Design of Readout Electronics for the DEEP Particle Detector

Bendik Husa

Master's thesis in Physics

University of Bergen
Department of Physics and Technology

November, 2021

Acknowledgements

I would like to extend my gratitude to my supervisor Kjetil Ullaland. I am grateful for being given the opportunity to work on this very interesting project, for all the helpful discussions we had, and for receiving good answers to every question I had along the way. I also want to thank Senior Engineer Shiming Yang. Your knowledge of PCB design has been an invaluable resource, and I always left your office with more knowledge than I had when going in. Thank you, Hilde Nesse Tyssøy for very helpful feedback on my thesis, and for providing insight into the world of particle precipitation.

I want to thank my friends on the third floor, Birger, Thomas, Amalie and Håvard for making this year memorable and for all the useful (and not so useful) discussions we had. A big thanks to Tobias Heggeli for proofreading and providing feedback on my thesis. Another thanks to Tarje Hillersøy and Vegard Milde for help and discussion about many aspects of the thesis.

Lastly, I want to thank Heidi for all your support and help throughout this year, and for being who you are.

Abstract

Along with electromagnetic radiation, the Sun also emits a constant stream of charged particles in the form of solar wind. When these particles enter Earth's atmosphere through a process known as particle precipitation, they can through a series of chemical reactions produce NO_x and HO_x gases. These gases are greenhouse gases and deplete the ozone in the mesosphere and upper stratosphere. It is important to quantify the rate of production of these gases to model the potential climate impact. Existing particle detectors in space are suboptimal because they cannot determine the energy flux and pitch angle distribution of precipitating particles. The primary scientific objective of the DEEP project is to design a particle detector instrument that is specifically designed for particle precipitation measurements.

This thesis investigates different data acquisition schemes for handling the signal from a pixel detector. The chosen approach is measuring the width of a shaped pulse to quantify the energy of the particle. Known as Time-over-Threshold, a detector circuit board is designed featuring high-speed comparators as threshold discriminators and the NG-MEDIUM FPGA from NanoXplore to implement the data acquisition.

Digitizing the comparator pulse width is done with a Time-to-Digital converter (TDC) implemented in the FPGA fabric. Since the difference in pulse width is small for different energies, a high conversion resolution is required. Two high-resolution TDCs are designed and compared, both of which feature a digital counter and a method of interpolating the counter clock period. The first interpolation method applies the use of a multitapped delay line implemented with hard carry chain resources, and the second method oversamples the input with several equally off-phase sampling clocks.

A resolution of 302 ps and a differential non-linearity of 3.26 was achieved with the delay line TDC clocked at 100 MHz. An automatic statistical calibration scheme is included to determine the actual delays of the delay line, utilizing a second asynchronous clock to generate uniformly distributed hits. The asynchronous oversampler resolution is clock frequency dependent and provides a 4-fold improvement to the clock period. The differential nonlinearity approaches zero with close matching of the off-phase clocks and operating frequency.

A complete firmware design for the data acquisition and rocket telemetry of the detector is proposed and demonstrated. A simulation of the firmware utilizing each TDC topology is conducted and the delay line TDC is demonstrated to be the most accurate at all operating frequencies and thus the recommended TDC for the DEEP data acquisition.

Contents

Acronyms

ADC Analog-to-Digital Converter.

ASC Andøya Space Centre.

ASIC Application Specific Integrated Circuit.

BCSS Birkeland Centre for Space Science.

BRAM Block RAM.

CERN European Organization for Nuclear Research.

CGB Coarse Grain Block.

CI Carry In.

CKS Clock Generation Switch.

CMOS Complementary Metal–Oxide–Semiconductor.

CO Carry Out.

CSA Charge Sensitive Amplifier.

CSV Comma-separated values.

DAQ Data Acquisition.

DEEP Distribution of Energetic Electrons and Protons.

DLL Delay-Locked Loop.

DNL Differential non-linearity.

DSP Digital Signal Processor.

dToT Dynamic Time-over-Threshold.

EDAC Error Detection and Correction.

ENOB Effective Number of Bits.

EPP Energetic particle precipitation.

FE Functional Element.

FIFO First In First Out.

FMC FPGA Mezzanine Connector.

FPGA Field Programmable Gate Array.

FSM Finite State Machine.

GATE GEANT4 Application for Emission Tomography.

ICI Investigation of Cusp Irregularities.

JTAG Joint Test Action Group.

LDO Low Dropout Linear Regulator.

LSB Least Significant Bit.

LUT Lookup Table.

PCB Printed Circuit Board.

PCM Pulse Code Modulation.

PLL Phase-Locked Loop.

PVT Process, Voltage and Temperature.

RFB Register File Block.

ROM Read Only Memory.

RSE Radiation Shutter Electronics.

SDF Standard Delay Format.

SERDES Serializer-Deserializer.

SMILE Solar wind Magnetosphere Ionosphere Link Explorer.

SNR Signal-to-Noise Ratio.

SPI Serial Peripheral Interface.

SRAM Static RAM.

SSP Single Shot Precision.

TDC Time-to-Digital Converter.

ToT Time-over-Threshold.

VCO Voltage Controlled Oscillator.

VDL Vernier Delay Line.

WFG Waveform Generator.

1 INTRODUCTION

1.1 Scientific objective

The primary scientific objective of the Distribution of Energetic Electrons and Protons (DEEP) project at the Birkeland Centre for Space Science is developing a particle detector that can accurately quantify the energy flux deposited into the atmosphere by the process of Energetic particle precipitation (EPP). Existing particle detectors in orbit are inadequate for determining the amount of particles precipitating into the atmosphere, largely due to the highly anisotropic pitch angle distribution of precipitating electrons requiring a large field of view [Nesse Tyssøy et al., 2016]. The DEEP instrument will include three electron detectors mounted in different angles such that they collectively cover a field of view greater than 180°. This will provide new information about the distribution of particles precipitating into the atmosphere as well as the distribution of particles backscattered by the atmosphere. In addition to this, three proton detectors are included to correct for proton contamination in the electron measurements [Nesse Tyssøy et al., 2017].

The contribution to the DEEP project and primary scientific objective of this thesis is developing a single scaled-down electron detector that is planned to be included in the payload of a sounding rocket launched by Andøya Space Centre (ASC) in 2023. This detector will serve as a proof of concept, and give valuable test information about the performance of the detector that can be used for further development. The thesis scope starts at the concept level, and extends as far as time will permit.

1.2 Energetic particle precipitation (EPP)

In order to understand the processes behind EPP, some background knowledge on the Earth-Sun environment and its dynamics is required. The information in this section is gathered from [Hargreaves, 1992].

1.2.1 Earth's atmosphere

The atmosphere of Earth is a layer of gases that is being confined by the gravitational pull. It is divided into layers based on the temperature gradient. These are, named in order of rising altitude, the troposphere, the stratosphere, the mesosphere and the thermosphere. The density of the atmosphere decreases rapidly with altitude, and since a major part of the energy from the Sun is absorbed in the thermosphere, it is particularly heated, reaching several hundred degrees Celcius. Though the temperature is high, the scarcity of particles means that there is almost no heat transfer. This point is important because it means that the situation is not quite as dire for temperature sensitive circuitry operating in the thermosphere as it might first appear. Because of the low particle density above 100 km from the surface, there is a permanent presence of ions and free electrons. This region of the thermosphere is known as the ionosphere, and it is electrically conductive.

1.2.2 Solar wind

The Sun is the centre of our solar system and its radiation is either directly or indirectly the primary source of all energy on Earth. In addition to the electromagnetic radiation the Sun emits, it also emits a continous stream of charged particle plasma known as solar wind. One property of plasma is that it is electrically conductive, and magnetic fields become frozen in it. Because of this, the magnetic field of the Sun is carried along with the plasma stream and couples with the magnetic field of Earth. The solar wind is cyclic, following an 11-year solar cycle of continuously increasing and waning intensity.

1.2.3 The magnetosphere

The near-Earth space dominated by the Earth's magnetic field is called the magnetosphere. It is formed by the Earth's iron core, but continually reshaped by the varying solar wind. If not for the solar wind, Earth's magnetic field could accurately be modelled as a dipole with north and south poles slightly tilted off from the rotational axis. The solar wind, with its embedded magnetic field, has a large impact on the shape of the magnetic field surrounding earth. The magnetic field is compressed by the solar wind on the side facing the sun, and stretches out on the opposite side, creating a long tail on the night side as shown in Figure 1. If not for the magnetic field, the solar wind would blow away the atmosphere [Ødegaard, 2016, p. 7].

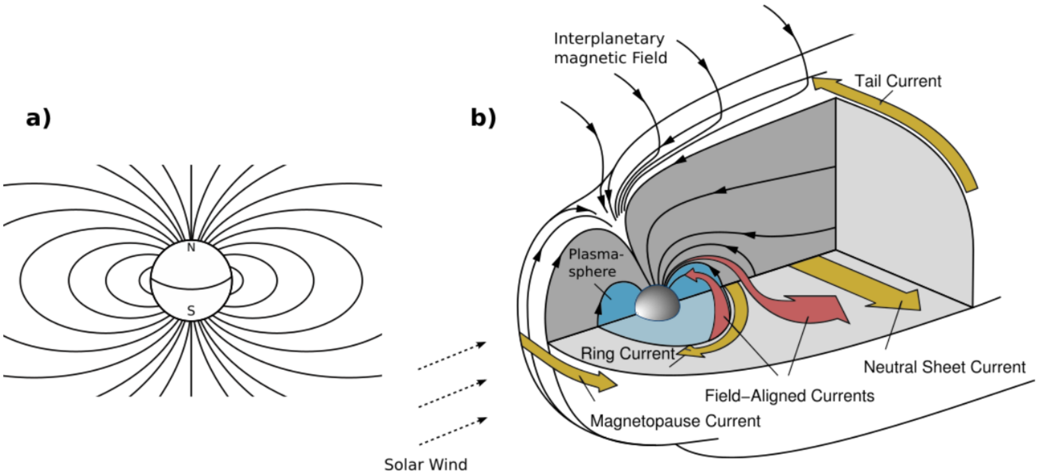


Figure 1: Earth's magnetic field [Ødegaard, 2016, figure 2.1].

1.2.4 Radiation belts

Suspended in the magnetosphere are charged particles, trapped by the influence of the magnetic field. The discovery of the radiation belts starts with the Explorer 1 satellite, launched on 31 January in 1958. The satellite was equipped with a Geiger counter built by James Alfred Van Allen's research group at the University of Iowa. Measurements from this Geiger counter revealed that at certain times the counting rate was so large that it was interpreted as a malfunctioning counter. The same was observed by the satellites Explorer 3 and Sputnik 3, and it was speculated that these particles were electrons of energy around 100-150 keV. With results from Explorer 4, it was determined that the measured results stemmed from protons in a certain region with a kinetic energy above 30 MeV. Pioneer 3, also launched in 1958, discovered a second region of particles at a further distance from earth. This led Van Allen to formulate a theory of particles trapped in two distinct radiation belts in the magnetosphere [Van Allen and Frank, 1959]. Because of this, these regions of trapped particles are also known as the Van Allen radiation belts. These regions of particles can be shown in Figure 2.

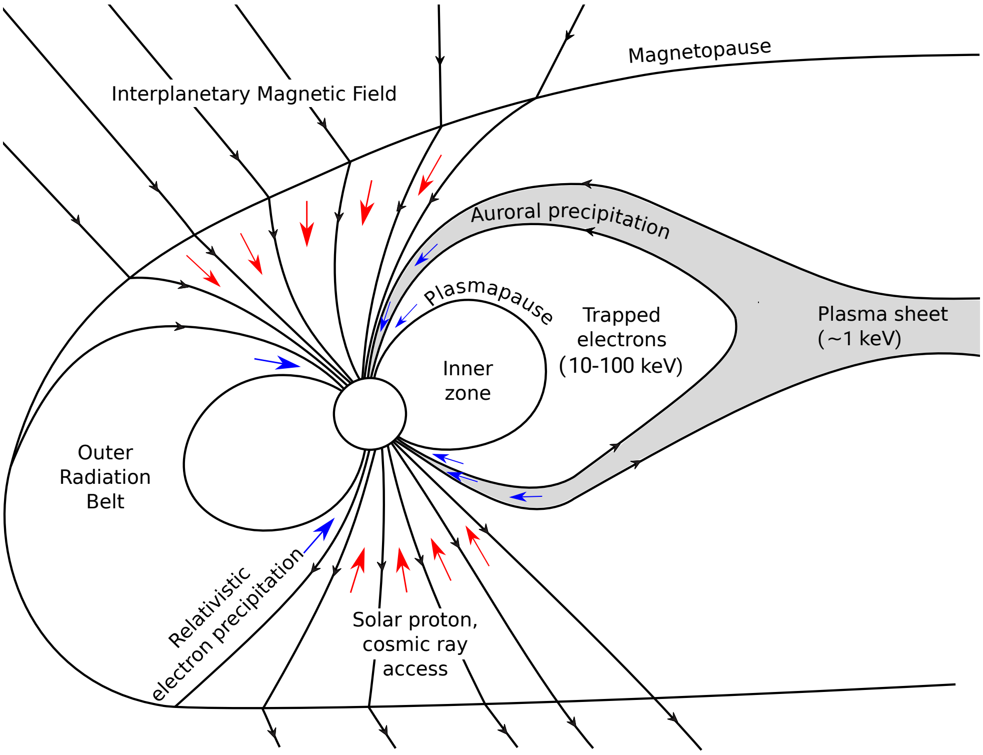

Interplanetary Magnetic Field
Magnetopause
Auroral precipitation
Plasmapause
Trapped electrons (10-100 keV)
Plasma sheet (~1 keV)
Inner zone
Outer Radiation Belt
Relativistic electron precipitation
Solar proton, cosmic ray access

Figure 2: Illustration of the magnetosphere with trapped particles [Ødegaard, 2016, figure 2.2].

1.2.5 Particle precipitation

Charged particles follow the field lines of the magnetosphere by spiralling around them. The angle of the velocity vector with respect to the geomagnetic field line is known as the pitch angle α, as illustrated in figure 3. The pitch angle of a particle is defined as the angle of the velocity vector with respect to the field line, which can be constructed from the two velocity components $v_{\parallel}$ and $v_{\perp}$, the components parallel and orthogonal to the field line respectively,

$$\alpha = \tan^{-1}\left(\frac{v_{\perp}}{v_{\parallel}}\right) \tag{1.1}$$

The pitch angle varies with the magnetic field, and since the magnetic field is weakest at the equatorial plane, the pitch angle of a particle is also at a minimum here [Ødegaard, 2016]. Because of the ever-changing nature of the pitch angle it is useful to characterize a particle by its pitch angle at this point, known as the equatorial pitch angle. As the particle moves along Earth's magnetic field, the pitch angle increases as it moves closer to the magnetic poles, and when the particle reaches a pitch angle of $\alpha = 90°$, it will turn around and subsequently bounce back and forth between the hemispheres. The point at which the particle turns around is known as a mirror point, and it causes the particle to be indefinitely suspended in the magnetic field. If the mirror point is below an altitude of 100 km, it is unlikely that the particle will be able to mirror without colliding in the denser layers of the atmosphere. Such a particle is said to be lost to the atmosphere, or to precipitate into the atmosphere.

In order to quantify the rate of precipitating particles, a loss cone is defined at the equatorial plane that divides precipitating particles from trapped particles. If a particle has an equatorial pitch angle α_{eq} within the loss cone, it has a mirroring point below the point at which it will be able to return before colliding with the gasses in the atmosphere, and thus it will be lost to the atmosphere. A typical pitch angle distribution has few particles within the loss cone, gradually decreasing with decreasing pitch angle. The loss cone pitch angle can be defined by the relationship,

$$\sin^2 \alpha_{LC} = \frac{B_{\text{eq}}}{B_0} \tag{1.2}$$

where B_{eq} is the magnetic field strength in the equatorial plane and B_0 is the magnetic field strength at some defined altitude, typically chosen to be 100 km. The geometry of the loss cone is shown in Figure 3.

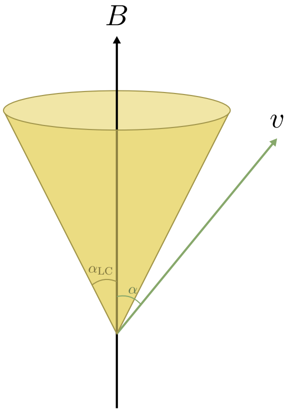


Figure 3: Loss cone of the equatorial pitch angle. In this case the equatorial pitch angle of the particle points out outside the loss cone and the particle is trapped. Adapted from [Rodger et al., 2013].

1.2.6 Effects

The most famous effect of particle precipitation is the appearance of aurora polaris, caused by the emission of photons at certain wavelengths as the precipitating particles excite the particles in the atmosphere. Different colours are associated with different atmospheric gasses, stemming from precipitation into different regions of the atmosphere.

There has been a lot of research into electron precipitation of energies less than 30 keV because of its ties to polar lights. Energetic particles with a kinetic energy greater than 30 keV are not typically associated with aurora, and their spectral energy flux is under-represented in atmospheric models [Nesse Tyssøy et al., 2016] [Nesse Tyssøy et al., 2019]. These particles are of interest mainly due to the generation of NO_x and HO_x gases in the upper atmosphere [Lam et al., 2010]. During polar winter, the NO_x gases descend to a lower altitude, and through a series of chemical reactions they can deplete the ozone in the upper stratosphere. This causes a disturbance in the thermodynamic equilibrium of Earth's atmosphere.

1.3 Detection

In order to accurately quantify the loss cone flux, not only must the particle energy be detected, but also its pitch angle. To measure the pitch angle distribution, multiple pinhole detectors are placed at different angles, measuring the fluxes of different pitch angles. The typical pitch angle distribution has a near constant level of fluxes at all angles that fall outside the loss cone, but the flux gradually decreases for smaller and smaller pitch angles. A better angle resolution is attained by placing a pixel detector behind the pinhole to determine the angle at which it entered the pinhole. Multiple detectors at different angles will however not be a part of the system launching on the sounding rocket, where the proof of concept relates to the energy distribution.

1.3.1 NOAA POES MEPED

The Medium Energy Proton and Electron Detector (MEPED) from National Oceanic and Atmospheric Administration Polar Orbiting Satellites (NOAA POES) is currently the best instrument available for measuring the flux of precipitating high-energy particles [Nesse Tyssøy et al., 2016]. This instrument is part of the Space Environment Monitor (SEM) package, which comes in two generations known as SEM-1 and SEM-2. This instrument package is included on 14 satellites in total; 12 from NOAA POES and an additional two from the European Organisation of Meteorological Satellites (EUMETSAT). In addition to the MEPED, the SEM package also contains a Total Energy Detector (TED) which measures the energy flux of electrons and protons in the range of 0.5 to 20 keV [Ødegaard, 2016].

The MEPED instrument is able to detect protons and electrons in the energy channels specified in Table 1 and 2.

Table 1: MEPED proton energy channels.

Energy channel	Proton energy range
P_1	30 keV to 80 keV
P_2	80 keV to 240 keV
P_3	240 keV to 800 keV
P_4	800 keV to 2500 keV
P_5	2500 keV to 6900 keV
P_6	6900 keV and greater

Table 2: MEPED electron energy channels.

Energy channel	Electron energy range
E_1	30 keV to 2500 keV
E_2	100 keV to 2500 keV
E_3	300 keV to 2500 keV

The instrument consists of two detectors, one pointed to 0° and another pointing to 90° of the local vertical. At high latitudes the 0° detector points nearly parallel to the magnetic field and measures particles that will be lost to the atmosphere. The 90° detector will detect precipitating or mirroring particles. In the common case of anisotropic pitch angle distribution, with fewer particles at the centre of the loss cone, the 0° will underestimate and the 90° detector will overestimate the precipitating fluxes. Several methods to mitigate this inaccuracy have been implemented such as applying the geometric mean, logarithmic mean, sinus fit and more advanced theoretical approaches [Nesse Tyssøy et al., 2016]. In addition to the insufficient pitch angle information, the MEPED instrument has a poor energy resolution, only providing three energy channels for electrons.

1.4 Project background

The DEEP instrument has been in development since the Birkeland Centre for Space Science (BCSS) first opened in 2013. Are Haslum started the work on the electronic design in 2017. He finished a prototype for the DEEP project that was onboard the ICI-5 sounding rocket launched from Ny-Ålesund on 26th November in 2019 [Lynnebakken, 2019] (rocket launch photograph in Figure 4). Contributions to the pixel design have also been made by Hogne Andersen in his 2018 Master's thesis.

Figure 4: ICI-5 launch from Ny-Ålesund 26th November 2019. Photo: Helge Markussen.

2 PRINCIPLES OF PARTICLE DETECTION

The theory in this section is retrieved from [Spieler, 2005], unless otherwise specified.

When a sufficiently energetic particle strikes a semiconductor, it will ionize the lattice atoms and generate electron-hole pairs along the particle track. Silicon is the predominant semiconductor used in detection of charged particles [Knoll, 2010, p. 354]. The energy required to ionize one atom is known as the ionization energy and is equal to roughly 3.6 eV for silicon, but this is subject to variation of the material. The ionization energy remains largely constant for incident energies, and the number of electron-hole pairs generated is thus approximately proportional to the kinetic energy of the particle.

The number of electron-hole pairs generated by incident radiation is equal to,

$$N = \frac{E}{E_i} \tag{2.1}$$

Where E is the energy imparted by the particle and E_i is the ionization energy. Each mobile carrier bears a charge equal to one elementary charge q. The total charge induced in a semiconductor by energetic radiation is thus,

$$Q = q\frac{E}{E_i} \tag{2.2}$$

The semiconductor is in itself not sufficient as a detector, because although the incoming particle generates mobile carriers, they quickly recombine. A p-n junction, known for its primary purpose as a diode, is better suited for the task. A p-n junction is formed by two neighboring regions in a silicon crystal: one negative (n-doped) and one positive (p-doped) region.

With no external bias, a small current can be sensed as the mobile carriers are propelled by the small built-in potential present in a p-n junction, which depends on the doping level in the semiconductor. However, because of spontaneous recombination and trapping, not all charge carriers make it to the electrodes, and it is considered a poor sensor. By applying a suitably large reverse bias to the junction, almost all the mobile carriers make it to the electrodes. The p-n diode with a reverse bias acts as a capacitor, where the ohmic contacts are the electrodes and the depletion region is the dielectric material. The concept is illustrated in Figure 5. An incident particle hits the diode and frees electrons and holes. These are swept up into the electrodes on their respective ends, inducing a current. In practice, a pixel detector does not look like this, but is implemented in a larger two-dimensional area in a heavily p-doped silicon substrate. The pixel contains a negatively doped area known as an n-well at the top that attracts electrons, and an ohmic contact at the p-substrate at the bottom. The reverse bias is then applied top-to-bottom, and hole current travels downwards while electron current travels upwards.

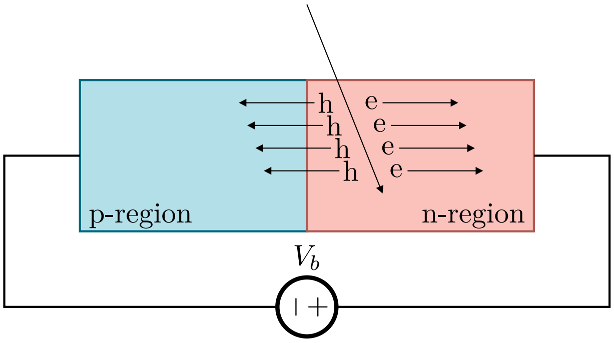


Figure 5: p-n junction acting as an ionization chamber, holes are denoted h and electrons denoted e. Adapted from [Spieler, 2005, p.9].

As both holes and electrons are majority carriers in their respective electrode, they both contribute to the total current and are added in accordance with the superposition principle. Since the propagation speed of majority carriers is not equal, the current generated has two components as shown in Figure 6 (top). Hole propagation happens at roughly one third of the rate of electron propagation. By equation (2.3), the charge is the integral of the current with respect to time, so the total induced charge will look like Figure 6 (bottom).

$$I = \frac{dQ}{dt} \tag{2.3}$$

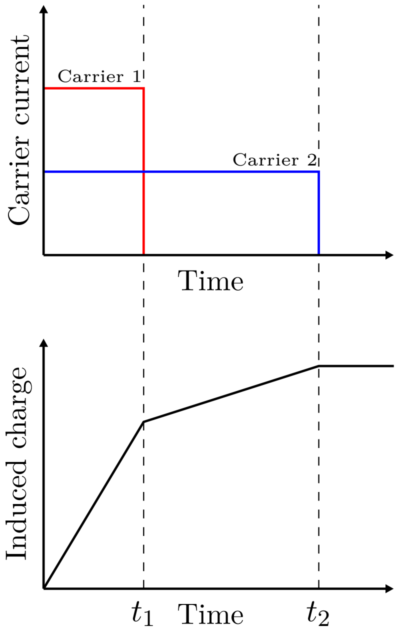


Figure 6: Current generated by the propagation of majority carriers (top) and the equivalent induced charge (bottom). Adapted from [Knoll, 2010, p. 367].

2.1 Amplification

The current that is generated by charge according to Figure 6 is amplified by a preamplifier that integrates the current to produce an output voltage. This is known as a Charge Sensitive Amplifier (CSA), and it outputs a voltage that is proportional to the total charge deposited in the sensor. The gain G of the CSA is commonly expressed in mV/fC.

2.2 Pulse shaping

The output voltage of the preamplifier can be measured directly by an Analog-to-Digital Converter (ADC), but this measurement is very susceptible to noise. To improve the Signal-to-Noise Ratio (SNR) of the detector system, a pulse shaper is included before the measurement. The pulse shaper consists of a low-pass and a high-pass filter that outputs a regular pulse shape with a known shaping time (or peaking time) τ_P. The signal power is distributed in the frequency space, and the shaper filter is designed to favor the signal and attenuate the noise. The ideal shaper output for maximized SNR is a pulse that peaks for an infinitesimally small amount of time. This is impractical, however, because it becomes impossible for the digitizer to measure the peak value of the pulse. A practical implementation of the pulse-shaper is one in which the peak value persists for a measureable amount of time.

By the convolution theorem, one can relate the Fourier transform in the frequency domain to the pulse shape in the time domain. This is defined as,

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau = F(f)G(f) \tag{2.4}$$

In practice, this relationship means that the shape of the signal in the time-domain is affected by the transfer function in the frequency domain.

The time response of a low-pass filter in response to a step input voltage V_i is,

$$f(t) = V_i\left[1 - \exp\left(\frac{-t}{\tau_i}\right)\right] \tag{2.5}$$

where τ_i is the time constant of the low-pass filter (also known as an integrator). Similarly, the transfer function of a high-pass filter, also known as a differentiator, is,

$$g(t) = V_i\left[\exp\left(\frac{-t}{\tau_d}\right)\right] \tag{2.6}$$

where τ_d is the time constant of the high-pass filter (differentiator).

Calculating the output response in the time domain is done by convolving the input with the transfer functions,

$$V_o(t) = V_i(t) * f(t) * g(t) \tag{2.7}$$

For a unit step this is,

$$V_o(t) = \frac{\tau_d}{\tau_d - \tau_i}\left[\exp\left(\frac{-t}{\tau_d}\right) - \exp\left(\frac{-t}{\tau_i}\right)\right] \tag{2.8}$$

And for $\tau = \tau_i = \tau_d$,

$$V_o(t) = \left(\frac{t}{\tau}\right) \exp\left(\frac{-t}{\tau}\right) \tag{2.9}$$

As the output shape is the same for any step input, the amplitude must be proportional to the step amplitude, which in turn is proportional to the induced current. It can be useful to normalize this function if the amplitude is known. Since the maximum of this function occurs at the peaking time $t = \tau$, the normalized function $\hat{V_o}(t)$ is attained by multiplying $V_o(t)$ with the inverse of the peaking value e^{-1},

$$\hat{V_o}(t) = \left(\frac{t}{\tau}\right) \exp\left(1 - \frac{t}{\tau}\right) \tag{2.10}$$

As the pulse is shaped, the energy that is proportional to the incident charge is now also proportional to the amplitude of the shaped pulse,

$$Q \propto E \propto A \tag{2.11}$$

A complete readout chain is presented in Figure 7.

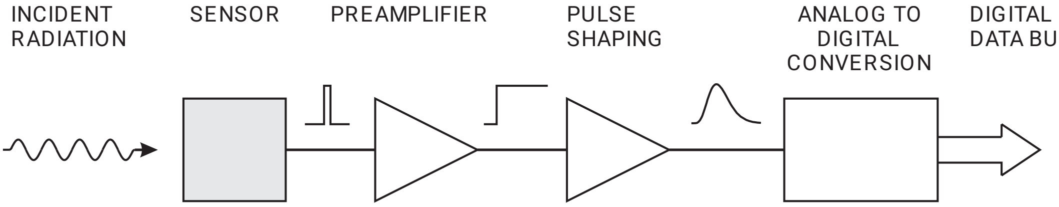


Figure 7: Particle detection signal chain [Spieler, 2005, p. 2].

2.3 Pile-up

After a particle has generated charge in the sensor, there is a certain amount of time until the output returns to its baseline value, decided by the shaping time of the pulse. In the event that two particles should strike the sensor in a time interval shorter than this, a condition known as pile-up occurs. It is useful to divide pile-ups into two kinds: tail pile-up and peak pile-up. In the first case, the pulses are sufficiently spaced so that they can be discerned, but the remnants of the tail of the first pulse, whether positive or negative (due to undershoot), is superimposed on the following pulse. Alternatively, in the case of peak pile-up, the particles strike the sensor with such a small time difference that the charge deposited appears as a single pulse. The superposition of the pulses causes their kinetic energy to be summed, and the fact that two particles struck the sensor is occluded. [Knoll, 2010]

Particles striking the sensor are randomly spaced in time and follow the Poisson distribution,

$$P(X = k) = \frac{\lambda^k \exp(-\lambda)}{k!} \tag{2.12}$$

where λ is the expectation value and $P(X = k)$ is the probability that k events happen within a time interval. Letting $\lambda = rt$, where r is the rate of events and t is time, the probability of another particle striking the sensor within the time period τ and causing a pile-up condition can be calculated. Let t_0 be the time that a particle strikes the sensor and t_1 be the time when the sensor circuitry is ready to detect another particle, and then define $\tau = t_1 - t_0$. The probability can then be calculated with (2.12) by finding the probability that no particle strikes within τ and subtracting this from the total probability 1. Since this means that $k = 0$, the equation reduces to the cumulative exponential distribution and the pile-up probability P is given as,

$$P = 1 - \exp(-r\tau) \tag{2.13}$$

Figure 8 shows the pile-up probability by varying the time window τ from 0 to 1000 ns and the event rate r from 10^4/s to 10^7/s. A low pile-up probability is desired, so ideally the shaping time needs to be low enough for a given flux so that the probability remains in the blue-shaded area on the graph. For low fluxes a long shaping time is fine, but the shaping time requirements become more stringent as the particle flux increases. A reduced shaping time can increase the complexity of the readout circuit if the pulses are so short that very high speed and high resolution methods need to be employed.

The expected count rate is on the order of $r = 10^5$ events/second [Nesse Tyssøy et al., 2017].

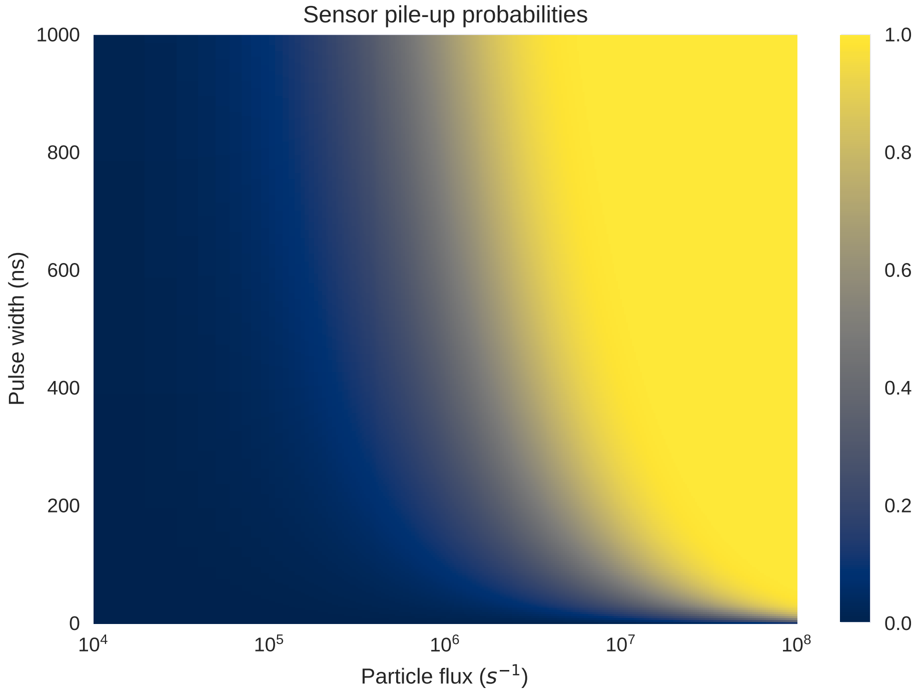


Figure 8: Color map of pile-up probabilities for varying particle fluxes and shaping times.

2.4 Baseline shift

Since there is a series capacitor in the pulse shaper, any DC component of the input is blocked. A sequence of pulses will have a DC component that depends on the event rate. The DC baseline shifts make the total transmitted charge equal to zero. Since the event rate can change over time or fluctuate randomly, the baseline will shift over time, and this contributes to the noise of the system. In both the cases of low and constant event rates the effect of baseline shift will be minimal, but in high event rate systems it may affect the resolution of the system significantly. A baseline restorer can be included in the circuit that temporarily grounds the signal line in the absence of any signal, thereby removing unwanted charge.

2.5 Data acquisition

The amplitude of the pulse generated by the CSA is proportional to the charge deposited in the pixel detector. Measuring the amplitude requires some form of analogue to digital conversion, and there are two primary ways to achieve this. The first method is a direct measurement of the pulse amplitude using an ADC. The other method exploits the fact that the shape of the pulse is the same for all incoming pulses, and the time a pulse spends over a given threshold is a function of the amplitude. A Time-to-Digital Converter (TDC) can be used to determine how long the pulse is above some threshold. This can be mapped to a corresponding amplitude of the signal pulse. Since the amplitude is proportional to the incident energy and deposited charge, these are readily determined also. The threshold must be selected such that it is below the lowest signal amplitude, but sufficiently greater than the noise floor.

2.5.1 Analogue-to-Digital Conversion

The signal pulse can continually be sampled by an ADC, and the current pulse value compared to some locally stored maximum. If the pulse value is above the local maximum, the maximum is updated with the new value. After a pulse has passed, the maximum is recorded as the amplitude of the signal, and the energy is determined. This results in a linear relationship between measured pulse heights and incident particle energies. Error sources in an ADC-based data acquisition circuit include circuit noise and quantization error, but the main driver of inaccurate conversion is insufficient sampling rate. The shaping time must be minimal to avoid pile-up, and the amount of samples per pulse should be sufficient to accurately determine the peak; low rates can lead to sampling on both ends of a peak, missing the true peak of the pulse. Another challenge with implementing an ADC is that since particles can scatter through several sensors, multiplexing cannot be used, and a separate channel is required for each pixel. A high number of channels and a high sampling rate causes high power dissipation to be a limiting factor.

2.5.2 Time-over-Threshold

An alternative approach to converting the pulse to a digital value is by the Time-over-Threshold (ToT) method. It has been applied successfully in detector readout systems such as the BaBar Silicon Vertex Tracker at the Stanford Linear Accelerator Center [Kipnis et al., 1997]. By letting the signal be the input to a comparator with an appropriately chosen threshold voltage, the signal charge deposited into the sensor may accurately be quantified by measuring the pulse width of the comparator output, illustrated in Figure 9 and Figure 10. The conversion is done by a TDC that converts time intervals into quantized digital representations. The energy resolution is constrained by the resolution attainable in the TDC. The ToT method is very sensitive to noise on the input of the comparators, so circuit noise must be minimized to discern the signal from the noise in the lower range of energies.

The benefit of the ToT method is simplicity, resulting in a low component count and low power dissipation.

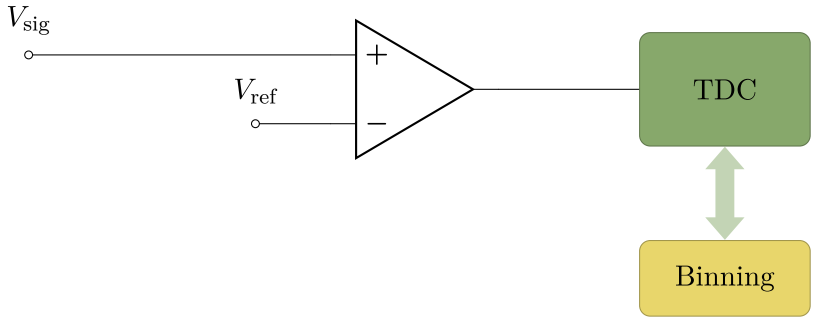


Figure 9: Implementation of Time-over-Threshold.

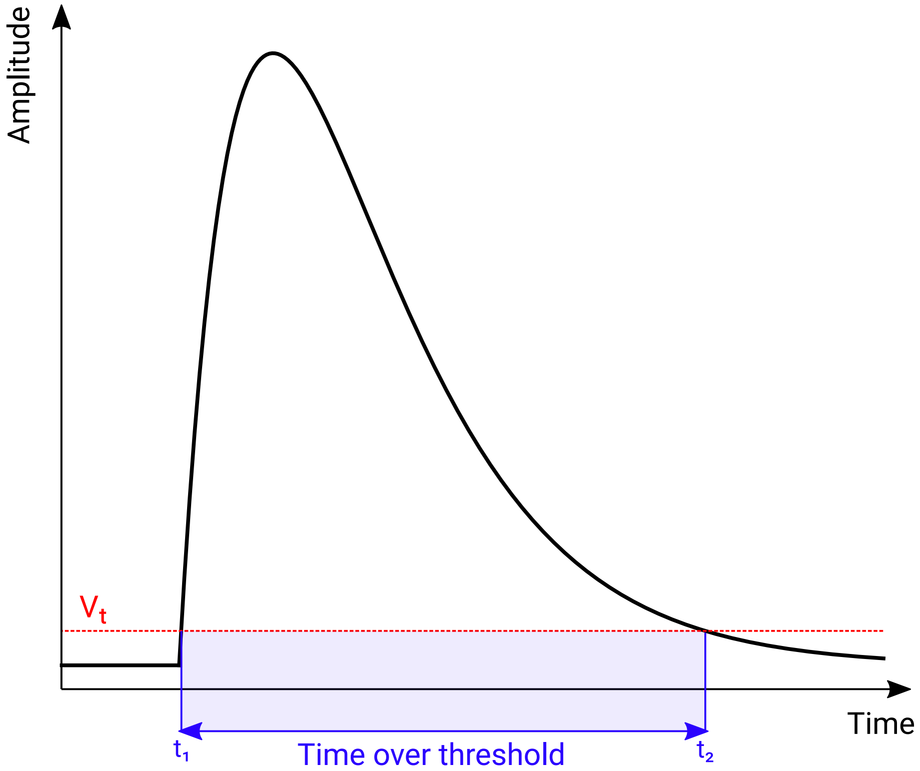


Figure 10: Time-over-Threshold demonstrated on a shaped pulse. The time interval shown in blue is digitized, and the pulse peak information is withdrawn from the pulse width.

One crux of the ToT method is that the width of the pulse generated by the comparator is highly nonlinear, and consequently the difference in width between short pulses is greater than the difference between two large pulses. This necessitates mapping the pulse width to energy nonlinearly with a conversion table. Another consequence of the nonlinear transfer function is that as the energy increases, the resolution decreases, because each step change in energy corresponds to a smaller and smaller difference in the time over threshold. A result of this is that the dynamic range suffers, as it becomes increasingly hard to detect differences in higher energies.

The ToT interval is dependent on the threshold voltage, the shaping time of the pulse shaper, as well as the incident energy.

The pulse peak value is known for a given incident energy and is the product of the deposited charge Q in coloumbs (given by equation (2.2)) and the gain G of the CSA in volts per coloumb,

$$V_{pk} = GQ = Gq\frac{E}{E_i} \tag{2.14}$$

Multiplying the peak with the function for the normalized shaped pulse in (2.10) gives the function for the pulse dependent on incident energy of the particle and gain of the CSA,

$$V_o(t) = Gq\frac{E}{E_i}\left(\frac{t}{\tau}\right)\exp\left(1-\frac{t}{\tau}\right) \tag{2.15}$$

Of main interest is developing a function $T(E)$ that maps from incident energies to time intervals. Note that the CSA gain G, the ionization energy of the semiconductor E_i, and the shaping time τ are all known design variables, and can be considered constants for the remaining analysis.

Let the threshold voltage be a constant V_t. $V_o(t)$ is zero both for $t = 0$ and $t = \infty$, and crosses the threshold exactly twice. The crossover points are the roots t_1, t_2,

$$Gq\frac{E}{E_i}\left(\frac{t}{\tau}\right)\exp\left(1-\frac{t}{\tau}\right) - V_t = 0 \tag{2.16}$$

Because of the exponential function, the roots of (2.16) are attained using the multivalued Lambert function $W(E)$. Each root is given by the two branches of the Lambert function W_0 and W_{-1},

$$t_1 = -\tau W_0\left(\frac{-e^{-1}V_t}{G\frac{qE}{E_i}}\right) \tag{2.17}$$

$$t_2 = -\tau W_{-1}\left(\frac{-e^{-1}V_t}{G\frac{qE}{E_i}}\right) \tag{2.18}$$

The time interval above a given threshold is thus decided by the difference in the roots,

$$T(E) = t_2 - t_1 \tag{2.19}$$

This is readily calculated by applying the Python package Scipy to calculate the Lambert function, and the function T(E) is solved for all energies in the interval up to 2 MeV. A plot of the resulting transfer function for ToT is given in Figure 11.

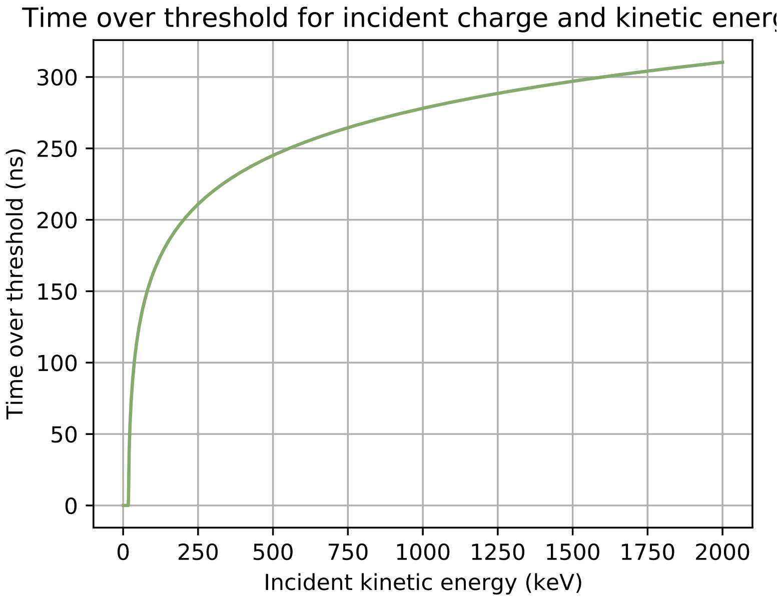


Figure 11: Time over threshold as a function of incident kinetic energy.

The resulting relationship between incident energy and time over threshold of the shaped pulse is highly nonlinear. For small energies, a small change in input energy results in large differences in the time interval produced by the comparators. This means that as the kinetic energy increases, the resolution of the readout electronics decreases in turn.

