Stony Brook University

Design of an FPGA Based Semiconductor Detector Control System for Synchrotron Radiation Powder Diffraction Studies

A Thesis Presented

by

Mohammad Atiquar Rahman

to

The Graduate School

In Partial Fulfillment of the

Requirements

for the Degree of

Master of Arts

In

Physics

Stony Brook University

May 2008

Stony Brook University

The Graduate School

Mohammad Atiquar Rahman

We, the thesis committee for the above candidate for the

Master of Arts degree, hereby recommend

acceptance of this thesis.

D. Peter Siddons – Thesis Advisor
Physicist, National Synchrotron Light Source, Brookhaven National
Laboratory

Tomas Weinacht
Professor, Department of Physics

Chris Jacobsen
Professor, Department of Physics

This thesis is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Thesis

Design of an FPGA Based Semiconductor Detector Control System for Synchrotron Radiation Powder Diffraction Studies

By

Mohammad Atiquar Rahman

Master of Arts

In

Physics

Stony Brook University

2008

This work introduces a new powder diffraction instrument for synchrotron radiation studies and describes the development an FPGA based control system for the silicon strip detector for that instrument. The device is basically a Guinier Camera with Rowland Circle geometry. The strip detector is based on a multi-element Si sensor and has a 32 channel dedicated readout application specific integrated circuit (ASIC) called HERMES3 developed at the Instrumentation division of Brookhaven National Laboratory. The control system is designed on a Xilinx Virtex-4 field programmable gate array (FPGA) with an embedded PowerPC processor. Three intellectual property (IP) cores based on on-chip-peripheral (OPB) bus have been devised to send, receive and generate a timebase respectively. A tail pulse generator was used to verify that the designed cores are communicating with the ASIC properly. Also, some studies for calibrating the individual ASIC channel threshold voltage were also performed.

Table of Contents

List of Figures:

Chapter 1

This 1ˢᵗ chapter gives an introduction to the history and background of the synchrotron radiation with a brief theory behind the synchrotron light generation.

Overview of synchrotron radiation

1.1 Introduction to Synchrotron radiation

Synchrotron radiation is the light radiated by an electric charge following a curved trajectory *i.e.* a charged particle under the influence of a magnetic field. When charged particles, in particular electrons or positrons, are forced to move in a circular orbit, photons are emitted. At relativistic velocities, these photons are emitted in a narrow cone in the forward direction, at a tangent to the orbit. In a high energy electron or positron storage ring these photons are emitted with energies ranging from infra-red to energetic (short wavelength) X-rays. This radiation is called Synchrotron Radiation.

1.2 Short history

In 1947, a staff member in the laboratory of Professor Pollock, observed radiation emitted by electrons as the moved circularly in the magnetic field chamber of a cyclic accelerator – a synchrotron that accelerated electrons up to 70 MeV [1]. The radiation was observed as a bright luminous patch on the background of the synchrotron chamber. In this way, 'electronic light' was experimentally seen for the first time as radiation emitted by relativistic electrons having a large centripetal acceleration. This radiation was called synchrotron radiation as it was observed for the first time in a synchrotron.
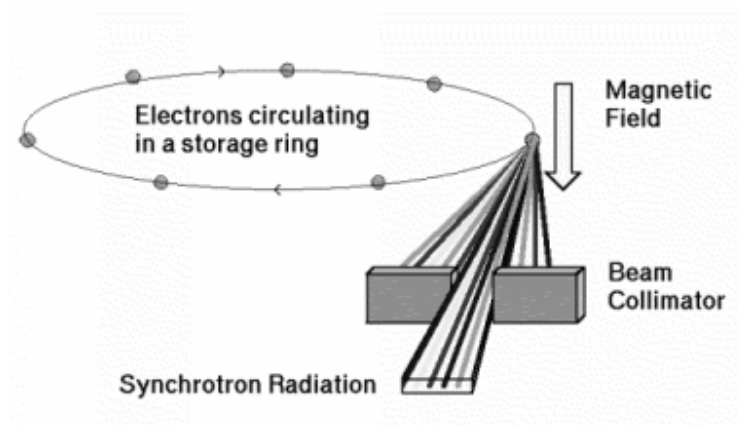
Figure 1.1: Generation of synchrotron radiation by circulating electrons in a circular path called storage ring [1].

The observation of synchrotron radiation was purely accidental. The cover of the chamber was removed to perform an adjustment and this allowed light to be seen outside the chamber [2].

But the idea of synchrotron light goes as far back as the 19th century. A French physicist Alfred Lienard of the Ecole des Mines in Paris described the concept of retarded potentials in the calculation of the effects due to the motion of charged particles, and worked out a basic theory of what is now known as synchrotron radiation. This work was supplemented by Emil Wiechert, so the formalism is generally known as the Lienard-Wiechert potentials.

The next major development of synchrotron radiation theory came in 1908 from G. A. Schott in a prizewinning paper about mechanical reactions of electromagnetic radiation [3]. The idea of synchrotron radiation lay dormant for a few decades before synchrotron radiation was seen on April 24, 1947. From there, continuous advancement of synchrotrons occurred and the brightness of the light produced also kept increasing. Third generations synchrotrons are the most advanced ones today and the radiation produced by them is exceptionally brighter than the first seen radiation.

Figure 1.2: x-rays emerging from a synchrotron beam port excite nitrogen in the air resulting luminesceing blue – Image obtained from www.nsls.bnl.gov.

1.3 Brief theory of synchrotron radiation

Though the discovery and first observations of synchrotron radiation were unexpected, a number of theoretical studies on the emission of a relativistic accelerating electron had been carried out long before that initial experiment in 1947[2].

In 1920, A. Lienard and O. Heaviside extended the familiar Larmor formula for the plane power of a relativistic electron

$$W = -\frac{\partial E}{\partial t} = \frac{2e^2 \dot{v}^2}{3c^3} \tag{1.1}$$

to a high velocity particle. In modern notation it takes the following form

$$W = \frac{2}{3}\frac{e^2}{m^2c^3}\left(\frac{dp_\mu}{d\tau}\frac{dp_\mu}{d\tau}\right) = \frac{2}{3}\frac{e^2\gamma^6}{c}\left[\dot{\beta}^2 - \left[\beta\dot{\beta}\right]^2\right], \tag{1.2}$$

where p_μ is the four-dimensional impulse, $d\tau = dt/\gamma$ is the intrinsic time, $\gamma = E/mc^2$ and $\beta = v/c$.

Lienerd also determined the fast growth of losses in the energy of electron describing in a circle of radius R, which is proportional to the fourth power of the particle speed and is inversely proportional to the square of the radius of the path

$$W = \frac{2}{3}\frac{e^2c}{R^2}\beta^4\gamma^4 \ . \tag{1.3}$$

Synchrotron radiation is an example of electromagnetic radiation produced by centripetal acceleration (as opposed to 'bremmsstrahlung', which is produced by tangential acceleration). The wavelength of this radiation is a function of the energy of

3

the charged particles and the strength of the magnetic field bending the charged particles.

The spectrum of the radiation is continuous and is characterized by its critical wavelength, which divides the spectrum into two parts with equal power (half the power radiated above the critical wavelength and half below). The critical wavelength can be expressed as follows [4]

$$\lambda_c = \frac{4\pi E_0^3}{cBE^2} \quad ,$$

(1.4)

and this reduces to the following equation when the charged particles are electrons

$$\lambda_c = \frac{1.864353}{B[T].(E[GeV])^2} \quad .$$

(1.5)

Chapter 2

X-ray diffraction and measurements

This chapter provides an overview of x-ray diffraction theory and discusses various methods of diffraction techniques with a detailed study of the powder diffraction process.

2.1 Diffraction of X-rays

A crystal is a complex but orderly arrangement of atoms and all atoms in the path of an x-ray beam scatter x-rays simultaneously. In general, the scattered x-rays interfere, and in certain specific directions, where the scattered x-rays are "in-phase", the x-rays scatter cooperatively to form a new wave. This process of constructive interference is *diffraction* [7].

The diffraction of X-rays by a crystal was first confirmed by Max von Laue. He suggested the use of crystal to act as a grating for x-ray diffraction and showed that if a beam of X-rays passed through a crystal, diffraction would take place and a pattern would be formed on a photographic plate placed at a right angle to the direction of the rays. The pattern would mark out the symmetrical arrangements of the atoms in the crystal. This was verified experimentally verified in 1912 by two of his students under his supervision, which showed that a diffracting crystal intercepts X-rays of all wavelengths, but only those that undergo constructive interference are transmitted efficiently to the detector [7].
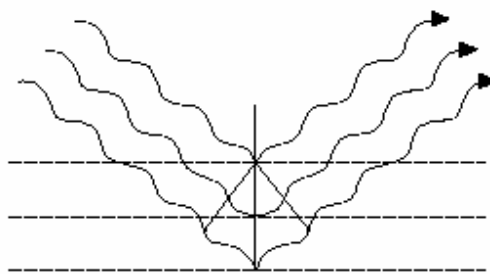
Figure 2.1: A representation of first-order x-ray diffraction. The path-length difference between X-rays diffracted from adjacent atomic layers corresponds to an integral number of wavelengths.

2.2   Bragg Diffraction

When a wave enters the crystal, some portion of it will be reflected by the first layer, while the rest will continue through to the second layer, where the process continues. By the definition of constructive interference, the separately reflected waves will remain in phase if the difference in the path length of each wave is equal to an integer multiple of the wavelength.
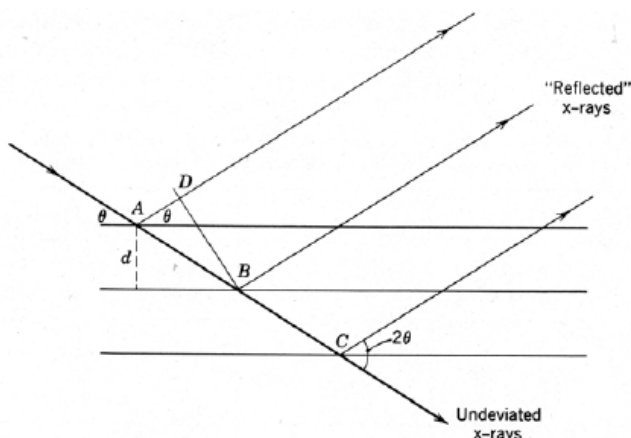
Figure 2.2: Bragg Law: The condition for diffraction [8].

Figure 2.2 shows a beam of parallel x-rays penetrating a stack of planes of spacing d, at a glancing angle of incidence, θ. Each plane is pictured as reflecting a portion of the incident beam. The "reflected" rays combine to form a diffracted beam if they differ in phase by a whole number of wavelength, that is, if the path difference $AB - AD = n\lambda$, where n is an integer. Therefore,

$$AB = \frac{d}{\sin\theta} \quad \text{and} \quad AD = AB\cos 2\theta = \frac{d}{\sin\theta}(\cos 2\theta) \ .$$

Hence,

$$n\lambda = \frac{d}{\sin\theta} - \frac{d}{\sin\theta}(\cos 2\theta)$$

$$= \frac{d}{\sin\theta}(1 - \cos 2\theta) = \frac{d}{\sin\theta}(2\sin^2\theta)$$

$$\therefore n\lambda = 2d\sin\theta \ . \tag{2.1}$$

This equation (2.1) is the Bragg law for diffraction [9].

2.3 Diffraction Theory with Fourier analysis

The mathematical process of connecting x-ray diffraction with the crystal structure is based on Fourier analysis. The spatial distribution of scattered x-rays is closely related to the Fourier transform of the crystal's electron density. In principle, an inverse Fourier transform can be used to directly convert experimental scattering data into a picture of the electron density (*i.e.* the atomic crystal structure). Of course, the Fourier Transform of the electron density is a complex-valued function, only the magnitude of which is actually measured, which is the amplitude squared of each coefficient. However, given just the intensity, without the phase information it's not possible to determine the Fourier Transform. Hence the "phase problem" of crystallography arises.

The primary goal of X-ray crystallography is to determine the density of electrons $f(\mathbf{r})$ throughout the crystal [10]. For this purpose, X-ray scattering is used to collect data about its Fourier transform $F(\mathbf{q})$, which is then inverted mathematically to obtain the density defined in real space, using the following formula

$$f(r) = \int \frac{dq}{(2\pi)^3} F(q) e^{iqr} \ , \tag{2.2}$$

and the corresponding Fourier transform is

$$F(q) = \int dr f(r) e^{-iqr} \ . \tag{2.3}$$

The vector $\mathbf{q}$ represents a point in reciprocal space, that is, to a particular oscillation in the electron density as one move in the direction in which $\mathbf{q}$ points. The length of $\mathbf{q}$ corresponds to 2π divided by the wavelength of the oscillation.

The Fourier transform $F(\mathbf{q})$ is generally a complex number, and therefore has a magnitude $|F(\mathbf{q})|$ and a phase $\varphi(\mathbf{q})$ related by the equation (2.4)

$$F(q) = |F(q)| e^{i\varphi(q)} \ . \tag{2.4}$$

As discussed above, the intensities of the reflections observed in X-ray diffraction give us the magnitudes $|F(\mathbf{q})|$ but not the phases $\varphi(\mathbf{q})$. To obtain the phases, full sets of reflections are collected with known alterations to the scattering, either by modulating the wavelength past a certain absorption edge or by adding strongly scattering (*i.e.* electron-dense) metal atoms such as mercury. Combining the magnitudes and phases yields the full Fourier transform $F(\mathbf{q})$, which may be inverted to obtain the electron density $f(\mathbf{r})$[10].

2.4   Diffraction Methods

Diffraction can occur whenever the Bragg law is satisfied. With monochromatic radiation, an arbitrary setting of a single crystal in a beam of x-rays will not in general produce any diffracted beams. Some way of satisfying the Bragg law must be devised, which can be done by continuously varying either wavelength (λ) or angle (θ) during the experiment. There are three primary diffraction methods that can be distinguished by their respective ways of variation of the wavelengths and incident angle.

2.4.1   Laue method

The Laue method is mainly used to determine the orientation of large single crystals [8]. White radiation is reflected from, or transmitted through, a fixed crystal. The diffracted beams form arrays of spots that lie on curves on the film. The Bragg angle (θ) is fixed for every set of planes in the crystal. Each set of planes picks out and diffracts the particular wavelength from the white radiation that satisfies the Bragg law for the values of d and q involved. Each curve therefore corresponds to a different wavelength. The spots lying on any one curve are reflections from planes belonging to one zone.
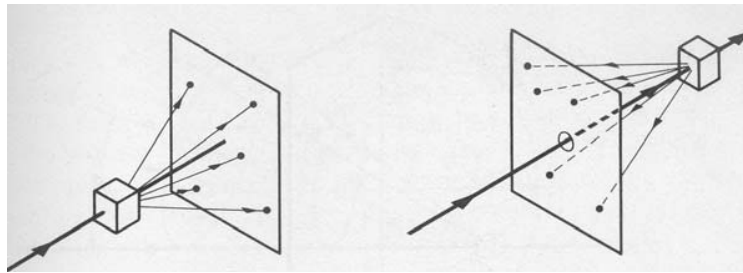
Figure 2.3: Transmission and back reflection Laue method [7].

There are two variants of the Laue method, the back-reflection and the transmission Laue. In the back-reflection method, the film is placed between the x-ray source and the crystal. The beams which are diffracted in a backward direction are recorded. One side of the cone of Laue reflections is defined by the transmitted beam. The film intersects the cone, with the diffraction spots generally lying on a hyperbola. In the transmission Laue method, the film is placed behind the crystal to record beams which are transmitted through the crystal. One side of the cone of Laue reflections is defined by the transmitted beam. The film intersects the cone, with the diffraction spots generally lying on an ellipse.

2.4.2 Rotating-crystal method

In the rotating crystal method, a single crystal is mounted with an axis normal to a monochromatic x-ray beam. A cylindrical film is placed around it and the crystal is rotated about the chosen axis. As the crystal rotates, sets of lattice planes will at some point make the correct Bragg angle for the monochromatic incident beam, and at that point a diffracted beam will be formed.
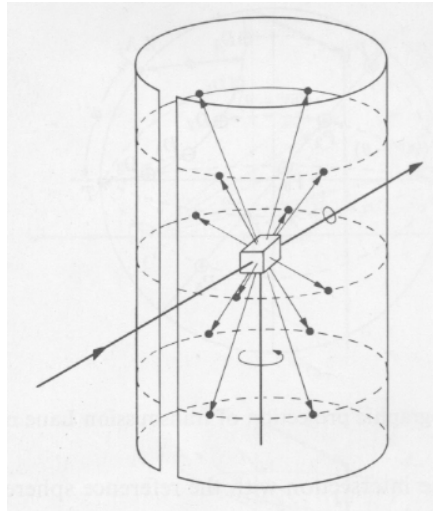
Figure 2.4: Rotating crystal method [7].

The reflected beams are located on the surface of imaginary cones. When the film is laid out flat, the diffraction spots lie on horizontal lines. The chief use of the rotating crystal method is in the determination of unknown crystal structures. Modern practice uses a flat electronic detector placed perpendicular to the incident beam, like the transmission Laue method. Then the zone lines are formed in conic sections [11].

2.4.3 Powder method

In the powder method, the crystal to be examined is reduced to a very fine powder and placed in a beam of monochromatic x-rays. Each particle of the powder is a tiny crystal, or assemblage of smaller crystals, oriented at random with respect to the incident beam. By probability, some of the crystals will be correctly oriented for (110) reflections, and so on. Other crystals will be correctly oriented for (110) reflections, and so on. The mass of powder is in fact, to a single crystal rotated, not about one axis but about all possible axes. When the scattered radiation is collected on a flat plate detector the rotational averaging leads to smooth diffraction rings around the beam axis rather than the discrete Laue spots as observed for single crystal diffraction [11]. The angle between the beam axis and the ring is called the *scattering angle* and in X-ray crystallography always denoted as 2θ. In accordance with Bragg's law, each ring corresponds to a particular reciprocal lattice vector in the sample crystal. This leads to the definition of the scattering vector as

$$q = 4\pi sin\theta/\lambda \,. \tag{2.5}$$
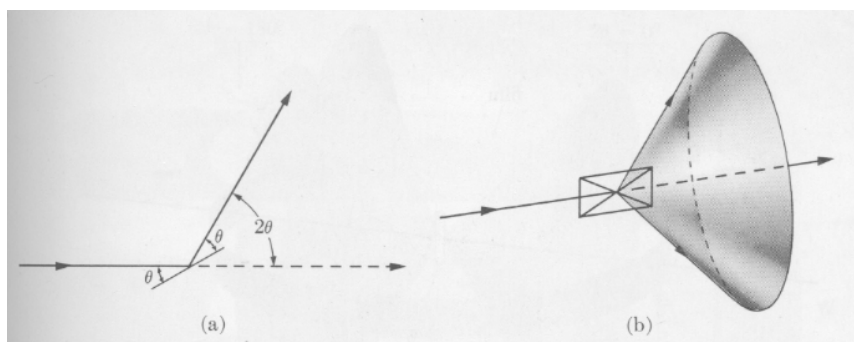
Figure 2.5: Formation of a diffracted cone of radiation in powder method [7].

