapter will discuss a testbeam that was in operation in the summer of 1994. It was used for testing the current version of the Felix chip and module as well as for testing communication between the modules and the control-hardware and -software.

The setup used in that testbeam can as a first approach be divided in two: The beam-line area and the control room. The beam-line area is the area in and close to the particle-beam. All the modules, scintillators, power-supplies for both, and (adjustable) mechanical support for the modules stood in the beam-line area. All the equipment for controlling and read-out of the modules stood in the control-room. The equipment in the beam-area were connected to the equipment in the control room with an electrical cable system incorporating both busses and single-signal cables.

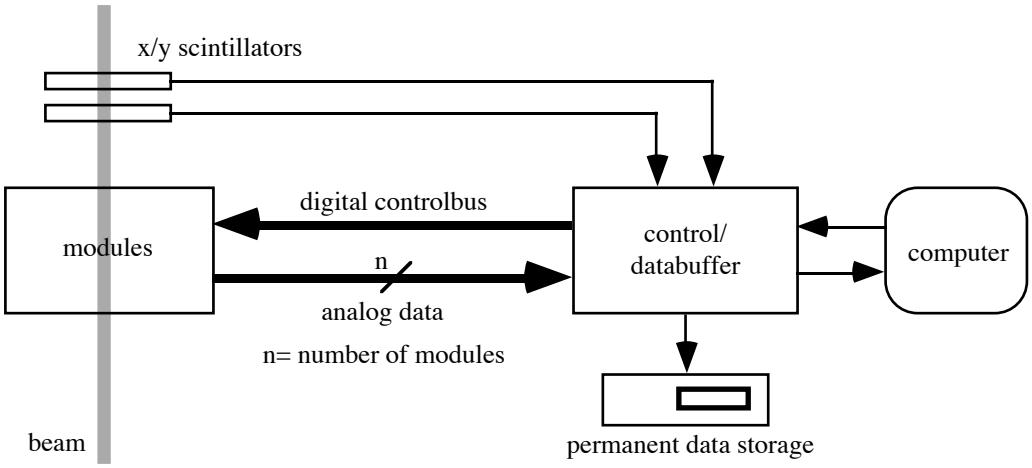


Figure 5.1: Schematic division of setup in beam-area and control-room area.

The physical setup and the division of the two areas were dictated by space-limitations and radiation-danger: The beam area was locked when the beam was switched on, due to the radiation. Thus nobody had access to the beam-area during data-taking, and all equipment that needed frequent interaction and/or adjustment therefore were placed in the radiation-shielded control-room.

5.2: The Modules

The equipment in the beam area was centred around the modules. The figure below depicts them as they were mounted onto a support structure. A solid granite baseplate of 84 cm length together with the other precise adjustment components yielded a very good placement reproducibility, in the range of 1–2 μm. The modules, mounted onto such a support structure as depicted in figure 5.2, is called a *telescope*.

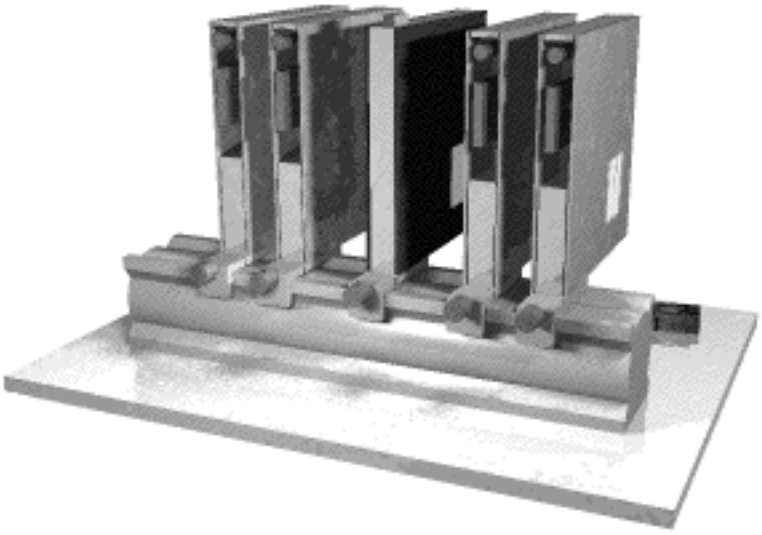


Figure 5.2: The modules as they stood in the beam-line. Four Viking reference-counters and the Felix module in the middle.

As the figure indicates the Felix module was not alone in the beam-line, but was surrounded by four *Viking* modules, two on each side. The Viking chips are the predecessor to the Felix chip, and the Viking modules acted as reference in the beam. This was necessary for two reasons. Firstly, the area of active microstrips were very small in the Felix modules in use at that time. It contained one 32 channels Felix-chip only. The silicon used had strips separated with 50 μm, which made the width of active silicon $32 \cdot 50$ micron = 0.16 mm and with a length of 6 cm. The beam-radius was approximately 1 cm and therefore quite easy to miss. The area of active silicon in the Viking modules was $3.2 \cdot 3.2$ cm, so much larger. It was thus much easier to find the beam in the Viking modules, and they were used to place the Felix modules in the centre of the beam. Secondly, once it was established, with the aid of the Viking modules, that the Felix microstrips were in the right place and were being hit by particles, it made the debugging easier. It made it possible to rule out misplacement and beam-problems, even if no hits could be seen in the Felix module. That was a probable situation, since the Felix and the Vikings had each their own separated read-out system, the Viking system being simpler and better tested.

The main role of the beam telescope was nevertheless to obtain precise reconstruction of particle tracks. The Viking modules, with their slow amplification and a signal to noise ratio of 70, could achieve a hit resolution of 2 μm. By interpolating the track seen in the Viking modules to the Felix module it could be predicted where a hit should be seen in the module under test. The response in the Felix module could then be examined, knowing where a signal should be seen.

The Viking modules were well suited for the job. The Viking chips were profoundly tested and quite reliable. They provided straightforward amplification and

processing of the silicon transducer signal. 10 chips reside inside each Viking module, each with 128 channels. They were connected to two layers of silicon. The microstrips in the two layers were rotated 90 degrees with respect to each other. This facilitated a determination of the x-y coordinates of a particle-hit.

The Viking modules themselves went through basic testing before they were applied in this test-run. Because of all this, the modules were to be trusted, and worked well as a fixed reference-point.

The Felix belongs in an entirely different league than the Viking. The Viking chip belongs to the 'old class' of FE chips. It features a slow, low power charge sensitive amplifier, and was specifically designed to be used with silicon transducers. The amplifier has a rising time in the order of 2 μs, and the amplifier output is transmitted directly to the chip output. The chip is not usable in ATLAS for the reasons of speed discussed in section 4.2.

Of course, the conditions of the testbeam were far from those planned for the finished ATLAS detector, in terms of event frequency and particles per event. In the testbeam, a *spill* occurred every fourteen seconds, with a bunch of approximately 10^4–10^5 particles. Under these conditions, the Viking chips and modules performed more than well enough for the testbeam.

5.3: The Control-System

The most interesting part of the system, seen from an electronics point of view, was in the control-part of figure 5.1. The control-logic was responsible for, upon receiving a trigger from the scintillators, initiating a readout of the Felix and Vikings, and clock all the resulting data from the MUXes of all the modules. The data, together with timing-information, was then written to tape and also displayed briefly on the computer screen used for online monitoring.

5.3.1: The Hardware

The testbeam control system consisted of a computer with relevant software and hardware modules in a VME-crate. The hardware can be described schematically as in the following figure:

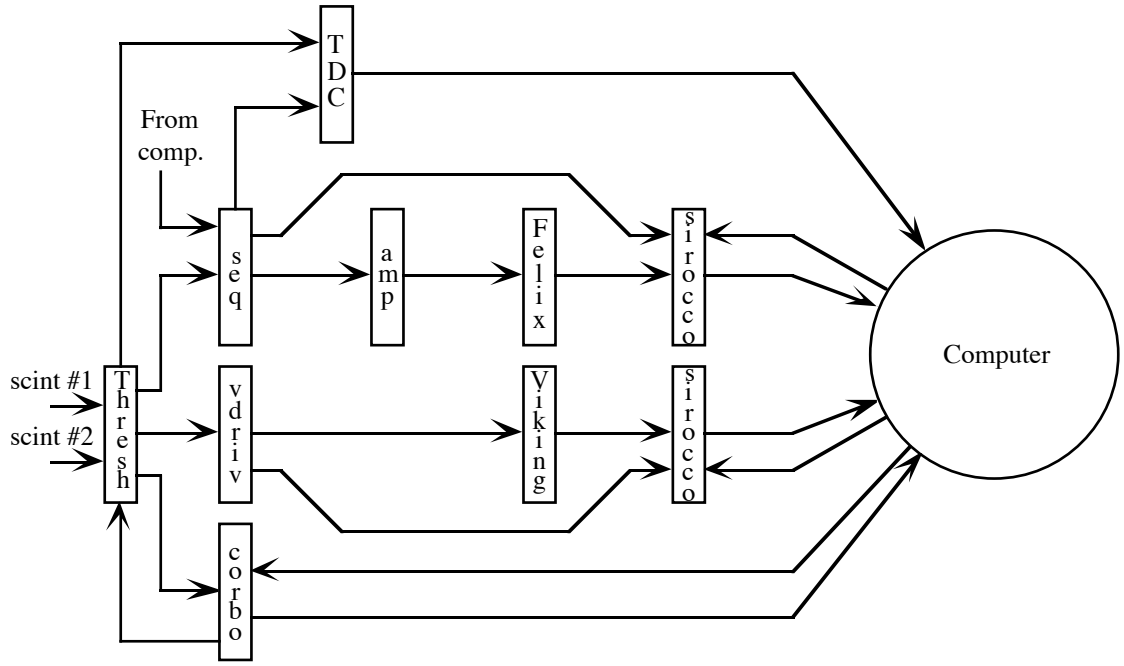


Figure 5.3: The logical setup of the control-system.

Collection of data from an event was initiated by a signal in one or both of the scintillators. These signals passed into a *threshold-unit*. The threshold-unit is a discriminator; it will only pass on a trigger-pulse if the input-signal exceeds a set value. In addition, a switch on the front determines if a coincidence of the two scintillators is needed to produce a trigger, or if either one will do. This and/or function utilised testing of the raw data-throughoutput capabilities of the system (*or* position) or more refined testing of the event-capture/interrupt capabilities (*and* position).

The trigger was distributed to several units from the threshold unit. The following describes the the testbeam hardware procedure for processing an event: The *Corbo*-unit is a programmable interrupt-generator. When it received a trigger from the Threshold unit, it immediately disabled any new triggers from the same unit, and sent a *VME interrupt* to the computer. It would then wait for the computer to signal it to enable the Threshold unit again. The trigger from the Threshold unit would also signal to the Sequencer and Viking Driver that a readout of the Felix and Viking modules should be performed. The samples from the Felix/Vikings would then be sent into the ADCs. The ADCs have an output-register that signals the computer when they have recieved the prescribed number of samples. Independent processes, started by the VME interrupt from the Corbo, ran on the computer that checked those registers. When the outputs got active, the computer would disable the ADCs from accepting more samples and read them. When the computer had read all data from the ADCs it would then enable them again, and also signal "All data read!" to the Corbo, which in turn would enable the threshold unit again. This made sure an event was read entirely before another event was accepted into the testbeam ROE.

The *Viking driver* is a fairly simple unit, reflecting the simplicity of the Viking-modules themselves. To be read out, the Viking-modules need only 1280 clock-pulses, and those were provided by the Viking-driver output upon receiving a trigger. The data from a Viking-module was sent to an ADC, also called a *Sirocco*. A delayed version of the 1280 strobes from the Viking driver made sure the data was read by the ADC.

The *sequencer* (*SEQSI*) is the Felix equivalent of the Viking driver. Given the more complex Felix-chip, it is no surprise that the Sequencer is more complex than the Viking-driver. The sequencer can be viewed as a dedicated, high-speed programmable pulse-generator with 32 parallel outputs. The Felix needs control-pulses to sample and place the samples in the ADB, to initiate and carry out a deconvolution, to scan the MUX to read out all the channels, and to reset the chip upon completion of a read-out. A suitable pulse-sequence was programmed into the sequencer for this (see next section for details about the programming). The output of the sequencer can logically be divided in two: A *default sequence* and a *trigger sequence*. The latter is output once when the sequencer receives a trigger. Thereafter it repeatedly outputs the default sequence until it again receives a trigger. The main signal in the default sequence is the *BCO-clock*. The BCO-clock is a 40 MHz pulse-train which the Felix, among several other things, uses to update a counter. The counter in turn controls a pointer that runs along the ADB and determines where samples from the pre-amp will be placed.

The Felix was read by its Sirocco in a similar way to the Viking. The difference was that the Felix Sirocco received only part of the output from the sequencer, namely the clock-pulses that also were passed to the Felix MUX.