2.5.3 Dynamic Time-over-Threshold

In an attempt to rectify the non-linear relationship described above, [Orita et al., 2015] presents a method of linearising the transfer function. Instead of having a static threshold value on the comparators, an alternative is a threshold value that dynamically changes with the same time constant as the shaping time of the pulse. This method is known as Dynamic Time-over-Threshold (dToT). An implementation is illustrated in Figure 12.

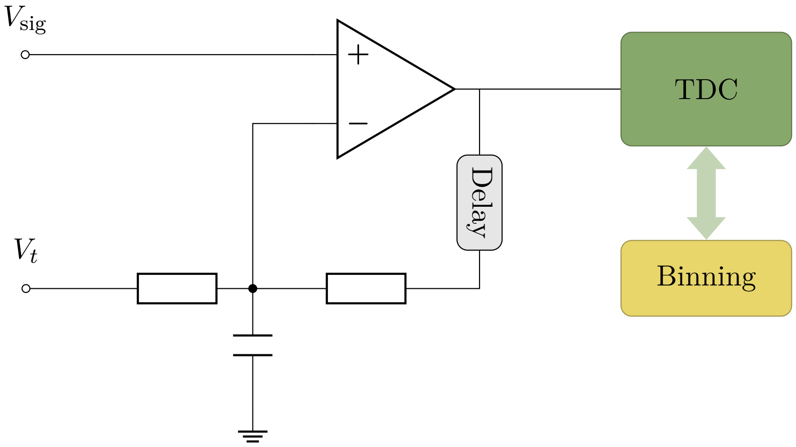


Figure 12: Implementation of Dynamic Time-over-Threshold. Adapted from [Orita et al., 2015].

Orita et al. used a monostable vibrator to create the delay, but it can be generated in many ways. The important thing is to closely match the delay to the peaking time of the pulse. If there is random variation in the delay, this will manifest itself as noise on the time interval. The working principle is shown in Figure 13. The red dashed line is the dynamic threshold, and the blue line is the conventional static threshold. Shown here are three pulses of amplitude 0.25, 0.5 and 1. The non-linearity can be shown with the blue arrows. In a linear system it is expected that each doubling of the amplitude corresponds to a doubling of the time interval. However, in the static case, the difference between 0.5 and 1 is as large as the difference between 0.25 and 0.5. This is rectified by the dynamic threshold, where it can be seen that the distance in time between 0.5 and 1 is twice as large as the difference between 0.25 and 0.5.

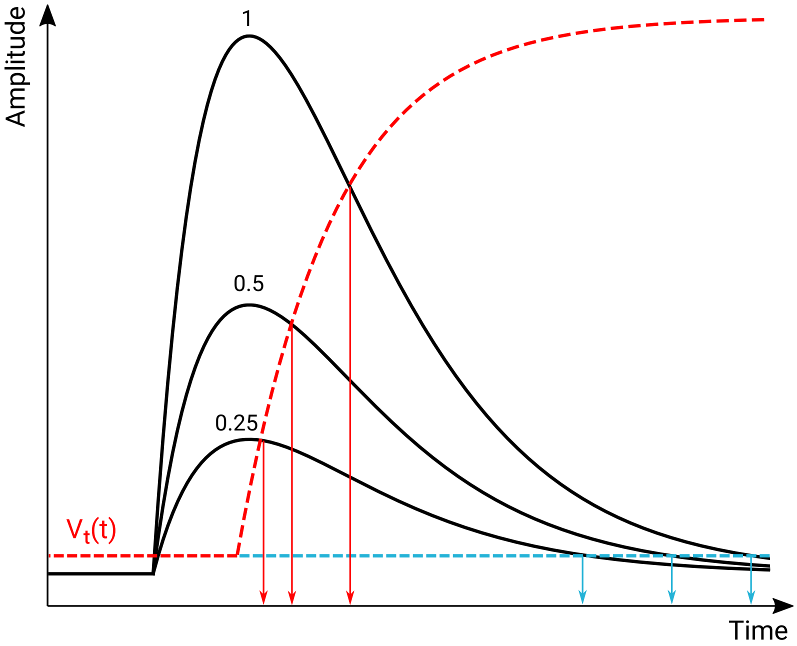


Figure 13: Principle of Dynamic Time-over-Threshold. The threshold is triggered to rise exactly at the peaking time of the pulse. The red arrows signify at which point in time a pulse goes below the dynamic threshold, while the blue arrows show the times it transitions below the static threshold.

The transfer function between kinetic energy and time over threshold is shown in Figure 14.

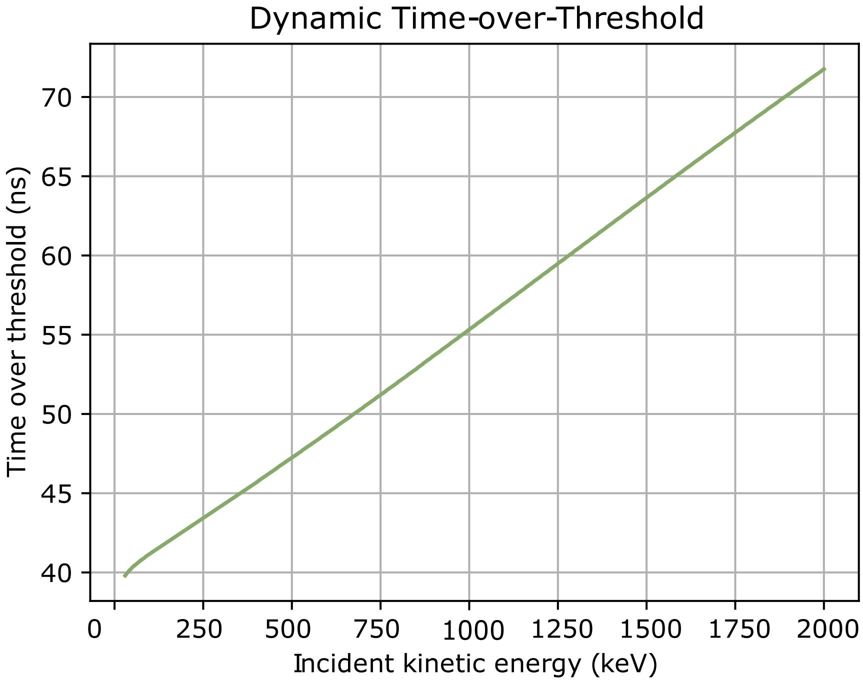


Figure 14: The linear transfer function of a dToT implementation.

3 SYSTEM OVERVIEW AND SPECIFICATIONS

The Investigation of Cusp Irregularities (ICI) rocket campaign is a scientific campaign that aims to investigate disturbances in the ionosphere through the use of sounding rockets. The last prototype of DEEP flew as part of the instrument package on the fifth ICI rocket known as ICI-5. This sounding rocket flew to an altidue of 253 km [lau, 2020]. The next iteration of the instrument is planned to be included in the payload the next ICI rocket in the campaign. Although this rocket is currently under development, preliminary specifications have been received, and it is assumed that many aspects of the integration will be comparable. The same detector house will be used, and the electrical interface between the instruments and rocket is confirmed to be backwards compatible with ICI-5 (Email correspondence, Geir Lindahl ASC via Ketil Røed (UiO) 30-04-2021). This section gives an overview of the system specifications, the sounding rocket integration, and work that has been done on the project prior to this thesis.

3.1 Detector house

The instrument house consists of both the detector house and the readout electronics house. The detector house for the detector itself is made out of 2.5 mm thick aluminium sides, further shielded by 2 mm tungsten on the inside to protect from high-energy particles that may penetrate the sensor housing and deposit energy into the pixel sensors. The particles enter the detector house through a conical hole at the top known as a pinhole or aperture. A conceptual drawing is illustrated in Figure 15.

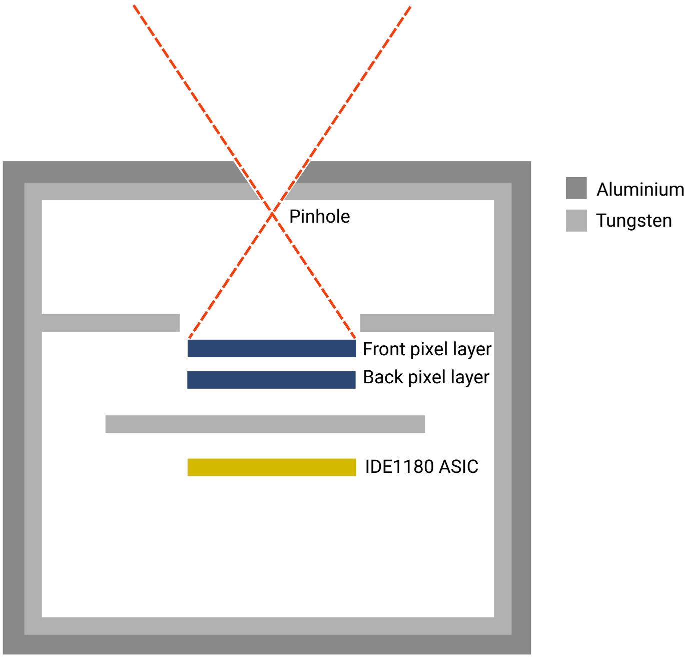


Figure 15: Conceptual detector house drawing. The outer walls are aluminium and the inner walls consist of Tungsten.

A CAD drawing is shown in Figure 16. The pinhole where electrons enter is shown at the top of the house. Inside, the pixel detectors are masked by a sensor mask made out of

Tungsten. It is thicker around the edge, and also provides a physical barrier between the pixel detectors. The purpose of the thick layer around the edge is to stop particles from scattering past the pixels. The thin dividing layer between the pixels is to stop electrons from scattering horizontally between pixels. Below the first pixel layer is the second pixel layer. This is another identical layer of pixels on the flip-side of the Printed Circuit Board (PCB), i.e. directly below the front layer. Below the pixel detectors is another layer of Tungsten shielding to protect the Application Specific Integrated Circuit (ASIC) from the radiation. Directly below the Tungsten shielding is the ASIC PCB. The output pulses from the ASIC travels on traces in the flex ribbon cable, leaving the detector house and entering the electronics house where it is bonded to the DEEP readout board.

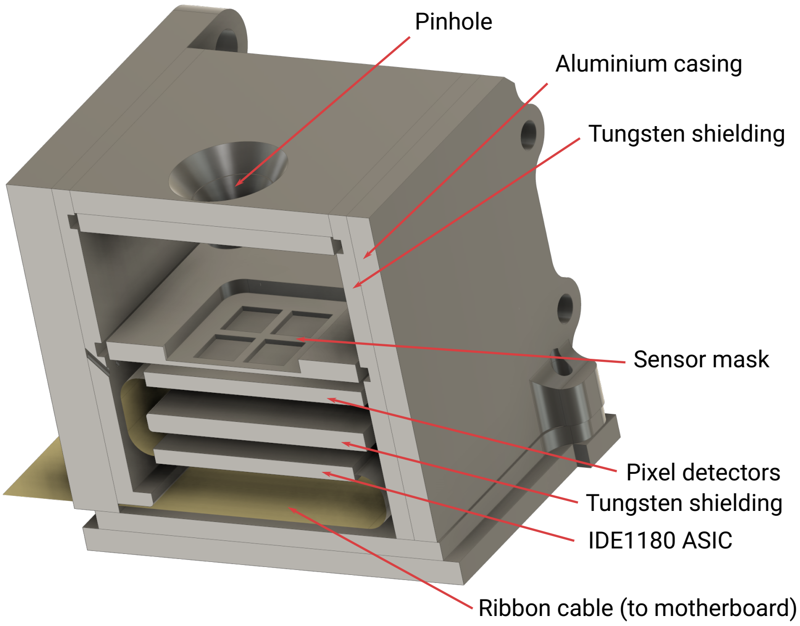


Figure 16: Cross-section of detector house (CAD drawing by Torstein Frantzen).

Inside the detector house are two silicon diode layers, each consisting of four pixel detectors for high angular resolution. Each semiconductor layer is 1 mm thick. The proposed pixel arrangement for the complete system is two layers of 2x4 sensor pixels for the electron detector, and two layers of 2x2 sensor pixels for the proton detector. The scaled down test detector considered in this thesis makes use of a 2x2 array of electron detectors.

The combination of the electronics house, detector house and rocket interface constitutes the DEEP instrument. It is shown integrated with the ICI-5 rocket in Figure 17.

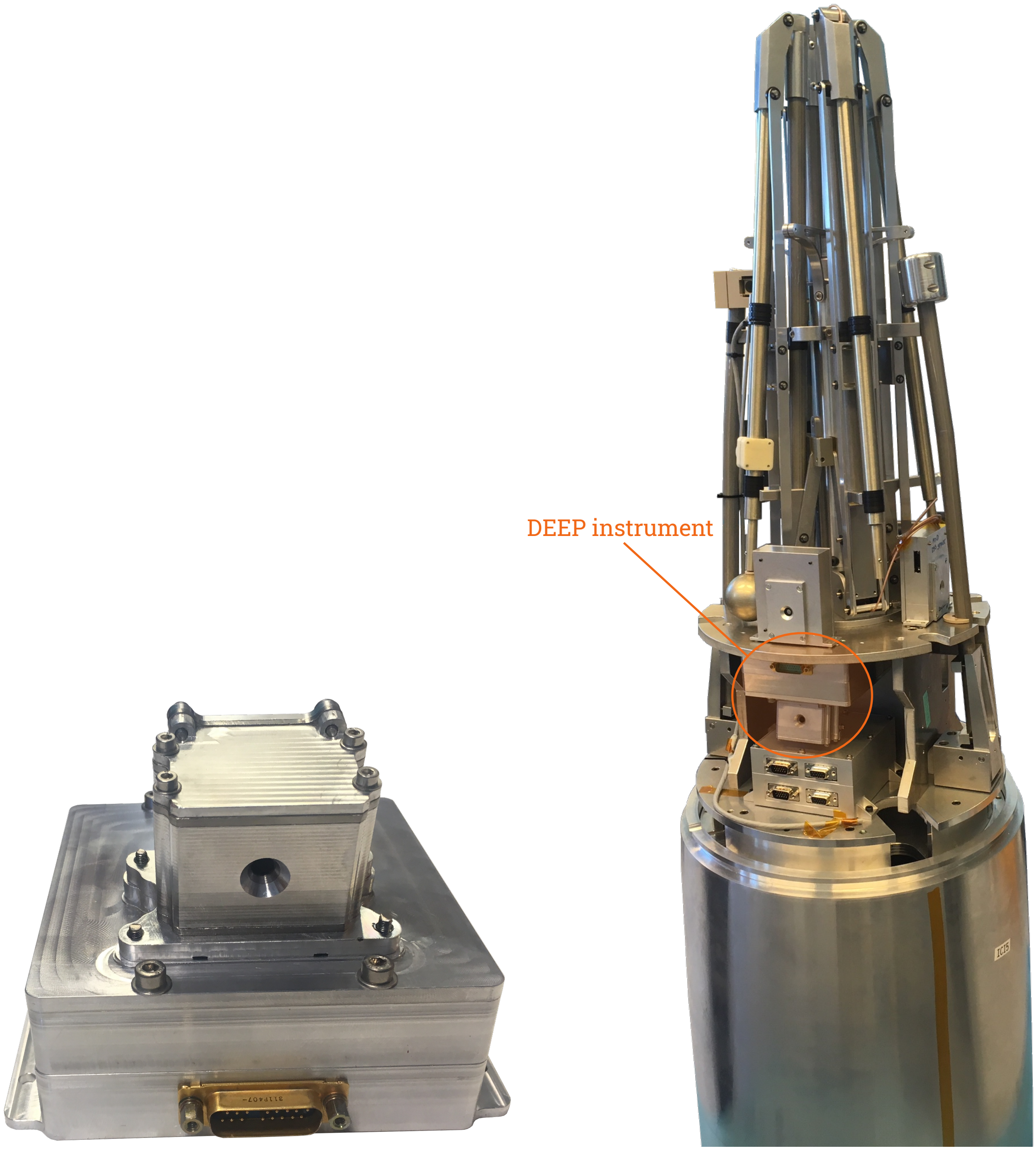


Figure 17: Left: Detector house consisting of detector house (top) and electronics house (bottom). Right: Detector house mounted on the ICI-5 rocket. Photo: Are Haslum.

3.2 Binning

The particle flux energies are expected to be exponential, and the incident radiation will be dominated by the lower range of energies. The number of bins must also be limited to not occupy a large amount of memory and telemetry bandwidth. The bins outlined in Table 3 are proposed for the histogram. Each bin width is doubled to account for the lower number of particles expected as energy levels increase.

Table 3: Desired bin widths in the DEEP histogram.

Bin	Energy range
1	>30 keV
2	>60 keV
3	>120 keV
4	>240 keV
5	>480 keV
6	>960 keV
7	>1500 keV

3.3 Sensor PCB

The sensor module is a separate PCB that consists of eight pixel detectors arranged in a 2x2 pixel array, shown in Figure 19. The pixel detectors were produced by SINTEF. It features a flex cable that is to be connected to the DEEP readout board. When an energetic particle hits the detector, charge is deposited and this generates an electrical signal that can be sensed and amplified. The IDE1180 AMADEUS ASIC from IDEAS is used as a combined charge sensitive preamplifier and pulse shaper. It has 16 input channels, adjustable gain (mV/fC) as well as an adjustable shaping time ranging from 20 ns to 40 ns. The shaping time is the time interval between an input charge is collected to the output pulse peaks.

The theory discussed in section 2 is not practically applicable to the actual sensor PCB system. While the shaping time of the CSA is known, the exact pulse shape is not given in the accompanying datasheet. A measurement of the CSA pulse in response to a step input is shown superimposed on the pulse generated by (2.15) in Figure 18.

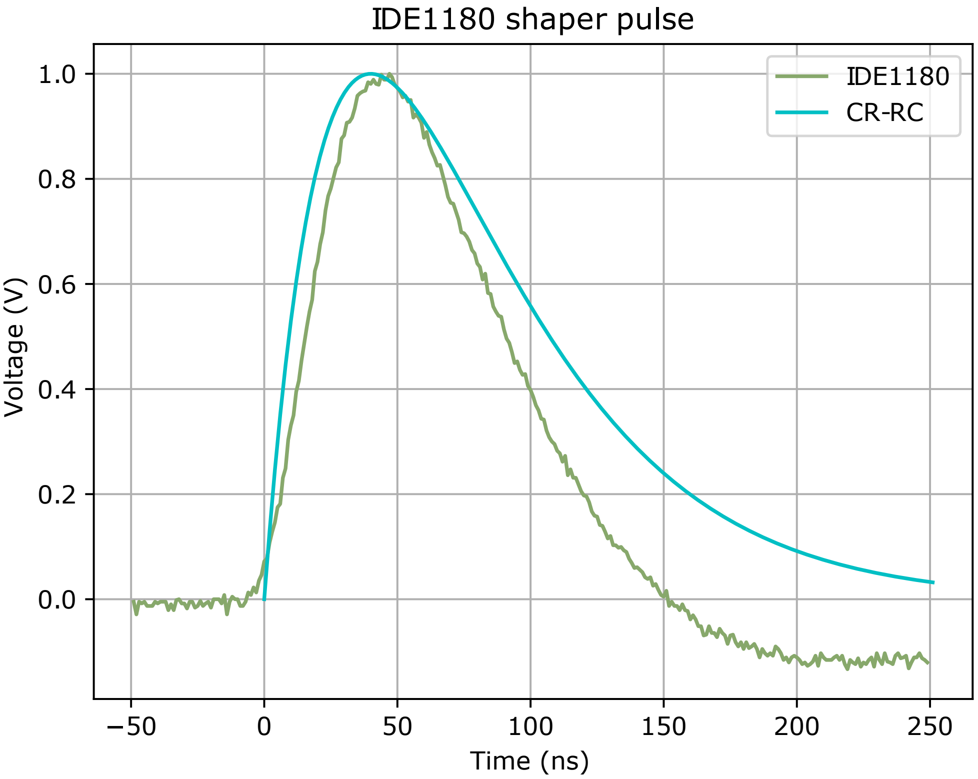


Figure 18: Idealized pulse versus measured IDE1180 pulse (normalized).

It can be seen that although the peaking time is the same, the IDE1180 pulse falls of at a steeper rate than the theoretical pulse. This is caused by an undershoot in the filter of the IDE1180 ASIC, and results in the ToT interval to be shorter than expected. The undershoot can also affect two pulses that are close in time, causing the second pulse to be smaller than expected.

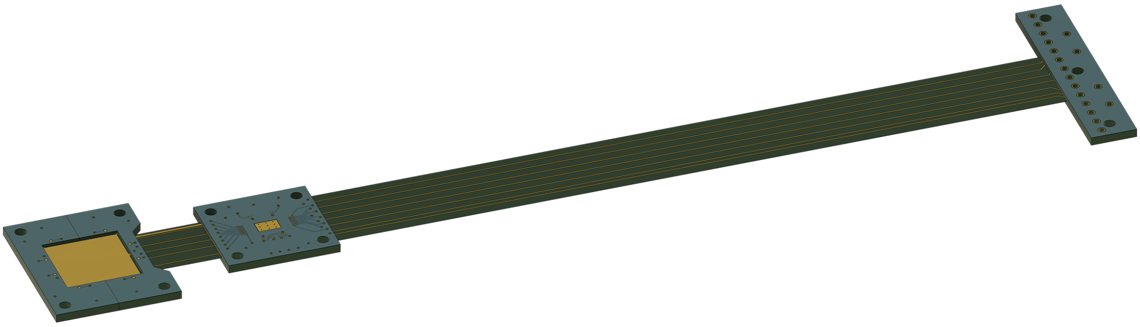


Figure 19: Flex PCB with sensor (left), IDE1180 ASIC (middle) and bonding connection (right).

3.4 Rocket interface

The rocket interface consists of a power connection to the battery, as well as signals to and from the rocket. It is an ASC requirement that these are galvanically isolated to eliminate ground loop issues. These signals are the same as requested for the ICI-5 launch and is listed in Table 4.

Table 4: Rocket interface signals.

Signal	Function	Direction
DATA	Data to PCM encoder	Out
GATE	Enable signal for data channel	In
SCLK	Synchronizing clock	In
MIN_FRAME	Minority frame counter	In
MAJ_FRAME	Majority frame counter	In
TIMER	Timer signal that goes high when rocket reaches a certain altitude, used for turning on 150V source	In
GSE_CONTROL	Ground Support Equipment (GSE) signal that can turn on and off 150V source when the rocket is on the ground	In

The telemetry system on the sounding rocket from ASC is a type of Pulse Code Modulation (PCM) system using Time-Domain Multiplexing, and it offers digital channels as well as analogue channels. For this project histogram of counts for each energy bin is the main information that will be transmitted. This is readily represented as a digital value, and thus the digital channels will be used.

Data in a PCM system is structured into major and minor frames. To know when a frame starts and stops, synchronization words are included in the data stream. This process is handled by the PUSEK PCM encoder onboard the rocket, and signals are supplied to the DEEP instrument to identify the start of minor and major frames. A major frame consists of minor frames, which are small periods of time in which the output spectrum of a given sensor is transmitted.

3.5 Specifications

Since multiple parties have requirements that must be met in the development of the detector, it is useful to collect these in a specifications list. These requirements are based on information about and previous integration with the ICI-5 sounding rocket, desired integration with the completed mechanical house and sensor PCB, as well as a conceptual design report of the detector. The following system specifications are compiled for the design of this detector:

- The previous iteration of the circuit board was approved for flight by ASC. The new board should have the same physical dimensions and similar weight to avoid necessitating a new approval.
- The total power consumption of the system should be less than 10 watts.
- The DEEP PCB is to be bonded to the sensor PCB and must supply the following to the IDE1180 ASIC: A bias current sink of 100 mA, a 150 V detector reverse bias, and a 0.5 V offset voltage. The 150 V bias supply needs to be disableable.
- Output pulses from the IDE1180 ASIC should be digitized and counts stored in a

histogram consisting of seven energy channels: 30 to 60 keV, 60 to 120 keV, 120 to 240 keV, 240 to 480 keV, 480 to 960 keV, 960 to 1500 keV as well as 1500 keV and greater.

- There is a total of eight pixel detectors, and each one should have its own dedicated digitizer to facilitate higher count rates.
- The pixel detectors must be shielded from high-energy particles not originating from the pinhole disturbing the pitch angle distribution measurement. Other methods of filtering these particles may also be employed.
- Count rates of up to 10^6 particles per second should be supported on each detector channel.
- Circuit components should be industry grade: automotive or military preferred.
- Preferrably, a space grade version of each component should be available even if it is not used on the proof-of-concept circuit.
- 28 V power will be supplied by the sounding rocket, but needs to be galvanically isolated to reduce ground loop issues. All locally required power lines must be generated on-board.
- The PCB needs to interface with the PCM encoder on the sounding rocket to transmit data through telemetry. Like the power line, these signals should be galvanically isolated and differential.
- The signal and power connections are interfaced to the rocket through a DSUB connector. The connector mounted to the detector house and is bonded to the PCB with flying leads.

4 MONTE CARLO SIMULATION OF ENERGETIC PARTICLE PRECIPITATION

In order to define an accurate algorithm for the processing of pixel data, it is useful to simulate the behaviour and movement of energetic particles once inside the detector house. Particles at relativistic velocities do not necessarily move in straight lines, and it is not guaranteed that they deposit charge in only one pixel. When a particle deposits energy in a medium, it is through a collision with atoms in the structure. When this happens, the path of the particle is altered. This process is known as scattering, and it should be accounted for when determining any algorithm involving the trajectory of particles. There are two key questions that should be answered before the data acquisition algorithm is determined:

1. At what rate do particles scatter from one pixel to another, and can it be determined which pixel the particle struck first?
2. What is the energy necessary to penetrate the Tungsten shielding and deposit energy in the back of the sensor, and can these particles be filtered to prevent them contaminating the pitch angle measurements?

This section extends upon the work carried out by Hogne Andersen in his 2018 Master's Thesis [Andersen, 2018]. In his thesis, he developed simulation models for the particle detector using GEANT4 Application for Emission Tomography (GATE), an open-source Monte Carlo simulation tool based on the Geant4 toolkit developed by European Organization for Nuclear Research (CERN). In particular, his simulation models of the pixel detectors has been reused, and his proposition for vertical coincidence checking is further examined. The simulation model is a detector house as described in 3.1, where the walls are shielded by aluminium and tungsten, and inside there are two Silicon plates that act as the sensors. In the simulation, these front and back sensor plates are singular units, and definitions of pixels are added during the processing stage based on the coordinates of the deposited energy.

The simulation environment used is shown in Figure 20.

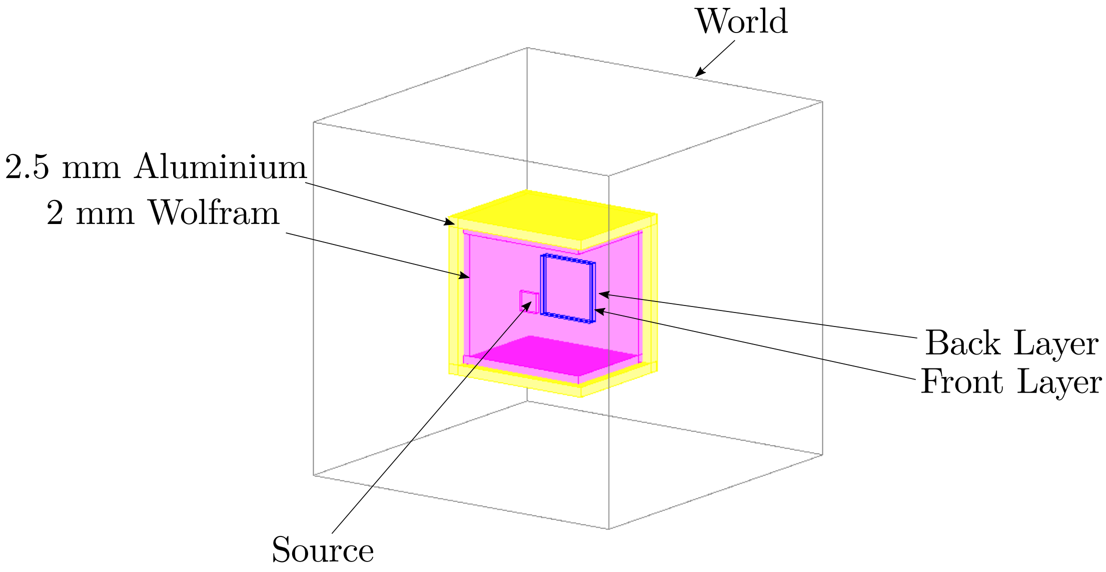


Figure 20: GATE simulation environment [Andersen, 2018].

The simulation input is specified in a MAC file, wherein the simulation environment, materials and geometry and particle radiation sources are defined. In the simulation, particles are emitted from the source towards the Silicon detector layers. The angle of the radiation source is specified by two angles, θ and ϕ.

The output of the simulation is a ROOT file, which is structured as a list of events. The file is read by a Python wrapper for ROOT known as PyROOT. Each event is a collision of the particle with a medium, and contains the particle ID, deposited energy, the spatial coordinates and a timestamp. The information is used to structure the hits into energy deposited in each pixel for each particle. The data can also be used to track the trajectory of the particles. Figure 21 shows the track of a particle that travels through the first sensor layer, hitting the second layer and bouncing back into the first layer again. It can be hard to see such patterns in a 2D plot, but since the particle data contains three-dimensional coordinates it can also be plotted in 3D as illustrated in figure 22.

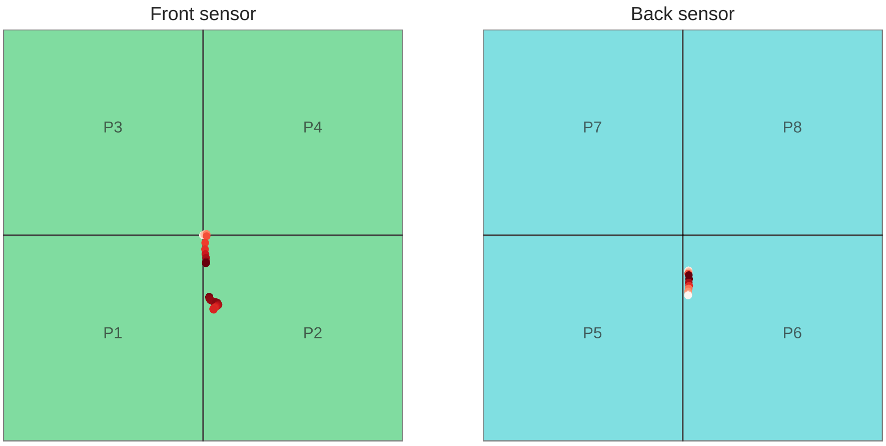


Figure 21: Electron hitting front and back sensors, simulated using GATE.

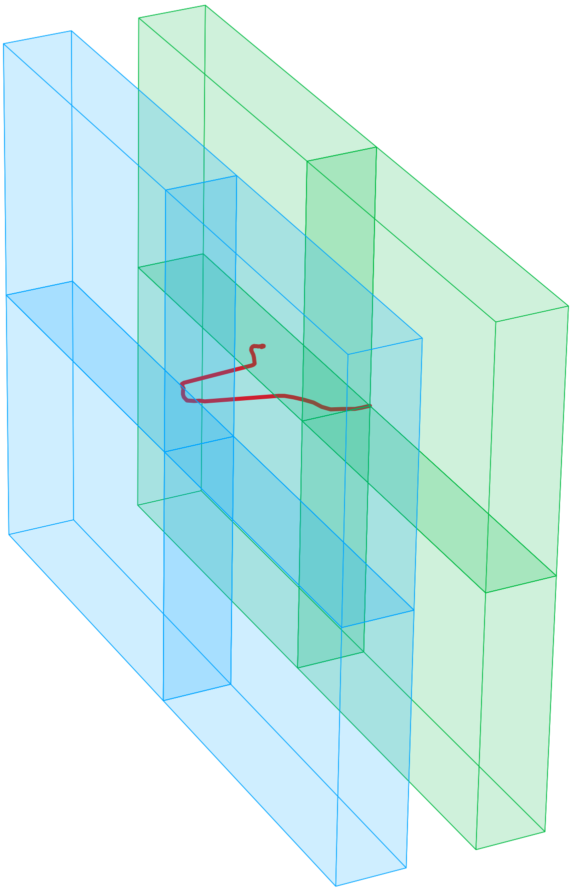


Figure 22: 3D visualization of the same electron scattering through sensor.

4.1 Horizontal coincidence

To be able to determine the pitch angle distribution, it is necessary to be able to determine from which direction a particle came from. This is a problem when detecting electrons because of scattering. An electron may scatter throughout many of the pixel detectors and deposit energy, known as a horizontal coincidence event. Determining if a particle struck several sensors can be done in the circuit by noting if the detectors both detect a hit in the same clock period, they belong to the same particle, and the energy must be summed. When a coincidence event has been established, the question arises of which histogram the particle belongs to. To uncover if the energy distribution can be used to determine which of the pixel was hit first, a simulation is needed.

The idea is that an electron is likely to deposit more energy in the pixel it first struck than in any of the other it hit. Therefore, the initial pixel and thus angular information about the particle can be retrieved by finding the pixel in the front layer that has the maximum energy in a hit. 100 000 electrons of energy 960 keV were fired towards the pixel array in GATE, and the coordinates of the first deposited energy event in the simulation were saved for each particle. The coordinates were used to determine which pixel a particle struck first and assign it to the particle. Next, the total deposited energy was sorted into a sum for each pixel. Since this only concerns the particles that hit multiple pixels, the data was filtered to only include coincidence events in the front layer.

The pixel with the most deposited energy for each particle was compared to the pixel each particle struck first. If they match, it is counted as a success. Of the 100,000 electrons, 20,713 of them deposited energy in multiple pixels. Of these, the initially struck pixel detector was the one with the most deposited energy in 10,319 cases, yielding a success rate of 49.8%. Since there are four pixels, it could be concluded that this is at least better than chance. In the vast majority of cases, however, the particle deposited energy in exactly two pixels. Only 1,214 cases had hits in more than two pixels. It is not really feasible to use this to determine the pitch angle of an electron based on this method. A proposed solution that retains information about both pitch angle distribution and flux energies is the inclusion of a ninth histogram. This histogram is used to collect all the particle hits for which the pitch angle is ambiguous.

4.2 Vertical coincidence

Andersen suggests in his thesis that as a particle scatters through both of the pixel layers, the energy deposited in the first layer relative to the second layer (so-called front/back ratio) is greater than some ratio R_c for lower energies. After a certain energy threshold E_c, the relationship flips and most of the energy deposited is in the back layer. [Andersen, 2018, p. 62]. In principle this ratio can be arbitrarily selected because each ratio has its own respective critical energy that can be determined by simulation. Since we want to say something about how the energy is distributed between the two pixel layers depending on the direction of the particle, the ratio is set to 1. It is important to recognize the fallacy of increasing detector efficiency by altering the ratio. By shifting the ratio, more hits that are known to originate from the pinhole (front pixel layer first) are accepted, but proportionally fewer of the hits from the back are rejected.

This is easily demonstrated by the inequality of accepted hits,

$$\frac{F}{B} > R_c \quad \Leftrightarrow \quad \frac{B}{F} < 1/R_c \tag{4.1}$$

where F is the energy deposited in the front layer and B is the energy deposited in the back layer. Essentially, this means that if ratios above R_c are accepted, only ratios above $1/R_c$ are rejected. Taken to an extreme, all hits can be accepted by setting the ratio to an infinitesimal ϵ, but this leads to rejecting all hits that are above $1/\epsilon$, which is such a large number that in reality it does not reject any hit that penetrated the shielding and hit the sensor from the back.

To investigate this proposed method of determining the direction of a hit, a simulation was defined that radiates electrons from a source towards the pixel array with energies ranging from 0 to 2 MeV. In order to sort them by energy, the electrons have been defined by a static energy and increased in steps of 10 keV with repeating GATE runs. The scatter plot generated from this simulation is shown in Figure 23. Along the x-axis of this scatter plot are the true energies of the electrons, that is the actual kinetic energy of the particle being spawned by the radiation source.

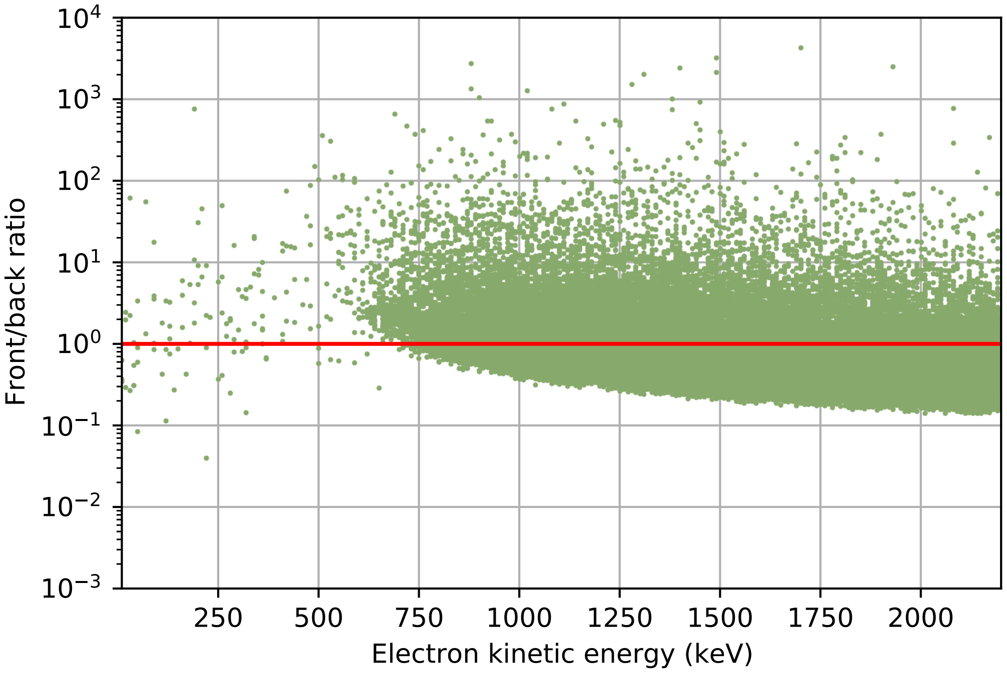


Figure 23: Simulation scatter plot of front/back ratios of the energy deposited by precipitating electrons across an energy range of 0 to 2MeV. Logarithmic Y-scale.

When looking at the particles that struck both the front and the back sensors, a pattern can be discerned in figure 24. The turning point is located almost exactly at E_c =1 MeV. In the right figure, the efficiency of coincidence detection when the ratio is reversed after 1 MeV is seen.

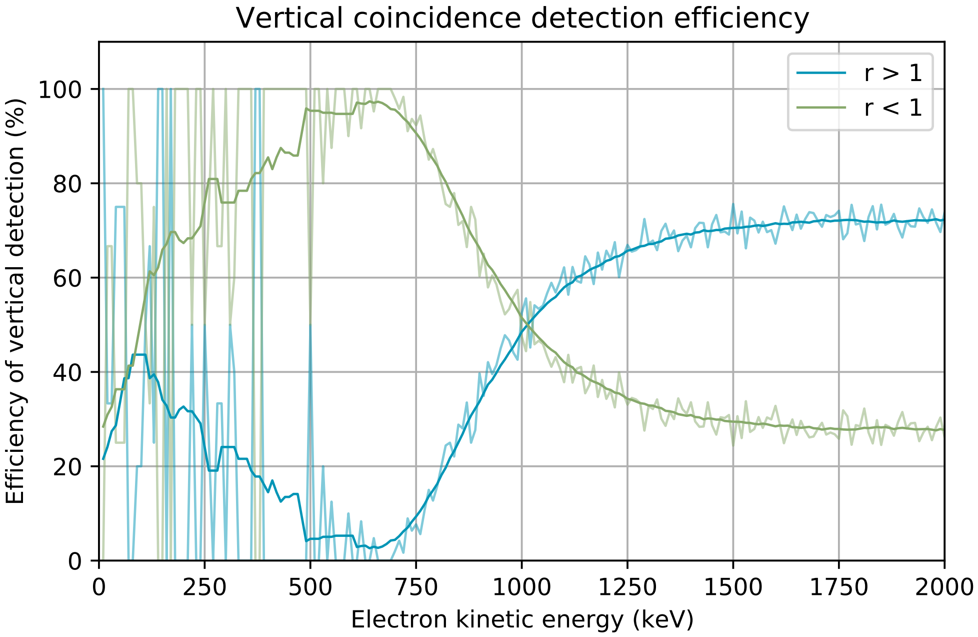


Figure 24: Efficiency of vertical coincidence discrimination for F/B ratios below and above 1. In this plot only the particles that hit both the front and back pixel array are included.

Combining these with the hits that did not scatter to the front sensor, the total efficiency of the vertical coincidence detection is shown in Figure 25.

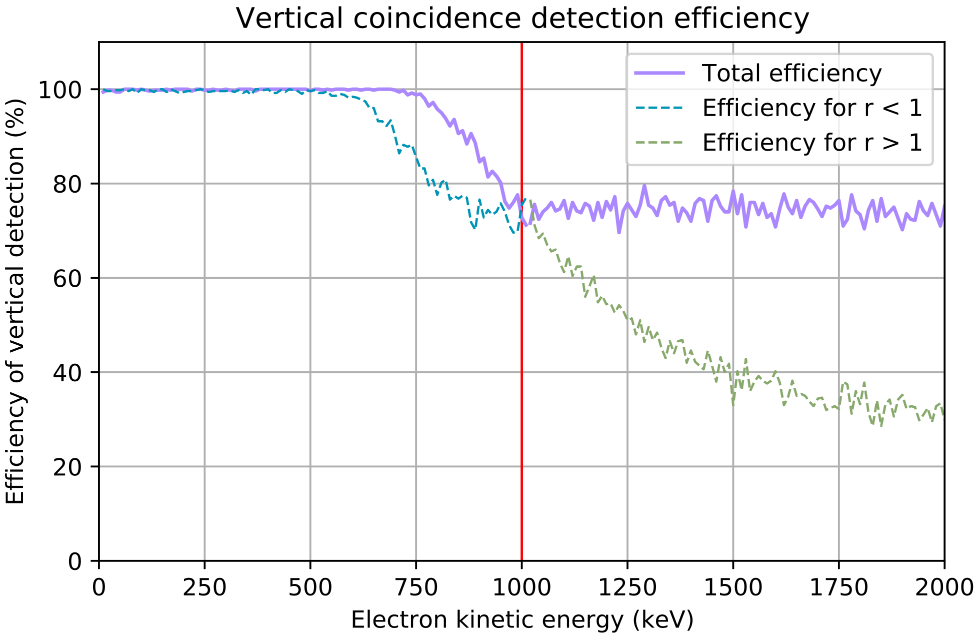


Figure 25: The total efficiency of the vertical detection efficiency including all particles, accepting different F/B ratios based on incident energy.

