Development of a new analog Sum-trigger for the MAGIC experiment with a continuously adjustable analog delay line and automatic calibration

Diplomarbeit

Fakultät für Physik

Ludwig-Maximilians-Universität
München

Dennis Häfner

München, den 08. November 2010

Frontispiece: Image of the Crab Nebula, published by NASA in 2002. In this picture X-ray data from the Chandra satellite (in blue) and an optical image from the Hubble Space Telescope (in red) are superimposed.
In the central region of the nebula the pulsar is located. Until recently it was not possible to observe the pulsar in the γ-ray regime above 20 GeV. In 2008, with the MAGIC telescope and a new type of trigger system it was possible to detect the pulsed γ-ray emission up to ≈ 100 GeV.

Für Anke

Zusammenfassung

Die hier vorgelegte Diplomarbeit befasst sich mit einer Entwicklung auf dem Gebiet der Hochenergie-Gammastrahlenastronomie. Im Speziellen habe ich ein neuartiges Triggerkonzept für das MAGIC-Teleskop entscheidend weiterentwickelt und verbessert. MAGIC ist das momentan größte Tscherenkowteleskop weltweit mit der niedrigsten Energieschwelle aller Systeme zur erdgebundenen Beobachtung kosmischer Gammastrahlung. Mit Hilfe dieser Teleskope observiert man das Tscherenkowlicht von sogenannten ausgedehnten Luftschauern, die durch hochenergetische kosmische Strahlung in der Atmosphäre ausgelöst werden. Auf Grund ihrer extrem geringen Helligkeit können diese Tscherenkowlichtblitze nur bei klaren, dunklen Nächten beobachtet werden. Durch Gammastrahlen verursachte Schauer sind mit einem Anteil von weniger als $0,01\,\%$ an allen Luftschauern relativ selten und zusätzlich durch Hintergrundlicht des Nachthimmels von $> 10^{12}\ \frac{\text{Photonen}}{\text{m}^2\ \text{s}\ \text{sr}}$ stark überdeckt. Diese dominierenden Hintergrundprozesse erfordern spezielle Triggertechniken und limitieren die erreichbaren Energieschwellen nach unten. Ziel meiner Arbeit war die Entwicklung und Fertigung einer völlig überarbeiteten und erweiterten Version eines neuen Triggerkonzeptes – des sogenannten Summentriggers – zur weiteren Absenkung der Energieschwelle. Durch die Reduzierung der Triggerschwelle eröffnen sich neue Möglichkeiten zur Beobachtung von Pulsaren, aktiven Galaktischen Kernen mit hoher Rotverschiebung und den bisher nicht verstandenen Gammastrahlenblitzen im Energiebereich von 20 bis 100 GeV. Zusätzlich kann der Trigger zur umfassenderen Untersuchung von vielen weiteren kosmischen Gamma-Quellen und zum tieferen Verständnis der Entstehungsprozesse dieser Strahlung beitragen. Für den von mir entwickelten Summentrigger, basierend auf einem ersten Prototyp der MAGIC Kollaboration aus dem Jahr 2007, habe ich eine wesentlich verbesserte, neue Triggerelektronik auf drei unterschiedlichen Platinen aufgebaut und erfolgreich getestet. Die umfangreichen Schaltungen kombinieren Analog- und Digitalelektronik, sowie programmierbare Logikbausteine. Um die aufwändige manuelle Einstellung und Wartung des bisherigen Prototyps zu ersetzen, wurde ein komplett softwaregesteuertes, vollautomatisches Kalibrierungsverfahren entworfen. Das zentrale Element der Schaltung ist eine ebenfalls neu entwickelte, stufenlos elektronisch regelbare analoge Verzögerungsleitung. Mit ihr können die zeitlich im Nanosekundenbereich streuenden Signale der Kameraphotosensoren aneinander angepasst werden, was unerwünschte Triggerimpulse durch Untergrundsignale erheblich minimiert. Diese neue Triggerschaltung konnte erfolgreich am MAGIC-Teleskop getestet werden und wird zukünftig in einer endgültigen Fassung im Experiment eingesetzt, vorrangig bei Pulsarbeobachtungen.

Summary

The thesis presented here deals with a development in the field of high energy gamma-ray astronomy. In particular, I have greatly improved and refined a novel trigger concept for the MAGIC telescope. MAGIC is currently the world's largest atmospheric Cherenkov telescope with the lowest energy threshold among ground-based systems dedicated to the detection of cosmic gamma radiation. With these telescopes one observes the Cherenkov light of so-called extended air showers induced by high energy cosmic rays impinging on the atmosphere. Due to their extremely faint intensity, these Cherenkov light flashes can only be observed in clear, dark nights. With a fraction of less than 0.01 % of all air showers, gamma-ray induced showers are rather rare and strongly masked by the night sky background light of $> 10^{12}\ \frac{\text{photons}}{\text{m}^2\ \text{s sr}}$. These dominating background processes require special trigger techniques and limit the lowering of the energy thresholds. The aim of my thesis is the development and fabrication of a completely revised and enhanced version of a new trigger concept – the so-called Sum-trigger – to further lower the energy threshold. The reduction of the trigger threshold opens up new prospects in the observation of pulsars, active galactic nuclei of high redshift and the so far not understood gamma-ray bursts in the energy range of 20 to 100 GeV. In addition, the trigger can contribute to the wider study of many other cosmic gamma-ray sources and to a deeper understanding of the formation processes of this radiation. The new Sum-trigger developed in this thesis is based on a first prototype made by the MAGIC collaboration in 2007. On three different printed circuit boards I built up and successfully tested the new trigger electronics. The extensive circuits combine analog and digital electronics, as well as programmable logic devices. To eliminate the need for costly manual tuning and maintenance demanded by the first prototype, a fully software-controlled, automatic calibration method was developed. The key element of the circuit is a novel, electronically controlled continuously variable analog delay line. It enables the temporal equalization of the signals from the camera photo sensors, whose timing differences are in the order of nanoseconds. This adjustment significantly reduces false triggers from background signals. The new trigger setup was successfully tested on the MAGIC telescope and a final version will soon be applied in the experiment, mainly intended for pulsar studies.

Contents

1. Introduction **1**
- 1.1. Cosmic rays . . . 3
- 1.2. Gamma-ray astronomy . . . 4
- 1.3. Methods of detecting Gamma-rays from cosmic sources (Gamma-astronomy) . . . 7
  - 1.3.1. Air showers . . . 7
  - 1.3.2. Cherenkov light emission in air showers . . . 9
  - 1.3.3. Imaging atmospheric Cherenkov telescopes (IACTs) . . . 11
- 1.4. The MAGIC experiment . . . 12
  - 1.4.1. Mechanical structure and mirrors . . . 12
  - 1.4.2. PMT camera . . . 13
  - 1.4.3. Readout chain and data acquisition . . . 14
  - 1.4.4. Calibration system . . . 14
  - 1.4.5. The standard digital trigger . . . 15
  - 1.4.6. Current analog Sum-trigger . . . 16

2. Concept of the new analog Sum-trigger **17**
- 2.1. Principles of operation . . . 18
  - 2.1.1. Analog sum . . . 18
  - 2.1.2. Large afterpulses and the need for clipping . . . 19
  - 2.1.3. Signal conditioning (amplitude and arrival time flatfielding) . . . 21
  - 2.1.4. Discriminator . . . 22
- 2.2. Advantages of the Sum-trigger . . . 22
- 2.3. Improvements to the current Sum-trigger system . . . 23
- 2.4. Principle of automatic adjustment . . . 25
  - 2.4.1. Sources of signal jitter along the transmission chain . . . 26
  - 2.4.2. Measurement process . . . 29
  - 2.4.3. Amplitude measurement and flatfielding . . . 30
  - 2.4.4. Deriving the optimal setting . . . 31
  - 2.4.5. D-flipflop as timing discriminator . . . 31
  - 2.4.6. Delay adjustment . . . 32
  - 2.4.7. Absolute calibration of clipping threshold . . . 33
  - 2.4.8. Setting of clipping thresholds . . . 34

3. Development of a new continuously adjustable analog delay line **35**
- 3.1. Basic concept of the adjustable delay line . . . 35
- 3.2. The varactor diode, a voltage controlled capacitor . . . 37
- 3.3. Simulation and selection of components . . . 38
  - 3.3.1. Impedance of the delay line . . . 40

3.4. Testing of delay line prototypes . . . 41
3.5. Further optimizations . . . 43

4. Sum-trigger prototype setup 45
4.1. Block diagram of the Sum-trigger prototype . . . 45
4.1.1. Clipping board . . . 45
4.1.2. Discriminator board . . . 46
4.1.3. Computer control board . . . 47
4.2. Electronics and components . . . 47
4.2.1. Amplifier stages . . . 47
4.2.2. DACs . . . 48
4.2.3. Adjustable delay line and bias voltage . . . 49
4.2.4. Attenuator . . . 50
4.2.5. Clipping amplifier . . . 51
4.2.6. Discriminator (comparator) . . . 52
4.2.7. One-shot (multivibrator) . . . 52
4.2.8. Timing measurement unit . . . 53
4.2.9. Board CPU . . . 54
4.3. Digital communication . . . 54
4.3.1. CPU-to-part communication (SPI) . . . 54
4.3.2. Board-to-board communication (SPI derivate) . . . 55
4.3.3. Board-to-computer communication (USB) . . . 56
4.4. Programmable logic . . . 57
4.4.1. Structure of the control data packages . . . 57
4.4.2. Computer control logic . . . 58
4.4.3. Clipping board logic . . . 59
4.4.4. Discriminator board logic . . . 60
4.5. Computer software . . . 61

5. Measurements, results and proof of concept 63
5.1. Installation on-site . . . 63
5.2. Calibration and component tests . . . 64
5.2.1. Precision of the timing measurement . . . 65
5.2.2. Results of the amplitude flatfielding process . . . 66
5.2.3. Performance of the clipping amplifier . . . 68
5.3. Bandwidth of the analog signal path . . . 71
5.4. Final trigger patch rate scan . . . 72

6. Conclusion & Outlook 75

A. Schematics of electronic circuits 77
A.1. Computer control board . . . 77
A.2. Clipping board . . . 78
A.3. Discriminator board . . . 80
A.4. Delay line . . . 82

B. Final prototype boards 83
B.1. Clipping board . . . 83

B.2. Discriminator board 84
B.3. Computer control board 84

C. CPLD program schematics **85**
C.1. Computer control board logic 85
C.2. Clipping board logic 86
C.3. Discriminator board logic 87
C.4. Schematics of encapsulated logic 88
C.4.1. Rate counting logic 88
C.4.2. Positive edge detector 88
C.4.3. 8bit wide bus switch 88
C.4.4. 8bit wide open drain output 89
C.5. Source code of VHDL based elements 89
C.5.1. SPI master 89
C.5.2. SPI slave 91
C.5.3. One-shot module 92

D. Computer software source code **93**
D.1. Computer control software source code 93
D.2. Gnuplot macro for fitting the error function 100

1. Introduction

This diploma thesis deals with the development of an important detector component in the field of high energy γ-ray astronomy, in particular a new trigger concept for ground-based observation of cosmic γ-ray sources with the MAGIC telescopes.

The two MAGIC telescopes, located on the canary island of La Palma, detect Cherenkov light from air showers produced by high energy particles impinging on the atmosphere. To distinguish between those faint events and noise triggers from night sky background light special efforts concerning the trigger system are necessary. Besides a standard digital trigger with a threshold of 55 GeV, a prototype analog Sum-trigger has been installed in October 2007 that enabled to lower the trigger threshold significantly down to 25 GeV [1, 2]. In principle this analog trigger system amplifies and sums up signals of patches of camera pixels using analog electronics. The main difference to other systems is that a discriminator is applied to this analog sum of pixels and not to each individual pixel. Having achieved excellent results in the detection of the pulsed γ-radiation from the Crab pulsar [2] the analog Sum-trigger now is being further optimized. One main improvement is the implementation of an adjustable analog signal delay line to compensate the different signal transition times of the telescope camera's *photomultiplier tubes* (PMTs) and small delay differences of the 162 m long optical fibers used for the analog transmission of the fast PMT signals. This newly developed delay line performs very well concerning its high bandwidth and steplessly variable delay – so far there is no comparable product on the market. Basically it consists of 25 stages of second order passive low-pass filter circuits made of chip inductors and variable capacity diodes. This enables to control the phase shift of each stage – and thus the overall delay of the whole chain of filter circuits – by just applying a voltage to the diodes. Measurements showed that the accuracy of delay adjustment solely depends on the precision of the reverse voltage applied to the varactor diodes. On the current delay line prototype a bandwidth of more than 400 MHz for minimum delay is achieved. The inductors and varactors were chosen such that the total delay range spans about 6 ns. The ideal field of application of this device is the alignment of pulses like in the case of the Sum-trigger. With this delay line the timing differences of pulses coming from different pixels, can be accurately compensated, which is crucial for the Sum-trigger operation.

Another key upgrade of the new trigger system is the computer controlled automatic adjustment of amplitude and delay of the signals. Important requirements were a simple and cost effective circuit and very high precision even for fast signals. To achieve this, a very fast *emitter-coupled logic* (ECL) comparator is used whose one input is connected to the output of the component to be tuned and the other to an adjustable voltage source, in particular a *digital-to-analog converter* (DAC). This setup will be referenced to as "discriminator" in the following. A simple logic implemented in a *complex programmable logic device* (CPLD) counts the rates of

the discriminator's output at several points while tuning through the corresponding values. Due to every signal's intrinsic jitter in delay and amplitude a normal cumulative distribution function can be fitted to the measuring points in order to minimize the number of measurements needed but still maintain highest precision in determining the optimal value, which is found at a count rate of 50 %.
With these concepts the new Sum-trigger will facilitate the process of calibration and further decrease the trigger threshold through a much more precise alignment of pulses and thus permits to detect even lower energy air shower events below 25 GeV.

In this thesis I developed, built, and tested a fully functional prototype of an improved version of the first-generation Sum-trigger setup for the MAGIC experiment. This work included:

- A complete redesign of existing schematics
- The design of new analog and digital circuits
- Detailed studies and ordering of electronic components
- Composing the layouts of three printed circuit boards, assisted by the electronic workshop at two boards
- Planing, designing, simulation and testing of the digital circuits of each board's central logic processor (CPLD)
- The elaboration of a data exchange protocol based on SPI
- Programming a computer software in C, including fully automatic measurement and adjustment of trigger parameters
- The development, simulation, prototyping (schematics, layout, selection of components) and testing of the new continuously adjustable analog delay line
- On-site tests with the new prototype integrated in the MAGIC I telescope electronics

By the tests performed on the MAGIC site, I was able to show that the new prototype is working properly and entirely meets the required specifications, in particular regarding bandwidth and accuracy of adjustment. I could also prove that the new method of automated calibration performs very well.

The text of this thesis is organized as follows:

In this first chapter I will continue with a brief overview of cosmic rays in general and indicate a few examples of observation targets requiring an improved trigger. In addition, I will shortly outline the MAGIC telescopes and the observation techniques. In chapter 2 the basics of the new Sum-trigger concept are presented in depth while chapter 3 describes the development of the key element, the continuously adjustable analog delay line. In Chapter 4 I explain the prototype setup and in chapter 5 the first measurements and validation of the new Sum-trigger is presented. Finally, in chapter 6 the conclusions are summarized and a short outlook is given. In the appendices the schematics of the electronic circuits are attached as well as the CPLD program structures and the listing of the computer program.

Hereafter I will very briefly summarize the fundamentals of cosmic ray physics and discuss a few physics targets that motivated the development of the Sum-trigger.

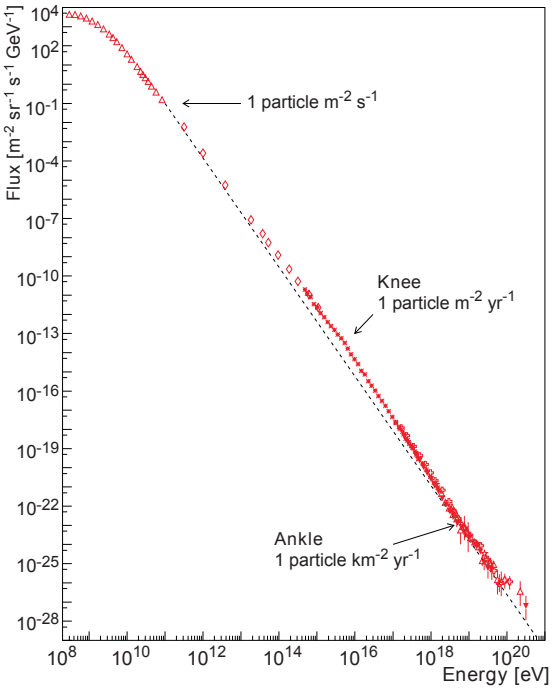


Figure 1.1.: The spectrum of cosmic rays. Below 1 GeV the flux is mainly influenced by the solar wind. Between 1 GeV and $10^{15.5}$ eV the spectrum follows a power law with a spectral index $\alpha \approx 2.7$ and above the so-called *knee* $\alpha = 3$. Taken from [3]

1.1. Cosmic rays

The study of cosmic rays began in 1912, when during balloon-borne experiments Victor Hess discovered that the up to then unidentified ionizing radiation was coming from space. Since then this new field of science has been intensely explored. Nevertheless, the sources of cosmic rays are mostly unknown. Only during recent years a few source candidates have been detected. These discoveries triggered an extensive search for cosmic sources emitting high energy particles.

In figure 1.1 the spectrum of cosmic rays is shown. Obviously, the flux is highly energy dependent: at lower energies, e.g. 10^8 eV, around 1000 particles per square meter and per second can be observed whereas at higher energies of 10^{19} eV the flux drops to less than 1 particle per square *kilo*meter and per *year*, which poses a serious problem for high statistic studies.

At lower energies of a few giga electron volts (GeV) cosmic rays mainly consist of protons (85 %), up to 2 % of electrons and only to less than 10^{-4} % of γ-rays[1] ("Gamma-rays") [4]. The rest are alpha particles, heavier nuclei, and – completely unknown in number – neutrinos. The crux is that up to an energy of 10^{19} eV the charged particles (protons, electrons) are deflected by galactic and extragalactic magnetic fields and thus the observed direction of these cosmic rays on ground does not point back to their origin in space. The only exception are *ultra high energy*

[1] In this thesis I will subsume γ-rays as cosmic rays, though in some publications γ-rays are strictly excluded from the term "cosmic rays".

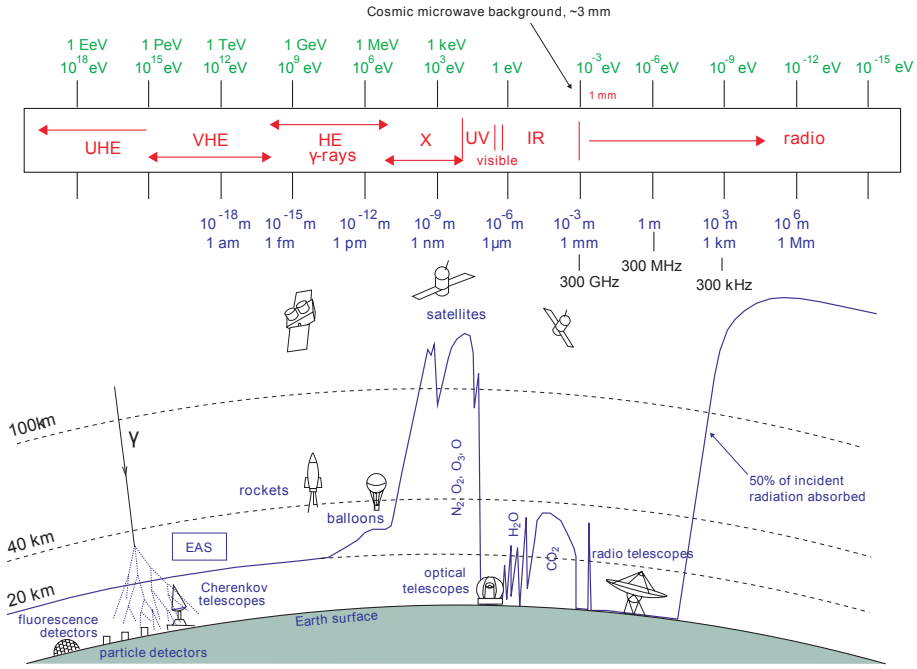


Figure 1.2.: The electromagnetic spectrum and its atmospheric windows for observation [6]. For each wavelength the blue line marks the altitude at which 50 % of the incident cosmic radiation is absorbed. The γ-rays reside on the left of the image, in the very- and ultra-high energy range. A sketch of an *extended air shower* (EAS) and a Cherenkov telescope are shown in the lower left corner.

cosmic rays (UHECR), whose Larmor radius ($r_L = \frac{p}{|q|B}$, with p the particle momentum, q its charge and B the magnetic flux density) is larger than the extension of our galaxy [4]. Only these highest energy particles can be traced back to their source. Unfortunately, recalling the spectrum, they are way too rare to be observed with high statistics, except with experiments distributed over huge areas, like AUGER, covering 3000 km^2 [5]. A way to detect the sources is to observe not the charged cosmic rays but the neutral ones, in particular the γ-rays (photons) which are not deflected by cosmic magnetic fields. This is the sector of γ-ray astronomy.