The *TDC* was necessary to extract precise timing information from the system. The problem one faces in a testbeam is this: The scintillator outputs a signal when it is hit by a particle. The same particle also traverses the silicon microstrips in the Felix module and produces a signal in the Felix chip. That signal is to be extracted from the Felix. Because the chip samples continuously, the signal is somewhere inside the ADB. But there is a lot of delaying signal-paths in the system, so there is no easy way to tell in which cell of the ADB it is to be found. Because the Sequencer, and therefore the Felix, operates at a frequency of 40 MHz, 25 ns is the time-resolution of the collected data. The sequencer starts outputting its trigger sequence at 40 MHz on the rising edge of the next BCO clock-pulse when it receives a trigger. The trigger sequence controls how long after that edge the Felix shall start deconvoluting. Or, if the peak-sample is used, how long after that it shall take the sample. What in reality is controlled is which cell(s) in the ADB shall be extracted, as the following figure illustrates. This control is accomplished by adjusting the trigger sequence. Another way of saying it is that a specific *time-bin* is sampled.

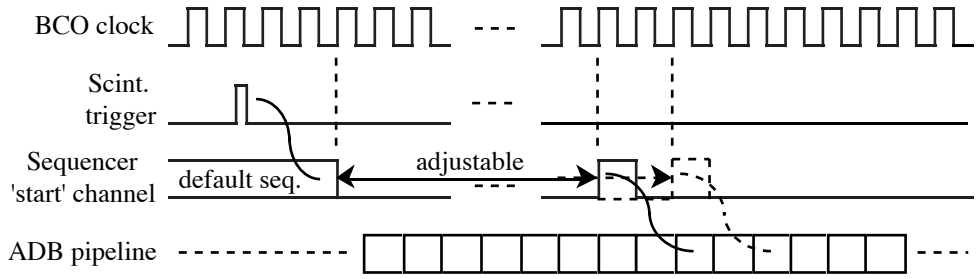


Figure 5.4: Illustrating the process of performing a deconvolution on the correct samples.

Both the scintillator trigger and the ADB pipeline is in the beam area. However, the signal has to go via the Sequencer, which resides in the control room. Thus there is a delay in the cables between the two locations, as well as in the Felix control logic. By stretching or shrinking the length indicated as 'adjustable' in figure 5.4, it is possible to compensate for this delay, ensuring that the sample(s) fetched from the ADB is indeed the one(s) containing the signal which also gave a signal in the scintillator. However, as the time between one time-bin and the next is 25 ns, 25 ns is the minimum adjustment that can be made. As I will explain, this is why there is a need for the TDC.

Before there is any need for the TDC function, however, the correct time-bin has to be found. This was done in the testbeam in a rather straightforward way: Sampling in different time-bins (i.e. with different 'adjustable' lengths, ref. figure 5.4) was tried until signals could be seen in the Felix on the computer-screen. Then data was recorded to tape, some thousand events for that and neighbouring time-bins. Because the Felix preamp has a rising-time of 75 ns, signals can be seen in several neighbouring time-bins. All the collected data was analysed offline and provided an answer to which time-bin had the best signal. That time-bin was then used in further tests. Still, since the time resolution of this scheme is 25 ns, sampling-time could have been off by as much as ±12.5 ns and there would be no way of knowing, had it not been for the TDC. It was important to know, among other things because we wanted to measure signal to noise (S/N) in the system. I explained in the previous chapter how the effectiveness of the deconvolution is dependant on sampling time, so it should be no surprise that S/N is sensitive to the sampling time, as well, and is highest when the sampling time is correct..

The TDC enhances the time resolution of the system, and its function is similar to that of a stopwatch. The unit simply records the time between the start-trigger, which it receives from the threshold-unit, and the stop-trigger, which is the positive edge of the next BCO clock-pulse.

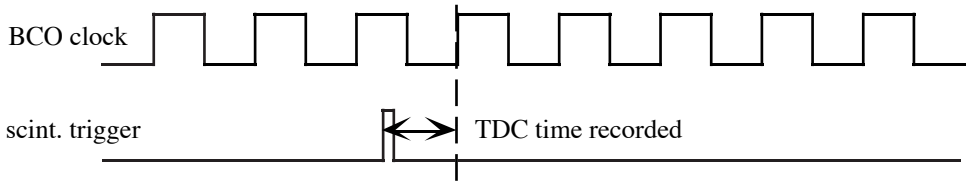


Figure 5.5: TDC function. Compare with figure 5.4.

The scintillator trigger is the product of a particle hit, and as such occurs at completely random moments. The Sequencer trigger-sequence, on the other hand, is synchronous to the BCO clock. Therefore the TDC times will spread out in the interval 0–25 ns. As discussed, there is also a specific time from the scintillator trigger to the moment the Felix ideally should have been sampled. When the Felix is sampled at precisely that moment, the highest sample pulse-height results, especially when deconvolution is used (the peak sample is less sensitive to small fluctuations from the correct sampling time). When sample-time is off, the pulse-height is reduced. Thus, if a plot is made with deconvoluted sample pulse-height versus TDC time in a specific run, a figure which looks more or less like this will result:

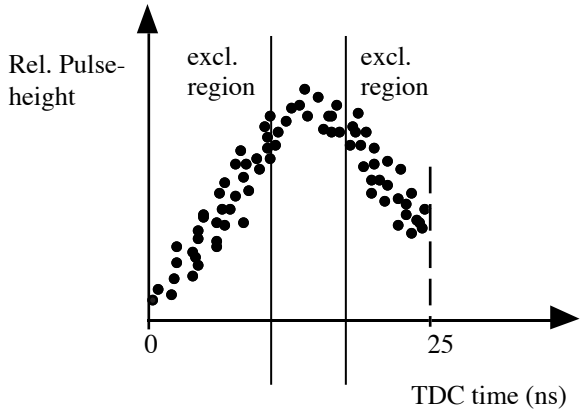


Figure 5.6: Felix sample pulse-height vs TDC time.

The samples are distributed evenly inside the time-bin due to the uncorrelated nature of the scintillator trigger and the BCO clock. The top represent the ideal sampling-time. Samples taken at precisely that moment will yield the correct signal to noise-ratio, because then the samples used for deconvolution is taken at the correct time. To allow for some statistics and small time-fluctuations, it is however usual to include samples from a time-region around the ideal in calculations. The more data has been taken, the narrower the region can be.

It should be mentioned that this is an extra complication which will not exist in the ATLAS detector, because the events will occur at a 40 MHz rate, and so can be made completely synchronous with the Felix sampling. The system will then be tuned so that

sampling occurs at exactly the right moment. Such a scheme is usually not possible in a Testbeam, and the TDC was a vital component in the -94 Testbeam hardware system.

5.3.2: The Software

I briefly mentioned some of the software that runs on the computer in the previous section in order to clarify the function of some of the hardware modules. Since the software is a major part of setting up a testbeam, it is worth discussing in more detail.

The operating system that was used in the -94 testbeam was *OS 9*. This not-too-well-known operating system can be seen as a predecessor to UNIX, and shares essential features such as multitasking with it. OS 9 is also a multi-user OS like UNIX. However, clear signs of age is showing, and CERN is porting more and more of its OS 9 software over to UNIX. Nevertheless, OS 9 has been much used at CERN and provided enough power for the -94 testbeam.

The software used to control the Felix sequencer was a single program. The program could be run once before starting a run, then exited. When run, it would load the default and trigger-sequence into the sequencer, which kept those in its own memory. The normal procedure, however, was to keep the program running. This because it was used to move the sample time-bins as has been explained in section 5.3.1. At startup *Runseq*, as the program was named, would prompt the user for a file, then attempt to read the trigger-sequence from that file. An earlier version of the program had the trigger sequence coded into the program-source. This approach was inflexible, as it required re-compilation each time a different sequence was needed. Taking a deconvoluted sample required a different sequence than taking a peak sample, for instance. Also, as has been discussed, Felix was not the only chip developed for the read-out. Other, similar chips existed, requiring slightly different trigger sequences.

For these reasons, Runseq was modified during the testbeam period to read the sequence from an easily editable file. The file simply contained rows with form of:

1010000100100000

The row would then specify the state of the output-channels in one period of the sequencer's clock. This example row would specify that the first channel is high, the second low, third high and so on for 16 of the sequencer's 32 channels. The rest of the channels are unspecified, but defaults to low. In addition to rows of the above form, the file also contained commands to the program to iterate certain rows a specified number of times. This to avoid actually having to write some thousand lines to specify the complete trigger sequence.

The speed of the sequencer clock was set by writing a specific word to a dedicated register in the sequencer. The word, and therefore the speed of the clock was hardcoded into the program, since its frequency would be fixed at 40 MHz during the testbeam.

The default sequence was also hardcoded into the program, as it needed less frequent modification. Having loaded both sequences, the program would enter a command prompt loop, in which the user could perform simple tasks. These included loading a new sequence and moving the sample time-bin(s) back and forth (ref. 5.3.1). This moving was accomplished by Runseq simply by shifting the trigger sequence that currently was loaded into Runseq back or forth one clock-period, then reloading the new sequence into the Felix sequencer. One of the channels contained a simple strobe which signalled 'start fetching/deconvoluting' to the Felix, and by moving that and all the other control-signal closer to or further away from the moment of the scintillator trigger, we could find the correct sample time-bin.

Sharing the same computer, but on a higher level of complexity, the Data-Acquisition program used to initialise the Corbo and to read out the Siroccos needed some low level programming. The latter can be attributed to the need for accepting interrupts from external hardware. The need to have the program contain concurrent processes, i.e. processes running independent of each other, explains the relative complexity. More so because the processes needed to pass information between them. The programming of both this software and Runseq was done in C, which supports the needed low level commands.

As was done for the hardware in the previous section, the following describes how an event was processed in the Testbeam software.

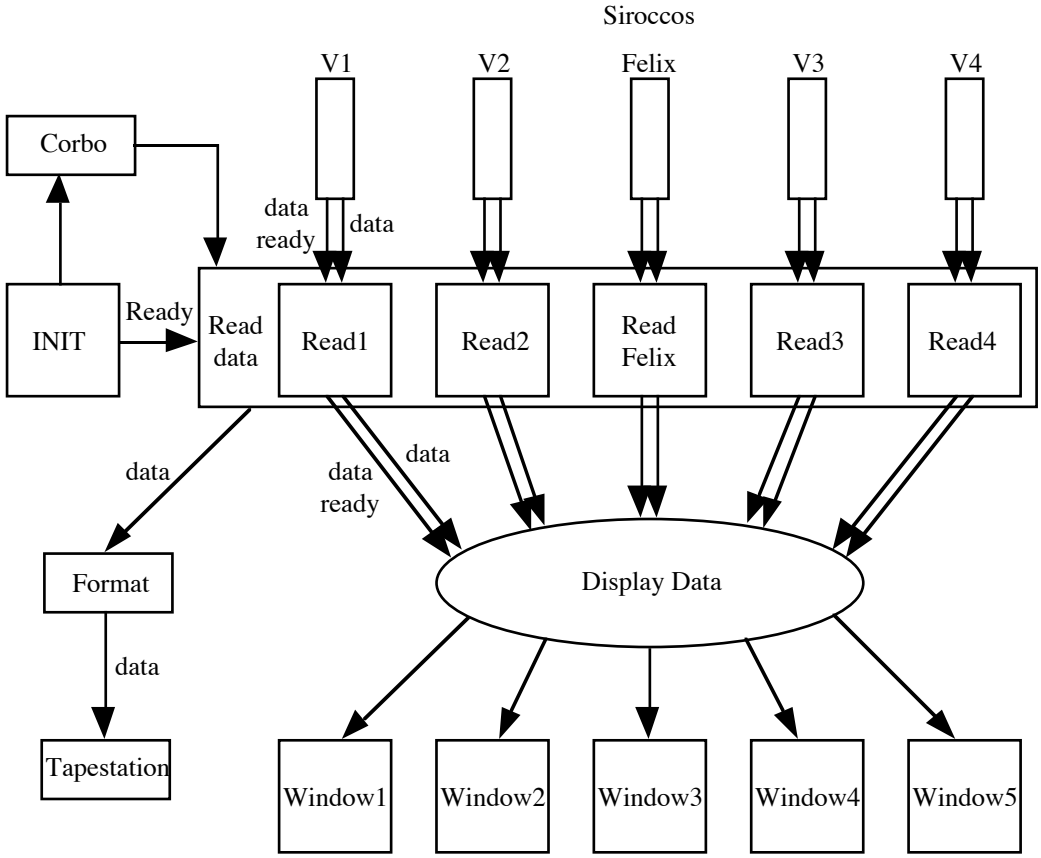


Figure 5.7: Schematic overview of the testbeam software and the relevant hardware. Notice how this figure is linked with figure 5.3 via the Corbo and Siroccos.