This simulation experiment confirms that there is a pattern that can be exploited in how the energy is distributed between the pixel layers, but this relies on a faulty premise as in the simulation the initial energy of the particle is known. In practice, this energy cannot be accurately determined, only the energy actually deposited in the sensor can be sensed and converted into an energy. Accounting for this, a second simulation was defined where the source is a uniform distribution of energies from the source. Here, the energy spawned is not known, and it is defined as the sum of energies deposited in any of the pixel sensors. The scatter plot of this simulation is shown in figure 26,

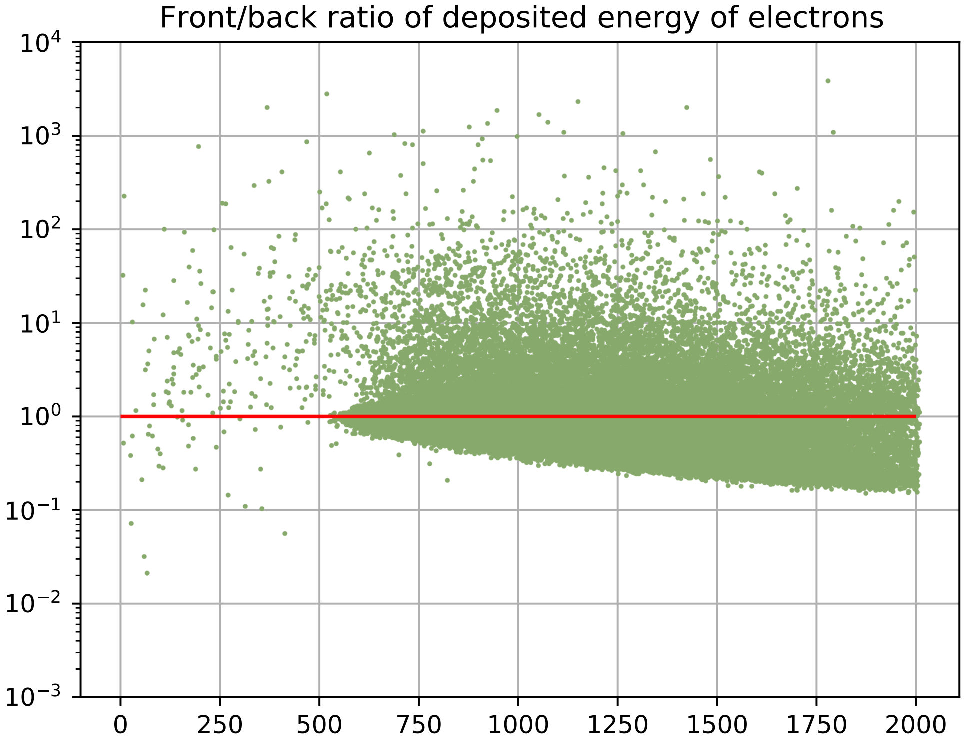


Figure 26: Simulation scatter plot of front/back ratios of the energy deposited by precipitating electrons across an energy range of 0 to 2MeV. Logarithmic Y-scale. Electron energy in keV on x-axis and front/back ratio on y-axis.

The accuracy when selecting $r > 1$ and $r < 1$ is shown in Figure 27. This plot has more noise stemming from the fact that particles can scatter away from the sensors and not deposit all of their kinetic energy in the simulation, disturbing the pattern. The thicker line is a moving average with an averaging window of 20.

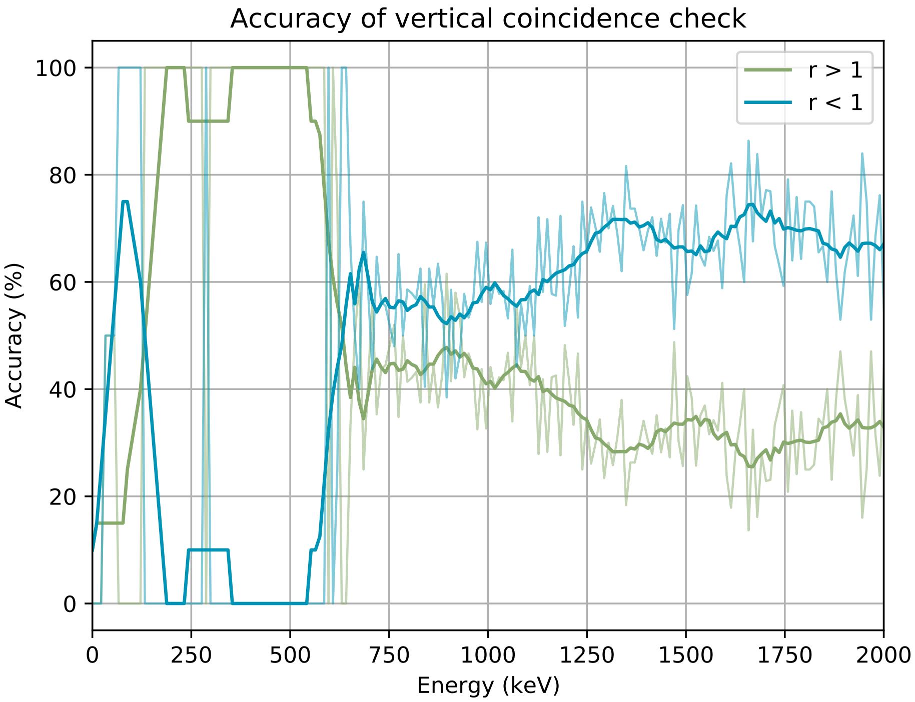


Figure 27: Efficiency of vertical coincidence discrimination for F/B ratios below and above 1. Only particles that hit both the front and back pixel array are included.

What this plot reveals is that there is little to no information to be gained by reversing the relationship at any energy point. As soon as there is a significant amount of particles that strike both pixel layers, the case for $r > 1$ quickly comes out on top. Since there is not a clear distinctive trend, this method is deemed infeasible. Even if the origin of the particle can be ascertained with 70% certainty, this means reducing the detector efficiency to 70% to stop 70% of the particles originating from the back. To investigate how common this is, 10 MeV electrons were fired into the detector house from the back. The simulation was run with 100,000 particles. A logarithmic histogram of the recorded energy bins in the pixel detectors are shown in Figure 28.

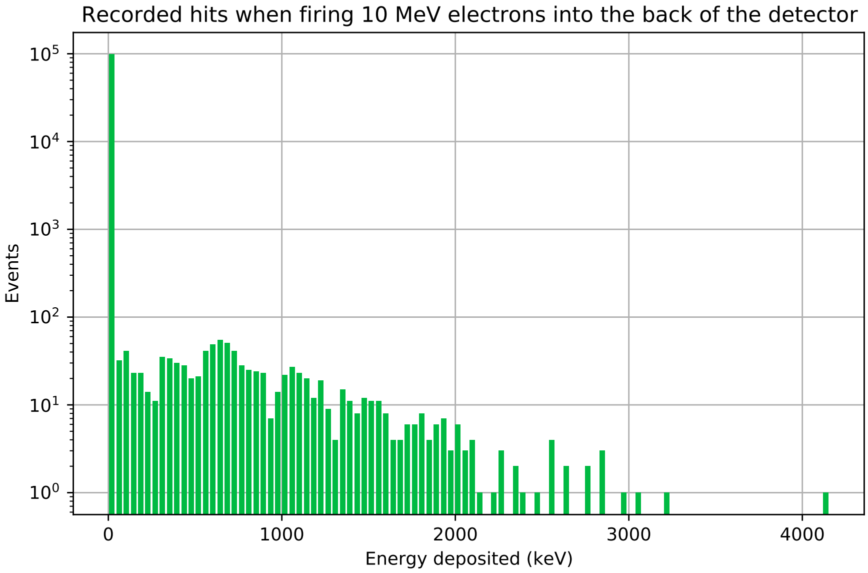


Figure 28: Deposited energy in the pixel detectors when firing 100,000 10 MeV electrons to the back of the detector house. Logarithmic Y-scale.

The number of particles penetrating the shield here are not likely to have an adverse effect on the measurement for two reasons. First of all, electrons of such high energy are relatively rare. Secondly, the simulation shows that almost all of these particles were stopped by the tungsten shield. The energy distribution discrimination scheme proposed by Andersen is thus not recommended.

5 TIME-TO-DIGITAL CONVERSION

Information in this section is gathered from [Henzler, 2010] if not otherwise specified. Time-to-Digital converters are used in a wide range of applications such as particle detectors, Time-of-Flight measurements, LIDARs and even Phase-Locked Loops (PLLs) where they serve as phase detectors. It can be implemented in many ways, but most commonly it is implemented in Field Programmable Gate Arrays (FPGAs), ASICs or Complementary Metal–Oxide–Semiconductor (CMOS) processes where control of the time domain is great. The simplest way to do a Time-to-Digital conversion is to employ a digital counter that is enabled by some input and counts clock edges. The main benefit of using a counter is the superb range; incrementing the number of bits in the counter doubles the range. The drawback is that the resolution of the conversion is limited by the clock period, requiring very high clock speeds for accurate conversion. This is impractical due to difficulty of clock generation and excessive power consumption. Methods exist to achieve sub-clock period resolution, and this is an actively researched field. There is a plethora of available choices for high-resolution TDCs, but many methods are designed for implementation in CMOS processes. Fortunately, a subset of the available methods are both physically and practically realizable in FPGA fabric, and these are described here.

5.1 Performance measures

There are several measures used to accurately quantify the performance of a TDC, and these will be described here. The first and most obvious performance measure is the resolution T_{LSB}. The resolution is a measure of conversion granularity and is defined as the time input period that corresponds to a change of 1 Least Significant Bit (LSB) in the binary output of the TDC. Similar to ADCs, TDCs inescapably must sample in a discrete fashion, and the time difference between the true analog input transition and the first subsequent sample is called the quantization error ε. Since the input signal is asynchronous, this error means that this is not an invertible relationship, i.e. the true interval cannot be repoduced exactly from the digitized value. The relationship between the input time and the binary output vector N may then be described as,

$$T_{in} = NT_{\mathrm{LSB}} + \varepsilon \tag{5.1}$$

If the input signal is not correlated with the TDC sampling clock, the quantization error is an equally distributed random error constrained between 0 and T_{LSB} with a mean value $\langle\varepsilon\rangle = \frac{1}{2}T_{\mathrm{LSB}}$. The quantization error contributes to the noise floor of the conversion, and cannot be improved without improving the resolution.

Offset error and gain error

Offset error and gain error can be described by the linear ideal transfer function $N = dt + N_{\mathrm{offset}}$. The offset is the output for $t = 0$, and can be adjusted by shifting the staircase to match the offset error. The gain error is the error in the transfer function with respect to the ideal transfer function. This happens in a delay line if, for instance, the operating temperature is increased, which in turn increases the propagation delay of each element. If a delay line is matched to the clock period, it would now be impossible for the last bits in the delay line to transition before the next clock edge.

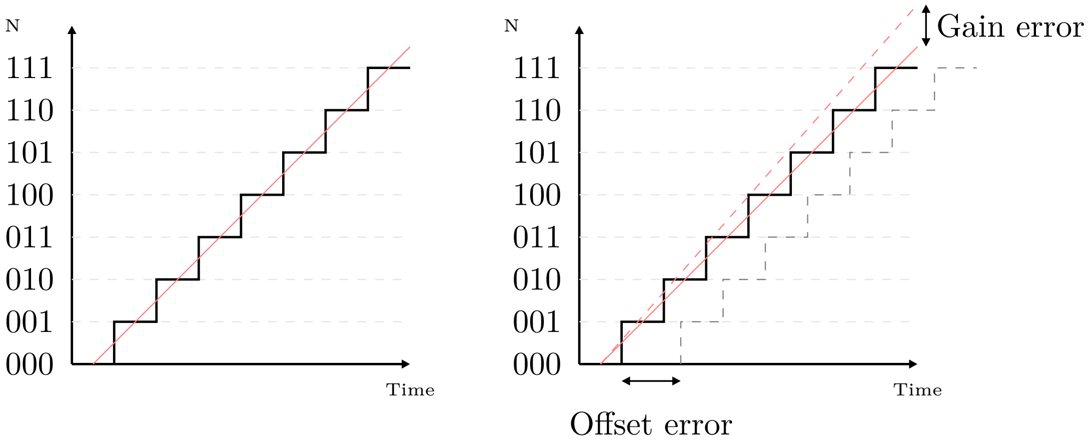


Figure 29: Left: Ideal input-output characteristic of a TDC. Right: Linear imperfections that may appear in a practical implementation.

Non-linearity

Differential non-linearity (DNL) is a measure of how much each step deviates from the nominal width T_{LSB}. It is a measure for each step, but usually a single figure is stated, which corresponds to the DNL of the worst step. It is normalized to 1 T_{LSB}, and can be both negative and positive. For instance, a step of width $\frac{1}{5}LSB$ has a DNL of -0.8.

Integral non-linearity is the stepwise distance from the ideal transfer function and the actual transfer function. In the same fashion as DNL, the number for the worst step is typically stated. It is tempting to think of this as the sum of the DNL, but the DNL contains both the gain and non-linearity error, whereas the INL only contains the non-linearity error. If every step in the transfer function is off by the same amount, the integral non-linearity would be zero, although each DNL step is non-zero.

5.1.1 Conversion in the presence of noise

The detrimental effects discussed above are obtained using static measurements, and are attained by averaging several measurements to reduce the presence of noise in the measurements. In a real situation, noise from peripheral circuitry does affect the measurement, and dynamic performance measures are required. For ADCs, these are typically SNR and Effective Number of Bits (ENOB). Measuring the SNR of an ADC is trivial, as an analogue signal, being continuous, inherently has a much higher resolution than the ADC, and a simple sinusoidal wave can be fed into the ADC. For TDCs there is no simple way to generate a sinusoidal sequence that encompasses all possible time intervals, so a different approach is needed to define equivalent performance measures.

The single shot experiment is introduced to create an equivalent of these measurements for TDCs. Instead of using a sinusoidal signal to convert each possible voltage level, a fixed time interval is repeatedly applied to the TDC. Without noise, the resulting value would be the same each conversion. The standard deviation of the difference in the actual measurements is defined as the Single Shot Precision (SSP).

The quantization error is shown for increasing timing uncertainty $\sigma_\mathcal{T}$ in Figure 30. In the ideal case, with no error introduced by noise, the lower bound on the precision is placed by the quantization error of $\pm\frac{1}{2}T_{\text{LSB}}$. This is illustrated by the experiment being carried out both when the time interval is placed exactly in the middle of a quantization interval, and when the interval is placed in the corner of a quantization interval. In the former case, there is no quantization error and the SSP approaches the ideal 0. For timing uncertainties of above $\frac{1}{2}T_{\text{LSB}}$, there is little difference, and single-shot precision increases linearly with timing uncertainty. Because of this, SSP is often used as a real measure of resolution.

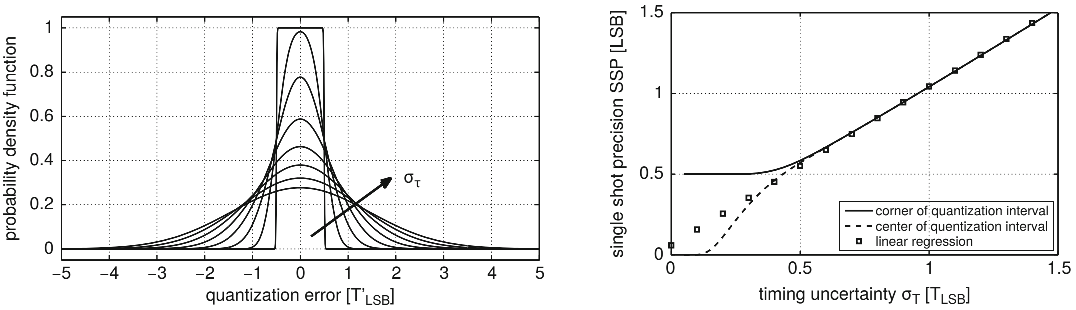


Figure 30: Left: Time interval dependent probability quantization error function. Right: Single-shot precision. [Henzler, 2010, p. 29, 31].

5.2 Calibration methods

Ideally each conversion step should be of equal width as shown in Figure 29 (left). One shortcoming of implementing a TDC in an FPGA is that the steps will have varying width, resulting in a high DNL compared to a CMOS implementation where control over delay and routing is much greater. The primary cause of DNL in an FPGA is uneven routing delays between cells. This is very pronounced in those interconnects that bridge logic cells together, resulting in what is known as *ultrawide bins*. Because of the uneven nature of FPGA circuits, it is very important to calibrate the bins to the circuit. Another worry is the variance of Process, Voltage and Temperature (PVT) affecting the speed of the circuit. While process variation can be treated as static in an FPGA (unless programming multiple FPGAs with the same configuration), temperature and voltage cannot. In a review paper of TDC methods, four methods to calibrate the bin widths are described [Tancock et al., 2019]. One of these is manual calibration which is not of interest because it cannot be performed by the FPGA post-programming and thus does not solve the PVT dependency. The remaining three are described in the following sections.

5.2.1 Statistical calibration

By inputing many hits that are known to be uniformly spread, the count of each bin should statistically approach a uniform distribution as the number of hits N gets large. The hits can be generated by anything asynchronous to the sampling clock, but the simplest way to generate them is to include a second and completely separate oscillator on the PCB to ensure complete phase independence. This clock should also be of a lower frequency than the TDC clock to ensure that a maximum of one transition happens for each TDC sampling clock period. Using the clock transitions as input to the TDC calibration, many asynchronous hits can be generated in a short time. Actual hits from the sensor can also be used, but the calibration would be slower because of the lower hit rate. By feeding numerous hits into the TDC, a histogram is generated of all the bin widths and their corresponding bin count. Because the input is uniformly distributed in time, the ideal TDC will generate a histogram in which every bin is of equal height. An uncalibrated TDC with non-zero DNL will generate a histogram with uneven bin counts. The bin counts of each bin is directly proportional to the delay of the corresponding cell. The width of the i'th bin τ_i is given by,

$$\tau_i = \frac{n_i T_p}{N} \tag{5.2}$$

where n_i is the number of hits in the i'th bin and T_p is the clock period. The transfer function of the conversion is then given as,

$$H(n) = \frac{\tau_n}{2} + \sum_{i=0}^{n-1} \tau_i \tag{5.3}$$

This equation gives the delay of each element before the current element in the line, centered on the current bin by the term $\frac{\tau_n}{2}$.

The number of hits M required to perform an accurate calibration can statistically be determined by requiring a given tolerance β and confidence α by the following equation,

$$M = \left(\frac{z_{\alpha/2}}{\beta}\right)(2^B - 1) \tag{5.4}$$

Here $z_{\alpha/2}$ represents the area under a Gaussian distribution outside the confidence interval given by α.

5.2.2 Double registration

If the delay line is made twice as long as the clock period, the same hit can be registered twice on two subsequent clock edges. The assumption is that the DNL of one bin is likely not the same as the corresponding bin in the second registration. The width of the bin is then given as the average of the two,

$$\tau_i = \frac{T_p}{K_{2_i} - K_{1_i}} \tag{5.5}$$

Where K_1 and K_2 are the positions of the two detected transitions.

5.2.3 Sliding scale

The sliding scale technique is a method to average the bin widths of the delay line. Like statistical calibration, this method requires asynchronous input signals. Unlike statistical calibration, however, it does not calculate the bin-by-bin widths, instead generating an average bin width across the delay line.

The principle of operation is that the same asynchronous pulse is applied repeatedly but with different delay offsets, generating multiple codes for the same pulse width, shown in Figure 31. By subtracting the respective delay offset from the generated codes, different codes are attained that are generated on exactly the same signal. Finally, the codes are averaged, and an average bin width is calculated [Tancock et al., 2019]. This does not account for ultrawide bins as it creates an average width for all bins, and will not perform well if any bin is substantially larger than others such as commonly is the case in FPGA implementations. There are still examples of this being used on FPGAs; [Tontini et al., 2018] applied this calibration method successfully in a FPGA implementation.

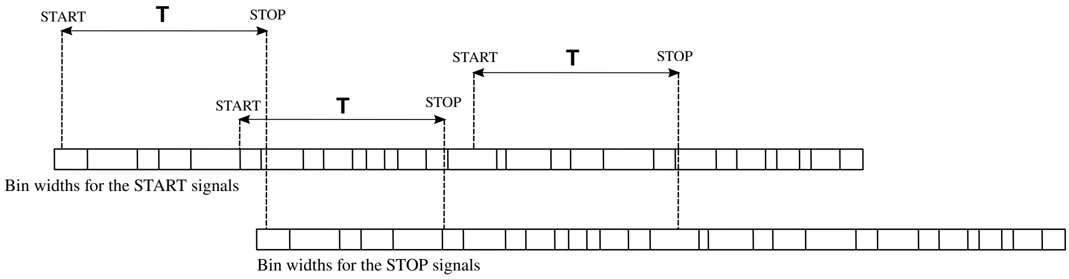

START
T
STOP
START
T
STOP
START
T
STOP
Bin widths for the START signals
Bin widths for the STOP signals

Figure 31: Sliding scale averaging, adapted from [Tancock et al., 2019].

5.3 Other potential issues

While the ideal delay line converter produces a thermometer code, mismatches in signal propagation speeds, clock skew and metastability of the sampling registers may result in what is known as a bubble error in the thermometer code, illustrated in Figure 32. In this case, there is not a clearly defined transition in the delay line, and determining the point in which a thermometer code transitions is not practically realizable. An alternative method to get a meaningful reading is by counting the number of ones in the thermometer output. If there is a bubble early in the thermometer code, it will not have a large impact on the conversion.

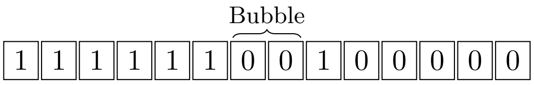


Figure 32: Bubble formation in thermometer code.

Carra et al. used a Wallace tree encoder consisting of full adders to count the number of ones in the thermometer code [Carra et al., 2019]. An alternative but slower approach is to feed the thermometer code into a clocked shift register and counter that cycles through each bit and counts the ones, which is fine in a pipelined solution.

5.4 Clock interpolation

A method of increasing the range of a TDC is by employing clock interpolation. In this hybrid solution, the high range of a low-resolution TDC is combined with the high resolution of a low-range TDC. The coarse TDC is typically implemented as an n-bit counter, while the fine TDC is any high-resolution TDC such as a multitapped delay line. Both of these are connected to the same start and stop signals. The coarse TDC counts the number of positive clock edges between the start and stop signals, while the fine TDC is used to determine at which point inside a clock period the transition happened. In order to achieve this, it must trigger asynchronously on transition changes and latch the output. If the input to the fine TDC is synchronized in any way, the resolution of the whole TDC will be reduced to that of the counter. To avoid metastability, the output of the fine TDC is fed through a synchronizer circuit before interfacing with the general digital circuitry.

5.5 Asynchronous oversampling

A simple method to improve the resolution by a factor n is by utilizing asynchronous oversampling [Balla et al., 2014]. The main clock feeds into a counter that is enabled by the input signal. By also sampling the signal at off-phase periods in time, each clock period can be divided into n distinct parts, improving the resolution by a factor of n. Figure 33 shows an implementation with 4x asynchronous oversampling. This same principle is also described as a Serializer-Deserializer (SERDES) TDC by [Tancock et al., 2019]. In this implementation, multiple SERDES blocks sample the same signal asynchronously, with the same result.

Within each clock period, the outputs of each row successively rise to a high logic level with a difference of $1/n$ of the clock period. If the input signal changes value within a clock period, the outputs of the different clock samples will reveal in which partition of the clock period the event happened. With a 1 ns clock period and 4x oversampling, this would give a conversion resolution of 250 ps. Asynchronous oversampling is a low resource implementation that gives a slight improvement to the resolution, but the amount of clocks that can readily be generated and distributed is a limiting factor. Low clock skew and good clock distribution are essential in not introducing a source of error in the conversion. Generating several high-frequency clocks comes with a penalty of higher power consumption. Reliably generating the off-phase clocks also requires a Delay-Locked Loop (DLL). Delays can be used to generate these clocks, but this introduces errors due to propagation delay drift with temperature and voltage.

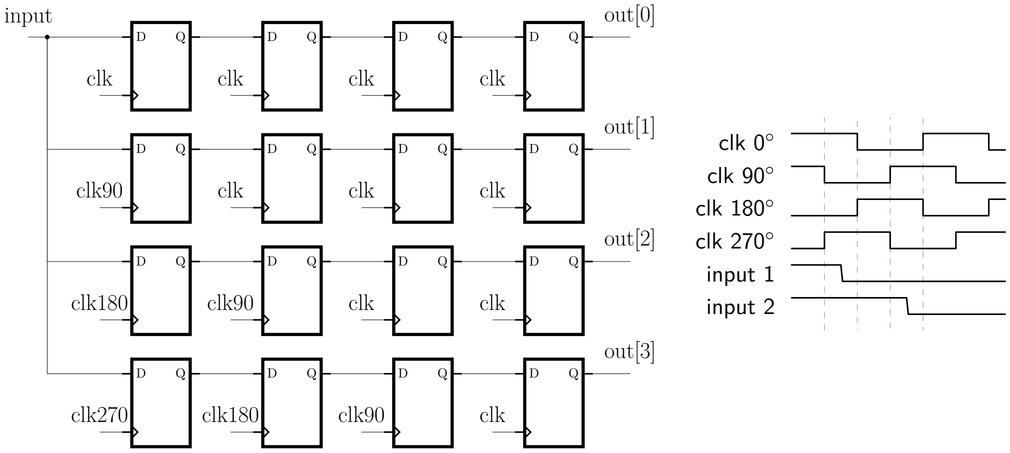


Figure 33: Asynchronous oversampling with four clocks. Input 1 and 2 both go low in the same clock period, but can be differentiated by the oversampling circuit. Adapted from [Balla et al., 2014].

5.6 Multitapped delay-lines

The multitapped delay line method involves feeding the input into a chain of buffers called a delay line. When the input changes, each of the buffer outputs transitions separated by a delay τ. At the next clock edge, the outputs are sampled by registers. The principle is illustrated in Figure 34.

It is challenging to create accurate delay lines in FPGA fabric, and resource utilization can be quite high. The main difficulty in implementing a multitapped delay line is that routing may introduce differences in propagation delays between buffers, which will be most pronounced if the buffer chains are long and individual propagation delays short. In addition, the propagation delays vary with PVT. To overcome these challenges, it is imperial to perform calibration on the conversion bins, which will alleviate the PVT dependencency and also reduce wide bins.

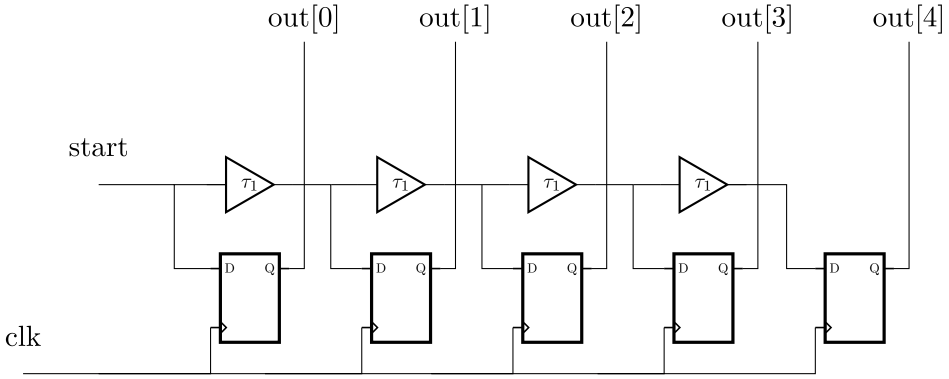


Figure 34: Multitapped delay line, adapted from [Henzler, 2010, p. 15].

It is worth noting here that a multitapped delay-line may also be implemented by delaying the clock and sampling with a stop signal. In practice this means feeding the comparator input into the clock input, and the clock into the data input.

5.7 Vernier delay-lines

An inherent problem with the simple delay line is that a delay step can never be lower than the gate delay of the delay units. To push resolution to sub-gate delays, a delay line known as a Vernier Delay Line (VDL) can be applied. The VDL is named after the Vernier scale, a widely used method by calipers to interpolate lengths that fall between graduation marks. It consists of two delay-lines, one for the start signal, and one for the stop signal, which may also be the clock if only a start signal exists. The Vernier delay line is shown in Figure 35.

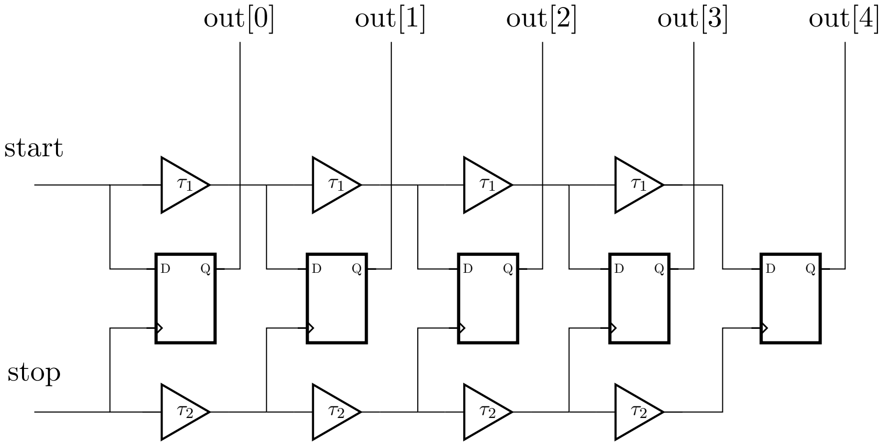


Figure 35: Vernier delay line, adapted from [Henzler, 2010, p. 74].

The delays τ_1 and τ_2 are chosen such that $\tau_2 < \tau_1$. Because of this, the stop signal chases the start signal, and will eventually pass it. The point in time when the stop and start signal arrive at the same time is detected by early-late detectors implemented with D registers. At each step the stop signal catches up to the start signal by $\tau_1 - \tau_2$. This means that the first output will transition if $t < \tau_1 - \tau_2$, and the second output transitions if $\tau_1 - \tau_2 < t < 2(\tau_1 - \tau_2)$, and so on. This defines the resolution of the TDC,

$$T_{\text{LSB}} = \tau_1 - \tau_2 \tag{5.6}$$

This can be made arbitrarily small, depending on how closely the two different delays can be designed. Designing the small delay difference is easy in CMOS processes by varying the transistor length-to-width ratios, but it can be challenging to implement a consistent delay difference in an FPGA. [Xie et al., 2005] connected an unused gate as a dummy load to increase capacitance on each node, thus increasing the propagation delay of each step. The higher resolution also comes with a higher resource cost. The sum of stages must be at least equal to either the longest interval that needs to be measured, or in the case of clock interpolation, the clock period. If the difference between the two delays is very small, the delay line must subsequently be very long to not reach the end before the stop signal.

5.8 Looped delay lines

In the linear TDCs discussed above, the length of the linear chain of delay elements increases with the length of the time interval to be measured. In order to increase the range of the counter without the occupation of FPGA resources increasing linearly with it, feedback can be included to create what is known as a looped delay line. In this topology, the start event traverses through the same delay line several times, and each traversal is recorded by a loop counter. The principle of the looped delay line is shown in Figure 36. This topology may be combined with a VDL to create a looped VDL.

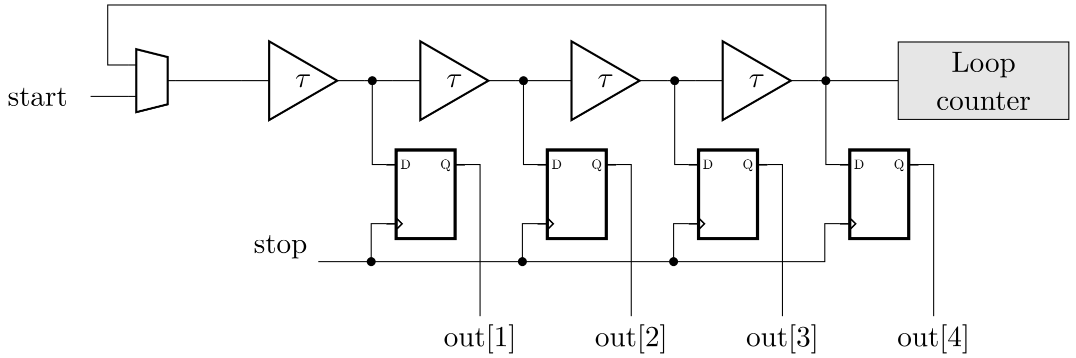


Figure 36: Looped delay line, adapted from [Henzler, 2010, p. 46].

The multiplexer is configured to pass the start signal. A control unit is required for the multiplexer that enables the feedback loop when the start signal has entered the loop. At each cycle of the delay line, a loop counter is incremented to keep track of the number of cycles through the delay line. This can also be combined with the Vernier technique to create a looped Vernier delay line. When the stop event occurs, the delay line and counter are sampled and the multiplexer is set to pass the start signal again.

This type of TDC is physically realizable in an FPGA, but only in limited scale. The output from the delay line that clocks the counter has to be considered a local clock, and in FPGA fabric it must be routed through the clock distribution network. Since only a limited number of clock lines exist in an FPGA, the routing tool will quickly fail if this TDC is scaled to many channels. This TDC is similar to clock interpolation, but it is self adjusting because a full period is decided by the delay line propagation and not any external clock.

6 PCB DESIGN AND LAYOUT

This section contains both the deliberations and decisions regarding the design of the readout motherboard PCB. It was designed using the Mentor Xpedition Enterprise software suite.

Are Haslum designed the flex PCB that houses the pixel detectors and pulse shaper ASIC, and in addition to this he designed the previous revision of the DEEP motherboard. Both of these PCBs were present on the ICI-5 rocket. The main issue observed with the electronics was noise on the detector, and the SNR was too low to accurately detect electrons with less than 60 keV energy. The board was based on the Time-over-Threshold method and had a ProAsic3e FPGA for the data acquisition. A new revision of the board is desired to implement a few key changes.

First and foremost, the ProAsic3e FPGA is replaced with the NG-MEDIUM BRAVE FPGA from NanoXplore. This is a new and promising FPGA specifically designed for operation in radiation heavy environments such as space. It was made commercially available in 2018, and it provides a promising alternative to existing FPGAs. ESA is currently investigating its feasibility for adoption in its own projects through the QUEENS-FPGA evaluation project [Maragos et al., 2018]. In addition to this, the Microelectronics group at UiB is designing radiation shutter electronics for the SMILE project featuring the NG-MEDIUM. Using the same FPGA in both projects allows the reuse of several systems common for both. With the change of FPGA on the PCB comes a complete redesign of the power distribution network, the reset network, programming interface and programming memory as well.

Secondly, it is vital to reduce general circuit noise, particularly on the comparator thresholds and detector reverse bias. As the previous revision did not have a sufficient SNR on the comparators to detect the low ranges, changes must be made to reduce the noise. This is done in conjunction with the need to rework the power delivery network, and special care is taken to create silent power supply lines.

A tertiary aim of this revision of the PCB is to add a general IO interface between the PCB and the NG-MEDIUM evaluation kit. This is included to enable the use of a headless mode where the PCB can be connected to and controlled from the evaluation kit, removing the need to solder the FPGA to the PCB for initial stages of development. Without including the FPGA or components supporting the FPGA such as power supplies, the component cost of the remaining circuit is much lower, making PCB respins comparatively cheap.

An overview of the PCB circuit is shown in Figure 37.

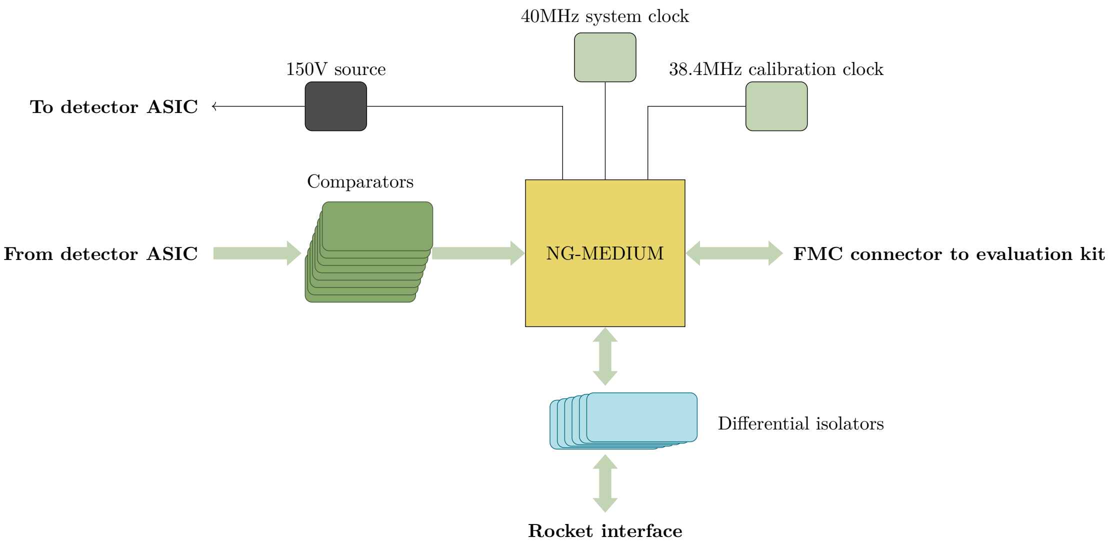


Figure 37: Circuit overview.

6.1 PCB specifications

There are a number of requirements that the PCB has to satisfy for correct operation and these are outlined in section 3.5.

6.2 PCB stackup

The PCB is a mixed signal board with a form of Analog-to-Digital conversion present at the ToT comparators. A common strategy for mixed signal boards is to keep a physical division between the analogue and digital circuits. When considering the layers of the PCB, this means keeping digital power and ground planes on one side, and the analogue power and ground planes on the other. [Brooks, 2003, p. 304]. Furthermore, Brooks presents two basic rules to abide by in order to achieve a low noise PCB design:

1: Every high-speed trace must be referenced to a continuous plane. The reasons include loop area control, impedance control and crosstalk control.

2: Every power supply should have a parallel power-ground capacitive plane pair. The reasons include help in capacitive decoupling circuits and for electromagnetic interference control.

The only way to meet these two requirements for each of the digital and analogue parts is to place the power and ground planes next to each other (rule 2), and the signal layer next to the continuous ground plane (rule 1). One such solution is the stackup shown in Figure 38, which is one of the 6-layer stackups recommended for signal integrity by [Hartley, 2000]. Analogue components such as the ToT comparators are placed on the top layer, while digital components like the FPGA are placed on the bottom layer.

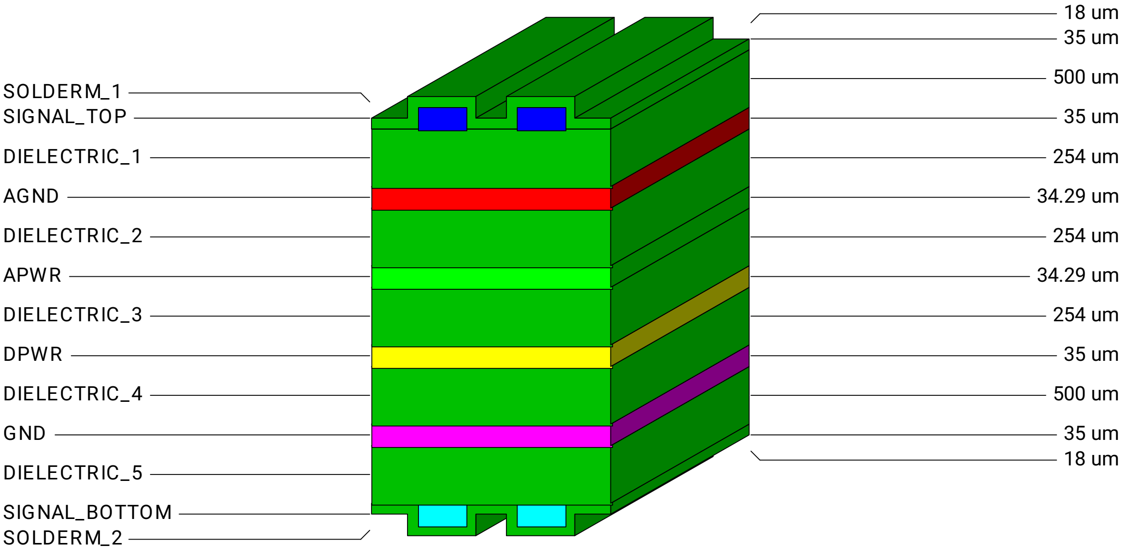


Figure 38: DEEP PCB stackup exported from Mentor Graphics Layout tool.

6.3 FPGA

The FPGA on the PCB is the radiation hardened NG-MEDIUM (NX1H35AS) from the French company NanoXplore. It is produced in the C65SPACE process from STMicroelectronics [Nan, 2020c, p. 7]. This is a 65 nm CMOS process that is radiation hardened and suitable for space applications [STM, 2015]. The FPGA was space qualified according to ECSS requirements as a result of the EU-funded VEGAS project in 2020 [veg, 2020], making it the first fully European-developed FPGA to be qualified for space operation.

The NG-MEDIUM comes in four different packages; the 625-pin land grid array (LGA-625), ball grid array (BGA-625) and column grid array packages (CGA-625), as well as the 352-pin ceramic quad flat package (CQFP-352). The package used in this project is the CQFP-352. Although the pin count is lower and the package is physically much larger, it is more resistant to the vibrations of the rocket during launch since the metallic leads of the package serve as load bearing elements [Ghaffarian and Evans, 2014]. Other benefits to this package include the fact that it can be soldered manually, and the pins can be probed. It is also possible to visually inspect solder joints and repair any weak contacts.

The CQFP-352 package is shipped as a lead frame, and has to be manually fitted onto the footprint of the package by crop-and-form. This requires specialist tooling that is not available at UiB and will be done by an external contractor. It is relatively expensive, and factoring in the self-cost of the FPGA itself, the cost of an assembled PCB is high. To enable development without physically fitting the FPGA to the PCB, a socket has been included. This connects directly into the socket on the NanoXplore NG-MEDIUM evaluation kit, and each signal going to the FPGA is dually routed to the socket. Two modes of operation are thus supported with the same PCB. With the FPGA mounted to the PCB, it operates as it will when integrated with the rocket. With the socket it functions as an expansion card to the evaluation kit that can be used for development and functional testing of the FPGA configuration. The socket will not be mounted during launch, but it introduces stubs as the signals are routed to the FPGA Mezzanine Connector (FMC). Stubs can cause issues with impedance matching and electromagnetic interference.

The socket on the evaluation kit is of the full 400-pin version, but since there are relatively few connections needed in this design, and with the added benefit of reducing pad count, the 160-pin version of the socket was selected for the DEEP PCB. The 160-pin socket is the same mechanical socket as the 400-pin connector, but whereas the full connector consists of ten rows of 40 pins labelled A-K, the 160-pin version only includes the C, D, G and H rows. Only pins that are available as pins on the evaluation kit have been used in the design to avoid having to maintain two separate pinouts. Ideally, the behaviour of the circuit in FMC mode should be identical to how it operates when fully integrated with an onboard FPGA. Small differences between the two modes are expected. The package of the FPGA on the development kit is of the 625-pin grid array type, meaning that the available IO banks are different. In addition to this, some difference should be expected due to the binning difference of the two FPGAs, as well as the difference in the signal path to peripheral circuitry, especially considering that in the FMC mode the signals have to cross boards.

To properly secure the FPGA package to the PCB, four mounting holes are included directly underneath it. These will be used to glue the FPGA to the PCB to increase vibration tolerance.

The DEEP PCB is shown connected to the NG-MEDIUM development kit in Figure 39. Note that in this photograph, not all components are mounted, as some are only needed when the actual FPGA is soldered onto the PCB.

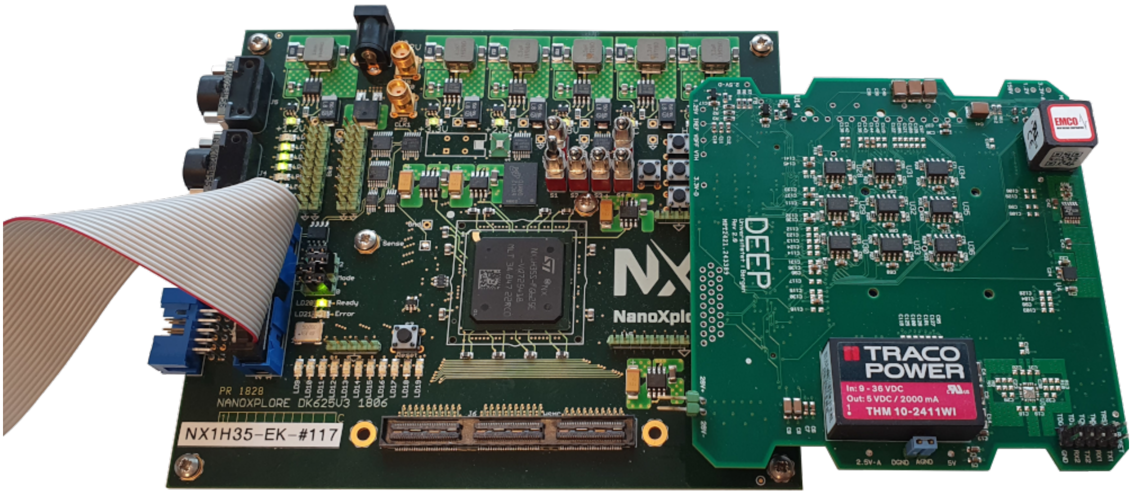


Figure 39: DEEP PCB connected to the development kit with the socket. Ribbon cable to the left goes to the ANGIE programmer.