Powder diffraction data are usually presented as a diffractogram in which the diffracted intensity I is shown as function either of the scattering angle 2θ or as a function of the scattering vector q. The latter variable has the advantage that the diffractogram no longer depends on the value of the wavelength λ. The advent of synchrotron sources has considerably broadened the choice of wavelength.

2.5 Applications of powder diffraction

Comparing to other methods of analysis, powder diffraction allows for rapid, non-destructive analysis of multi-component mixtures without the need for extensive sample preparation. This gives laboratories around the world the ability to quickly analyze unknown materials and perform materials characterization in such fields as metallurgy, mineralogy, forensic science, archeology and the biological and pharmaceutical sciences. Identification is performed by comparison of the diffraction pattern to a known standard. The major applications of powder diffraction method are pointed in the next sections [8].

2.5.1 Phase identification

The most widespread use of powder diffraction is in the identification and characterization of crystalline solids, each of which produces a distinctive diffraction pattern. Both the positions (corresponding to lattice spacing) and the relative intensity of the lines are indicative of a particular phase and material, providing a "fingerprint" for

comparison. A multi-phase mixture will show more than one pattern superposed, allowing for determination of relative concentration.

2.5.2 Crystal structure determination

In many cases, Rietveld refinement allows the full fitting of the intensity profile of a diffractogram to determine the crystal structures as well as other characteristics like preferred orientation (texture), crystallite size and -in the case of neutron patterns- even the magnetic structure, *i.e.* the spatial distribution of magnetic moments in the structure if the material shows magnetic order.

2.5.3 Crystallinity

Comparing to a crystalline pattern consisting of a series of sharp peaks, amorphous materials (liquids, glasses etc.) produce a broad background signal. Many polymers show semi-crystalline behavior, *i.e.* part of the material forms an ordered crystallite by folding of the molecule. One and the same molecule may well be folded into two different crystallites and thus form a tie between the two. The tie part is prevented from crystallizing. The result is that the crystallinity will never reach 100%. Powder XRD can be used to determine the crystallinity by comparing the integrated intensity of the background pattern to that of the sharp peaks.

2.5.4 Phase transitions

Powder diffraction can be combined with *in situ* temperature and pressure control. As these thermodynamic variables are changed, the observed diffraction peaks will migrate continuously to indicate higher or lower lattice spacing as the unit cell distorts. This allows for measurement of such quantities as the thermal expansion tensor and the isothermal bulk modulus, as well determination of the full equation of state of the material. At some critical set of conditions, a new arrangement of atoms or molecules may become stable, leading to a phase transition. At this point new diffraction peaks will appear or old ones disappear according to the new phase symmetry. This is what our detector is particularly useful for, due to its large angle range and rapid readout ability.

Chapter 3

This chapter gives an overview of the major powder diffraction methods and provides a detail description of our proposed diffraction instruments for synchrotron radiation sources.

Powder Diffraction techniques

3.1   Debye-Scherrer/ Hull method

The first powder diffraction camera was invented by Debye and Scherrer [12] in 1916 and Hull [13] in United States. Fig. 3.1 shows the original optical arrangements of a Debye-Scherrer camera. This instrument uses a pinhole collimeter (PC), which allows radiation from the course (S) to fall onto the rod-shaped sample (SP). A secondary collimeter (SC) or beam stop is placed behind the sample to reduce the secondary scatter of the beam by the body of the camera. A film (FM) is placed around the inside circumference of the camera body. Also, a β- filter is typically placed between the source and the specimen to partially monochromatize the incident beam [14]. Modern day Debye-Scherr instruments use position-sensitive bent detectors instead of cameras.
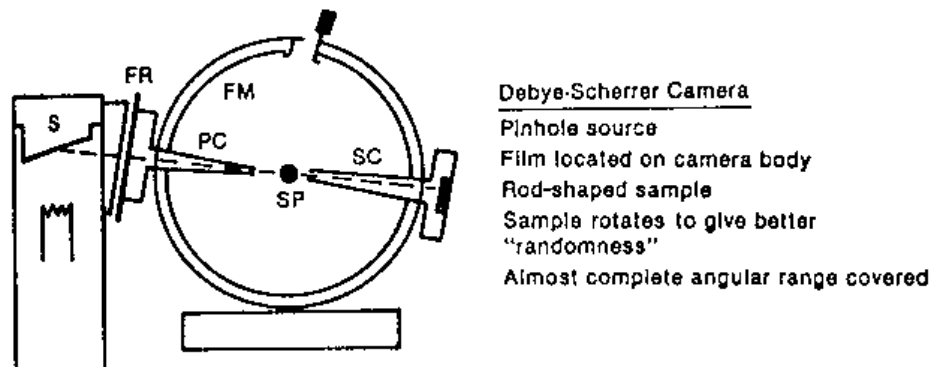
Figure 3.1: Initial Debye-Scherr camera setup [14].

In the Debye Scherrer camera, a narrow strip of film is curved in a short cylinder with the specimen placed on the axis and the incident beam is directed at right angles to the axis. The cones of diffracted radiation intersect the cylindrical strip in lines and when the strip is laid straight the resulting pattern is as shown in the figure 3.1. Each pattern is made up of small spots each from one particle and the spots are so close to each other that they appear as a continuous line. These lines are generally curved and when $2q = 90°$, they form a straight line. From the measured position of a given diffraction line, θ can be determined and if the wavelength λ is known, we can calculate the d-spacing of the lattice planes.
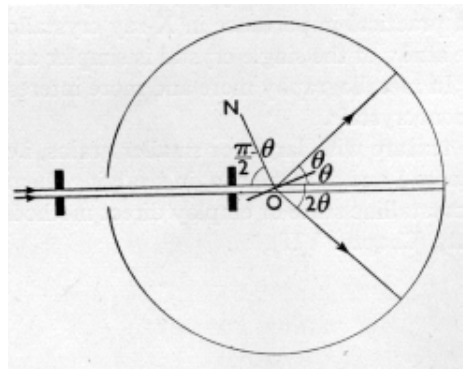
Figure 3.2: Principle of the Debye-Scherrer method [7].

As this setup utilizes position sensitive detection technique, the data collection is very fast and has a high throughput. But this method suffers from modest resolution.

3.2   Crystal Analyzer/ Diffractometer method

In this method a crystal is used as an angular filter to provide high angular resolution, and an electronic single-point detector is used instead of the film [15]. A collimated incident ray beam is used to give a good angular resolution. The powder is filled in the hole of a sample holder. The set up has minimum systematic errors if reflection is obtained when the beam is incident at an angle θ with the flat sample surface and the reflected beam is recorded at an angle of 2θ in what is referred to as θ-2θ scan. This is shown in Fig. 3.3. The peak positions and the intensities are readily obtained from the chart. A cylindrical sample can also be used.
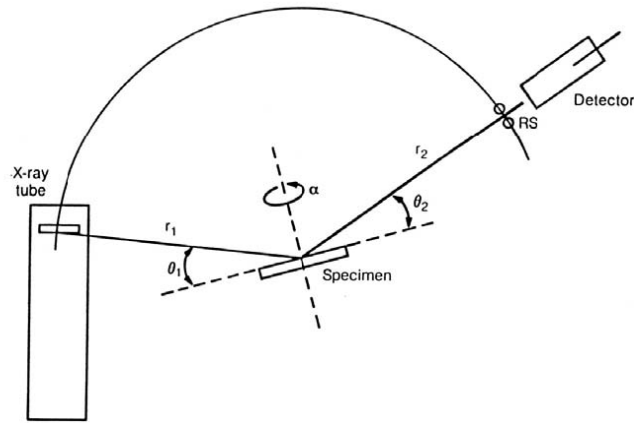
Figure 3.3: Common mechanical movements in powder diffractometers [15].

3.3   Our proposed diffraction instrument

It is evident from the above discussion that the Debye-Scherrer method has its advantages and disadvantages. Using the above mentioned techniques, beamlines with focusing optics typically use a position-sensitive detector and generally achieve lower resolution than the analyzer instrument, but with much higher throughput. The analyzer instrument achieves the best possible resolution, but is very slow and cannot provide the high throughput required for time-resolved studies. Our proposed powder diffraction instrument uses the Guinier geometry [16] along with Rowland circle mounting combines the high throughput of a position sensitive detector system with the high resolution of a crystal analyzer [17].

3.4   Guinier Camera

The combination of a focusing monochromator and a focusing camera is known as a Guinier camera, invented by Guinier in late 1930s [16]. Usually the focusing monochromator is best used with powder cameras especially designed to take advantage of the focusing action of the reflected beam. In Guinier camera (Fig. 3.5) a cylindrical camera (curved detector in our proposed system) is used with both the specimen and film arranged on the surface of the cylinder.
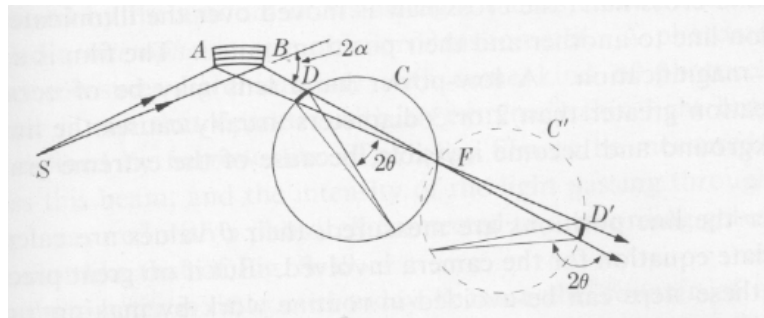
Figure 3.4: Guinier camera used with focusing monochromators. Only one diffracted beam is shown in each case [7].

Low angle reflections are recorded with camera placed in position C, while the high angle reflections are obtained by back reflections with camera in position C', shown dashed and the specimen at D'. In either case, the diffracted rays from the specimen are focused on the film for all 'hkl' reflections; the only requirement is that the film or detector be located on a circle passing through the specimen and point [16].

3.5   Rowland Circle

The Rowland circle is a geometric construction for a spherical concave grating with radius R, where the source and exit slit are pointed on an imaginary circle with diameter R so as to match both diffraction and focusing conditions. The lines of the grating are normal to the plane of the circle and the radius of the grating sphere passes through the center of the circle. An entrance slit positioned on the Rowland circle produces a focused spectrum on the Rowland circle [7].

16
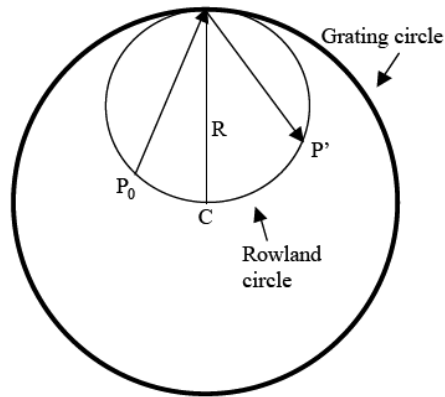
Figure 3.5: Rowland circle geometry [18].

The Rowland circle limits the loss of light that necessarily arises when diffracted rays are focused by means of lenses [15]. The Rowland circle identifies the correct locations of the entrance slit (P0) and the exit slit (P'). If P0 is anywhere on the circle whose diameter is equal to R, the radius of curvature of the grating, and which contacts the centre of the grating at O as shown in Fig. 3.7, then the specular beam and the dispersed beams in all orders will be focused at other points on the same circle.

To prove this, we will show that that P0 and P' satisfy the grating equation if they lie on the Rowland circle [18]. We start by considering figure 3.7, where the reflection point is in the middle of the grating. The coordinate origin is defined at O(0,0), so that P0(x0,z0) and P'(x', z') are identified by:

$$x_0 = -r\sin\theta \qquad\qquad z_0 = r\cos\theta \qquad\qquad (3.1)$$

$$x' = r'\sin\theta' \qquad\qquad z' = r'\cos\theta' \qquad\qquad (3.2)$$
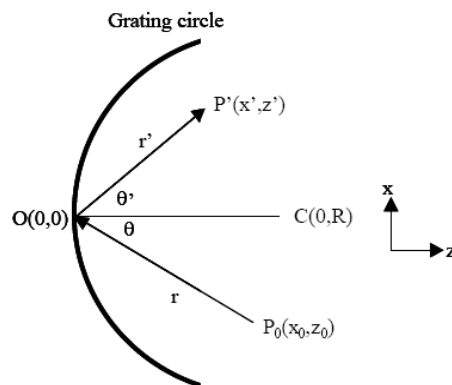
Figure 3.6: Concave geometry for O at the centre of the grating [18].

17

Next we consider a different position of reflection on the grating circle, namely P(x,z), as in Figure 3.8. Note that the angles q and q' are still being reference d to the line connecting C and the point of reflection on the grating.

The arc OP is required to contain an integral number of grooves that is approximately x/a, where a is the groove spacing (assuming that the arc length is small compared to R). If P' is to be at an interference maximum, then each groove contributes a beam whose optical path length as measured from P_0 differs from contributions due to adjacent grooves by a factor ml (for dispersion into the order m). The total optical path length difference between contributions from O an P is
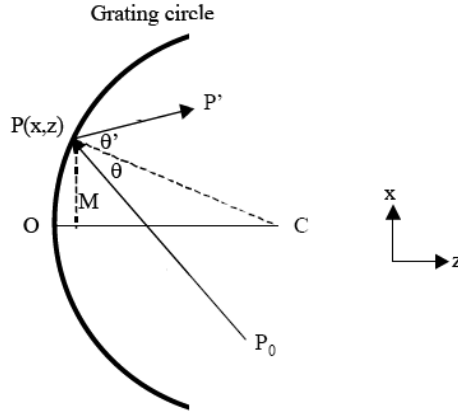
Figure 3.7: Reflection point P not in the middle of the grating [18].

$$\overline{P_0OP'} - \overline{P_0PP'} = \frac{m\lambda x}{a} \tag{3.3}$$

$$\overline{P_0OP'} = r + r' \tag{3.4}$$

$$\overline{P_0PP'} = \left[(x_0 - x)^2 + (z_0 - z)^2\right]^{1/2} + \left[(x' - x)^2 + (z' - z)^2\right]^{1/2}. \tag{3.5}$$

As the dimension of z is very small compared to the other dimensions in Figure 3.8, we can ignore the terms of z^2 when expanding equation (3.5); so that we have the approximation

$$\overline{P_0PP'} \approx \left[x_0^2 + z_0^2 - 2(xx_0 + zz_0) + x^2\right]^{1/2} + \left[x'^2 + z'^2 - 2(xx' - zz')^2 + x)^2\right]^{1/2}. \tag{3.6}$$

Equation (3.6) can be simplified even further by using a standard approximation to the

18

curvature of a spherical surface as given in [18] as

$$z \approx x^2 \big/ 2R \; .$$ (3.7)

By substituting equations (3.1), (3.2) and (3.7) into equation (3.6) leads to

$$\overline{P_0PP'} \approx \left\{ r^2 - 2\left[x(-r\sin\theta) + \frac{x^2}{2R}(r\cos\theta) \right] + x^2 \right\}^{1/2}$$
$$+ \left\{ r'^2 - 2\left[x(-r'\sin\theta') + \frac{x'^2}{2R}(r'\cos\theta') \right] + x^2 \right\}^{1/2}$$ (3.8)

$$\therefore \overline{P_0PP'} \approx r\left\{ 1 + \frac{2x\sin\theta}{r} - \frac{x^2\cos\theta}{rR} + \frac{x^2}{r^2} \right\}^{1/2} + r'\left\{ 1 + \frac{2x\sin\theta'}{r'} - \frac{x'^2\cos\theta'}{r'R} + \frac{x^2}{r'^2} \right\}^{1/2} . \quad (3.9)$$

We now use the fact that x/r and x/r' are small to simplify the square root operation. Because $(1+p)^{1/2} \approx 1 + p/2 - p^2/8 + ...$, we can state for our case that, for the leftmost term in equation (3.8) for example

$$p = \left(\frac{x}{r}\right)^2 + \frac{2x\sin\theta}{r} - \frac{x^2\cos\theta}{Rr} \; .$$ (3.10)

By expanding the leftmost term of equation (3.8), we can approximate it by

$$\overline{P_0P} \approx r\left\{ 1 + \frac{x^2}{2r^2} + \frac{x\sin\theta}{r} + \frac{x^2\cos\theta}{2Rr} + \frac{x^2\sin^2\theta}{2r^2} \right\}$$

$$\approx r + x\sin\theta + \frac{x^2}{2R}\cos^2\theta - \frac{x^2\cos\theta}{2R} \; .$$ (3.11)

If only the terms of second or lower order in x are retained and x/r is assumed small. Similar steps for the rightmost term of equation (3.8) leads to the approximation

$$\overline{P_0PP'} \approx (r+r') + x(\sin\theta - \sin\theta') + \frac{x^2}{2}\left[\left(\frac{\cos^2\theta}{r} - \frac{\cos\theta}{R} \right) + \left(\frac{\cos^2\theta'}{r'} - \frac{\cos\theta'}{R} \right) \right] . \quad (3.12)$$

Equation (3.12) can now be substituted in equation (3.3) (which expresses the conditions for an interference maximum at P′ given the source at P₀)

$$\overline{P_0OP'} - \overline{P_0PP'} \approx x(\sin\theta' - \sin\theta) + \frac{x^2}{2}\left[\left(\frac{\cos^2\theta}{r} - \frac{\cos\theta}{R} \right) + \left(\frac{\cos^2\theta'}{r'} - \frac{\cos\theta'}{R} \right) \right] = \frac{m\lambda x}{a} \; .$$

(3.13)

The first term on the left of equation (3.13) is the familiar form of the grating equation, while the second term will be zero (at this level of approximation) provided that

$$r = R\cos\theta \qquad\qquad (3.14)$$

and

$$r' = R\cos\theta' \ . \qquad\qquad (3.15)$$

Equations (3.14) and (3.15) imply that OP_0C and OP'C define right triangles, and because OC is common to both triangles, points P and P' must lie on a common circle whose diameter is OC, namely the Rowland circle.

3.6   *Guinier camera w/ Rowland circle configuration*

This instrument uses Guinier geometry which is traditionally used with an X-ray tube source. In this setting, a focusing monochromator will be used to focus the x-ray
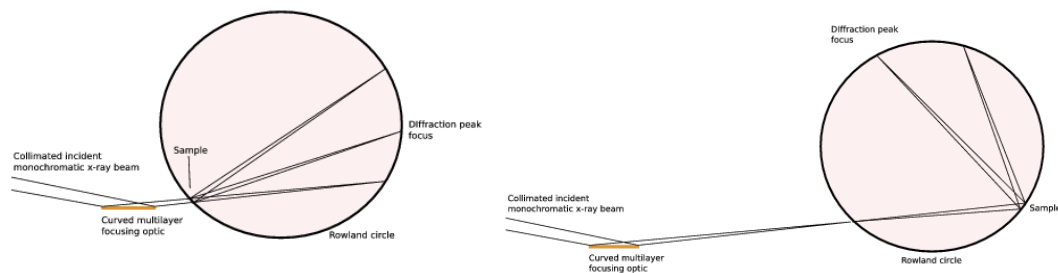
Figure 3.8: The Guinier geometry with Rowland Circle configuration has two forms, (L) is optimal for low Bragg angle peaks and operates on transmission, (R) is optimized for high Bragg angle peaks which operate on reflection [17].

beam. The sample and detector will be on the perimeter of a Rowland circle which means the radius of the curved powder crystal will be equal to the diameter of the circle. This will guarantee that that diffracted (for low Bragg angle) rays and back reflected rays (for high Bragg angle) will be focused on the detector as long as it's also on the perimeter of the Rowland circle. We propose to use a polygonal approximation to a circular silicon strip detector array to cover the appropriate range on the Rowland circle, and a flat sample. These approximations both introduce negligible errors for the parameters chosen. The instrument will have a curved multilayer optic, with good figure error and would be long enough to allow operation at higher energies without losing beam aperture. It would have a full detector array covering 90 degrees of Rowland circle (45 degrees in 2-theta) [17].