The INIT process is run once, at the startup of the software. It initialises the Corbo and Siroccos. An Event-Read is triggered by the Threshold Unit in figure. 5.3, and it in turn makes the Corbo send an interrupt to the Read Data process. The Read Data process will then start waiting for the Sirocco register to signal that data can be fetched from the Siroccos. When that happens, the data starts flowing downwards in figure 5.7. The Read Data process reads all the Siroccos *sequentially* (the Read1, Read2... need not be, and are not concurrent processes) and hands them over to the Format process and the Display Data process. Display Data is a *Random Consumer*. It runs as an independent process with quite low priority, and will only process a sub-sample of the data from the Read Data process. Display Data has an internal structure similar to Read Data, although this is not indicated in the figure. Display Data formats the data in that it turns each sample into a point in the corresponding window, showing the signal strength and the placement in the silicon strips.

The data is also written to tape for offline analysis. The data-path forks from the Read Data process. One fork goes to the Display Data process, one goes to the tape preformatter, which have the highest priority since it must process all samples. Additional

information is added in the preformatter, such as a time-stamp and the TDC time-stamps for the Felix samples. The data is eventually written to tape through a tape-station driver.

The important part of the user interface, the graphical part which told us if we got hits in the modules, was developed during the testbeam period for online monitoring. Previously it had been a crude text-display, outputting raw numbers from the Siroccos. This provided little more information than if the modules were being hit and some indication of the strength of the signal. A more graphical output was needed, which could supply information on *where* in the module the beam was hitting. The Felix module only provided one dimensional hit-information, but the Viking modules had silicon strips in two planes, as has been discussed in section 5.2. Their output could therefore be used to generate a two-dimensional *hit-map*.

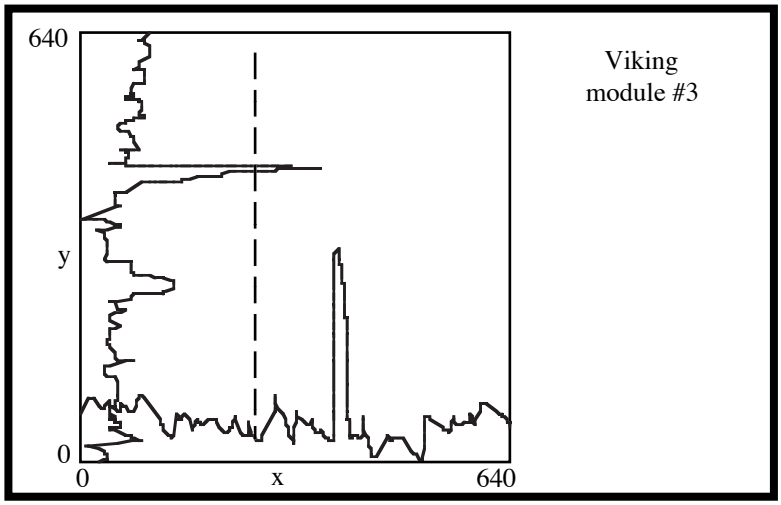


Figure 5.8: Output-window with a hit-map of one of the Viking modules. The placement of the Felix silicon strips in the paper-plane is indicated with the dashed line.

OS 9 can be expanded with a windows package, just like UNIX usually runs with X-windows or other window-servers. This package was used to generate online hit-maps of all modules. The package could be easily controlled from C, with C function calls to generate windows and write in them. The above figure resembles what such a hit-map window looked like. when the Read Data process had read all data into the buffers, the process released the buffers, and they could be read by the Display Data process. The display-process in turn sent the numbers in the buffers to the correct window. The correct window would be the window which belonged to that specific buffer, the buffer in turn belonging to one specific module. The first 640 samples in the buffer came from the five Viking chips connected to the x-direction strips, and the last 640 came from the strips in the y-direction. Thus they were drawn horizontally and vertically in the window, respectively. This made up a window like the one in figure 5.8. The figure indicates a hit in both directions with the two biggest peaks. The rest of the data just shows the

pedestals, i.e. each silicon strip's steady DC level. All the windows were updated for each new scintillator trigger, regardless of if there was a hit in the module or not. Displayed like this, it was easy to distinguish a hit from the pedestals because hits were seen to be clearly sticking out, with one spike on each axis, from the repeated pattern of the pedestals. Furthermore, by observing the output from several consecutive scintillator triggers, a good estimate for where in the modules the beam hit, could be extracted. The strength of the signals was also easily seen.

It is clear that this was no sophisticated graphical tool. There was not much processing of the data (such as pedestal subtraction) before they were displayed, and no output which showed hits over time. But it was a great step forward from the prior stage, and allowed much quicker and better placement of the Felix module into the beam, as well as better positioning of the Viking modules inside the beam.

5.4: Testbeam Results

The testbeam here discussed ran in July/August of 1994. Data was gathered over a few weeks and stored on tape for offline analysis. Although that analysis is not to be discussed here, I will discuss some of the results to justify the set-up.

The modules used were not completely identical to the ones that will sit in the ATLAS silicon detector. For this testbeam we used 3x6 cm wafers with a strip-pitch of 50μm and with readout of every strip. The modules foreseen in ATLAS will be of 12 cm length (two wafers bonded together) and with a readout-pitch of 112.5 μm and one intermediate strip (cap. charge coupling scheme, ref. section 3.4.3). The shorter strips yields a lower Felix input capacitance, something which influences the noise in a positive way. As I will show in the next chapter, the serial noise of the preamplifier is linearly dependant on the input capacitance. The effect was believed, based on earlier measurements, to increase the S/N with a little less than 1/3. Furthermore, we used one single 32 channels Felix chip bonded to a wafer with only 32 strips. The plan for the ATLAS silicon detector is to have double-sided modules with four 128 channels chips bonded to each side, making a total of 1024 strips and channels. Although no 128 channels Felix existed at that time, this was expected to have minor effects on the noise, compared with the effect of the strip-length. Other important issues which would affect S/N included correct sample time, transducer signal shape, correct APSP weight-values and the preamplifier response function. It is possible to come very close to the optimum noise performance by carefully adjusting the three latter quantities to each other. The results from this testbeam would give us an indication as to how far in that process we had come.

It is worth noting that these internal Felix-adjustments only has a potentially positive effect on the *signal-amplitude* after deconvolution; it does not alter the noise. This

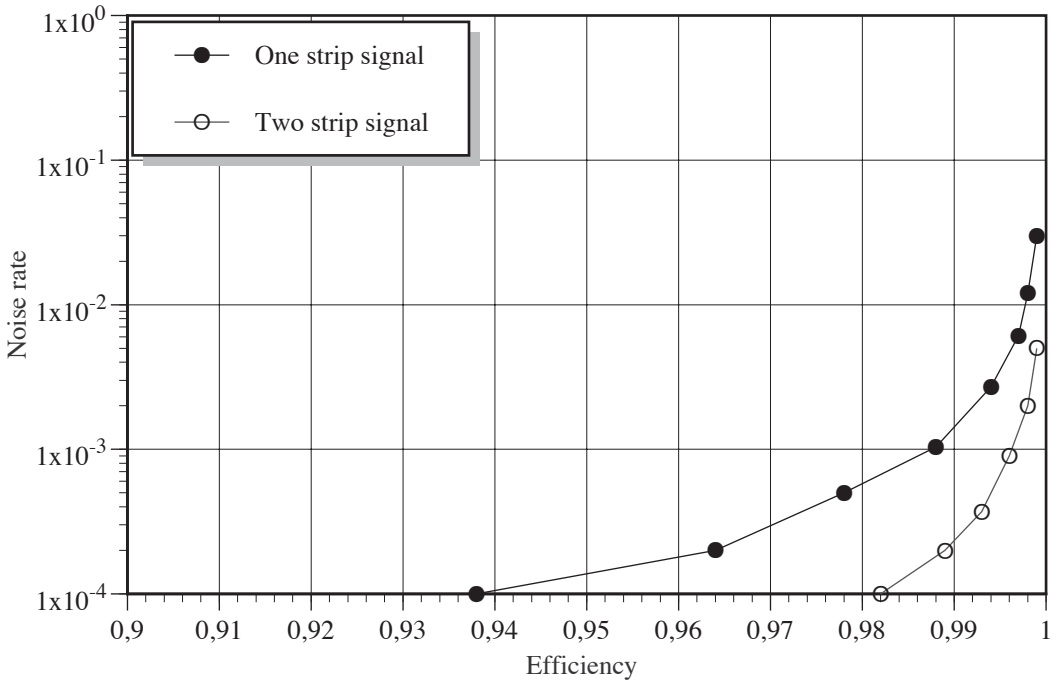
in contrast to the input capacitance, which mostly has an effect on the noise. I will elaborate on this in the next chapter.

After we had gotten the system operational, the scintillators were aligned with the beam. This was easy, as the only requirement was to place them in the beam, no precision alignment was necessary. The two scintillators used where rotated 90° with respect to each other, to get an x/y hit functionality. Then the modules, standing on a*telescope* similar to the one shown in figure 5.2, were aligned with the beam. This was done using the Viking hit-maps. Since the relative placement of the Felix strips to the Viking strips were known, we could be sure that the Felix silicon strips were in the beam after the alignment. Thereafter signals had to be found in the Felix. That is equivalent to finding the correct sample time-bin with the aid of the program Runseq, as discussed. After some scanning, a signal was found, and after some days of data gathering and offline analysis, the correct time-bin for the peak and deconvoluted sample(s) were determined. Once we had found the right time-bin, the rest of the data gathering was done in that time-bin.

There are four quantities that give an estimate of the quality of the Felix used for this testbeam: *Signal over noise* (S/N), space *resolution*, *efficiency* and the *noise hit probability*.

The resolution is defined and found by plotting the distance between the hit-positions found in the Felix module and the position found from interpolation tracks from the Viking modules. When properly aligned, this will typically yield a distribution around zero, from which a function is fitted and then the resolution found from eq. 3.4.

The efficiency and the noise hit probability (also called noise rate) are related to each other, as demonstrated by figure 5.9:


One strip signal
Two strip signal
Noise rate
Efficiency
1x10^0
1x10^-1
1x10^-2
1x10^-3
1x10^-4
0,9
0,91
0,92
0,93
0,94
0,95
0,96
0,97
0,98
0,99
1

in figure 5.9). Unfortunately, it also means that more real signals with amplitudes under the limit will be rejected as noise. If the limit is set lower, the efficiency improves, but more noise passes through as signals. These probabilities can be calculated for a given cutting algorithm when the noise-distribution is know, and the more noise passes as signals, the higher the uncertainties of the results. Having high efficiency with low noise rate is therefore of major importance.

The open circles in figure 5.9 show a way to achieve this by using a better cutting algorithm, where the algorithm looks at two neighbouring strips when determining whether there has been a real hit or not. This improves noise-rate/efficiency by a great deal, because the amplitudes on two neighbouring strips resulting from a particle passing between them will both be high, and will add up to the total 1 MIP signal, as shown in figure 2.4. The noise-amplitudes on two neighbouring strips, on the other hand, do not correlate, and the probability that they both are so high as to pass for a real signal is very small. As figure 5.9 shows, it was possible to achieve an efficiency of almost 99% with a noise rate of $2 \cdot 10^{-4}$ in 1994, i.e. one out of 5000 strips where there was no hit will give a fake signal that pass the cutting algorithm as a hit.

The results from this testbeam, being the first time a complete readout chip with LHC speed-specifications was operated in a testbeam, were very encouraging, and are listed below[14]:

- Signal over noise: 10.5
- Resolution σ: 9.1 μm
- Efficiency: 98.9%
- ENC 1850 e^-
- Noise hit probability; 10^{-4}

These results showed us that we certainly were on the track. Of course, the parameters listed above are also subject to the ATLAS specifications. Among many other demands, the specifications for the silicon detector system state that S/N must be above 15 and the noise must be below 1500 e^-. As can be seen, these demands were not met. That was however not the goal for this testbeam. When the result were considered encouraging, it was because the team felt confident that several further optimisations could be made, enhanching this ‘first edition’ Felix chip into a readout chip meeting all the ATLAS specifications.

6: Felix Input Capacitance Experiment

It should be clear from the previous chapter that minimizing the noise added by the read-out electronics is a high priority issue. Each component in the read-out chain needs to be examined with scrutiny in search of ways to minimise its contribution to the total system signal to noise. The Felix chip is no exception.

The experiment here to be described was conducted at CERN in September -94. Its aim was to investigate the correlation between noise in the preamplifier and input capacitance.

6.1: Preamplifier Noise

The Felix chip is a complex system in itself. To achieve the highest possible Felix signal to noise (S/N), the chip is broken down in its individual components and each component is optimised with respect to S/N. The performance of signal to noise of the Felix chip depends on a large number of parameters. The biggest contributor to the noise is the preamplifier. The other stages, except for the APSP, only provide storage of the samples and has smaller influence on the noise. Optimising the preamp should therefore be done very carefully. The preamp noise depends on both internal and external characteristics. Of the latter one of the more important is input capacitance, i.e. the apparent capacitance the input of the pre-amp sees from the silicon microstrips. To optimise the signal to noise for the capacitance of the micro-strips as it will be in the ATLAS-detector is therefore very important.