Test and probe access to the signals going between the boards is provided by an FMC breakout board sandwiched between the evaluation kit and the DEEP board.

6.3.1 Programming interface

NG-MEDIUM is a Static RAM (SRAM) based FPGA, and the configuration bitstream is transmitted to the onboard SRAM, where it is read by the on-die configuration engine. This can be done through several interfaces, including Joint Test Action Group (JTAG), Serial Peripheral Interface (SPI) and SpaceWire. The package has four pins named MODE[3:0] that are sampled at power-up. The state of these pins decide which configuration mode the FPGA should enable, but JTAG is always available regardless of MODE [Nan, 2020a]. Because the configuration memory is SRAM, the configuration is volatile and does not persist through a power-cycle. To keep the configuration permanent, an external flash memory is connected to the SPI interface. NanoXplore recommends the S25L256L serial NOR flash as a configuration memory (Email correspondence with support via Kjetil Ullaland 15-02-21). MODE is set by pullups and pulldowns on the PCB, and can be toggled between the nominal SPI master mode and 8-bit slave parallel mode by a short for flash memory write operations. When the master SPI is enabled and a bitstream exists in the flash memory, the bitstream is automatically fetched from the flash memory by the configuration engine, and loaded to the configuration SRAM.

6.4 Comparator circuits

There are nine comparator circuits on the board: eight for each pixel sensor, and one test input. It is important that the comparators have a low propagation delay to enable accurate readings of the short pulses from the CSA. The selected comparators are of the type AD8561. This comparator is specified with a 7 ns propagation delay, and has a space-grade equivalent in AD8561S, making it easy to swap this component for the full space-grade implementation of the particle detector. It requires a supply voltage in the range between 3 V and 10 V. Since there are 3.3 V components on the board requiring a power line of this voltage, the same is used for the comparators. The propagation delay increases for decreasing supply voltage, and is specified to 8.5 ns for 3 V. The comparator uses TTL logic, but it has both an inverted and non-inverted output [Ana, 2014]. Both of these are routed directly to the positive and negative pin of the same IO buffer on the FPGA to enable the possibility of specifying a return path for the current that is isolated from the ground plane. Isolating the signal avoids the possibility of one comparator causing ground bounce noise affecting all comparators [Spieler, 2005, p. 398]. Return current is guaranteed in differential mode but this cannot currently be used because the output current of the AD8561 is too large, measured to be 20-25 mA in shorted operation. The NG-MEDIUM supports programmable series termination, but this cannot exceed 100 Ω. A differential mode of operation may be enabled in a future revision of the readout board by adding proper series termination resistors of 350 Ω. Both the outputs of each comparator are configured as single-ended inputs so that while one output sources current to the FPGA, the other output sinks current nearby. Since each output goes to the same IO buffer, the theory is that these should cancel out and if not eliminate ground bounces, at least reduce them.

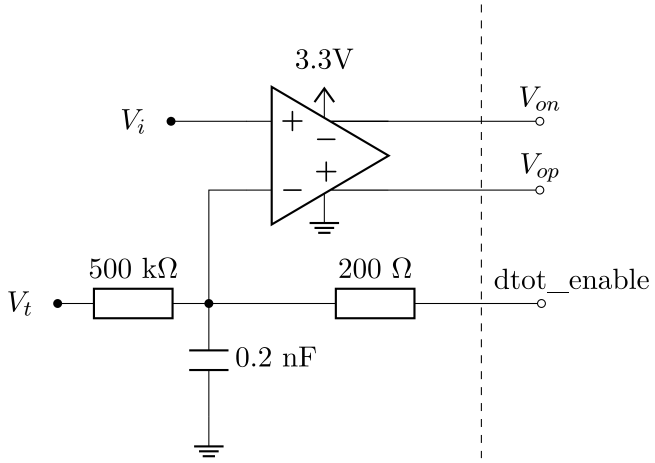


Figure 40: Comparator circuit.

The threshold of the comparator is connected to a globally distributed threshold that is common for comparators, as shown in Figure 40. To add dynamic ToT support, each threshold has a series resistance of 500 Ω to the global reference point. This is to reduce the impact on the other comparator thresholds when switching the dynamic threshold locally. The output pin for the dynamic threshold enable is connected with a series resistance of 200 Ω a shunt capacitance of 0.2 nF. This gives the dynamic threshold a time constant τ of 40 ns when enabled, which is the same as the time constant in the CSA pulse.

6.5 Power delivery network

Power to the board is supplied through a 28 V rail in the D-SUB connector to the battery onboard the rocket. Since the main power line must be galvanically isolated from the rocket, the THM-10 isolated DC/DC converter converts this to a 5 V main power line. Downstream of the main 5 V rail, there are several derived power lines required by the circuit. The FPGA itself requires a main 1.2 V supply, as well as an auxiliary 2.5 V analogue supply. Each I/O bank used needs a supply. Since the I/O supply on the development kit is configured to be 2.5 V, the same voltage has been used for the I/O banks on the motherboard in the interest of compatibility. Lastly, a 3.3 V supply is needed for some peripheral ICs such as the AD8561 comparators and SRAM. The 2.5 V and 3.3 V rails have been separated into analogue and digital domains to reduce noise on sensitive analogue components. An overview of the power distribution network is shown in Figure 41.

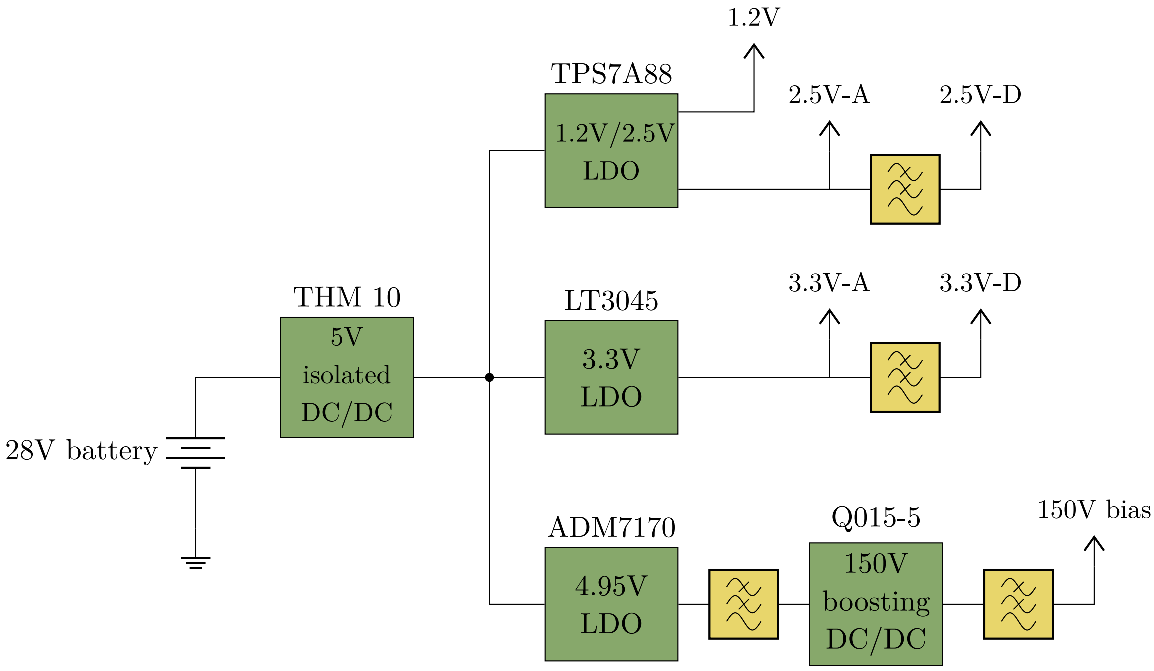


Figure 41: Overview of the power distribution network on the PCB. The yellow boxes indicate low-pass filters.

In converting the supply to the low voltages required by the circuit components, there are two main converter topologies to consider; switched-mode converters and linear regulators. While switched-mode regulators can achieve far higher conversion efficiency than Low Dropout Linear Regulators (LDOs), they contribute a significant amount of noise through the harmonics of the switching operation inherent to this type of regulator [Tran, 2010, p. 69]. The TDC conversion is very noise sensitive, so the sensor and comparator circuitry should be powered by a linear regulator to limit noise. The main shortcoming of the linear regulator is that because of the low efficiency, it has to dissipate a large amount of power, and thus, heat. The power dissipated is the difference in power in and power out. Since the input current is the same as the output current, the dissipated power can be described as

$$P_D = I_{\text{out}} \left(V_{\text{in}} - V_{\text{out}} \right) \tag{6.1}$$

While power efficiency is important, maintaining a quiet and noise-free environment is even more important in a detector instrument, so all power lines that directly interface with circuit components are selected to be LDOs. This includes the TPS7A88 dual-rail LDO for the main FPGA power lines, the ultra-low noise LT3045 for the 3.3 V comparator rail, as well as a 5 V ultra-low dropout ADM7170 for the detector reverse bias. The low dropout means that the voltage drop over the regulator is so low that it can regulate its output almost to 5 V with a 5 V input from the main 5 V rail.

6.5.1 Bias supply

The pixel detectors require a high voltage reverse-bias. It needs to be high enough to dominate the built-in potential in the diode and minimize trapping and recombination of ionized charge. It is especially critical to achieve low noise and low ripple on this bias because a change in the bias voltage ΔV will induce charge in the sensor according to (6.2), which is amplified by the CSA,

$$Q = C_d \Delta V \tag{6.2}$$

Where C_d is the capacitance of the diode.

The bias voltage is generated by the XP-POWER Q015-5, a 5 V to 150 V boosting DC/DC converter, which was also present on the previous iteration. Since the bias voltage was too noisy to accurately detect particles in the 30 keV to 60 keV energy channel, the filter of the bias supply has been redesigned to reduce the ripple and noise. This includes the addition of a Pi filter with a ferrite bead on the input, as well as adding to the RC network on the output. The attenuation of the new filter-design compared to the previous iteration is shown in Figure 42.

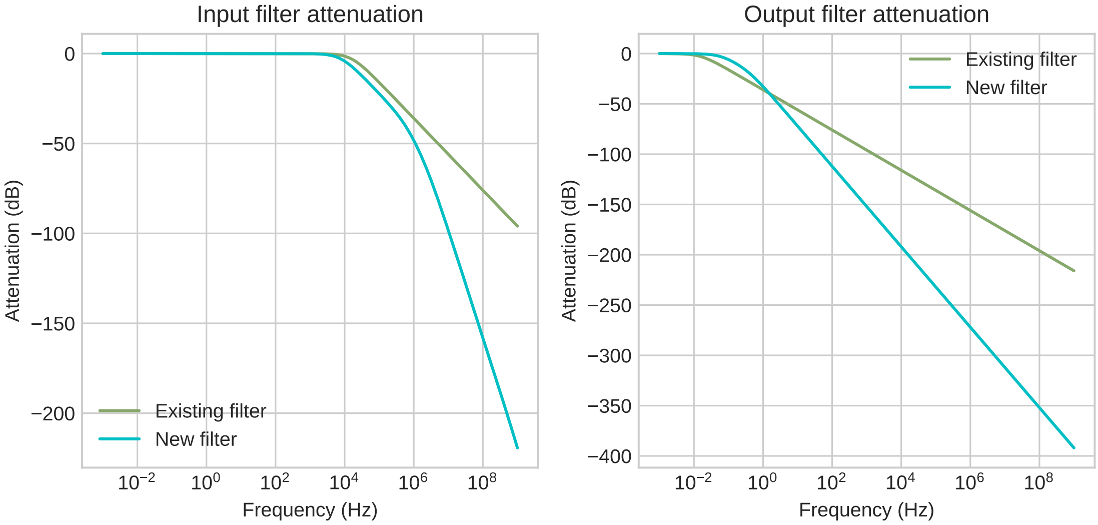


Figure 42: SPICE simulation of input and output filter attenutation of the filter on the first version of the DEEP PCB (green) and the new improved filter network (blue).

Another concern is that the DC/DC converter does not regulate the output to changes on the input, so ripple on the input appears as ripple on the output. The Q015-5 has an inherent ripple on the output of 0.5%, corresponding to 750 mV unsmoothed [XP-, 2020]. In addition to this, the THM-10 specifications states a 30 mV peak-to-peak ripple on the output, corresponding to 0.6% on the 5 V output voltage. To prevent this ripple from being superimposed on the bias output, the ADM7170 linear regulator post-regulates the 5 V output from the DC/DC converter. This LDO was selected because it has a very low dropout voltage of 42 mV, which is the voltage drop from the input to the output. This allows the LDO to regulate the 5 V rail to an output voltage that is sufficiently close to 5 V, while smoothing the rail in the process.

High voltage circuits are susceptible to corona discharge and must be turned off during the critical launch phase. The enable pin on the ADM7170 is used from the FPGA to disable the 150 V source by cutting off the input to the Q015-5 DC/DC converter. Andøya Space Center recommends that high voltage circuits are not turned on until the rocket reaches an altitude of at least 80 km [And, 2018]. For this purpose, the rocket supplies a 28 V timer signal that is asserted at a given altitude. This feeds into a differential receiver and subsequently into the FPGA. When this signal is asserted, the 150 V source is enabled at a safe altitude.

6.6 Reset network

There are two reset signals on the board, RST_N and USR_RST_N. The purpose of RST_N is to reset the FPGA, and the purpose of USR_RST_N is to initialise all logic upon power-up. Whenever the FPGA is programmed or powered on, the RDY signal will initially be pulled down, and then driven high when the device enters user mode and is ready for operation [Nan, 2020b, p. 11]. This signal is delayed and fed back into the FPGA, as recommended by NanoXplore. The signal is delayed by Schmitt triggers applied as buffers. This delay is decided by the propagation delay of each Schmitt trigger as well as the time constants of the RC networks on the inputs of the Schmitt triggers. The reset network is illustrated in Figure 43.

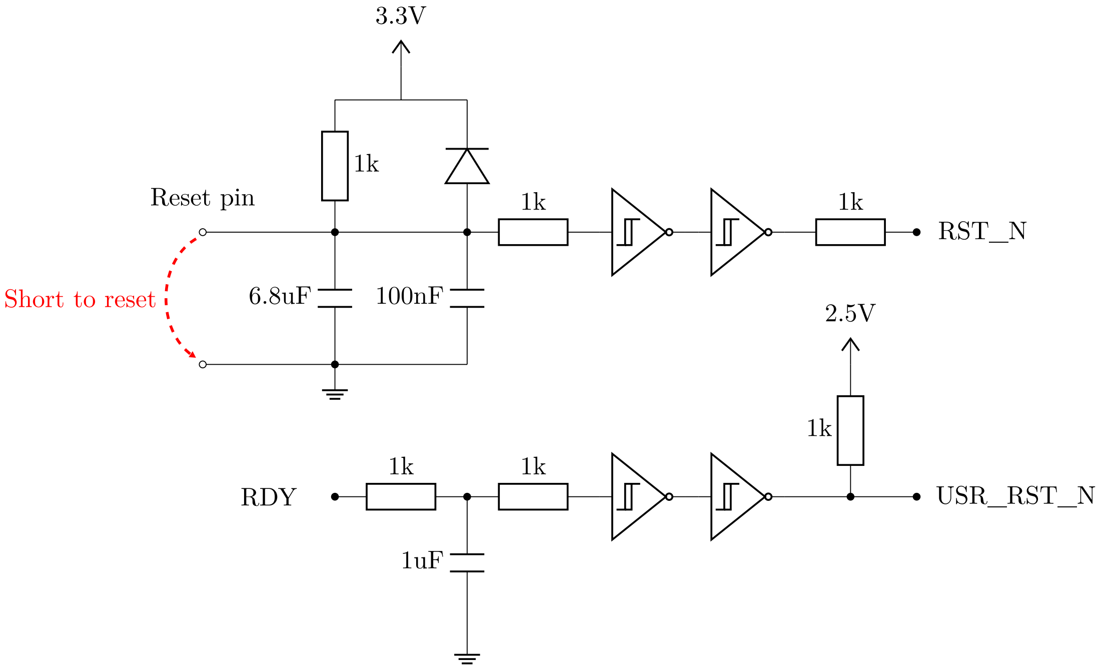


Figure 43: Reset network. The switch is not a physical switch, but a reset pin that can be shorted to ground.

6.7 Rocket interface

The readout board is connected to the rocket through a male 26-pin D-SUB connection. This connection supplies power to the board from a 28 V battery, and implements communication points between the readout board and the PUSEK board onboard the rocket. The PUSEK board includes a PCM encoder for communication with the ground station. It is a requirement to galvanically isolate all contact points with the rocket circuitry in order to not disturb the rocket circuitry. In the previous iteration of the PCB, the differential optocouplers HCPL0631 were used for inputs to the board. These have been replaced by the IL 611 isolators, as recommended by the preliminary interface specification sheet from

Andøya Space Centre. These have an input impedance of around 85Ω at 25°C and are thus close enough to the characteristic impedance (100Ω) of the twisted pair cable connecting the boards [Lindahl, 2021]. There are current limiting resistors on the PUSEK board, so there is no need to include additional series termination on the instrument side. For outputs from the board, the previous SN65LBC179 differential driver has been replaced by ISL3179E, also in accordance with ASC recommendations. An illustration of the interface is shown in Figure 44. The rocket electronics consist of exactly the same circuits, and the illustration is valid for both directions, with the instrument board and rocket electronics switching roles between transmitter and receiver depending on the direction of the signal.

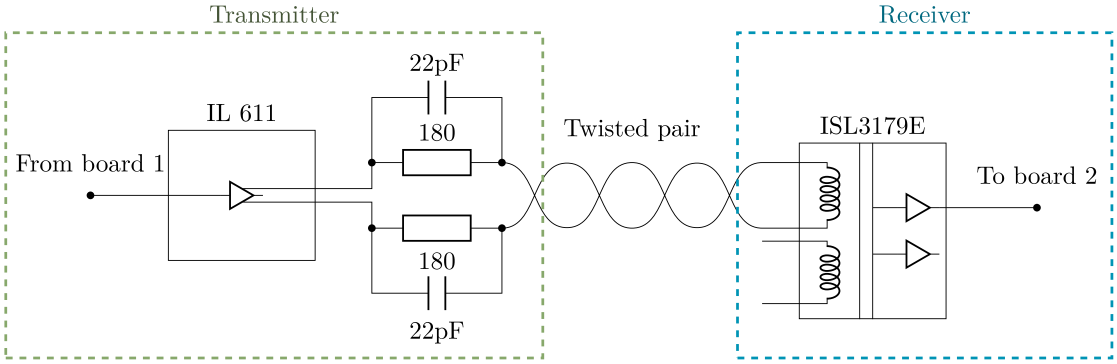


Figure 44: Differential encoder interface, adapted from [Lindahl, 2021].

3D models of the PCB exported from Mentor Expedition are shown in figures 45 and 46. The top view includes the sensor PCB bonded to the PCB.

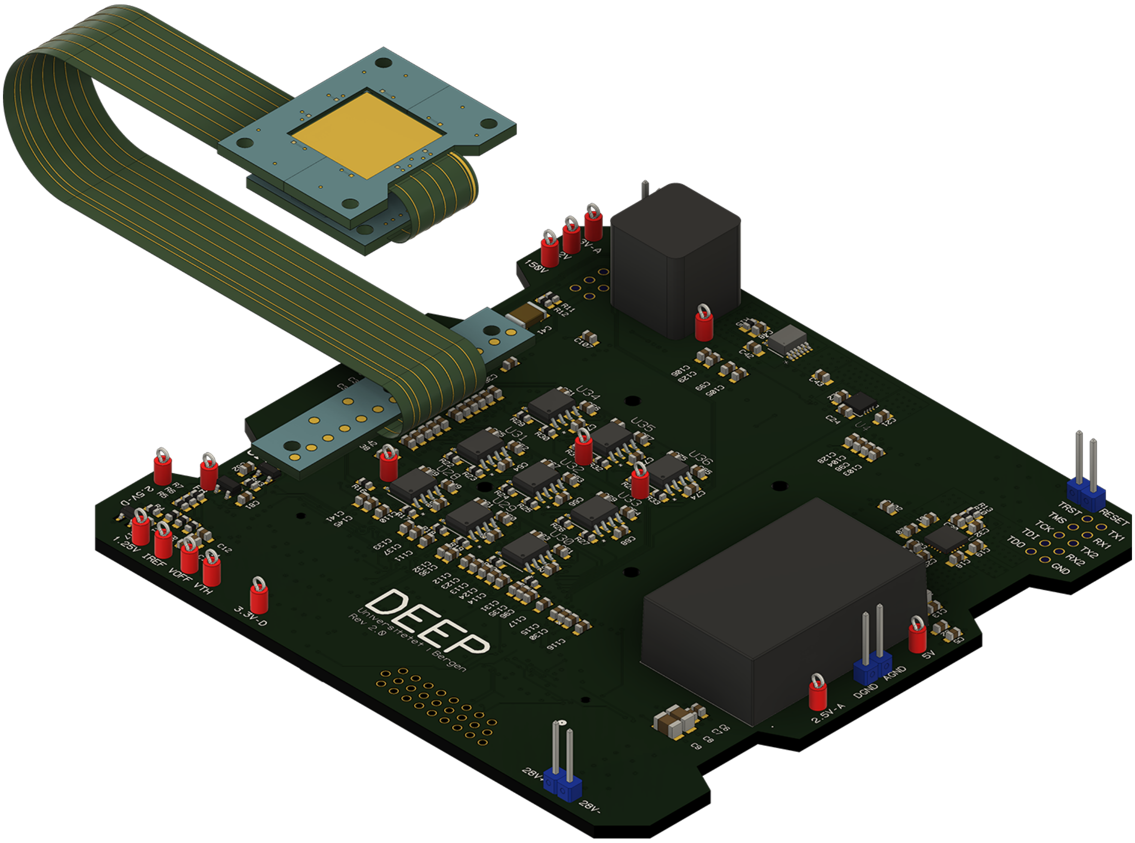


Figure 45: DEEP PCB with ASIC PCB bonded.

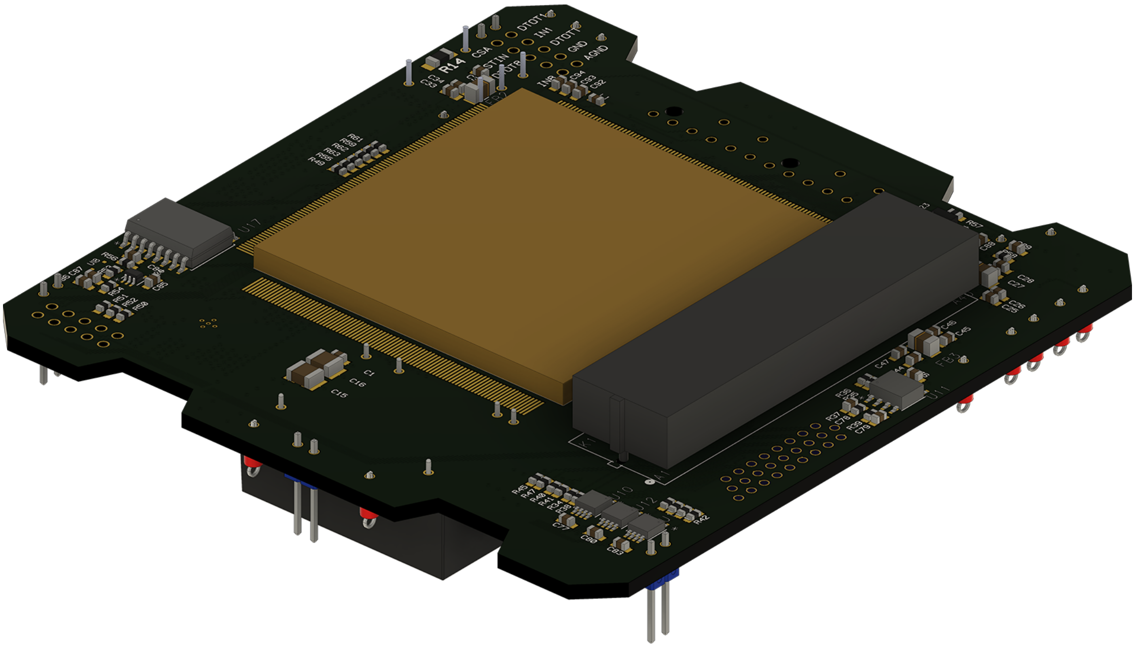


Figure 46: DEEP PCB bottom side.

7 HARDWARE TESTING

The purpose of this section is to test for the correct functionality of important circuit components. This is by no means an extensive test, but is meant to verify basic circuit functionality.

7.1 Detector bias

As discussed in section 6.5.1, a change of output voltage on the bias supply of the detector induces a charge that is proportional to the voltage change. This means that ripple and noise on the bias supply directly affects the SNR of the detector, and that there is a maximum allowable ripple on this bias to rule out noise hits on the lowest energy channel of 30 keV. Therefore, it is important to determine the ripple of the bias supply. As the ripple is expected to be a small percentage of the DC bias, it is measured with an oscilloscope and AC coupled probe. A special measurement method is needed as the conventional method of connecting a probe directly on the output terminal and ground somewhere on the circuit will create a loop antenna effect and collect noise from the environment. By not recognizing this, the ripple performance of the DC/DC converter will be underestimated [Ric, 2016]. To reduce this effect, the ground loop formed between the probe and ground wire must be minimized. The ripple and noise has been measured using the tip and barrel method as demonstrated in 47, which minimizes the area covered inside the loop. Here, the measured output is the voltage across a decoupling capacitor on the output.

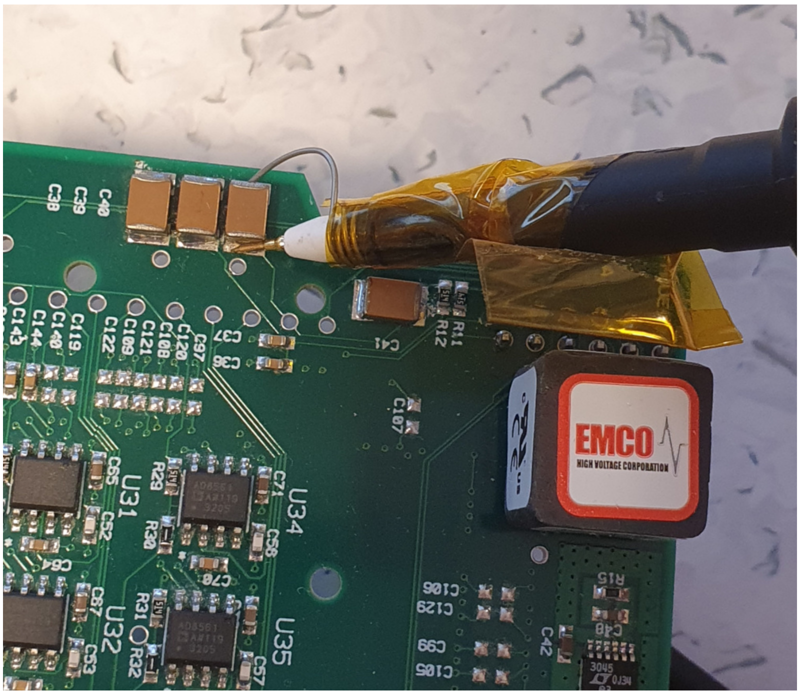


Figure 47: Tip and barrel measurement of output ripple voltage.

The ripple of the bias is illustrated in Figure 48.

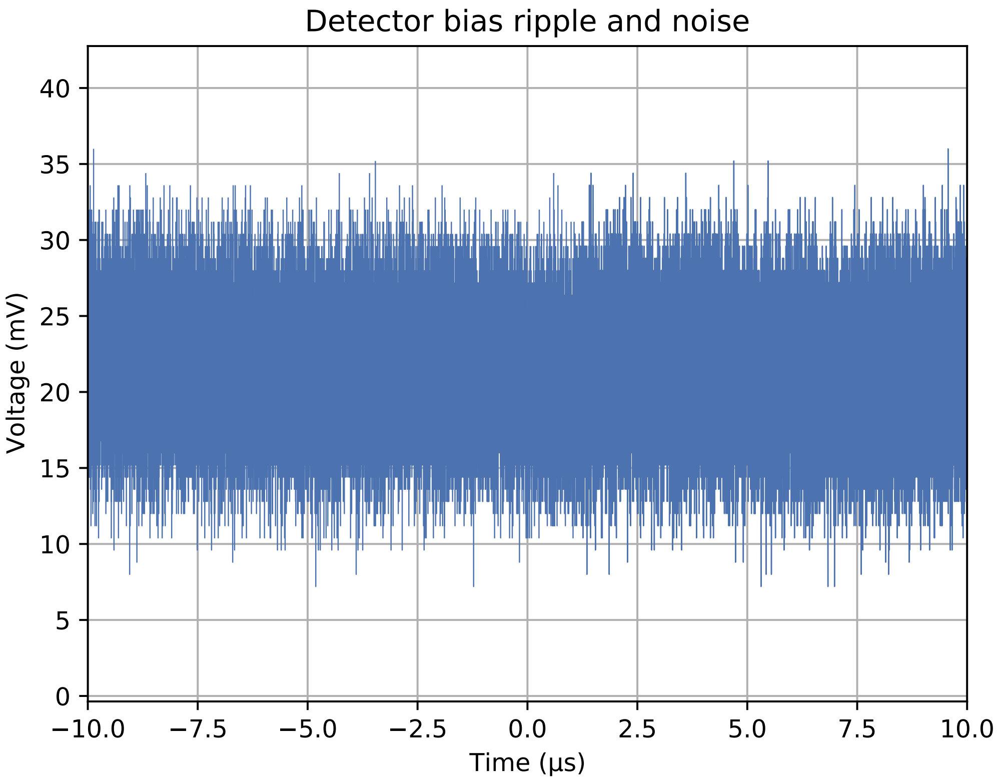


Figure 48: Noise and ripple of bias supply.

As per equation (6.2), the allowable ripple is defined by the minimum charge that is to be detected and the capacitance of the diode. The capacitance was measured by SINTEF during production and is dependent on the bias voltage, shown in Figure 49. A 30 keV particle deposits 1.35 fC of charge, which means that together with the capacitance of roughly 5.2 pF, we require that the ripple and noise amplitude is lower than 25.9 mV. As shown in 48, this requirement is met.

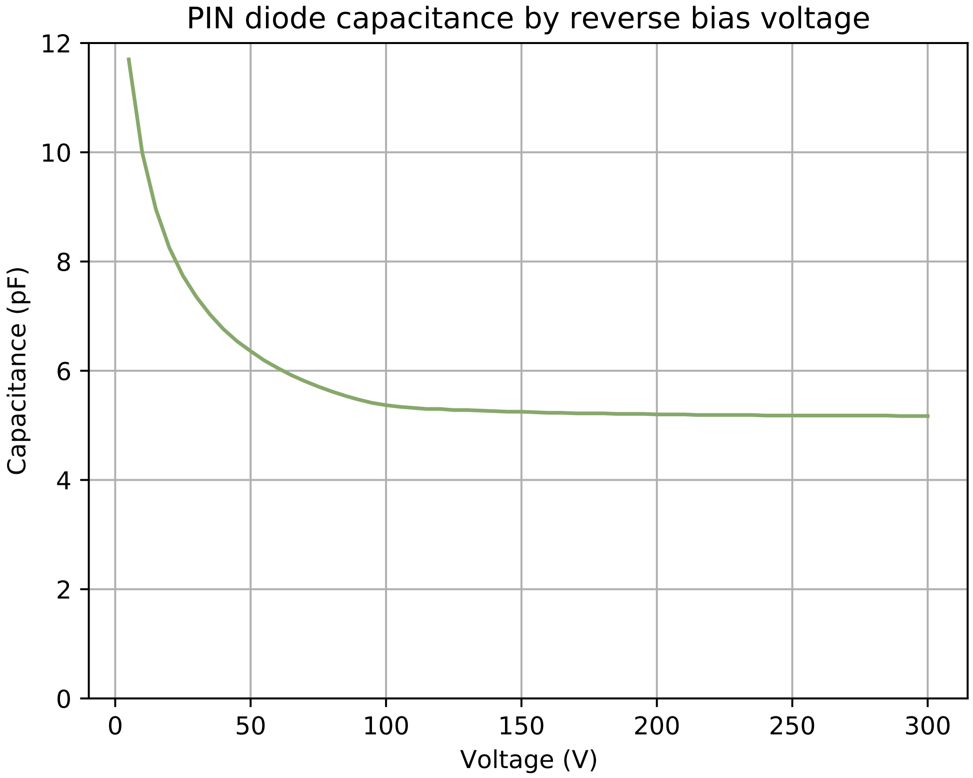


Figure 49: Capacitance of the diode as a function of reverse bias voltage.

Another thing of interest is the rise time from when the bias supply is enabled until it reaches its operating voltage. After enabling the bias, the detector should be disabled until the bias supply is stable to avoid inaccurate readings. The measurement in Figure 50 shows that the rise time is roughly 6 seconds. The relatively slow rise-time is expected because the bias supply boosts the input voltage from 5 V to 150 V. Note that in this configuration there was no load attached, and the output was left floating, resulting in a slightly higher output voltage of 210 V. The rise time may change with a load attached, but in general the rise time should be overestimated to allow the bias voltage to stabilize.

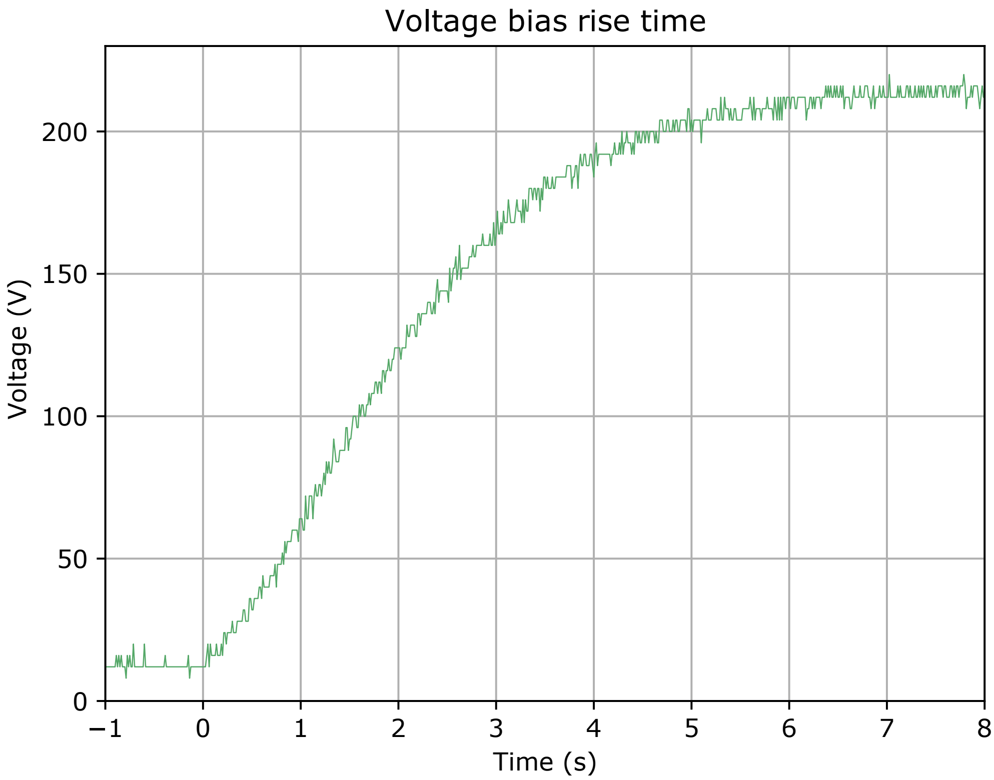


Figure 50: Rise time of detector voltage.

7.2 Comparator circuit

The comparator circuit has been tested and is working as expected; the output correctly switches to an applied stimulus on the input, and the FPGA on the evaluation kit is able to read the output from the comparator through the socket. The response of the inverting output to a stimulus can be shown in Figure 51, showing a propagation delay of around 7 ns. It is worth noting here that for correct operation of the comparator output, the FPGA must be programmed and both of the pins on the FPGA must be configured as inputs. If the FPGA is unprogrammed or the pins are not specified as inputs, the FPGA will drive these lines high, and this means that the comparator will have to overpower the FPGA to switch correctly, leading to longer propagation delays.

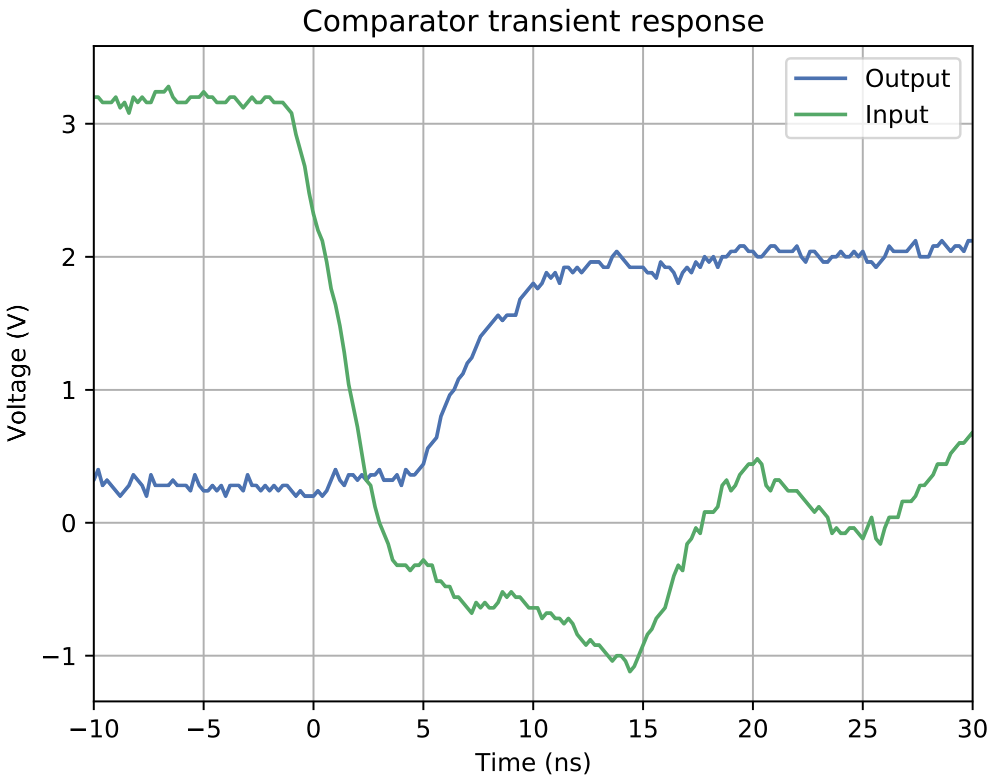


Figure 51: Comparator transient response to input stimulus.

The circuit has two modes of operation in terms of the threshold: the ToT mode with a static threshold, and the dToT mode where the threshold can be switched to increase with the transient response of an RC network with the same time constant as the shaping time of the pulse from IDE1180 on the sensor PCB. Both of these modes have been tested to be working, although the time constant of the RC pulse is slightly off. Figure 52 shows the measured pulse superimposed on RC pulses with the time constants of 40 ns and 55 ns. As seen, the actual pulse tracks closely with the time constant of 55 ns. The increased time constant can likely be attributed to stray capacitance, and this can be rectified by reducing either the capacitance or resistance of the circuit.

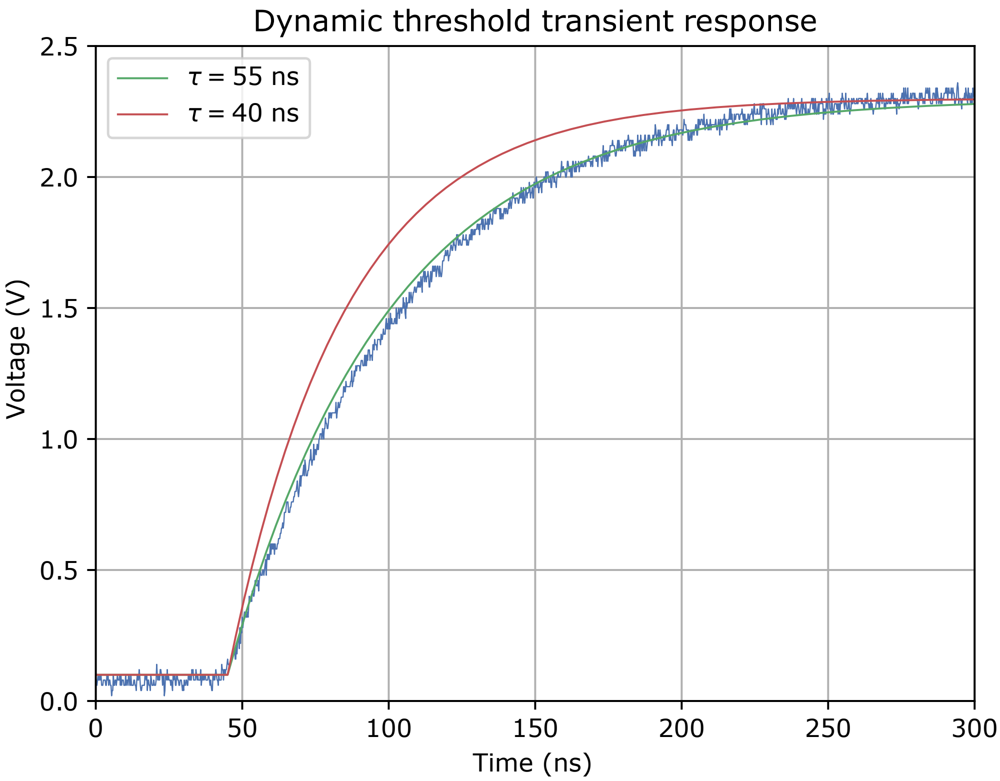


Figure 52: Dynamic threshold transient response.

8 FIRMWARE IMPLEMENTATION

This section contains the complete firmware design of the detector, developed in VHDL. Before the details of the implementation, a quick summary of the NG-MEDIUM architecture is included.

8.1 NG-MEDIUM architecture

The NG-MEDIUM architecture can be divided into three blocks: the user IO ring, including four clock generators, the FPGA core, and the FPGA configuration logic and interface [Nan, 2020b]. An overview is shown in Figure 53.

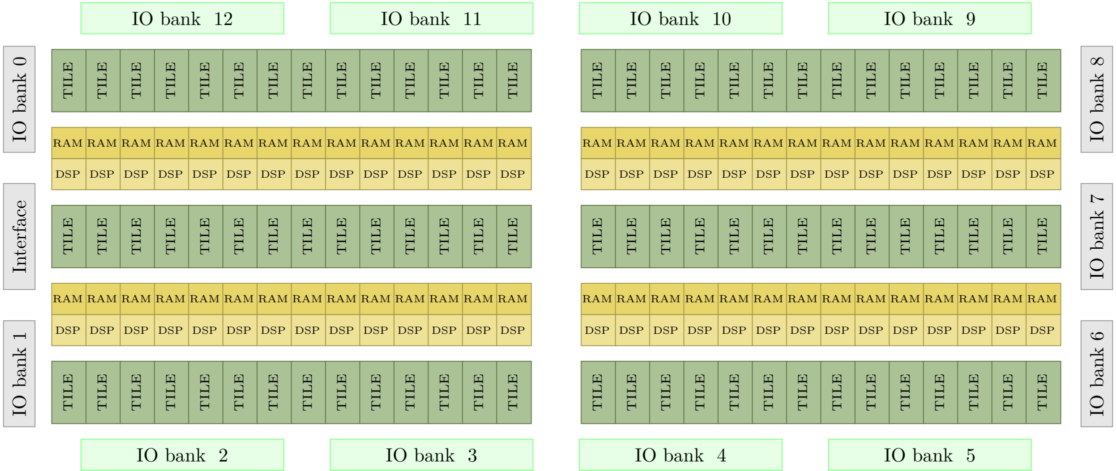


Figure 53: NG-MEDIUM architecture, adapted from [Nan, 2020b, p. 72].