To realize the advantage of this geometry we need an efficient, low noise, high resolution position sensitive detector. Such a detector will be based on BNL's HERMES integrated circuit and silicon microstrip arrays [19]. This consists of a monolithic array of silicon diodes coupled to a set of application-specific integrated circuits designed at BNL. This allows each diode to detect x-rays with good energy resolution and hence low noise. The sensitive thickness of the diodes is 0.4mm, allowing efficient detection up to about 12keV. The HERMES detector will be discussed in detail in next chapter.

This instrument is primarily a strongly focusing device, identical to the classical Guinier geometry, but scaled up to allow angular resolutions at the 10^4 level or better. Its inherent resolution is determined by the size of the focal spot at the Rowland circle and the spatial resolution of the detector [17].

3.7 Focusing optic

To achieve high resolution, we need an optical system which provides a superior energy resolution, and at the same time provides a strongly demagnified image of the source at the detector. We would like the beam size at the sample to be large, to ensure as many powder particles are intercepted by the beam as possible, and we need a small focal spot (or line) to provide the required good Bragg angle resolution. While asymmetric crystal monochromators can provide reasonable energy resolution and a small focus, they cannot provide a large beam cross-section at the sample in a synchrotron context because of the large asymmetry necessary to maintain the Cauchois geometry and hence good energy resolution. The standard beamline 2-crystal monochromator has been chosen to provide the high energy resolution, and we use a secondary optic to provide the strong focusing [17].

The collimated beam needs to be taken from the beamline monochromator (1.5mm x 4mm) and focus it down to as small a line focus as possible. The ultimate resolution of this instrument depends on the size of this focus and the detector spatial resolution. To perform this focusing, the beam downstream of the focusing optic will have a convergence of about 2 mr. This implies a range of incidence angles on the optic of 1 mr. A grazing-incidence mirror could be used, since typical grazing angles are 3 mr or more (depending on the mirror coating). Assuming an average grazing angle of 2 mr,

the mirror would need to be almost 1m long. But this setup would make the instrument too big to fit in the available space. So, a multilayer mirror has been chosen to increase the grazing angle and maintain a high reflectivity.

Chapter 4

This chapter gives an overview of radiation detection methods and describes the theory and operation principle behind the solid state silicon strip detector, which will be used in our proposed powder diffraction instrument.

Radiation detection with Semiconductor detector

4.1   Overview of semiconductor detectors

A semiconductor detector uses a semiconductor (usually silicon or germanium) to detect traversing charged particles or the absorption of photons.

Historically, semiconductor detectors were conceived as solid-state ionization chambers. To obtain a high-electric-field, low-current, solid-state device for detection and possibly spectroscopy of ionizing radiation, conduction counters (highly insulating diamond crystals) were first used. However, such crystals were quickly rejected because of poor charge collection characteristics resulting from the deep trapping centers in their bandgap. After the highly successful development of silicon (Si) and germanium (Ge) single crystals for transistor technologies, the conduction-counter concept was abandoned, and silicon and germanium ionizing radiation detectors were developed by forming rectifying junctions on these materials. A semiconductor detector is a large silicon or germanium diode of the p-n or p-i-n type operated in the reverse bias mode [20]. At a suitable operating temperature (normally around 300 K for silicon detectors and 85 K for germanium detectors), the barrier created at the junction reduces the leakage current to acceptably low values. Thus an electric field can be applied that is sufficient to collect the charge carriers liberated by the ionizing radiation [21].

4.2   Advantages of semiconductor detectors

In many radiation detection applications, the use of a solid detection medium is of great advantage. For the measurement of high energy electrons, detector dimensions can be kept much smaller than the equivalent gas-filled detector as solid densities are some 1000 times greater than that for a gas. Scintillation detectors offer one possibility of providing a solid detection medium but one of the major drawbacks of scintillation counters is their relatively poor energy resolution. The chain of events that must take place in converting the incident radiation energy to light and the subsequent generation of electrical signal involves too many inefficient steps. Therefore, the energy required to produce one information carrier (a photoelectron) is of the order of 100 eV, so the number of carriers created for a typical x-ray of 10keV is about 100. The statistical fluctuations in such a small number place an intrinsic limitation on the energy resolution that can be achieved under the best circumstances and nothing can be done about improving the energy resolution beyond this point.

The only way to reduce this energy resolution statistical limit is to increase the number of photoelectron per pulse. As we discuss in this chapter, the use of semiconductor materials as radiation detectors can result in a much larger number of carriers (more than 20,000) for a given incident radiation event than is possible with any other common detector type [22]. As a result, the best energy resolution from radiation spectrometers is achieved using semiconductor detectors. In this case, the fundamental information carries are electron-hole pairs created along the path taken by the primary radiation through the detector.

Besides better energy resolution, solid-state detectors can also have a handful of other desirable features. Among these are compact size, relatively fast timing characteristics, and an effective thickness that can be varied to match the requirements of the application.

4.3   Operation principles of semiconductor detectors

4.3.1 Process of ionizing radiation

When a charged particle passes through a semiconductor with the band structure of Fig. 4.1, many electron-hole pairs are produced along the track of the particle. The average energy expended by the primary charged particle to produce one electron-hole pairs often called the *ionization energy*, which is largely independent of both the type and energy of incident radiation. This important simplification allows interpretation of the number of electron-hole pairs produced in terms of the incident energy of the radiation.
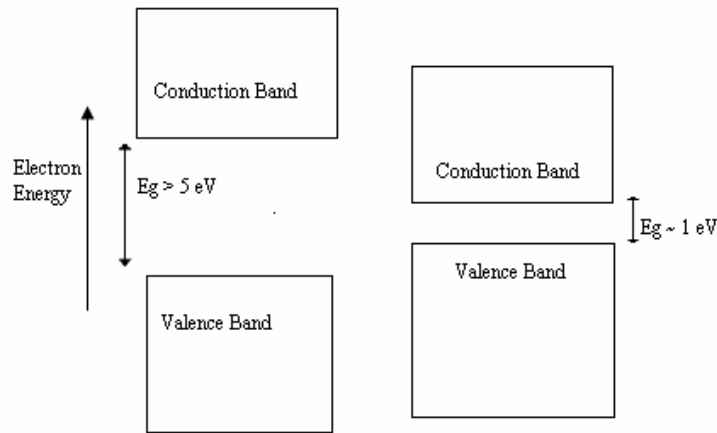
Figure 4.1: Band structure for electron energies in insulators and semiconductors.

Radiation is measured by means of the number of charge carriers set free in the detector, which is arranged between two electrodes. Ionizing radiation produces free electrons and holes. The number of electron-hole pairs is proportional to the energy transmitted by the radiation to the semiconductor [20]. As a result, a number of electrons are transferred from the valence band to the conduction band, and an equal number of holes are created in the valence band. Under the influence of an electric field, electrons and holes travel to the electrodes, where they result in a pulse that can be measured in an outer circuit. The holes travel into the opposite direction and can also be measured. As the amount of energy required to create an electron-hole pair is known, and is independent of the energy of the incident radiation, measuring the number of electron-hole pairs allows the energy of the incident radiation to be found.

4.3.2 Pulse formation

After a particle deposits energy in a semiconductor detector, equal numbers of conduction electrons and holes are formed within a few picoseconds. The detector configuration ensure that an electric field is present throughout the active volume, so that both charge carriers feel electrostatic forces that cause them to drift in opposite directions. The motion of either the electrons or holes constitutes a current that will persevere until those carriers are collected at the boundaries of the active volume [22].
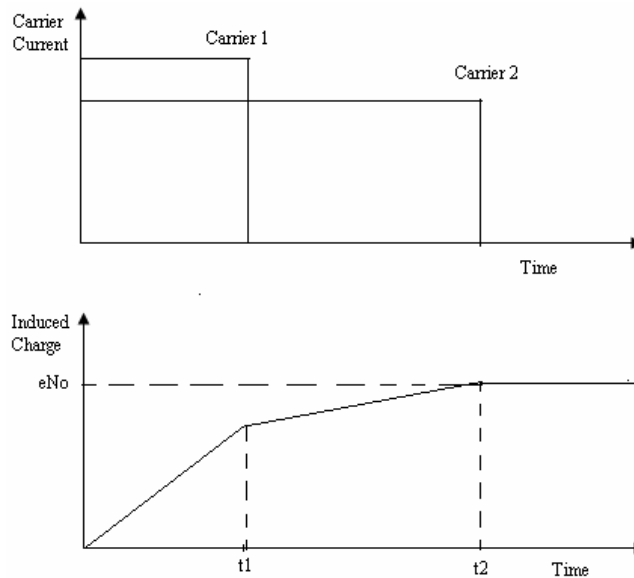
Figure 4.2: The upper plot shows a representation of the electron and hole currents flowing in a semiconductor following the creation of N_0 electron-hole pairs. In the lower plot, t_1 represents the collection time for the carrier type that is collected first, and t_2 is the collection time for the other carrier. If both are fully collected, a charge of eN_0 is induced to form the signal, where e is the electron charge.

Assuming all the charge carriers are formed at a single point, the resulting currents can be represented by the upper plot of Figure 4.2. As the charge collection times are not the same because of differences in drift distance and carrier motilities, on of the two currents will persist for a longer time than other. When a measuring circuit integrates these currents with long time constant, the measured induced charge has the time characteristics of the lower plot of Figure 4.2. This time profile will also be that of the rise of the pulse produced by a conventional preamplifier used to process the pulses from the detector. In silicon semiconductor detectors, the hole mobility is within a factor of about two or three of the electron mobility, hence the collection times are much closer to

26

being equivalent [20]. As a result, standard silicon detectors rely on complete integration of the currents duo to both electron and holes. Both carrier types must therefore be completely collected for the resulting pulse to be a reliable measure of the particle deposition energy.

4.4 Position sensitive silicon strip detector array

A combination of BNL's HERMES integrated circuit and silicon microstrip arrays will provide us with an efficient, low-noise, high-resolution position sensitive detection system. It is based on a multi-element Si sensor and dedicated readout application specific integrated circuit (ASIC). It will have a full detector array of about 7000 elements covering 90 degrees of the Rowland circle (45 degrees in 2-theta) [17]. Our Rowland circle diameter will be around 1 meter large, which would allow us to have greater depth of focus while minimizing aberrations and better angular resolution for the same spatial resolution. Temperature-controlled circulation of water will be used to control the operating temperature of both the detector and the integrated circuit.
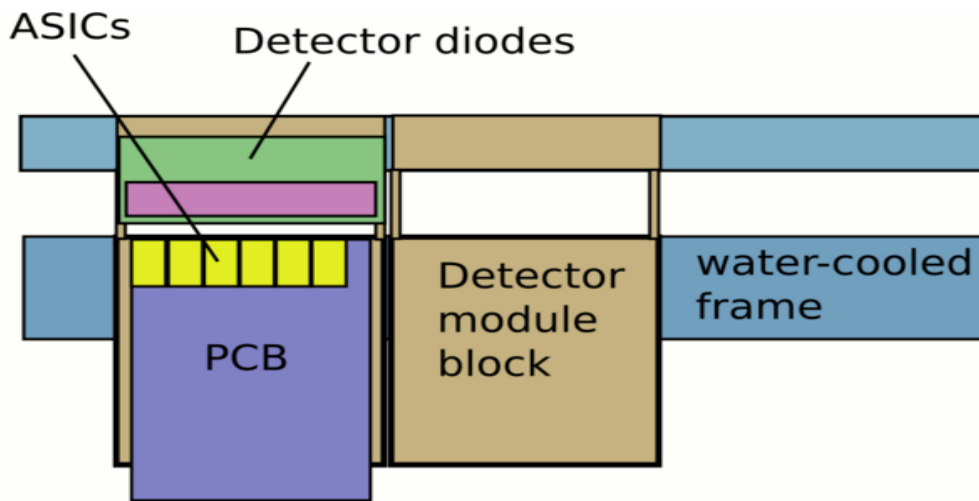
Figure 4.3: Schematics of the detector system.

The sensor is composed of 640 pixels, each having 0.125 X 4 mm² area, arranged in a linear array. It is wire-bonded to twenty 32-channel ASICs. Each channel implements low-noise pre-amplification with self-adaptive continuous reset, high-order shaper, bandgap referenced baseline stabilizer, one threshold comparator, and two digital-analog

converter (DAC) adjustable window comparators, each followed by a 24-bit counter [19]. Fabricated in 0.35 m CMOS, the ASIC dissipates about 8 mW per channel.

The primary components of the detector system are addressed in the following sections.

4.4.1 Multi-element silicon sensor

The silicon diode microstrip array (Fig. 4.4) was fabricated at the Semiconductor Detector Laboratory of the Instrumentation Division, BNL. The pixels were formed using boron implant (at 40 keV) on a high-resistivity (4–6 kilohm-cm) n-type 400 micrometers thick wafer with (111) orientation. An ohmic contact is formed on the back side, using a phosphorous implant (at 150 keV). The sensitive thickness of the diodes is 0.4mm, allowing efficient detection up to about 12 keV [19].
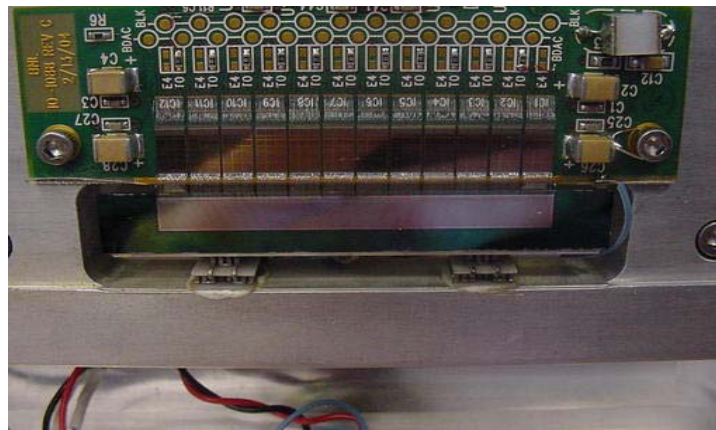
Figure 4.4: The silicon diode microstrip array [17].

4.4.2 Frond-end ASIC configuration

A mixed-signal ASIC was designed at the Instrumentation Division and fabricated in 0.35 μm CMOS, dual-poly, four-metal (DP-4M) 3.3 V technology. It is composed of 32 channels plus bias circuitry and digital interface, a total of 180,000 MOSFETs, and it dissipates about 8 mW per channel. The layout, which is optimized for minimization of pick up from mixed signal activity, measures 6.3 x 3.6 mm² (Fig. 4.5). Each channel implements (Fig. 4.6) a low-noise preamplifier, a high-order shaper with baseline stabilizer, one threshold and two window discriminators with digital-analog converters (DACs) for fine adjustments and one counter per discriminator. The preamplifier incorporates an n-channel MOSFET in feedback, operating in saturation to realize a low-

noise fully compensated continuous reset circuit of the type described in [19]. The configuration is self-adaptive to leakage currents from the sub-pA to the nA scale. The reset circuitry contributes parallel noise, which, in the worst case of weak inversion operation, is equal to the shot noise of the pixel leakage current. A charge gain of 32 is provided. A cascade of two of these stages, the second of which, based on p-channel MOSFET in feedback, provides an overall gain of 1024(i.e. 32x32) from the detector to the input of the first stage of the shaper.
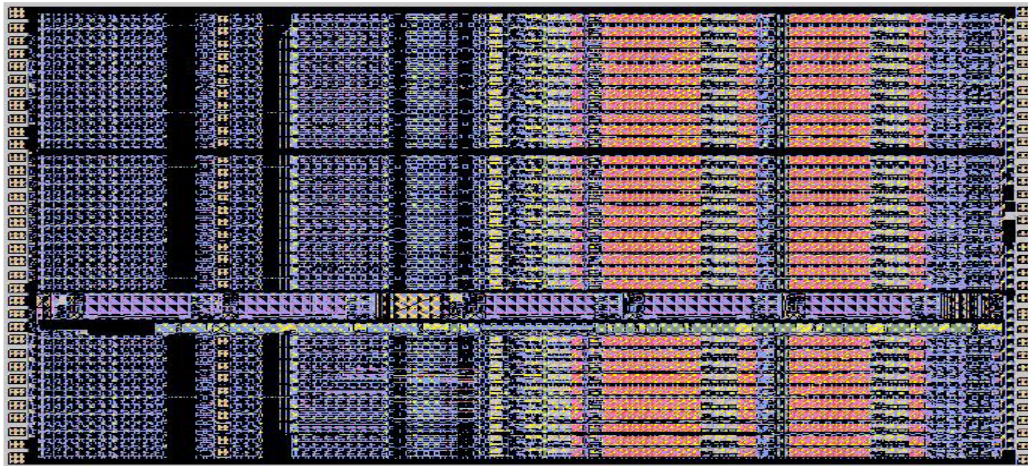
Figure 4.5: Photograph of 32 channel ASIC, composed of 18, 0000 MOSFETs [19].

Inside the shaper, there is a 192 kΩ feedback resistor that contributes an equivalent input shot noise of 240 fA. The second and third stages of the shaper provide two pairs of complex conjugate poles, forming a fifth order complex semi-Gaussian shaper. The high-order shaper was chosen for its better noise filtering performance (up to 2.6 times compared to a low order) in the high-rate regime, where white series noise dominates.

The output stage adds a gain of four. Its output baseline is referenced to a bandgap circuit and an error voltage is fed back to the input of the second stage of the shaper through a slew-rate limited follower and a very low frequency filter, thus realizing a BLH configuration [5] for the baseline stabilization. The analog processing chain has an adjustable peaking time (0.5 μs , 1 μs, 2 μs, 4 μs), and gain (750 mV/fC, 1500 mV/fC).