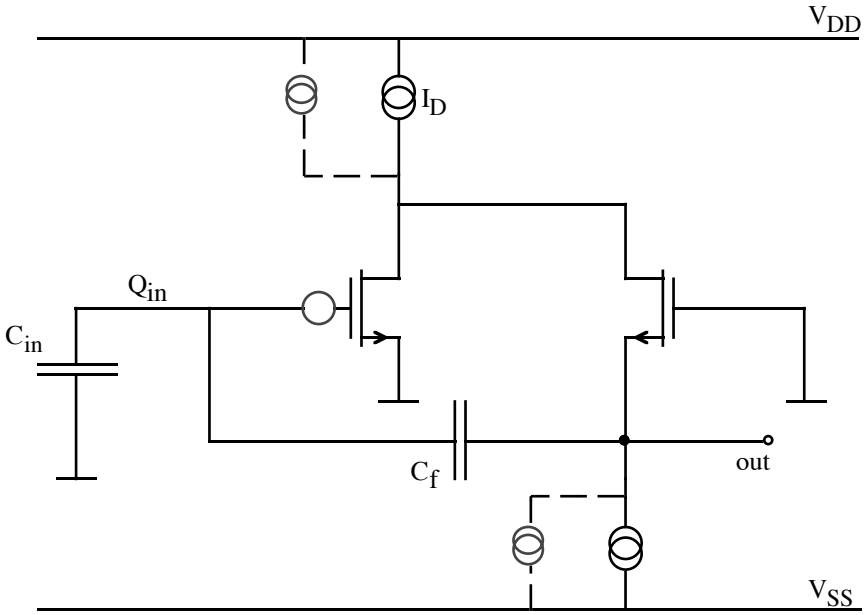


Figure 6.1: Standard charge sensitive amplifier (CSA) schematic diagram, including the dominant noise sources (greyed)[2].

There are several noise sources. However, not all of them are dependant of the input capacitance. This chapter is mostly concerned with those arising from characteristics in the silicon transducer. These are dominant, and the noise from internal sources such as I_D are not significant in comparison.

6.1.1: Serial Noise

For the Felix preamplifier (including the CR-RC shaper) the dominant serial noise in ENC (Equivalent Noise Charge) is given as[3]:

$$ENC(C_{in}) = \frac{e \cdot C_{in}}{q} \sqrt{\frac{\Gamma \cdot kT(\eta + 1)}{3T_p \cdot g_m}} \quad [e^-] \qquad \text{Eq. 6.1}$$

The definition of the quantities follows:

e	=	2.71
q	=	electron charge
C_{in}	=	total capacitance at input
T_p	=	shaper peaking time
g_m	=	transconductance of input FET
Γ	=	quality factor of the amplification process
η	=	ratio of bulk to channel and gate to channel transconductances

Here is $\Gamma = 1$ the theoretically optimal value. The manufacturing process of Felix should give a Γ very near 1. Likewise, the process should give a negligible η in most cases.

C_{in} is the total capacitance the amplifier input sees, and is made up from several contributions:

$$C_{in} = C_{is} + C_{BP} + C_{inpFET} + C_{stray} \qquad \text{Eq. 6.2}$$

Here, C_{is} is defined as the interstrip capacitance one strip sees from all the neighbouring strips. Similarly, C_{BP} is the total capacitance to the backplane seen by one strip. All the components are expected to add up like $C_{in} = 6 + 1.2 + 4 + 2 \approx 15pF$ for 6 cm modules with a pitch of 100 μm[3]. For 12 cm long modules $C_{in} \approx 23pF$ is expected[3].

T_p is given for the Felix application to be 75 ns, so the only possibility left for minimizing the serial noise is by increasing the amplifier's transconductance g_m. However, since $g_m = i_d / v_{gs}$ and thus depends on the drain current, g_m is restricted by the input current to the amplifier. The preamplifier of the most recent Felix chip, the 128-channels version, has a current drain of 700 μA and a transconductance of 4.5 mA/V. The version we worked with in this experiment , the 32-channels version, had a current drain of 400 μA.

As can be seen, the serial noise increases linearly with the total input capacitance. A quite straightforward application of the equation with the numbers given, and setting $\Gamma = 1$ and $\eta = 0$, yields a noise-slope

$$ENC / C \approx 35e^- / pF \qquad \text{Eq. 6.3}$$

for $T_p = 75ns$. However, the effect of the deconvolution has to be taken into account. With a deconvolution that has the effect of reducing the effective T_p with 2/3, as is the case here, the slope of the serial noise after deconvolution becomes[3]

$$ENC / C \approx 64e^- / pF \qquad \text{Eq. 6.4}$$

The Felix chip we measured on had however a lower transconductance than 4.5 mA/V, and we consequently could expect a steeper noise-slope. Also, the values of $\Gamma$and η were not necessarily close to the optimal values.

6.1.2: Parallel (Shot Noise) From Leakage Current

The other dominant noise source in a silicon detector module is that from leakage current in the silicon transducer junctions. It will give a contribution in the CR-RC filter of the Felix preamp as[3]

$$ENC(I_l) = \frac{e}{q}\sqrt{\frac{q \cdot I_l \cdot T_p}{4}} \text{ [e}^-\text{]}, \qquad \text{Eq. 6.5}$$

where I_l is the leakage current. The shot noise does not depend on the input capacitance, but is a major noise source, not to be ignored. And since the leakage current will increase with time, as discussed in section 3.4.2, the noise will also increase with time, and will eventually become the dominant noise source. The leakage current in the ATLAS silicon detector modules is dependant of a number of factors which it is beyond the scope of this thesis to discuss. It is however expected to be in the range 1–15 μA[3]. If the ENC for $I_l = 2\mu A$ is calculated, again using $T_p = 75ns$, the result will be $ENC(I_l) \approx 1300$. The deconvolution has an effect here, as well. It will reduce the shot noise with a factor of 0.35, yielding $ENC(I_l) \approx 500$ after deconvolution[3].

6.1.3: Expected Noise

The primary goal of this experiment was to investigate the noise-slope of the Felix at hand, which is dependant on C_{in} in the way shown above. To the effect of leakage current are attached quite substantial uncertainties, especially regarding its variation over time due to radiation in the operating detector. Equation 6.5 is anyhow not relevant for this experiment, since the preamp input was not connected to any detector.

We can therefore expect results which yield a noise-slope of over 70 e^-/pF, since the Felix we used was not believed to be fully optimized. And we expect a constant noise-factor, as well. This arises from the ever-present capacitance of the input FET and other stray capacitance. Its magnitude is hard to predict, but earlier experiments suggest something in the vicinity of 500–800 e^-.

6.2: Experimental Setup

Since the biggest contribution to the noise comes from the pre-amp, the Felix chip as it was designed in the testing stages utilised a so-called *broken channel*. This is an output-pin on the chip connected directly to the output of the pre-amp, and the output of the pre-amp is disconnected from the rest of the chain. This ensures that cause and effect is isolated to the preamp alone.

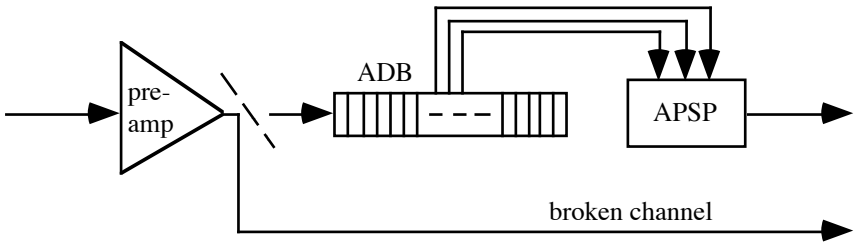


Figure 6.2: Felix broken channel diagram.

One might ask why the noise characteristics of the preamp were not already carefully tested when it existed as its own chip. It was, but the preamp that was included on the Felix chip incorporated several optimizations not present in the versions of the preamp that existed as separated chips, and so the noise characteristics of the old preamps were no longer valid for the preamp in the Felix.

Since we wanted to measure signal to noise, we needed a signal. A 'real' signal from a particle hit was not necessary, if only the signal resembled the shape of such a signal. Of course, radioactive sources are available in the lab, but an output-signal from a high-speed pulse generator proved sufficient. The signal was tuned to feed the equivalent charge of a 1 MIP (22000 electrons) particle into the preamp.

Since we took the signal off the Felix right after the pre-amp, we had access to neither the ADB nor the APSP. Measuring the deconvoluted S/N was a necessity, thus we had to sample and deconvolute externally. For this we needed some storing/processing capabilities. The test-setup that we used is shown schematically in the figure below:

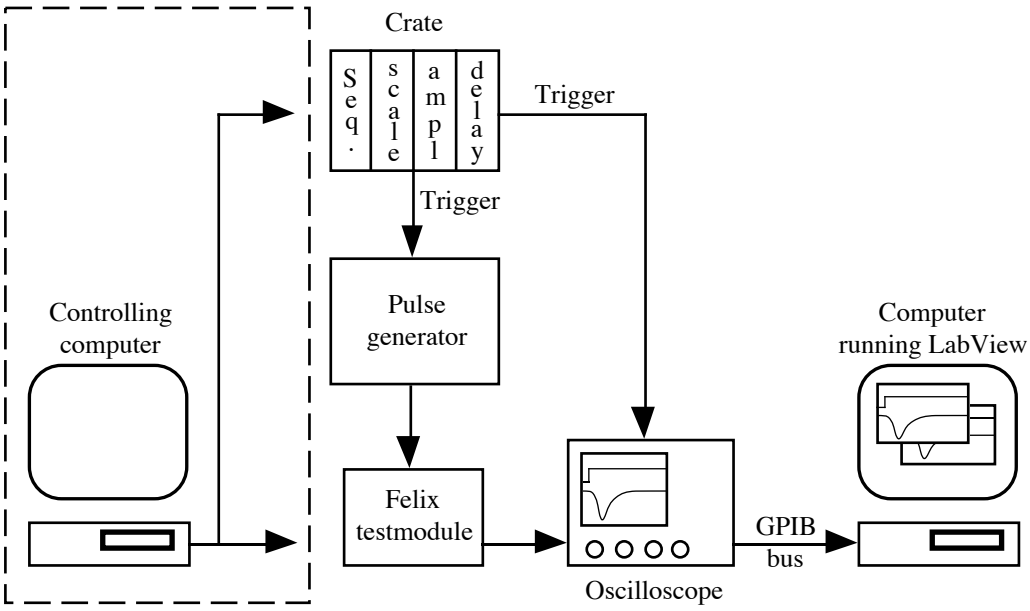


Figure 6.3: Schematic diagram of the test-setup, showing the control-signal generation equipment, the object under test (Felix), and the measuring equipment.

According to our needs, the setup consisted of these components: The Felix chip itself, mounted on a PCB. The PCB contained pins for connecting the Felix to the necessary voltages, as well as facilities for connecting different capacitors across the preamp input of a broken channel. Power supplies are not shown in the figure. The reason the computer is shown 'hooked off', is because it provided no real Felix control. Since we were taking the samples off before the ADB, there was no need for control-signals to the

ADB, APSP and the MUX. Still, such control-signal had a potential influence on the noise, and we wanted to run the experiment with the signals the chip would receive in normal operation. The trigger was simply a repeated pulse coming from the Sequencer. Upon receiving the trigger, the Felix would output a signal, which was fed to a high-speed, digital oscilloscope. The oscilloscope-display was triggered by a delayed version of the same trigger as the pulse-generator received. A PC was connected to the scope over a GPIB-bus. The scope could output the received waveforms through this bus, thus the PC was fed the output-signal from the Felix. *LabView* was running on the PC. LabView can be described a programmable multi-purpose software-instrument. With it we could perform a multitude of calculations on the input, and we could view the results numerically or graphically in real-time, accumulate and average sample continuously, as well as view other relevant dynamic or static information as we chose.

6.3: Experiment Progress

Describing how LabView is programmed is beyond the scope of this text, but it employs a form of graphical programming where pre-defined input, output and computing elements are connected with lines representing the information that flows between those elements. The elements are easy to modify to suit the needs of a broad range of tasks. We used it to make a system for fetching the samples from the scope and display them in real-time, as well as calculating noise, peak- and deconvoluted signal-amplitudes and S/N. Figure 4.4 is a direct screen-dump from LabView[4].

Before we applied a signal to the Felix, we wanted to measure the *pedestals*, i.e. the DC voltage-level on the input. The noise is then measured by accumulating pedestal amplitudes from many triggers, averaging them and then calculating their fluctuations around that average value.

After that a signal from the programmable pulse-generator is applied, and the resulting output from the preamp is measured, also by measuring the signal over many triggers and averaging it. These two steps were repeated for every value of the input capacitance we wanted to measure for.