1.2. Gamma-ray astronomy

In the electromagnetic spectrum γ-rays energetically reside above ultraviolet light and x-rays, with energies above several 100 keV (figure 1.2). Since the most energetic thermal photons in our universe at most reach up to a few keV in energy, high radiation energies can only be achieved in non-thermal processes like in π^0 decays, inverse Compton scattering or in electromagnetic fields by bremsstrahlung or synchrotron radiation [6]. According to the current theoretical models the formation of γ-rays and the origin of the charged cosmic ray particles are mostly locally related, in particular γ-rays should normally point back to the so-called cosmic accelerators. Up to now (fall 2010) nearly a hundred very high energy cosmic γ-sources have

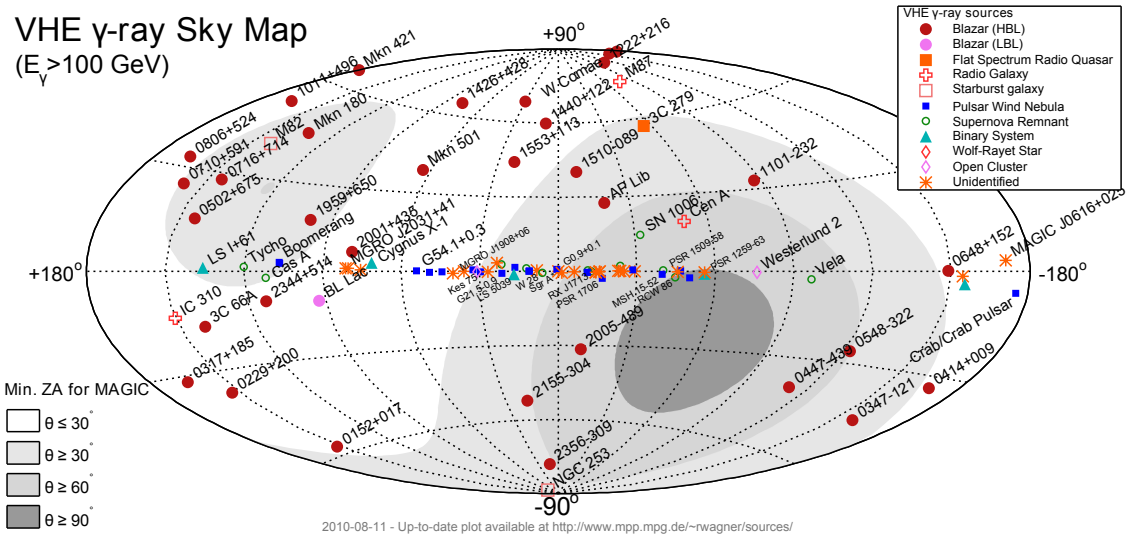


Figure 1.3.: The sky map of cosmic γ-ray sources. Galactic sources are found on the horizontal centerline (area of the galactic plane), extragalactic sources mostly above and below. Note the large number of galactic sources detected so far. The accessible regions for the MAGIC telescopes are displayed in white and different shades of gray according to the telescopes' *zenith angle* (ZA). Taken from [7]

been detected (see figure 1.3). Nearly all detections have been made by ground-based Cherenkov telescopes such as MAGIC. Among the sources there are classes which are either limited in their energy spectrum to maximum $50 - 100$ GeV, such as pulsars, or they are located very far in the universe such that gammas of above $50 - 100$ GeV are interacting with the extragalactic background light (EBL) and do not reach the earth [5]. In the past decades Cherenkov telescopes were improved to achieve lower thresholds, from a TeV 20 years ago to about 100 GeV nowadays. Only the MAGIC telescopes provide a slightly lower threshold. To further reduce the threshold, a new trigger principle had to be introduced in order to observe sources of the above mentioned classes, which are the following:

- **Active galactic nuclei (AGN) at high redshift**

  The term AGN subsumes various astrophysical objects with similar characteristics, usually quasars, blazars and radio galaxies. They are the brightest objects in the universe and thus can be observed over large distances. In general they consist of a central super-massive black hole surrounded by an accretion disc and perpendicularly emitted jets, which frequently are the location of powerful particle acceleration processes.
  AGNs with their jets pointing towards the earth are called blazars.
  Of particular interest are studies of high redshift AGNs, which can only be observed up to about 50 GeV or a few hundred GeV, depending on the redshift [5]. From AGN observations the intensity of the EBL as a function of the wavelength can be deduced, which indirectly determines the intensity of early star formation.

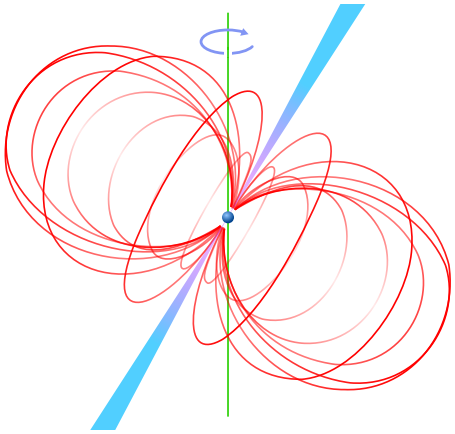


Figure 1.4.: Illustration of a pulsar. The small blue sphere in the middle represents the neutron star, the red curves depict the magnetic field lines and the bluish cones mark the outgoing electromagnetic beam.
Based on http://en.wikipedia.org/wiki/File:Pulsar_schematic.jpg

- **Gamma-ray bursts (GRB)**

  GRBs are phenomena whose source process is basically unknown. They are observed as sudden and extreme γ-ray fluxes lasting only between milliseconds to a few minutes with a longer lasting afterglow in the optical and x-ray spectrum. GRBs occur unpredictable once or twice per day anywhere in the sky. Most GRBs have a high redshift with the most distant one observed at $z \approx 6$. Due to the high redshift (and hence large distance) one can observe γ-rays only below the EBL absorption edge (cutoff in the γ-spectra). Therefore detectors with a γ-threshold of at most $20 - 25$ GeV are needed. During the short outburst GRBs set free more energy than the sun in billions of years, which makes them the brightest γ-ray sources detectable. One possible origin of the very short outbursts is the collision of two very heavy neutron stars or a neutron star and a black hole.

- **Pulsars** ("**Pulsa**ting source of **r**adio")

  The third class of stellar objects to be studied in γ-astronomy with low threshold Cherenkov telescopes are pulsars. If the core of a star undergoing a supernova explosion has between 1.4 and 3 solar masses it can end up in the densest known state of stable matter, a fast rotating *neutron star* with a diameter of only about 20 km and a very strong magnetic field of roughly 10^8 T [4]. Along the magnetic axis a strong electromagnetic beam is emitted induced by the motion of the magnetic field, accelerating protons and electrons. If the magnetic axis and the rotational axis are not aligned, the pointing direction of this beam rotates around the axis like the light cone of a lighthouse (figure 1.4). If the beam points towards the earth, it can be observed as pulsating radiation.
  From current lower energy measurements with satellites it is not possible to determine where pulsed γ-rays are produced within the pulsar system. Two

important models have been developed over the last years: the so-called *polar cap model*, according to which gammas are produced very close to the neutron star surface, and the *outer gap model*, in which gammas are supposed to be produced quite far away from the polar cap, close to the light cylinder limit. Only by measurements at the upper end of the pulsed spectrum one can discriminate between these models. For the Crab pulsar the critical energy range is between 10 and 100 GeV [4].

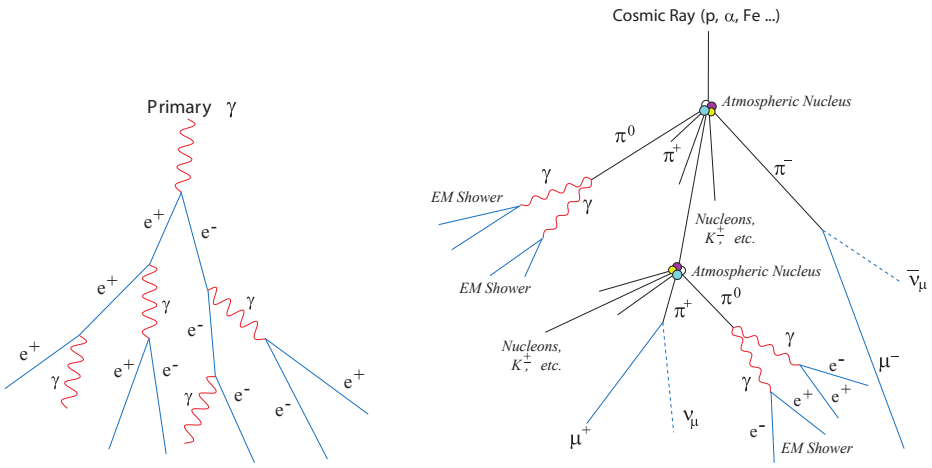
All these examples require observation of γ-rays below 100 GeV, where current cosmic instruments with standard triggers are insensitive. Obviously, also observations from other classes of γ-sources will profit very much from a lower trigger threshold because in general one is very much interested in measuring spectra over the maximum possible range, and in particular, merge the spectra measured by Cherenkov telescopes with the measurements of satellite borne detectors which at most extend to 1 - 50 GeV (depending on the intrinsic flux).

1.3. Methods of detecting Gamma-rays from cosmic sources (Gamma-astronomy)

The first ever detection of cosmic γ-rays took place in July, 1967 when the US military satellites Vela 3 and 4 detected "Gamma-Ray Bursts of Cosmic Origin" [8, 9]. Since then many more space-borne and ground-based systems were installed dedicated exclusively to the detection and measurement of celestial γ-rays. The direct measurement of cosmic γ-rays is solely possible by detectors on spacecrafts or very high flying balloons outside the earth's absorbing denser atmosphere. Ground-based telescopes can detect γ-rays only indirectly by observing the light effects from *air showers* triggered by high energy γ-rays entering the atmosphere. Producing too small air showers, lower energy γ-rays (above some MeV) can be observed from space only. The downside of those satellite detectors is the very limited size of their light collection area. As mentioned in figure 1.1 the flux for higher energies dramatically drops requiring huge detection areas. This induces a strong upper energy limitation of space-borne observatories. The currently most successful γ-ray satellite is the *Fermi Gamma-ray Space Telescope* (FGST, formerly GLAST) with a supposed energy range from 10 keV to 300 GeV for the strongest sources [http://fermi.gsfc.nasa.gov/science/]. Though ground-based telescopes have a much higher energy threshold (usually above 50 to 200 GeV) and observe indirectly, using the atmosphere as a gigantic calorimetric absorber, they have a detection area of several thousand square meters allowing to detect signals from far lower fluxes (higher γ-ray energies).
The basis of this indirect observation will be described in the following section.

1.3.1. Air showers

High energy cosmic particles, either γ-rays (photons) or the far more numerous charged hadrons (e.g. protons), initiate air showers of secondary particles when impinging earth's atmosphere. Though similar in their appearance, electromagnetic and hadronic showers differ fundamentally in their formation and composition.



(a) Electromagnetic shower (b) Hadronic shower

Figure 1.5.: Schematics of the formation of air showers triggered by (a) γ-rays and (b) hadrons such as protons, α-particles or heavy ions [5]. The electromagnetic subshowers (blue) in the hadronic cascade (b) develop like the purely electromagnetic air showers (a) but with much wider lateral spread.

Electromagnetic showers

The γ-ray based electromagnetic showers emerge from γ-rays above an energy of around 20 MeV interacting with an air nucleus resulting in an electron-positron pair creation.

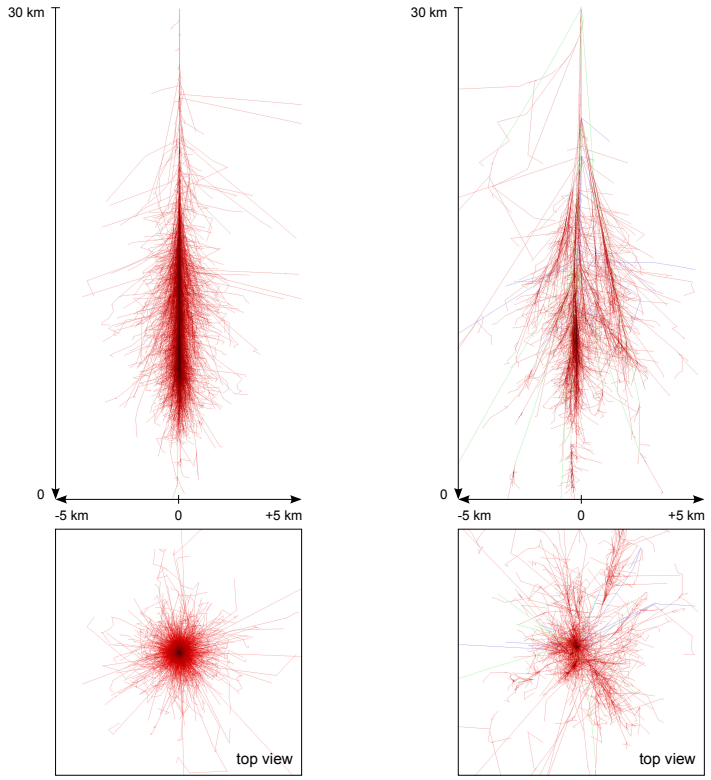
$$\gamma \longrightarrow e^+ + e^-$$

Having enough energy (> 83 MeV) the electrons and positrons generate high energy photons through bremsstrahlung which then can again create electron-positron pairs. Thus an exponentially increasing cascade develops, ending when the average energy of the generated electrons and positrons drops below the necessary limit to produce photons by bremsstrahlung.

The incident γ-photon being ultra-relativistic, the secondary particles are emitted under low angles, thus the spatial progression of those showers is very narrow and mainly concentrated around the incoming direction of the primary photon (figure 1.5 (a)), only widened by multiple scattering and weak deflection by the magnetic field of the earth.

Hadronic showers

Unlike electromagnetic showers, the much more complex hadronic cascades are initiated by charged hadrons interacting strongly with air nuclei. In these strong interactions mesons such as pions (≈ 90 %), kaons (≈ 10 %) and light baryons (protons, anti-protons, neutrons, anti-neutrons) are created. These secondary particles can either decay or interact again with other air molecules, building up the shower. Unlike the other particles, muons and neutrions as decay products of charged pions can even reach the ground. The generated neutral pions usually decay into two



(a) Electromagnetic shower (b) Hadronic shower

Figure 1.6.: Simulations of air showers from cosmic-ray particles with 100 GeV interacting with an air molecule at an altitude of 30 km [10]. The progression of a photon induced electromagnetic shower is shown in (a) whereas (b) illustrates a hadronic shower triggered by a proton. The line colors reveal the particles' types: red lines denote tracks of electrons, positrons and γ quanta, green lines imply muons and blue lines stand for hadron traces. Remarkably is the very different lateral dispersion of the two shower types. Hadronic showers are dominated by electromagnetic subshowers at their "tail".

γ-photons ($\pi^0 \longrightarrow \gamma + \gamma$) which in turn trigger electromagnetic subshowers (figure 1.5 (b)). A shower dies out when the energy of the secondary particles reaches the pion production threshold (≈ 1 GeV), because then ionization losses will dominate.

The strong hadronic interactions pass high transverse momenta on their decay products, spreading the shower wider than electromagnetic cascades (figure 1.6).

1.3.2. Cherenkov light emission in air showers

As visible in figure 1.6 (a) even at γ-ray energies of 100 GeV practically no shower particles reach ground. Thus only shower tail particles from γ-rays with more than several hundred GeV can be directly measured. Another way to observe air showers is to detect *Cherenkov radiation* emitted by ultra-relativistic secondary shower particles.

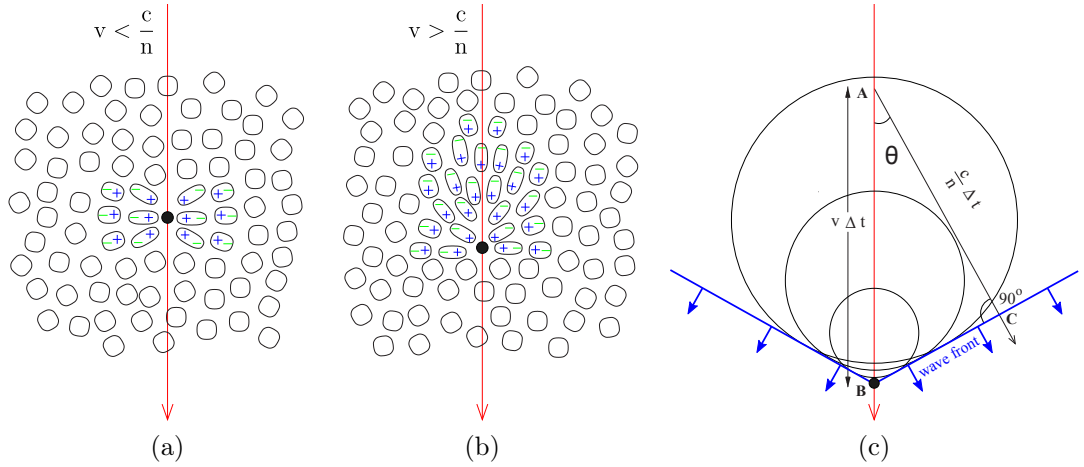


Figure 1.7.: Illustration of the Cherenkov effect. In depiction (a) the charged particle moves through the medium (e.g. air in the atmosphere) slower than the speed of light inside the medium. The medium's electric dipoles can react "fast enough" on the particle trajectory and thus the electromagnetic waves from the dipoles' re-orientation overlay destructively resulting in no effective light emission. In figure (b) the particle moves faster through the medium than the local speed of light such that the electromagnetic pulses from the dipoles are in phase causing a net macroscopic field which is the Cherenkov radiation. In drawing (c) the Cherenkov effect according to Huygen's principle is shown: the elementary spherical waves pile up *behind* the particle forming a wave front seen as the Cherenkov light cone, propagating with the opening angle θ. From [11]

Cherenkov radiation occurs when a charged particle moves through a dielectric medium (e.g. air) at a speed greater than the propagation speed of light in that medium, like in the case of a large fraction of shower particles in the atmosphere. A moving charged particle distorts the surrounding local electromagnetic field in a medium by polarizing the atoms of the medium in close vicinity of the particle. After the particle has passed, electromagnetic waves are emitted through the atomic relaxation processes. In the usual case the particle travels slower than the medium's speed of light ($v_{particle} < \frac{c}{n}$, where c is the speed of light in vacuum and n is the refractive index of the medium) causing the locally emitted waves to interfere destructively with each other and no emission can be seen (figure 1.7 (a)). However, when the particle moves faster than the light inside the dielectric ($v_{particle} > \frac{c}{n}$) then the local waves constructively interfere, thus emitting Cherenkov radiation (figure 1.7 (b)). According to Huygen's principle the elementary Cherenkov light waves of a "superluminal" particle add up to a wave front propagating in a cone-shaped manner (figure 1.7 (c)), similar to the sonic shock wave front (*"Mach cone"*) of a supersonic aircraft. Contrary to the shower particles themselves, the Cherenkov light cones which they evoke, can easily reach the ground. Albeit they represent a very faint occurrence: only around 500 Cherenkov photons per GeV of primary γ-ray energy are created in an electromagnetic shower [5].

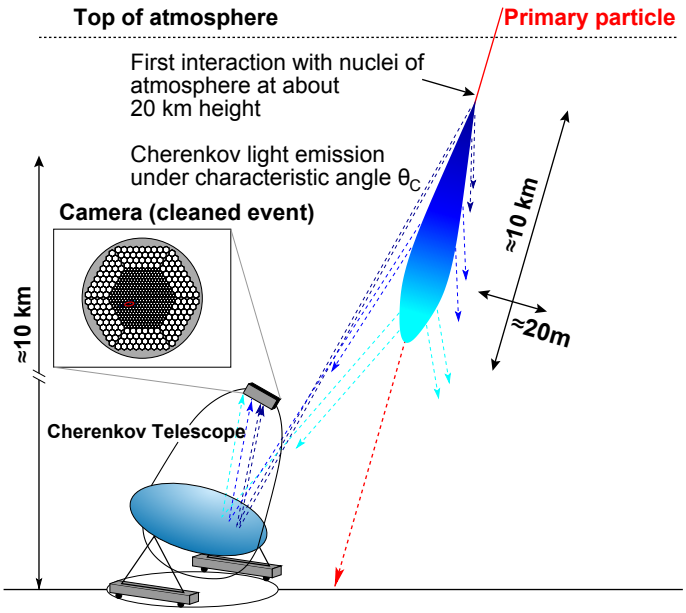


Figure 1.8.: Schematic of the working principle of an IACT. The primary cosmic particle interacts with an air nucleus at a height of 20 km and triggers an air shower of secondary particles which emit Cherenkov radiation (bluish club and arrows). The image of this light emission is recorded by a highly sensitive camera in the focal plane of a telescope mirror. Based on [5]

1.3.3. Imaging atmospheric Cherenkov telescopes (IACTs)

Though being a twofold indirect aftereffect, the Cherenkov light emission from secondary air shower particles enables the ground-based detection of the primary cosmic γ-ray photons with energies below several hundred GeV. Various observation methods for the atmospheric Cherenkov light phenomena have been developed during recent decades. Among them the *imaging* technique, which is currently the most successful one. Besides the basic information where and when an air shower occurred, with the imaging method the type of the primary particle, its energy, its incoming direction and the altitude of interaction can be concluded from the Cherenkov light flashes. The major installations of *imaging atmospheric Cherenkov telescopes* (IACTs) are Whipple/VERITAS (Arizona, USA), CANGAROO (Woomera, Australia), H.E.S.S. (Namibia) and MAGIC (Canary Islands). They all have a common functional principle: The faint and very short atmospheric Cherenkov light flashes from secondary γ-ray induced air shower particles are focused by an optical mirror on a multi-pixel highly quantum efficient photo-detector camera (figure 1.8). Basically a photographic image of the projection of the super-positioned Cherenkov light cones on ground is recorded from which all the relevant parameters of the primary particle can be derived, though with limited precision. The light pulses' short appearance of only $2-3$ ns for γ-ray induced showers and the very small amount of Cherenkov photons triggered by less energetic γ-rays pose a high technological challenge due to a large background from night sky light ($> 2 \cdot 10^{12}\ \frac{\text{Photons}}{\text{m}^2\ \text{s}\ \text{sr}}$).