The analog output and a pixel leakage current monitor, with equivalent gain~8 GΩ, can be routed to two global analog outputs through digital setting. The shaper output

is fed to one single-threshold and two window-discriminators. The five threshold levels are coarsely set through external voltages common to all channels. Each threshold level of the two windows can be individually adjusted through a 6-bit DAC with 1.6 mV step. Each discriminator is followed by a 24-bit counter (three counters per channel). During the data readout phase, the counters of all 32 channels are converted into a single shift register for serial reading.
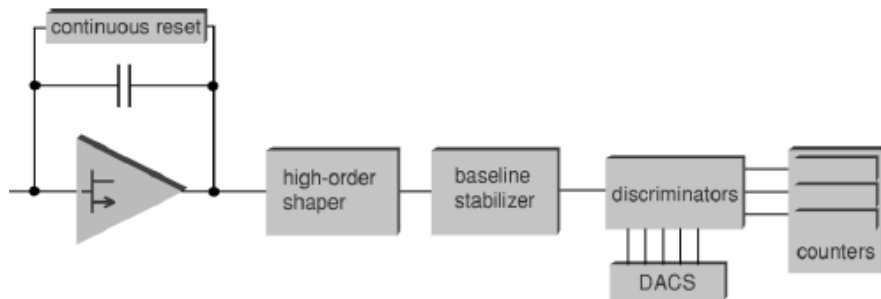
Figure 4.6: Simplified schematics of the one channel of the ASIC [19].
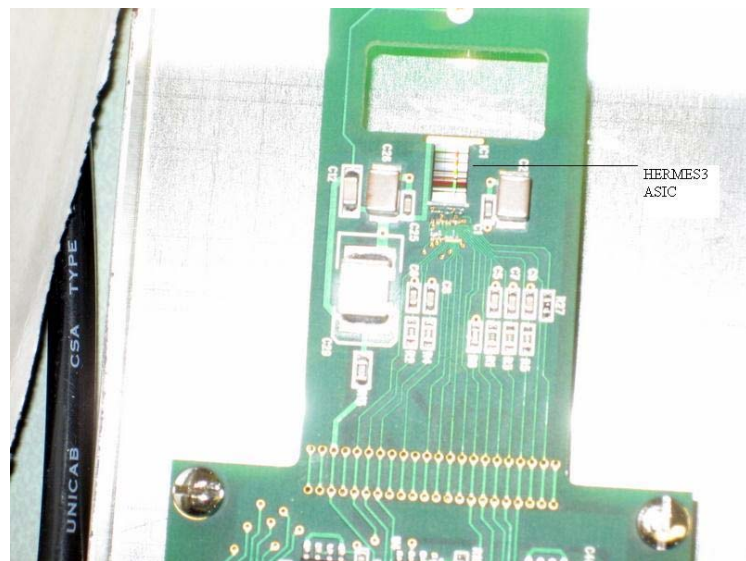
Figure 4.7: Picture of an actual HERMES3 ASIC chip.

A serial peripheral interface is also included for global settings, monitor enabling, channel masking, DAC adjustment, and counter readout. The channel area is 5854 μm by 102 μm. In the long dimension, 1929 μm is dedicated to the comparators and DACs while 690 μm is devoted to the three 24-bit counters. The power dissipated by the overall chain is 8 mW, with 3 mW dedicated to the preamplifier.

Chapter 5

This chapter provides an introduction to 'Field Programmable Gate Arrays' (FPGA), with a detail discussion of Xilinx Virtex-4 FPGA device which we have used to build the detector control system.

Detector control system design with Xilinx FPGA

5.1   Overview of programmable logic devices

Ordinarily, *a programmable logic device (PLD) is* a common term referring to any type of integrated circuits that can be configured by the end user for a particular design implementation. As PLDs are programmed 'in the field' by the end user, they are also called *field programmable logic devices* (FPLDs). Unlike a logic gate, which has a fixed function, a PLD has an undefined function at the time of manufacture. Before the PLD can be used in a circuit it must be programmed.

5.1.1   Programmable array logic

The first programmable logic devices were called 'PAL's, for *programmable array logic*. The programmable array contains logic gates, themselves fixed in function, with programmable interconnections between them. The array has a number of inputs and outputs, and can create any Boolean function of a selection of the inputs at any of its outputs [23]. A single PAL can replace a circuit containing a large number (perhaps a few hundred) of fixed logic gates. In a PAL the logic gates are arranged as a sum-of-products array. A PAL is programmed by fitting it into a machine called a PAL programmer. PAL

programmers are usually general-purpose machines that can program all types of PLD from all manufacturers. A PAL may be programmed only once.
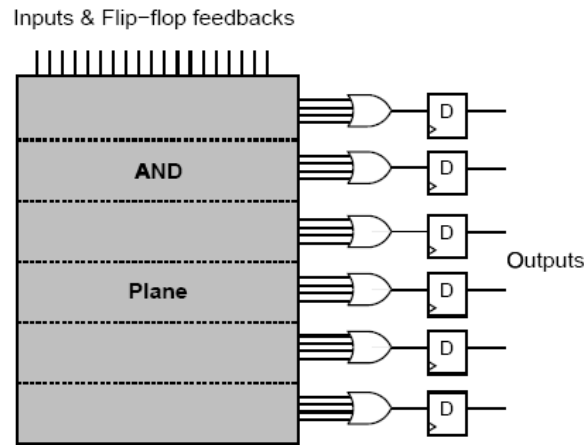
Figure 5.1: Structure of a PAL showing programmable AND- plane with fixed OR-plane [23].

5.1.2 Generic array logic devices

Generic array logic device, or 'GAL', has the same logical properties as the PAL but can be erased and reprogrammed. It was invented by Lattice Semiconductor. The GAL is very useful in the prototyping stage of a design, when any bugs in the logic can be corrected by reprogramming.

5.1.3 Complex programmable logic devices

Both PAL and GALs were available only in small sizes. As chip densities increased rapidly, it was natural for the PLD manufacturers to evolve their products into larger parts called Complex Programmable Logic Devices (CPLDs). For most practical purposes, CPLDs can be thought of as multiple PLDs (plus some programmable interconnect) in a single chip. The larger size of a CPLD helps to implement either more logic equations or a more complicated design.
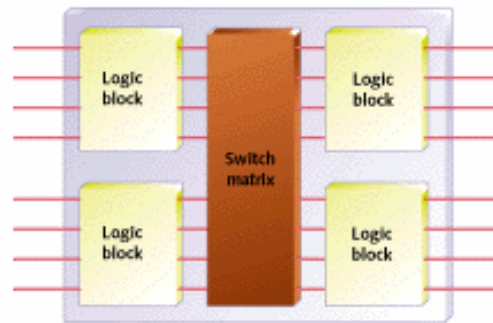
Figure 5.2: Block diagram of a CPLD. Each of the four logic blocks shown here is the equivalent of one PLD [23].

Though Figure 5.2 shows four logic bocks, in an actual CPLD there may be more (or less) than four logic blocks. These logic blocks are themselves comprised of macrocells and interconnect wiring, just like an ordinary PLD.

Unlike the programmable interconnect within a PLD, the switch matrix within a CPLD may or may not be fully connected. In other words, some of the theoretically possible connections between logic block outputs and inputs may not actually be supported within a given CPLD. The effect of this is most often to make 100% utilization of the macrocells very difficult to achieve. Some hardware designs simply won't fit within a given CPLD, even though there are sufficient logic gates and flip-flops available [24].

Because CPLDs can hold larger designs than PLDs, their potential uses are more varied. They are still sometimes used for simple applications like address decoding, but more often contain high-performance control-logic or complex finite state machines.

5.1.4   Field Programmable Gate Array

While PALs were busy developing into GALs and CPLDs (all mentioned above), a separate stream of development was happening. This type of device is based on gate array technology and is called the field-programmable gate array (FPGA)[25]. Early examples of FPGAs are the 82s100 array, and 82S105 sequencer, by Signetics, introduced in the late 1970s.

33

FPGAs use a grid of logic gates, similar to that of an ordinary gate array, but the programming is done by the customer, not by the manufacturer. The term "field-programmable" means the array is done outside the factory, or "in the field"[24].

FPGAs are usually programmed after being soldered down to the circuit board, in a manner similar to that of larger CPLDs. In larger FPGAs the configuration is volatile, and must be re-loaded into the device whenever power is applied or different functionality is required. Configuration is typically stored in a configuration PROM or EEPROM. EEPROM versions may be in-system programmable via JTAG cable.
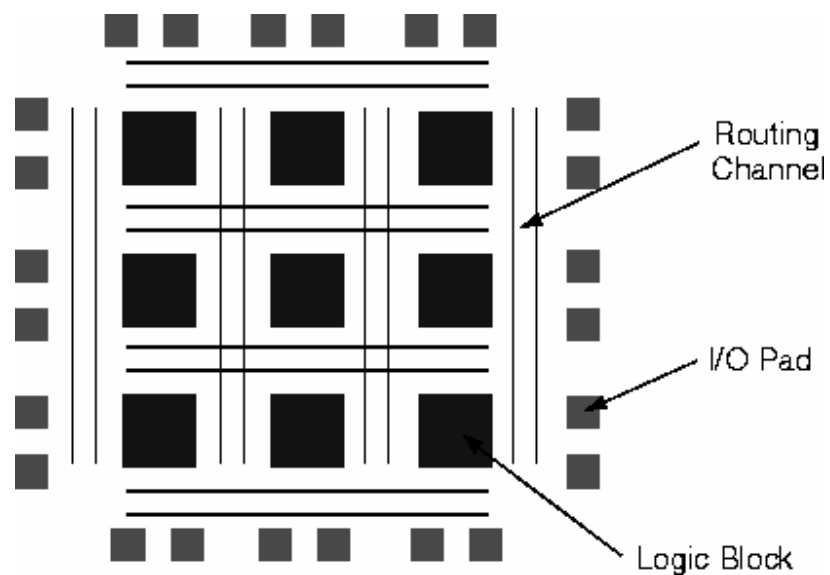
Figure 5.3: Simplified FPGA structure [25].

5.2   FPGA Architecture

The common basic architecture consists of an array of configurable logic blocks (CLBs) and routing channels. A generic FPGA consists of numerous programmable logic blocks which have the capability to implement some digital logic functions. In between these logic blocks are programmable routing switches which connect the input and output pins of each logic block.

A classic FPGA logic block consists of a 4-input lookup table (LUT), and a flip-flop, as shown below. In recent years, manufacturers have started moving to 6-input LUTs in their high performance parts, claiming increased performance.
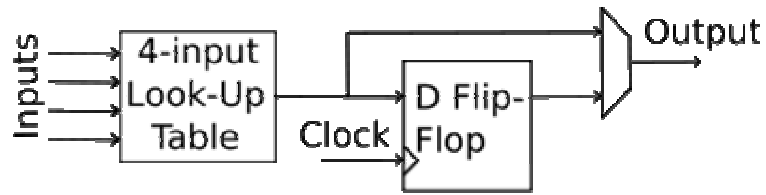
Figure 5.4: Typical logic block of an FPGA.

There is only one output, which can be either the registered or the unregistered LUT output. The logic block has a clock input and four inputs for the LUT. As the clock signals (and often other high-fanout signals) are normally routed via special-purpose dedicated routing networks in commercial FPGAs, they and other signals are separately managed. For this specific example architecture, the locations of the FPGA logic block pins are shown below [26].
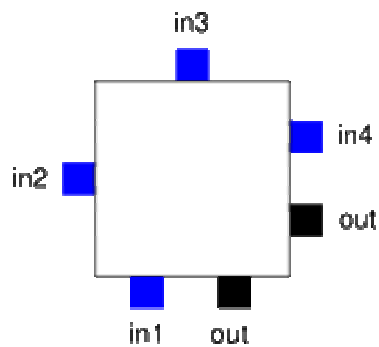
Figure 5.5: Logic block pin location for the specific example in Fig. 27.

Each input is accessible from one side of the logic block, while the output pin can connect to routing wires in both the channel to the right and the channel below the logic block. Each logic block output pin can connect to any of the wiring segments in the channels adjacent to it.

Modern FPGA families expand upon the above capabilities to include higher level functionality fixed into the silicon. Having these common functions embedded into the silicon reduces the area required and gives those functions increased speed compared to building them from primitives. Examples of these include multipliers, generic DSP blocks, embedded processors, high speed IO logic and embedded memories.

5.3 Xilinx Virtex-4 FPGA

For our detector control system, we have selected the Virtex-4 FPGA, which is the first multi-platform FPGA family based on the Advanced Silicon Modular Block (ASMBL) architecture. The Virtex-4 family offers a programmable logic solution closely matching our needs. It has several new architectural elements designed for maximum throughput, higher integration and lower power consumption. As shown in Figure 5.6, the configuration architecture is frame-based, where a frame spans 16 rows of configurable logic blocks (CLBs). Clock distribution regions are also aligned in blocks of 16 CLB rows, unlike earlier Vertex devices, where clock regions were defined to be quadrants [27].

Also the I/O blocks are arranged in columns (like all other resources) rather than a ring. The Virtex-4 shares the glitchless dynamic reconfiguration property of earlier devices, but this now applies to all primitives including LUT RAM and SRL16 logic.

The primary components of a Virtex-4 FPGA are discussed below.
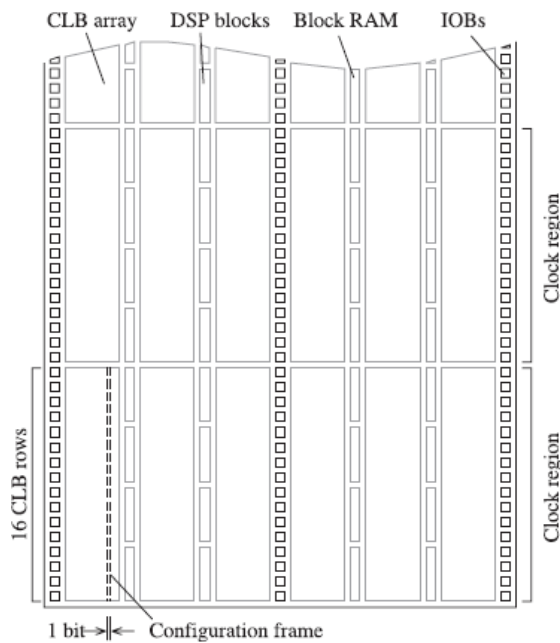
Figure 5.6: Xilinx Virtex series Architecture [27].

36

5.3.1 Input/Output Blocks

I/O blocks provide the interface between package pins and the internal configurable logic. Most I/O standards are supported by programmable I/O blocks (IOBs). The IOBs are designed for source-synchronous applications. Source synchronous optimizations include per-bit deskew, data serializer/deserializer, clock dividers, and dedicated local clocking resources. The IOB registers are either edge-triggered D-type flip-flops or level-sensitive latches.

5.3.2 Configurable Logic Blocks

Configurable Logic Blocks (CLBs), the basic logic elements for Xilinx FPGAs, provide combinatorial and synchronous logic as well as distributed memory and SRL16 shift register capability. CLB resources are made of four slices. Each slice contains two function generators (F & G), two storage elements, arithmetic logic gates, Large multiplexers and fast carry look-ahead chain.

The function generators F & G are configurable as 4-input look-up tables (LUTs). Two slices in a CLB can have their LUTs configured as 16-bit shift registers, or as 16-bit distributed RAM. In addition, the two storage elements are either edge-triggered D-type flip-flops or level sensitive latches. Each CLB has internal fast interconnect and connects to a switch matrix to access general routing resources.

5.3.3 Block RAM

The block RAM resources are 18 Kb true dual-port RAM blocks, programmable from 16K x 1 to 512 x 36, in various depth and width configurations. Each port is totally synchronous and independent, offering three "read-during-write" modes. Block RAM is cascadable to implement large embedded storage blocks. Additionally, back-end pipeline registers, clock control circuitry, built-in FIFO support, and byte write enable features are also supported in the Virtex-4 FPGA.

5.3.4 DSP Slices

The Virtex-4 DSP slices are organized as vertical DSP columns. Within the DSP column,

two vertical DSP slices are combined with extra logic and routing to form a DSP tile. The DSP tile is four CLBs tall. Each DSP48 slice has a two-input multiplier followed by multiplexers and a three-input adder/subtracter. The multiplier accepts two 18-bit, two's complement operands producing a 36-bit, two's complement result. The result is sign extended to 48 bits and can optionally be fed to the adder/subtracter. The adder/subtracter accepts three 48-bit, two's complement operands, and produces a 48-bit two's complement result. Each multiplier or accumulator can be used independently. These blocks are designed to implement extremely efficient and high-speed DSP applications.

5.3.5   Clocking Resources

The DCM and global-clock multiplexer buffers provide a complete solution for designing high-speed clock networks. Up to twenty DCM blocks are available. To generate deskewed internal or external clocks, each DCM can be used to eliminate clock distribution delay. The DCM also provides 90°, 180°, and 270° phase-shifted versions of the output clocks. Fine-grained phase shifting offers higher resolution phase adjustment with fraction of the clock period increments. Flexible frequency synthesis provides a clock output frequency equal to a fractional or integer multiple of the input clock frequency. Virtex-4 devices have 32 global-clock MUX buffers. The clock tree is designed to be differential. Differential clocking helps reduce jitter and duty cycle distortion.

5.4   Memec Virtex-4 FPGA Mini-module

The Memec Virtex-4 FX12 mini module was chosen to integrate with our HERMES3 detector ASIC to devise the detector control system. This mini module is a complete system on a module, packaging all the necessary functions needed for an embedded processor system onto a tiny footprint slightly bigger than a stick of chewing gum [28]. Figure 5.7(a) provides an overview of the module while Figure 5.7(b) and Figure 5.8 shows clocking components and the module functional block diagram respectively.
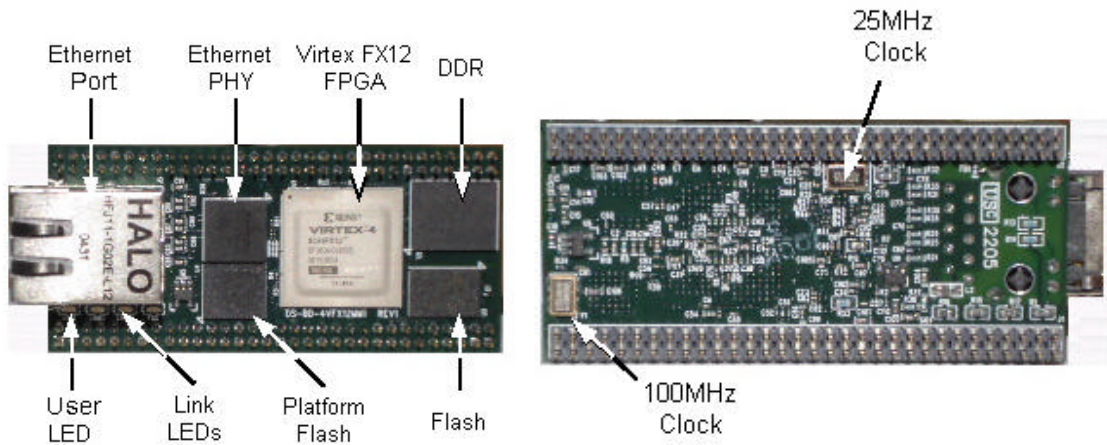
Figure 5.7: (a) Top view of Xilinx Virtex-4 FPGA Mini Module. (b) Bottom view of the module showing the clocking components onboard [28].
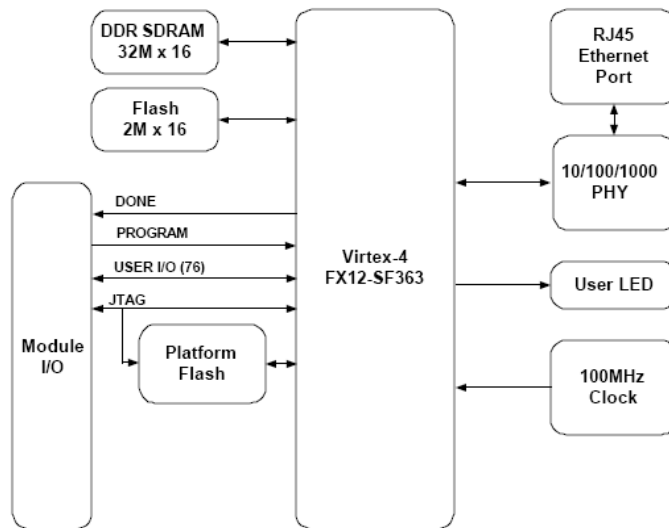
Figure 5.8: Block Diagram of the Virtex-4 FPGA Mini module [28].