There are two ways in which the signal can be measured: In peak-mode or in deconvoluted mode. See chapter 4 for a discussion on these. When we wanted to measure deconvoluted we used LabView to apply the three correct weights to the sampled preamp signal. Deconvolution gives the most correct view on S/N, as discussed in chapter 4. However, the effectiveness of the deconvolution technique is depending on the shape of the input signal, the amplifier response and that the samples are taken at the right moment. This chapter deals only with the results obtained in deconvoluted mode, because

4 Figure 4.5 is rebuilt from a bad scan of such a screen-shot.

the aim of this experiment was to investigate the quality of the current Felix chip in terms of its preamp/APSP characteristica.

6.4: Results

Capacitors from the standard series were used for this experiment, with values ranging from 1.8 to 22 pF, as can be se in table 6.1. The table shows the result from the deconvoluted run.

Table 6.1: Results from input capacitance experiment.

Capacitance (pF)	Pedestal (mV)	Noise (mV)	Deconv. signal mean (mV)	S/N
0	-0.5	2.3	37	16.3
1.8	-0.2	2.4	36	15.1
3.3	-0.6	2.6	32	12.4
10	-0.5	2.8	29	10.5
15	0.2	3.4	25	7.3
22	0.5	3.2	23	7.0

As can be readily seen from the table, the noise increases and S/N sinks with higher input capacitance. This is what was expected from the preceding sections. That the signal itself should sink, however, was unexpected. The signal-reduction of almost 40% over the range of input capacitors is not completely understood.

It can be seen from the table that S/N for 15 pF input capacitance was measured to 7.3. This may seem to disagree with the result quoted for the testbeam in the previous chapter, since 15 pF was the total capacitance found for 6 cm strips. But as a matter of fact the two results for S/N are in very good agreement. The reason is the that the capacitance in this experiment included the internal preamp contributions, whereas it is more usual to treat them as a constant factor, since they are independent of what kind of transducer-module Felix is connected to. The capacitance per strip in the 6 cm, 50/50 μm modules used for the testbeam was 10 pF, so to compare the results from this experiment to the testbeam-results, the values at $C_{in} = 10\,pF$ have to be used. S/N at $C_{in} = 10\,pF$ was measured to 10.5 in this experiment. This is in fact the same value measured in the testbeam , and so is in exellent agreement with those results.

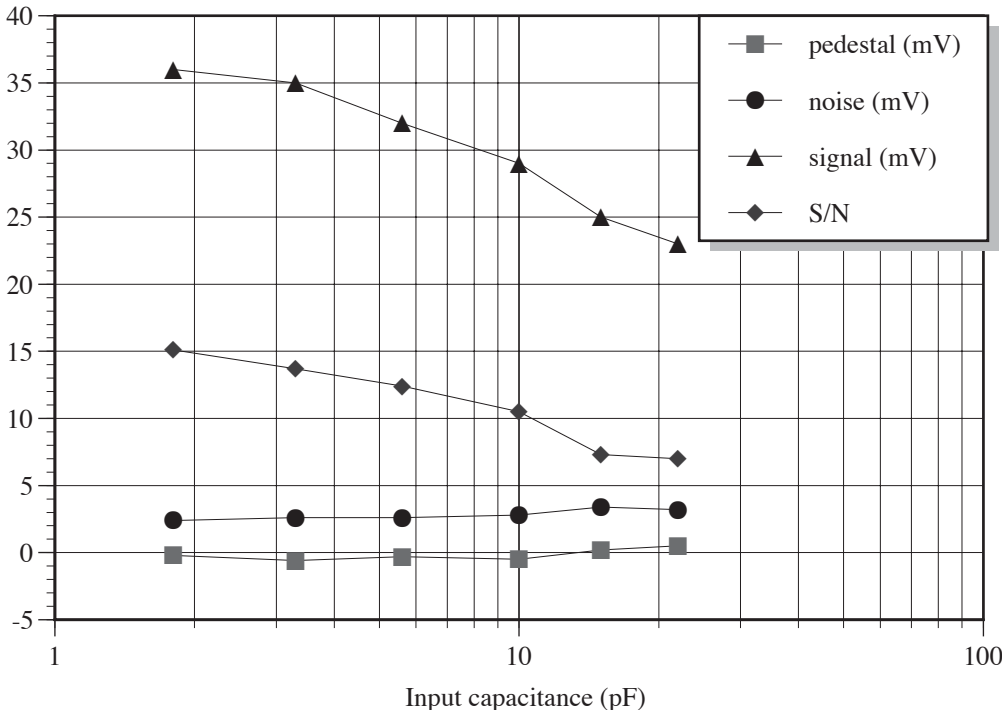
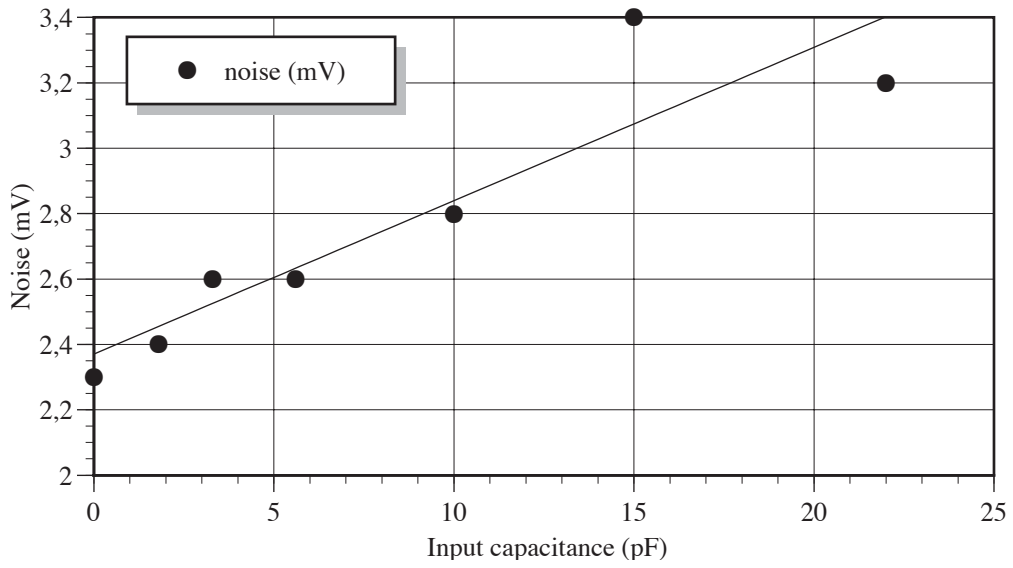


Figure 6.4: Signal over noise as a function of input-capacitance in pF.

Figure 6.4 shows the results graphically. The figure seems to indicate some deviation from the otherwise quite smooth curves at $C_{in} = 15pF$. As can be seen, S/N makes a marked dip from $C_{in} = 10pF$ to 15 pF, and sinks only very little from 15 to 22 pF. Looking at the noise with a bit more scrutiny, it is clear from section 6.1.1 that the noise is expected to rise linearly with input capacitance. Figure 6.5 shows a straight-line curve-fit to the noise values. As can be seen from the figure, the line does not fit extremely well to the values, and the deviation is especially large at $C_{in} = 15pF$.



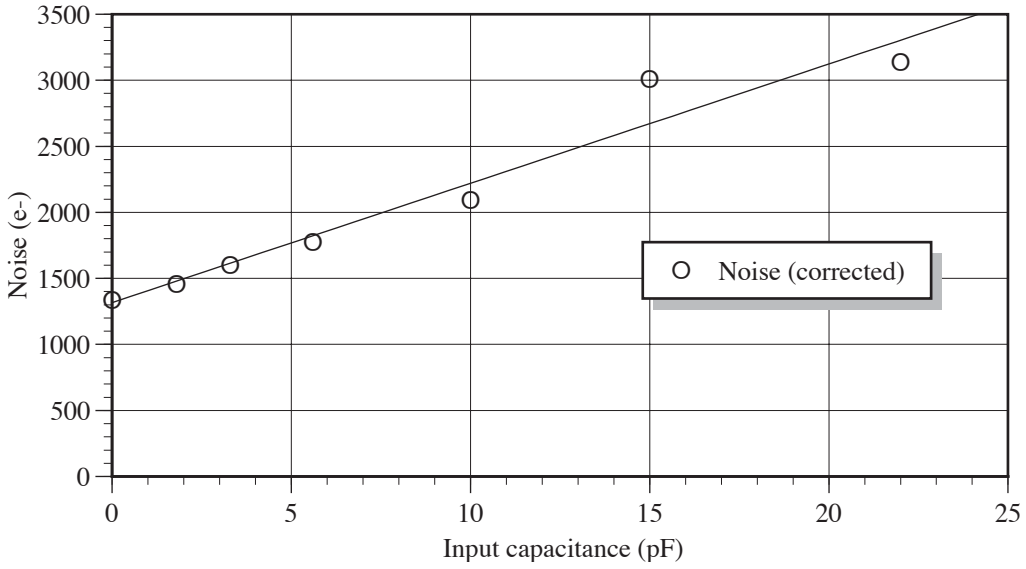
The resulting function for the straight line is

$$f(x) = 0.047x + 2.37 \text{ [mV]} \qquad \text{Eq. 6.7}$$

with x measured in pF. If this function is taken to represent our results for the noise, the slope can be calculated in ENC. 0.047 mV/pF yields $(0.047 / 37) \cdot 22000 = 28 e^{-} / pF$. This is under half of what was predicted for a 'perfect' Felix in section 6.1.1. The constant part of the noise is on the other hand very high: 2.37 mV is equivalent to 1410 e^- for the Felix in our experiment. There is no reason to believe in a too low noise-slope. As we have assumed a perfect Felix in the calculation, there is no way of making the expected noise-slope less steep without increasing the transconductance. For this Felix the transconductance was probably *lower*, not higher, than the value used in the calculation of the noise-slope.

From these results we could only conclude that the Felix on which this experiment was conducted, or our setup, must have contained one or more noise-sources other than those we wished to have, and that they had the effect of reducing the signal and introducing an additional constant noise.

One possible way to 'rescue' the results is to use the S/N ratio. We may expect the unknown factor of our setup to affect both the signal and the noise in a similar manner, which is to basically introduce a gain-reduction. Thus the ratio S/N may still be valid. And if we assume that the input-signal was constant, with the value 1 MIP, we can divide 1 MIP with the S/N ratio to obtain a corrected noise-function, as in this figure:



. The fitted function is in this case

$$f(x) = 90.3x + 1317 \ [e^-] \qquad \text{Eq. 6.8}$$

so that the result is a noise-slope of 90.3 e^-/pF and a constant noise of 1317 electrons. This result is more in agreement with previous experiments and the theoretical calculations in section 6.1. The slope calculated in section 6.1 was 64 e^-/pF, but that was for a fully optimized Felix with higher transconductance. If this result is to be believed, further optimization should be performed to get closer to a slope of 64.

Of course, caution should be taken in adopting these results as true just because they seem more agreeable. It can be noted that this result for the noise, some 2200 electrons at 10 pF, is quite high compared to the value obtained in the testbeam in the previous chapter.

All in all the conclusion had to be that we probably had a set-up that corrupted our output somewhat, and that we therefore should be careful in making definite claims about the correctness of our results. In particular it seems likely there was a noise source in the system which introduced an additional constant noise factor. The drop of signal-amplitude may be due to the increased load at the input.

In retrospect we also know (from later versions of the chip) that the output driver of the broken channel introduced an additional time-constant and possibly additional noise in the system.

However, the S/N result suggest that there was still work to be done to optimize the Felix chip in terms of deconvoluted read-out of signals. This has indeed been done since the experiment here discussed was conducted, and the last section of the next chapter will discuss some later results.

7: KEK Testbeam Experiment

From the 23rd of February to the 2nd of March 1995 an experiment using a particle-beam at KEK, Japan, was conducted to investigate the effect of a high magnetic field upon the silicon microstrip transducers. In addition, the intention was to test a new Felix 32 channel chip before submitting a 128 channel version later the same Spring. The following discusses the motivation, setup and running of that experiment. The results are also outlined and discussed.

7.1: KEK

The National Laboratory for High Energy Physics (*KEK*), is a laboratory-site outside the scientific city of Tsukuba, Japan, approximately 50 km north of Tokyo. The facility is younger than CERN, it came into operation in 1971. Its first major facility was a proton synchrotron with a centre-of -mass energy of 12 GeV. The biggest accelerator at KEK is at present TRISTAN, a 30 GeV electron-positron colliding beam accelerator. Building of a new accelerator inside the TRISTAN ring, designed for B-physics, is underway.

KEK is an active participant in international particle physics collaborations. Groups form several countries participates in the experiments at KEK, and the laboratory has a generally open policy, enabling short-term experiments like the one that will be discussed in this chapter.

The reason for doing the experiment at KEK was quite simple. CERN is a laboratory buzzing with activity, and the planning of LHC only increases that activity. Schedules can be tight, as can access to testbeam facilities. For this specific experiment, a very high magnetic field was needed. That meant the use of a strong and large magnet, of which there are only a one or two at CERN. KEK could provide such a magnet *and* the time needed to use it.