Figure 1.9.: The two MAGIC telescopes on the Roque de los Muchachos. Image courtesy of Robert M. Wagner

1.4. The MAGIC experiment

Currently the IACT with the lowest γ-ray energy threshold in the world is the *Major Atmospheric Gamma-ray Imaging Cherenkov* (MAGIC) telescope system (figure 1.9) located on the "Roque de los Muchachos", a mountain of volcanic origin on the Canary Island La Palma at a height of 2225 m above sea level (a.s.l.) and 28°46′ N, 17°53′ W. It is a stereoscopic system consisting of two telescopes, MAGIC I and II, each with a tessellated mirror of 17 m in diameter and a camera system comprising a matrix of 577 (MAGIC I) and 1039 (MAGIC II) *photomultiplier tubes* (PMTs). With its standard trigger system it reaches a lower γ-ray energy threshold of 55 GeV and with the recently installed first version of an analog Sum-trigger in MAGIC I, the limit even goes down to 25 GeV for certain cosmic objects.

1.4.1. Mechanical structure and mirrors

Both telescopes are designed as lightweight as possible to enable very fast repositioning for the observation of the ephemeral GRBs. Hence the mirrors' support structures are made up of a very stiff space-frame of carbon fiber tubes (see figure 1.10), allowing to limit the total weight of each telescope to only 72 t. The mirror dishes with a diameter and focal length of 17 m each have a parabolic shape to preserve the timing information of the incoming wave fronts such that a Cherenkov light pulse is not widened in the focal point (camera plane) by the geometry of the reflector. This enables short trigger coincidence windows and improves the separation of γ-ray showers from *night sky background* (NSB) light and hadronic showers. The mirrors are composed of spherically dished glass and aluminum mirror segments with a size of 1 m by 1 m adding up to a total reflective area of ≈ 235 m^2. For opti-

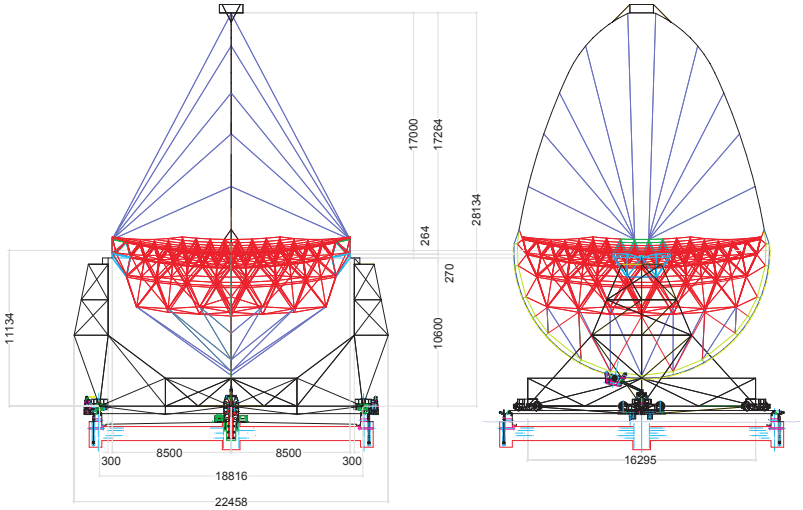


Figure 1.10.: Basic structure of the MAGIC I telescope. On top is the camera box mounted on a 17 m wide arch laterally fixed by steel cables (blue lines). The mirror dish support structure (red) is made of a lightweight and stiff carbon fiber space-frame. Unit of length is mm. Courtesy of Max-Planck-Institute for Physics, construction department

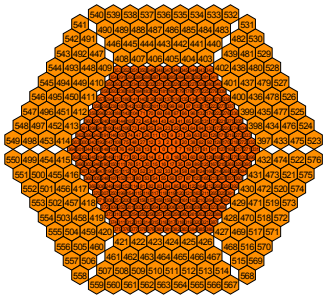
mal focusing each mirror element is mounted on an actuator controlled by the *active mirror control* (AMC) system to compensate for the small deformations of the mirror structure when the telescope's position is changed. Moreover the AMC permits to focus the mirror on various distances, usually 10 km (at small zenith angles) as this is the most probable altitude of a 100 GeV air shower maximum.

1.4.2. PMT camera

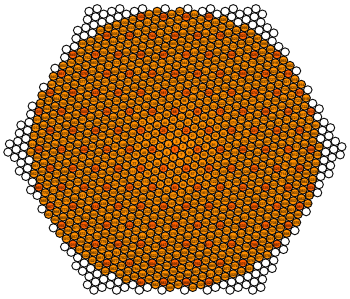
For an IACT the camera is one of the most essential parts. Desirable features are a high quantum efficiency to detect photons even from low energy showers, high resolution and a fast signal response to resolve the very short Cherenkov pulses. On both telescopes the cameras are mounted in the focal point of the parabolic mirror and have a diameter of 1.5 m corresponding to a field of view of 3.6°. They are composed of hexagonally arranged round photomultiplier tubes to convert the Cherenkov photons into amplified electrical signals. Each PMT is topped with a hexagonal light collector ("Winston cones") to guide photons, which would otherwise hit the dead area in between the PMTs, to the photon-sensitive part. The MAGIC I camera has 577 pixels[2] with a peak *quantum efficiency* (QE) of $25 - 30$ %. It has two sectors, an inner part with higher resolution consisting of 397 small 25 mm diameter PMTs and a coarser outer part with 180 larger 39 mm diameter PMTs (figure 1.11 (a) and (b)). MAGIC II has a homogeneous camera with uniform pixels of 1039 PMTs with 25 mm in diameter (figure 1.11 (c)). These PMTs have a more peaked QE of 34 %.

[2] *PMT*, *pixel* and *channel* will be used synonymic from now on

(a) MAGIC I camera



(b) MAGIC I camera scheme



(c) MAGIC II camera scheme

Figure 1.11.: The cameras of the MAGIC telescopes. MAGIC I has a camera with two sectors of differently sized PMTs, visible in (b) as well as on the photo of the camera housing with open lids (a). The MAGIC II camera (c) has a homogeneous arrangement of pixels and an overall higher resolution. Currently only the orange area is equipped with PMTs. Photo (a) courtesy of Robert M. Wagner

1.4.3. Readout chain and data acquisition

In the following I will briefly describe the MAGIC I telescope installation, as the current Sum-trigger and the prototype of the new Sum-trigger was installed and tested there.

As depicted in figure 1.12, the electrical signals from the PMTs are preamplified and converted into optical pulses using *vertical cavity surface emitting lasers* (VCSEL) inside the camera housing. The optical signals are sent through 162 m long multi-mode fiber optic cables (one fiber per pixel) to the nearby counting house where the data processing and recording takes place. The optical transmission has several advantages over the electrical transfer, namely no crosstalk or noise pickup, less dispersion and attenuation. On top of that the optic fibers are much lighter, which is helpful to reduce the total weight since a considerable part of the cable loom runs along the telescope structure. In the counting house the optical signals are first split and then converted back to electrical ones, one signal path going to the trigger system, the other to the digitizing *Flash analog-to-digital converter* (FADC) system, sampling the analog signals sequentially at a rate of 2 GHz using optical delay lines [12]. *Multiplexing* (MUX) is used to reduce the number of readout channels to provide a cost effective FADC system [13]. In the trigger section the electrical signal is split once more to provide dedicated signal paths for the standard trigger and the Sum-trigger. When a trigger outputs a signal then the digitizing process is interrupted and the data present in the subsequent ring buffer is read out and stored by the data acquisition (DAQ) system.

1.4.4. Calibration system

In order to deduce the shower properties it is necessary to know the number of Cherenkov photons that hit each pixel. To get the conversion factor from the number of photons arriving at a PMT to the corresponding output value of the ADC at the

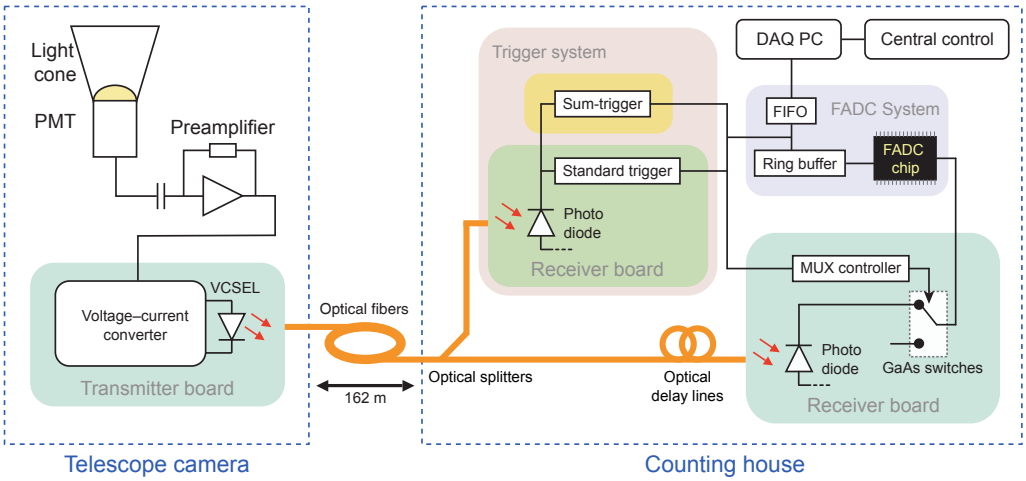


Figure 1.12.: Schematic of the readout chain of MAGIC I. In the camera the PMT signals are preamplified, converted to optical pulses by VCSELs, and injected into optical fibers (one fiber per channel). After 162 m transmission to the counting house, each optical signal is split. One part is digitized by a *multiplexed* (MUX) FADC system using optical delays, the other part is used for the trigger system. There, the optical signal is converted back into an electrical one and split again to provide separate signal outputs for the standard trigger and the Sum-trigger. If a trigger "fires", the digitizing is halted and the digital data in the ring buffer is read out by the data acquisition (DAQ) system.

end of the signal transmission chain a calibration with a well defined light source is necessary. Therefore differently colored *light emitting diodes* (LEDs) are mounted in the center of the mirror dish illuminating the camera homogeneously. The LEDs provide fast light pulses at different wavelengths and variable intensity to calibrate the whole dynamic range of the camera photo sensors and their readout chain. The calibration of the signals recorded by each pixel is done in two steps: first a relative and then an absolute calibration. Due to production tolerances practically every single element of the transmission chain has a slightly different gain. Therefore, the relative calibration is used to equalize the response of different channels when exposed to the same amount of light. An absolute calibration is needed in order to convert the FADC value of the signal recorded by a pixel into physical quantities related to the flux of photons arriving at the camera. The absolute calibration is performed by means of a method which exploits the fact that a photomultiplier generates a small *excess noise* which is linearly related to the initial number of *photoelectrons* (Phe) that produce its output signal. More details can be found in [14].

1.4.5. The standard digital trigger

Permanently the PMTs are hit by photons from the night sky background and only occasionally by the Cherenkov light produced by a cosmic air shower. With a

continuous sampling rate of 2 GHz the amount of data from all pixels is simply too large to be processed within reasonable time and the data acquisition system would be saturated. Thus a Cherenkov telescope needs a trigger to preselect whether a sample might represent photons from an air shower and should be recorded or not. As Cherenkov light flashes from cosmic showers are extended and very short-lived events they coincidentally expose several neighboring camera pixels within a time window of ≈ 10 ns (for primary particle energies in the range of $30 - 300$ GeV). Consequently a trigger has to scan for such distinctive footprints in the continuous data stream. In MAGIC this is usually done with the standard digital trigger system, built up from very fast but power consuming *emitter coupled logic* (ECL) elements. It consists of three logic levels, L0, L1 and L2, whereas L2 is a freely re-programmable logic layer of *field programmable gate arrays* (FPGAs) which is still not used and will not be further explained. The output of L1 actually triggers the data acquisition process.

L0-Trigger. The L0-trigger covers the 325 innermost pixels of the MAGIC I camera. Each channel has a dedicated discriminator that generates a logic signal if the output of a PMT reaches a certain threshold, usually above 5-6 photoelectrons.

L1-Trigger. In the L1-trigger a logic AND is applied to the L0 output of a compact group of neighboring pixels, also known as *next neighbor* (NN) logic. In the standard mode 4 neighboring pixels (4NN) have to emit a signal on the L0 within a time window of 6 ns to produce a trigger signal at the L1 output.

1.4.6. Current analog Sum-trigger

In an attempt to further lower the triggering energy threshold below 55 GeV, a new trigger concept was developed and tested by M. Rissi et al. [1]. This analog Sum-trigger is currently installed besides the standard digital trigger in MAGIC I. It was set up in October 2007 and mainly intended for pulsar studies [4]. Having a separate signal path it works completely independent of the standard trigger. Instead of using a discriminator on every single pixel and then digitally form coincidence patterns, the analog Sum-trigger rather applies a discriminator to the analog sum of signals from a patch of neighboring pixels. A detailed description of its working principle will be given in the next chapter. With this new trigger concept it was possible to reduce the minimal γ-ray energy necessary to detect its air shower photons by a factor of 2, giving rise to the first-ever detection of the pulsed γ-rays from the Crab pulsar [15]. Due to this and other very promising results, this first version of the Sum-trigger will be upgraded, including a twin to be made for the MAGIC II telescope. The purpose of this thesis was to develop the concept and a fully functional prototype for this upgrade.
The analog Sum-trigger concept will be explained in detail in the next chapter.

2. Concept of the new analog Sum-trigger

For studies of the Crab pulsar in the γ-ray regime, the low energy emission ($E \lesssim 50$ GeV) of the pulsating component of the γ-radiation coming from the Crab nebula poses a difficult task.

Before 2008 there was no IACT sensitive enough to provide evidence for these pulsating γ-rays above 25 GeV and a new concept to detect γ-radiation below 50 GeV had to be found. MAGIC currently is the most sensitive ground-based Cherenkov telescope with the standard setup, and it was possible to further improve the sensitivity with a new trigger design [1].

The standard MAGIC setup is recording γ-ray events with energies down to only ≈ 50 GeV, leaving the gap of $10 - 50$ GeV unexplored. The Sum-trigger concept, developed by the MAGIC group of the Max-Planck-Institute for Physics, provided a reduction of the telescope energy threshold, giving significant overlap with the energy range of the FERMI γ-ray satellite.

The aim of the Sum-trigger system was to lower the minimal energy threshold but still enable the separation of γ-ray air showers from the night sky background. In addition, the resulting trigger rate had to be kept low enough not to saturate the *data acquisition queue* (DAQ), which is capable of recording events at a rate of up to 2 kHz [5, 4]. Consequently, at the end of 2007 a prototype setup of the analog Sum-trigger was installed at the MAGIC I telescope and first measurements of the Crab pulsar revealed that the new concept was successful [4]. The functional principle of the trigger is described in the next section.

Up to now the first version of the Sum-trigger was successfully working for 3 years but required intensive maintenance from experts to occasionally readjust the system by hand.
The main intention to develop a new version of the Sum-trigger along with this thesis is the implementation of automated tuning of the trigger parameters, to eliminate the need of time-consuming manual adjustment and improve the system's reliability. In addition, the upgrade will include a larger camera area on which the trigger is sensitive and an optimized bandwidth[1] of the analog signal path.

[1] *Bandwidth* generally is defined as the difference between the lowest and highest frequencies of a signal. A high bandwidth of an electronic circuit usually describes its ability to transmit especially high frequency components of a signal. If the bandwidth is low, short input pulses are widened due to the limited signal rise time (see chapter 3.4).

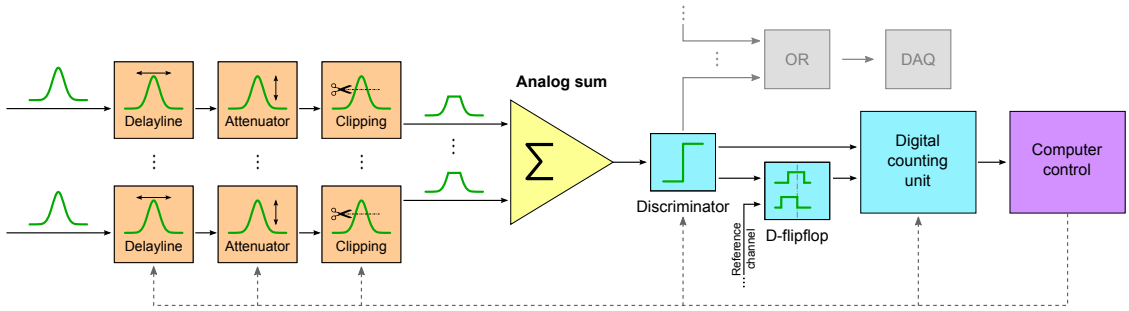


Figure 2.1.: Simplified schematic of the Sum-trigger prototype concept. The electric pulses (green) coming from the receiver unit first go to the *"clipping board"* (orange tiles) where they are adjusted in delay and amplitude and then clipped. After the separate analog *"summing board"* (yellow triangle) the summed analog signal is fed into the discriminator located on the *"discriminator board"* (blue tiles). The discriminator's digital output is split. One path is routed to the digital counting unit (big blue tile). The other path is connected to a D-flipflop that compares the arrival times of the pulses with a reference signal. The flipflop output is also connected to the counting unit. The digital communication system is located on the *"computer control board"* (purple) which transfers data between the boards and the computer program, in order to read out results from the counting unit and to set the values calculated by the control software.

2.1. Principles of operation

The challenge for both triggers, the Sum-trigger and the standard trigger, is the same: the electronics have to be sensitive to simultaneously arriving pulses from a group of PMTs, which were illuminated by an extended atmospheric Cherenkov flash, while ignoring accidental signals from single PMTs provoked by the night sky background light or intrinsic noise. In order to suppress such accidental events, both trigger types apply topological and timing constraints on the incoming signals, adapted to the Cherenkov flash properties. The special properties of the Sum-trigger will be described in the following.

2.1.1. Analog sum

As the name "analog Sum-trigger" suggests, its basic task is to sum up the signals of several neighboring camera pixels with analog summing amplifiers (see also figure 2.1). Such a group of adjacent PMTs whose signals are summed is called a *patch* of pixels. The size and shape of a patch is critical for the trigger rate: if the patches are too small they capture only a fraction of a widespread Cherenkov light cone image and their signal sum usually is to weak to trigger, especially for the very faint lower energetic showers. Conversely, if the patch size is too large, accidental triggers from background light can dominate the rate. The optimal size of the patches was derived from *Monte Carlo* (MC) simulations as described in [4].

In general, an extended Cherenkov flash illuminates several pixels of a patch which then (in the ideal case) synchronously output signals. These signals simultaneously arrive at the summing stage where they are added up to one big pulse that can exceed the threshold of the subsequent discriminator, which then emits a trigger signal.

The physical motivation of adding up several PMT signals before the discriminator is the detection of low energy showers (< 50 GeV) whose faint Cherenkov light is more widespread over a group of pixels (> 6) and does not fulfill the condition of the standard trigger, that 4 next neighbor pixels all reach at least 6 Phe.
Associated with this is the demand to achieve a higher *signal-to-noise ratio* (SNR)[2] for the trigger decision: the noise from NSB light is suppressed in the sum because the *root mean square* (RMS) levels of uncorrelated Gaussian noise sources only add up like

$$U_{SummedNoise} = \sqrt{U^2_{NoiseSource1} + U^2_{NoiseSource2} + \cdots + U^2_{NoiseSourceN}}$$

[16] whereas the simultaneously arriving signals form a Cherenkov light flash sum up regularly. According to the triangle inequality

$$\sum_{i=1}^{N} U_i \geq \sqrt{\sum_{i=1}^{N} U_i^2}$$

the sum of correlated signals will always yield a higher value than the sum of uncorrelated noise if the RMS noise level per channel does not exceed the amplitude of the actual signal.

In other words, the analog sum provides the smallest possible time coincidence window, neglecting very close precedent or following noise pulses: in a perfect scenario (synchronous Gaussian-shaped pulses) the analog sum of all the signals is not wider than one single pulse.