The Mini-Module is a small plug-in board, containing a Xilinx Virtex-4 FPGA, configuration memory, RAM, parallel Flash memory, an Ethernet port, a clock source and a user LED. The module provides 76 user definable IO pins through the 2x32 pin headers along each side of the circuit board. The module connects via a socket or thru-hole mounting to a baseboard, which provides the various module power requirements, JTAG interfaces, some status and control, and any user application circuitry.

5.5   Intellectual Property (IP) Core

An IP (intellectual property) core is a block of logic or data that is used in making a field programmable gate array (FPGA) or application-specific integrated circuit (ASIC) for a product. As essential elements of design reuse, IP cores are part of the growing electronic design automation industry trend towards repeated use of previously designed components. IP cores are typically offered as generic gate netlists. The netlist is a boolean-algebra representation (gates, standard cells) of the IP's logical-function, analogous to an assembly-code listing for a high-level program application. The netlist protects the vendor against reverse-engineering, while maintaining portability to multiple foundry targets. Synthesizable cores are delivered in a hardware description language such as Verilog or VHDL. Both netlists and synthesizable cores are called 'soft cores', as both follow the SPR design-flow (synthesis, placement and route) [29].

For our purpose of devising the control system, we have designed three custom IP peripherals to be attached to the on-chip-peripheral bus of the virtex-4, which will be used to communicate with the HERMES3 chip of the detector. These peripherals are discussed in detail the following chapter.

Chapter 6

In this chapter, we will discuss the custom IP cores in Xilinx Virtex-4 FPGA mini-module, which have been designed to communicate with the detector ASIC for sending and receiving data. Also test results will be provided to demonstrate their operational accuracy.

Custom IP cores for the detector control system

6.1   Required peripherals and ports

Three custom peripherals are required on our Virtex-4 mini module embedded processor system that will be interacting with the HERMES3 ASIC (Appendix 1). These are discussed below.

6.1.1   Timer

This timer works as an accurate timebase to time the period of data collection. It will generate a single pulse with a duration of anywhere between 1 milisecond and 100 seconds. A 20-bit down counter was implemented with a clock rate of 1 kHz. The output pulse is given by a flip flop, which is set by first clock pulse after loading, and cleared by the OR of all counter bits. Then we can load a number of milliseconds into the counter, enable the flip-flop and wait for the counter to reach zero, which clears the flip-flop. The timer output will be connected to the *counter enable* input of the ASIC.

6.1.2   Data Out to ASIC

This peripheral will be responsible to send out instructions from the CPU to the detector ASIC. It has the following outputs.

6.1.2.1 SCK

This is the serial clock output, generated by a clock divider. The 100 MHz system clock has been divided by 10 to generate this 10 MHz serial clock. This clock output idles high. Negative edge shifts into the write mode, while positive edge shifts into the read mode.

6.1.2.2 SDI

This is the serial output that goes from the FPGA to the ASIC. When the 10-bit DAC is disabled (SDAC = 0), it has 904 bit serial data stream containing 28 bits per channel (ECH, ECAL, ELK, EAN, D0-D5/VL1, D0-D5/VH1, D0-D5/VL2, D0-D5/VH2] for 32 channels and 8 bits containing gain (G), timing controls (T2, T1), internal bias leakage enable (EBLK), DAC monitors enable to OAN (SDA2:SDA0). The first bit of data stream is D5/VH2 of Ch31, last bit of data stream is 0.

When 10-bit DAC is enabled (SDAC = 1), it has 50 bit serial output, which contains P0:P9 (VH2), P0:P9 (VL2), P0:P9 (VH1), P0:P9 (VL1), P0:P9 (VL0), the desired contents of the five on-chip 10-bit threshold voltage DACs. The first bit of the data stream is P9 (VL0), the last bit of the data stream is P0 (VH2).

6.1.2.3 Other outputs

Other output ports include chip select (CS), enable (EN), read/write mode (WR) and global reset (RST). CS is active high. EN=1 enables the counter, while En = 0 disables it. We will be using the timer output to perform counter enabling duties for the ASIC. WR= 1 shifts to read mode and WR = 0 shifts to write mode for the ASIC. Also, RST is active low [30]. Figure 6.1 illustrates the FPGA setup with the ASIC input and output ports.
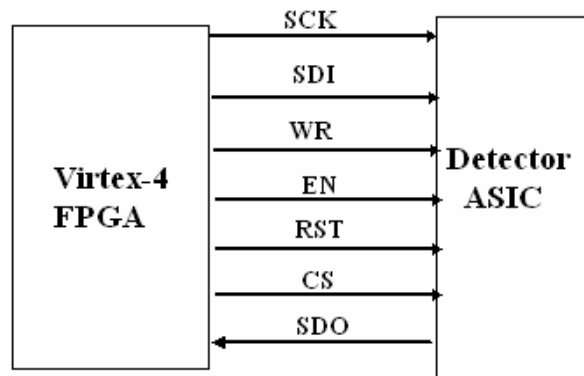
Figure 6.1: Block Diagram of the control system hardware setup.

6.1.3   Data In from ASIC (SDI)

This serial input contains 2304 bit data stream from the ASIC. It incorporates 24 bits per comparator, which results 72 bits per ASIC channel (C23-C0/W0, C23-C0/W1, C23-C0/W2) for 32 channels. There are 3 comparators per channel; 1 threshold comparator and 2 window comparators. The first bit of data stream is C23/W0 of Ch31, while the last bit of data stream is C0/W2 of Ch0 [30].

6.2   Embedded system structure

To design the above custom peripherals to interact with the HERMES3 ASIC, we have developed an embedded PowerPC processor system using Xilinx EDK 9.1 software. The required peripherals were created inside that baseboard as custom Intellectual Properties (IP) and integrated into the system. The 'On-Chip peripheral Bus' (OPB) is a fully synchronous bus that functions independently at a separate level of bus hierarchy. It is not intended to connect directly to the processor core. All the peripherals were connected to the PowerPC via this OPB. The OPB interfaces provide separate 32-bit address and up to 32-bit data buses. Since the OPB supports multiple master devices, the address bus and data bus are implemented as a distributed multiplexer. The processor core can access the slave peripherals on this bus through the "PLB to OPB" bridge unit (PLB = Processor Local Bus). Peripherals which are OPB bus masters can access memory on the PLB

43

through the "OPB to PLB" bridge unit (Figure 6.2). OPB arbiters can be implemented on FPGA fabric and are available as soft IP cores.
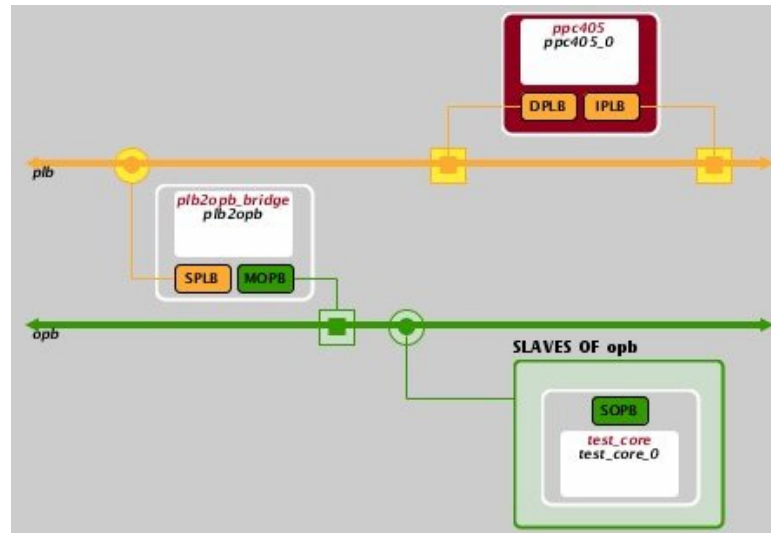
Figure 6.2: Overview of the OPB bus architecture connectivity shown here with the PowerPC processor with a test OPB IP.

The architecture and operation of the three peripherals are discussed below.

6.2.1 Timer

The timer IP takes an input in milliseconds unit and has a range from 1 to 10^6 milliseconds. This is basically a counter which runs at 1 KHz clock, so we had to divide our 100 MHz system clock to scan at a lower rate of 1 KHz. After receiving a specific input time within its range, the timer output stays high for that period of time and then goes low when the count is over. In this IP, we have devised two software controlled 32-bit registers, one of them to receive the input time and the other one act as a control/status register. After setting the input time, if the user sets a '1' at bit location '0' of the command/status register, the timer will start counting down and the output stays high. As soon as the counter hits zero the timer outputs low and the location '0' of the command/status register will become zero informing the cpu/user that the timer is ready to accept the next counter value.

6.2.2 Data output to ASIC & Data input from ASIC

As discussed before (6.1.2), this output to the ASIC contains a serial bit stream of 904 bits for total 32 ASIC channels or 50 bits depending on the status of the SDAC bit. The input data is a serial beat stream of 2304 bits. As both the output and input from the cpu is parallel 32 bits we have devised two shift registers for the data streams. One of them is a parallel to serial shift register for the output data to ASIC and the other one is a serial to parallel shift register for the data input from ASIC. The complete peripheral works as a Serial Peripheral Interface (SPI) bus, where the FPGA works as the master device and initiates the data frame. During one clock cycle, the FPGA sends out a single bit via SDI port to the ASIC on the falling edge while the ASIC sends a single bit to the FPGA via the SDO port on the rising edge of the serial clock.
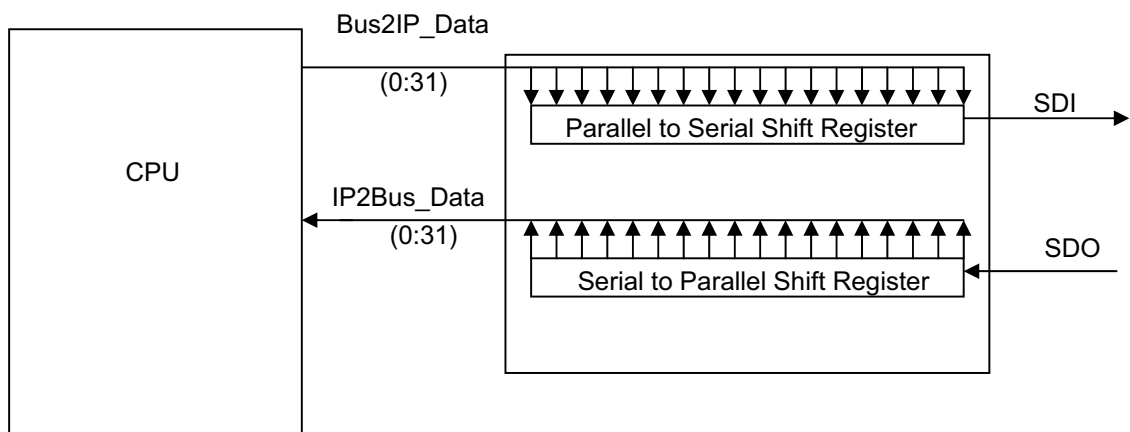
Figure 6.3: Architecture of the 'Serial Data Out' (SDO) and 'Serial Data In' (SDI) peripherals. SDI operates on the basis of a 32 bit parallel to serial shift register, while SDO is based on a 32 bit serial to parallel shift register.

There are also three more registers which handles instructions of numbers of transferable bits, a control/status register and a register for the other single bit inputs to the ASIC (i.e. write/read enable, counter enable, global reset and chip select). The parallel to serial shift register shifts data out from least significant bit (LSB) to most significant bit while the serial to parallel shift register does it the other way around, as required by the ASIC data sheet (See Appendix. 1).

6.3   Operational details of the control system

The operation starts by setting the chip select high for the desired chip (test data was taken with a single chip), write/read mode to low (write mode) and 10-bit DAC enable to high (SDAC=1). Then, 50 bits of serial data stream are sent to SDI to setup the 10-bit DAcs. After that, the SDAC is turned low and 904 bits of serial data are sent to the ASIC via SDI port by the 'Data Out to ASIC' IP. This data stream controls the gain, peaking time control, internal leakage enable and all the settings for three comparators. After that, the chip select is set to low to prepare the ASIC for counting. Then, the timer IP gets input for the desired milliseconds (integration time) and stays high for that time period for which the detection counters on the ASIC will be enabled. After the timer goes low, counters get disabled and the 2304 bit serial data is ready to be read back by the 'Data In from ASIC' IP via the SDO port. At this point, the write/read mode and the chip select are both set to high (read mode) and the CPU reads back the number of counts for two window and one threshold comparators for the user defined timeframe. The timing diagram of 'send' and 'receive' data is illustrated in Figure 6.4.
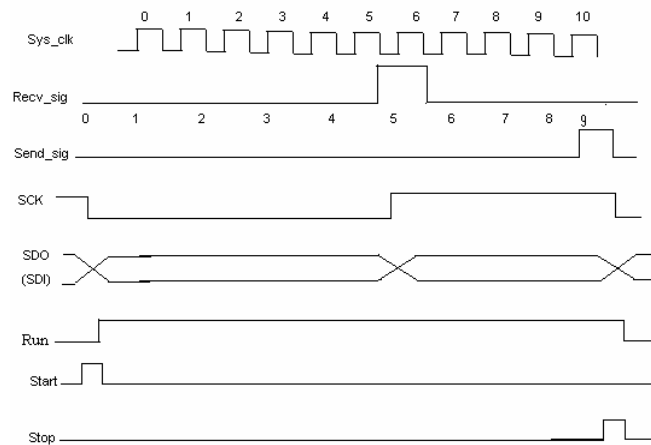
Figure 6.4: Timing diagram for the send and receive data signal. It shows all the internal signals used to program the IP in Xilinx ISE software. Serial Clock (SCK, running at 10 MHz) is driven from the system clock (Sys_clk, running at 100 MHz). Receive signal gets high at 5th system clock edge, while send signal gets high at 10th clock edge for 1 cycle. First bit of 'Data out to ASIC' (SDI) is available at the falling edge of the SCK, while 'Data out from ASIC' (SDO) is available at the rising edge of the serial clock. The 'Run' signal stays high for a complete SCK clock cycle. The activation of start signal loads the data into the parallel to serial shift register for sending out to ASIC while the stop signal makes the data stream coming from ASIC available for CPU reading.