8.1.1 IO ring

The IO ring consists of thirteen IO banks in total, divided into eight complex IO banks and five simple IO banks. In Figure 53, the complex IO banks can be seen as the ones running along the top and bottom of the die (green), and the simple banks are located on the left and right of the die (gray). Since the CQFP-352 package has a reduced pin count, only eight of these are available. These are listed in Table 5.

Table 5: QCFP-352 IO banks [Nan, 2020b, p. 26].

IO bank	Type	Number of IOs	Location
0	Simple	14	Left
1	Simple	12	Left
2	Complex	30	Bottom
5	Complex	30	Bottom
6	Simple	22	Right
8	Simple	24	Right
9	Complex	30	Top
12	Complex	30	Top

Each IO bank requires its own IO supply, and can operate on 1.8 V, 2.5 V or 3.3 V. Simple and complex banks have the same core functionality, but the complex banks have a few extra features. An overview of the difference between simple and complex banks is shown in Table 6.

Table 6: IO bank feature summary [Nan, 2020b, p. 28].

Feature	Complex	Simple
Number of IOs	30	30/22
Operating voltages	1.8V, 2.5V, 3.3V	1.8V, 2.5V, 3.3V
IO standards	LVCMOS, SSTL, HSTL, LVDS	LVCMOS, LVDS (2.5V only)
Single DFF	Yes	Yes
Input termination	Yes	No
Programmable delay	Yes	Yes
Clock domain changer	Yes	No
Shift register	Yes	No
DDR mode	Yes	No
SpaceWire	Yes	No

8.1.2 Core logic

The core logic is constitued by five rows of logic resources, of which three rows consist of tiles, and two rows consist of Coarse Grain Blocks (CGBs), containing Block RAM (BRAM) and Digital Signal Processor (DSP) blocks. Each tile row is composed of 28 tiles, yielding a total of 84 tiles. Each tile consists of four stripes containing 96 Functional Elements (FEs) each for a total of 384 per tile. The functional elements are the core logic blocks of the NanoXplore FPGA fabric analogous to Xilinx's Configurable Logic Block and Intel's Adaptive Logic Module. Each functional element contains a 4-input Lookup-table and a D-type flip-flop as illustrated in Figure 54. Die-wise this amounts to a total of 32,256 FEs. In stripes 1 and 2 of each tile the input of each Lookup Table (LUT) is connected to the carry logic in the cell, while stripe 3 includes 24 extra 4-input extended LUT units known as X-LUTs. Each of the four X-LUT inputs are connected to the output of four LUTs, constituting a 16-input LUT. Stripe 4 contains 64 16-bit dual-port BRAM blocks and Clock Generation Switches (CKSs).

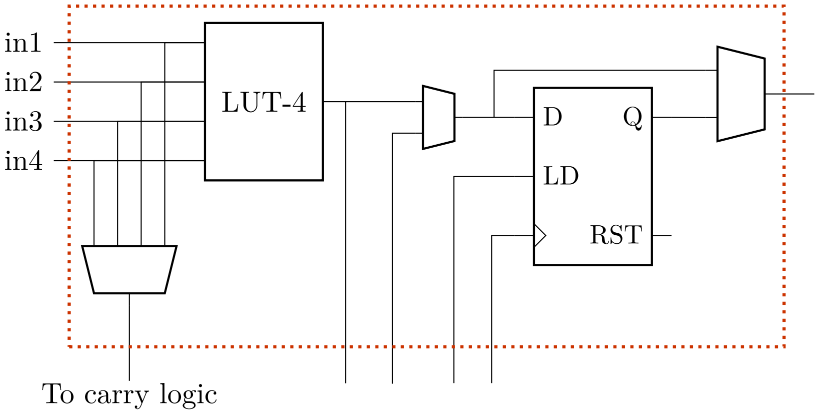


Figure 54: Functional element, adapted from [Nan, 2020b, p. 76].

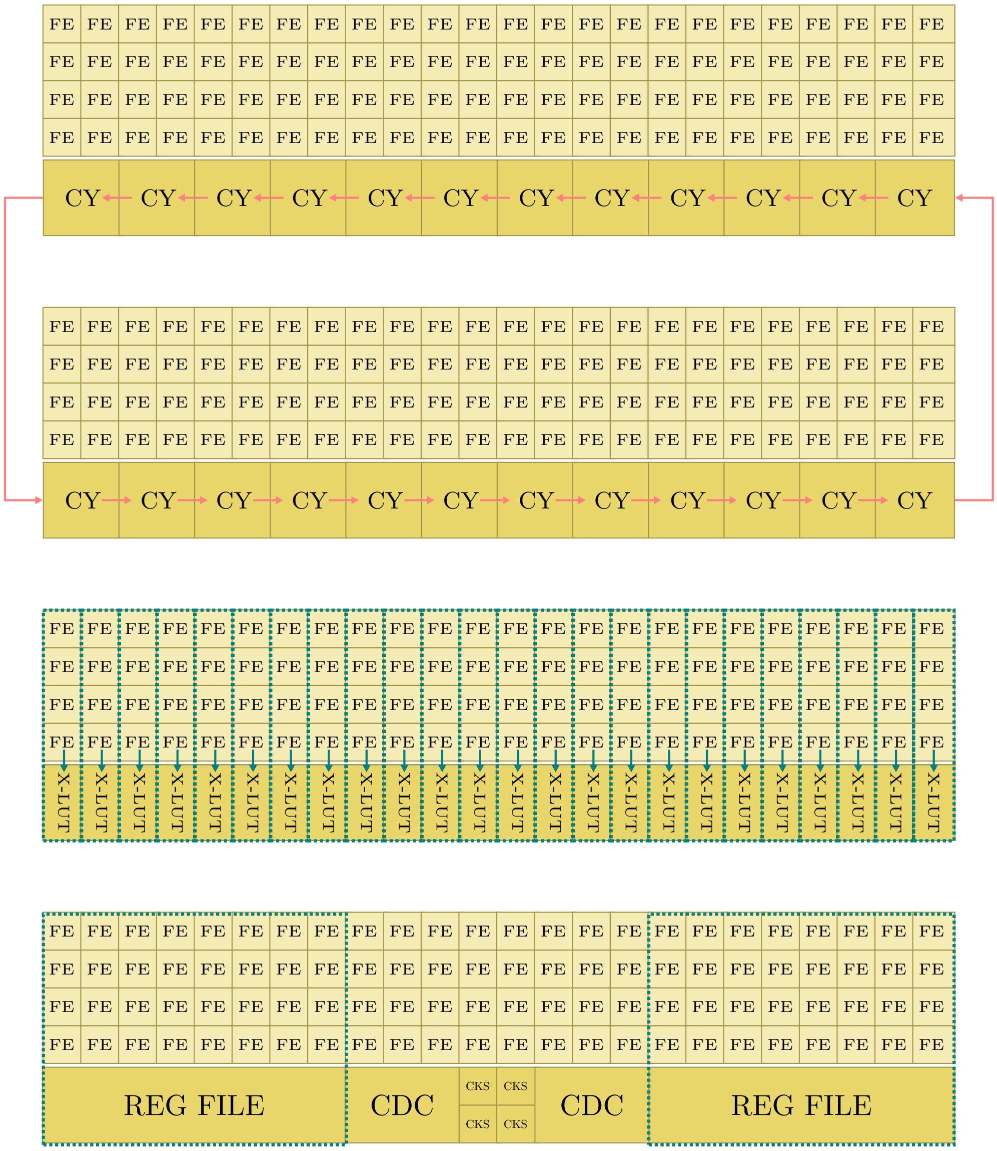


Figure 55: Tile architecture, adapted from [Nan, 2020b, p. 75].

Each carry logic unit is directly connected to four of the eight FEs that reside directly over it as shown in Figure 56, while the remaining four FEs are free to implement other logic. The carry logic unit is a carry lookahead circuit that is either inferred from arithmetic operators during synthesis, or can be instantiated as a component NX_CY from the NXLibrary. This component includes four output ports that constitute the sum output of the adder, as well as carry interconnects between NX_CY blocks. As shown in Figure 55, these can be chained to create up to a 96-bit carry chain in each tile.

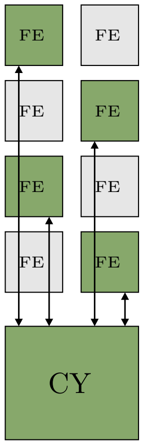


Figure 56: Carry logic interconnect, adapted from [Nan, 2020b, p. 78].

The carry logic in the FPGA fabric is mainly of interest for this project because it offers a resource that can be utilized to create short propagation delays with known and consistent routing delays for implementing a reliable multitapped delay-line.

8.2 RSE protocol

The microelectronics group at UiB has developed a communication protocol for use in the Solar wind Magnetosphere Ionosphere Link Explorer (SMILE) project. This is a register-based wrapper protocol expanding upon on the UART physical protocol. The Radiation Shutter Electronics (RSE) protocol is used in this configuration for debugging operation, but will not be present at time of flight. The protocol features a 7-bit addressing scheme, which in conjunction with the R/W bit constitutes the first byte of any transfer.

A read operation is initiated by transmitting the address of the register with the R/W bit set to zero and is illustrated in Figure 57. After the FPGA receives the address it will return three bytes. The first byte contains the address of the register that was accessed, the second byte contains the contents of that register, and the third byte is the sequence number of the communication. The sequence number is incremented for each read or write operation and is used to detect the loss of transmission data.

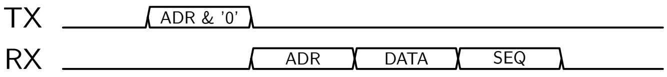


Figure 57: RSE protocol read operation.

Similar to the read operation, the write operation starts with sending the address, but with the R/W bit is set to one as shown in Figure 58. Following this, the data that is to be written to the address must also be sent. When the FPGA has received these two bytes, it returns the address, the data, and finally the sequence number.

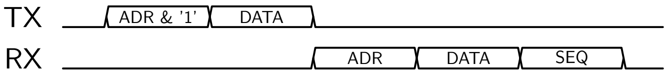


Figure 58: RSE protocol write operation.

8.3 Clock generation

The main system clock is generated by the 40 MHz Si510 oscillator which is connected to the internal clock distribution network through one of the clock inputs on IO bank 8. In addition to this, the counter in the data acquisition module requires a sampling clock of as high frequency as practical. The NG-MEDIUM FPGA contains four CKSs, one in each corner of the die. [Nan, 2020d, p. 8] Each CKG is composed of one PLL and eight Waveform Generators (WFGs). These are included as components in the NXLibrary package. The PLL requires a reference clock input, and supports an optional external feedback input. It can supply up to three clock outputs with different division factors from the Voltage Controlled Oscillator (VCO) output. When the PLL has locked onto a phase and frequency, the RDY signal is asserted. The highest VCO frequency that the PLL can reliably generate is 1.2 GHz.

The WFGs support two modes of operation specified by the *mode* generic input. Setting this to normal mode results in the WFG acting as a clock buffer. If the *mode* generic is set to 1, the WFG enters pattern sampling mode, in which the output waveform can be specified by the use of the pattern generic. In both modes, optional delays can be configured, in addition to clock inversion through the *wfg_edge* generic.

The WFGs have programmable delay lines with 64 taps of 159 ps each. These are set with the *delay* generic and enabled by the *delay_on* generic at instantiation. If the *delay_on* generic is set to 1, a *delay* value of 0 corresponds to a delay tap value of 1.

8.4 Top level entity

The top level entity acts a framework that connects all the modules together. An overview of the top-level entity can be seen in Figure 59. Each of the modules in the figure are explained in the following sections.

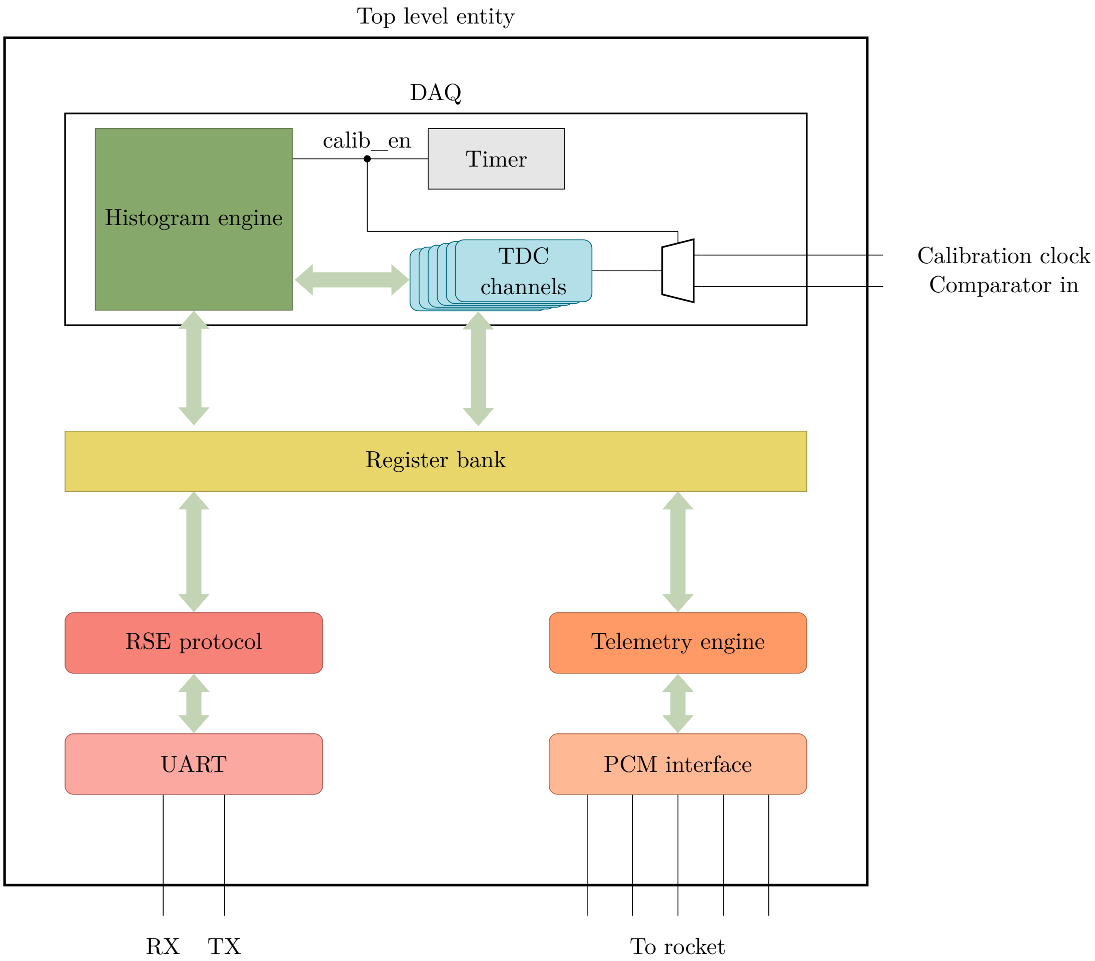


Figure 59: FPGA firmware overview. The clock generation circuit has been left out of the drawing.

8.5 Time-to-digital converter

In this thesis, three TDC topologies are suggested and implemented. Each of them carry their own strengths and weaknesses, and the topology most suitable for the particle detector has to be determined by testing and analyzing performance metrics. An additional concern beyond the scope of the sounding rocket launch is the scalability when more than eight pixels are included in the design. The VHDL code is written with scalability in mind, but the modules are still limited by the available hard resources on the FPGA.

The three topologies treated in this section are counters with no interpolation (Type I), counter with delay line interpolation (Type II) and counter with asynchronous oversampling (Type III). The most accurate TDC is the Type II TDC, but it is challenging to implement the asynchronous nature of it correctly. It may also pose issues for scalability as more channels are added. Conversely, the Type I counter is easy to implement and scales well, but the resolution may be too limited for accurate particle detection. Type III lies somewhere in the middle of these two, improving the conversion characteristic without demanding too many resources. It is also slightly easier to implement than the delay line TDC owing to the efficient clock distribution network of the FPGA.

8.5.1 Type I: Interpolationless counter

This TDC consists solely of a synchronous counter. It has no method of interpolating the clock period, and the resolution is thus limited by the clock period. It may seem redundant to include a TDC with no interpolation, as both of the other TDC topologies include a counter providing the same coarse resolution, and should result in strictly superior TDCs. However, one particular strength of the interpolationless counter is the simplicity, resulting in a very low resource and low power TDC that should scale with little effort. In addition, there is no need for calibration as this is a purely synchronous design, further reducing the circuit complexity. This fact alone means that the counter cannot be altogether ruled out, as the relative performance of the circuit may not be as bad as it seems at first glance. The counter also serves as a baseline case, which will be useful in answering whether clock interpolation needs to be included.

8.5.2 Type II: Counter with delay line interpolation

The second type of proposed TDCs is of the type discussed in section 5.6

There are many ways to create a delay element in an FPGA. The lookup-tables in the functional elements have a gate delay and can be chained together to create a delay line, but routing between the FEs must be done manually to ensure that routing delays are uniform, and even then it is challenging to exactly match the delays. On top of this, NanoXplore's place and route tool is primitive and does not yet support entirely manual place and route, only allowing the designer to constrain modules to specific tiles. An alternative approach is to utilize hard resources on the FPGA to generate more symmetrical delays. On the NG-MEDIUM, available hard delay resources are the carry logic units and the DSP cores. Since this is discrete circuitry embedded in the die and not part of the general logic, the circuits are closely matched and not prone to random fluctuations. The adder carry chain is particularly attractive as there is no way to physically directly route the carry out of one adder than to the carry in of the next, and as a consequence, the order of implementation is constrained to the physical chain of adder blocks. There is no way to directly route the carry chain across tiles physically, as per Figure 55.

Generating a delay with carry logic is relatively simple. The carry logic on the NG-MEDIUM is discussed in section 8.1.2. The port map of the NX_CY component is shown in Figure 60.

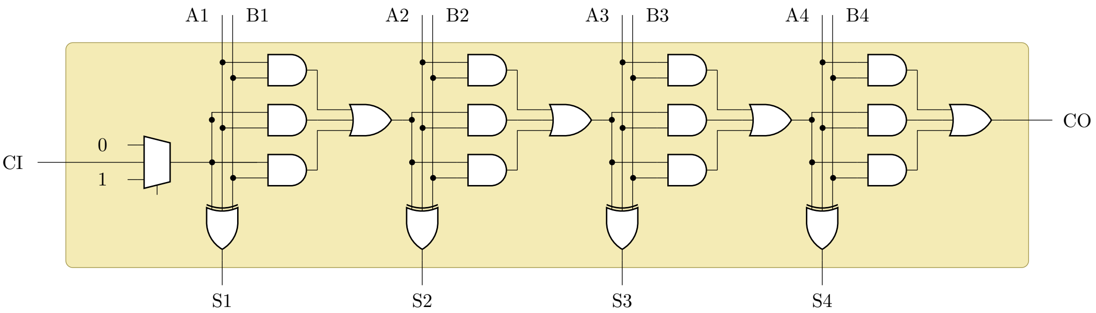


Figure 60: NX_CY port map, adapted from [Nan, 2020d, p. 28].

In adder terminology, a full adder either generates a carry, propagates it, or does not propagate it (also known as a kill). A carry propagates if,

$$P_i = A_i \oplus B_i \tag{8.1}$$

which means that the carry out is equal to the carry in. This is known as a ripple carry because the carry propagates through chained full-adder blocks in sequential fashion, and each full adder introduces a delay. This delay can be used to implement a delay line. The sum of a full adder that propagates is not ready until the carry in is ready,

$$S_i = C_{\text{in}_i} \oplus P_i \tag{8.2}$$

Figure 60 shows a simple 4-bit carry ripple adder where the sum S_i is the XOR of A, B and CI. The carry out is the majority of A, B and CI, here directly implemented as sum of products,

$$C_{\text{out}_i} = AB + AC_{\text{in}_i} + BC_{\text{in}_i} \tag{8.3}$$

This figure is meant to be illustrative of an adder and does not correspond to the internals of the actual adder circuit, which is a carry lookahead [Nan, 2020d, p. 28]. This is verified by the Standard Delay Format (SDF) file produced by the synthesis tool as listed in Listing 1.

```
(CELL
  (CELLTYPE "NX_CY")
  (INSTANCE i_carry_chain_i_gen_5_i_nx_cy)
    (DELAY (ABSOLUTE
      (IOPATH B1 S1 (0.770::0.783) (0.770::0.783))
      (IOPATH B1 CO (0.527::0.539) (0.527::0.539))
      (IOPATH B2 CO (0.546::0.557) (0.546::0.557))
      (IOPATH B3 CO (0.435::0.438) (0.435::0.438))
      (IOPATH B4 CO (0.395::0.401) (0.395::0.401))
      (IOPATH CI S1 (0.748::0.761) (0.748::0.761))
      (IOPATH CI CO (0.280::0.284) (0.280::0.284))
    ) )
)
```

Listing 1: SDF content of a NX_CY element

The propagation delays shown here indicate that the adder circuit has matched delays from the carry in to the sum outputs, which means that these will change value at roughly the same time and can therefore not be used as individual taps in the delay line. Instead, only one of the sum outputss is used as a tap, and each NX_CY module will only supply one delay. Since the Carry Out (CO) of the last NX_CY block is ready roughly 280-284 ps after the Carry In (CI), the sums of each consecutive NX_CY module will change logic level separated by this delay. The delay from CI to sums S1-S4 will contribute to the first tap delay in each tile, so the first tap delay is systematically imbalanced. This comes in addition to the interconnect delay imposed by crossing tiles.

By configuring a chain of full adders to propagate and connecting the carry in to the output of the CSA comparators, the sums will sequentially change value, each with a delay τ_c corresponding to the carry propagation delay. This is inherently an inverting operation, as P_i is always true in this configuration. The NG-MEDIUM does not have a direct connection available between carry in, carry out and the functional elements that constitute general FPGA fabric; only the inputs A and B and the sum output S is exposed to general logic. For this reason it is not possible to directly connect the comparator outputs to the carry in of the delay line. To get around this architectural problem, the first carry out is generated by connecting the comparator output to A4 and B4 of the first NX_CY unit in the chain. When the comparator output is 0, this creates a kill K which means that the carry out of the first block will also be 0 independent of carry in. Conversely, when the comparator output is 1, this bit position is a generate G which gives a carry out of 1.

By adding 0000_b with 1111_b, it is ensured that every bit propagates the carry in to the carry out.

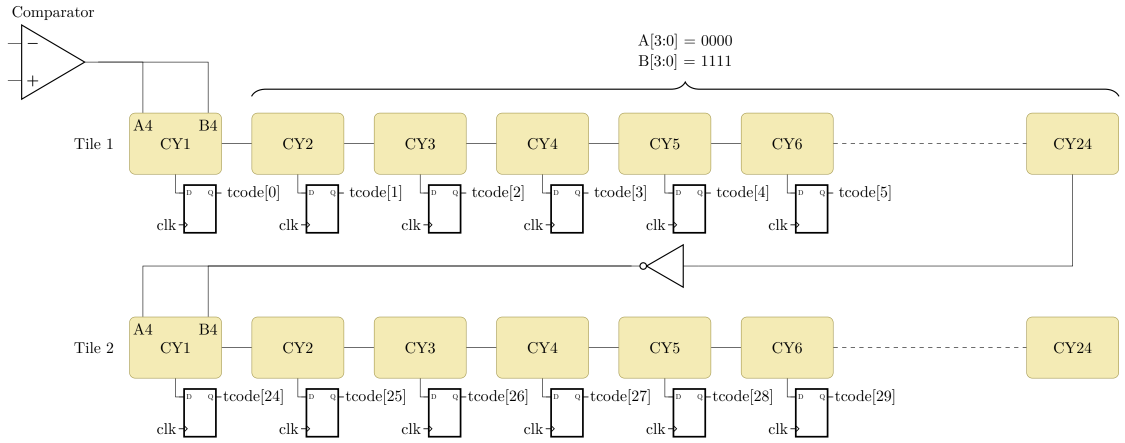
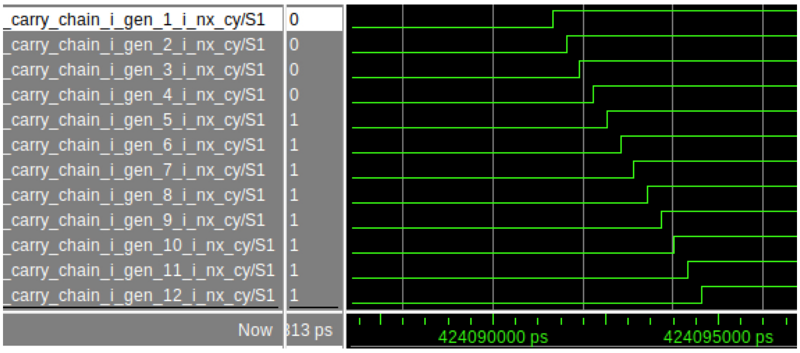


Figure 61: Multitile delay line implemented with NX_CY units.

Since each tile is limited to a maximum of 24 carry units by hardware, longer chains need to be bridged together by the tile interconnect, and fed through a NX_CY unit to enter the next carry chain. Attempts to route this directly will result in a place and route failure. The delay line can still be bridged by using the adder inputs and sum outputs to use the last block in the chain as a carry arbitrator. An example of how this can be achieved is shown in Figure 61, and example VHDL code of how this is accomplished with nested generate statements is shown in appendix C.1. Since the sum output is inverting, the sum output from the carry arbitrator is inverted before being fed into the carry generator in the next tile. This method presents a way to extend the delay chain into any number of tiles. It is not recommended unless necessary, as the bridge interconnect will add an ultrawide bin and be a detriment to TDC performance.

As the carry in needs to be routed through a carry generator block, the carry generator block sum output cannot be used. This is because the delay line must be used on both rising and falling edges, and since a carry in cannot be routed, the correct sum output is not dynamically alterable. If the TDC input is high and the carry in is one, the sum output of the carry generator block is also one. However, if the TDC input is low and the carry in remains one, the sum output is still statically one. The same is true for the rising edge case if the carry in is zero. Effectively, this means that for each tile only 23 carry adder blocks can be used, as the first block is required to route the signal into the carry chain and has a static sum output. A back-annotated example of the sum outputs of the delay line propagating is illustrated in Figure 62.



```
_carry_chain_i_gen_1_i_nx_cy/S1   0
_carry_chain_i_gen_2_i_nx_cy/S1   0
_carry_chain_i_gen_3_i_nx_cy/S1   0
_carry_chain_i_gen_4_i_nx_cy/S1   0
_carry_chain_i_gen_5_i_nx_cy/S1   1
_carry_chain_i_gen_6_i_nx_cy/S1   1
_carry_chain_i_gen_7_i_nx_cy/S1   1
_carry_chain_i_gen_8_i_nx_cy/S1   1
_carry_chain_i_gen_9_i_nx_cy/S1   1
_carry_chain_i_gen_10_i_nx_cy/S1  1
_carry_chain_i_gen_11_i_nx_cy/S1  1
_carry_chain_i_gen_12_i_nx_cy/S1  1
                             Now  313 ps
```

Figure 62: Back-annotated simulation of the sum outputs of the NX_CY delay-line. Each delay is roughly 300 ps.

One particular detail that should not be overlooked is that the difference in propagation delay between the input to the edge detector and the input to the delay line should be minimal. If the propagation delay of the former is much lower than the latter, the difference in delays will add to the first bin. This was in fact observed with back-annotated simulation (shown in Figure 64a), where the first bin was roughly 10 times larger than the remaining bins. The cause was discovered to be a sampling register in the IO buffer that gets implemented when the input is sampled, such as by an edge detector. This causes the output of the delay line to be sampled early. In order to rectify this, a somewhat peculiar routing scheme is required. Instead of detecting an edge in the TDC input signal, the edge detector monitors the first sum output of the chain. With this routing arrangement, it is guaranteed that the TDC input is not routed directly to a nearby edge detector, causing a premature sampling of the delay line as the input propagates to the carry unit blocks. Since the edge detector will not fire before the signal has propagated through the first delay element, the first bin will have a width of 0 and be excluded from the chain.

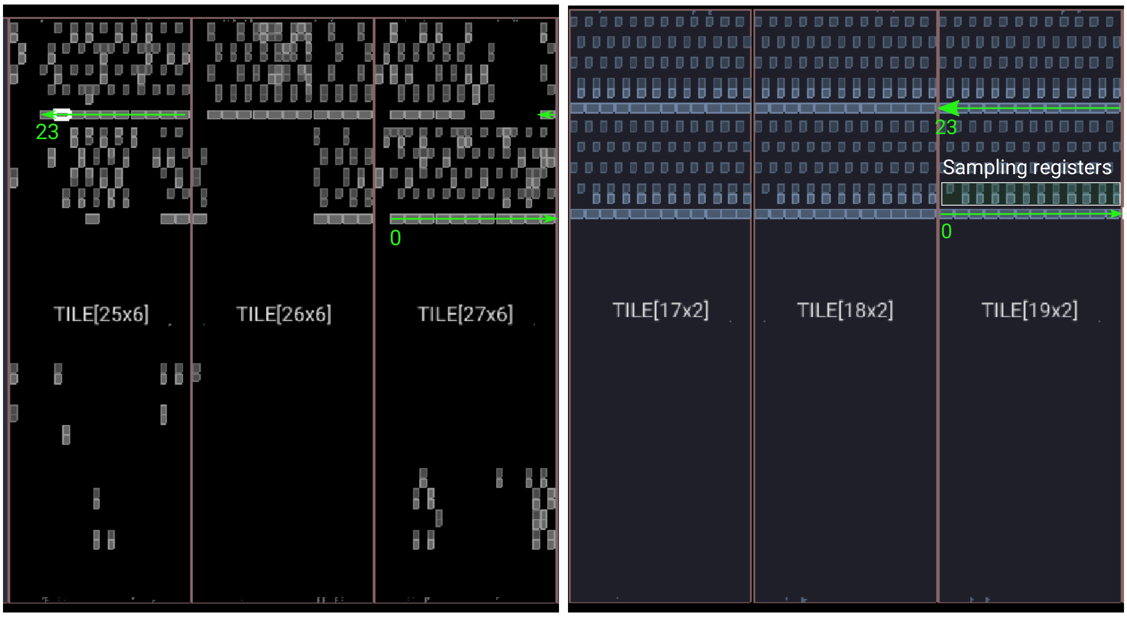


Figure 63: Left: Fractional implementation of a delay line, causing an ultrawide bin in the interconnect between tile 27 and tile 25. Right: Perfect implementation of the delay line.

The performance of the delay line is very dependent on the physical placement of the circuit. Left to its own devices, the placement tool has a tendency to place unrelated CY units in the same tile as the delay line, causing a fractional implementation of the delay line that stretches over several tiles. This can be shown in figure 63 (left). For a proper implementation, the chain should be continuous within a tile, and only the sampling registers should be allowed within a delay line tile.

There are placement constraints that can be passed to the placement tool to ensure that the delay lines get their own exclusive areas. First, a module must be defined for each delay line. As there are several delay lines, these are mapped to their own respective module by the % operator. Then, a region is defined, and in this case it is desired that the TDCs are as close to the input IO bank as possible. Finally, the module is constrained to the region. This process is outlined in Listing 2. In the example, the TDCs are anchored to the tile in the top right corner and ordered in descending columns.

```
# Must be defined before synthesis
proj.addModule('.*delay_line.*', '.*delay_line.*', 'delayline-%')

# Synthesize circuit
proj.synthesize()

for channel in range(num_channels):
    #Function parameters: Region name, col1, row1, col2, row2, exclusive
    proj.createRegion(f'DELAY_REGION{channel}', 28-channel, 2, 28-channel,
   2, True)

    proj.confineModule(f'delayline-{channel}', f'DELAY_REGION{channel}')
```

Listing 2: Adding placement constraints to NanoXmap.

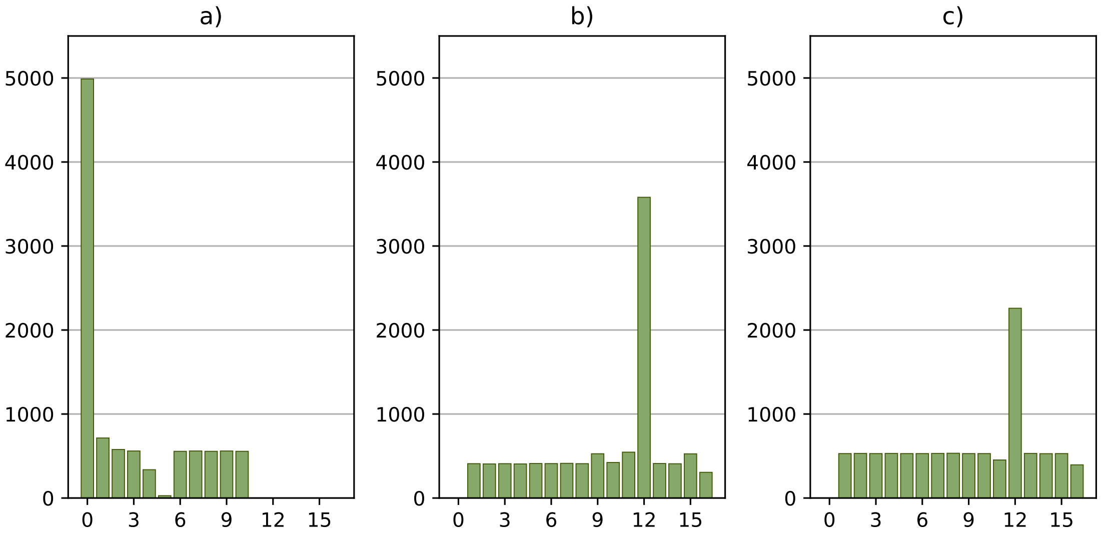


Figure 64: Delay line bins from a uniform distribution of 10 000 hits: a) With edge detector directly from TDC input b) With edge detector connected to first tap c) Same as b but with placement constraints. Bins are shown on the x-axis.

Figure 64b shows how the tap delays look like with the fractional implementation shown in 63 (Left). When the appropriate placement constraints are added, the 12th bin is reduced as shown in Figure 64c.

This is the best possible implementation in this technology, and there is no way to further reduce the 12th bin as it is inherent to the FPGA architecture and stems from the interconnect between the upper and lower rows of NX_CY blocks. By running the TDC with a sampling clock speed sufficiently high enough so that the clock period is shorter than the total propagation delay of 12 NX_CY elements, this interconnect is not needed. With this implementation two delay lines can be implemented in a single tile, and the bin widths of the delay line are very uniform. With a propagation delay of 300 ps, this means running the TDC at speeds exceeding 277 MHz.

8.5.3 Type III: Counter with oversampling interpolation

This implementation of a TDC is based on the principle discussed in section 5.5. The oversampling TDC demonstrated in that section is a 4-clock implementation. There are 32 WFGs in the NG-MEDIUM, and each of them support delay taps from 0 to 64. Since there is no DLL resource in the NG-MEDIUM, these lines constitute the only option to generate off-phase clocks. The delay spacing period d is 159 ps, meaning that to create n equally off-phase clocks, the clock period T should ideally be a multiple m of this,

$$T = mnd \tag{8.4}$$

At first glance, it may seem that this TDC can be implemented with up to 32 (as many as WFGs available on the die) equally off-phase clocks, but in practice this is limited by the clock distribution network. Since all the TDCs should ideally be placed in the upper right corner where the IO bank of the input is, there is a further restriction on available WFGs as each corner only contains eight WFGs. Another limiting factor with this method is the large power consumption from powering many clocks. The dynamic switching power of a clock is,

$$P_{\text{sw}} = CV^2 f \tag{8.5}$$

The PLL clocks use the 1.2 V main core power rail, which is limited to 1.2 W. A power calculation was done using a spreadsheet supplied by NanoXplore, and it shows that of the available power on the PCB, and five clocks of frequency 500 MHz contributes a power dissipation of roughly 0.5 W. This means that five clocks is possible within the power budget of the PCB.

Since each of the channels that implement this TDC would use the same clocks, the limiting resource as the number of channels scale in this topology is the number of D flip-flops available. The limiting resource in terms of accuracy is the number of clocks that can reliably and practically be implemented.

[Balla et al., 2014] only considers 4x oversampling with four clocks, but this method can be extended to an arbitrary number of clocks. To generalize the method, first a pattern should be established. It may be useful to define the clock phases that sample each register as a matrix C. The matrix of Figure 33 is a 4x4 matrix of registers, and the clock phase of each register can be represented as,

$$C = \frac{2\pi}{4} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \tag{8.6}$$

In component form, the matrix is defined as,

$$c_{i,j} = \begin{cases} 0 & i < j \\ i - j & i \geq j \end{cases} \tag{8.7}$$

A proposed general implementation in VHDL is listed in appendix C.2.

This TDC implementation is entirely inferred and will work on any FPGA. Only the clock array needs to be populated manually. This TDC is optimally implemented with delay locked loops to supply the off-phase clock signals, but calibration can adjust the bin widths in the same fashion as for the delay line.

One WFG is required per clock division. These must be synchronized, and the input to each WFG should come from a PLL [Nan, 2020b, p. 48]. The WFGs should only operate once the PLL has achieved a phase-lock, at which point the PLL component will assert the RDY signal, which is connected as inputs into each WFG. To synchronize the WFGs to each other, one of them is selected as master. Each WFG has a synchronization out (SO) and synchronization in (SI) signal, and the master WFG should feed the SO back into its own SI input. Each remaining WFG leaves the SO output open and connects the SI to the master SO.

The clock generation circuit is shown in Figure 65. The clock lines are shown in red and blue.

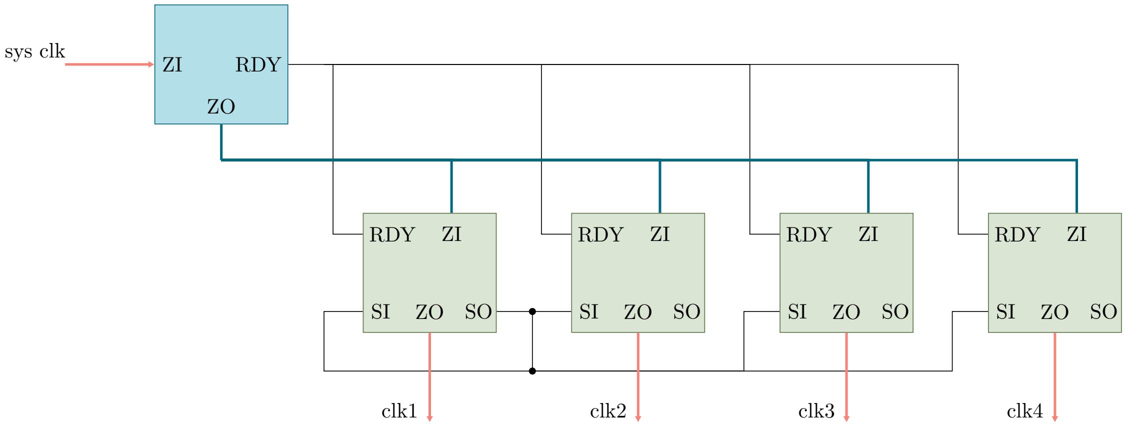


Figure 65: Clock generation circuit. WFG1 is the master WFG that synchronizes the output of the WFGs.

Each of these WFGs must have their tap delay set in such a way that a full phase cycle is covered. In this case each clock should have a phase offset of $\frac{2\pi}{4}$, but in general the phase delay can be set for each element by the number of taps θ_i,

$$\theta_i = 2\pi \left(\frac{i-1}{n}\right)\left(\frac{1}{f_{\text{clk}} d}\right) - 1 \tag{8.8}$$

where n is the number of clocks, i is the index of the clock and d is the tap delay of 159 ps. One is subtracted due to the design of the WFG. Setting the delay_taps generic to 0 results in a delay of 159 ps, while a delay of 0 is only achieved by setting the delay_on generic to 0.

This method of generating off-phase clocks can only be used if the total delay spans at least $(n-1)/n$ of the sampling clock period, as there is only a given amount of delays that can be used to generate an off-phase clock. For instance, in the 4x oversampling case, it must be possible to generate a clock that is delayed by 3/4 of the clock period. If the longest delay possible is determined by the FPGA architecture, the clock frequency has to increase to reduce the clock period. This places a lower bound on the optimal running frequency, and failure to meet this frequency will result in the last bin being wider than the others. Conversely, the frequency is limited on the upper end by the timing of the circuit.

The longest delay possible to make is 64 taps of 159 ps, resulting in 10.17 ns. In the 4x oversampling case, this must be 3/4 of the minimum clock period, meaning that the minimum clock frequency that must be used with 4x oversampling is 73.75 MHz. Taken to an extreme, the limiting frequency as the number of clocks is equal to the maximum available number of delays is the period of the whole delay line, corresponding to 98.33 MHz. This is a reasonable minimum frequency that is not likely to cause any issues.

8.6 Thermometer-to-binary encoder

With all fine TDC architectures discussed, the output of the TDC is thermometer code, a vector in which there is a single transition from 0 to 1 or vice versa. Each bit in the thermometer code corresponds to the input sampled at a difference in time from the preceding or following bit. This must be converted to a binary number to quantify the time interval measured by the TDC. A simple solution is to implement a lookup-table that converts the thermometer inputs to binary outputs. As discussed in section 5.3, this is complicated by the fact that in practical applications there is not always just a single transition in the priority code, which means all possible permutations of the input vector must be defined in the lookup-table. If the thermometer code contains multiple transitions, any solution that depends on edge detection will produce erroneous results. To account for bubble errors, the thermometer-to-binary encoder is implemented as a population counter (also known as a ones counter). It is possible to synthesize the code in Listing 3 as a chain of adders [Jaworski, 2015, orig. in Verilog]:

```
process : (clk)
variable cnt_temp : std_logic_vector(cnt'range);
begin
    if rising_edge(clk) then
        for i in tcode'range loop
            cnt_temp := std_logic_vector(unsigned(cnt_temp) + tcode(i));
        end loop;
    cnt <= cnt_temp;
    end if;
end process;
```

Listing 3: Combinatorial ones counter

This relies on the synthesis tool being able to recognize the structure and find a suitable circuit implementation. This has been tested, and NanoXmap manages to implement this using adder units. The binary output is equal to the count of the input zeroes.

8.7 Memory

There are two main types of hard memory resources available in the NG-MEDIUM, Register File Blocks (RFBs) and BRAM. The block ram is intertwined between tile-rows as shown in Figure 53. These blocks are synchronous dual-port implementations of SRAM. There are 56 of these blocks, and each block can store a total of 6,144 bytes. In total, this allows the instantiation of 2.6Mb of BRAM [Nan, 2020c, p. 17]. The RFBs are also synchronous dual-port RAM and are contained within tiles as shown in Figure 55. There are two of these per tile, and each one can store 64 words of 16-bits, for a total of 10.7 Kb across all tiles on the die. Both BRAM and RFB supports Error Detection and Correction (EDAC) mode, which is implemented with SECDED (Single Error Correction, Double Error Detection) codes. When EDAC is enabled, the maximum amount of memory in one BRAM is reduced from 6,144 bytes to 4,068 bytes. Furthermore, EDAC mode restricts the available address lengths and word lengths. These are shown in Table 7.

Table 7: Available modes of BRAM storage [Nan, 2020c, p. 17].