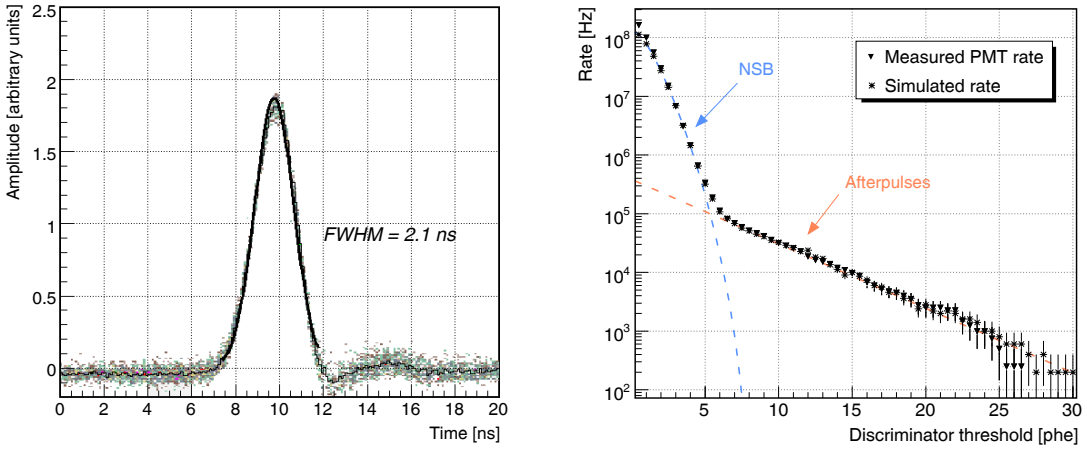
The essential peculiarity of the analog sum compared to the standard trigger's next neighbor logic is that even very weak signals can contribute to the trigger decision, while in the standard trigger system only signals above the L0 threshold (6 Phe) are considered.

A nice side effect of the analog summing circuit is its simplicity compared to the complex digital logic necessary for the standard trigger. Since it is located on separate *printed circuit boards* (PCBs) it also allows a quite flexible arrangement of patterns in which the channels are summed.

2.1.2. Large afterpulses and the need for clipping

Unfortunately, the analog sum concept has a drawback: it is sensitive to a specific type of PMT noise, called *afterpulses*. These very short noise pulses (figure 2.2 (a))

[2] The term SNR might be misleading since it is normally used to define the quality of the recorded images. Here it is utilized for the sensitivity of the trigger: a high SNR means the trigger is sensitive to very weak signals "hidden" in the noise floor. Actually the images from lower energy showers detected by the Sum-trigger naturally have a lower SNR than the much higher energetic shower images captured by the standard trigger.



(a) Shape of an afterpulse

(b) Single pixel NSB and afterpulse spectrum

Figure 2.2.: Typical shape of a PMT's afterpulse (a) at the input of the current Sum-trigger and rate scan over a range of discriminator thresholds (b) using a single pixel exposed to NSB (without signal clipping). The Monte Carlo (MC) simulation in (b) was performed with an NSB level of 0.18 Phe/ns. Above 6 Phe the rate is completely dominated by afterpulses. Due to the flat spectrum, even for very large amplitudes (high discriminator thresholds) the afterpulse rate is not negligible and dominates largely over the rare Cherenkov events. Images kindly provided by Maxim Shayduk.

appear with a small probability after any signal (Cherenkov flash or single Phe from the huge NSB rate) with a variable delay in the range of µs. They follow an exponential amplitude spectrum, which is much broader than the pure NSB spectrum (see figure 2.2 (b)), implying that some afterpulses can have huge amplitudes. They originate from impurities (foreign molecules) on the surface of a PMT's first dynode, which can be ionized by accelerated photoelectrons. These positively charged ions (mainly H^+) are accelerated from the first dynode back to the photo cathode, where they induce the emission of electrons, which then in turn occasionally produce large PMT output signals (figure 2.6) [17].
Yielding a large amplitude, one single afterpulse alone can already exceed the discriminator threshold of the sum and trigger the data taking. Respectively, the number of afterpulses of all channels together would completely dominate the trigger rate and saturate the DAQ with accidental events. Therefore the idea of signal clipping[3] was introduced, preventing any PMT pulse to surpass a certain limit, which is much lower than the discriminator threshold [4]. As the clipped afterpulses of different pixels are not coincidental, their analog sum does not reach the trigger level, in contrary to the many small simultaneously arriving pulses from a Cherenkov flash which "pile up" in the analog sum, even when clipped (figure 2.3). By clipping, the

[3]Clipping a signal means to limit (or "cut off") its amplitude at a certain level, preventing any pulse to exceed this limit.

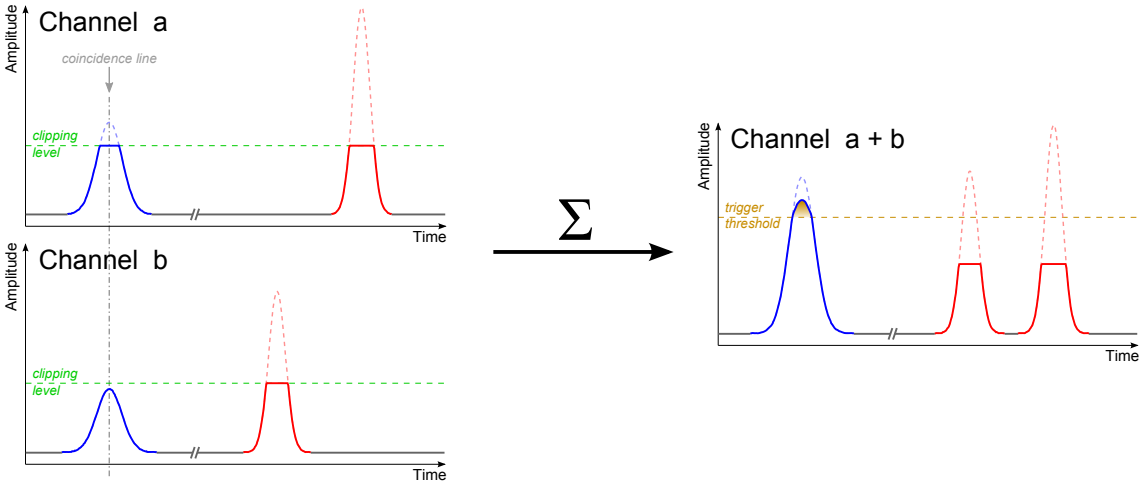


Figure 2.3.: Principle of signal clipping and analog sum. On the left, artificial signals of two channels are displayed and on the right their analog sum. The pulses from a Cherenkov flash is tinged blue, afterpulses red. Dashed curves represent the non-clipped signals. Without clipping, both afterpulses would exceed the trigger threshold and cause false triggers (right). If clipping (green dashed lines) is introduced the afterpulses do not trigger anymore, but the Cherenkov pulse components, simultaneously arriving on both channels (left) still pile up and trigger correctly when summed (right).

sensitivity of the Sum-trigger is slightly reduced, but this is accepted in favor of the efficient reduction of false triggers [4].

2.1.3. Signal conditioning (amplitude and arrival time flatfielding)

Since the Sum-trigger's functional principle is based on a topological and temporal analysis of the signals, it is crucial to equalize the gains and transit times of all channels in the trigger area. Individual pixels with a higher gain would have a bigger share in the sum (only restricted by the clipping level), consequently dominating the amplitude of the sum and hence the accidental trigger probability. Conversely, channels with a differing delay would not (fully) add to the summed pulse which then could be too small to trigger, thus affecting the Sum-trigger's efficiency to detect weak shower signals.
Uniform gains are also required to correctly translate the pulse amplitudes to a physical relevant value, namely the corresponding amount of photoelectrons released in the PMTs.

Aberrations of gains can occur despite the general camera flatfielding[4], because some channels have a differing amplification factor due to tolerances in the trigger electronics (from the optical splitter up to the summing stage).

[4]In the *flatfielding* process or *flat-field correction* the gains of all PMTs in the camera are equalized in order to achieve homogeneous sensitivity of the camera and improve the image quality.

Different delays among the channels arise mostly due to two reasons:

- The optical fibers, guiding the signals from the camera to the counting house, have a tolerance in length and thus in signal propagation time of around 2 to 3 ns.

- In the camera flatfielding procedure the gains of the PMTs are equalized by adjusting the high voltage applied to their dynodes (see figure 2.6). This can change the signal transit times inside the PMTs by up to 2 ns.

While the individual transit times of the optical fibers remain unchanged and could be compensated with fixed delays, the transit times of various PMTs can change after each camera flatfielding, which is performed once a year. Hence, equalizing delay elements are required that have to be readjusted on a regular basis.

2.1.4. Discriminator

To issue a final trigger decision from the analog signals, a discriminator is used that outputs a digital pulse when the summed analog input signal exceeds a previously set threshold. In case of the Sum-trigger, the discriminator circuit is placed directly after the analog summing stage, applying one discriminator per sum patch.
To avoid large trigger rates of a patch when a bright object (e.g. a star) is in its field of view, a trigger rate control system automatically adjusts the discriminator thresholds such that the rate of each patch equals the current average rate per patch[5].

2.2. Advantages of the Sum-trigger

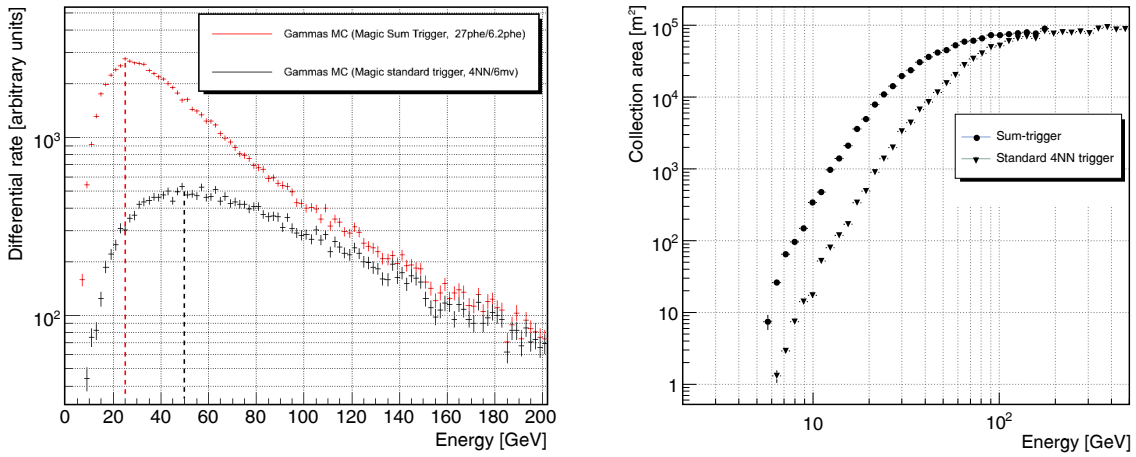
In order to determine the Sum-trigger efficiency, detailed Monte Carlo studies were carried out [4]. As can be seen in figure 2.4 (a), the main advantage of the analog Sum-trigger over the standard digital trigger is the significantly reduced trigger threshold. It is a lot more sensitive to Cherenkov signals "hidden" by NSB light than the standard trigger. As a result, the Sum-trigger substantially extends the telescope's collection area[6] to lower energies (figure 2.4 (b)).

The Sum-trigger's exceptional performance in the energy regime above 25 GeV allows to set new physical targets. Many interesting phenomena like the pulsating γ-ray components of pulsars, GRBs, or high redshift AGNs can solely be observed with the high detection sensitivity provided by the Sum-trigger.

Additionally, the rather simple Sum-trigger circuits turned out to be very cheap and less power consuming compared to the complex and expensive fully ECL based

[5]Strictly speaking, this approach has an adverse effect: it equalizes the "noise levels" of the different patches. However, due to its simple implementation and good performance it is further applied.

[6]The energy dependent *collection area* characterizes the performance of an air Cherenkov telescope (for a detailed explanation refer to [5]). It is the area in which air showers can be observed by the telescope folded with the detection efficiency.



(a) Differential rate

(b) Collection area

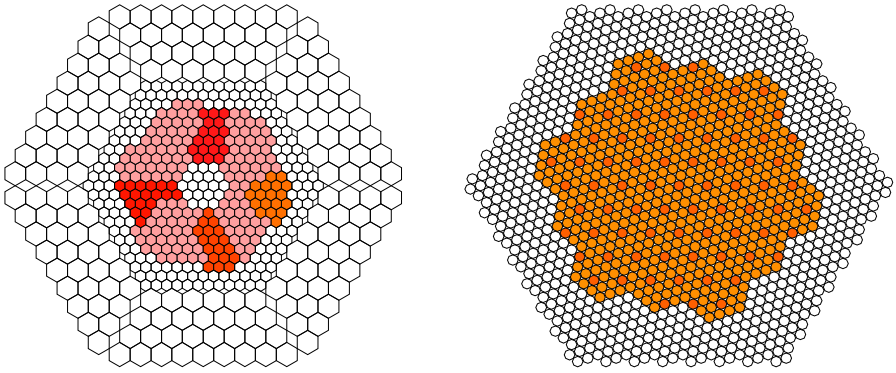
Figure 2.4.: Monte Carlo simulations of the differential rate (a) and collection area (b) for the MAGIC telescope energy range, comparing the standard trigger and the current Sum-trigger. In (a) the standard trigger rates (black) fall off below 50 GeV, whereas the Sum-trigger rates (red) follow the cosmic ray energy spectrum even down to 25 GeV. A comparison of the collection areas of both triggers in (b) shows an improvement of around one order of magnitude at 20 - 30 GeV when the Sum-trigger is applied. Plots courtesy of Maxim Shayduk.

standard trigger. With the separation of the summing stage from the other electronics, the Sum-trigger also offers the possibility to easily adopt the composition of the groups of summed pixels to future physical demands.

A critical drawback of the currently installed first Sum-trigger version is the need of intensive maintenance to sustain its functionality. This will be solved in the new Sum-trigger design, which then will attain the reliability of the standard trigger.

2.3. Improvements to the current Sum-trigger system

The new analog Sum-trigger will have various improvements compared to the current Sum-trigger system.
First of all it will feature completely computer controlled automatic calibration and adjustment of all its parameters.
Second, the introduction of a continuously adjustable analog delay line addresses the problem of changing signal transit times and enables a much more precise and faster alignment of pulses than the coaxial cables used in the present installation (for details of the current setup see [4]). This will probably lower the trigger energy threshold even further. Additionally, avoiding the large amount of cables considerably reduces the space requirement of the setup.
And third, through the application of high speed components and small optimiza-



(a) MAGIC I Sum-trigger area (b) MAGIC II standard trigger area

Figure 2.5.: Figure (a) shows the sensitive area of the first generation Sum-trigger in MAGIC I. To simplify the electronics only the 216 reddish colored channels were used. The ring-shaped topology is optimized for shower energies below 50 GeV, with asymmetric summing patches of 18 pixels each [4]. For comparison, in (b) the much larger sensitive region (orange colors) of the standard trigger in MAGIC II is displayed. The new Sum-trigger will use the same large area, enabling wobble observation with the Sum-trigger and improve its trigger efficiency also for showers with energies above 200 GeV [4].

tions of the circuit, the overall bandwidth of the analog signal paths will be raised. This reduces the widening of the pulses and hence minimizes the time coincidence window, helping to further suppress background noise artifacts in the trigger decision.

The most crucial optimization concerning trigger efficiency represents the complete redesign of the trigger's sensitive camera area and patch sizes. Initially, the current setup was intended to observe at lowest energy thresholds in ON-OFF mode[7] and hence was equipped with only a small annular trigger region (figure 2.5 (a)). This ring shape was chosen because the majority of the images of showers with energies below 50 GeV lies in this area, when the γ-ray source is centered in the telescope's field of view [4]. For the lately introduced *wobble observation mode* the pointing of the telescope is slightly shifted next to the source, thus also the image on the camera is displaced from the center (for details see [18]). Consequently, the current Sum-trigger can not be efficiently used in wobble mode.
The new system will have a much larger trigger region, covering the same area as the MAGIC II standard trigger (figure 2.5 (b)). This will improve the trigger efficiency for widespread showers with energies above 200 GeV, making it equal to the standard trigger in this energy regime [4]. In addition, wobble observation with the Sum-trigger will be possible without any restrictions.

Concerning the electronics, two main modifications were already implemented in the new prototype developed in this thesis:

[7] In *ON mode*, the telescope points directly towards the observed source, whereas in *OFF mode* a near-by region without any known γ-ray source is observed to measure the current NSB [18].

- In the current Sum-trigger, simple potentiometers are used to tune amplitudes by hand, requiring intensive expert's maintenance. The new prototype is equipped with a digitally adjustable attenuator chip on each channel.

- Also the delay adjustment is updated: for the previous setup a large number of coaxial cables of different lengths was needed to add compensating delays to each channel. Now this is done by a newly developed continuously adjustable analog delay line module (see chapter 3).

Both upgrades enable an automatized flatfielding procedure which is essential for the reliability of the Sum-trigger, especially for the new system with a large number of over 500 trigger channels, which is unfeasible to maintain by hand.
In addition, the much more precise new delay adjustment should improve the trigger sensitivity further by a better matched temporal "stacking" of pulses in the sum.
It is worth to point out that the adjustable delay line is indispensable for an efficient Sum-trigger with a large number of trigger channels.

All important trigger parameters, like clipping and discriminator thresholds or even the trigger area, will be easily adjustable with a computer interface to adapt the Sum-trigger to future observation methods, making it a very flexible system.

2.4. Principle of automatic adjustment

The major upgrade of the Sum-trigger will be the completely automatic adjustment and calibration system, whose functionality was evaluated with the new prototype created in this thesis. Many crucial components of the telescope gradually change their properties. Especially the PMTs are critical, since their amplification factor can shift with time. Accordingly, they need to be re-calibrated periodically by varying their operation voltage. As mentioned in section 2.1.3 this also slightly alters the propagation delay between incident light pulse and electrical output signal (transit time). This way both, amplitude and delay can change for all pixels individually. With the current Sum-trigger installation one faces the problem that it can hardly be adapted to those changes or it is rather time-consuming. Thus, the circuits had to be extended by adjustable elements and a measuring unit, ideally equipped with a digital computer interface. On one hand the new electronics should be as simple and robust as possible, while on the other hand high precision in measuring and setting the correction factors was demanded and on top of that the process should not take too long to adjust all channels of the camera's trigger region.

Usually, to determine signal amplitudes, high-speed *analog-to-digital converters* (ADCs) are used. Likewise, *time-to-digital converters* (TDCs) are applied to translate the time difference of two pulses into a digital value.
Though these devices offer the advantage of presenting immediate measurement results and being commercially available, their application is a huge cost factor, especially taking into account the required very high temporal resolution and the large number of channels of the new Sum-trigger. With additional ADC and TDC components, also the circuit's intricacy will increase significantly. In particular, the implementation of TDCs is very challenging [19].

Thus, to keep the complexity of the new Sum-trigger circuits as low as possible, a novel measuring technique based on the evaluation of a series of rate measurements is introduced, for which in principle only very few additional electronics is required. In particular, the discrete trigger output of the discriminator unit is used to perform the measurement of amplitudes by counting rates, which means to count the number of events that surpassed the discriminator threshold in a certain time span (section 2.4.3).
Similarly, the rates to determine the delay of each channel are derived from the output of a regular one bit memory device ("flipflop") that can be used to compare the arrival times of two pulses (see section 2.4.6).

For the rate counting procedure the artificial light flashes from the calibration LEDs in the mirror plane are used. During the normal data taking they are emitted with a constant rate of 25 Hz. An electronic gate signal is sent from the calibration unit to the Sum-trigger counting unit each time when the LEDs flash[8], making it possible to separate the calibration pulses from noise or shower signals: Only in the very short time window when the gate signal is in logic HIGH state the rate counter can increment its value.

The distinctive feature of the new measuring technique is to take advantage of intrinsic variances in amplitude and arrival time of pulses, to efficiently derive the optimal settings for the trigger components.

2.4.1. Sources of signal jitter along the transmission chain

The signal coming from each camera pixel underlies certain fluctuations in both, amplitude and delay, even if the telescope camera were exposed to perfectly reproducible and frequency stable light pulses (idealized calibration pulses). The relative amplitude variations are in the order of $\approx 10\,\%$ [20] and the signal transit time jitters[9] slightly below 1 ns. These fluctuations primarily originate from the PMTs, while other sources are imperfections in the calibration unit and in signal transmitting components of the telescope. The main sources of these jitters are described in more detail below.

Signal properties of PMTs

The calibration pulses used for the measurement process induce roughly 50 Phe in each PMT. Hence, the PMT output amplitudes should follow a Poisson distribution. However, a PMT introduces additional fluctuations to the signals. As shown in figure 2.7 the PMT response to a single photon already has a large variance in amplitude (charge). This is due to the stochastic nature of the amplification process of the PMT dynode system (figure 2.6). The rather low number (order of 10) of secondary

[8] Actually, for my prototype measurements I used a self-made "coincidence gate signal" which was generated by discriminating the analog sum of 5 arbitrary pixels loosely spread over the whole camera. Only the very uniform calibration light pulse can illuminate all 5 pixels simultaneously and initiate coincident signals whose sum can reach the discriminator threshold. The gate signal from the calibration unit of MAGIC I was simply too wide and arriving much too late. This problem is planned to be solved in future upgrades of the MAGIC I telescope.