6.4   Peripheral test and evaluation

After building the three custom OPB peripherals to communicate with the HERMES3 chip, we have integrated them to the EDK embedded processor system. XMD debugger was used to test the operational ability of the timer, data-out and data-in IPs. After that, the HERMES3 ASIC was connected to the Xilinx Viretex-4 FPGA mini module for real-time hardware testing. To simulate detector events, a 20 kilovolt tail pulse generator with 1 KHz frequency was used to provide the calibration input for the ASIC and the output was observed on the oscilloscope.
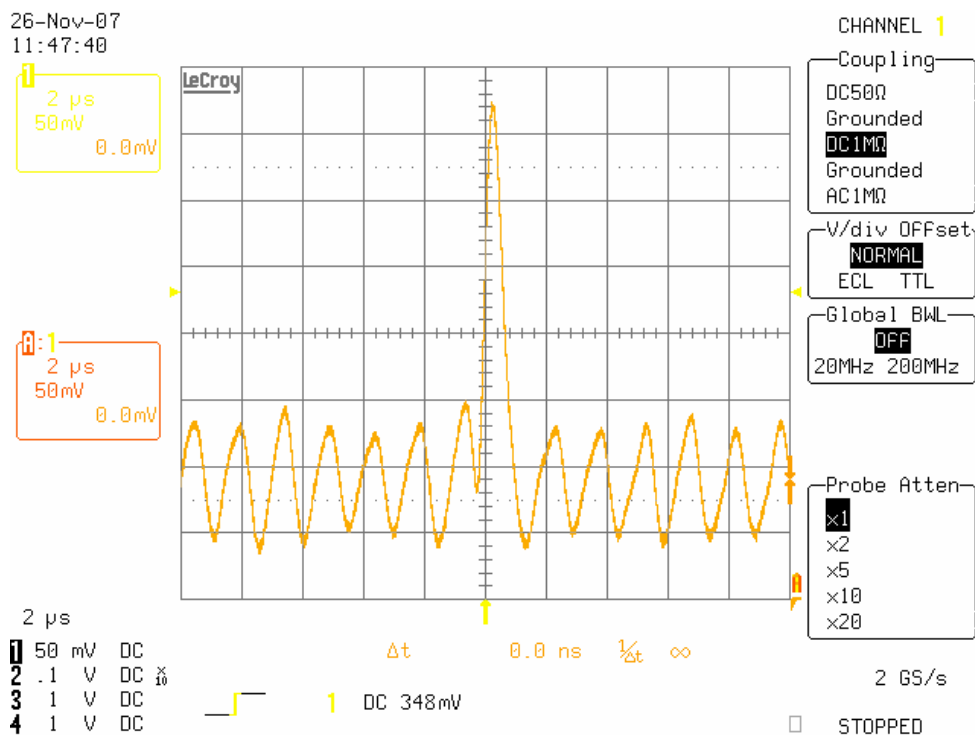
Figure 6.5: Output signal from the ASIC for 0.5 µs peaking time control input.
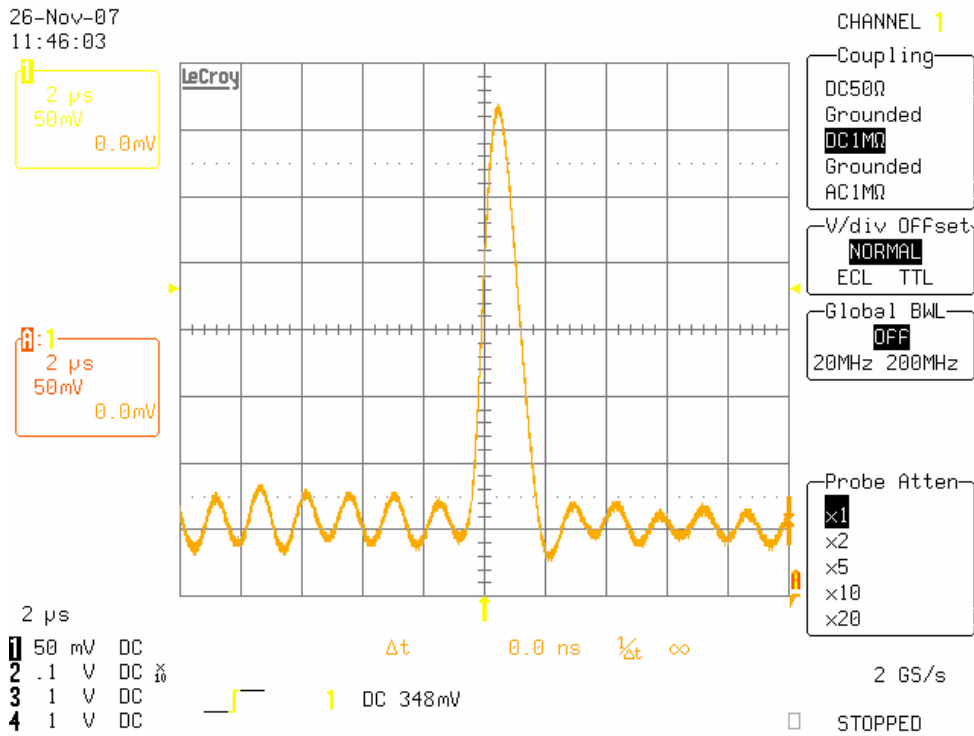
Figure 6.6: Output signal from the ASIC for 1 μs peaking time control input.
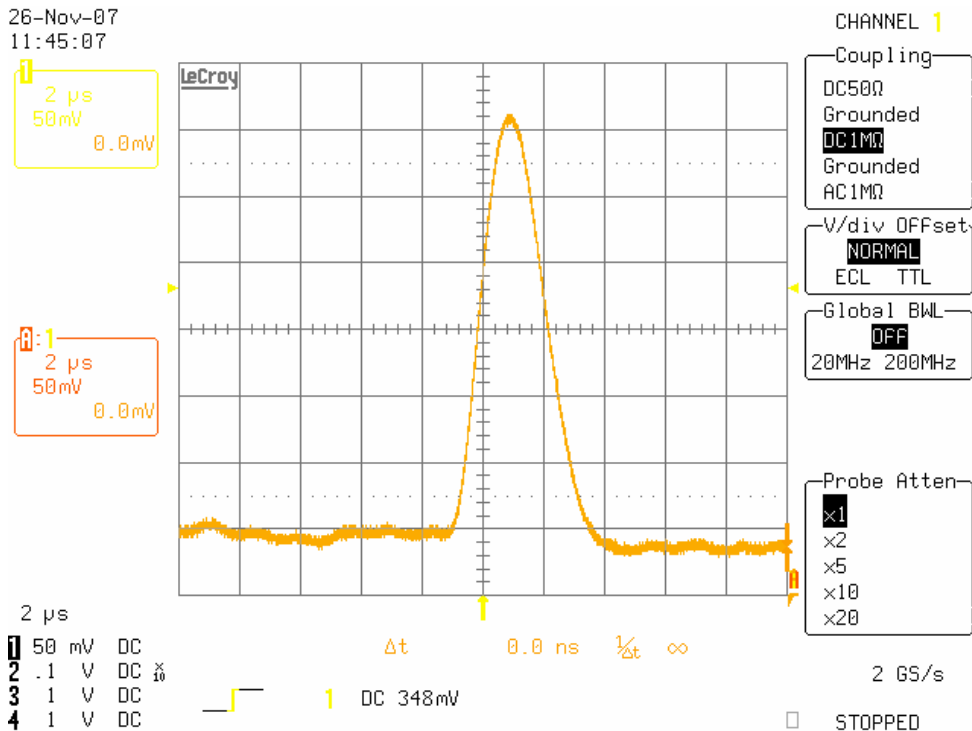
Figure 6.7: Output signal from the ASIC for 2 μs peaking time control input.
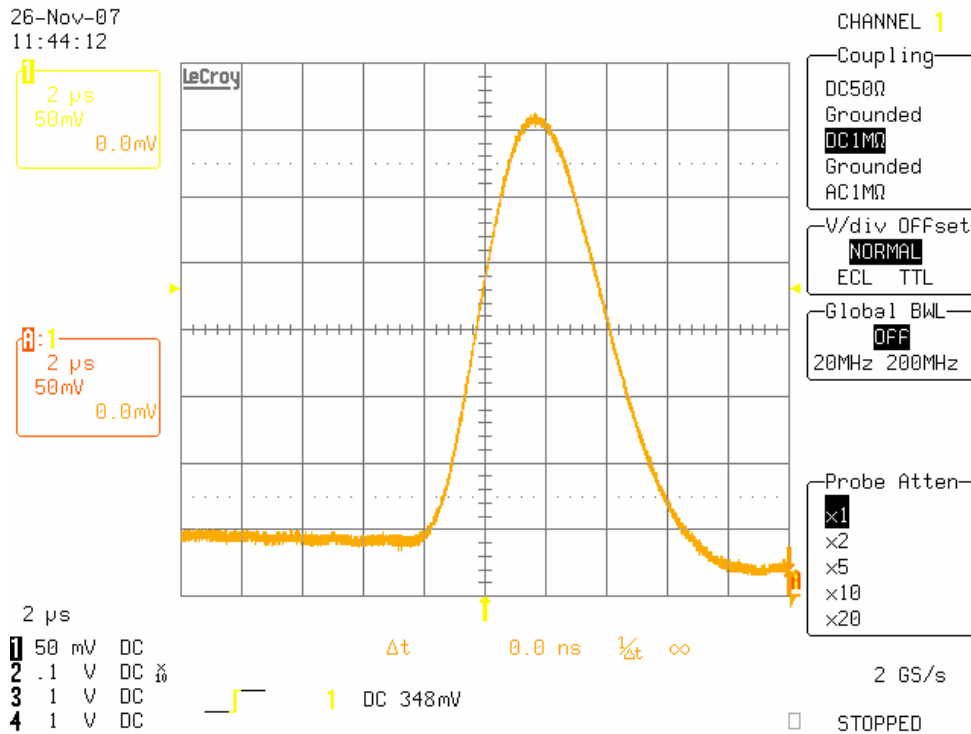
Figure 6.8: Output signal from the ASIC for 4 μs peaking time control input.

By varying the peaking time controls (T_2, T_1) input for four settings defined in the HERMES3 data sheet (Appendix 1), we have achieved the expected signal width at half maximum (0.5 μs, 1 μs, 2 μs and 4 μs respectively) for the output Gaussian signal (Figure 6.5, 6.6, 6.7, 6.8), which denotes that our IP that is responsible for sending out data to ASIC is performing optimally. Also, effective gain (G) was also varied to get the desired effect on the analog output pulse height (Fig. 6.9, 6.10).

To test the 'Serial Data Out' (SDO) IP, we have also used the tail pulse generator to simulate the detector events to calibrate the ASIC. Serial data stream was sent to the ASIC via SDI port along with all other digital control bits using the abovementioned procedure (Sec. 6.3). Then, we have recorded the total number counts (coming via SDO port) for three comparators for 100 milliseconds integration time. After that the counting time was increased at steps of 100 ms up to 90000 milliseconds. TCL scripting language was used to automate the whole process. Plot 6.1 shows number of counts along with the integration time. A linear fit with excel with correlation coefficient r = 1 shows clear linear relationship between the integration time and number of counts, which denotes that 'Serial Data Out' is communicating with the ASIC properly and performing optimally.

49
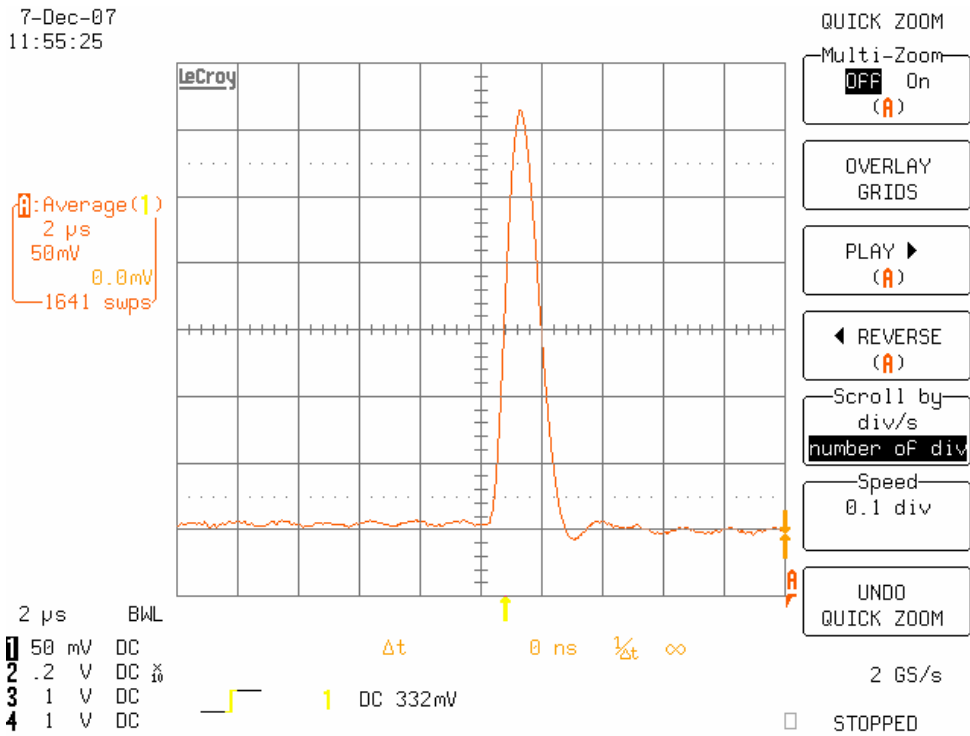
Figure 6.9: Analog output pulse for Gain =1.5 V/fc for 1 μs shaping time.
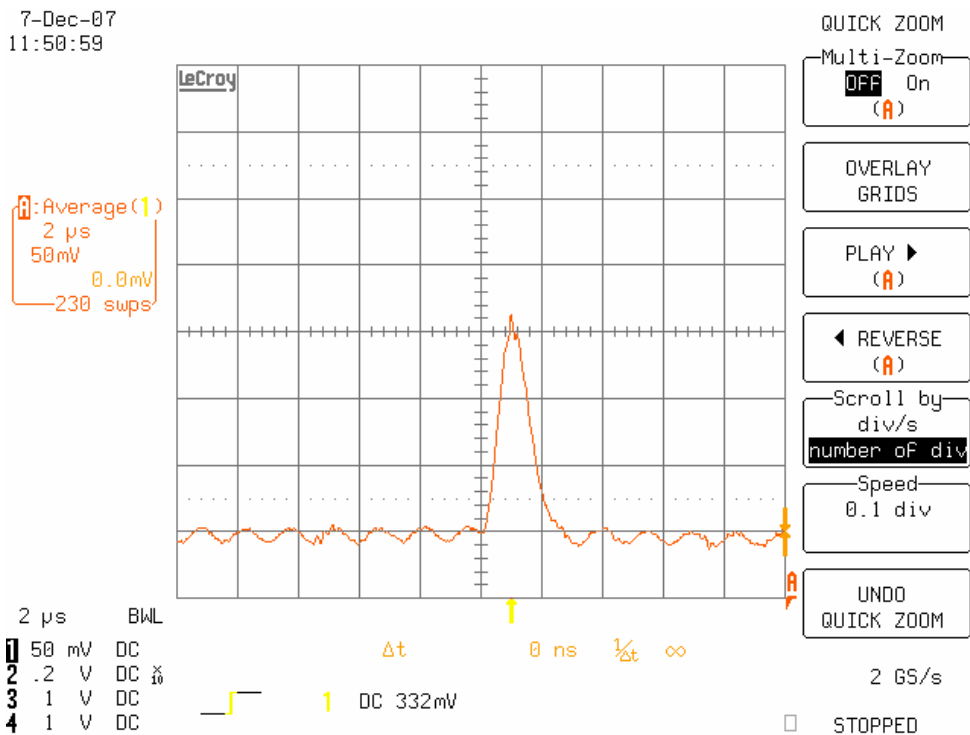
Figure 6.10: Analog output pulse for Gain =0.75 V/fc for 1 μs shaping time. Pulse height is half of the height of Fig. 6.9.

Number of counts vs. integration time

$y = 0.9608x + 0.2553$
$R^2 = 1$
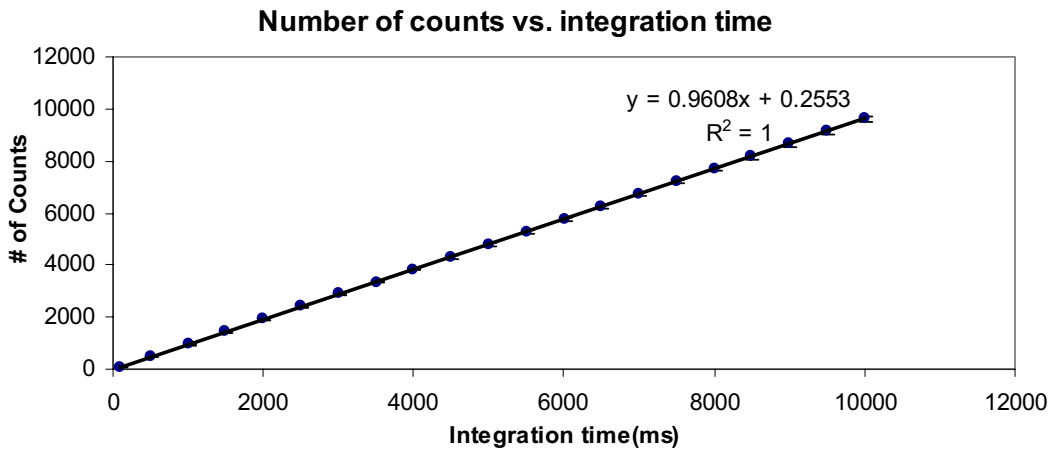
Figure 6.11: Plot of number counts vs. counter integration time. Data was taken for all 32 channels and 3 comparators. Comparator values were set in the middle. Only a single channel and comparator value is shown as they were all identical as window comparators were disabled. A linear regression was performed with Excel. The correlation coefficient (r) was determined as 1 which suggests that total count number is strongly dependent on integration time. From the tabulate values of correlation coefficient [31], we see that our correlation coefficient (r = 1) exceeds the tabulated r value even at p=0.001. This denotes that there is a very highly significant positive correlation between the integration time and their respective number of counts. Also, it suggests that the correlation probability and confidence level is about 99.99% [32].

Comparison between exptected and detected counts

♦ Expected # of counts

● Detected # of counts
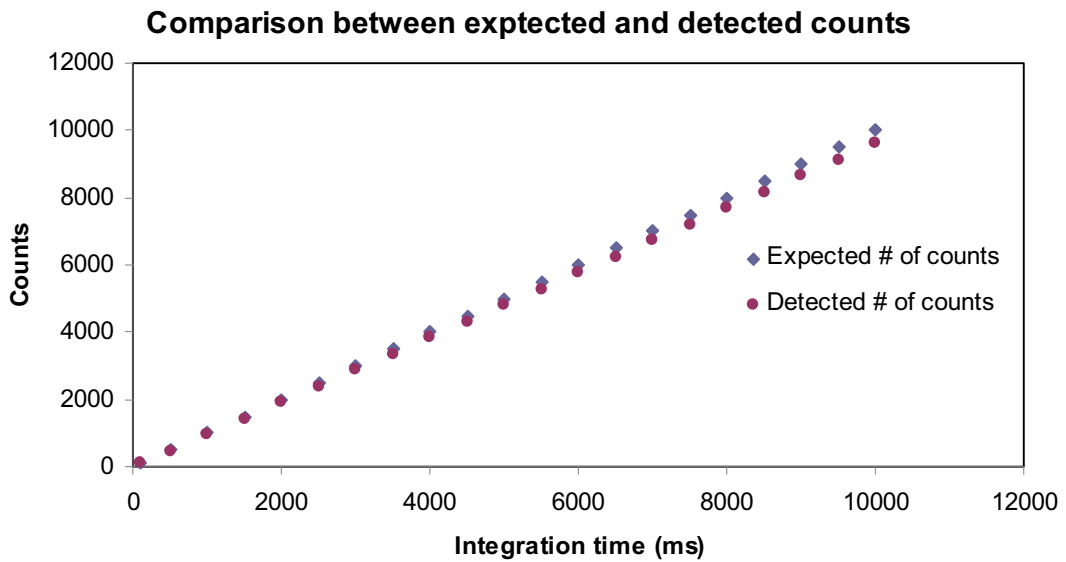
Figure 6.12: This plot shows the linear deviation of the experimental data from the theoretical count values (as expected from a 1 KHz pulse generator), which suggests the existence of small systematic residual error. About 96% of the total events were detected by the counter.