With EDAC		Without EDAC	
Address length	Word length	Address length	Word length
2048	1	49 152	1
2048	2	24 576	2
2048	6	12 288	8
2048	9	6144	8
2048	18	1024	12
		2048	24

8.7.1 Transfer function ROM

The transfer function between a time interval and incident energy is dependent on several factors: the gain of the CSA, the intrinsic energy of the detector material, as well as the voltage threshold of the comparators. In order to facilitate further calibration and fine-tuning, this transfer function has been implemented in a Read Only Memory (ROM). The ROM contains the energy for each combination of coarse and fine count. There are many ways to create a ROM in VHDL, but most commonly it is implemented either as an array of standard logic vectors, for which the index of the array is the address, or it is implemented as a case statement where the address serves as the switch variable. Both of these memory types can be laborious to define and maintain, but fortunately NanoXmap supports a synthesis directive *addMemoryInitialization* and proprietary file format *.nx* for initialising indexed array ROMs [Nan, 2020e, p. 75].

The initialisation file syntax is simple: each line in the file is the contents of the memory address corresponding to the line number, starting at zero. The contents of the file are subsequently linked in the NXPython compilation script like so:

```
project.addMemoryInitialization('getModels(<entity>_<rom_signal>)', 'NX', '
    <nx_file_path>')
```

Here <entity> is replaced by the entity in which the ROM appears, <rom_signal> is the signal name of the array of vectors, and <nx_file_path> is replaced by the path to the *.nx* file with the ROM contents. This relies on the synthesis tool being able to recognize the ROM structure.

This same principle can be used to fill the ROM for simulation: Questasim supports a command,**mem load**, that can inject the contents of a *.mem* file into memories in simulation [Men, 2015, p. 194]. The structure is the same, but the .mem file must contain a header with information about the memory, as well as the address in hexadecimal numbers on each line.

The benefit of using this method of applying the transfer-function ROM is that it is independent of the VHDL configuration, and can be fine-tuned without making changes to the actual firmware.

A Python script named *gen_rom.py* is used to generate the *.nx* and *.mem* files. The relationship from Figure 11 cannot be used as the pulse is not exactly defined by any equation as illustrated by Figure 18. To improve accuracy, the script uses the actual measured pulse from the IDE1180 pulse shaper, scaling it up and down based on the calculated amplitude for a range of energies. In the configuration used in this thesis, the gain was set to 19 mV/fC and the ionization energy of the material was set to 3.6 eV. The range of energies is converted to a range of pulse amplitudes, and these are multiplied with the normalized pulse shape to approximate the pulse shape for each energy. Next the crossover points for the threshold are defined by finding the intersection, and each energy is mapped to a time interval.

The next step is mapping the time intervals to coarse and fine counts to be used in the FPGA based on the sampling clock frequency. The coarse count is found by doing the integer division of the ToT and the clock period. The fine count is found by taking the modulus of the same integer division and multiplying it by the number of fractions that each count should be mapped for. After this, duplicate coarse and fine counts that map to different energies are removed.

Because the function is non-linear, and the input energy range is linear, the lower energy range is underrepresented. This is because a step in the input energy may correspond to a change of several fine counts. Increasing the sampling resolution of the input energy is impractical because the program is quickly starved of memory. Some improvement is realized by including more samples at the lower energy range than the higher energy range. To ensure that every single combination of coarse and fine count is included in the memory, linear interpolation is applied on the data set. The energy bins are also offset by half the distance to the next energy, as per (5.3). Each coarse and fine count that can be mapped to an energy is copied into a table that contains every single combination of coarse and fine count. Then the empty rows are interpolated to fit the rows that exist.

The table is saved to *.nx* and *.mem* files, and each row corresponds to its address in the ROM. Each coarse count address is separated by the number of fractions, and all the addresses between are for different fine counts. This means that the relationship between the address a, the coarse count C and fine counts is,

$$a = Cn_{\text{fractions}} + R - F \tag{8.9}$$

Where R is the fine count for the rising edge and F is the fine count for the falling edge.

To avoid multiplication when addressing the ROM it can be useful to make the number of fractions a power of 2: for example 4, 8 or 16. Multiplicating a binary number by a power of two is the same as shifting the bits leftwise by the exponent.

8.7.2 FIFO

A First In First Out (FIFO) memory is needed for the raw particle hit data that is to be transmitted through the telemetry circuitry, and this is ideally implemented in RAM blocks. The more blocks that are used, the longer the data can be sampled before being transmitted. The FIFO is written in the inferred style, with signals named head and tail that keeps track of where data should be inserted and extracted from the array. When the FIFO is full, it asserts a signal saying it is full, and similarly it asserts a signal when empty. It has been tested and confirmed that NXMap is able to synthesize this style of FIFO into RAM, and it can be mapped to any of the RAM implementations available.

8.7.3 Synchronization

Since the input signals are asynchronous signals, they must be synchronized. Sampling the input with a clocked register is an effective synchronizer, but it includes a non-zero risk of metastability, disturbing the operation of the circuit. With the inclusion of a second clocked register, this risk is minimal.

The path through the two registers of a clock synchronizer should not be considered during the static timing analysis, because even if the first output is not ready before the clock flank, it is likely to resolve itself before the next clock flank. This applies to any circuit with an explicit clock synchronizer, be it a clock-domain crossing or sampling of an asynchronous signal [Bhasker, 2009, p. 38]. To avoid these paths being regarded during timing analysis, they can be specified as false paths. This is done in the compilation script,

```
project.addFalsePath(’getRegisters(<launchingregistername>), getRegisters(<
    receivingregistername>)’)
```

8.8 Histogram engine

The histogram engine is the central module of the data acquisition, and together with the calibration engine and TDCs it forms the entirety of the Data Acquisition (DAQ) module. It is responsible for monitoring the TDC channels, converting the coarse and fine counts into energies, and binning the hit in the appropriate histogram bin. The histogram is continually constructed, but every 50 milliseconds it is prompted by the timer circuit to sample and reset the histogram. The sampled histogram is stored in the register bank, from where it will be transmitted to the PCM encoder by the telemetry module.

The histogram engine is clocked by the main system clock, and since it directly interfaces with the faster clocked TDC, synchronization is required between the clock domains. This is realized by control signals going between the histogram engine and the TDCs. When a hit is detected in a TDC, the coarse count and fine thermometer code is loaded into registers and the TDC asserts a control signal to inform the histogram engine of the hit. Once this control flag is raised, the TDC goes dormant. When the histogram engine has extracted the required information and binned the hit, it in turn asserts a service flag to enable the TDC again. The hit flags are cleared, and then the service hit flags are cleared. The time period in which the TDC channel is disabled until it is serviced and re-enabled is what constitutes the dead-time of the detector. The histogram engine is structured as a Finite State Machine (FSM) and the states are illustrated in Figure 66. The green states are the states that constitute normal operation, whereas the yellow states are the states that are forced during calibration. These are mutually exclusive, and no hit will be serviced when calibration is being performed.

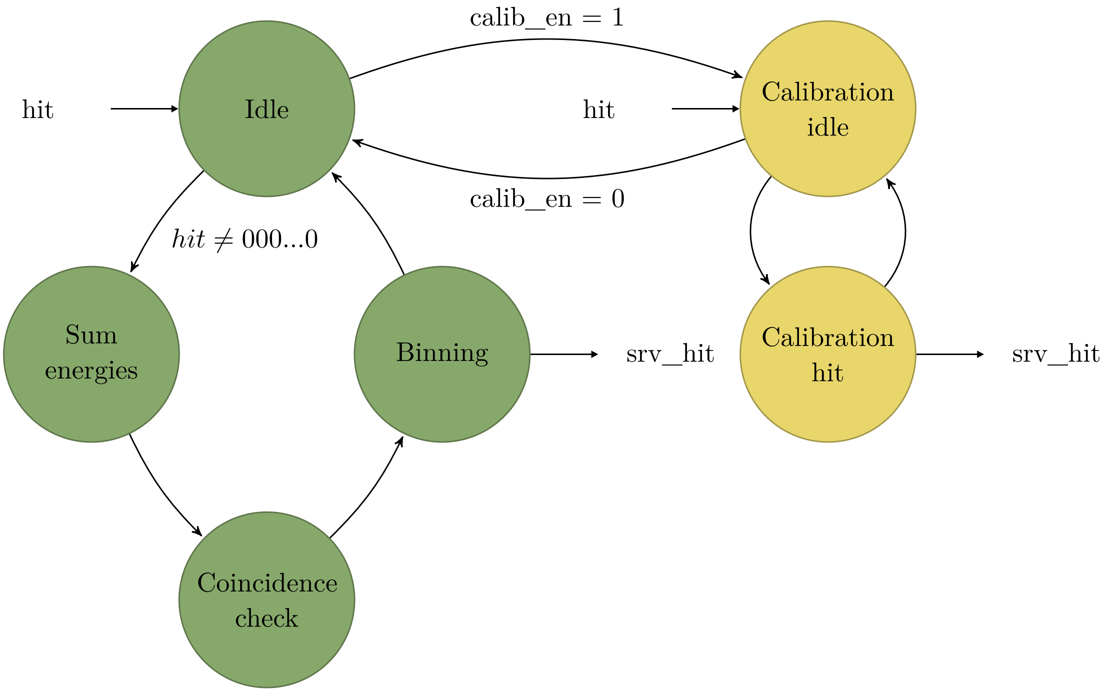


Figure 66: Histogram engine state machine.

8.8.1 Binning

Instead of creating delimiters for coarse and fine counts and binning the hits directly, the exact energy is first determined with a ROM and stored as an unsigned number. Converting the counts into energies explicitly produces circuit overhead, but is necessary because the transfer function is not linear. Each ToT interval must be converted to an energy before it can be summed. The stored energy is a multiple of 10 so that some fractional information is included, i.e. an energy of 30405 corresponds to 3040.5 keV. As a consequence, the maximum energy detectable with a 16-bit word is 6553 keV.

An added benefit of converting to an energy in the circuit is that it is much easier to alter the histogram definition at a later point of development, needing only the number of bins and an array of energy delimiters. The incoming energy signal is also useful in the testbench to compare input times to output energies.

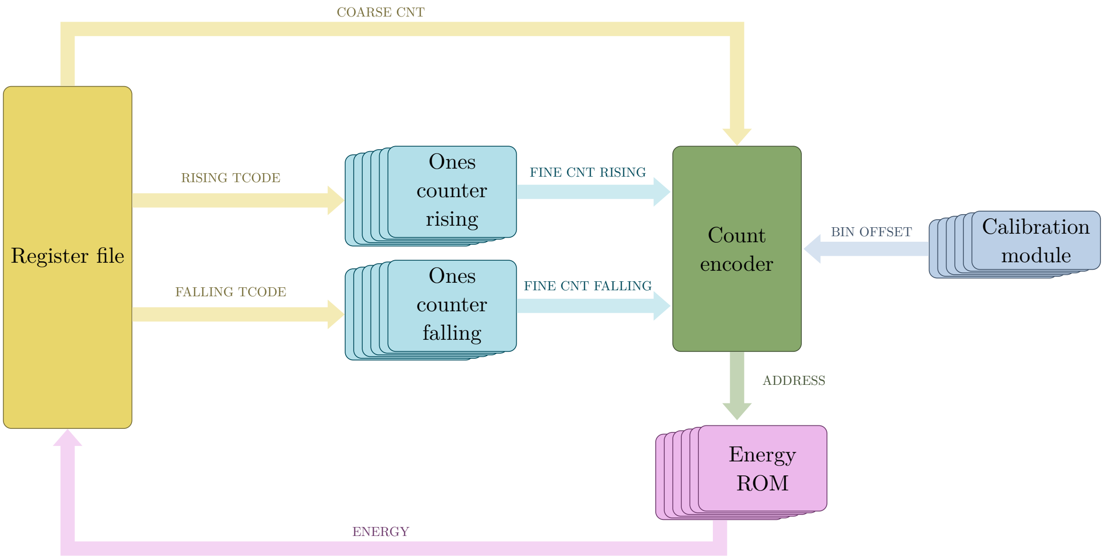


Figure 67: Histogram engine data flow.

The data flow of converting TDC outputs to energies is outlined in Figure 67. The count and thermometer code from each TDC is accessed from the register file. The thermometer code for both rising and falling edges of the comparator input are fed directly into ones counters. The output from these are together with the coarse count fed into the count encoder. Also included as an input to the count encoder is a predefined bin offset for each fine count from the calibration module. The offset is defined by the known bin widths and points a fine count to the appropriate energy bin. The inner workings of this module is described later in section 8.10.

The purpose of the count encoder is to convert the counts from the TDC into addresses that can be used to look up the energy in the energy ROM. To do this, it subtracts the falling count from the rising count. This is because the number of ones in the rising thermometer code add to the total interval, while the number of ones on the falling flank subtracts from it. This is not immediately obvious, but is demonstrated in Figure 68.

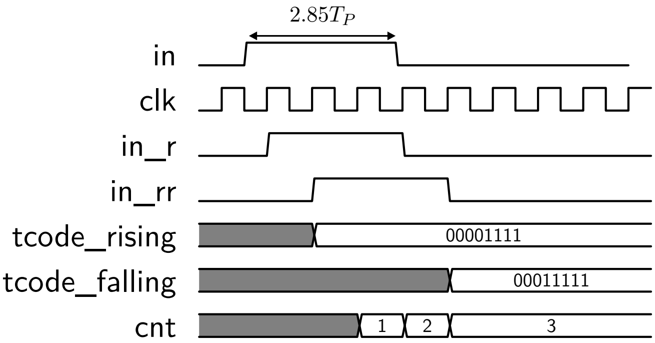


Figure 68: Interpolating TDC example with thermometer code.

In this example, the true total length of the input pulse is 2.85 times the clock period T_p, and the circuit features 8 tap delays that in total sum to one clock period. The input is high for three clock flanks, so the coarse count is 3. Sampling the 8-bit thermometer code on the rising flank yields a count of four ones, while on the falling flank the result is five ones. The thermometer code is inverted in the falling case to create comparable patterns on both clock flanks. As the input rises halfway in a clock period, the four out of eight positive bits in the thermometer code means that the input signal transitioned already four tap delays before the sampling clock flank, with the result that half of a clock period should be added to the sum. Similarly, the count of five on the falling edge means that the input signal went low five tap delays before the sampling point. Obviously, this means that on this edge the count was overestimated by 5/8 clock periods, and this should be subtracted. The resulting calculation and converted time interval is,

$$T = T_P \left(3 + \frac{4}{8} - \frac{5}{8}\right) = T_p \left(3 - \frac{1}{8}\right) = 2.875 T_p \tag{8.10}$$

The coarse count and summed fine count are used to construct an address for the ROM, and since the fine count can be negative, it is desirable to convert these measures so that they are strictly positive. The above time interval T of $(3 - 1/8)T_p$ can also be expressed as $(2+7/8)T_p$, and this is used to create the positive counts. If the fine count is a negative, the coarse count is subtracted by one and the fine count is subtracted from the number of tap delays. If the fine count is positive, however, no conversion is needed as the coarse count can only be positive.

Since there are eight pixels, and each pixel should have its own histogram, eight histograms are needed. However, as discussed in section 4, the hits that purely hit the back sensor should not be included. This reduces the number of histograms to four. Furthermore, any particle that hits multiple pixels will go into a shared histogram for coincidence hits, bringing the total number of histograms to five: one for each front pixel, and one shared for coincidence hits. If a hit uniquely hits one pixel in the front layer and also hits one or more pixels in the back layer, the hit is accepted and binned in the histogram corresponding to the front pixel it hit.

8.9 Coincidences

If there is energy deposited in two or more pixels, a coincidence event has occurred. This can be caused by two distinct mechanisms, either several particles struck the detector, or the same particle has deposited energy in several pixel sensors. In the first case, each hit should be treated as a separate event, and in the latter case the energy of the pixels should be summed. When a particle hits the semiconductor material, it will not immediately deposit all its energy in the vicinity of where it struck. Instead, it will scatter through the material, continually depositing energy as it propagates through the material. The process of scattering allows the particle to radically change trajectory, and it may therefore move in any direction, even to other pixels in the same layer. The aim of the coincidence check is thus to discern the scattering particles from multiple particle hits.

8.9.1 Horizontal coincidence check

The factor of distinction is the time difference between the hits being recorded. The velocity of any incident >30 keV electron is relativistic, so the timescale of charge deposition is many orders of magnitude shorter than the clock period of the sampling circuitry. A simple implementation of this coincidence check is to determine if there is a particle event in more than one sensor in the same clock period. This is limited by the timing constraints placed on the clock by the TDC circuit. A slower clock increases the probability that multiple separate and unique events happen within the same clock cycle and are counted as the same event. This probability can be found with exactly the same equation as that for pile-up given in (2.12). With a 100 MHz clock and 10^6/s event rate, this probability is less than 0.01% per clock period and deemed negligible. If more accuracy is needed in the horizontal coincidence check, the use of the delay line to reduce the time window that defines a coincidence event.

8.9.2 Multi-hit capability and pipelining

The dead-time between when a hit is first asserted until it is recorded, binned and serviced inescapably causes hits to be missed, either by simply not registering them or by mistakenly grouping them together with independent hits. This is due to the random nature of incident hits. If a hit event has occurred in a pixel, there is a non-zero probability that another hit will occur within any given time frame. For this reason, adding multi-hit capability improves the efficiency of the detector by not ignoring hits while previous hits are treated. Due to how the signal is generated, there is no way to avoid pile-up in a single pixel, as discussed in section 2.3. However, as each channel and accompanying TDC are completely separate instances there is no reason for a hit in one pixel to introduce dead-time in another pixel. Still, the data acquisition module must at some point funnel all the pixel data into a shared histogram, and this poses an architectural problem, as a hit cannot be serviced in zero-time.

The proposed solution is the inclusion of pipelining to register hits sequentially. The hit signals from each TDC correspond to a bit in a vector in the histogram engine. If a hit occurs in a pixel, the hit vector is registered and the histogram FSM moves into a sequence of states to service the hit. Before it returns to the idle state, it returns a service vector with the exact same bits as the initially registered hit vector. If another hit happens while the first hit is being serviced, that hit will not be serviced because the histogram engine FSM is blind to changes in the hit vector while it is servicing a hit. The result of this is that once the FSM returns to the idle state, it will immediately start servicing the second hit consisting of the remaining bits in the hit vector. One particular disadvantage with this approach is that it is limited to two concurrent hits. By making the FSM blind to changes in the hit register, it is also blind to determine coincidence events. In the unlikely event that several particles hit the sensor in the exact time window that it is treating a prior hit, the hit signal from all of those particles will be registered as the FSM returns to the idle state. The result of this is that it will treat the hits as a coincident event, and their energy will be summed erroneously.

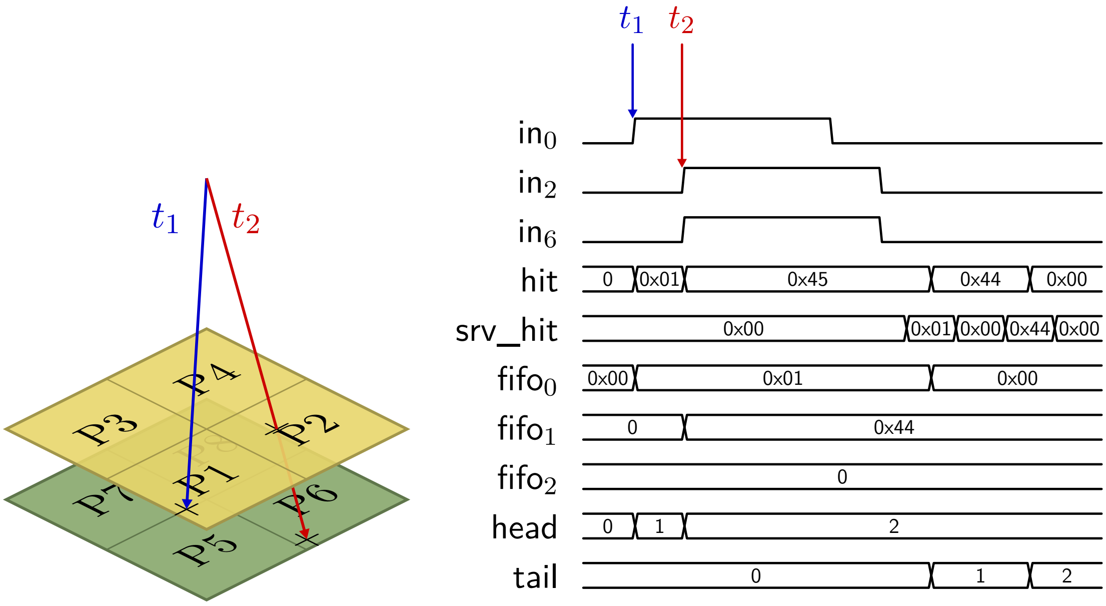


Figure 69: Particle pipelining with hit registers. Hits occurring in separate clock cycles are buffered in a FIFO to be treated sequentially after the hits have been recorded.

Extending this principle to any number of closely occurring hits can be done with a small FIFO and a separate coincidence monitor process. The coincidence process will continually monitor the hit vector for a change, and if there is a change, the hit vector will be placed into the FIFO. This process is shown in Figure 69. Each hit occurring in the same clock cycle are registered in a FIFO. To not store a hit in a pixel that is actively being processed, the hit vector being stored in a FIFO is XORed with the current hit vector. In the example, the first hit is stored as 0x01, and when the second hit occurs, the histogram engine is busy processing the hit vector 0x01, so the 0x45 vector is XORed with the 0x01 vector, resulting in 0x44. This correctly identifies the second particle as a coincidence in pixels 2 and 6.

Since the occurrence of multiple hits not sufficiently spaced from each other is relatively rare, the FIFO does not need to be long. A FIFO is length 3 is probably sufficient for all but the rarest of cases.

8.10 Calibration

The purpose of calibration (as discussed in section 5.2) is to determine the delay separating each tap in the delay line in the presence of routing and PVT variations. More concretely, the aim is to determine how to interpret the thermometer code output by the delay line and how to place the bin delimiters to reflect actual delays. Calibration is better explained with an example, and therefore this section will consider the circuit in Figure 70. Illustrated here is a delay line wherein each delay element contributes a nominal delay of τ_{nom}. If the first delay element were to be placed in a separate tile from the rest of the delay elements, and the routing between tiles incurs a delay of τ_i, the output of the second bit in the thermometer code is ready after $\tau_{\text{nom}} + \tau_i$. All subsequent bits are delayed by the additional cross-tile routing delay with the result that the entire delay line after this delay is shifted from the nominal. In practice, this means that the first bit now encompasses any time interval between 0 and $\tau_{\text{nom}} + \tau_i$, because $\tau_{\text{nom}} + \tau_i$ is how long it would take for the signal to propagate to the second bit.

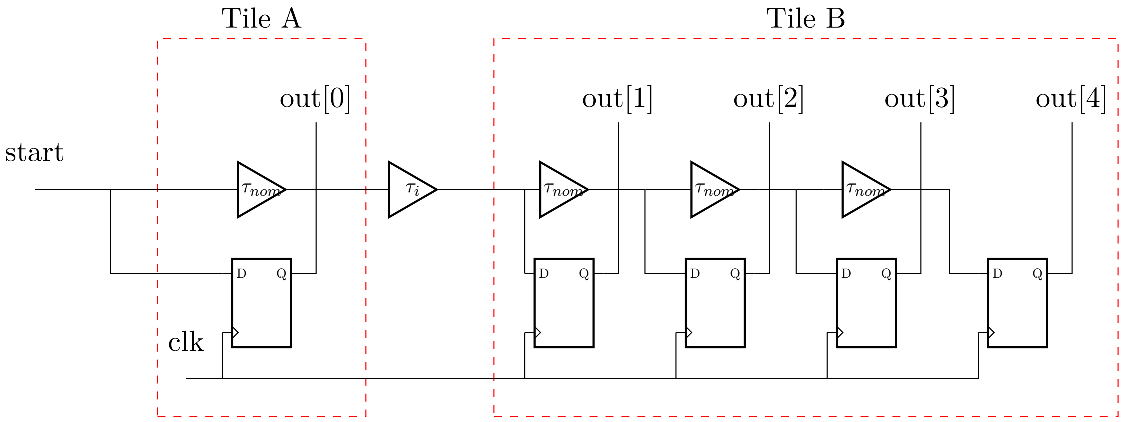


Figure 70: Delay line with unwanted routing delay.

For simplicity, the case where the interconnect delay is equal to one nominal delay is considered. The time diagram of this circuit in response to an input is detailed in Figure 71. The interconnect delay τ_i is added to the nominal delay, with the result that the bin width increases and information of transitions within this time period is lost.

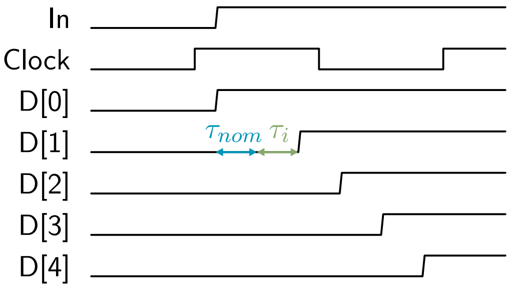


Figure 71: Timing diagram of an input in the unbalanced delay line.

The calibration method applied for the DEEP firmware is the statistical calibration method from 5.2.1. This is a bin-by-bin calibration which means that ultrawide bins are reflected accurately, making it a good choice for a FPGA based TDC. The other two calibration methods discussed are averaging bin width methods that are more suited for implementation in CMOS where seemingly random and long delays are not expected. As an example, repeatedly stimulating the above discussed TDC delay line with a uniform distribution of inputs yields a histogram similar to the one shown in Figure 72.

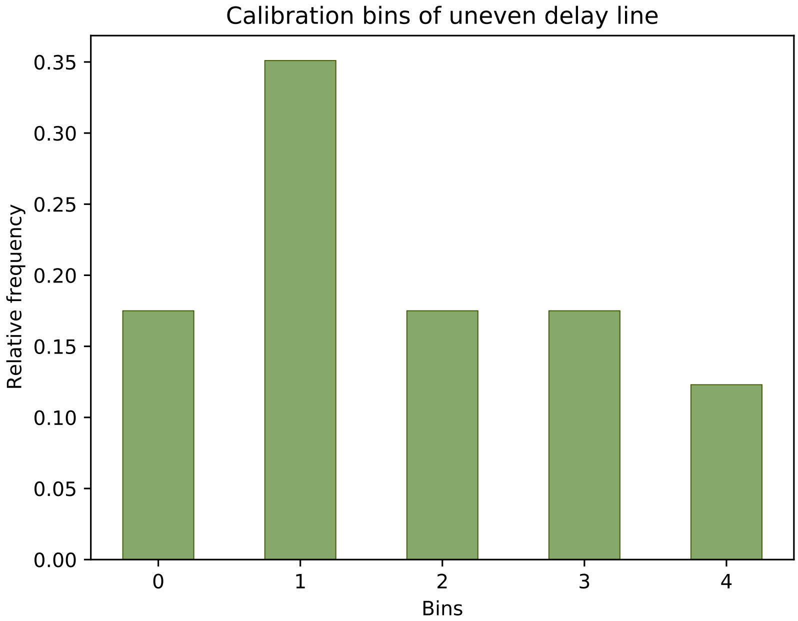


Figure 72: Histogram of uniform inputs being applied to the above unbalanced delay line.

Since the input is a uniform distribution, the ideal TDC is expected to generate a histogram in which each bin has the exact same height. That is, the output is ideally uniform to reflect the uniform nature of the input. What is shown here is that the bin heights of the histogram is shaped by the imperfect delay line, and the height of each histogram bin is proportional to the delay of the bin. This makes sense intuitively because if a bin is twice as large as it should be, it would encompass a doubled fraction of the clock period.

Practically, the uniform distribution of inputs in this circuit is generated by an external clock asynchronous to the main system clock. The asynchronous clock has a different phase and frequency from the sampling clock. Rising edge clock flanks from the calibration clock will slide relative to the sampling clock period and hit every part of it uniformly. This process is further aided by clock drift and clock jitter. A calibration clock derived from a system clock through a PLL cannot be used because the phase is locked to the system clock. If the calibration clock had some multiple frequency of the clock it is derived from, it would be synchronized to strike in the same parts of the clock period every time.

To dynamically adjust the delimiters of the thermometer code in the FPGA, an error array is generated by the calibration circuit. This error array is equal in length to the number of delay elements in the delay line, and consists of precalculated offsets for each bin in each TDC. To create the error array, both the nominal delimiters and target calibrated delimiters must be calculated. Both of these are the prefix sum of all delays preceding the current tap. In the nominal case, all delays being equal reduces the prefix sum to a product, and the nominal delimiters nd_k can readily be calculated as,

$$nd_k = k\tau_{\text{nom}} \tag{8.11}$$

In practice, this array is precalculated before synthesis and included as a constant array in the FPGA firmware.

The calibration delimiters cd_k are defined by the prefix sum of its bin heights, and must be calculated element-wise after calibration hits are recorded,

$$cd_k = \sum_{i=0}^{k} \tau_i \tag{8.12}$$

Doing this prefix sum in VHDL is achievable with a double loop as in Listing 4.

```
    v_calibration_delims := (others => 0);
    for i in 0 to c_num_delays - 1 loop
      for j in 0 to i loop
        v_calibration_delims(i) := v_calibration_delims(i) +
calibration_delims(j);
      end loop;
    end loop;
```

Listing 4: SDF content of a NX_CY element

Since the prefix sum is cumulative, the last delimiter should be equal to the number of calibration hits that were input to the delay line.

Once both the nominal and calibrated prefix sum arrays are attained, the error array is simply the element-wise integer division of the calibrated array subtracted from the nominal array with the number of hits separating each memory location,

$$\mathrm{err}_k = \frac{(cd_k - nd_k)}{n_{\mathrm{hits}}/n_{\mathrm{fractions}}} \tag{8.13}$$

The denominator of this equation is a precalculated constant, and corresponds to the number of hits that are expected in each bin.

The error for each tap in the delay line represents how the actual delay from the start of the delay line to the current tap deviates from what one would expect in the nominal case. As the conversion is limited by the ROM granularity as well, implementing a higher granularity in the ROM results in greater accuracy, as otherwise the calibration error can only offset by a whole tap difference. Accuracy being improved by calibration offset accessing nominally unused ROM locations is demonstrated in Figure 73. In the nominal case, the taps would move directly in a downwards fashion and map to each second energy location. Because the second bin is twice as large as expected, the prefix sum shifts each subsequent element right-wise by one. Note that the last bin is typically never larger than the nominal because that can only happen when the delay of the entire delay line is shorter than the clock period.

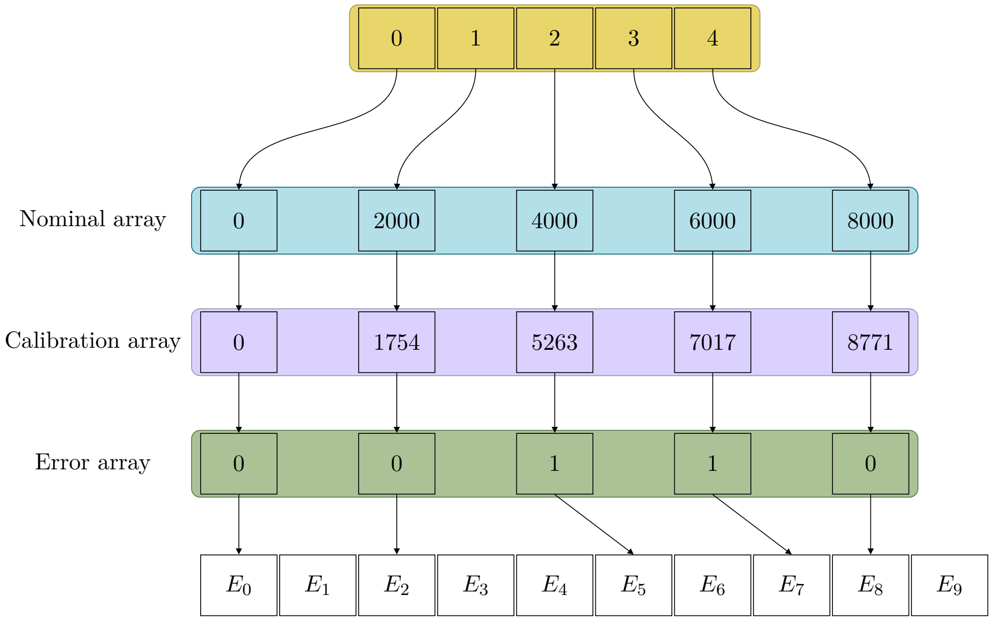


Figure 73: A four-tap delay line calibrating into a ROM with ten fractions per coarse count and 10000 calibration hits.

Similarly, it may be useful to have more tap delays than ROM fractions due to timing and resource constraint as low clock frequencies demand longer delay lines. This is equivalently solved by the calibration circuit since several delay taps are mapped to the same ROM location through the integer division.

The calibration process is fully automatic and will run until the required number of hits is counted up. A calibration enable signal is periodically raised by the timer module, and the frequency is specified by a constant in the package file. Figure 74 shows the calculation of the error offset in a delay line where the first bin is twice as large as the remaining bins. Each bin initially points to the first bin and sequentially increments the offset as each calibration hit is sorted into each bin. Once the calibration process is done, each bin is incremented by one to account for the initial large bin.

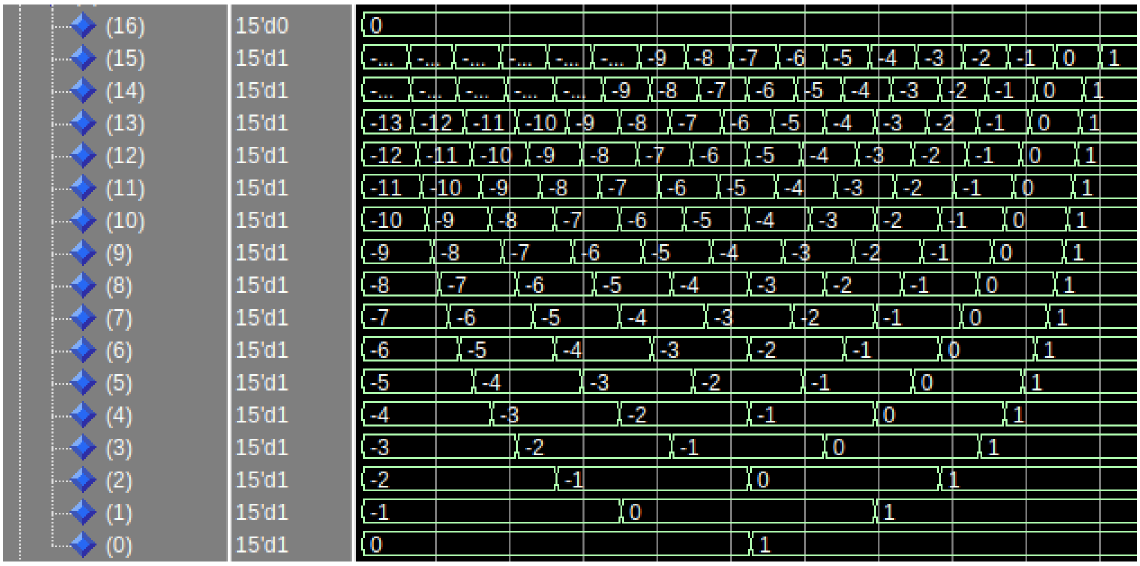


Figure 74: Calibration scheme calculating offset for each bin in a 16 bin delay line where the first bin is twice as large as the remaining bins.

With the calibration offsets calculated, the address equation from (8.9) is modified to include the offset,

$$a = Cn_{\text{fractions}} + (R + \text{err}_R) - (F + \text{err}_F) \tag{8.14}$$

Here, R is the fine count for the rising edge, err_R is the rising edge offset, F is the fine count for the falling edge and err_F is the falling edge offset.

8.11 Telemetry communication

The telemetry of the firmware is structured into two layers, where the PUSEK board in the rocket can be thought of as a third and physical transmission layer and not controlled by the DEEP firmware.

The bottom layer on the readout PCB is the PCM interface developed by Kjetil Ullaland for the firmware for the previous iteration of the DEEP project. The module has four input control and synchronization signals and one serial output to the rocket electronics. It has a data out vector with a word length of 16 bits, which corresponds to a minor frame. In addition to this, it requires a system clock, a system reset and a high frequency sampling clock *hclk*.

Since two channels of the PCM communication are available for the DEEP instrument, the PCM interface has two output signals, re0 and re1. These correspond to read enable 0 and 1 on respective channels. When either of these signals are asserted, the data on the *data_in* vector is loaded into a register and transmitted to the PUSEK PCM encoder.

To handle the communication at a higher abstraction level, the telemetry module constitutes the third and topmost layer of the transmisson chain. It monitors the read enable signals from the PCM interface to know when to load data, and it reads registers from other parts of the circuit to know when and what to transmit to the PCM encoder.

On channel 0, the histogram data is transmitted, while on channel 1, raw data is transmitted. The data for both channels comes from the histogram engine. The raw data is stacked in a FIFO and transmitted whenever the FIFO has filled up, while the histogram engine is prepared at regular intervals, triggered by an adjustable timing signal. The system is illustrated in Figure 75.

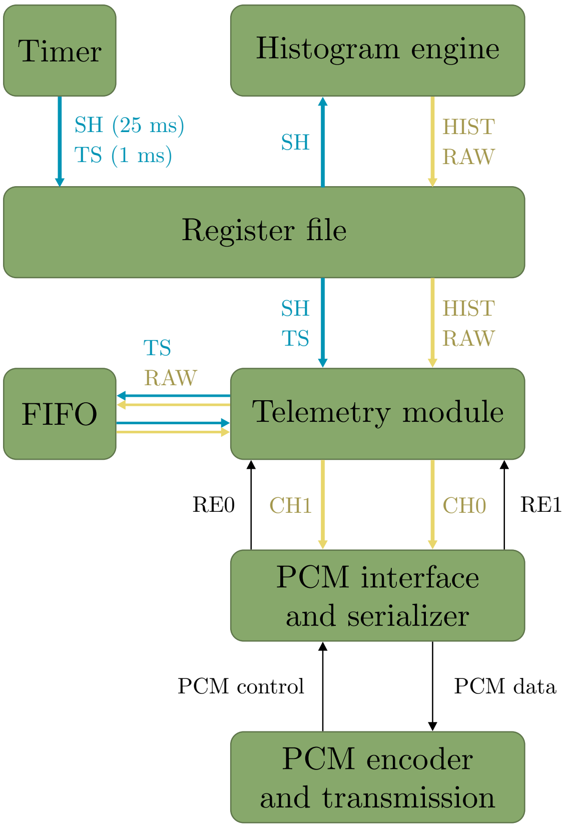


Figure 75: Telemetry communication chain, SH is the sample histogram signal, TS is the timestamp signal. HIST and RAW are histogram and raw particle data from the histogram engine.

Whenever a hit occurs, the appropriate header, bin and channel are stored in a byte vector. Simultaneously, a data valid signal is asserted for one clock cycle. The signal is monitored by the telemetry engine, which stores the byte in a FIFO. To be able to discern when a hit occurred in the raw data stream, a timestamp is also stored in the FIFO every millisecond. This is an empty package that consists only of a timestamp header, and every hit that is positioned between timestamps in the FIFO occurred in the same millisecond. An example of how a FIFO stack can look like is illustrated in Figure 76. Information about how many hits happened in a millisecond is obtained by looking at how many are placed between two consecutive timestamps in the FIFO stack.

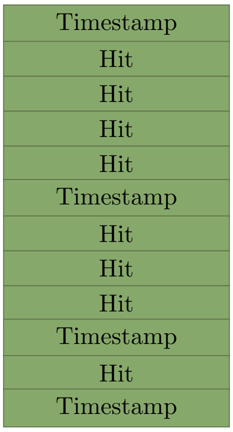


Figure 76: Raw data FIFO structure.

There are two PCM channels available, each with a bandwidth of 5,787 bytes per second. The first channel will be dedicated to the histogram. Transmitting a total of five histograms consisting of 16-bit bin counts and seven bins results in a total data size of 70 bytes. The histograms are transmitted with a header, followed by each count in pixel-wise order, as illustrated in Figure 77.

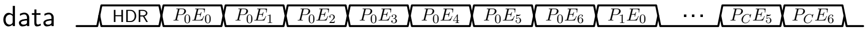


Figure 77: Transmitting histogram data with telemetry.

A total of 82 histogram sample events can be supported per second, or one every 12 ms. Another limiting factor is counter overflow in any one bin. If considering the worst case that every particle enters the same b-bit energy bin, the maximum event rate r that can be sustained is,

$$r = \frac{2^b - 1}{h} \tag{8.15}$$

Here, h is the histogram sample period. The rate decreases for increasing histogram sample periods, as the counters have to count for longer periods at a time. Setting this sample period to 25 ms gives a margin on the telemetry bandwidth while at the same time supporting an event rate of 3.28×10^6 which is greater than the expected count rate. Note that this is a very conservative estimate, and on average each pixel should receive a quarter of the hits, and longer sampling periods can be supported.

On the second channel, raw data will be transmitted. Considering that a hit is one byte with encoded information, and assume that the hits are buffered by a FIFO such that they can continually be transmitted at the maximum data rate. This means that 5,787 hits can be recorded in a second. Obviously this is much lower than the expected count rate on the order of 10^5 to 10^6 particles per second. The raw data channel will yield detailed snapshots of the particles hitting the detectors, but will only contain a fraction of the observed hits.

9 GATE/VHDL TESTBENCH FRAMEWORK

The GATE simulation outputs the information in a ROOT data file. ROOT is a data analysis framework developed by CERN for C++. The simulation output can be read in ROOT, but the ROOT project also includes bindings to read and modify ROOT files in Python through a package named PyROOT. Since these are bindings to the compiled ROOT framework and not a reimplementation in Python, the performance of PyROOT should be similar to ROOT C++.

The simulation testbench process is two-fold. First the data from the GATE simulation is read and structured in PyROOT, where it is analyzed. This part of the simulation is the ideal case in which raw data of particle hits is analyzed and transformed into time-intervals that correspond to the output of the comparators on the input channels. These intervals are analyzed on a high abstraction level to develop an efficient algorithm to generate a high-level output spectrum that can be compared with the input. The second part of the simulation is transferring the algorithm from the high-level analysis into the hardware domain with VHDL. The hits in the sensor are transformed into time intervals for each of the pixel channels, and the comparator inputs of each channel are stimulated by the testbench. The FPGA configuration converts each of the time intervals back into energies and the internal histogram engine will create a low-level output spectrum that can be compared to both of the aforementioned spectrums. An overview of the system is shown in Figure 78.

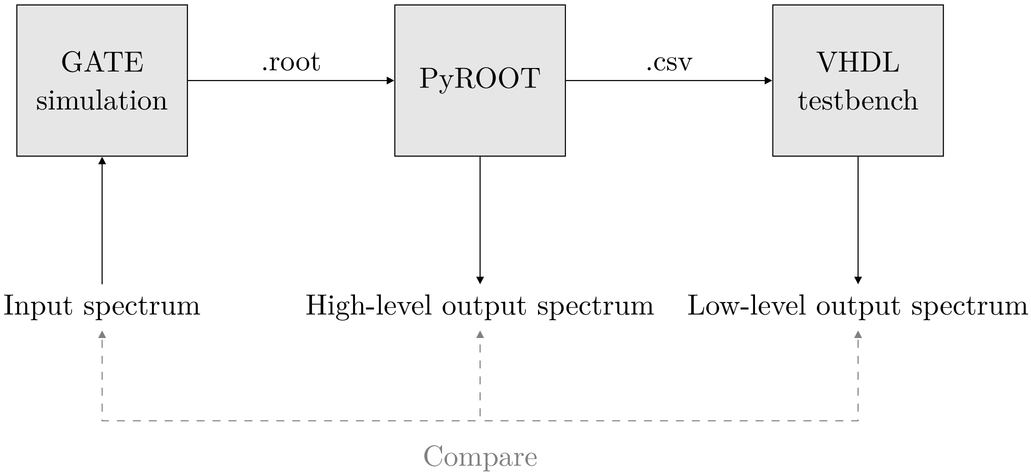


Figure 78: GATE/VHDL simulation overview.