Once the decision was made, the group of four picked to go there gathered the equipment necessary, the modules, scintillators, power-supplies, cables, computer etc. and shipped it for Japan well ahead of the scheduled testbeam time, 23. of February to the 2. of March. One person (me) went ahead of the others, one week before the testbeam would start, to receive the equipment and set it up as far as possible. Unfortunately, time proved tight, partly due to the equipment being held in Japan custom control, partly due to ‘standard-problems’. Power-lines in Japan are 100 volts, all the cable connectors that are ‘standard’ in Western Europe are not in Japan etc. This to mention but a few of the problems.

A new Felix 32-channel version was used in this testbeam. Later the same Spring and Summer the tests continued at CERN with a new 128-channel Felix optimized further. At the end of this chapter some of the results obtained with the new chip are quoted.

My role in the testbeam work terminated with the KEK-run and therefore only a few final results are given to indicate how the program developed throughout 1995.

7.2: High Magnetic Field Impact

As the development of the various components of the future ATLAS-detector continues, the test-conditions of these components are gradually moved more and more into those they will operate under in the finished detector. It is of course important to pin down and remove problems in the early stages of development.

One of the potentially severe sources of problems/errors in a silicon microstrip detector is the effect of the high magnetic field under which the detector will operate. More specifically, the presence of a high magnetic field will distort the distribution of the charge the strips of a silicon transducer receive when a particle passes through. This will potentially shift the charges so much that it will appear as if the particle passed near a neighbouring strip.

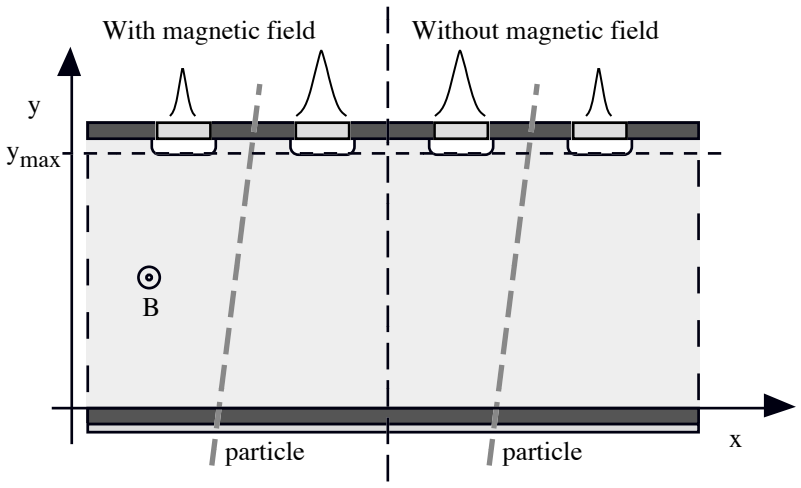


Figure 7.1: Charge displacement caused by a magnetic field in silicon strip transducers.

The e-h pairs knocked loose by the passing particle will be influenced by the magnetic field, according to

$$\vec{F} = q\vec{v} \times \vec{B}. \qquad \text{Eq. 7.1}$$

Since it is the holes which drift towards the strip junctions, at the same time they will drift rightwards in the figure, and the result is that the hit will appear to have occurred nearer the right strip than is the case.

Equation 3.2 expresses the electric field as a function of y in the n-type bulk of a silicon transducer. Since the force F on a charge-carrier due to the electric field is $F = ma = qE$, and the velocity such a carrier gains between collisions is $v_d = at$, the drift velocity v_d is proportional to the field E and is

$$v_d = \mu E, \qquad \text{Eq. 7.2}$$

where μ is the mobility of the charge-carriers. In the depleted n-type bulk, the holes are the signal charge-carriers, and $\mu_p = 0.048 m^2 V^{-1} s^{-1}$. The velocity of the holes in the n-type bulk due to the electric field is therefore given as

$$v_d(y) = \frac{\mu_p q n}{11.7} y. \qquad \text{Eq. 7.3}$$

The coordinate-system is defined in figure 7.1, and all of these expressions are only valid in the interval $0 \le y < y_{\max}$, i.e. inside the n-type bulk. By using the expression for the drift velocity at point y on equation 7.1, we arrive at an expression for the force on a hole along the x-axis due to the magnetic field B:

$$F_x(y) \approx \frac{1}{11.7} \mu_p q^2 n B \cdot y. \qquad \text{Eq. 7.4}$$

This expression is only an approximation because the magnetic field will bend the direction of the drift velocity, and hence the direction of F. That will result in a curved hole-trajectory, whereas the holes in this approximation will drift in a straight line, as shall be demonstrated. The approximation can therefore be expected to under-estimate the charge-displacement caused by the magnetic field.

Let us however for the moment assume that equation 7.4 is correct, and use it to calculate the expected maximum displacement. In section 3.4.1 we calculated an expression for E(y), and we can use that to find

$$F_y(y) = qE(y) = \frac{q^2 n}{11.7} y. \qquad \text{Eq. 7.5}$$

From eq. 7.4 and 7.5 we see that the ratio F_x / F_y is constant in our approximation, and some simple algebra yields:

$$\frac{F_x}{F_y} \approx \mu_p B = 0.096 \qquad \text{Eq. 7.6}$$

for a 2 Tesla magnetic field. This will cause a maximum single-charge displacement of $300\mu m \cdot 0.096 = 28.8\mu m$. This displacement is slightly smaller than the measured experimental value, and our approximation does indeed yield an under-estimation, although small.

Keep in mind that this effect should not be confused with the bending of a passing particle, indicated with the dashed, thick lines in figure 7.1. That effect is of course desirable, and is the reason why there are magnets in the detector in the first place. It is the effect on the e-h pairs inside the silicon transducer that is an undesired, complicating factor.

It is possible to alleviate the problems caused by this effect by slightly *tilting* the transducer-module to compensate for the drift due to the magnetic field. If we accept the value of a 28.8 μm displacement, the modules should be rotated through an angle $|\varphi|$ so that $\varphi = a\tan(28.8/300) = 5.5^{\circ}$. To determine which way to rotate the modules is left as an exercise for the reader, but figure 2.3 shows the correct direction for a magnetic field pointing outwards from the paper-plane.

Finally it must be mentioned that the effect of the magnetic force is not really a shifting of the entire signal distribution, rather that the magnetic field will cause a certain *smearing* of the signal. This is due to the fact that holes created near the backplane of the transducer have the longest drift-distance, and therefore are more displaced by the magnetic field. The holes produced near the strips will be under the influence of the magnetic field only for a short period, thus the magnetic field will extend the signal-distribution.

Since the charge distribution of two neighbour-strips is not straightforward even in the non-magnetic field case (the pulse height distribution is not linear with respect to the hit position between the strips), the effect of a magnetic field has to be investigated, theoretically as well as experimentally.

7.3: Experimental Progress

To investigate experimentally the effect discussed in the previous section, we wanted to place the Felix module in a high magnetic field in a way shown below. The Viking modules were also present, like in the testbeam discussed in chapter 5, although they are not shown. The field is set up by a electromagnetic magnet in the beam-line, also holding the strips under test. The module is placed so as to make the strips parallel to the field. This is the same situation shown in figure 7.1, and will also be the situation in the ATLAS detector, where the strips and the magnetic field will be parallel to the Z beam-axis.

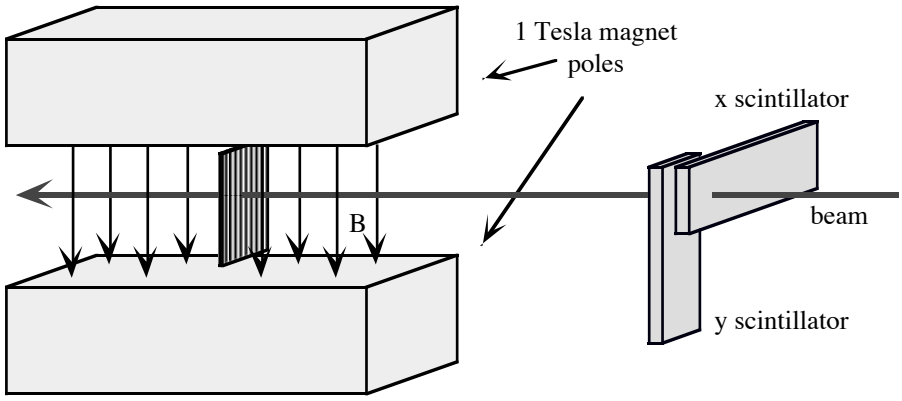


Figure 7.2: Position of scintillators, magnet and modules relative to the beam-line.

Generally, the testbeam setup was much the same as in the testbeam of the summer of -94, discussed in chapter 5. We had a setup which we knew should work well, and there was no reason to change too much. The only new thing we wanted to introduce, was the magnetic field and a few modules with the new Felix version, as mentioned. We had our modules, power supplies and scintillator in the beam area, and our VME crate with Sequencer, Siroccos and the other controlling hardware and software in the control room.

As in the testbeam discussed in chapter 5, the first task was to place the Felix module into the beam, i.e. to align all the modules correctly in the beam-line, first without the magnet switched on. Once we had the modules in place, we would proceed by finding the correct time-bin. In fact, even the cables stretched from the beam area to the control room were mainly the same as the ones used for the -94 testbeam. Therefore the delay in our system should be approximately the same, and we could expect to find the best signal in the same time-bin, or in one of the neighbouring, as in the -94 testbeam.

It seemed, however, that one of the Felix modules we had brought along had turned faulty as a result of the transportation, and we had problems finding any signal in it whatsoever. The other Felix module we had brought looked better, though. But still alignment in the beam and finding the correct time-bin took time. It did not help that our usual offline-analysist were at CERN. This meant that each run we had taken, had to be stored on the KEK computer system for him to analyse. And since Japan time is off by eight hours compared with CET, that usually took a night.

With all our problems with seeing hits in the Felix, it began to be clear that we needed a better tool for seeing exactly where the beam hit the modules. To meet that demand, the display routines of the software were altered, and enhanced to produce a Viking hit-map that showed hits over time. When left to run during a data-taking period, the routine would produce a *hit density map* similar to the one showed in the figure below. This was done using a quite simple signal threshold cutting algorithm (ref. section

5.4). The result was a very reliable source of information regarding the beam position in the modules.

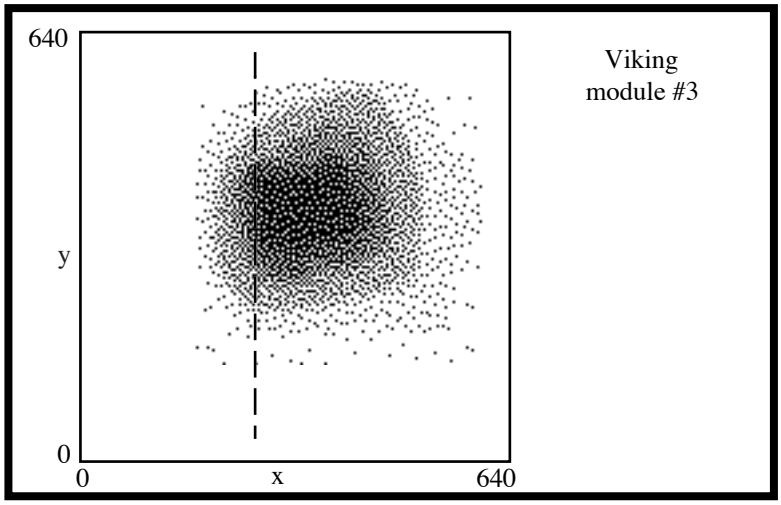


Figure 7.3: Enhanced Viking hit map, showing hits over time.

In this way we could establish that the beam was hitting the Felix module properly. The offline analysis told us eventually that signals could be seen in the good Felix. Thereafter we did enough runs to obtain some statistics without the magnetic field.
The next step after that was to turn on the magnet, re-align the modules to compensate for the beam bending in the magnetic field, and take new data.

7.4: Later Felix Testbeam Results.

The results from the KEK testbeam were not as good as hoped for. Later, after I left the project to concentrate on the writing of this thesis, it was discovered that the new 32-channel version of the Felix chip was very sensitive to fine-tuning of the control input-signals. Also, the data-taking period at KEK became too short.

Several other testbeam-runs have been performed since the KEK testbeam, yielding good statistics with the improved 128-channels version of the Felix chip, and using the same testbeam setup as did the testbeam I have discussed in this thesis.

I will in this section quote some results achieved in the testbeam of November 1995. The results show that the 128-channel Felix chip today meets the ATLAS silicon detector specifications.