6.5   Calibration of the window comparators

The HERMES3 ASIC has two individual DAC settings for three comparators (two window and one threshold). There are five 10-bit DACs (Vl0, VL1, VH1, Vh2, and VL2) to control the threshold comparator voltages for all the channels, while there are also five 6-bit DACs to control the individual threshold offset for each channel. As the threshold window works only as a high pass filter, its calibration is not necessary. The other window comparators with their individual offset for channels are required to be calibrated for optimum performance. At first, the lower threshold of the window 1 (VL1) was scanned for the entire counting range (245 to 295 counts or 0.538 to 0.648 volts) and the count data was recorded for all the channels (Fig. 6.9). The individual offsets for each channel were set in the middle of the range to get the maximum number of counts. From the plot of count number for each channel vs. VL1 counts, the median value for the counts were chosen and corresponding SDAC values (x intercept) were calculated for each channel. The average of the SDAC values were selected as our reference VL1 voltage (273 counts) and all other channel offsets were adjusted in respect to that value to calibrate lower threshold of window 1 for all the channels (Fig. 6.10). The threshold dispersion were reduced from 70 e$^-$ rms to 2 e$^-$ rms by using our correction for lower threshold value obtained by adjusting individual channel offsets for VL1. Also, a full scan of individual channel offsets was also performed for the corrected coarse threshold counts value (Fig. 6.14).
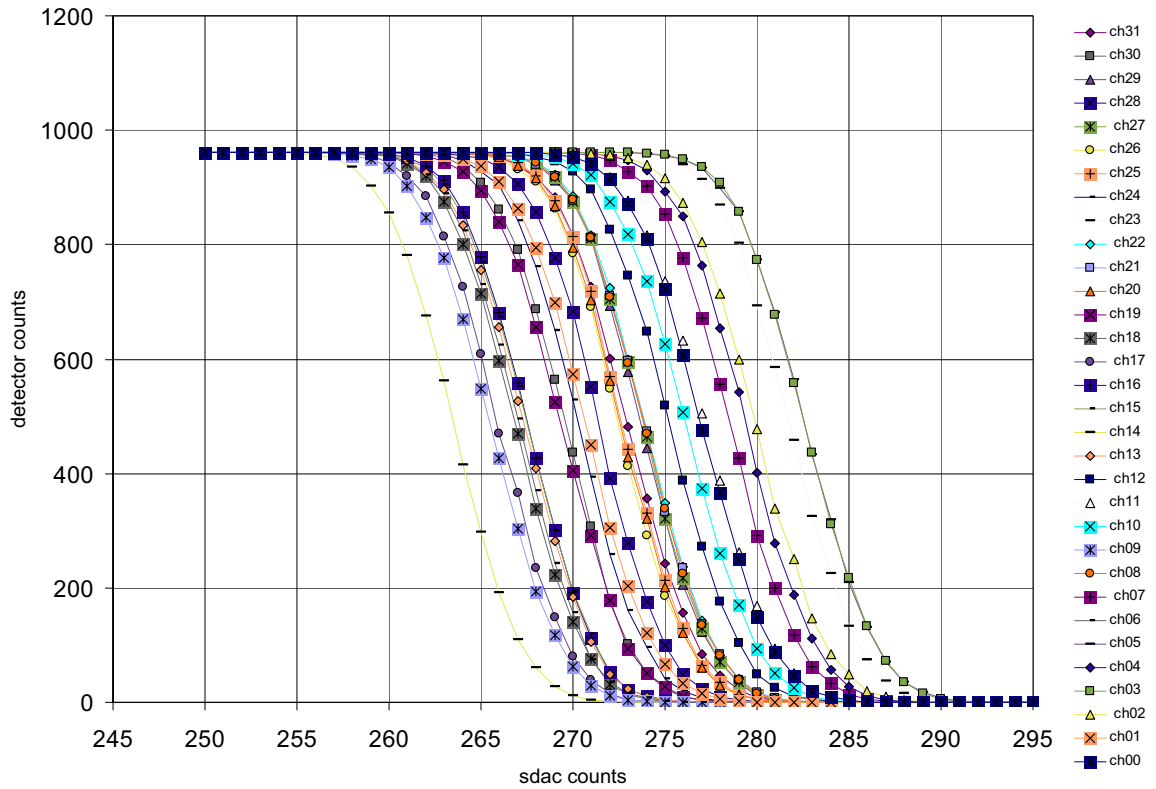
Figure 6.13: Plot of detector counts vs. SDAC voltage (VL1), expressed here in number of counts. Individual channel DAC voltages are uncalibrated, which results the spread in the data.
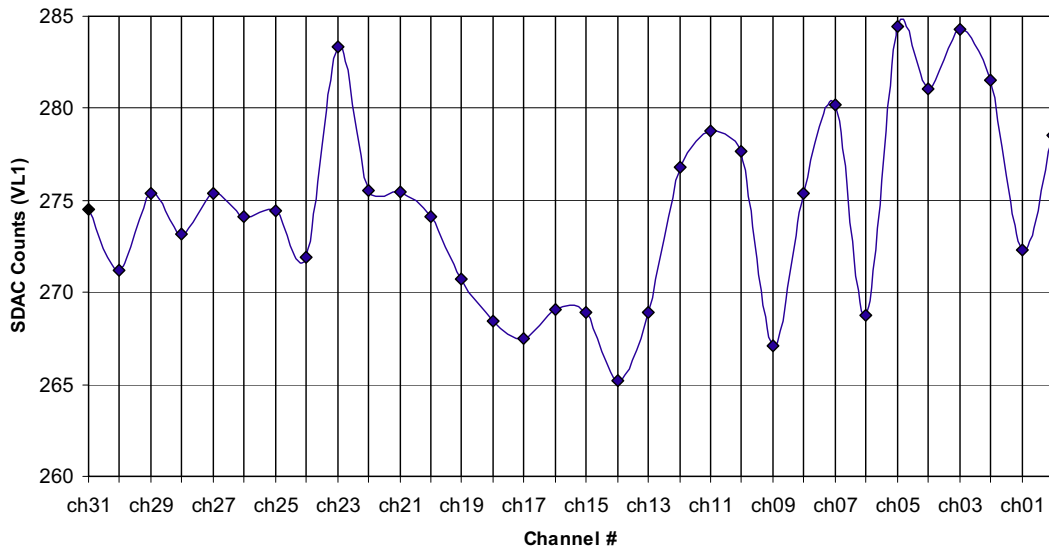
Figure 6.14: Uncorrected SDAC intercept dispersion, plotted here for all 32 channels. Threshold deviation for VL1 was recorded as 70 e⁻ rms.
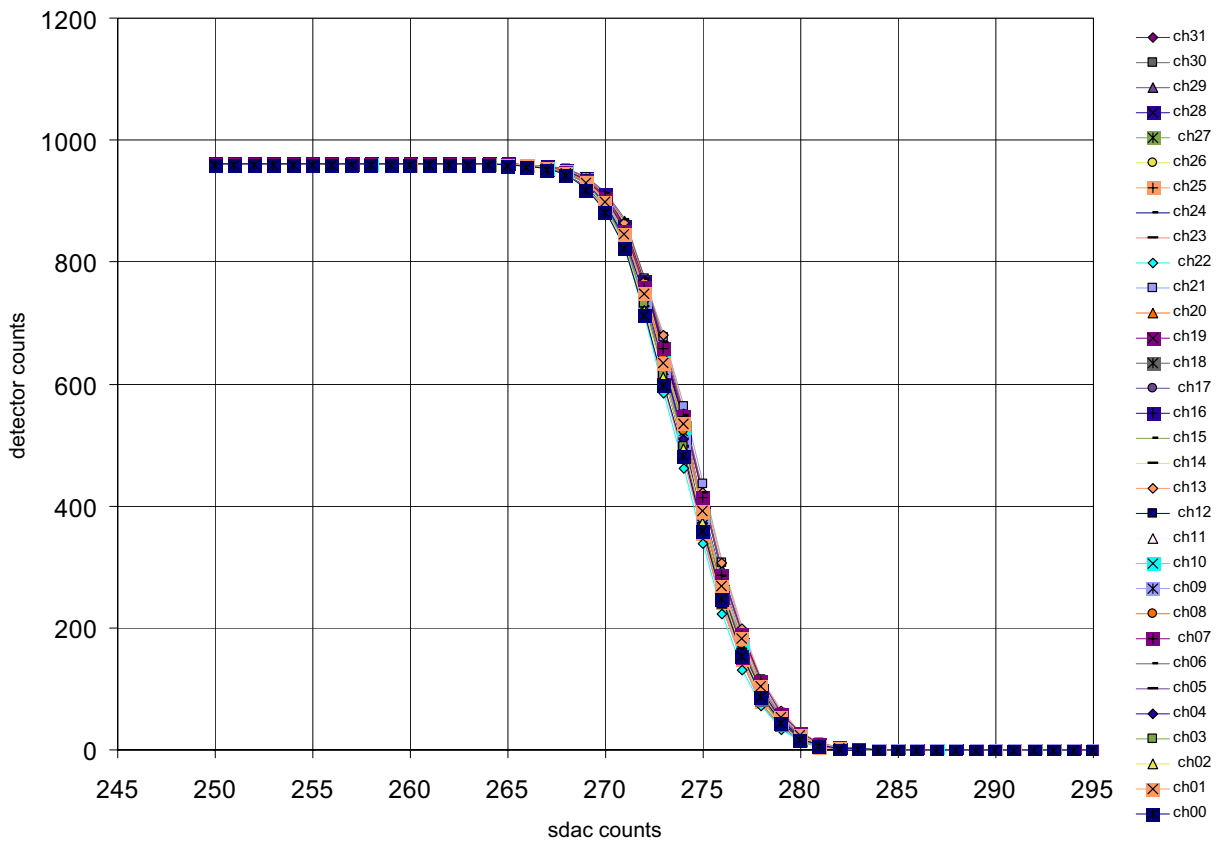
Figure 6.15: Plot of detector counts for all 32 channels vs. SDAC counts. In this plot, individual 6-bit DACs for each channel threshold ware adjusted in respect to channel 14 to reduce the dispersion, which results calibrated count data for window 1 lower threshold (VL1).
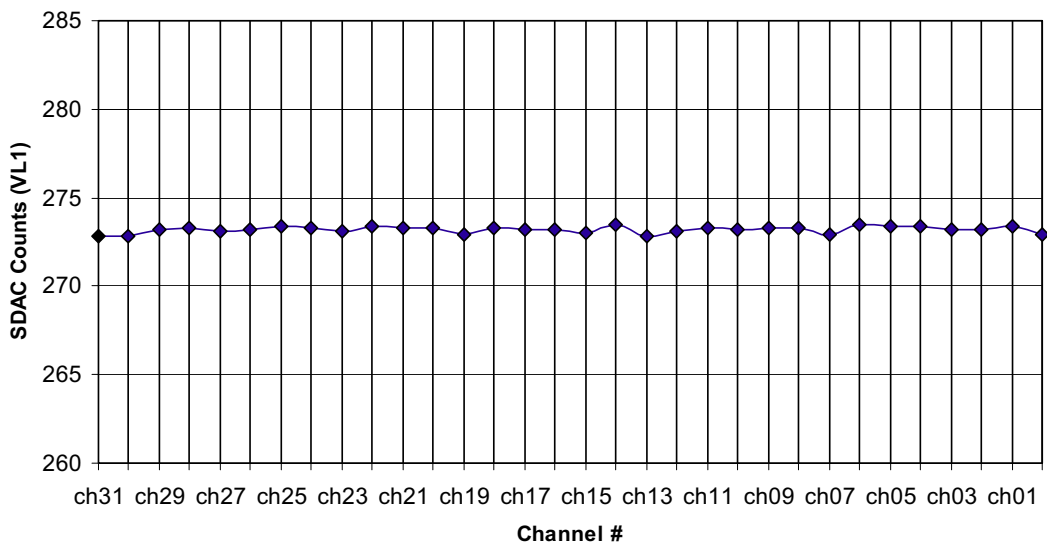
Figure 6.16: Corrected SDAC intercept dispersion, plotted here for all 32 channels. The data shows the properly calibrated individual offsets for each channel. Deviation decreased to about 2e⁻ rms.
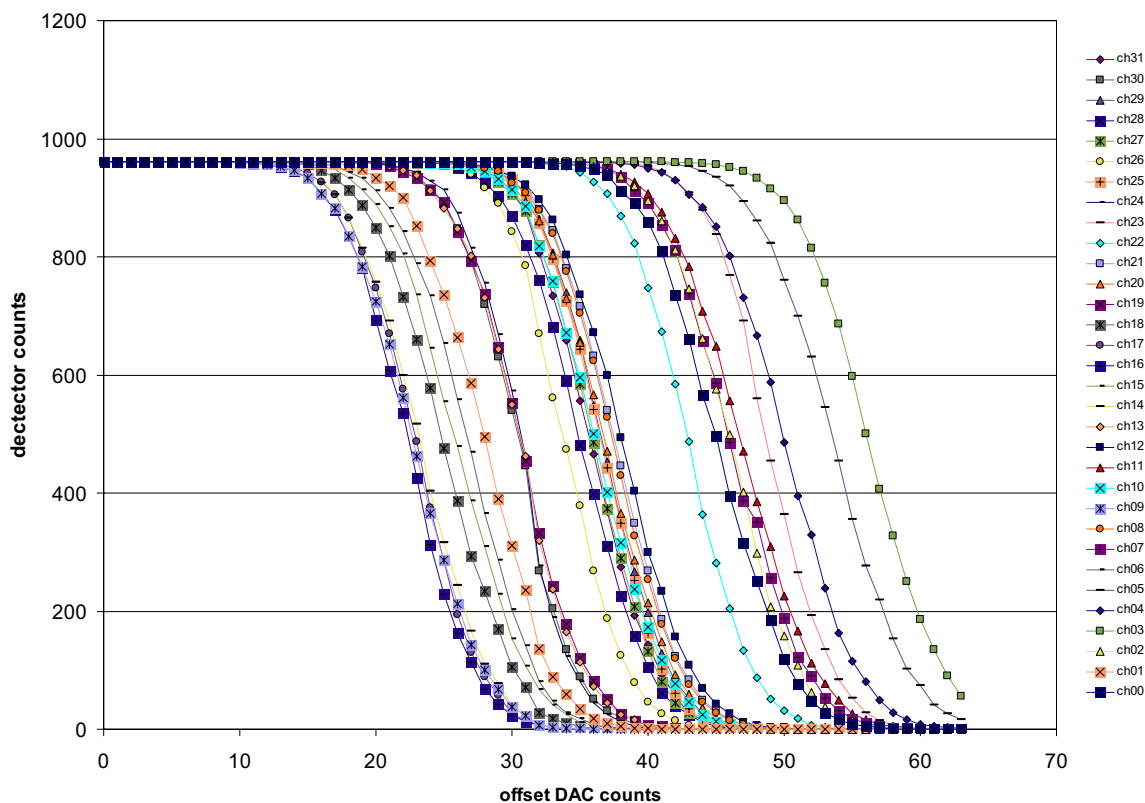
Figure 6.17: Plot of detector counts vs. VL1 offset count ranges for all channels for referenced SDAC value (273 counts) before correction. The spread shows the need for individual channel adjustment.

6.7   Conclusion

The test results presented above suggests that we have successfully developed an FPGA based control system for the HERMES3 detector ASIC, which will be used in powder diffraction studies for synchrotron radiation sources. Three custom designed peripherals (intellectual property cores) were designed to send serial data to the ASIC, read serial data back from ASIC and generate a time base from 1 to 10^6 milliseconds respectively. Also, the calibration of one of the window comparators (VL1) by tweaking the individual voltage threshold for each channel denotes that similar calibration can be performed for all other thresholds (VH2, VL2, VH1), and at that point the detector will be fully calibrated for any threshold voltage. Future evaluations can be performed by varying the peaking time controls and analyzing its effect on the individual channel calibration.

Reference

1. H. Wiedemann. *Synchrotron Radiation.* Springer, New York, 2003.

2. F.R. Elder, A.M. Gurewitsh, R.V. Langmuir, H.C. Pollock, Phys. Rev. 71: 829, 1947.

3. O. Heaviside. Nature, 67: 6, 1902.

4. V. A. Bordovitsyn. *Synchrotron Radiation Theory and Its Development.* World Scientific, New Jersey, 1999.

5. I.M. Ternov, V.V. Mikhailin, V.R. Khalilov. *Synchrotron Radiation and Its Application.* Harwood Academic Press, London, 1985.

6. D.H. Bilderback, P. Ellaume, E. Weckert. Review of third and next generation synchrotron light sources. *J. Phys. B.* 38, S773-S797, 2005.

7. B. D. Cullity, *Elements of X-ray Diffraction.* 2nd edition, Addison-Wesley, MA, 1978.

8. R. Jenkins, R. Snyder. *Introduction to X-ray Powder Diffractometry*, John Wiley & Sons, New York, 1996.

9. W. L. Bragg. The Specular Reflexion of X-rays. *Nature* **90**: 410, 1912.

10. W.H. Zachariasen, *Theory of X-ray diffraction in Crystals,* John Wiley & Sons, New York, 1945.

11. Dept of Chemistry, University of Wisconsin, *Lecture notes on X-ray crystallography and structure determination,* obtained from http://www.chem.uwec.edu/Chem406_F06/Pages/lectnotes.html#lecture7

12. P. Debye, O. Scherr. Inference of X-rays: employing amorphous substances. *Phys. Z.* 17, 277-283, 1916.

13. A. Hull. A new method of X-ray crystal analysis. *Phys. Rev.* 10, 661-696, 1917.

14. D.L. Bish, J.E. Post, (ed). *Modern Powder Diffraction*. Mineralogy Soc. of America, Washington, DC, 1989.

15. C. Giacovazzo (ed). *Fundamentals of Crystallography*. Oxford University Press, New York, 1992.

16. A. Guinier. *X-ray Crystallographic Technology.* Hilger & Watts, London, 1952.

17. D.P. Siddons, S.L. Hulbert, P.W. Stephens. A Guinier Camera for SR Powder Diffraction: High Resolution and High Throughput. *AIP Conference Proceedings,* 879, 1767-1770, 2006.

18. University of Johannesburg, electronic thesis & dissertation archive, *Rowland Circle*, obtained from
www.etd.rau.ac.za/theses/available/etd-04192004-120219/restricted/
finecontentsappendix.pdf

19. G. De Geronimo, P. O'Connor, R.H. Beuttenmuller, Z. Li, A.J Kuczewski, D.P. Siddons. Development of a high-rate high-resolution detector for EXAFS experiments. , *IEEE Trans. Nucl. Sci*, 50, 885-891, 2003.

20. G.F. Knoll, *Radiation Detection and Measurement.* John Wiley & Sons, New York, 1999.

21. G. Dearnaley, D.C. Northrop. *Semiconductor Counters for Nuclear Radiations.* 2nd ed., Wiley, New York, 1966.

22. R. S. Gilmore, *Single Particle Detection and Measurement,* CRC Press, New York, 1992

23. S. Brown, J. Rose. Architecture of FPGAs and CPLDs: A Tutorial. *IEEE Design and Test of Computers*, Vol. 13, No. 2, 42-57, 1996.

24. A.K. Sharma. *Programmable Logic Handbook*. Mcgraw-Hill, New York, 1998.

25. W. Kleitz, *Digital Electronics: A Practical Approach.* Prentice Hall, New Jersey, 2002.

26. J.J. Rodriguez-Andina, M.J. Moure, M.D. Valdes. Features, Design Tools, and Application Domains of FPGAs. *IEEE Trans. Indust. Elec,* Vol. 54, No. 4, 2007.

27. Z. Salcic, A. Smailagic. *Digital System Design and Prototyping using Field Programmable Logic.* Kluwer Academic Publishers, Boston, MA, 2000.

28. Virtex-4 Data Sheet, *Xilinx Corporation*, San Jose, California, 2007.

29. P. Sedcole, B. Blodget, T. Becker, J. Anderson, P. Lysaght. Modular dynamic reconfiguration in Virtex FPGAs. *IEEE Proc.-Comput. Digit. Tech.,* Vol. 153, No. 3, 157-16, 2006.