The time of the collision can be extracted from the GATE simulation data, but the distribution of the simulation is uniform, meaning that each event happens exactly at evenly spaced intervals over one second. As discussed in section 2.3, particles will strike the sensor at random time intervals, and is thus best modelled by a Poisson distribution. To account for this, the particle data is extracted with no information about the time-domain, and then the interarrival times are modelled using a Poisson process. The time between each particle hitting the sensor is an exponential distribution with a cumulative probability distribution of [Devore, 2012, p. 199],

$$F(t) = 1 - e^{-\lambda t} \tag{9.1}$$

A random interval can be generated by using the inverse transform sampling method [Thomopoulos, 2013, p. 28]. The inverse exponential function is defined as,

$$F^{-1}(p) = \frac{-\ln(1-p)}{\lambda} \tag{9.2}$$

By randomly generating a probability p using a uniform distribution $U(0,1)$, a Poisson process interevent time can be obtained by feeding it into (9.2). An array of time intervals is easily generated with NumPy in Python,

```
    randarr = np.random.rand(num_particles)
    poisson_process = -np.log(1-randarr)/rate
```

The array of interarrival times are saved as a Comma-separated values (CSV) file and read by the VHDL testbench along with the ToT intervals. The interarrival times are handled by a process that reads a line from the CSV and toggles a particle event, then it waits for the number of nanoseconds in that line before it reads another line. When the particle event is triggered, a separate process reads a time from the CSV and formats the time intervals that each pixel should have its ToT signal driven high. Simultaneously, each pixel has its own process that reads the formatted time interval and drives the signal high for the specified time. Structuring the testbench like this ensures full parallelism between the pixel events, allowing the same particle to strike several sensors and drive the comparators high for different time intervals. An example of hits being generated in the testbench is shown in Figure 79.

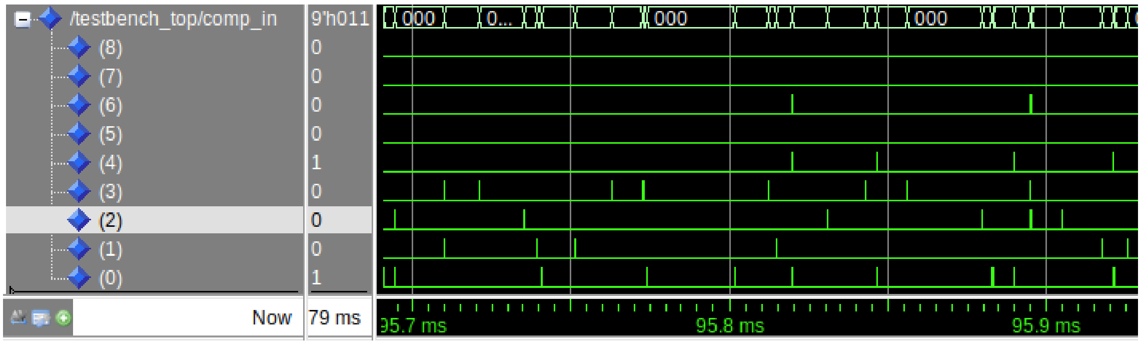


Figure 79: Particle hits from GATE simulation striking the sensor with varying pulse widths and random interarrival times.

10 RESULTS AND DISCUSSION

The GATE-VHDL framework discussed in section 9 is used to compare the different TDC architectures implemented. Although the energy ROM is generated through a simulated transfer function, some losses are expected when mapping true energies to energies from the ROM. These errors stem from insufficient energy resolution in the ROM, and also the interpolation method used when generating it. An example input spectrum is compared to the same spectrum when converted into ToT intervals and converted to energies with a generated energy ROM in Figure 80. This is done directly in Python and the conversion from energy to ToT intervals may be assumed perfectly synchronous.

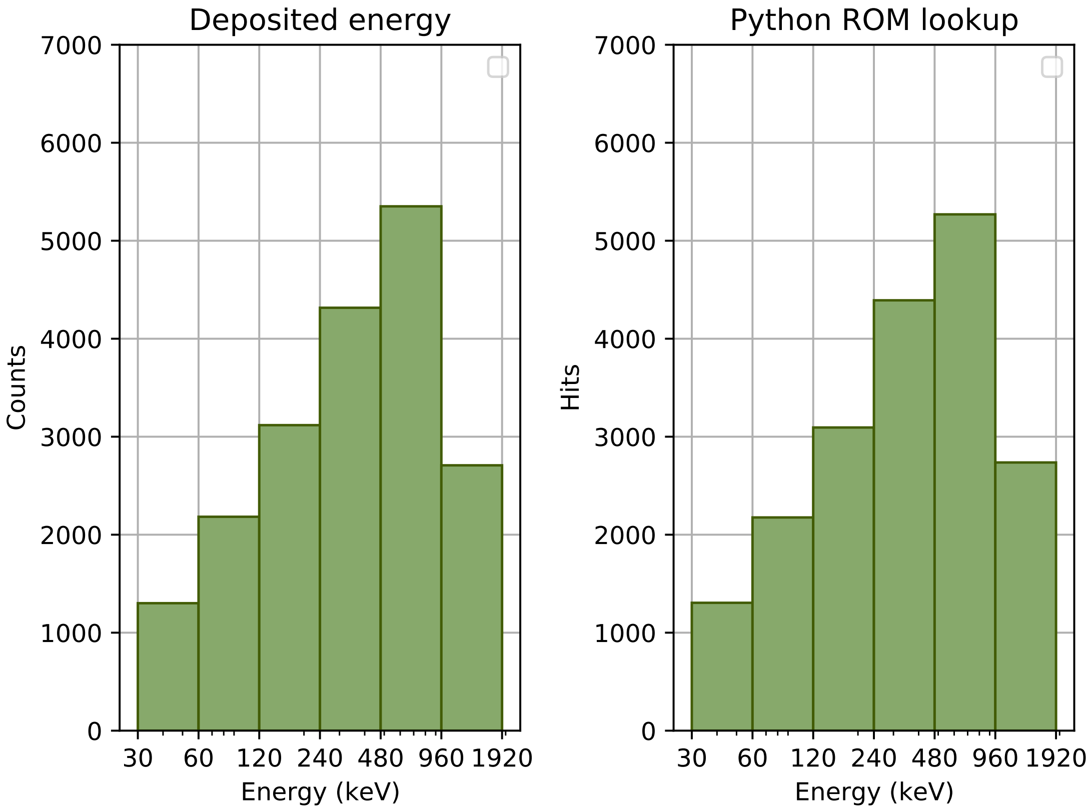


Figure 80: True input spectrum compared to ROM lookup of the same spectrum.

Because the firmware is limited by the conversion ROM and has no information about the true input spectrum, it makes sense to compare the converted energies to the ROM spectrum. If the converted energy spectrum in the testbench is closer to the true spectrum than the energy ROM, it must be regarded as an error, even though it is an error in the right direction. Furthermore, the conversion ROM is different for each clock frequency and has to be regenerated if changing the frequency to account for this. Small variations between the ROMs of different frequencies are also expected, and may be fine-tuned at a later point.

To demonstrate how the resolution of the TDC affects particles of known energies, a simple simulation is run with a radiation source that emanates electrons at 50 keV, 150 keV, 250 keV, 500 keV and 750 keV at exactly the same rate. These are first converted to energies in the testbench using a 100 MHz sampling clock and no fine TDC, and then another time with exactly the same parameters and with the inclusion of the delay line fine TDC. The result from this can be seen in Figure 81. As expected, the counter can mostly discern the particles of the lower energies, but starts to struggle at 500 keV and 750 keV. This is caused by the transfer function; at these energies a small difference in ToT corresponds to a large difference in energy. The delay line TDC has an effective resolution of roughly 300 ps, which is far superior to the 10 ns period of the counter clock. This TDC has no problem discerning the lower energies, and remains fairly accurate even when converting the higher energy particles, although a slight underestimation can be observed at these energies.

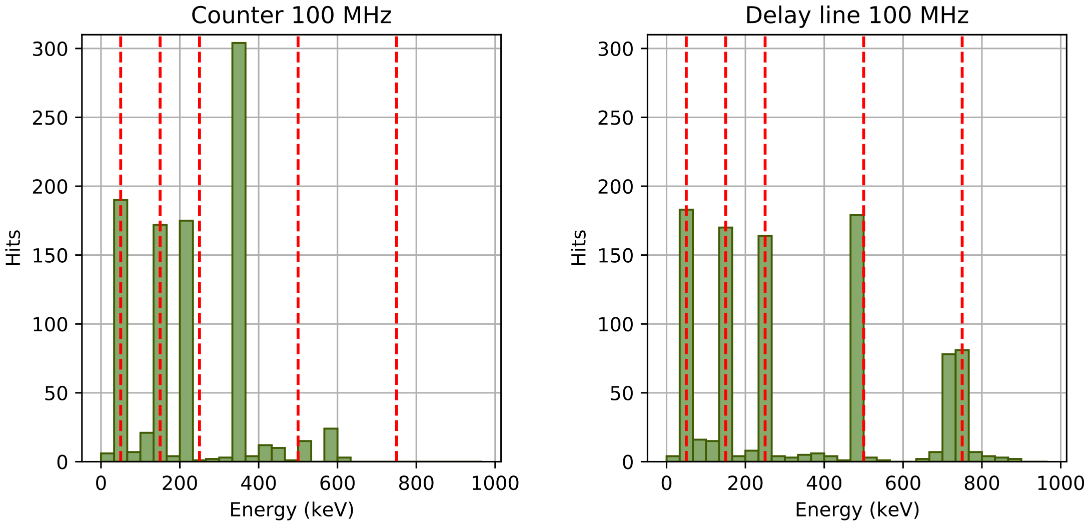


Figure 81: Left: Conversion of 50, 150, 250, 500 and 750 keV particles with a counter sampled at 100 MHz. Right: The same conversion with a delay line TDC.

A simulation was set up using an input spectrum that encompasses energies ranging from 30 keV to 1920 keV. This was done to obtain a picture of what the performance looks like in a real implementation with the actual expected bins. High energy particles are overrepresented in this simulation to not leave the high energy bins unpopulated. The particle hits are sorted into a histogram with the same bins as described in Table 3. The performance of the three TDCs implemented have been compared to the ROM at three different frequencies: 100 MHz, 200 MHz and 400 MHz. In each case, the counts expected by the ROM from the ToT generated by GATE is shown in a dashed red line, and the observed counts retrieved from the testbench are shown in green. The ideal TDC would match the count exactly to the ROM count, and the red dashed outline would match perfectly with the observed histogram. Thus, the performance measure of this exercise is how closely the histogram fits into the red dashed outline, and better resolution yields a better fit. All simulations have been run using exactly the same GATE particle data, and all runs have been implemented in a ROM that is fractioned into 16 fine counts for each coarse count. The resolution is expected to improve as the frequency increases, since this increases the number of sample points.

The results for the interpolationless counter is shown in Figure 82. Observed here is that the TDC generally does not have a sufficient resolution until the counter reaches a sampling frequency of 400 MHz, and even then some error is discernible.

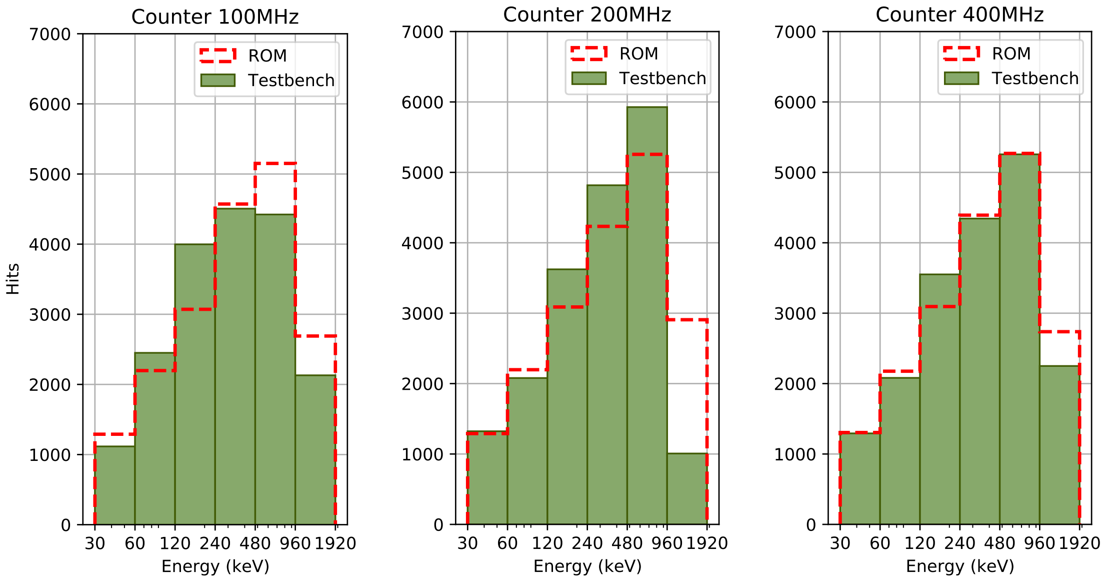


Figure 82: Accuracy of a pure counter TDC.

The next TDC under review is the asynchronous oversampler, and this is shown in Figure 83. It is implemented with four clocks and is expected to perform strictly better than the counter at all frequencies. It can be seen that this is indeed the case, but it still does not reach a sufficient resolution until 400 MHz.

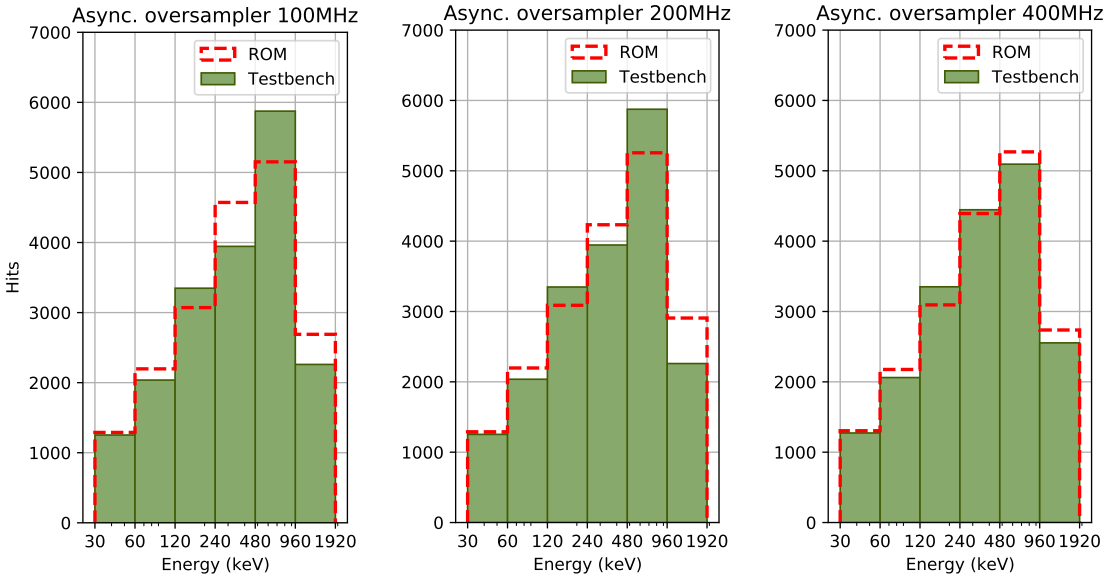


Figure 83: Accuracy of a 4x asynchronous oversampling TDC.

Finally, the multitapped delay line TDC is shown in Figure 84. This implementation stands out because the resolution is not derived from any clock, and comes from the propagation delay of the delay elements. Thus, the resolution is not expected to deviate much between the frequencies. It can be seen that this is the case, with a notable deviation observed for the 200 MHz simulation run. This can probably be attributed to some combination of frequency and delay line length that fits poorly with the conversion ROM, and is not indicative of the delay line performing worse at this frequency. The notable result here is that the delay line resolution most accurately recreates the expected shape of the histogram. The high resolution even at low frequencies is attractive as it makes the timing requirements of the circuit easier to meet.

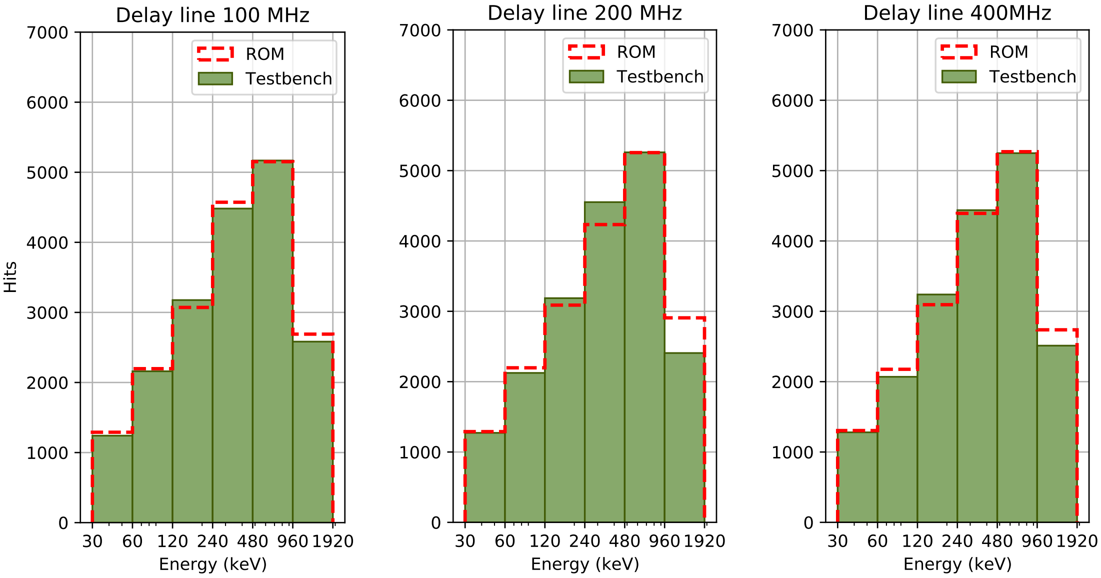


Figure 84: Accuracy of a delay line TDC.

The above analysis has only looked at the case of the ToT method and not included any dynamic threshold. This is in spite of the fact that in the course of the thesis, the dToT method was proposed, the circuit was included on the PCB, and also tested to be working as intended. The primary reason for this is that it is complicated to implement and model correctly, and ultimately it is another potential source of error. Therefore, it was deemed sensible to first determine whether it is needed, or if regular ToT is adequate. Furthermore, although the method can readily be modelled in the testbench, in a physical implementation it depends on being able to reliably create a delay in the FPGA that is exactly equal for all channels and consistent over time. A clock-based delay can decidely not be used, and this is apparent when considering that the generated delay then suffers from the same low resolution as the counter TDC. The variation in the threshold enable delay is manifested directly as a random variation in the Time-over-Threshold. The variation from a counter is on the order of nanoseconds, and completely negates the resolution of TDC. The input and output buffers of the FPGA can include calibrated delay lines of up to 10 ns, but this only enables the implementation of a 20 ns delay line. Some alternatives were investigated, but nothing definitive was concluded. The same delay-lines as implemented in the TDC may be used, but these must be manually calibrated to account for internal routing delays, and creates a problem of consistency across channels. Another alternative is possibly using the SERDES blocks to create the extra delay needed. Finally, if a new PCB iteration is done, a monostable vibrator solution can be included.

The discussion and implementation of dToT is left in the thesis as a suggestion to potential improvement that may be realized at a later point in the project, but as of now it is not feasible and best left disabled.

The firmware has been simulated at three select frequencies: 100 MHz, 200 MHz, and 400 MHz. Little mention has been made of the actual frequencies these TDC solutions can support. In general, it is expected that the counter will have no problem running at 400 MHz and possibly more, but the other two solutions are not likely to run at such a high frequency. The reason for this is that many registers have to be sampled in a single period to create the thermometer code. The firmware currently depends on combinatorial constructs that restrict the operating frequency of the two interpolating TDCs to somewhere between 100 and 150 MHz. Little effort has been made to pipeline the design, and the higher frequencies may still be unlocked by this. Fortunately, however, it has been demonstrated in thesis that the delay line TDC provides a high resolution alternative even at 100 MHz. The static timing analysis tool of NanoXmap currently provides no information about critical paths and their timing requirements, it only provides a maximum frequency that each clock can sustain. It is hard to determine exactly which paths are limiting the operating frequency. The main possible offenders for the system clock are the ones counters, the address count encoder and the calibration offset calculation, all of which can be pipelined to increase the operating frequency.

11 SUMMARY AND CONCLUSION

Ionizing particle radiation can be detected by a diode with a suitably large reverse bias. The pulse is amplified for detection and shaped to maximize the SNR. The amplitude of the pulse is proportional to the deposited charge and can be converted to determine the energy of the incident particle. Because of the regular pulse shape, the time the pulse is above a threshold can be used to uniquely determine the amplitude of the pulse in what is known as the Time-over-Threshold method. A DEEP detector readout unit was designed based on the Time-over-Threshold principle. Because the transfer function from time to energy is highly nonlinear, a dynamic threshold with the same time constant as the pulse shape time constant can be used to linearise the transfer function. The circuit contains the NG-MEDIUM FPGA to convert the time intervals to energies and record the particle hits into histograms. The PCB supports 8 channels in addition to a test channel and contains one comparator for each channel. It has three separate power lines to power the components on the circuit. In addition to this, it has a 150 V power line that serves as a high-voltage reverse bias for the pixel detector diodes.

Initial hardware testing of the PCB was conducted, and the detector reverse-bias was confirmed to have a ripple and noise that is sufficiently low to detect 30 keV particles, but this was not tested in practice. The comparator circuits were also confirmed. The connection of the board to the evaluation kit of the FPGA was tested and continuity was confirmed on all the pints. The onboard FPGA and supporting circuitry has not been tested.

In order to accurately quantize the time intervals, a Time-to-Digital converter is needed. A counter enabled by the input signal can determine the length of the pulse, but is ultimately limited by the clock frequency as it cannot discern transitions happening within a clock period. Two high-resolution TDC architectures were developed that can be used in conjunction with a counter to interpolate the clock period and improve accuracy. The first is a multitapped delay-line that consists of a chain of delay elements with equal propagation delay. As the signal transition propagates through the delay line, the output taps sequentially transition. On the rising edge of the clock, the taps are sampled producing a thermometer code. The second type of TDC is the asynchronous oversampler, where several clocks are generated that are equally off-phase from each other to constitute a full clock period. Each clock is used to sample registers in a grid to produce the output thermometer code. Instead of improving the resolution to a gate delay, it divides the clock period into equal parts. The multitapped delay-line TDC has the highest accuracy and lowest power consumption, but the benefit of the asynchronous oversampling is the ease of implementation and low resource cost.

Previous work on Monte Carlo simulations of the sensor geometry was evaluated, and although a proposed solution of vertical coincidence checking at first seemed promising, it was concluded that this method reduces detector efficiency by an unacceptable amount. Furthermore, the rate of particles penetrating the detector house and contaminating the pitch angle distribution measurement was investigated, and found to be negligible. A method of discering which pixel the particle struck first was not found, and the scattering electrons are recommended to be collected in a separate histogram.

A complete firmware was developed for the readout, binning and transmission of particle hit data. A Python script was made that generates the ROM contents based on the simulated ToT interval of a range of energies. A packet and message structure for the telemetry was proposed for the histograms and the raw data. Two high-resolution TDCs were developed and simulated.

A framework was developed for feeding particle hits from the GATE simulation into the VHDL testbench. This was further used to investigate the performance of the developed TDCs. The delay-line TDC was concluded to have a better resolution than the counter and asynchronous oversampler even when clocked at much lower frequencies. Consequently, the power consumption is also much lower with this implementation. The delay-line TDC can be implemented in a single tile, and can therefore likely be scaled to the planned 32-channel instrument in the future without immediate problems.

11.1 Conclusion

Through the work in this thesis, some important contributions to the DEEP project has been made. A lot of information about the previous work of the project was available at the start of the thesis, but little of it was collected in a concise work. One of the main contributions of this thesis is thus to form a cohesive overview of all the parts of the system, from first principles to implementation. Secondly, a new PCB was designed that seems through testing to have alleviated the noise problem that was observed on the reverse bias for the pixel detectors. It also enabled the use of a new and promising FPGA and a method of using the PCB as an expansion card to the FPGA evaluation kit.

A few key contributions were also made to the data acquisition strategy. Although the previous iteration of the PCB also implemented a Time-over-Threshold based data acquisition, the idea of implementing a TDC to improve resolution was first formulated in this thesis. It was also demonstrated in this thesis that such an implementation is in fact needed to accurately determine the flux energy distribution of precipitating particles. The dToT method was also proposed and partially implemented, and remains a viable strategy for the future. A new firmware design was made that included data acquisition, binning, telemetry communication and UART functionality. Although the design is synthesizable and meets basic timing requirements, improvements in this area are both possible and recommended by redesigning certain circuit elements to include pipelining.

The Monte Carlo simulations were useful in determining an accurate algorithm for coincidence events, and the GATE/VHDL testbench framework was useful in validating the work performed in this thesis and will likely have future applications in the testbench. Further contributions were made in several scripts useful for developing for the NG-MEDIUM FPGA going forward, such as working examples of applying IO, timing and placement constraints, and generation of a memory table and the subsequent linking to physical memory in the FPGA.

11.1.1 Future work

The work of this thesis has mainly been design oriented, producing a prototype of the PCB and the FPGA firmware. Though bring-up of the PCB was performed and a testbench produced for the FPGA firmware, this is by no means exhaustive. Unfortunately, my year with the DEEP particle detector has come to an end, but there is still a lot of work to be done and improvements to make. Though the PCB was based on a previous revision, a lot of changes were made, and the layout is completely new. A comprehensive test of the PCB must be done and potential errors of the design rectified. The firmware needs a more extensive testbench that can verify all the modules in each mode of operation. The firmware must be tested on hardware, and particularly the calibration bins of the delay line should be compared with the back-annotated calibration bins. When all of this is in place, the energy transfer function should be calibrated with a radiation source to ensure accurate conversions of time intervals. When everything is working in headless mode on the development kit, the FPGA must be mounted to the PCB. This is a stage that can introduce new errors, because at this point the FPGA relies on the onboard supporting circuitry that previously was located on the evaluation kit. Once the board and FPGA configuration are verified, and the conversion ROM is properly calibrated, the detector is hopefully ready to fly.

References

Norwegian launch record. https://www.sworld.com.au/steven/space/norway-rec.txt, 2020.

Project vegas project summary. http://vegas-h2020.eu/index.html, 2020.

Ultrafast 7 ns Single Supply Comparator. Analog Devices, 2014. URL https://www.analog.com/media/en/technical-documentation/data-sheets/AD8561.pdf. Rev D.

H. Andersen. Design and simulation of pixel layout and data processing algorithms for the deep instrument. Master's thesis, University of Bergen, 2018.

Questionaire concerning experiment data for payload construction and design for ICI-5. Andøya Space Centre, 2018.

A. Balla, M. Mario Beretta, P. Ciambrone, M. Gatta, F. Gonnella, L. Iafolla, M. Mascolo, R. Messi, D. Moricciani, and D. Riondino. Low resource FPGA-based time to digital converter. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 739:75–82, 2014. ISSN 0168-9002. doi: https://doi.org/10.1016/j.nima.2013.12.033. URL https://www.sciencedirect.com/science/article/pii/S0168900213017245.

J. Bhasker. *Static timing analysis for nanometer designs : a practical approach*. Springer, New York, 2009. ISBN 9780387938202.

D. Brooks. *Signal integrity issues and printed circuit board design*. Prentice Hall, Upper Saddle River, NJ, 2003. ISBN 0-13141-884-X.

P. Carra, M. Bertazzoni, M. G. Bisogni, J. M. Cela Ruiz, A. Del Guerra, D. Gascon, S. Gomez, M. Morrocchi, G. Pazzi, D. Sanchez, I. Sarasola Martin, G. Sportelli, and N. Belcari. Auto-calibrating tdc for an soc-fpga data acquisition system. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 3(5):549–556, 2019. doi: 10.1109/TRPMS.2018.2882709.

J. Devore. *Modern mathematical statistics with applications*. Springer, New York, NY, 2012. ISBN 978-1-4614-0390-6.

R. Ghaffarian and J. Evans. Enabling more than moore: Accelerated reliability testing and risk analysis for advanced electronics packaging. 2014.

J. K. Hargreaves. *The solar-terrestrial environment: An introduction to geospace - the science of the terrestrial upper atmosphere, ionosphere, and magnetosphere*. Cambridge University Press, Cambridge England New York, NY, USA, 1992. ISBN 0-521-32748-2.

R. Hartley. Controlling Radiated EMI Through PCB Stack-up. *Printed Circuit Design Magazine*, pages 16–23, 2000.

S. Henzler. *Time-to-digital converters*. Springer, Dordrecht London, 2010. ISBN 978-90-481-8627-3.

Z. Jaworski. Choosing the optimal hdl model of thermometer-to-binary encoder. pages 297–300, 2015. doi: 10.1109/MIXDES.2015.7208530.

I. Kipnis, T. Collins, J. DeWitt, S. Dow, A. Frey, A. Grillo, R. Johnson, W. Kroeger, A. Leona, L. Luo, E. Mandelli, P. Manfredi, M. Melani, M. Momayezi, F. Morsani, M. Nyman, M. Pedrali-Noy, P. Poplevin, E. Spencer, V. Re, and N. Roe. A time-over-threshold machine: the readout integrated circuit for the babar silicon vertex tracker. *IEEE Transactions on Nuclear Science*, 44(3):289–297, 1997. doi: 10.1109/23.603658.

G. Knoll. *Radiation detection and measurement.* John Wiley, Hoboken, N.J, 2010. ISBN 978-0-470-13148-0.

M. M. Lam, R. B. Horne, N. P. Meredith, S. A. Glauert, T. Moffat-Griffin, and J. C. Green. Origin of energetic electron precipitation >30 keV into the atmosphere. *Journal of Geophysical Research A: Space Physics*, 115(A4):1–15, 2010. ISSN 21699402. doi: 10.1029/2009JA014619.

G. Lindahl. *Telemetrienkoder til bruk i vitenskapelige sonderaketter.* Andøya Space Centre, 2021.

H. Lynnebakken. Forskningsraketten ICI-5 fløy fra Svalbard. https://titan.uio.no/2019/forskningsraketten-ici-5-floy-fra-svalbard, 2019. [Retrieved 20/09/21.].

K. Maragos, V. Leon, G. Lentaris, D. Soudris, D. Gonzalez-Arjona, R. Domingo, A. Pastor, D. M. Codinachs, and I. Conway. Evaluation methodology and reconfiguration tests on the new european ng-medium fpga. pages 127–134, 2018. doi: 10.1109/AHS.2018.8541492.

ModelSim® Command Reference Manual. Mentor Graphics, 2015.

NG-MEDIUM NX1H35AS Configuration user's guide. NanoXplore, 2020a. Rev 1.2.

NG-MEDIUM NX1H35AS Cookbook. NanoXplore, 2020b. Rev 1.5.

NG-MEDIUM NX1H35AS Datasheet. NanoXplore, 2020c. Rev 2.2.

NXmap2 Library Guide. NanoXplore, 2020d. Rev 1.6.

NXMAP User Manual. NanoXplore, 2020e. Rev 3.0.

H. Nesse Tyssøy, M. Sandanger, L.-K. Ødegaard, J. Stadsnes, A. Aasnes, and A. Zawedde. Energetic electron precipitation into the middle atmosphere - constructing the loss cone fluxes from meped poes: Energetic electron loss cone fluxes. *Journal of Geophysical Research: Space Physics*, 121, 05 2016. doi: 10.1002/2016JA022752.

H. Nesse Tyssøy, J. Stadsnes, F. Søraas, K. Ullaland, and D. Röhrich. *Distribution of Energetic Electron and Proton (DEEP) Instrument - Conceptual design report.* Birkeland Centre for Space Science, 2017.

H. Nesse Tyssøy, A. Haderlein, M. I. Sandanger, and J. Stadsnes. Intercomparison of the poes/meped loss cone electron fluxes with the cmip6 parametrization. *Journal of Geophysical Research: Space Physics*, 124(1):628–642, 2019. doi: https://doi.org/10.1029/2018JA025745. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018JA025745.

T. Orita, K. Shimazoe, and H. Takahashi. The dynamic time-over-threshold method for multi-channel APD based gamma-ray detectors. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 775:154–161, 2015. ISSN 01689002. doi: 10.1016/j.nima.2014.12.014. URL http://dx.doi.org/10.1016/j.nima.2014.12.014.

Output Ripple Measurement Methods for DC-DC Converters. Richtek, 2016. URL https://www.richtek.com/Home/Design%20Support/~/media/DT_PDF/Ripple_measurement_tips.pdf.

C. J. Rodger, A. J. Kavanagh, M. A. Clilverd, and S. R. Marple. Comparison between poes energetic electron precipitation observations and riometer absorptions: Implications for determining true precipitation fluxes. *Journal of Geophysical Research: Space Physics*, 118(12):7810–7821, 2013. doi: https://doi.org/10.1002/2013JA019439.

H. Spieler. *Semiconductor detector systems.* Oxford University Press, Oxford New York, 2005. ISBN 978-01-915-2365-6.

Rad hard 65nm CMOS technology platform for space applications. STMicroelectronics, 2015. URL https://www.st.com/resource/en/data_brief/c65space.pdf.

S. Tancock, E. Arabul, and N. Dahnoun. A review of new time-to-digital conversion techniques. *IEEE Transactions on Instrumentation and Measurement*, 68(10):3406–3417, 2019. doi: 10.1109/TIM.2019.2936717.

N. Thomopoulos. *Essentials of Monte Carlo simulation : statistical methods for building simulation models.* Springer, New York, 2013. ISBN 9781461460220.

A. Tontini, L. Gasparini, L. Pancheri, and R. Passerone. Design and characterization of a low-cost fpga-based tdc. *IEEE Transactions on Nuclear Science*, 65(2):680–690, 2018. doi: 10.1109/TNS.2018.2790703.

T. Tran. *High-speed DSP and analog system design.* Springer, New York London, 2010. ISBN 978-1-4419-6308-6.

J. A. Van Allen and L. A. Frank. Radiation around the earth to a radial distance of 107,400 km. *Nature*, 183, 2 1959. doi: 10.1038/183430a0. URL https://www.osti.gov/biblio/4292622.

D. K. Xie, Q. C. Zhang, G. S. Qi, and D. Y. Xu. Cascading delay line time-to-digital converter with 75 ps resolution and a reduced number of delay cells. *Review of Scientific Instruments*, 76(1):014701, 2005. doi: 10.1063/1.1829931.

Q Series datasheet. XP-POWER, 2020. URL https://www.xppower.com/portals/0/pdfs/SF_Q_Series.pdf.

L.-K. G. Ødegaard. *Energetic particle precipitation into the middle atmosphere - optimization and applications of the NOAA POES MEPED data.* PhD thesis, University of Bergen, 2016.

Appendices

A PINOUT

Name	Location	Standard	Drive
usr_rst_n	IO_B08D01N	LVCMOS	4mA
clk_40	IO_B08D01P	LVCMOS	4mA
calib_clk	IO_B08D02N	LVCMOS	4mA
tx1	IO_B6D10N	LVCMOS	4mA
rx1	IO_B6D10P	LVCMOS	4mA
tx2	IO_B6D11N	LVCMOS	4mA
rx2	IO_B6D11P	LVCMOS	4mA
comp_in[1]	IO_B9D14P/IO_B9D14N	LVCMOS	4mA
comp_in[2]	IO_B9D15P/IO_B9D15N	LVCMOS	4mA
comp_in[3]	IO_B9D05P/IO_B9D05N	LVCMOS	4mA
comp_in[4]	IO_B9D08P/IO_B9D08N	LVCMOS	4mA
comp_in[5]	IO_B9D09P/IO_B9D09N	LVCMOS	4mA
comp_in[6]	IO_B9D01P/IO_B9D01N	LVCMOS	4mA
comp_in[7]	IO_B9D06P/IO_B9D06N	LVCMOS	4mA
comp_in[8]	IO_B9D04P/IO_B9D04N	LVCMOS	4mA
dtot[0]	IO_B6D05P	LVCMOS	4mA
dtot[1]	IO_B8D02P	LVCMOS	4mA
dtot[2]	IO_B8D03P	LVCMOS	4mA
dtot[3]	IO_B8D04P	LVCMOS	4mA
dtot[4]	IO_B8D05P	LVCMOS	4mA
dtot[5]	IO_B8D06P	LVCMOS	4mA
dtot[6]	IO_B8D07P	LVCMOS	4mA
dtot[7]	IO_B8D08P	LVCMOS	4mA
dtot[8]	IO_B8D10P	LVCMOS	4mA
pcm_data	IO_B6D12P	LVCMOS	4mA
pcm_timer	IO_B6D07N	LVCMOS	4mA
pcm_gse_ctrl	IO_B6D07P	LVCMOS	4mA
pcm_maj_frame	IO_B6D08N	LVCMOS	4mA
pcm_min_frame	IO_B6D15N	LVCMOS	4mA
pcm_gate	IO_B6D15P	LVCMOS	4mA
pcm_sclk	IO_B6D08P	LVCMOS	4mA
sensor_bias_en	IO_B6D12N	LVCMOS	4mA

B REGISTER MAP

Table 8: Register map part 1.

Name	Address	Access	Width	Default	Description
DEEP_SBI_HEARTBEAT	0x00	RW	8	0x0	Heartbeat enable
DEEP_SBI_SYSTEM_CONTROL	0x02	RO	8	0x0	System control
DEEP_SBI_TIMER	0x04	RO	8	0x0	Timer
DEEP_SBI_PIXEL0_HIST0	0x06	RO	16	0x0	Pixel 0 histogram bin 0
DEEP_SBI_PIXEL0_HIST1	0x08	RO	16	0x0	Pixel 0 histogram bin 1
DEEP_SBI_PIXEL0_HIST2	0x0A	RO	16	0x0	Pixel 0 histogram bin 2
DEEP_SBI_PIXEL0_HIST3	0x0C	RO	16	0x0	Pixel 0 histogram bin 3
DEEP_SBI_PIXEL0_HIST4	0x0E	RO	16	0x0	Pixel 0 histogram bin 4
DEEP_SBI_PIXEL0_HIST5	0x10	RO	16	0x0	Pixel 0 histogram bin 5
DEEP_SBI_PIXEL0_HIST6	0x12	RO	16	0x0	Pixel 0 histogram bin 6
DEEP_SBI_PIXEL1_HIST0	0x14	RO	16	0x0	Pixel 1 histogram bin 0
DEEP_SBI_PIXEL1_HIST1	0x16	RO	16	0x0	Pixel 1 histogram bin 1
DEEP_SBI_PIXEL1_HIST2	0x18	RO	16	0x0	Pixel 1 histogram bin 2
DEEP_SBI_PIXEL1_HIST3	0x1A	RO	16	0x0	Pixel 1 histogram bin 3
DEEP_SBI_PIXEL1_HIST4	0x1C	RO	16	0x0	Pixel 1 histogram bin 4
DEEP_SBI_PIXEL1_HIST5	0x1E	RO	16	0x0	Pixel 1 histogram bin 5
DEEP_SBI_PIXEL1_HIST6	0x20	RO	16	0x0	Pixel 1 histogram bin 6
DEEP_SBI_PIXEL2_HIST0	0x22	RO	16	0x0	Pixel 2 histogram bin 0
DEEP_SBI_PIXEL2_HIST1	0x24	RO	16	0x0	Pixel 2 histogram bin 1
DEEP_SBI_PIXEL2_HIST2	0x26	RO	16	0x0	Pixel 2 histogram bin 2
DEEP_SBI_PIXEL2_HIST3	0x28	RO	16	0x0	Pixel 2 histogram bin 3
DEEP_SBI_PIXEL2_HIST4	0x2A	RO	16	0x0	Pixel 2 histogram bin 4
DEEP_SBI_PIXEL2_HIST5	0x2C	RO	16	0x0	Pixel 2 histogram bin 5
DEEP_SBI_PIXEL2_HIST6	0x2E	RO	16	0x0	Pixel 2 histogram bin 6
DEEP_SBI_PIXEL3_HIST0	0x30	RO	16	0x0	Pixel 3 histogram bin 0
DEEP_SBI_PIXEL3_HIST1	0x32	RO	16	0x0	Pixel 3 histogram bin 1
DEEP_SBI_PIXEL3_HIST2	0x34	RO	16	0x0	Pixel 3 histogram bin 2
DEEP_SBI_PIXEL3_HIST3	0x36	RO	16	0x0	Pixel 3 histogram bin 3
DEEP_SBI_PIXEL3_HIST4	0x38	RO	16	0x0	Pixel 3 histogram bin 4
DEEP_SBI_PIXEL3_HIST5	0x3A	RO	16	0x0	Pixel 3 histogram bin 5
DEEP_SBI_PIXEL3_HIST6	0x3C	RO	16	0x0	Pixel 3 histogram bin 6

Table 9: Register map part 2.