[9] In analog signal transmission the term *jitter* usually is used for Gaussian distributed variation of pulses, with respect to the amplitude, delay, and width. [16]

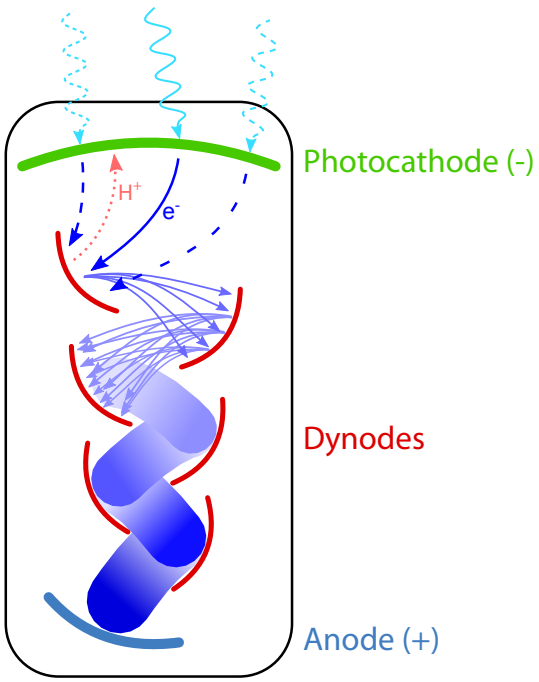


Figure 2.6.: Working principle of a PMT. Photons (wavy arrows) hit the photo cathode producing electrons (blue) by the photoelectric effect. The electrons are accelerated in the electric field between photo cathode and the first dynode, hitting the dynode surface such that many more electrons are set free through secondary emission. With each dynode being held at a higher voltage than the previous one an electronic cascade develops with an ever-increasing number of emitted electrons from one dynode to the next. When the bunch of electrons hits the anode a sharp current pulse appears. Occasionally an electron can punch out positively charged ions (e.g. H^+, red dotted arrow) from foreign molecules accumulated on the dynode surface, which are accelerated back to the photo cathode where they produce a large amount of electrons which in turn are accelerated forward again, resulting in a big afterpulse. Depending on the geometric position of the spot where an incident photon hits the photo cathode the length of the path along which the electrons and the following cascades travel varies (dashed lines), causing jitter in the transition times of the signals (typ. 0.5 - 1 ns).

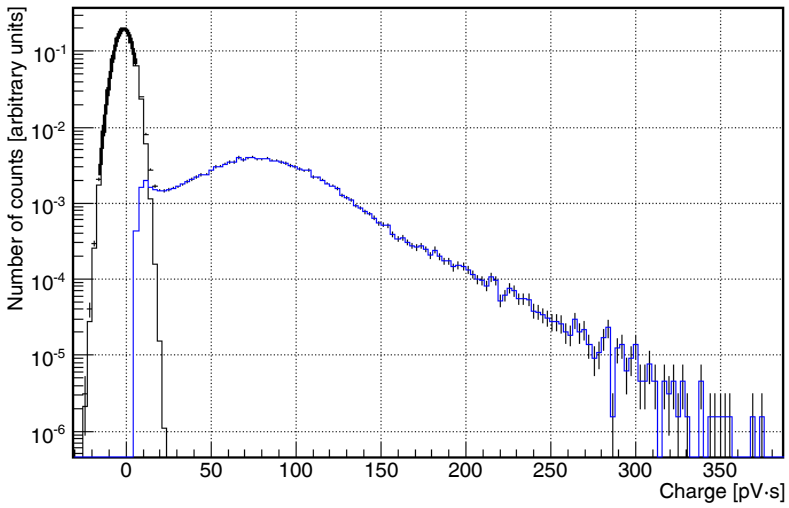


Figure 2.7.: Charge distribution of pulses from a PMT exposed to pulsed low intensity laser light ($\ll 1$ photon per pulse on average, FWHM < 1 ns). The width of the pedestal or *zero charge peak* (black curve) reflects the electronic noise of the experimental setup. The blue curve with rather large variance is the so-called *single photoelectron peak*. The right slope of the curve is slightly distorted by rare 2 and 3 Phe events. Measurement by Matthias Kurz, Max-Planck-Institute for Physics, Munich.

electrons evoked from the first dynode follows almost Poisson statistics and hence has a large relative variance (order of 30%). By the further amplification stages, this relative variance can not be reduced and thus dominates the output charge fluctuations of the PMT. This specific noise property is used for the calibration process of the MAGIC camera and readout chain, as mentioned in chapter 1.4.4.

The PMT transit time jitter (typ. 0.5 - 1 ns) is mainly caused by differing path lengths along which the electrons travel inside the PMT (see figure 2.6) [21, 20].

Even though the amplitude distributions and time jitters of single photon signals do not follow Gaussian statistics, for the further calibration methods the output signals can be assumed to be Gaussian distributed due to the high number of photons involved (> 30 per event and PMT).

Variances in signal transmission and calibration pulses

Also other components in the signal transmission chain, like the electronics or VCSELs, induce small Gaussian distributed fluctuations in amplitude and delay. However, with only ≈ 0.5 Phe and ≈ 10 ps, these variances are negligible compared to the large jitter of the PMTs.
In contrast, the optical calibration pulses emitted by the LEDs of MAGIC I show considerable variations in luminosity and emission frequency, which are clearly visible in the PMT output signals (see section 5.2.2).

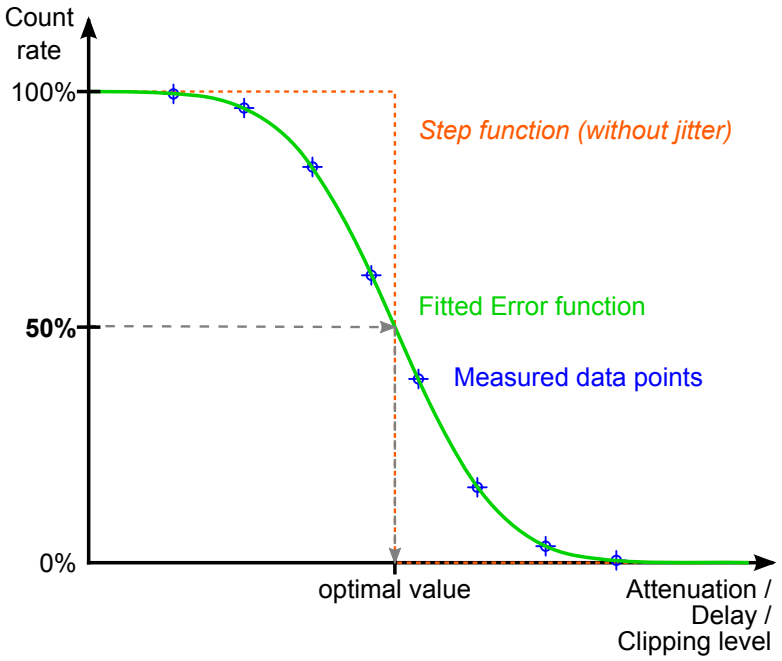


Figure 2.8.: Principle of deriving the optimal setting of either the attenuation, the delay or the clipping level by counting rates. An error function (green) is fitted to the measured rates (blue) in order to find the optimal value at a rate of 50 % (inflection point of the error curve). If there would be no Gaussian distributed jitter in the signals, the data points would follow a step function (orange dashed line).

2.4.2. Measurement process

A simplified block diagram of the functional elements including the measuring units of the new Sum-trigger prototype is shown in figure 2.1. The adjustable components, which alter the analog signals in terms of amplitude or delay, are located on the *"clipping board"*, before the summing stage. On the subsequent *"discriminator board"* a digital counter records the output pulses of the discriminator or a *D-flipflop* (see section 2.4.5).

For the measurement process the number of trigger signals from the discriminator (or D-flipflop respectively) is counted over a certain period of time, while varying either the attenuation, delay, or clipping threshold. This way one obtains a series of rate measurements (rate scan) of the transition region from maximum to minimum number of trigger signals (or *counts*) (figure 2.8).

For a better understanding, I will first elucidate the measurement procedure by the example of amplitude flatfielding, before describing the method to derive the optimal setting and explaining the processes to adjust the delay and clipping threshold.

2.4.3. Amplitude measurement and flatfielding

To equalize the gains of all channels in one trigger patch, the required attenuation per channel is determined. Therefore, rates[10] of calibration pulse triggers are measured for various different attenuator settings (hence amplitudes), for each channel individually. Thus, the rate versus attenuation can be plotted, as shown in figure 2.8.

The gain adjustment is performed by the following procedure:

1. Clipping of pulses is disabled completely
2. The discriminator threshold is set to a fixed value[11]
3. Since the channels have to be measured one after the other, only the channel which is currently measured is active, the others are switched off. Thus, only the signals on the active channel reach the discriminator (see figure 2.1)
4. On the active channel, the adjustable attenuator is tuned stepwise[12] from minimum to maximum attenuation
5. At each attenuation level the digital counting unit (see figure 2.1) measures the rate of trigger signals induced by calibration pulses
6. The rates are transferred to the control software which computes the optimal attenuation value by fitting an error function to the data (see next section)
7. Finally, the channel's attenuator is set to the derived value, then the next channel is measured, starting with step 3

The discriminator threshold is kept unchanged for all gain measurements in one patch, so that each channel's calibration pulse amplitude is tuned to the same value. Presuming the calibration pulses do not change their average amplitude[13] while the patch is calibrated, all channels have the same gain in the end.

Due to the signal's intrinsic amplitude jitter, the measured rates follow a sigmoid curve, as outlined in figure 2.8. The method to derive the ideal attenuation setting from this curve is explained in the following section.

[10] Here *rate* means the number of logic HIGH level signals (e.g. from output of discriminator) per given period of time.

[11] The discriminator threshold has to be just low enough such that the calibration pulse amplitudes on all channels of the patch can be attenuated to it

[12] The step "widths" were carefully chosen to provide enough data points in the transition region while keeping the total number of measurements small.

[13] Occasionally the calibration pulses slightly change their average amplitude when the telescope's pointing is changed in wobble mode. This is a technical problem of the MAGIC I telescope and will be corrected in a future upgrade.

2.4.4. Deriving the optimal setting

In a perfect scenario where the analog pulses had no variance in amplitude (or timing), the rate scan would follow a Heaviside step function[14]: either the analog signal does not reach the adjusted threshold, giving a rate of 0, or the signal exceeds the threshold, yielding 100 % of the the maximal rate (25 Hz when the standard calibration pulses are used).

With Gaussian distributed fluctuations in the signal, there is no sharp discrimination of the (varying) pulses and the step function is "smoothed". The resulting curve corresponds to a *normal cumulative distribution function* (normal CDF), which is identical in shape to the Gauss error function[15] $\mathrm{erf}(x)$, differing only in scaling and translation [22]. The error function is defined by

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-k^2} dk$$

[23], which is the integral of the normal distribution (normalized Gaussian function) with finite limits. In order to derive the value, where the threshold equals the averaged pulse amplitude or delay, the error function is fitted to the measured rate values. The position of 50 % of the maximum rate, which corresponds to the inflection point of the error function and hence to the peak of the underlying Gauss distribution, represents the optimal setting.

Generally, due to the fluctuations of the pulses and the consequent possibility to fit the smoothly shaped error function to the rate values, only few measurements are needed to derive the optimal value with adequate precision. With the absence of any jitter, the accuracy in finding the ideal value (which then was represented by a step function) would depend only on the step width of the rate scan, requiring a large number of rate measurements to precisely find the optimal setting. Since each rate counting process takes a certain time (≈ 7 s), the intrinsic jitter helps to speed up the whole calibration process by minimizing the number of necessary measuring points.

2.4.5. D-flipflop as timing discriminator

For the determination of signal transit time differences, a simple but efficient circuit with high temporal precision was needed. For this purpose, we found the "D-flipflop" as a suitable device.

A D-flipflop (figure 2.9) is a special type of *bistable multivibrator* which usually serves as a one bit memory in digital circuits (for further details the reader is referred to [16]).
However, it can also be applied to discriminate the arrival times of two digital pulses.

[14] The discontinuous Heaviside step function is the integral of the Dirac delta function. Without jitter the results of a repeated pulse measurement would have no variance and could be described by the delta function. In the real world with random signal jitter the delta function is replaced by the Gaussian distribution, whose (finite) integral is the Gauss error function.

[15] For a series of measurements with normal distributed outcome, a standard deviation σ and expected value 0, the Gauss error function $\mathrm{erf}\left(\frac{a}{\sigma\sqrt{2}}\right)$ yields the probability that for a single measurement the error lies between $-a$ and $+a$.

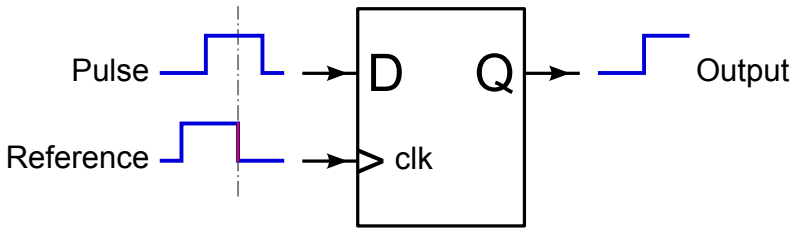


Figure 2.9.: Schematic drawing and working principle of a D-flipflop. If the rising edge (red marking) of a digital pulse (blue) enters the clock input ("clk") the output ("Q") is set to the logic value present at the data input ("D"). In the drawing the data pulse arrives before the clock pulse so the output will toggle to HIGH state.

A minimal D-flipflop has two inputs, namely data ("D") and clock ("clk"), and one output ("Q"). When the clock line is pulled to a logic HIGH level, the logic state given at the input D is set to the output Q. If the clock goes LOW, the last known state at the output Q is hold ("memorized"), no matter what happens at the input D, until the clock goes HIGH again.
To use this property for timing measurements, the D-flipflop has to be *edge-triggered*: A positive-edge-triggered D-flipflop is sensitive to the rising edge of a clock signal, which means that in the moment when a rising edge of a digital pulse enters the flipflop's clock input, the output Q is set to the logic state given at the data input D. Hence, the flipflop outputs a logic HIGH signal, when a data pulse arrives before a clock pulse. Conversely, a logic LOW is output when the clock pulse arrives first (see also caption of figure 2.9). This characteristic enables to discriminate between the arrival times of two digital pulses.

2.4.6. Delay adjustment

The procedure to equalize the delays among all trigger channels is mostly identical to the amplitude flatfielding process, only two different components are used.

- Instead of the attenuator, the adjustable analog delay line is applied to vary the signal's transition time.
- A very fast ECL D-flipflop takes over the discriminator's measuring functionality, in order to compare the arrival time of the pulses on the active channel with the ones on a separate reference channel. The reference channel's circuit is identical to the other Sum-trigger channels[16] and is connected to a selected camera pixel, whose signal has medium delay, such that all other channels can be adjusted to it.

To get the maximum rate of the calibration pulses the discriminator thresholds are set to a rather low limit.

[16]The reference channel was designed identical to the other channels to ensure that the reference pulse is delayed and deformed in the same way as the pulses on the other channels.

With the D-flipflop can be determined whether the reference pulse or the pulse on the active channel arrived first (see section 2.4.5). For this purpose its data input D is connected to the active channel's discriminator output, whereas the clock input is tied to the discriminator output of the reference channel. As described in section 2.4.5, with this setup the flipflop outputs a logic HIGH signal when the pulse on the active channel arrives at the D input before the reference pulse reaches the clock input. In contrary, a logic LOW is output when the reference pulse arrives first.

While tuning the delay line of the currently active channel stepwise from maximum to minimum delay, the rates of logic HIGH level pulses from the flipflop are recorded and evaluated the same way as the discriminator's output in case of the amplitude flatfielding process (section 2.4.3). Here it is the timing jitter inherent in the signals that causes the measured rates to follow an error curve. The optimal delay value is derived from the rates in exactly the same manner as the attenuation value in the amplitude measurement. Finally, the delay line is adjusted to the optimal setting and the next channel is being measured.

2.4.7. Absolute calibration of clipping threshold

After the amplitude flatfielding procedure all calibration pulses of the measured patch have the same average peak voltage which is well-defined by the discriminator threshold applied during the flatfielding.

The number of Cherenkov photons emitted by an air shower is proportional to the shower's energy [4]. Consequently, for an absolute calibration of the Sum-trigger thresholds it is necessary to know the conversion factor from the number of photoelectrons evoked in the PMT's photo cathode to the corresponding peak voltage of the pulse arriving at the Sum-trigger's discriminator. During an observation the number of photoelectrons triggered by an average calibration pulse is automatically evaluated (for details refer to [14]). Unfortunately, the current calibration pulses are considerably wider than pulses induced by Cherenkov flashes or by afterpulses. Since the amount of charge (photoelectrons) is related to the area enclosed by the pulse shape, a calibration pulse and an afterpulse can not be compared only by means of their amplitude voltage. In order to properly compare the peak voltages a conversion factor has to be found. This is done by using the afterpulses coming from the PMTs. The factor is derived in terms of widths of afterpulses (and not for the actual Cherenkov light induced pulses) because the absolute calibration is only necessary to set the optimal clipping threshold for the reduction of afterpulse induced triggers. For the trigger decision itself an absolute calibration in number of photoelectrons is not relevant. For the time resolution of the MAGIC telescope electronics, afterpulses are like delta functions representing the shortest pulses possible, as if they were initiated by single photoelectrons. Measuring their FWHM and comparing it to the FWHM of the calibration pulses the correction factor C can be derived with the formula

$$C = \frac{U_{calibPulse}}{N_{phe_{calibPulse}}} \cdot \frac{\mathrm{FWHM}_{calibPulse}}{\mathrm{FWHM}_{afterpulse}} \tag{2.1}$$

[4]. To get the discriminator threshold voltage that corresponds to a certain number of photoelectrons the following equation holds:

$$U_{discr} = C \cdot N_{phe} \tag{2.2}$$

For a proof of this concept see [4].

With the knowledge of the conversion factor C it is possible to set the clipping thresholds to a physically significant value in number of photoelectrons N_{phe}. In Monte Carlo simulations and measurements of the standard trigger and the old Sum-trigger a value of 6 Phe gave the best trade-off between the suppression of afterpulse triggers and a low NSB rate [4].

The absolute calibration only has to be performed once when the Sum-trigger is installed or after significant changes in the signal transmission chain have been made.

2.4.8. Setting of clipping thresholds

The clipping thresholds are adjusted in exactly the same manner as the attenuation in the flatfielding procedure (section 2.4.3). The only difference are:

- The discriminator threshold is set to the clipping value that was derived in the absolute calibration (section 2.4.7)
- Instead of the attenuation the clipping level is varied

Like in the amplitude measurement, the rates measured at different clipping levels also follow an error function shape due to the signals' amplitude variations, and the method described in section 2.4.4 can be applied to derive the correct clipping level for each channel.
Finally, each channel's signal is clipped at a voltage level that equals the amplitudes of pulses induced by 6 photoelectrons.

3. Development of a new continuously adjustable analog delay line

To superposition the PMT signals of a Cherenkov light flash with high precision, the equalization of the arrival times of the signals from different channels was a crucial prerequisite for the working principle of the analog Sum-trigger. In the old setup the delays have been equalized by coaxial cables of different lengths which had to be replaced by an adjustable and more compact signal delay unit, called *analog delay line*. No commercial products are available that meet our demands of a delay span of 6 ns and a bandwidth[1] of at least 300 MHz. Consequently, a new device had to be designed that allowed a continuous variation of delay of an analog signal. Again, this had to be simple, robust and cheap, like the other components of the Sum-trigger. The development of this continuously adjustable analog delay line was a major part of my work.

3.1. Basic concept of the adjustable delay line

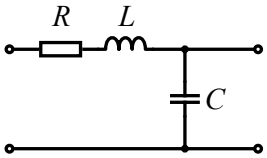
An electrical high frequency signal traveling through a (short) coaxial cable is only little distorted and damped but delayed by the time it needs to propagate from one end to the other. The basic idea of an analog delay line is to maximize the delaying feature while keeping the signal's properties intact. Electrically, an infinitesimal section of a coaxial cable can be expressed by a simplified replacement circuit (figure 3.1 (a)) of a series resistance and inductance and a shunt capacitance [16]. The values of these elementary components determine the bandwidth, impedance and signal propagation time of the transmission line. According to

$$T_{delay} = \sqrt{LC} \tag{3.1}$$

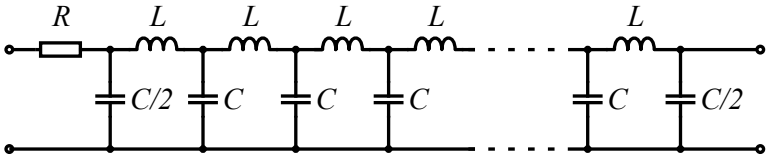
[24, 25] the total signal delay depends on both, the inductance and capacitance. Consequently, to vary the delay one either needs to change L or C[2]. Since there is no compact electronic component available that allows to alter its inductance electrically (by voltage or current), the only solution is to vary the capacitance and utilize a variable capacity diode (also called "varactor diode" or "varicap"), which

[1] Here *bandwidth* is sloppy language when actually the upper frequency limit is meant. But I will stick to this word because it is commonly used in this context. For the Sum-trigger pulse components below 1 MHz anyway are not relevant, only the higher frequencies are interesting.

[2] Actually both, L *and* C should be varied in order to avoid a changing impedance (equation 3.3).



(a) Equivalent circuit for a coaxial cable



(b) Circuit of an analog delay line

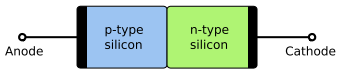
Figure 3.1.: Equivalent circuits of a coaxial cable. An infinitesimal coaxial cable element can be expressed by the simplified circuit in (a): a serial ohmic resistance R of the metal conductor, a serial inductance L inherent in any wire and a parallel capacitance C provoked by the spatial vicinity of the inner and outer conductor. Joining together several of these equivalent circuits gives an analog delay line (b). The capacitor with half a value of C at each end of the delay line terminates the line in order to reduce the frequency dependence of the delay.

changes its capacity according to a voltage applied in reverse-bias[3]. Thus the signal delay just depends on a *direct current* (DC) voltage.