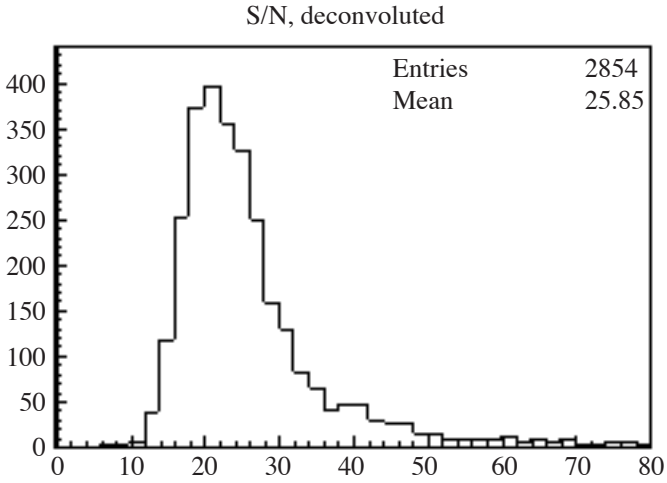


Figure 7.4: Histogram showing signal over noise for Felix connected to 6 cm modules[12].

The figure shows results directly comparable to the results from the –94 testbeam. As can be seen, a signal over noise of 21 was achieved with 6 cm modules in deconvoluted mode. This was performed with a new version of the Felix. The chip was equipped with 128 channels, and was further optimized in terms of noise performance. The testbeam performed experiments with 12 cm modules, as well, and the results were good:

- Signal over noise: 18
- Resolution σ: 7.8 μm
- Efficiency: 99.4%
- ENC: 1420 e^-
- Noise rate; 9.9×10^{-5}

These results are all within the ATLAS specifications that were quoted at the end of chapter 5, and the noise-values obtained are very close to the theoretical limit.

Now it should finally be possible, by recollecting some of the statements made throughout this thesis, to understand why the lenght of the Silicon modules will be 12 cm. In section 2.8 I said that the manufacturing technology restricts the silicon wafers to a maximum of $6 \cdot 6$ cm. That means that modules longer than 12 cm have to be made from three or more wafers. From section 5.8 we have that the specifications limit the noise to 1500 electrons or less. And the latest result is a noise of 1420 electrons. In section 6.1.1 the minimum noise slope vs input capacitance was calculated to 64 e^-/pF. The strip capacitance per cm depends on strip pitch, strip thickness and wafer thickness, but is approximately 1.2 pF per cm. From this it follows that if we add 1 cm to the lenght of the modules, it will result in a noise-increase of minimum $64 \cdot 1.2 = 77$ electrons, which will make the total noise very near 1500. If 2 cm is added, the limit is exceeded. To be able to use longer modules, one could try to decrease the strip capacitance per cm, or decrease the leakage current, both which would decrease the noise. Neither of the two are easy to

accomplish. As one would have to use three wafers to extend the modules beyond 12 cm, it would mean a leap in cost and complexity, and it would probably not be worth doing for a possible increase of only 1–2 cm. That is the main reason, since leakage current will increase with time, as mentioned in section 3.4.2.That will make the total noise exceed the 1500 e^- limit, but the modules will still be useable. So one would probably use modules of 14 cm, had it not been for the fact that three wafers would have to be used. It would be another story if noise around 1500 electrons could be achieved with 18 cm long modules.

8: Conclusion and Prospects

Planning, designing and building a detector like the ATLAS is a huge project. Only by international cooperation can there be hope of pulling such a task through. Even with the best of planning and the joint forces of 120 groups and some 1500 people working on the ATLAS detector, the proposed first-time operation is 2004.

A keyword to describe the process of planning and designing the ATLAS detector is *convergence*. 2–4 of the groups participating in the ATLAS project usually participated in each sub-project. The individual group would then propose a solution to the sub-project, and start developing that solution towards workable equipment. At regular intervals, the sub-project would be evaluated by coordinating groups, and at some level a decision would be made as to which sub-project solution would be developed further, with the goal to use that solution in the ATLAS detector. The building of ATLAS is a true dynamical process, in that each such decision is based on a number of factors. This includes compatibility with other solutions, solutions which in turn are chosen to be compatible with another solution, etc. Another factor is cost, which may vary due to technological progress, the availability of materials etc. And of course, technological progress in itself is a factor, making new solutions available, thus making the decisions even harder.

It should from this be clear that it is no easy task to build a machine like ATLAS, even having a reasonable overview of it is a formidable task! However, by breaking it into smaller projects and spread those projects out to many groups from many countries, while keeping the project firmly coordinated from CERN, the hope is that all the proposals should converge into solutions which will fit each other and which will make a very good detector.

There are first and foremost two components of the system I have looked into in this thesis: The silicon transducers and the read-out chip. Both belong to the Front-End Electronics of the Silicon Detector System of the Inner Detector. As I have stressed several places, the solution for these two components are not yet chosen, when I have discussed silicon microstrips for transducers and the Felix circuit for read-out, it has been because those were the favoured options in the testbeams I have worked with, and also as likely candidates for the ATLAS detector as anything else. Besides, both solutions incorporates certain general concepts, which have to be obeyed in any read-out scheme. I hope I in this thesis have made it clear what those concepts are, and that I have managed to communicate how the two components I have discussed attempts to solve the challenges present within the concepts.

The form of the work I undertook during the master-degree study is reflected in this thesis. This is not a paper which discusses a specific problem, and attempts to solve that problem. This is not a lab-report as such. Rather, it is an attempt to show that I have gained an understanding of the components and demands of the ATLAS Silicon Detector Read Out Electronics, *and* that I have gained an understanding of the process of planning, designing and building such an electronics system. Hopefully I have also managed to contributed to the solving of some of the problems encountered in building the ATLAS Silicon Detector System.

Appendix: The *Runseq* program sourcecode.

The Felix chip is a rather complex system, and needs several controlling inputs. The different input signals control the ADB, the APSP and the MUX. These signals are provided by a programmable high-speed pulse generator, called a sequencer. The Sequencer used for Felix was custom-made by Rutherford Appleton Laboratory (RAL). The sequencer is controlled by the Runseq program. Runseq loads the Sequencer with its default and trigger sequence, as well as setting its speed. The program has been refined several time. Prior to the testbeam of summer -94, Runseq was heavily rewritten by me to accommodate reading the trigger sequence from a file. This instead of the inflexible solution to have it hardcoded into the program sourcecode.