Name	Address	Access	Width	Default	Description
DEEP_SBI_PIXEL4_HIST0	0x3E	RO	16	0x0	Pixel 4 histogram bin 0
DEEP_SBI_PIXEL4_HIST1	0x40	RO	16	0x0	Pixel 4 histogram bin 1
DEEP_SBI_PIXEL4_HIST2	0x42	RO	16	0x0	Pixel 4 histogram bin 2
DEEP_SBI_PIXEL4_HIST3	0x44	RO	16	0x0	Pixel 4 histogram bin 3
DEEP_SBI_PIXEL4_HIST4	0x46	RO	16	0x0	Pixel 4 histogram bin 4
DEEP_SBI_PIXEL4_HIST5	0x48	RO	16	0x0	Pixel 4 histogram bin 5
DEEP_SBI_PIXEL4_HIST6	0x4A	RO	16	0x0	Pixel 4 histogram bin 6
DEEP_SBI_COINC_HIST0	0x4C	RO	16	0x0	Pixel 5 histogram bin 0
DEEP_SBI_COINC_HIST1	0x4E	RO	16	0x0	Pixel 5 histogram bin 1
DEEP_SBI_COINC_HIST2	0x50	RO	16	0x0	Pixel 5 histogram bin 2
DEEP_SBI_COINC_HIST3	0x52	RO	16	0x0	Pixel 5 histogram bin 3
DEEP_SBI_COINC_HIST4	0x54	RO	16	0x0	Pixel 5 histogram bin 4
DEEP_SBI_COINC_HIST5	0x56	RO	16	0x0	Pixel 5 histogram bin 5
DEEP_SBI_COINC_HIST6	0x58	RO	16	0x0	Pixel 5 histogram bin 6

C CODE

C.1 NX_CY delay line

```
  -- There are 24 carry units in each tile, need to generate in chunks of
   24 otherwise routing fails
  long_carry_chain_gen :
  for tile in 0 to c_num_carry_logic_modules-1 generate
  begin
    -- Carry in can only be routed from a previous carry out, use the first
     adder as a carry generator
    i_gen_nx_cy : entity nx.NX_CY
      generic map(add_carry => 0) --set CI to 0
      port map(A1 => sum(tile*24), A2 => sum(tile*24), A3 => sum(tile*24),
   A4 => sum(tile*24),
               B1 => sum(tile*24), B2 => sum(tile*24), B3 => sum(tile*24),
   B4 => sum(tile*24),
               S1 => sum(tile*24 + 1),
               CI => ’0’, CO => carry_chain(tile*24)
               );

    carry_chain_i_gen :
    for cy in 1 to 22 generate
    begin
      i_nx_cy : entity nx.NX_CY
        generic map(add_carry => 2)     --propagate CI as carry in
        port map(A1 => ’0’, A2 => ’0’, A3 => ’0’, A4 => ’0’,
                 B1 => ’1’, B2 => ’1’, B3 => ’1’, B4 => ’1’,
                 -- all sums are ready at the same time in NX_CY carry
   lookahead, only use S1
                 S1 => sum(tile*24 + cy + 1),
                 CI => carry_chain(tile*24 + cy - 1), CO => carry_chain(
   tile*24 + cy)
                 );
    end generate;

      i_arbitrate_nx_cy : entity nx.NX_CY
        generic map(add_carry => 2)       --propagate CI as carry in
        port map(A1 => ’0’, A2 => ’0’, A3 => ’0’, A4 => ’0’,
                 B1 => ’1’, B2 => ’1’, B3 => ’1’, B4 => ’1’,
                 S1 => sum((tile+1)*24),
                 CI => carry_chain((tile+1)*24 - 2)
                 );
  end generate;

-- feed input into first sum
  sum(0) <= tdc_in;
```

C.2 Asynchronous oversampler

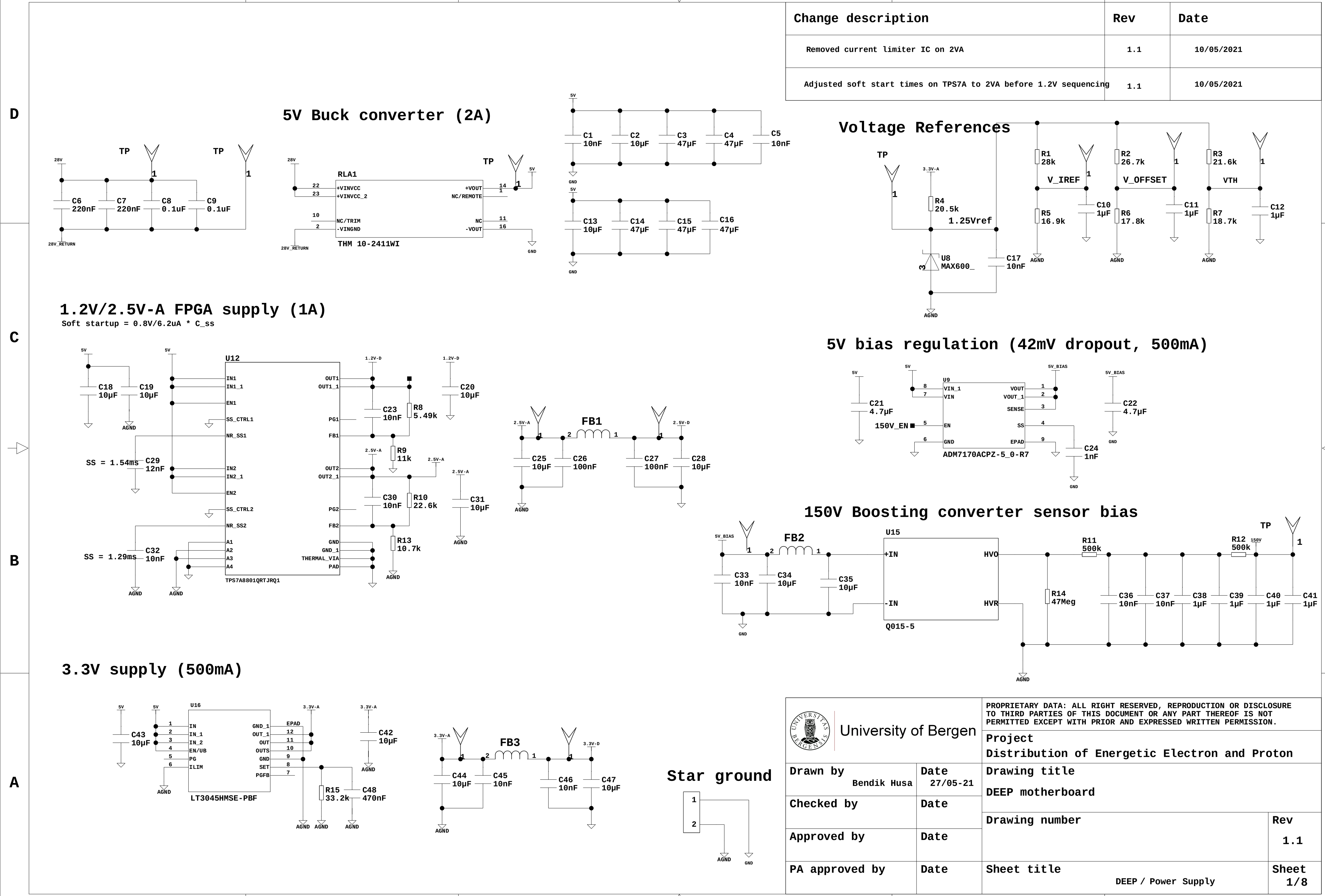
```vhdl
-- Generate register for each clock in matrix
  i_gen_rows : for row in 0 to c_num_clocks-1 generate
    i_gen_cols : for col in 0 to c_num_clocks-1 generate
      p_oversampling : process (clk_matrix(col)(row))
      begin
        if rising_edge(clk_matrix(col)(row)) then
          sample_matrix(col+1)(row) <= sample_matrix(col)(row);
        end if;
      end process p_oversampling;
    end generate;
  end generate;

-- Assign the correct clock to each register
  p_construct_clk_matrix : process (all)
  begin
    for row in 0 to c_num_clocks-1 loop
      for col in 0 to c_num_clocks-1 loop
        if (row < col) then
          clk_matrix(col)(row) <= clk_array(0);
        else
          clk_matrix(col)(row) <= clk_array(row - col);
        end if;
      end loop;
    end loop;
  end process p_construct_clk_matrix;

  sample_matrix(0) <= (others => tdc_in);
  tcode            <= sample_matrix(c_num_clocks);
```

D SCHEMATIC

Change description	Rev	Date
Removed current limiter IC on 2VA	1.1	10/05/2021
Adjusted soft start times on TPS7A to 2VA before 1.2V sequencing	1.1	10/05/2021

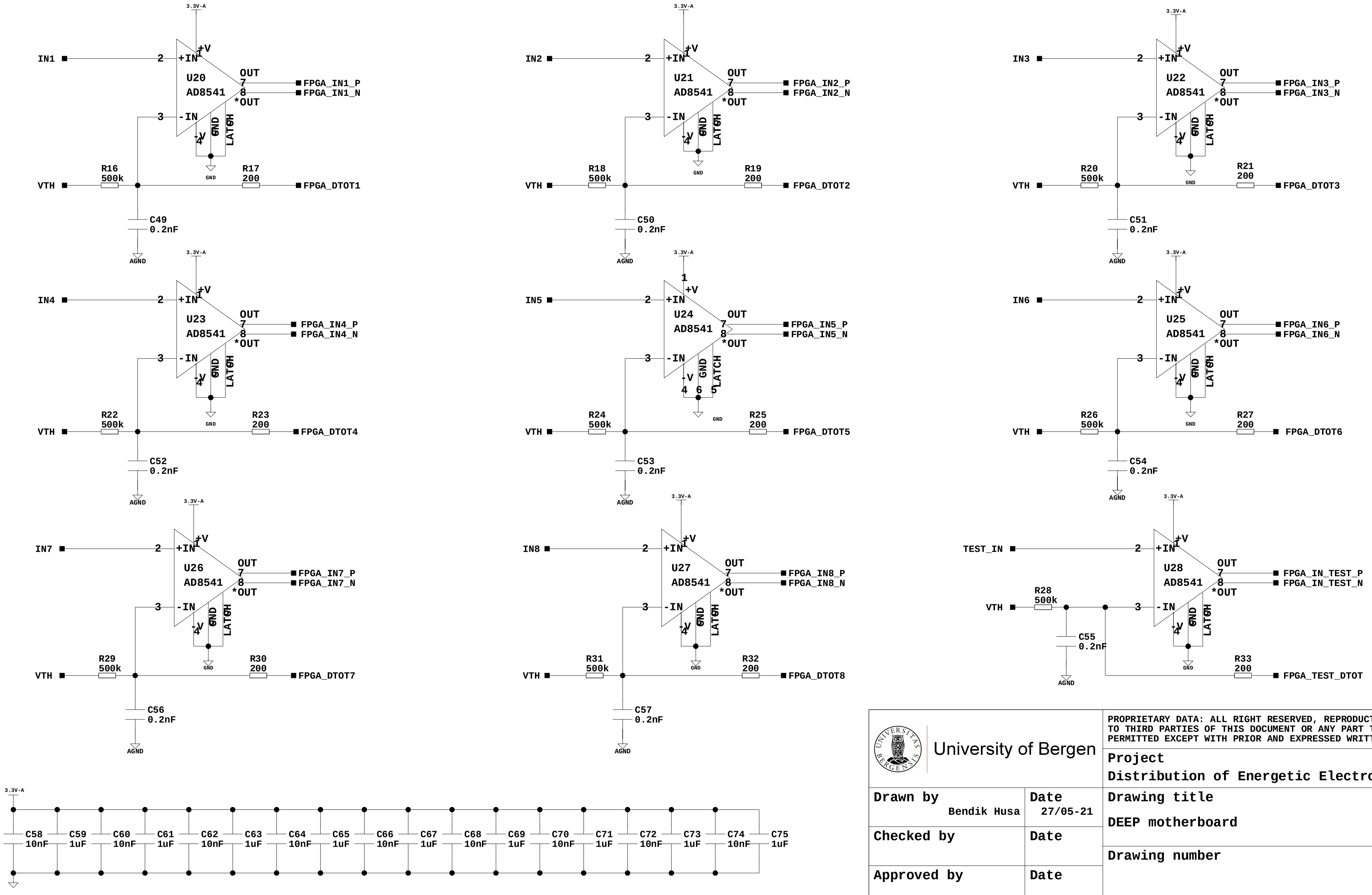


University of Bergen

PROPRIETARY DATA: ALL RIGHT RESERVED, REPRODUCTION OR DISCLOSURE TO THIRD PARTIES OF THIS DOCUMENT OR ANY PART THEREOF IS NOT PERMITTED EXCEPT WITH PRIOR AND EXPRESSED WRITTEN PERMISSION.

Drawn by Bendik Husa	Date 27/05-21	Project Distribution of Energetic Electron and Proton	
Checked by	Date	Drawing title DEEP motherboard	
Approved by	Date	Drawing number	Rev 1.1
PA approved by	Date	Sheet title DEEP / Power Supply	Sheet 1/8

Change description	Rev	Date
Changed from single-ended to differential output, moved to IO bank 91.1		08/05/2021



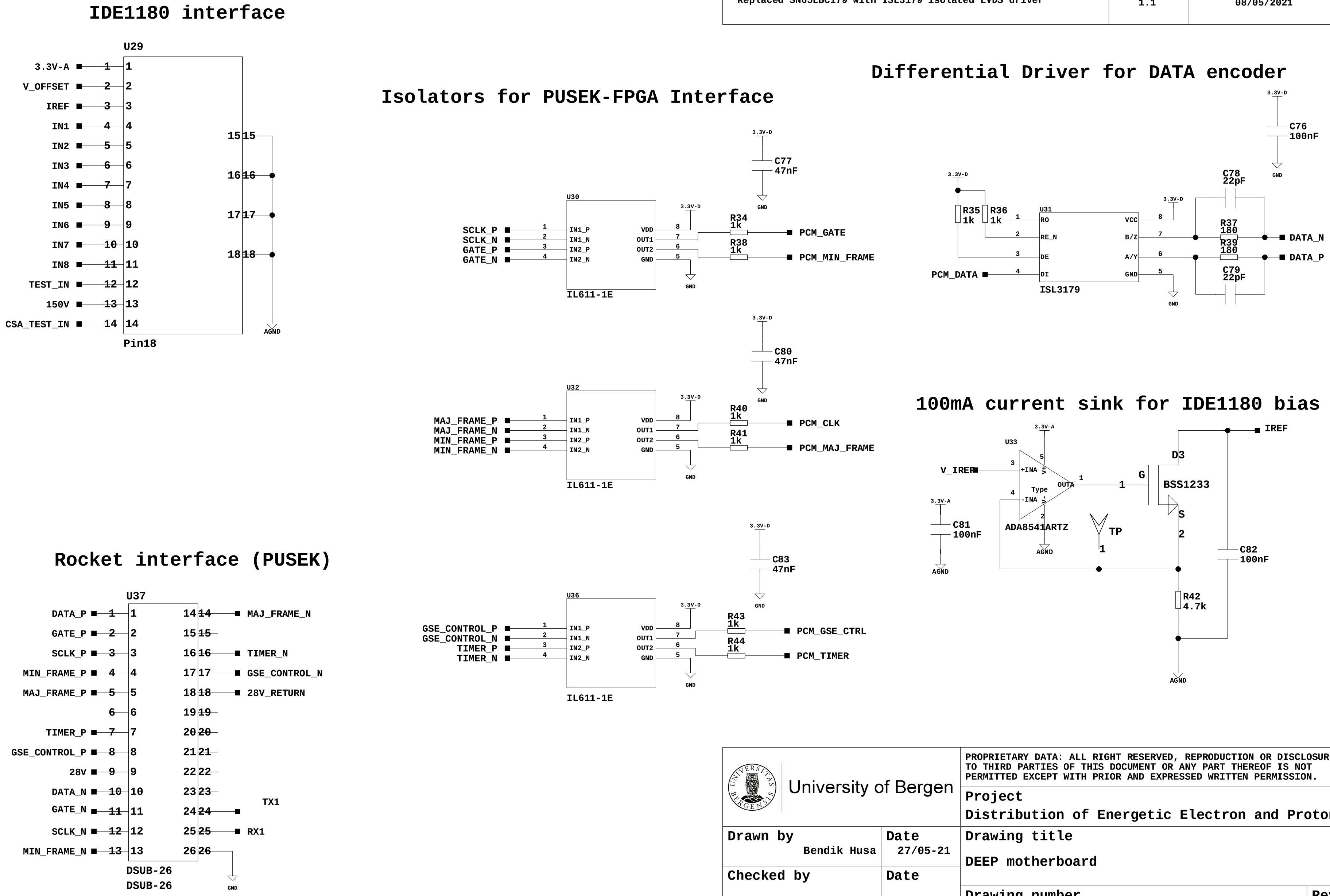
University of Bergen

PROPRIETARY DATA: ALL RIGHT RESERVED, REPRODUCTION OR DISCLOSURE TO THIRD PARTIES OF THIS DOCUMENT OR ANY PART THEREOF IS NOT PERMITTED EXCEPT WITH PRIOR AND EXPRESSED WRITTEN PERMISSION.

Project
Distribution of Energetic Electron and Proton

Drawn by Bendik Husa	Date 27/05-21	Drawing title DEEP motherboard	
Checked by	Date	Drawing number	Rev 1.1
Approved by	Date		
PA approved by	Date	Sheet title DEEP / Comparators	Sheet 2/8

Change description	Rev	Date
Replaced HCPL0631 optocouplers with IL611 isolators in accordance with ASC recommendations	1.1	08/05/2021
Replaced SN65LBC179 with ISL3179 isolated LVDS driver	1.1	08/05/2021

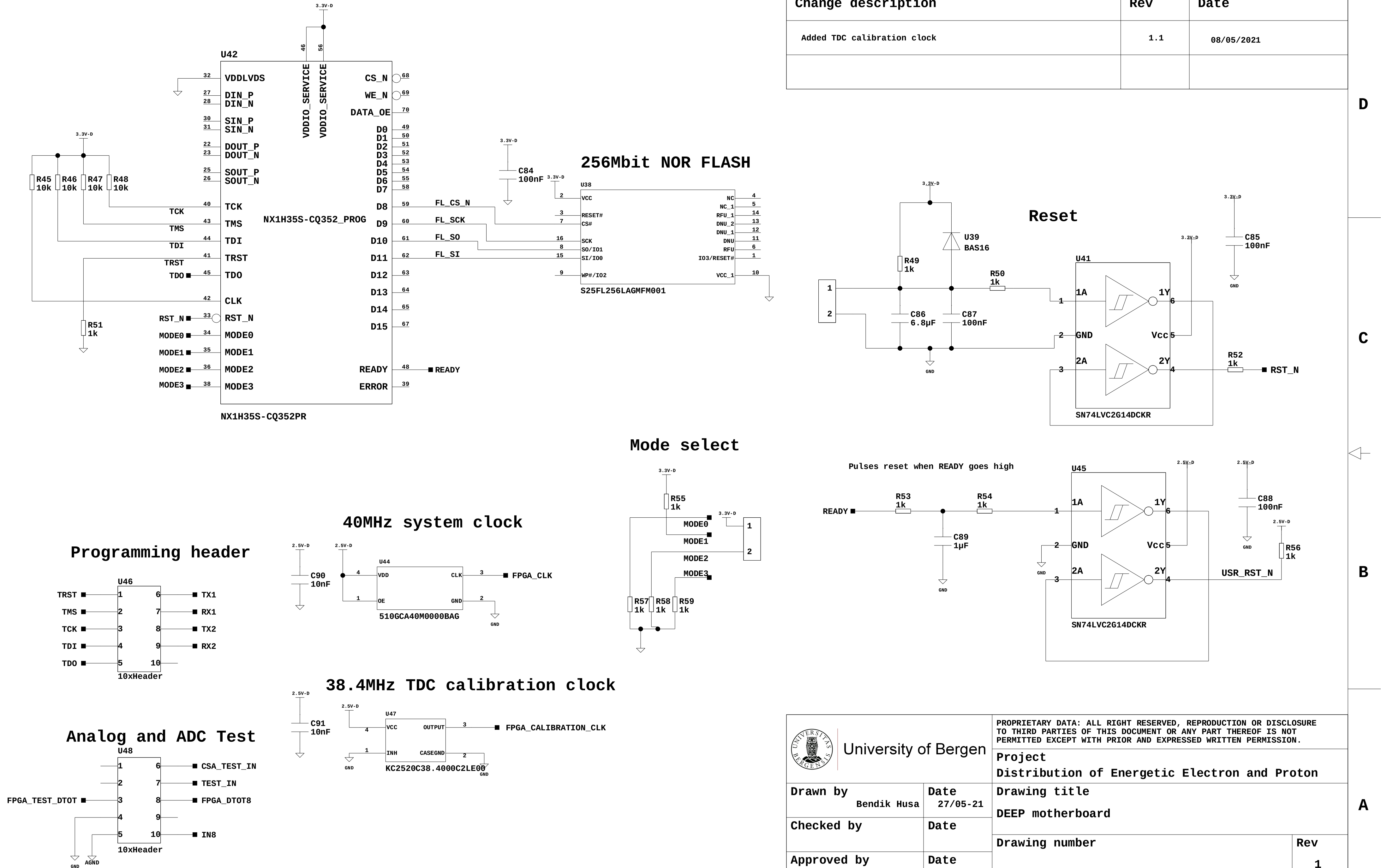


University of Bergen

PROPRIETARY DATA: ALL RIGHT RESERVED, REPRODUCTION OR DISCLOSURE TO THIRD PARTIES OF THIS DOCUMENT OR ANY PART THEREOF IS NOT PERMITTED EXCEPT WITH PRIOR AND EXPRESSED WRITTEN PERMISSION.

Drawn by Bendik Husa	Date 27/05-21	Project Distribution of Energetic Electron and Proton	
		Drawing title DEEP motherboard	
Checked by	Date		
Approved by	Date	Drawing number	Rev 1.1
PA approved by	Date	Sheet title DEEP / PCM interface	Sheet 3/8

Change description	Rev	Date
Added TDC calibration clock	1.1	08/05/2021

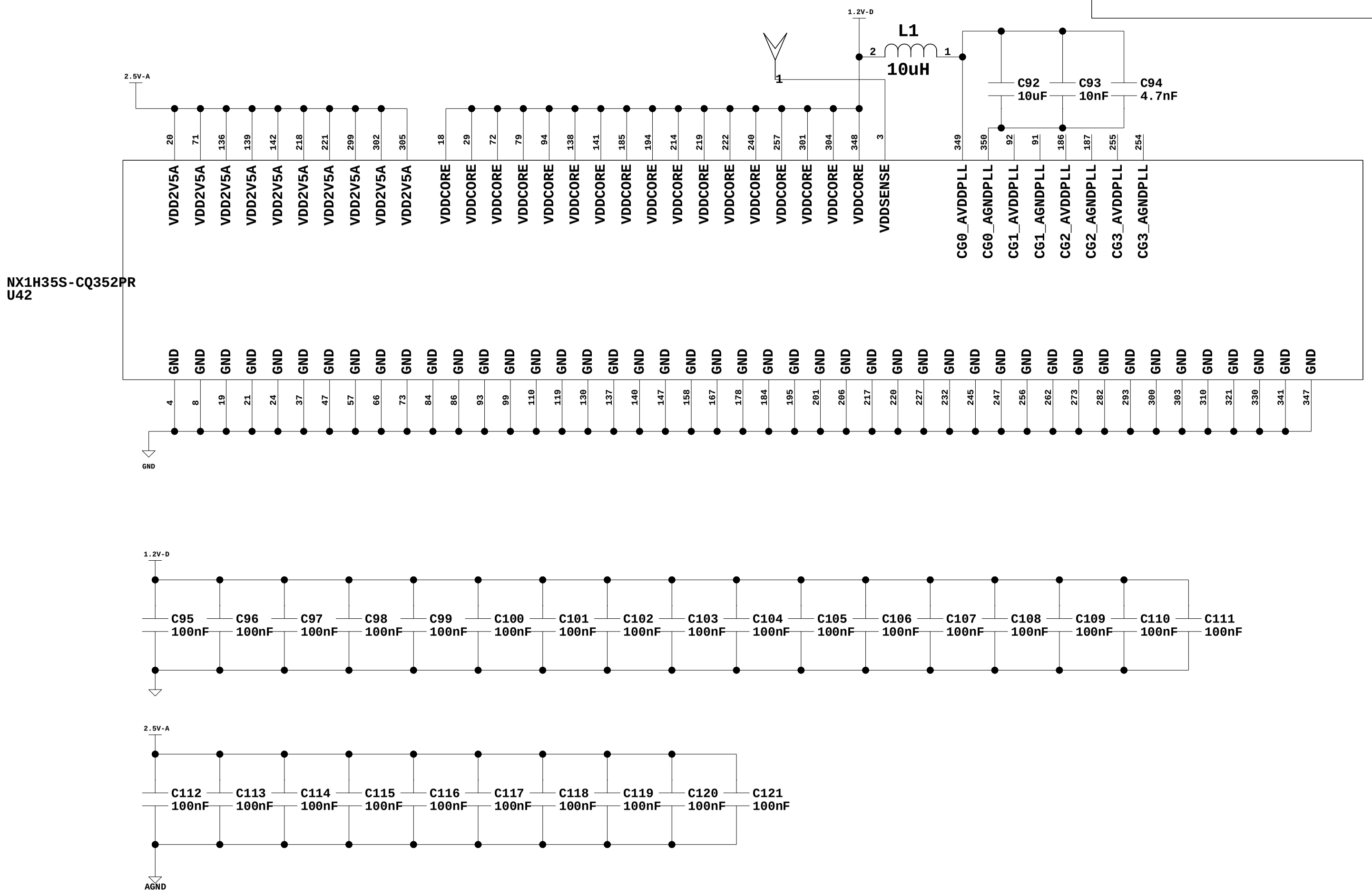


PROPRIETARY DATA: ALL RIGHT RESERVED, REPRODUCTION OR DISCLOSURE TO THIRD PARTIES OF THIS DOCUMENT OR ANY PART THEREOF IS NOT PERMITTED EXCEPT WITH PRIOR AND EXPRESSED WRITTEN PERMISSION.

University of Bergen

Drawn by Bendik Husa	Date 27/05-21	Project Distribution of Energetic Electron and Proton	
Checked by	Date	Drawing title DEEP motherboard	
Approved by	Date	Drawing number	Rev 1
PA approved by	Date	Sheet title DEEP / FPGA prog	Sheet 4/8

Change description	Rev	Date

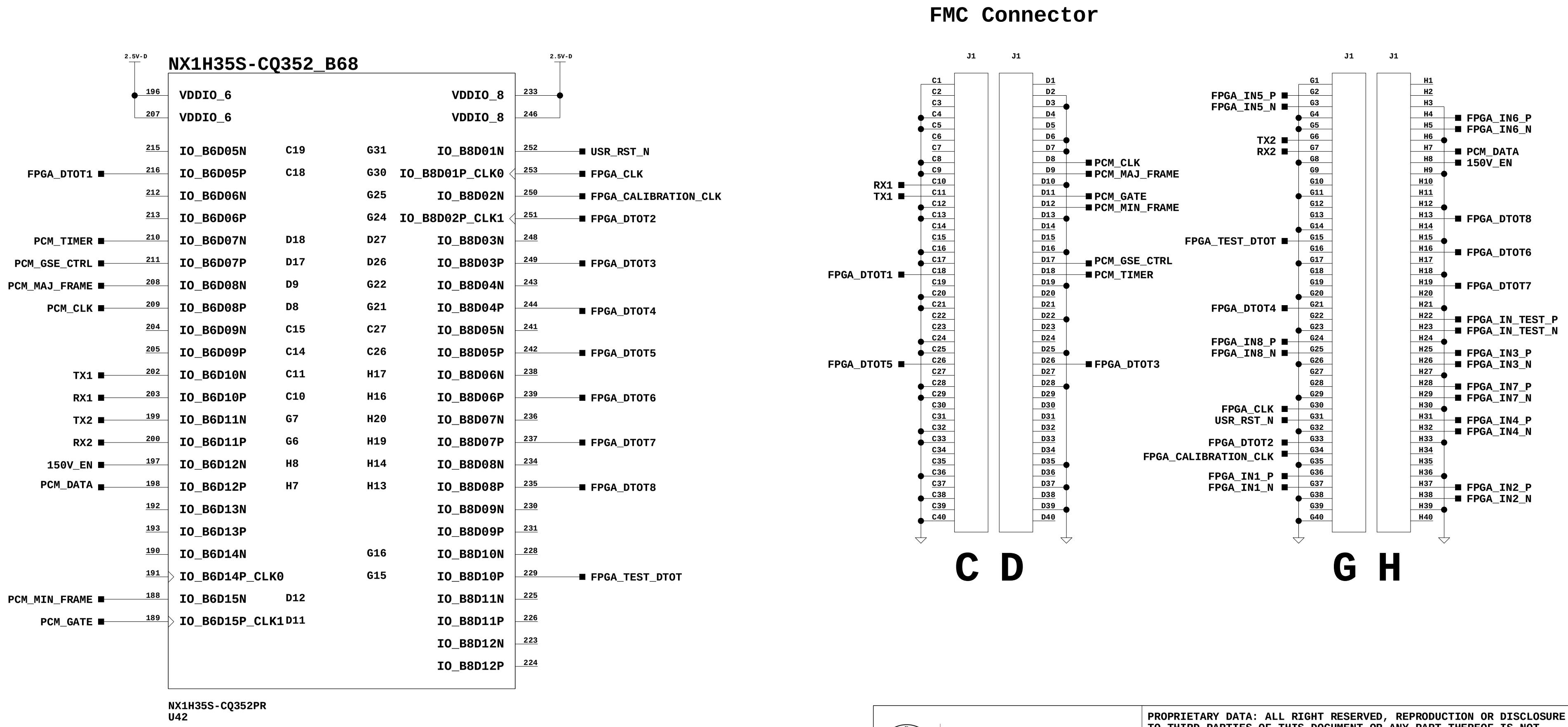


University of Bergen

Project
Distribution of Energetic Electron and Proton

Drawn by Bendik Husa	Date 27/05-21	Drawing title DEEP motherboard	
Checked by	Date		
Approved by	Date	Drawing number	Rev 1.1
PA approved by	Date	Sheet title DEEP / FPGA power	Sheet 5/8

Change description	Rev	Date
Moved comparator inputs to bank 9	1.1	08/05/2021

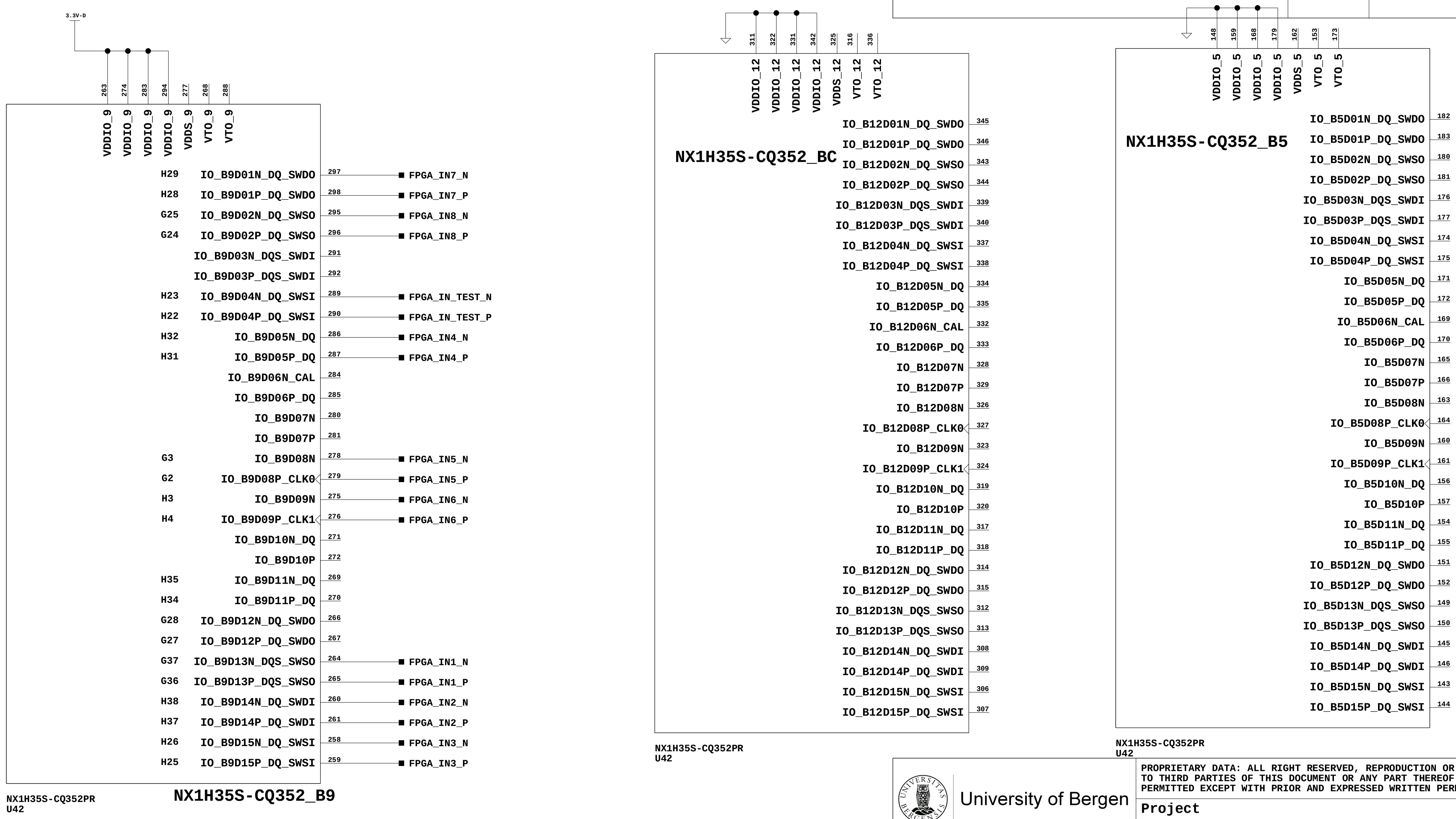


University of Bergen

PROPRIETARY DATA: ALL RIGHT RESERVED, REPRODUCTION OR DISCLOSURE TO THIRD PARTIES OF THIS DOCUMENT OR ANY PART THEREOF IS NOT PERMITTED EXCEPT WITH PRIOR AND EXPRESSED WRITTEN PERMISSION.

Drawn by Bendik Husa	Date 27/05-21	Project Distribution of Energetic Electron and Proton	
Checked by	Date	Drawing title DEEP motherboard	
Approved by	Date	Drawing number	Rev 1.1
PA approved by	Date	Sheet title DEEP / FPGA B6B8	Sheet 6/8

Change description	Rev	Date



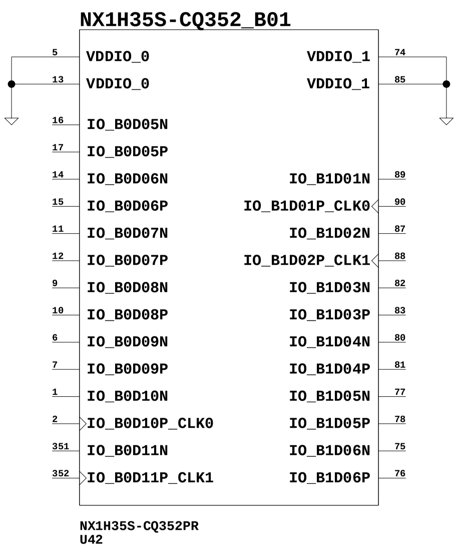
University of Bergen

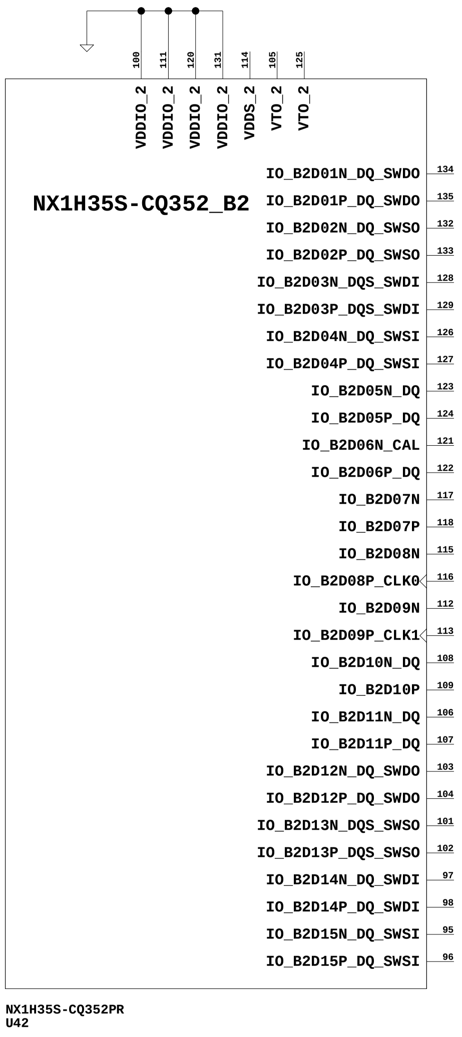
PROPRIETARY DATA: ALL RIGHT RESERVED, REPRODUCTION OR DISCLOSURE TO THIRD PARTIES OF THIS DOCUMENT OR ANY PART THEREOF IS NOT PERMITTED EXCEPT WITH PRIOR AND EXPRESSED WRITTEN PERMISSION.

Project
Distribution of Energetic Electron and Proton

Drawn by Bendik Husa	Date 27/05-21	Drawing title DEEP motherboard	
Checked by	Date	Drawing number	Rev 1
Approved by	Date		
PA approved by	Date	Sheet title DEEP / FPGA B5B9B12	Sheet 7/8

Change description	Rev	Date





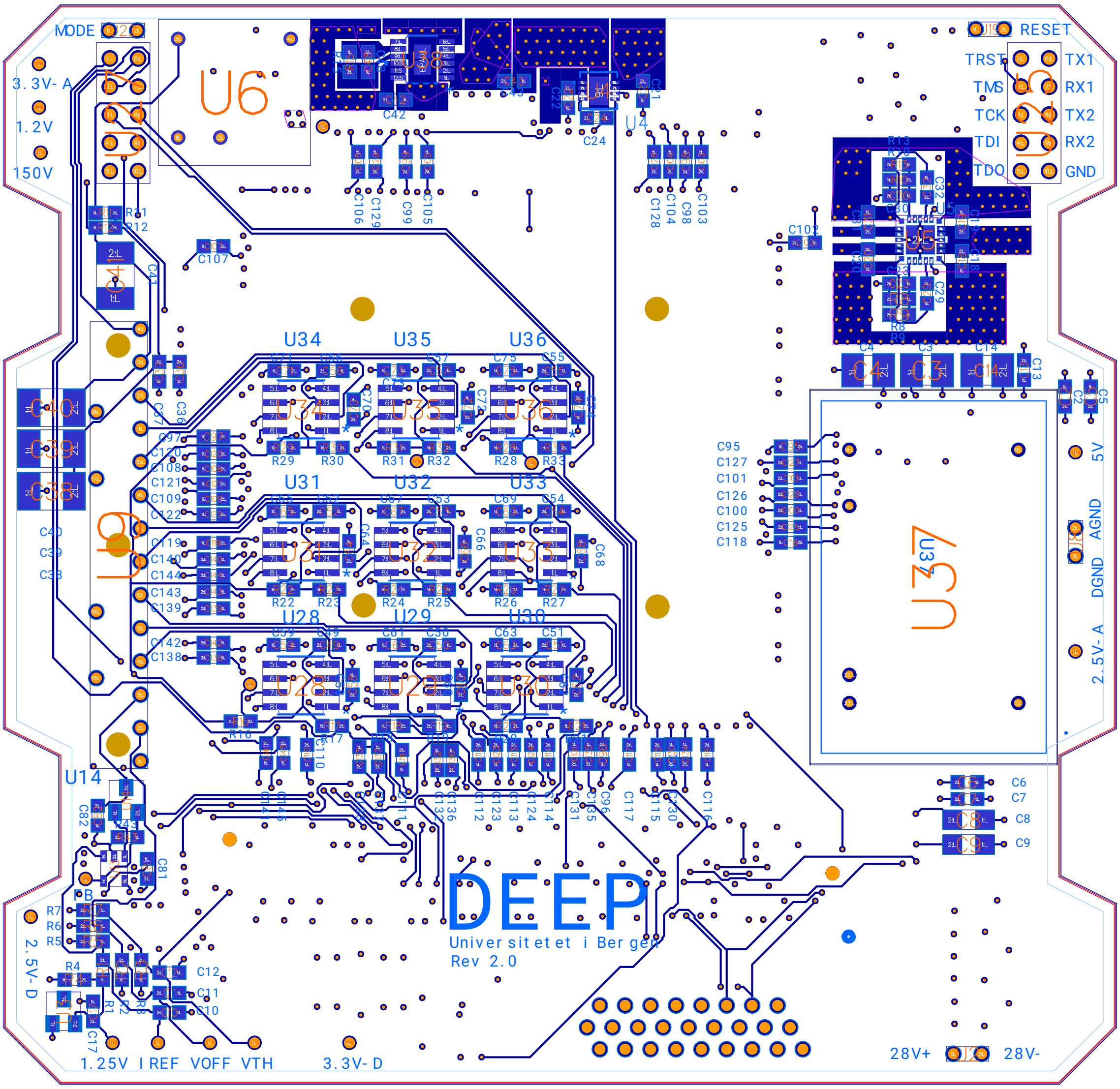
University of Bergen

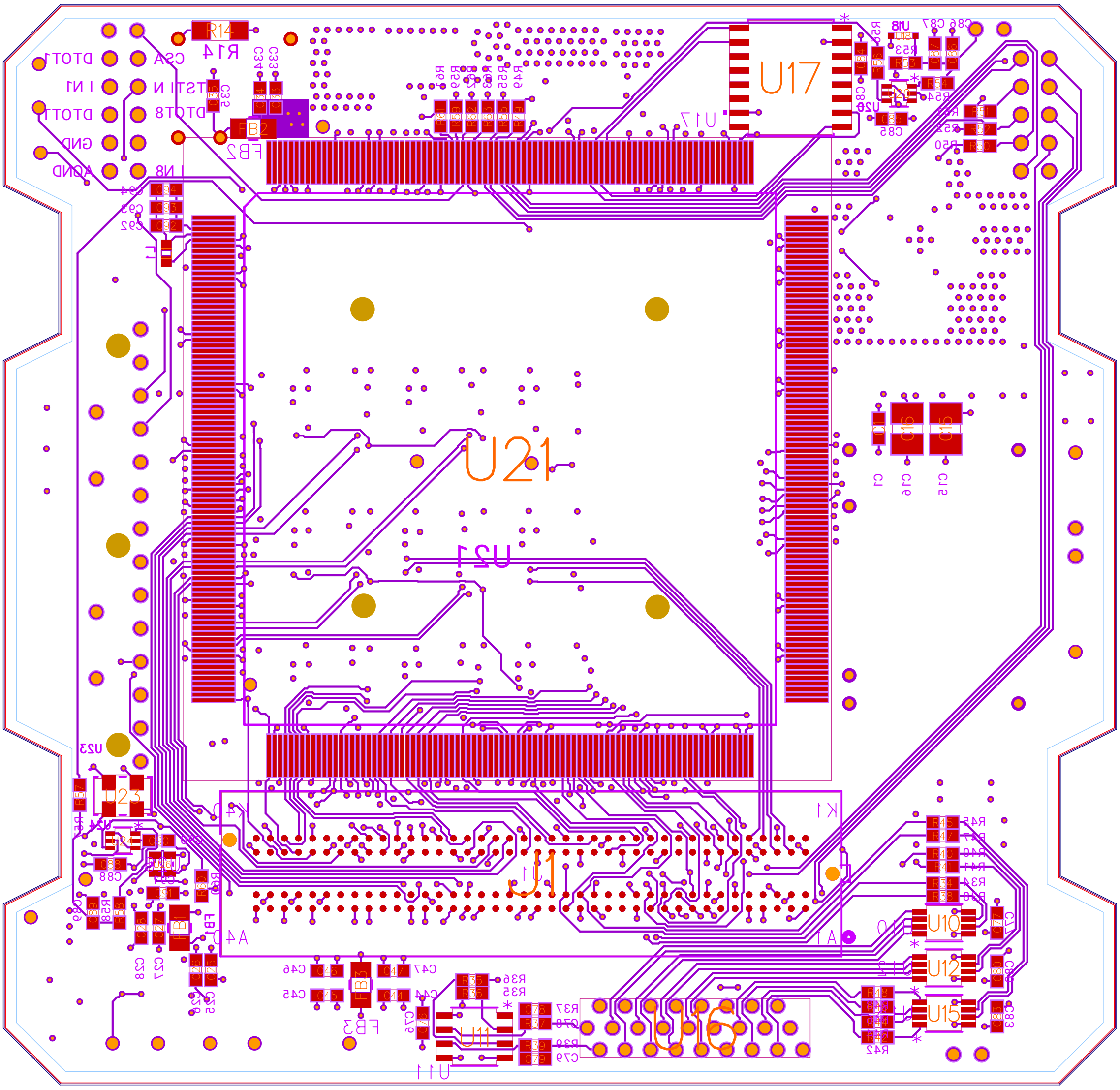
PROPRIETARY DATA: ALL RIGHT RESERVED, REPRODUCTION OR DISCLOSURE
TO THIRD PARTIES OF THIS DOCUMENT OR ANY PART THEREOF IS NOT
PERMITTED EXCEPT WITH PRIOR AND EXPRESSED WRITTEN PERMISSION.

Project
Distribution of Energetic Electron and Proton

Drawn by Bendik Husa	Date 27/05-21	Drawing title DEEP motherboard	
Checked by	Date	Drawing number	Rev 1.1
Approved by	Date		
PA approved by	Date	Sheet title DEEP / FPGA B0B1B2	Sheet 8/8

E LAYOUT


MODE
3.3V-A
1.2V
150V
U6
U4
RESET
TRST
TMS
TCK
TDI
TDO
TX1
RX1
TX2
RX2
GND
U34
U35
U36
U31
U32
U33
U28
U29
U30
U14
U37
5V
AGND
DGND
2.5V-A
2.5V-D
1.25V
IREF
VOFF
VTH
3.3V-D
28V+
28V-
DEEP
Universitetet i Bergen
Rev 2.0


R14
U17
U21
U1
U16
U10
U12
U15
U23
U11
FB2
FB3
C46
C47
C45
C44
R36
R35
R37
U18
U20
C85
C86
C87