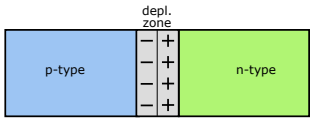
The basic replacement circuit for a transmission line shown in figure 3.1 (a) is nothing else than a passive low pass filter of second order because two bandwidth dependent elements (serial inductor and shunt capacitor) are put together [24, 16]. A low pass filter does not only suppress higher frequencies but it also shifts the phase of an incoming sinusoidal signal, which is tantamount to a time delay of that signal. In matters of high bandwidths it is not possible to use just one stage, meaning using only one capacitor and one inductor, when generating delays in the order of nanoseconds. The reason is simply the strongly bandwidth limiting property of a low pass filter: The larger the phase shift the lower the frequency limit. Furthermore, the phase can only be shifted by less than 180∘ per stage [26]. It is much more efficient to put several stages in a row to form a delay line, with each single section performing only a small phase shift but allowing a high bandwidth (figure 3.1 (b)). This way a signal's phase is shifted a bit more from stage to stage while propagating through the circuit. In order to get a larger delay one has to raise the phase shift per section which reduces the bandwidth. Thus for maximum delay the delay line has the lowest bandwidth.

Unfortunately the calculation of the properties of such a passive low pass filter of N-th order (N bandwidth dependent elements, so $N/2$ stages) is becoming impractical when adding more and more stages because each inductor and capacitor

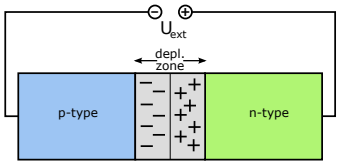
[3]Voltage in *reverse-bias* means to apply a voltage to a diode "reversely", such that no current flows.



(a) Standard diode



(b) Depletion zone



(c) Reverse-biased

(d) Varactor symbol

Figure 3.2.: Principle of a semiconductor diode. A diode consists of an interface layer of negatively and positively doped semiconductors (a). Due to local electron-hole recombinations in the p-n junction a depletion layer (depleted of charge carriers) emerges which behaves like an insulator (b). If an external reverse voltage is applied, the electric field "expands" the depletion region (c), lowering the intrinsic capacity of the diode. In (d) the graphical symbol of a varactor diode is shown. Drawings based on images from Wikimedia Commons (http://commons.wikimedia.org).

electromagnetically influences all the other ones in the delay line. As a consequence I used a circuit simulation software to derive the optimal values for the inductors and varactor diodes by systematically probing all parameter of the delay line model.

3.2. The varactor diode, a voltage controlled capacitor

The key element of the *adjustable* analog delay line is the variable capacity diode, enabling the variation of the delay. A diode is formed by joining a positive-doped and a negative-doped semiconductor together creating a p-n junction (figure 3.2 (a)). Conduction electrons can flow from the n-type side to the p-type side (forward current) but not vice versa (reverse direction). Basically every diode has a variable capacity when used in a reverse-bias setup (figure 3.2 (c)). This is due to the widening of the depletion layer between the p-n junction induced by the electric field of the applied reverse voltage. Like the capacitance of a classical capacitor depends on the distance of the conductive plates, a diode's capacity is defined by the width of the depletion region. The higher the reverse voltage, the wider the depletion zone, the smaller the capacitance (see equation 3.2). Compared to a regular diode a

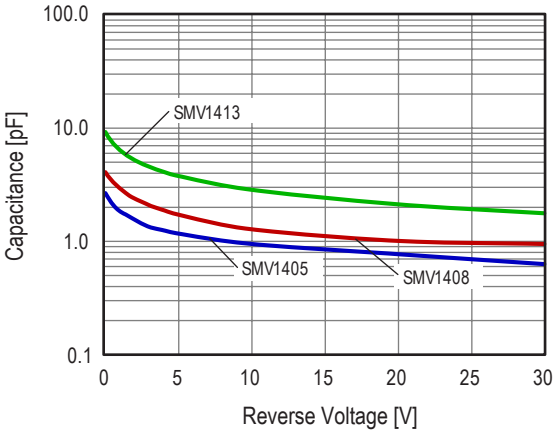


Figure 3.3.: Capacity-voltage relation of three different varactors from the company Skyworks. Mind the logarithmic scale of the vertical axis: close to 0 Volts is the largest change in capacity. Chart taken from the data sheet of *Skyworks SMV1405-SMV1413 series* varactor diodes.

varactor diode is optimized for wider capacity ranges and specific voltage-capacity characteristics, which are produced through different doping density profiles around the p-n junction.

For the capacity of a general varactor diode the following relation holds:

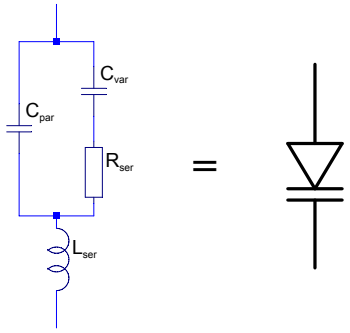
$$C(U_r) = \frac{C_0}{\left(1 + \frac{|U_r|}{\varphi}\right)^{\gamma}} \tag{3.2}$$

where C_0, φ and γ are diode specific constants and U_r is the applied reverse voltage [27, 28]. Examples of capacity-voltage curves are shown in figure 3.3.

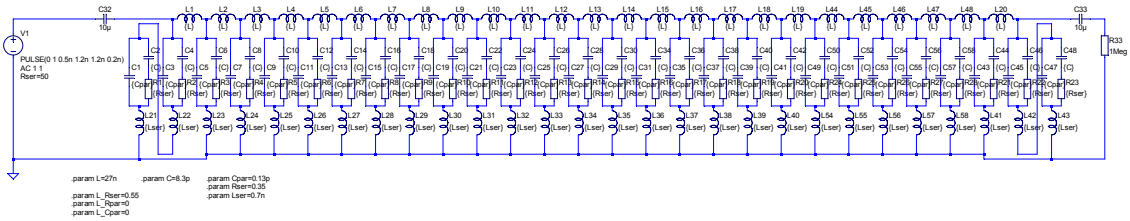
3.3. Simulation and selection of components

As already mentioned, the theoretical modeling of an analog delay line with many stages is rather extensive and would go beyond the scope of this thesis. Thus I chose to simulate the delay line with the circuit design simulation software *LTspice* from *Linear Technology*, which is freely available on http://www.linear.com/designtools/software/. In this way I could test all critical parameters of the inductors and varactor diodes to optimize the delay line for a good trade-off between high bandwidth and big delay span. A challenging task was to achieve a viable solution with the limited selection of suitable parts available on the market.

With LTspice I emulated a delay line circuit consisting of 25 identical inductors and 28 identical varactor diodes. The higher number of varactor diodes compared to the number of inductors comes from the termination of both ends of the delay line: to minimize the frequency dependence of the propagation speed a delay line needs terminating elements with one half the value of the components in each stage (figure 3.1 (b)) [29]. A varactor element of half capacitance is easily built with two



(a) Varactor model



(b) Simulated delay line

Figure 3.4.: Simulation model of a varactor (a) and delay line circuit (b) made in LTspice. The varactor model consists of a serial resistance R_{ser} and inductance L_{ser}, a fixed parallel capacitance C_{par} and of course the variable capacity C_{var}. These values are specific for each type of varactor. In the complete simulation circuit of the delay line the varactor model is already incorporated (b).

varactor diodes in series because for capacitors in series the inverse capacitance adds up: $\frac{1}{C_{total}} = \frac{1}{C_1} + \frac{1}{C_2}$ with $C_1 = C_2$ and thus $C_{total} = \frac{C_1}{2}$.

For the simulation the varactor diodes had to be modeled as realistic as possible. The model I used is proposed by *Skyworks*, one manufacturer of varactors (figure 3.4 (a)). Unfortunately only very few manufacturers provide the values necessary to correctly simulate their varactors.

The complete delay line simulation circuit consisting of 25 delay sections including the varactor model is shown in figure 3.4 (b). The parameters of the inductors and varactor diodes were set using variables defined in the parameter set below the schematic in order to ease the change of properties of all components at once.

As it turned out during the test of several parameters of the components, the varactor's serial resistance is the most critical one concerning the bandwidth. The lower the parasitic serial resistance the higher the achievable cut-off frequency.

Consequently, I tried to find varactors with the lowest serial resistance on the market. Additionally, having many delay stages the minimal capacitance per section should be low not to limit the bandwidth too much. Also due to the large number N of stages a large variability of the capacitance is less important because the parallel capacities sum up ($C_{total} = N \cdot C$) making the total delay T_{delay} scale with $\sqrt{N}$. This way more stages compensate a lack in capacitive variability. With these prerequisites I found only two possibly suitable varactor diodes: The *SMV1413* from *Skyworks Solutions, Inc.* (green curve in figure 3.3) and the *BB149A* from

NXP/Philips Semiconductors. Both offer a low serial resistance (0.35 Ω and 0.6 Ω respectively), low minimal capacity (1.7 pF and 2.1 pF) and a moderate capacity ratio (factor 6 and factor 9) still large enough to span a total delay range of at least 6 ns. They have a reverse voltage range from 0 to 30 V with the biggest change in capacity close to 0 V.

3.3.1. Impedance of the delay line

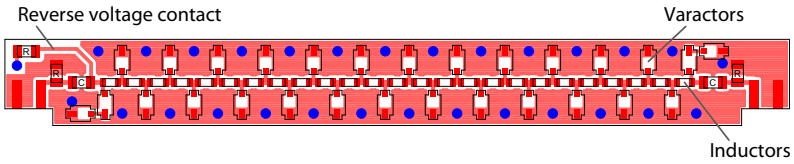
A major problem of such a tunable delay line is the change of the impedance when varying the capacity of the varactor, according to

$$Z_0 = \sqrt{\frac{L_{total}}{C_{total}}} \tag{3.3}$$

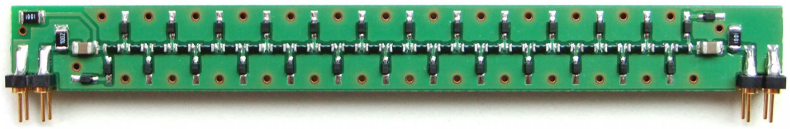
where Z_0 is the impedance of the delay line, and L_{total} and C_{total} are the total inductance in μH and capacity in pF [24]. This approximated relation only holds, when the phase shift per section is small [30], which is the case for the delay line prototype. Not having a fixed impedance there will always be unwanted reflections on the line traveling back and forth when put in an environment with a well defined impedance, usually 50 Ω. In the simulation and later on with the tests of the prototype I found out however, that these reflections are damped enough such that they are not problematic for the operation of the Sum-trigger. A reflection that occurs at the end of the delay line has to travel two more times through the circuit (back and forth again, including another reflection at the beginning of the line) than the actual pulse in order to appear at the output. Thus the reflected wave experiences two more times the line loss, which is mainly defined by the serial resistance of the inductors, which equivalents to the inner conductor of a coaxial cable. With this observation in mind I chose to use inductors with a slightly higher serial ohmic resistance of 0.55 Ω to reduce the reflections to an unproblematic level while keeping the propagation loss of the actual signal acceptable (less than 30 % of the incoming pulse amplitude).

Along with the changing impedance another downside of the tuning process arises: the amplitude of the outgoing signal varies with the delay. For the functionality of the Sum-trigger this would not be a serious difficulty because the different amplitudes would be compensated in the amplitude flatfielding process afterwards (see chapter 2.4.3). However it would be preferable to keep the overall change in amplitude as low as possible in order to avoid the necessity of a flatfielding procedure after each delay correction. The variation of the pulse heights is the result of reflections: the energy inherent in a signal is partially transfered into the reflected wave. The larger the reflected signal due to impedance mismatches, the smaller the actual pulse amplitude at the output. To achieve a minimal overall amplitude change it is advisable to chose the inductor and varactor properties in a way such that for medium delay the line impedance matches the impedances of the adjacent network.

Fortunately, the delay line circuit equipped with the Skyworks SMV1413 varactors shows only very small deviation from the maximum output pulse amplitude for the whole delay span. The reason for this is not yet completely clear and will be further investigated in the future.



(a) Delay line PCB Layout



(b) Image of the delay line module

Figure 3.5.: The delay line module. In (a) the layout of the PCB is shown. The top copper layer (red) is electrically connected to the bottom ground layer through many vias (blue) to enable a good current reflow. The centerline consists of the serially connected inductors and is alternating current (AC) coupled with a capacitor ("C") on both ends. To the inductors, the bias voltage is applied. Next to each capacitor is a resistor ("R") from the signal path to ground for impedance adjustments and to keep the input and output of the signal line free from offsets, which occur when the capacitors are recharged during a raise of the reverse voltage. The photograph of the module (b) shows the final delay line prototype equipped with the very small Skyworks SMV1413 varactors.

During the simulation phase many different delay line setups were tested, for example a varying number of sections, the influence of inductors and varactors of increasing or decreasing values along the line and several terminations. The current configuration yielded the best results concerning bandwidth and delay range. The final number of 25 stages was chosen because of two reasons. First of all, because of space limitations: The prototype delay line circuit board had to be shorter than 9 cm in order to fit on the main prototype board. Second, a larger number did not show significant improvements in the simulation and would have made the fabrication difficult since components with suitable values are barely available.

3.4. Testing of delay line prototypes

The optimal delay line circuit derived in the simulation process was manufactured for verification tests. I decided to put the circuit on a small separate PCB, pluggable onto the main board as a piggy-back board to make it easily exchangeable for testing. For the schematic and layout design I used the *PADS* software package from *Mentor Graphics*. The separate delay line PCB had to be as compact as possible due to the limited space available on the main board. Its outline was restricted to 9 mm in height and 85 mm in length. Two different delay lines were made using the same PCB: one version was equipped with Skyworks SMV1413 varactors and inductors with 27 nH, the other with BB149A varactors from NXP together with 6.8 nH inductors. For the final tests the module with the Skyworks varactors was

chosen because it showed a slightly better frequency response and the amplitude was nearly constant over the whole delay range. The NXP module showed a bigger total delay span than necessary.

The basic test setup consisted of a fast pulse generator producing a pulse with 1.7 ns *full width at half maximum* (FWHM) and alterable amplitude which was fed into the delay line, a 3 GHz LeCroy oscilloscope recording the output of the line with 20 giga samples (GS) per second, and an adjustable precision voltage supply providing the reverse voltage. The oscilloscope was triggered by the trigger output of the pulse generator. Further tests were carried out with a network analyzer from Agilent Technologies, measuring the frequency response and impedance of the delay line circuit. At least three Skyworks modules were tested with the following results:

- **Temperature stability.** A simple temperature dependency test with ice spray revealed that at least between 5 °C and 30 °C the delay does not change for any fixed reverse voltage, taking into account the measuring accuracy.

- **Precision of delay adjustment.** Setting the voltage back to a previously measured value resulted in the same delay, not showing any hysteresis. Thus the accuracy in setting the delay seems to depend solely on the applied voltage.

- **Tolerances among the various modules.** Each module showed the same time delay for a given reverse voltage. No variance could be seen. Apparently the tolerances of the components "average out" for the whole module.

- **Bandwidth.** According to the network analyzer the maximum transmitted frequency (−3 dB limit[4]) is only 190 MHz for the worst case (maximum delay), see figure 3.6, which does barely meet the requirements for the Sum-trigger. In contrast, regarding the pulse shapes (figure 3.7) the FWHM for a minimally delayed pulse is the same as the input pulse. Even the output pulse with maximum delay shows just a slight widening of 0.2 ns compared to the input signal. However, comparing the FWHM of the pulses is questionable because their shapes diverge from a Gaussian curve and differ among each other. Hence it is much more conclusive to evaluate the change of the pulse's maximal slope. Using the signal rise time (measured from 10 % to 90 % of amplitude) the bandwidth of a delay line can be approximated with

$$f_{limit}\,[\mathrm{GHz}] \approx \frac{0.35}{t_{rise}\,[\mathrm{ns}]} \qquad t_{rise} = \sqrt{t^2_{riseOutput} - t^2_{riseInput}} \tag{3.4}$$

  [24], yielding ≈ 300 MHz for maximum signal delay and even ≈ 420 MHz for minimal delay.

  The discrepancy of these results compared to the measurement with the network analyzer has a simple explanation: the frequency spectrum of the pulses barely contains frequencies below ≈ 50 MHz. Hence, the effective −3 dB limit in the frequency response plot (figure 3.6) should not be applied with respect to the attenuation of the lowest frequencies (1 MHz) but to frequencies above

[4] 3 dB means a change in amplitude by a factor of $\sqrt{2}$ or the signal's power by a factor of 2.

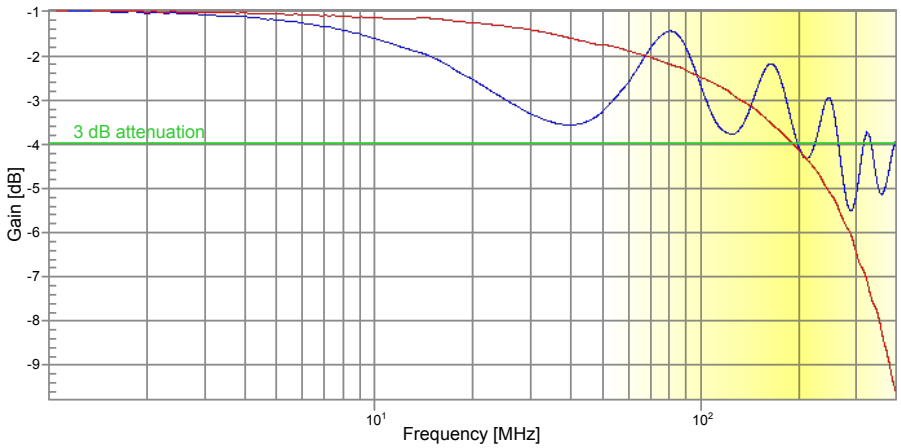


Figure 3.6.: Frequency response of the delay line module measured with a calibrated network analyzer. The red curve shows the response for maximum delay, whereas the blue one represents minimum delay. Here the 3 dB attenuation (base line at -1 dB) crosses the red line already at 190 MHz. Since the pulses are composed mainly of frequencies above ≈ 50 MHz (yellow area) the zero level of the -3 dB limit can be shifted further down (to ≈ -2 dB), yielding significantly higher cutoff frequencies. It also can be seen that the impedance of the measured line is optimized for large delays: the red curve is very smooth in contrast to the response for small delays which shows many humps coming from reflections on the line.

≈ 50 MHz. This way, the limit in the graph can be lowered by ≈ 1 dB. Then, the resulting cutoff frequencies are similar to the higher values derived in the direct pulse shape analysis.

- **Total span of delay.** Also visible in figure 3.7 is the dynamic range of the delay of 6 ns which is more than enough for the Sum-trigger. Actually even a smaller delay span (and thus higher bandwidth) would be fine because just very few pixels show a transition time deviating from the average of all channels of up to ± 3 ns. These outliers could be handled with fixed delay lines before the actual circuit. Throughout the rest of this thesis I will refer to relative delay values, defining 0 ns as minimum delay.

3.5. Further optimizations

As already mentioned the impedance changes with the delay, producing reflections of the signal. These reflections should be analyzed and suppressed further or an environment has to be created where they do not affect the adjacent signal paths, like placing the delay line in between two operational amplifiers[5] (OpAmps). Also,

[5] An *operational amplifier* is an optimized *integrated circuit* (IC) with two inputs (one inverting and one non-inverting) and one output. The circuit regulates the output voltage to match the

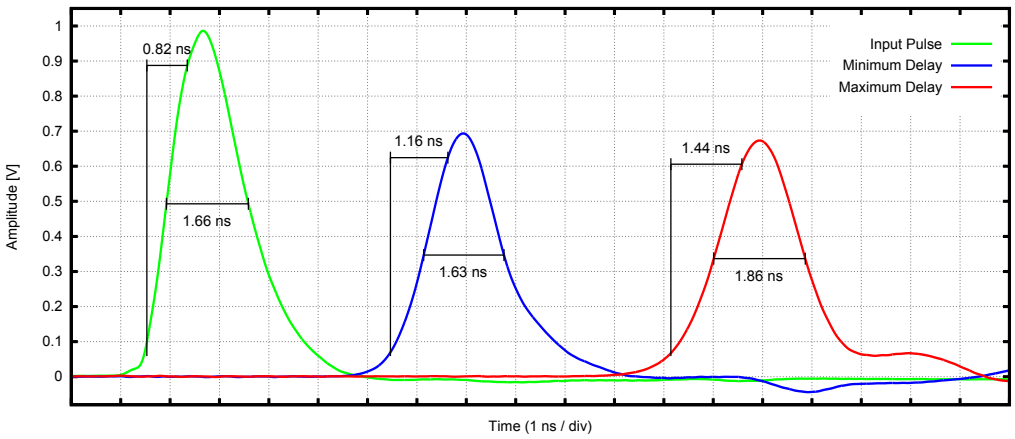


Figure 3.7.: Measured incident and outgoing pulse shapes of the delay line. In case of minimal delay (blue pulse) the incident pulse (green) is not widened at all, only the signal rise time (1.16 ns) is slightly longer. The maximally delayed pulse (red) is broader but still inside the tolerance limit. The trailing hump might come from oscillations along the transmission line and is small enough compared to the actual pulse that it does not affect the operation of the Sum-trigger.

the bandwidth could be optimized through a different choice of components and improved PCB layout which might also help to prevent possibly occurring oscillations.

difference of the two input voltages. The inputs have an extraordinary high input resistance in the order of several MΩ or even GΩ such that in the ideal case no current flows. In contrary, the output has a very low resistance, ideally 0 Ω.