This appendix lists the sourcecode for Runseq. The great majority of comments are by me, added during the development.

```
#include <stdio.h>
#include <time.h>
#include <math.h>

/*  Defines the different channels */
#define CLOCK_DRIVERS_ENABLE 0x8000000
#define SCOPE_TRIG 0x40000000
#define JUMP 0x80000000
#define T1 0x00000000
#define RESET 0x00000020
#define T1_BAR 0x00000002
#define TRIG 0x00000100
#define TDC_PULSE 0x00020000
#define HOLD 0x00000000
#define HOLD_BAR 0x00000010
#define CLK 0x00001000
#define CLK_BAR 0x00000000
#define RB 0x00000000
#define RB_BAR 0x00000001
#define RESET_BAR 0x00000000

long *big_buff_ptr;
short  *CLOCK_CONTROL,  *MEMORY_DATA_REG,  *JUMP_ADDRESS_REG,
*MEMORY_ADDRESS_CTR;
short *MEM_LOW_WORD, *MEM_HIGH_WORD, *TRIGGER_CTR;
short *POLARITY_CONTROL, *DIRECT, *SIGNAL_CONTROL;
short  *INTERRUPT_ADDR_REG,*STORE;
int T1_OFFSET,SEQ_END,INTERRUPT_START,INTERRUPT_END,SEQ_LENGTH;
int INTERRUPT_LENGTH,T1_LENGTH,RESET_OFFSET,VALID_CHAR,RESET_LENGTH;
int HOLD_OFFSET,CLK_OFFSET,RB_OFFSET,NUM_CLKS,MUX_CYCLE;
```

```
int HOLD_LENGTH,RB_LENGTH,PFIRST=1;

short *MEMORY,*TDC_BASE,*VERSION,*MANUFACT;
short *RESET_REG,*INTERRUPT_REG;
short *CONTROL_REG,*RANGE_REG,*THR_HI_REG,*THR_LO_REG;
short *OP_BUFFER;
int FIRSTLENGTH, ADC_MEM;

void *addr;
u_int32 size, perm;
process_id ppid;

main()
{
        short i;
        long li;
        int mod_length,freq,c,iev,iev2,sct, out=0;
        short a,b,test,FIFO_empty,mult,ev_ctr,buffer_info;
        short header;
        short ch_num,ch_data;
        float range,times;
        int idel,BASE1,BASE2;
        char seqintr[20], seqclock[20];

/* Sequencer setup */

       addr=(void *)0xfe0f0000;
      size=200;
      perm=3;
       _os_permit(addr, size, perm, ppid);

        MEM_LOW_WORD = (short *)(0xfe0f0000);
        BASE1 = (int)MEM_LOW_WORD;
        MEM_HIGH_WORD = (short *)(BASE1+0x2);
        JUMP_ADDRESS_REG = (short *)(BASE1 +0x4);
        MEMORY_ADDRESS_CTR = (short *)(BASE1 +0x16);
        MEMORY_DATA_REG = (short *)(BASE1 +0x18);
        POLARITY_CONTROL = (short *)(BASE1 +0x0A);
        DIRECT = (short *)(BASE1 +0x0C);
        SIGNAL_CONTROL = (short *)(BASE1 +0x0E);
        CLOCK_CONTROL = (short *)(BASE1 +0x08);
        TRIGGER_CTR = (short *)(BASE1 +0x1C);
        INTERRUPT_ADDR_REG = (short*)(BASE1 +0x06);

/* The different TDC registers */
/*
        TDC_BASE = (short *)(0x00000000);*/ /*TDC base address, set by
                                                switch */
/*       BASE2 = (int)TDC_BASE;
        VERSION =  (short *)(BASE2+0xFE);
        MANUFACT = (short *)(BASE2+0xFC);
```

```
        RESET_REG= (short *)(BASE2+0x1C);
        INTERRUPT_REG= (short *)BASE2;
        CONTROL_REG= (short *)(BASE2+0x1A);
        RANGE_REG= (short *)(BASE2+0x14);
        THR_HI_REG= (short *)(BASE2+0x12);
        THR_LO_REG=(short *)(BASE2+0x10);
        OP_BUFFER=(short *)(BASE2+0x18);*/
        SEQ_END = 68;
        INTERRUPT_START = 70;
        INTERRUPT_END = 2900;
        SEQ_LENGTH = 2902;
        INTERRUPT_LENGTH = 2830;
        range=770.0;
        freq=2;

        big_buff_ptr = (long*)calloc(0x7ff0,4);
        if(big_buff_ptr == NULL)
        {
                printf("big_buff_ptr 0\n");
                exit(0);
        }

        get_filename("interrupt",seqintr);  /*Ask the user for sequence-file */

        getchar(); /* Chews the newline- character */
        nullify_big_buff(); /* Sweeps big_buffer.  Probably superfluious..eh
                                        somebody correct this word */
        define_clockseq();

        read_sequence(seqintr, INTERRUPT_START);
        load_sequence();
        printf("return from load_seq\n");
/*      test_memory(); */
        printf("return from test_mem\n");
        start_sequence(freq);
        printf("Sequence started, clocks on.\n");
        printf("f & b moves offset, r reads in new interrupt- sequence, e quits.\n");
        printf("First signalchange occurs after %d timeunits.\n", FIRSTLENGTH);
        mod_length=FIRSTLENGTH; /* Safekeeps the offset at program- launch */
        while(out!=1)
        {  /* The user- interface.  Not excactly Macintosh... */
                printf("Run_seq > ");
                c = getchar();

                  if(c=='b')
                  {

                        modify_sequence(seqintr,-1);
                        nullify_big_buff();
                        define_clockseq();
                        read_sequence(seqintr, INTERRUPT_START);
```

```
                load_sequence();
                start_sequence(freq);
                printf("First signalchange occurs after ");
                printf("%d timeunits.\n", --mod_length);
        }
        else if(c=='f')
        {
                modify_sequence(seqintr,1);
                nullify_big_buff();
                define_clockseq();
                read_sequence(seqintr, INTERRUPT_START);
                load_sequence();
                start_sequence(freq);
                printf("First signalchange occurs after ");
                printf(" %d timeunits.\n", ++mod_length);
        }
        else if(c=='e')
                {
                        printf("\n");
                        out=1;
                }
        else if(c=='r')
        {
        /*      getchar();*/
                PFIRST=1;
                pkeep_offset(mod_length, seqintr);
                get_filename("interrupt", seqintr);
                nullify_big_buff();
                define_clockseq();
                read_sequence(seqintr, INTERRUPT_START);
                mod_length=FIRSTLENGTH;
                printf("First signalchange occurs after %d timeunits.\n",
mod_length);
                load_sequence();
                start_sequence(freq);
        }
        else
                {
                        printf("\nInvalid command.\n");
                        printf("f & b moves offset, r reads in new interrupt-
sequence, e quits.\n");
                }
        getchar(); /* Swallows newline */
    }
    pkeep_offset(mod_length, seqintr );
    printf("\nGoodbye\n");
    exit(0);
}

/******************     SUB_  ROUTINES     **********************/
```

```
/* Asks the user at program- exit whether the new offset is to be kept
or disregarded */
pkeep_offset(offset_length, seqintr)

char seqintr[20]; int offset_length;
{
        char c;
        if(offset_length!=FIRSTLENGTH)
        {
                while(c!='y' && c!='n')
                {
                        printf("Keep the modified offset [y/n]? ");
                        c=getchar();
                        if(c=='y')
                                printf("The file \"%s\" now reflects the new offset.\n",
seqintr);
                        else if(c=='n') /* Note that since the sequence file is modified
                                                when the offset is changed, is it necessary
                                                to change
                                                it back if the new offset is to be
                                                disregarded.  If the
                                                program exits abnormally, the file will
                                                contain the
                                                modified offset. */
                                modify_sequence(seqintr, FIRSTLENGTH-offset_length);
                        else
                                printf("Please answer y or n;\n");
                        getchar();
                }
        }
}

/* Loads big_buff into memory at the (hopefully) rigth address */

load_sequence()
{
        int i;
        short a,b;
        set_clock(0);
        printf("ret from set_clk\n");
        *JUMP_ADDRESS_REG=0;
        *MEMORY_ADDRESS_CTR=0;
        *MEMORY_DATA_REG=0;
        *DIRECT=0;
        *INTERRUPT_ADDR_REG=0;
        *MEMORY_ADDRESS_CTR=0;
        printf("Start to load sequence\n");
        for(i=0;i<SEQ_LENGTH;i++)
        {
                a = (short) (big_buff_ptr[i] & 0xffff);
                b = (short) ((big_buff_ptr[i] >>16) & 0xffff);
```

```
                *MEM_LOW_WORD= a ;
                *MEM_HIGH_WORD= b ;
        }
        printf("Finished loading\n");
        *JUMP_ADDRESS_REG=0;
        *MEMORY_ADDRESS_CTR=0;
        *MEMORY_DATA_REG=0;

}

get_filename(seqname, seqintr)
char seqname[10], seqintr[20];
{
        FILE *fp;
/*        printf("hei!\n"); */
        printf("File to define %s sequence: ", seqname);
        scanf("%s", seqintr);

}

start_sequence(clock)
int clock;
{
        *POLARITY_CONTROL=0;
        *SIGNAL_CONTROL=0;
        *JUMP_ADDRESS_REG=0;
        *MEMORY_ADDRESS_CTR=0;
        *MEMORY_DATA_REG=0;
        *INTERRUPT_ADDR_REG=INTERRUPT_START;
        set_clock(clock);
}

nullify_big_buff()
{
        int i;
        for(i=0; i< SEQ_LENGTH; i++)
                big_buff_ptr[i]=0;
}

set_clock(value)
int value;
{
        switch (value)
        {
                case 0: *CLOCK_CONTROL=0xFFF0;
                break;
                case 1: *CLOCK_CONTROL=0;
                break;
                case 2: *CLOCK_CONTROL=1;
                break;
                case 3: *CLOCK_CONTROL=2;
```

```
                break;
                case 4: *CLOCK_CONTROL=3;
                break;
                case 5: *CLOCK_CONTROL=4;
                break;
                default:        *CLOCK_CONTROL=5;
        }
}

init_tdc()
{
        float range;

        printf("Checking TDC:\n");
        printf("VERSION: %8lX\n",*VERSION);
        printf("MANUFACTURER: %8lX\n",*MANUFACT);
        *CONTROL_REG=0x1;
        *RANGE_REG=0xE0;
        range=770.0;
        *THR_HI_REG=0xA0;
        *THR_LO_REG=0x00;
}

/* Defines the states of all the channel during the clock- sequence.
Not very general! */

define_clockseq()
{
        int i;

        for(i=0; i<INTERRUPT_START; i++)
        {
                big_buff_ptr[i]=CLOCK_DRIVERS_ENABLE;
                big_buff_ptr[i]|=RESET_BAR;
                big_buff_ptr[i]|=T1_BAR;
                big_buff_ptr[i]|=HOLD;
                big_buff_ptr[i]|=CLK_BAR;
                big_buff_ptr[i]|=RB_BAR;
                big_buff_ptr[i]|=0x2000;
           big_buff_ptr[i]|=TDC_PULSE; /* defines state of ch. 18 */
           }
            big_buff_ptr[SEQ_END]|= JUMP;

}

/* Reads the sequencefile file and fills big_buff from address according
to that file */
read_sequence(file, address)
char file[20]; int address;
{
        int a,i,j,blocks=0,offset,length,width,offset_old=0,length_old=0,buffline=0;
```

```
        char bitpat[32];
        FILE *fp;
        if ((fp = fopen(file,"r"))==NULL)
        {
                printf("Can't open file \"%s\".\n", file);
                exit(1);
        }
        for(i=0; i<SEQ_LENGTH-address; i++)
        {
             big_buff_ptr[address+i]|=TDC_PULSE; /* defines state of ch. 18 */
               big_buff_ptr[address+i]=CLOCK_DRIVERS_ENABLE; /*Every timeunit
                                                              must have this */
        }
        fscanf(fp,"%*s%d", &blocks); /* Number of blocks must be correct! */
        for (a=0; a<blocks; a++)
        {
                fscanf(fp,"%*s%d%*s%d%*s%d", &offset, &length, &width);
                if(offset_old+length_old!=offset)
                {
                        printf("****WARNING:  There is an inconsistency between
                                block  %d and %d\n",a, a+1);
                        printf("      in the file %s.  Moving the offset will remove the
                                inconsistency,\n", file);
                        printf("      by assuming that the _lengths_ are correct.  Exit
                                now and edit the file\n");
                        printf("      if that's not what you want. ****\n");
                }
                offset_old=offset; length_old=length;
                if(PFIRST==1) { FIRSTLENGTH=length; PFIRST=0; } /* read_sequence
                                                    is called every time offset is
                                                    modefied.  This statement assures
                                                    that the offset as the program
                                                    is launched is kept for
                                                    reference (see pkeep_offset) */
                for (i=0; i<width; i++) /* Scans in the bitpattern from file and
                                           modefies big_buff. No check for EOF is
                                           performed. File- format must be
                                           correct!*/
                {
                        fscanf(fp, "%s", bitpat);
                        for (j=0; j<32; j++)
                                if ( bitpat[j]=='1' )
                                        big_buff_ptr[address+buffline]|=
                                        power(2,j);
                        buffline++;
                }
                if ( length/width > 1 )
                        for (i=0; i<width; i++)
                                for (j=1; j<(length/width); j++)
                                        big_buff_ptr[address+buffline-
width+i+(j*width)]=
```

```
                                        big_buff_ptr[address+buffline-width+i];
                buffline=buffline+length-width;
        }
        fclose(fp);
        big_buff_ptr[address]|=TRIG;
        big_buff_ptr[address] &= 0xFFFDFFFF; /* mask out TDC_PULSE */
        /* big_buff_ptr[address]|=TDC_PULSE; */
        big_buff_ptr[INTERRUPT_END]|=JUMP;
}

power(x, y) /* Need this, lib. func. pow() can't handle zero exponent */
int x, y;
{
        int i, temp=1;
        for(i=y;  i>0;  i--)
                temp=temp*x;
        return temp;
}

/* offset is moved deltaoffset units, actually modifies the sequence-
file filename, and reads it back in */

modify_sequence(filename, deltaoffset)
char filename[16]; int deltaoffset;
{
        char bitpat[32];
        int blocks,tegn, offset_new, length_new, offset_old, length_old, width, a, i, j;
        FILE *fp, *fp1;

        printf("\n File  %s\n",filename);
        if((fp=fopen(filename,"r"))==NULL) /* Opens the sequence-file for reading
* /
        {
                printf("Can't open file \"%s\".\n", filename);
                exit(1);
        }
        if((fp1=fopen("mod_seq.temp", "w"))==NULL) /* Creates a file for writing
                                                        modified sequence */
        {
                printf("Couldn't create the necessary file.\n");
                exit(1);
        }       /* Scans the seq.file and writes modified data into mod_seq file */
        fscanf(fp, "%*s%d", &blocks);
        fprintf(fp1, "blocks= %d\n", blocks);
        fscanf(fp, "%*s%d%*s%d%*s%d", &offset_old, &length_old, &width);
        fprintf(fp1, "offset= %d length= %d ", offset_old, length_old+deltaoffset);
        fprintf(fp1, "width= %d\n", width);
        for(a=0; a<blocks-1; a++)
        {
                for(i=0; i<width; i++)
                {
```

```
                fscanf(fp, "%s", bitpat);
                fprintf(fp1, "%s\n", bitpat);
        }
        fscanf(fp, "%*s%d%*s%d%*s%d", &offset_new, &length_new,
                &width);
        if(offset_new!=offset_old+length_old)
                printf("Fixed a problem in block %d in the file
                %s.\n",a+2,filename);
        if(a!=blocks-2)
                fprintf(fp1, "offset= %d length= %d ",
                offset_old+length_old+deltaoffset, length_new);
        else
                fprintf(fp1, "offset= %d length= %d ",
                offset_old+length_old+deltaoffset,
                length_new+offset_new-offset_old-length_old);
        fprintf(fp1, "width= %d\n", width);
        offset_old=offset_old+length_old;
        length_old=length_new;
    }
for(i=0; i<width; i++)
{
    fscanf(fp, "%s", bitpat);
    fprintf(fp1, "%s\n", bitpat);
}

fclose(fp); fclose(fp1);

fp=fopen(filename, "w"); fp1=fopen("mod_seq.temp", "r");

while((tegn=getc(fp1)) != EOF)
    putc(tegn, fp);
        /* Overwrites seq.file with contents of mod_seq file */
fclose(fp); fclose(fp1);
}
test_sequence()
{
    int i,k;
    for(i=0;i<SEQ_LENGTH;i++)
    {
        big_buff_ptr[i]=CLOCK_DRIVERS_ENABLE;
    }
    for(i=0;i<INTERRUPT_LENGTH;i=i+10)
    {
        k=INTERRUPT_START+i;
        big_buff_ptr[k]|=RB;
/******      big_buff_ptr[(k+1)]|=RB_BAR;*/
        big_buff_ptr[(k+1)]|=CLK;
/******      big_buff_ptr[(k+3)]|=CLK_BAR;*/
        big_buff_ptr[(k+2)]|=1;
/******      big_buff_ptr[(k+5)]|=HOLD_BAR;*/
        big_buff_ptr[(k+3)]|=RESET;
```

```
/******          big_buff_ptr[(k+7)]|=T1_BAR;*/
                big_buff_ptr[(k+4)]|=RESET_BAR;
                big_buff_ptr[(k+5)]|=T1;
        }
        big_buff_ptr[SEQ_END]|=JUMP;
        big_buff_ptr[INTERRUPT_END]|=JUMP;
}
test_memory()
{
        long  li;
        short  i,a,b,errct=0;

        *JUMP_ADDRESS_REG=0;
        *MEMORY_ADDRESS_CTR=0;
        *MEMORY_DATA_REG=0;
        for(i=0;i<SEQ_LENGTH;i++)
        {
                a = (*MEM_LOW_WORD & 0xFFFF);
                b = (*MEM_HIGH_WORD & 0xFFFF);
                li=((long)b<<16)|a;
                if(li!=big_buff_ptr[i])
                {
                        errct++;
                        if(errct<200)
                        {
                                printf("li:%8lX   %8lX   %4d\n",li,big_buff_ptr[i],i);
                        }
                }
        }
        printf("test memory: errors = %d\n",errct);
}
```

References

[1] Peter WeilHammer: A series of lectures on semiconducors

[2] G. Lutz and R.H. Richter: Some comments on Si-strip detectors and reaout schemes

[3] S. Roe and P. Weilhammer: Feasibility of Si strip detectors with capacitive charge division and deconvolution front-end electronics in the ATLAS Si tracker

[4] ATLAS Trigger-DAQ Steering Group: Trigger & DAQ Interfaces with Front-End Systems: Requirement Document

[5] B.R Martin and G. Shaw: Particle Physics

[6] Jacob Millman and Arvin Grabel: Microelectronics, Second Edition

[7] Paul Horowitz and Winfield Hill: The Art of Electronics, Second Edition

[8] P. Weilhammer: Full Analog Readout Architecture for the ATLAS SCT

[9] W.W. Armstrong et al.: ATLAS Thechnical Proposal: CALORIMETRY, ftp://www.cern.ch/pub/Atlas/TP/NEW/HTML/tp9new/node16.html

[10] The LHC Study Group: Machine Performance, http://www.cern.ch/CERN/LHC/YellowBook95/LHC95/node4.html

[11] http://atlasinfo.cern.ch/Atlas/ATLASFIGS/ATLASPIC/ATLAS.PS

[12] Shaun Roe: The Felix Chip Performance, http://wwwcn.cern.ch/~sroe/felix/Performance

[13] http://atlasinfo.cern.ch/Atlas/ENGINEER/detector/IT-GEOMETRY-K.PS

[14] P. Allport et al.: Performance of Analogue readout Silicon Strip detectors for ATLAS