30. G. De Geronimo, *HERMES3 Data sheet*. Instrumentation Division, Brookhaven National laboratory, Upton, NY.

31. University of Edinburg, *Correlation Coefficient, r table*, obtained from
http://helios.bto.ed.ac.uk/bto/statistics/table6.html#Correlation%20coefficient

32. Clemson University dept of physics, *Linear Regression and Excel*, obtained from
http://phoenix.phys.clemson.edu/tutorials/excel/regression.html

Appendix A:

HERMES3 Data Sheet – Version 3

Technology : *CMOS 0.35μm 2-poly, 4-metal*
Channels : *32*
preamplifier / high order shaper
settable gain: *0.75V/fC, 1.5V/fC*
settable peaking time: *0.5μs, 1μs, 2μs, 4μs*
two window-comparators and one threshold comparator
comparator thresholds controlled by five 10-bit DACs
comparator thresholds individually adjustable (6-bit DAC)
24-bit counter for each comparator
Power per channel : $\approx$ *8mW*
Serial interface.
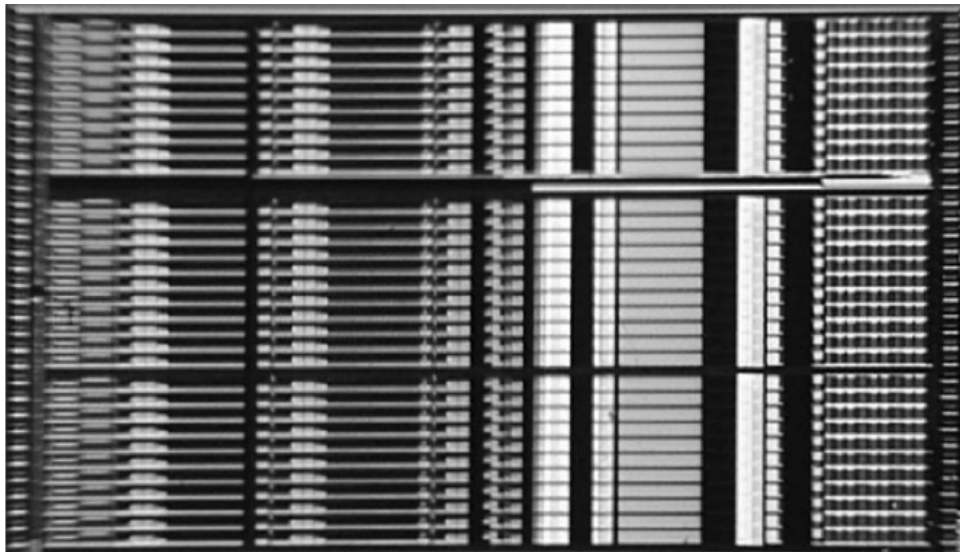
Figure 2: Photo of 32 channel HERMES3 ASIC

LOGIC

Chip select via CS
Chip is selected when CS (chip select) is high.
If SDAC low, WR low sets write mode in standard register, WR high sets read mode.

If SDAC low, transition high to low of CS resets the counters (internal control CRST) and sets ELK and EAN (internal control SAN).
If SDAC high, writing to 10-bit DACs is enabled (independently of WR).

Chip select via token
Chip selection can also be performed via token by connecting TO (token out) to CS of the next chip.

Enabling Counters and data SDI/SDO interface
Counters are enabled/disabled through EN.
Read/write mode is selected through WR.
In write mode (WR=0) SCK and SDI are enabled through internal control WE (write enable).
In read mode (WR=1) SCK and SDO (tristated) are enabled through internal control RE (read enable).
If CS is low, WE and RE are forced to low.

SUPPLIES
Vdd   +3.3V analog
Vddb   +3.3V analog common bias
Vddp   +3.3V analog input MOSFET
Vddd   +3.3V digital
Vss   0V analog
Vssb   0V analog common bias
Vssd   0V digital
Vsubd   0V substrate digital

ANALOG INPUTS
I0-I31   inputs
CAL   calibration input
VL0   low threshold, window 0
VL1   low threshold, window 1
VH1   high threshold, window 1
VL2   low threshold, window 2
VH2   high threshold, window 2

ANALOG OUTPUTS
O0-O31   analog outputs
GUARD   reproduces input voltage for guard biasing, on input side
OAN   analog output
OLK   leakage current monitor

ANALOG CONTROLS (self biasing)
BLK   internal leakage current control
BLN   output baseline control
BDAC   DAC range control

DIGITAL INPUTS
SCK   clock input   positive edge shifts in write mode
                             negative edge shifts in read mode (latch on positive edge)
                             clock must be idling high

SDI (SDAC low)	904 bit serial input; first bit written at first clock (positive edge), ending to Ch31; 0,SDA0,SDA1,SDA2,EBLK,T1,T2,G,[ECH,ECAL,ELK,EAN,D0 - D5/VL1,D0-D5/VH1,D0-D5/VL2,D0-D5/VH2]x32 first bit of data stream is D5/VH2 of Ch31, last bit of data stream is 0.	
SDI (SDAC high)	50 bit serial input; first bit written at first clock (positive edge), ending to VL0 DAC; P0:P9(VH2), P0:P9(VL2), P0:P9(VH1), P0:P9(VL1), P0:P9(VL0); first bit of data stream is P9(VL0), last bit of data stream P0(VH2).	
CS	chip select	1 active
	if SDAC low, CS transition 1 to 0 resets the counters and sets EAN and ELK	
EN	counters enable/disable	EN=1 enables the counters
		EN=0 disables the counters
WR	write/read mode	WR=1 read mode
		WR=0 write mode
RST	global reset	0 active
	OR-ed to internal power-on reset	
SDAC	enables writing to 10-bit DACs (independent of WR)	

DIGITAL OUTPUTS (clocked)

SDO	2304 bit counter serial output; first bit available before first clock, starting from Ch31 [C23-C0/W0, C23-C0/W1, C23-C0/W2]x32, first bit of data stream is C23/W0 of Ch31, last bit of data stream is C0/W2 of Ch0.

DIGITAL CONTROLS

EBLK	internal bias leakage enable	0 disable (default)
		1 enable
T2,T1	peaking time controls	00 1μs (default)
		01 0.5μs
		10 4.0μs
		11 2.0μs
SDA2:SDA0	enables DAC monitors to OAN	000 disabled (default)
		001 DAC VL0
		010 DAC VL1
		011 DAC VH1
		100 DAC VL2
		101 DAC VH2
P9:P0	10-bit DAC setting	0:0 ~300mV (default)
		1:1 ~ 2.25V
G	gain control	0 1500mV/fC
		1 750mV/fC
ECH	channel enable	0 enable
		1 disable
ECAL	calibration input enable	1 enable

		0	disable
ELK	leakage current monitor enable	1	enable
		0	disable
EAN	analog output enable	1	enable
		0	disable
E4T	ch.4 EAN, ECAL, ELK enable	1	enable
		0	disable

also forces T2 high (peaking time to 4μs from 1μs default)

Appendix B:

TCL scripting language was used to send the instruction dataset for communicating with the HERMS3 ASIC via the IP cores for testing and evaluation. The codes are shown as follows; comments are followed by a '#' sign.

```
#########################################################################
###########
## XMD Connect
##
##  host    IP address of XMD server
##  port    port number of XMD server (4731)
##
##  returns server socket
#########################################################################
###########
proc xmd_connect {host port} {
  global s
  set s [socket $host $port]
  fconfigure $s -buffering none -translation binary -blocking 1
  return $s
}

#########################################################################
###########
## Check ASIC Status
##
##   addr  address of status
##
##   returns when status is 0 else loops
#########################################################################
###########
proc ck_stat {addr} {
  global s
  set done 1
  while {$done} {
   puts -nonewline $s mrd\x20$addr\n
   scan [gets $s] "%*s%x" done
  }
 return done
}

#########################################################################
###########
## XMD Command
##
##   cmd     XMD native command
```

```
##    chan   output channel (e.g. null, stdout, stderr, or open file
channel)
##    returns ouput line from server
########################################################################
###########
proc xmd {cmd chan} {
  global s
  puts -nonewline $s $cmd\n
  flush $s
  set line [string trim [gets $s]]
  if {$chan != -1 && $line != ""} {puts -nonewline $chan [format "%s\n"
$line]}
  return $line
}


########################################################################
###########
## Read ASIC
##
##    addr   addr (hex)
##    returns value (dec)
########################################################################
###########
proc rd {addr} {
    scan [xmd mrd\x20$addr -1] "%*s%x" value
  return $value
}


########################################################################
###########
## Write ASIC
##
##    addr   addr (hex)
##    value  (hex or dec)
########################################################################
###########
proc wr {addr val} {
    xmd "mwr\x20$addr $val" -1
  return
}


########################################################################
###########
## Write Chip Select Bits to ASIC
##
##    val  chip select bits (hex)
########################################################################
###########
proc chip_sel {val} {
    xmd "mwr 0x41400010 $val" -1
  return
}


########################################################################
###########
## Gating Timer
##
```

```
##     value   time in milliseconds (hex or dec)
########################################################################
###########
proc gate_time {val} {
  wr 0x41300000 $val
  wr 0x41300004 0x80000000
  ck_stat 0x41300004
  return
}


########################################################################
###########
## Send SPI Bits
##
##     value   (hex or dec)
##     cnt     number of bits
########################################################################
###########
proc snd_spi {val cnt} {
  wr 0x41400000 [leftjust $val $cnt]
  wr 0x41400004 [expr {$cnt-1}]
  wr 0x41400008 0x80000000
  ck_stat 0x41600008
  return
}


########################################################################
###########
## Receive SPI Bits
##
##     cnt   number of bits
##     returns value (dec)
########################################################################
###########
proc rcv_spi {cnt} {
  wr 0x41400004 [expr {$cnt-1}]
  wr 0x41400008 0x80000000
  ck_stat 0x41400008
  return [rd 0x4140000c]
}


########################################################################
###########
## decimal to hex conversion
##
##     value   in dec or hex
##     returns value in hex
########################################################################
###########
proc dec2hex {value} {
 set p [string first "." $value]
 if {$p >= 1} {
  set value [string range $value 0 [expr {$p-1}]]
  return [format "0x%08x" [expr $value]]
 } else {
  return [format "0x%08x" [expr $value]]
 }
```

```
}

#########################################################################
###########
## Left Justify to a 32 bit HEX Number
##
##    value  in dec or hex
##    cnt    number of bits to left justify
##    returns value in hex
#########################################################################
###########
proc leftjust {value cnt} {
 if {[string first "0x" [string tolower $value 0 1]] < 0} {
  return [dec2hex  [expr {pow(2,[expr {32 - $cnt}]) * $value}]]
 } else {
  return [dec2hex [expr {pow(2,[expr {32 - $cnt}]) * [format "%d"
$value]}]]
 }
}

#########################################################################
###########
## Send Channel Setup
##
##    vh2,vh2,vh1,vl1  6 bit offset dacs (dec or hex)
##    enb              enable bits in hex (|EAN|ELK|ECAL|*ECH|)
#########################################################################
###########
proc chan_setup {vh2 vl2 vh1 vl1 enb} {
 snd_spi [dec2hex [expr {[leftjust $vh2 10] + [leftjust $vl2 16] +
[leftjust $vh1 22]+ [leftjust $vl1 28] + [leftjust $enb 32]}]] 28
}

#########################################################################
###########
## read counts from 32 channels of HERMES3
##
##    chan  output channel (e.g. null, stdout, stderr, or open file
channel)
#########################################################################
###########
proc rd_cnts {chan} {
  global s
  set n 1
  set val [rcv_spi 23]
  if {$chan != -1} {puts -nonewline $chan [format "%02d\t%08d\t" 31
$val]}
  for {set i 1} {$i<96} {incr i} {
  set val [rcv_spi 24]
   if {$chan != -1} {
    if {[incr n] >=3} {
     set n 0
     puts -nonewline $chan [format "%08d\n" $val]
    } elseif {$n == 1} {puts -nonewline $chan [format "%02d\t%08d\t"
[expr {31 - $i/3}] $val]
    } else {puts -nonewline $chan [format "%08d\t" $val]}
   }
```

```
  }
  return
}

################################################################
###########
## read calibration counts from 32 channels of HERMES3
##
##   nsec   gate time in seconds
##   chan   output channel (e.g. null, stdout, stderr, or open file
channel)
################################################################
###########
proc calib_cnts {nsec chan} {
  global s
  global tot_cnt
  set n 1
  set val [rcv_spi 23]
  for {set i 1} {$i<96} {incr i} {
  set val [rcv_spi 24]
   if {$chan != -1} {
    if {[incr n] >=3} {set n 0
    } elseif {$n == 2} {puts -nonewline $chan [format "%08.2f\t" [expr
{double($val) / $nsec}]]}
   }
  }
}


################################################################
###########
## Start XMD client session (main)
################################################################
###########

set s [xmd_connect "130.199.194.131" "4731"]

# CHIPSEL <- |RST=0|EN=0|*WR=1|SDAC=0|...|CS0=0| - 32 bits
chip_sel 0x20000000

# CHIPSEL <- |RST=0|EN=0|*WR=0|SDAC=0|...|CS0=0| - 32 bits
chip_sel 0x00000000

# CHIPSEL <- |RST=0|EN=0|*WR=0|SDAC=1|...|CS0=1| - 32 bits
chip_sel 0x10000001

# Send 50 bits to setup HERMES3 SDACS
<VL0[p9..p0]><VL1[p9..p0]><VH1[p9..p0]><VL2[p9..p0]><VH2[p9..p0]> - 50
bits

set sdac_vl0 100
set sdac_vl1 278
set sdac_vh1 1023
set sdac_vl2 276
set sdac_vh2 1023

snd_spi  $sdac_vl0 10
```

```
snd_spi  $sdac_vl1 10
snd_spi  $sdac_vh1 10
snd_spi  $sdac_vl2 10
snd_spi  $sdac_vh2 10

# CHIPSEL <- |RST=0|EN=0|*WR=0|SDAC=0|...|CS0=1| - 32 bits
chip_sel 0x00000001

# send setup to HERMES3 ((32*28)+8 = 904 bits)

# channels (31..0):
<VH2[d5..d0]><VL2[d5..d0]><VH1[d5..d0]><VL1[d5..d0]><ENB:|EAN|ELK|ECAL|*
ECH|> - 28 bits per channel

# setup channels 31..0 ----------->
#
#         <VH2><VL2><VH1><VL1><ENB>
#ch31:
chan_setup  32   32   32   32   0x2
#ch30:
chan_setup  32   32   32   32   0x2
#ch29:
chan_setup  32   32   32   32   0x2
#ch28:
chan_setup  32   32   32   32   0x2
#ch27:
chan_setup  32   32   32   32   0x2
#ch26:
chan_setup  32   32   32   32   0x2
#ch25:
chan_setup  32   32   32   32   0x2
#ch24:
chan_setup  32   32   32   32   0x2
#ch23:
chan_setup  32   32   32   32   0x2
#ch22:
chan_setup  32   32   32   32   0x2
#ch21:
chan_setup  32   32   32   32   0x2
#ch20:
chan_setup  32   32   32   32   0x2
#ch19:
chan_setup  32   32   32   32   0x2
#ch18:
chan_setup  32   32   32   32   0x2
#ch17:
chan_setup  32   32   32   32   0x2
#ch16:
chan_setup  32   32   32   32   0x2
#ch15:
chan_setup  32   32   32   32   0x2
#ch14:
chan_setup  32   32   32   32   0x2
#ch13:
chan_setup  32   32   32   32   0x2
#ch12:
chan_setup  32   32   32   32   0x2
```

```
#ch11:
chan_setup  32   32   32   32   0x2
#ch10:
chan_setup  32   32   32   32   0x2
#ch09:
chan_setup  32   32   32   32   0x2
#ch08:
chan_setup  32   32   32   32   0x2
#ch07:
chan_setup  32   32   32   32   0x2
#ch06:
chan_setup  32   32   32   32   0x2
#ch05:
chan_setup  32   32   32   32   0x2
#ch04:
chan_setup  32   32   32   32   0x2
#ch03:
chan_setup  32   32   32   32   0x2
#ch02:
chan_setup  32   32   32   32   0x2
#ch01:
chan_setup  32   32   32   32   0x2
#ch00:
chan_setup  32   32   32   32   0x2

# |G|T2|T1|EBLK|SDA2|SDA1|SDA0|0| - 8 bits
snd_spi  0x70 8

# throw away first gate time of 500 ms after ASIC setup is run
chip_sel 0x00000000
gate_time 500
chip_sel 0x20000001

# CHIPSEL <- |RST=0|EN=0|*WR=0|SDAC=1|...|CS0=1| - 32 bits
chip_sel 0x10000001

# Send 50 bits to setup HERMES3 SDACS
<VL0[p9..p0]><VL1[p9..p0]><VH1[p9..p0]><VL2[p9..p0]><VH2[p9..p0]> - 50
bits


puts -nonewline [format "sdac\tch31\tch30\tch29\tch28\t\
                        ch27\tch26\tch25\tch24\tch23\t\
                        ch22\tch21\tch20\tch19\tch18\t\
                        ch17\tch16\tch15\tch14\tch13\t\
                        ch12\tch11\tch10\tch09\tch08\t\
                        ch07\tch06\tch05\tch04\tch03\t\
                        ch02\tch01\tch00\n"]

set gtime 50000
for {set k 250} {$k<=295} {incr k} {
#for {set j 0} {$j<96} {incr j} { set tot_cnt($j) 0}

# CHIPSEL <- |RST=0|EN=0|*WR=0|SDAC=1|...|CS0=1| - 32 bits
chip_sel 0x10000001
set sdac_vl0 100
set sdac_vl1 $k
```

69

```
set sdac_vh1 1023
set sdac_vl2 276
set sdac_vh2 1023

snd_spi  $sdac_vl0 10
snd_spi  $sdac_vl1 10
snd_spi  $sdac_vh1 10
snd_spi  $sdac_vl2 10
snd_spi  $sdac_vh2 10

puts -nonewline [format "%d\t" $k]

# CHIPSEL <- |RST=0|EN=0|*WR=0|SDAC=0|...|CS0=0| - 32 bits
chip_sel 0x00000000
gate_time $gtime
puts -nonewline stderr "."
# CHIPSEL <- |RST=0|EN=0|*WR=1|SDAC=0|...|CS0=1| - 32 bits
chip_sel 0x20000001
calib_cnts [expr {$gtime / 1000.0}] stdout
puts -nonewline [format "\n"]
flush stderr
}
puts -nonewline stdout "\n"
flush stderr
catch {close $s}
exit




########################################################################
########################################
# HERMES3 Control Bits:
#
# CS   -  chip select 1 active (if SDAC low, CS transition 1 to 0
resets the counters and sets EAN and ELK)
# EN   -  counters enable/disable EN=1 enables the counters, EN=0
disables the counters
# *WR  -  write/read mode *WR=1 read mode, *WR=0 write mode
# *RST -  global reset 0 active, OR-ed to internal power-on reset
# TO   -  token output, to be connected to CS of the next chip
# TCK  -  token clock input positive edge shifts token
# SDAC -  enables writing to 10-bit DACs (independent of *WR)
#
# EBLK internal bias leakage enable:
# 0 disable (default)
# 1 enable
#
# T2,T1 - peaking time controls:
# 00 1s (default)
# 01 0.5s
# 10 4.0s
# 11 2.0s
#
# SDA2:SDA0 - enables DAC monitors to OAN:
# 000 disabled (default)
# 001 DAC VL0
```

```
# 010 DAC VL1
# 011 DAC VH1
# 100 DAC VL2
# 101 DAC VH2
#
# P9:P0 - 10-bit DAC setting:
# 0:0 ~300mV (default)
# 1:1 ~ 2.25V
#
# G - gain control:
# 0 1500mV/fC
# 1 750mV/fC
#
# *ECH - channel enable:
# 0 enable
# 1 disable
#
# ECAL - calibration input enable:
# 1 enable
# 0 disable
#
# ELK leakage - current monitor enable:
# 1 enable
# 0 disable
#
# EAN analog - output enable:
# 1 enable
# 0 disable
########################################################################
```