4. Sum-trigger prototype setup

The prototype of the new Sum-trigger developed in this thesis essentially consists of four different printed circuit boards, named according to their main function: The "*clipping board*", the "*summing board*", the "*discriminator board*" and the "*computer control board*". The analog summing boards were taken from the current Sum-trigger installation without changes. The three other boards I re-designed completely based on the analog circuits of the old Sum-trigger schematics. The clipping board comprises eight channels with identical circuits, where only the first one is split in two outputs, one being used as the separate timing reference channel to which the delays of the other channels are adjusted. All eight output signals are added up using one summing board, which is only a simple analog summing amplifier circuit with eight inputs and three identical outputs. The discriminator board has three equivalent circuits, two of them for the summed signals (whereas only one was used for the tests) and one for the timing reference.

The main purpose of the prototype was to elaborate and to prove the new conceptual design of the Sum-trigger with automatic adjustment of gains and delays, while improving the analog part of the current system. Finally, several tests to verify the functionality and to evaluate the performance were accomplished.

4.1. Block diagram of the Sum-trigger prototype

The functional block diagram of the Sum-trigger prototype is shown in figure 4.1. In the following I will explain the basic structure of the circuit boards in more detail. For the full schematics including all components and values the reader is referred to appendix A.

4.1.1. Clipping board

From the trigger receiver unit (see figure 1.12) the differentially transmitted[1] signals first are routed to the clipping board. There, the weak pulses (in the order of 1 $\frac{\mathrm{mV}}{\mathrm{Phe}}$) are converted from differential to single-ended ones and amplified in two steps by a total factor of 20 to reduce the influence of external noise along the analog signal path. In order to switch off channels for the calibration process (measuring one channel after the other) the amplifiers can be disabled. The amplified pulses then pass through the delay line module and the attenuator chip where they are trimmed

[1] Differential signaling is an efficient way to improve noise immunity of a transmission path, especially when transmitting low-voltage signals. The signal is transfered via two conductors in close vicinity to each other. One wire carries the signal in its primary polarity, the other wire the inverse of it. At the end of the line the receiving device reads the difference between the two signals, revealing the original waveform. This way, electromagnetic pickup, which acts in the identical way on both conductors is canceled out.

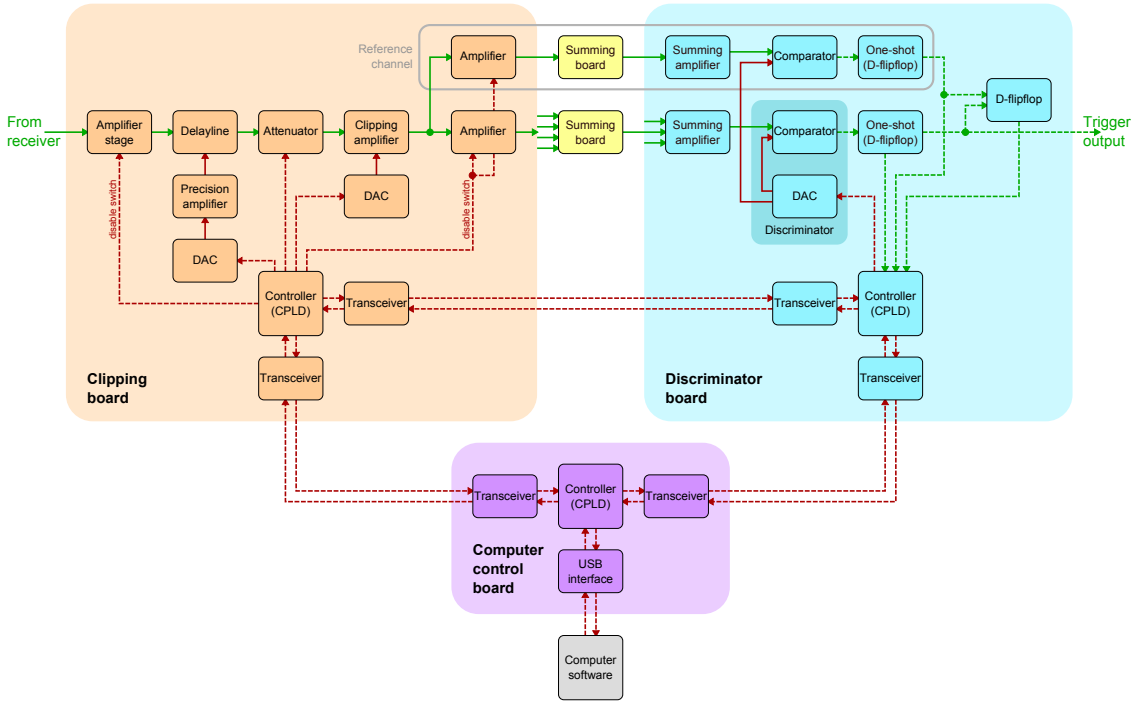


Figure 4.1.: Simplified functional block diagram of the Sum-trigger prototype for one channel, including the reference channel. Solid lines symbolize analog signals, dashed lines discrete signals or digital data. The paths of the signals coming out of the receiver board are marked with green arrows, signals for adjustment, setting and communication with red ones.

properly in amplitude before the clipping amplifier. The clipping voltage is provided by a digital-to-analog converter (DAC). After the clipping the first channel on the board is split to sideline the timing reference channel, which passes through the same stages as all other channels but on a separate path, to make the pulse undergo the same transformations according to delay and amplitude. At the end of the clipping board, amplifiers with a total gain of 1 act as line driver for the coaxial cable connections to the summing board.

4.1.2. Discriminator board

The summed signal is fed into the discriminator board, where only one input of the summing input amplifier was used for the tests. The following entity is the discriminator, consisting of a comparator and a DAC to supply the reference voltage (discriminator threshold). The comparator's very short digital ECL output pulse is stretched by the adjacent one-shot[2] circuit to roughly 60 ns. Otherwise the pulse would be too short (only few ns) for the following electronics to sense it properly. Here, the digital signal is divided up into two paths: for the amplitude measurements

[2]In digital electronics a circuit that keeps its output at a logic HIGH for a fixed amount of time after its input was triggered by a rising (or falling) edge of a pulse is called *monostable multivibrator* or *one-shot*.

one is directly connected to the *complex programmable logic device* (CPLD) which contains the rate counting unit and the communication logic. The other path, together with the reference channel, first goes to the D-flipflop and then also to the CPLD to carry out the timing measurements.

4.1.3. Computer control board

The computer control board only houses the digital components necessary for data transfer (CPD and transceiver) and a USB interface module for the communication with a standard computer.

On each board (except the summing boards) a central CPLD takes over the communication on and in between the boards and executes the board specific functions. For the data transfer from board to board industry standard RS-485 transceiver chips are used as line drivers to enable the usage of regular twisted-pair Ethernet cables for the transmission even over long distances.

On the clipping board the voltage required to adjust the delay line is provided by a DAC whose output is amplified by a high precision operational amplifier (OpAmp) in oder to achieve a total range from 0.0 V up to 25 V. Likewise, the reference clipping voltage for the clipping amplifier is provided by the same type of highly precise and stable DAC, though without a separate amplifier since maximally only 2.5 V are required.

4.2. Electronics and components

The general criteria for the selection of electronic components were a sufficient bandwidth and high speed, low power consumption, price and availability in manageable packages for later re-soldering in case of design errors.

4.2.1. Amplifier stages

For the amplification of the analog signal at the input and at the output stage of the clipping board, as well as for the summing circuits, the ultra-wideband current-feedback operational amplifier "OPA695" from *Texas Instruments* is used. It offers high bandwidth (up to 450 MHz at a gain of 8) amplification with low noise and a very high slew rate[3] of 4300 V/µs. The OPA695's properties are well understood as it is utilized in many subsystems of the MAGIC telescopes, also in the old Sum-trigger circuits. A very helpful feature is the OpAmp's disable pin. If it is pulled LOW (to ground) the amplifier is switched off resulting in a very high forward path isolation of roughly 70 dB. No hint of the input signal (up to 100 mV) could be seen at the output during the Sum-trigger tests when the amplifiers were disabled. By making use of the disable feature, no separate switches were required in order to to shut down the Sum-trigger channels for the calibration process.

[3]The *slew rate* indicates how fast an amplifier can ramp up (or down) the output voltage in a certain amount of time. It is the maximum rate of change of the output for a square wave signal fed into the input. If the absolute change in the output voltage is large this is a crucial bandwidth limiting factor.

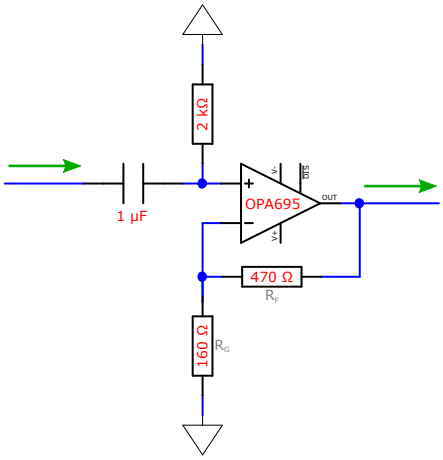


Figure 4.2.: Schematic of a non-inverting operational amplifier circuit with a gain of 4. The OPA695's output is connected to the inverting input (−) via the feedback resistor R_F. The gain G is given by $G = \frac{R_F}{R_G} + 1$. The non-inverting input (+) is AC coupled by a capacitor and leveled to ground with a 2 kΩ resistor to eliminate a *direct current* (DC) offset. For simplification the supply voltages (V+ and V-) and the disable pin ($\overline{\text{DIS}}$) are not connected in this drawing.

All OPA695s on the clipping board are used in non-inverting mode, where the feedback resistor is connected to the amplifier's inverting input (figure 4.2). Hence, the output voltages have the same polarity as the input signals. The amplifier stage on the clipping board is made up of two serially connected OpAmps. The first one receives the differential input signal from the receiver board and acts as differential-to-single-end converter with an amplification factor of 5. The second OPA695 is fully single-ended and has a gain of 4. The two step amplification results in a higher bandwidth compared to a single amplifier with the same total gain because the gains multiply, whereas the timing properties that determine the cut-off frequency just add [16]. To prevent oscillations and avoid unpredictable behavior from a constant voltage offset each amplifier's non-inverting input is AC coupled with a capacitor and tied to ground by a resistor.

4.2.2. DACs

For the adjustment of the delay line, the clipping level, and the discriminator threshold a precisely tunable voltage source is required. For this purpose I found the "DAC7568" from *Texas Instruments* as a perfect candidate. It offers 8 independent output channels with 12 bit resolution (equals 4096 equidistant voltage steps), very low offset voltage, optimal linearity, and excellent temperature stability. Additionally it has an highly accurate internal voltage reference, and an SPI interface for fast and easy data transfer from the CPLD. The DAC7568 comes in two version: a type A with an output voltage ranging from 0 to 2.5 V and a type C with a maximal output of 5 V. The A type is used to provide the clipping voltages and the

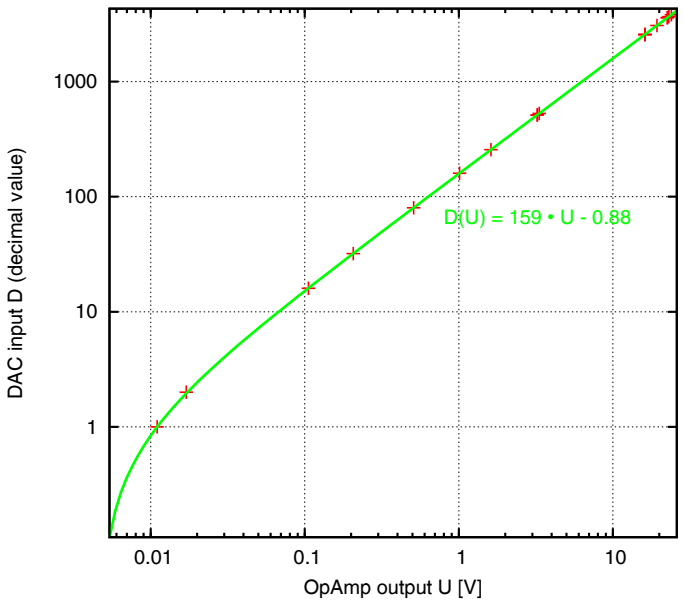


Figure 4.3.: Measurement of the decimal input value set at the DAC7568-C versus the output voltage of the OPA4277. Mind the double logarithmic scale. A straight line (green) has been fitted to the data points (red crosses). The curved section comes from the small zero point offset (≈ 1 mV) of the DAC. Remarkable is the perfect linearity of both components combined and the absence of measurable hysteresis (several points were re-measured, approached from higher and lower values) even for the very low voltages.

discriminator voltages, the type C controls the bias voltage for the delay lines.

4.2.3. Adjustable delay line and bias voltage

As already described in chapter 3, the analog delay line allows the continuous variation of a signal's delay by applying a voltage between 0 and 30 V. For high precision and long-term stability of the delay setting the voltage should be adjustable in very fine steps and must not drift significantly over a wide temperature range and a long period of time (at least several months). To achieve this I decided to use the highly accurate DAC DAC7568-C together with a high precision operational amplifier (*Texas Instruments* "OPA4277") capable of handling the large voltage span of 0 to 30 V. The OPA4277 contains four precision OpAmp circuits in one housing. Thus, to adjust eight delay lines on one board, one DAC and two OPA4277 are required. To control the delay line the OpAmp circuits of the OPA4277 operate in non-inverting mode, with high-ohmic (10 kΩ) feedback resistors to avoid a large current flow even at the very high output level of 30 V. In tests, the optimal gain to span the maximal voltage range was found to be factor 5.2. As mentioned in chapter 3.3 the delay changes at most at very low voltages close to 0 V. Consequently, to get an adequate span between minimum and maximum delay it is essential to approach 0 V as close as possible, whereas voltages above 25 V do not noticeably

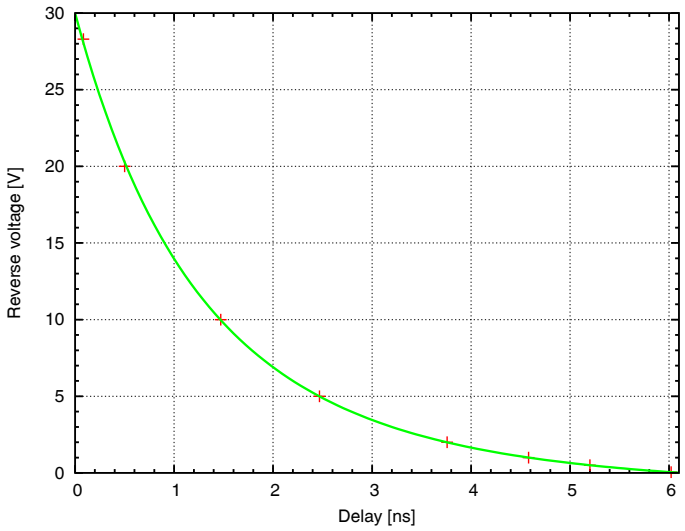


Figure 4.4.: Measurement of the reverse voltage applied to the delay line versus the delay. Equation 3.2 plugged into the relation 3.1 has been fitted (green curve) to the data points (red crosses). Obviously, the model (3.2) correctly describes the voltage dependency of the utilized varactor diodes.

contribute. Accordingly, the supply voltages for the OPA4277 were chosen to have a small negative component (−5 V and 25 V) such that the amplifiers can actually level down to 5 mV. Even less would be easily possible but this is inhibited by the DAC whose outputs can not be set much lower than 1 mV. Measurements revealed that the combination of both components shows perfect linearity and no observable hysteresis (figure 4.3.)

In order to find the conversion from the required DAC input value to a certain delay I first measured the delay for various reverse voltages and fitted equation 3.2 plugged into the relation 3.1 to the data. Then I combined the result with the equation derived from the fit shown in figure 4.3. In this way one can find the decimal value D that has to be set at the DAC to obtain a certain (relative) delay T_{delay}[ns], by the formula

$$D = 159 \cdot \varphi \left[\left(\frac{L \cdot C_0}{(T_{delay} + T_{offset})^2} \right)^{\frac{1}{\gamma}} - 1 \right] - 0.88 \tag{4.1}$$

where $\varphi \approx 1.05$, $\mathrm{L} \cdot \mathrm{C}_0 \approx 164$ (ns)^2, $\mathrm{T}_{\mathrm{offset}} = 6.67$ ns, and $\gamma \approx 0.38$.

4.2.4. Attenuator

For the amplitude flatfielding a component capable of varying the signal's amplitude was needed. Unfortunately no sufficiently fast *variable gain amplifier* (VGA) with properly tunable gain was available on the market. Internally virtually all VGAs are composed of an amplifier with a fixed gain and a subsequent adjustable attenuator. Hence, instead of a VGA I just installed a digitally adjustable attenuator, since

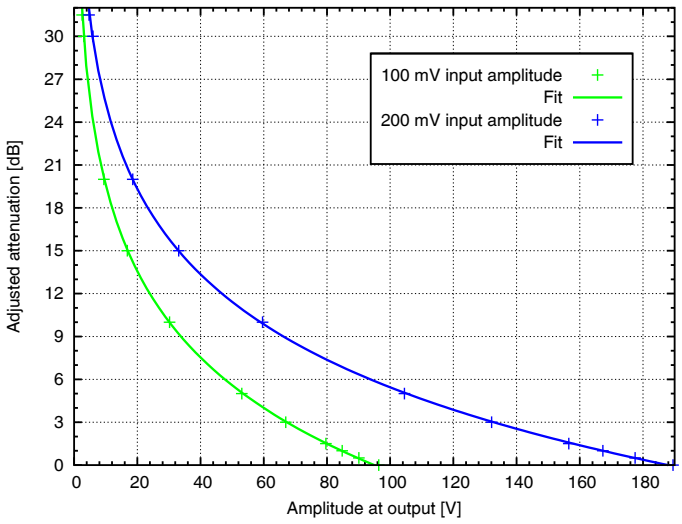


Figure 4.5.: Plot of the attenuator setting (G) in dB versus the measured output pulse amplitude U for input signals with two different amplitudes U_0. The data points (crosses) comply with the relation $G \propto -\log_{10}\left(\frac{-U}{U_0}\right)$ (continuous lines).

fixed gain amplifiers are already included in the clipping board circuit before the attenuator. The requirements of a wide attenuation range and fine tuning could be met by the digital step attenuator "DAT-31R5-SP+" from *Mini-Circuits*, which provides an attenuation up to 31.5 dB adjustable in 0.5 dB steps, and a 6 bit serial interface very similar to SPI. Its high bandwidth from DC up to 2.4 GHz is more than sufficient and the insertion loss of about 1.3 dB is unproblematic.

To calibrate the amplitudes it is appropriate to test the pulses at equidistant amplitude levels, since the finer (absolute) segmentation of the logarithmic dB scale for lower amplitudes is not required. Hence I derived a conversion from the dB scale to a linear scale expressing the attenuation in percent of the non-attenuated pulse amplitude. The formula

$$G = -\alpha \log_{10}\left(\frac{-x[\mathrm{mV}] + 100}{100\ \mathrm{mV}}\right)$$

delivers the value G in dB to which the attenuator has to be set in order to attenuate the signal by x % of the unattenuated amplitude.
Fits to measurements of the output signal of the attenuator at various attenuation steps for input pulses with constant amplitudes (figure 4.5) showed, that the factor α has to be adjusted to 17.3 (instead of usually 20) to correctly describe the properties of the attenuator.

4.2.5. Clipping amplifier

The clipping amplifier in the old Sum-trigger circuit was the key bandwidth limiting factor. The used OpAmp (*Texas Instruments* "OPA699") has an upper frequency limit of only 260 MHz and a low slew rate which reduces the overall bandwidth

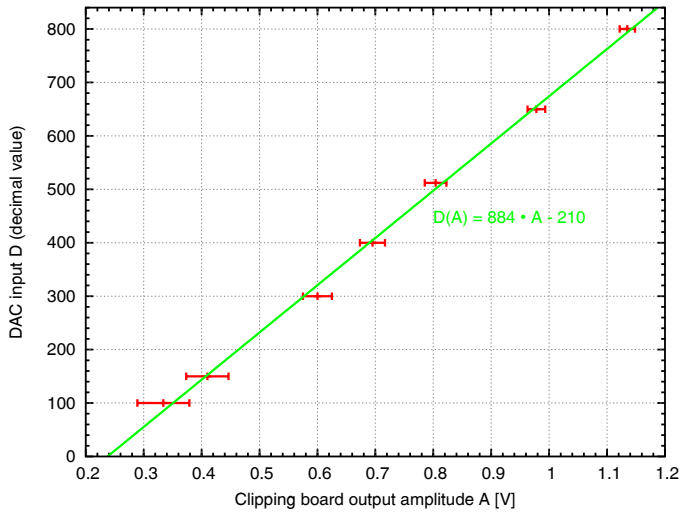


Figure 4.6.: Measurement of the clipping performance of the LMH6553 using calibration pulses. The clipping voltage is set with a DAC7568-A, whose decimal input values label the ordinate. The error bars are just roughly estimated regarding the reading accuracy and the variance of the amplitudes. For smaller pulses the dispersal got larger. A straight line (green) has been fitted to the data to retrieve the proper DAC value for a certain clipping voltage.

significantly and thus widens the pulses. As a substitution I picked the "LMH6553" from *National Semiconductor* with up to 900 MHz bandwidth. Unfortunately it is a fully differential amplifier, so a adapting circuit had to be designed enabling its usage in a fully single-ended environment. The circuit strongly affects the performance of the amplifier and it becomes problematic to clip signals below a threshold of 500 mV when the input pulses are large (see figure 4.6 and chapter 5.2.3). For this reason the amplifier was put at the very end of the clipping board, where the signal is already amplified enough.

4.2.6. Discriminator (comparator)

The discriminating circuit on the discriminator board consists of the ultra-high speed ECL comparator "MAX9600" from *Maxim Integrated Products* and a DAC7568-A which provides the reference voltage. The MAX9600 has already been successfully applied in the old Sum-trigger. It has a very low propagation delay skew (10 ps) and dispersion (30 ps), ideal for tracking of narrow pulses, supported by a low timing dispersion.

4.2.7. One-shot (multivibrator)

For the subsequent electronics the ultra-fast output pulses ($t_{HIGH} < 10$ ns) of the comparator were too short to be processed properly. Accordingly, a one-shot circuit is implemented after the discriminator to extend the pulses to roughly 60 ns.

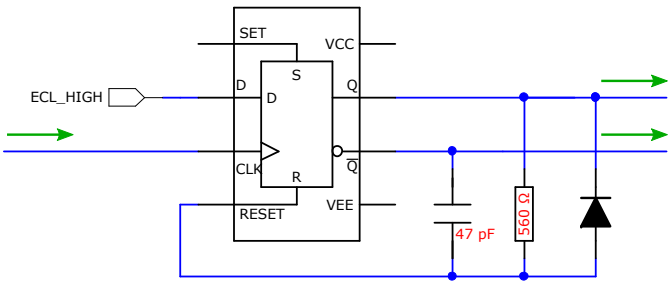


Figure 4.7.: Monostable one-shot circuit used to widen the ECL trigger pulse. The flipflop's D input is permanently set HIGH and the clock input (clk) is sensitive to the rising edge of an incoming signal. Both, the regular (Q) and the inverted output ($\bar{Q}$) are used by the adjoining components. The supply voltages are omitted in the drawing.

A one-shot is a monostable multivibrator generating a single output pulse of a specified width when a trigger pulse is applied. The trigger signal initiates a timing cycle which causes the output of the multivibrator to change its state at the start of the cycle and remains in this second state, which is determined by the time constant of the capacitor C and the resistor R until it resets itself back to its original (stable) state. Such a behavior can be achieved by attaching an RC element to a D-flipflop (figure 4.7). Since the circuit has to cope with an extremely short ECL trigger pulse, a very fast ECL-based D-flipflop ("MC100EP31" from *ON Semiconductor*) is necessary. In idle state the flipflop's output Q is LOW and its inverse $\bar{Q}$ is HIGH (in terms of ECL a HIGH level equals -0.9 V and LOW -1.75 V). When the clock input is triggered by an incoming pulse the output Q goes HIGH and $\bar{Q}$ LOW. Consequently, the capacitor is charged via the resistor until the voltage at the capacitor reaches the threshold to activate the flipflop's reset input. Then, the outputs Q and $\bar{Q}$ switch state and the capacitor is immediately discharged through the diode, setting the circuit back to the initial state. Hence, the values of R and C and the threshold of the reset input define the width of the output pulses.

The (differential) ECL output of the one-shot is connected to both, the discriminator board's central CPLD via a high speed ECL-to-TTL converter ("MC100EPT25" from *ON Semiconductor*) to count the rates for the amplitude measuring process, and to the timing measurement logic to determine the delay of the signal.

4.2.8. Timing measurement unit

In order to differentiate the arrival times of the adjusted pulse and the reference pulse with high temporal precision, the same ultra-fast ECL D-flipflop type MC100EP31 is used as for the one-shot circuit. It offers a very short setup time[4] of 100 ps, minimizing the time window in which the flipflop is in a metastable state[5], and

[4] A flipflop's *setup time* is the minimum amount of time the data signal should be held constant before the clock event so that the data is reliably sampled by the clock.

[5] Clocked flipflops are prone to a problem called *metastability*, which happens when a data or control input is changing at the instant of the clock pulse. The result is that the output may

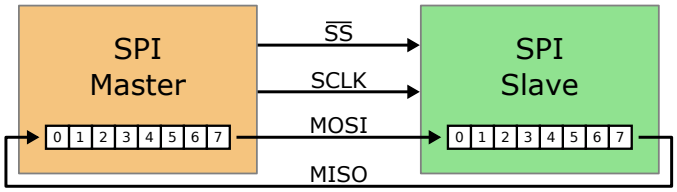


Figure 4.8.: Sketch of an SPI connection between a Master and a Slave device. Here, symbolical 8 bit shift registers are shown.

an exceptionally low cycle-to-cycle jitter of only 0.2 ps, which is a factor of $\approx 10^3$ smaller than the pulse's fluctuations and assures their thorough detection.

Like for the one-shot, also the flipflop's output is converted to TTL and connected to the counting logic inside the CPLD, where the rates are measured.

4.2.9. Board CPU

The central control and readout logic on each board (except the fully analog summing board) had to be freely and easily re-programmable for the test phase. I decided to use the EPM1270 CPLD of the "MAX II" series from *Altera*. It has 1270 logic elements and is available in a *thin quad flat pack* (TQFP) package which can be soldered by hand. For the prototype this large CPLD was picked to leave the door open for later extensions of the control logic. Easy programming is provided by the fully-featured integrated development environment "Quartus II" from Altera. Quick in-system re-programming can be done using a standard JTAG-cable. This way I can immediately test new versions of the controller logic.

4.3. Digital communication

For the readout and setting of values on the several boards the data somehow had to be transfered from the software to the hardware, from board to board and from CPLD to each adjustable component. As all adjustable electronic parts that I selected for the prototype can be addressed via the *serial peripheral interface* (SPI) I also chose a modified version of this interface to perform the transfer of data from one board to another. In that way I only had to work with one type of interface, except for the USB connection to the computer.

4.3.1. CPU-to-part communication (SPI)

For the adjustment of the attenuators and the output voltages of the DACs, SPI links between the devices and the central CPLD were used. SPI is a digital standard intended for the communication between a host processor ("Master") and a peripheral component ("Slave"). In standard mode it is using four wires: two of them are

behave unpredictably, taking many times longer than normal to settle to its correct state, or even oscillating several times before settling. However, such a behavior was never observed during the Sum-trigger tests.

control lines, namely a clock line (SCLK) and one slave select (SS), and two are serial buses, one data in (DIN) and one data out (DOUT) line. For clarification the latter two are also called *Master-out-Slave-in* (MOSI) and *Master-in-Slave-out* (MISO). Principally, the SPI works like a shift register. When SS is LOW, data bits are serially transferred out of the Master into the Slave via the MOSI line and simultaneously from the Slave to the Master via the MISO line, shifting one bit per clock cycle (figute 4.8). Thus a bi-directional link is established. After the last bit has been shifted, the SS is set back to HIGH and the data can be processed. One Master can handle many Slaves using the same clock, MISO, and MOSI lines. Only dedicated SS lines are necessary. In many cases, like the DACs and the attenuator for the Sum-trigger, a MISO line is not used since there is no need for the devices to transfer data to the Master. Then only three wires are required. The DAC and attenuator chips are directly connected to the CPLD, which acts as an SPI Master. Accordingly, the CPLD has to provide a clock signal, which was generated with an external 10 MHz quartz oscillator ("CFPS-69" from *IQD Frequency Products Ltd.*). The functional logic to make the CPLD work as SPI Master was implemented with a *very high speed integrated circuit hardware description language* (VHDL) program, which was synthesized[6] and included in the main CPLD control logic.

The 12 bit DAC7568 is fully SPI compliant and is controlled by a 32 bit long data word including control bits. The 6 bit attenuator DAT-31R5-SP+ differs slightly from the usual SPI standard: after the shifting of the data it requires the SS signal to go HIGH and then LOW again in order to acquire the data correctly.

4.3.2. Board-to-board communication (SPI derivate)

For the data transfer between the computer control board and the clipping and discriminator boards a parallel wiring scheme based on SPI was chosen. All boards are connected to the same SS, SCLK and MOSI lines in parallel, such that all discriminator and clipping boards (SPI Slaves) receive the data from the computer control board (SPI Master) simultaneously. To avoid that more than one board outputs data on the MISO wire (to which also all boards are linked), each board has an electronic switch which disconnects it from the MISO line. Solely a board, which was addressed by a previous data packet, hooks to the MISO bus and sends data to the computer control board when new data is shifted into the MOSI bus 4.9. Hence, to address a certain board, first a data packet containing the binary address of the board (and if necessary, commands that trigger certain functions on the device) has to be sent to the interface. Then, with the next sending attempt, data can be shifted from the specified board into the Master (computer control board).

For the Sum-trigger this setup has various advantages compared to a usual serial (daisy chained) SPI topology, where the MISO of a SPI Slave is connected to the MOSI of the next Slave:

- data can be sent to any board within only one write cycle and read out within two cycles, no matter how many boards are used. On a usual serial setup one needs N write cycles to shift the data up to the N-th device and even more

[6] *Synthesizing* VHDL code means to convert the structural logic of the source code into a functional assembly of logic gates and elements available on the programmable chip.

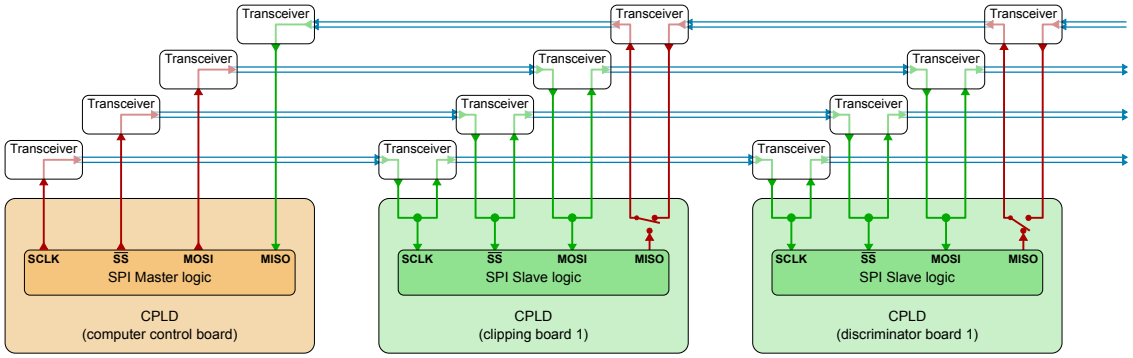


Figure 4.9.: Schematic of the board-to-board wiring based on a modified SPI architecture. The blue double lines denote differential RS-485 based board-to-board connections, for which regular Ethernet cables (with four wire pairs) are used. At the MISO terminals of the SPI Slaves switches (bus mux elements in the CPLD) hook in a Slave to the common MISO bus when it is addressed by data sent over the MOSI bus. The other SPI signal lines (SS, SCLK, MOSI) are just looped through each board's CPLD. In the drawing, exemplarily the "discriminator board 1" is active and can send data to the computer control board via the MISO bus.

cycles (depending on the number of devices in the chain) to read new data from it.

- the order of the boards in the chain does not matter. They are only identified by their address. In a usual daisy chain the control software has to be aware of the order of the devices in order to address the correct one.

- since the bits are shifted much less than in a daisy chained setup, the probability that a bit error occurs is much lower.

The electrical connection from one board to the next is realized with RS-485 transceiver chips ("MAX3490E" from *Maxim Integrated Products*). RS-485 is an electrical interface standard for digital serial data transfer via differential lines. It can be used over long distances and in electrically noisy environments. For the wiring in between the boards I use regular Ethernet cables with four twisted pairs of wires. The transceiver chips are powerful enough to drive even long cables with up to 12MBit/s (a length of 25 m was successfully tested with the prototype).

4.3.3. Board-to-computer communication (USB)

The data transfer from the computer control board to the PC and back had to be as simple as possible but should not require special hardware on the computer side. Thus I picked the "IO-Warrior56" module from *Code Mercenaries*, which was placed on the computer control board next to the CPLD. Essentially it is a very simple universal input/output controller for the USB interface and hence can be

connected to any modern computer. In normal operation it has 50 general purpose input/output pins that can be individually addressed by software. However, its key feature is a special SPI Master mode, making it a direct USB-to-SPI converter with a data rate up to 12MBit/s. On the computer side no extra driver is needed to install the IO-Warrior because on any current operating system it is identified as a generic *human interface device* (HID) which can be easily accessed from application level. In addition to that the manufacturer provides a dynamic library as a simple *application programming interface* (API) to access all IO-Warrior functions using any programming language on Windows or Linux-based operating systems.

Unfortunately, due to the architecture of the HID device class, consecutive read or write requests to the USB device can not be performed faster than every 1 or 2 ms. But this is important for the Sum-trigger setup because each sent or received data package is only 6 Bytes long. Then a new request is initiated. Between each request I had to insert a break of several ms in order to avoid unpredictable crashes of the program. This slowed down the calibration process enormously.

The SPI side of the IO-Warrior module is connected to the computer control board's CPLD, which acts as an SPI Slave on this link. Inside the CPLD the data is transfered from the Slave section to an SPI Master core, which transmits the data further to the other boards.

4.4. Programmable logic

All boards of the new Sum-trigger required a controller logic to perform the tasks necessary for measuring, readout and setting of values for the adjustment of the analog signal paths. I used the Quartus II by Altera to design, test, and compile the logic which was then transferred to each board's CPLD. The general logic was designed graphically with block diagrams. Some logic units were included using VHDL code.

As a general concept I decided to design asynchronous logic because it is simpler to implement and a strict synchronous processing was not needed. Nevertheless I had to deal with several timing issues for which I applied one-shot or positive edge detector logic, which require a clock. This is not the most beautiful way of system architecture, but it simplifies the circuit and is more intuitive.

The common section on each of the three newly developed boards is the communication part. A modified version of the SPI standard was used for the inter-board data transfer, altered to work in a parallel-wired setup (figure 4.9).

The more complex clipping and discriminator boards additionally included subroutines addressable by function codes for the measuring and adjustment procedures.

For detailed functional block diagrams refer to appendix C.

4.4.1. Structure of the control data packages

To distribute control data to the various boards and their logic subroutines and readout data from the counting unit, I use a 48 bit (6 Byte) wide data word structured in the following way:

1. The boards are selected by the first Byte (bit 48 to 41) of the data which contains the binary encoded board address (maximum number of addressable boards: $2^8 = 256$).

2. A specific function on the selected board is addressed using the next 4 bits (bit 40 to 37), representing the binary encoded function code that defines which sub-logic on the addressed board is executed (maximum number of distinct functions: $2^4 = 16$).

3. 4 bits (bit 36 to 33) are reserved for optional future use.

4. The remaining 32 bits (bit 32 to 1) consist of function specific data, such as 32 bits to control the DAC7568s, 6 bits to set the attenuation value, or 24 bits from the output of the counter.

4.4.2. Computer control logic

The computer control processor (appendix C.1) has a very simple logic circuit. It is completely "passive", meaning that it is not addressed directly and that is does not execute any special function. It only forwards the digital data packets coming in from the USB-to-SPI converter to the SPI-based board-to-board bus and vice versa. For this it incorporates an SPI Slave logic ("core") to receive/send the serial data from/to the computer side, and an SPI Master core to manage the board-to-board communication. Both, the Master and the Slave core are based on synthesizable VHDL code, available at [31] and [32], providing fully SPI compatible functionality.

The Slave core has the standard SPI lines MOSI, MISO, SCLK, and SS for the external linkage, and parallel 48 bit (6 Byte) wide Data_in and Data_out buses for the connectivity to the other logic-internal components. When the external SPI Master (USB-to-SPI converter) initiates a data transfer by setting SS to LOW, the bits parallel present at the Data_in are serially shifted out through the MISO bus, beginning with the 48th bit (*most significant bit*, MSB first). When the SPI transfer is finished, SS is set back to HIGH state and the bits that were serially shifted into the Slave via the MOSI line are output at the parallel Data_out bus, which is connected to the internal SPI Master.

Additionally to the SPI lines, the SPI Master core has a TX_Start input that triggers the serial data transfer when raised to HIGH level, and an input for an external clock source (clk), necessary to provide a clock on the SPI side (SCLK). It also has a TX_Done output which goes HIGH when the data transfer is finished. When activated by TX_Start, the Master sets its SS output LOW and bits applied at its Data_in are shifted out on the MOSI bus. Incoming data from the MISO line is given out to at Data_out bus when the transfer is done.

When the external SPI Master sent new data to the Slave, the internal Master has to be activated to transmit the data further on the board-to-board SPI link. For this purpose I used a D-flipflop element, identical in function to the device utilized for the timing measurement on the discriminator board (section 2.4.6), here in its usual manner of application. Its D input is permanently pulled HIGH (which equivalents to VCC) and its clock input, sensitive to rising edges, is connected to the external SS line. If the SS transitions into HIGH state (meaning that new data

is available at the SPI Slave's parallel output), the flipflop output (Q) pulls HIGH and activates the internal SPI Master via TX_Start. When the Master finishes the data transmission, its TX_Done goes HIGH and resets the flipflop's output through the clear input (CLRN), making the circuit ready for the next transfer cycle.
The flipflop is required as a buffering element to avoid the interruption of the SPI Master's data transmission.

4.4.3. Clipping board logic

The combinational logic of the clipping board (appendix C.2) can be divided into various sections, which are activated depending on the function that is being executed.

- The common communication part (upper left corner), consisting of the same SPI Slave core used in the computer control logic, receives data sent via the inter-board SPI bus and hands it over to the subsequent logic units through the Data_out. By default, the SPI Slave's MISO output is unlinked from the actual MISO output of the CPLD chip and the external MISO_in signal is just looped through the CPLD via the bus MUX module "lpm_mux0" (right of the Slave). If the board is addressed, lpm_mux0 maps the Slave's MISO signal to the external MISO output. This functionality is represented by the symbolic switch shown in figure 4.9. The Data_out bus is connected to the data inputs all subroutines of the processor logic.

- A bitwise comparator logic ("lpm_compare0", below SPI Slave) is used to check whether the 8 address bits of the data packet received on the SPI bus match the board address which is set with an external hardware dip switch. If both bit sequences are identical, the comparator's output goes HIGH and enables the entire subsequent board logic, including the MISO output switch of the communication part.

- For a correct sequence and save timing of the control signals, I applied a small group of delaying and single pulse generating logic one-shot elements. The leftmost one-shot module with its inverted output (small circle) delays the start signal which triggers the further data processing, to ensure that the bus comparator "lpm_compare" finished processing and that new data from the SPI link is available at the data input of the function code decoder ("lpm_decode0").

- The decoder module "lpm_decode0" (below the comparator) maps the 4 bit binary encoded function code number to a dedicated output representing its decimal value. This way each of the four following subroutines can be individually activated:

  - With function code 1 each of the 8 analog channels can be individually disabled using a pattern of 8 bits: if a bit is set to '1', the corresponding channel is disabled, if set to '0' it is enabled. An 8 bit wide flipflop module maintains the bit pattern at its output until the next disable command arrives.

  - Function code 2 activates the SPI Master that sets the attenuation values. Similar to the disabling routine, an 8 bit pattern is used to select the attenuators by their SS inputs, to allow individual or combined adjustment of the channels' attenuations.
  - The clipping voltage is set with an SPI Master addressed with function code 3, controlling the 8 channel DAC7568-A.
  - Likewise, the voltages for the delays are adjusted by the DAC7568-C, managed by another SPI Master core activated by function code 4.

- In order to not disturb the sensitive analog signal in any way, the external quartz oscillator which provides the clock required for the one-shots and the SPI Master modules is disabled when the board is not addressed or when the selected clocked process terminates. The oscillator is enabled when the board is selected. This toggling of the clock generator is performed by the final logic section of the clipping board CPLD (lower right corner).

4.4.4. Discriminator board logic

Most of the combinational circuit of the discriminator board is identical to the clipping board logic (see appendix C.3). The crucial difference is the inclusion of the rate counting unit which is the centerpiece of the newly introduced measurement process and hence will be described in more detail in the following.

The counting section consists of

- the main rate counter logic
- a 24 bit flipflop memory to hold the result of the counting process until the next readout request
- and an input switch (bus multiplexer) to select one of 5 signal sources which serve as input of the rate counter. The 5 sources are:
  - Discriminator output (trigger signal) of the discriminator board's first channel ("trigger1_in", not used during the tests)
  - Trigger signal of the second channel ("trigger2_in")
  - Discriminator output of the reference channel ("triggerRef_in")
  - Output of the D-flipflop which compares the pulse arrival times of the first channel and the reference channel ("timing1_in", not used during the tests)
  - Output of the second D-flipflop comparing the first channel and the reference channel ("timing2_in")

For the actual rate counter logic I designed a rather simple circuit using three standard binary counter elements.
The first unit "lpm_counter0" is a 12 bit counter which divides the externally provided 10 MHz clock by a factor of 4096 to ≈ 2.44 kHz. The slower clock is fast enough as reference for the counting of the calibration pulses whose frequency of

emission usually is only 25 Hz. In addition, the fast primary clock would require unnecessary large registers of the subsequent counter.
The delayed clock signal serves as clock source for the following 14 bit counter ("lpm_counter1"), that defines the maximum count time ($\frac{2^{12} \cdot 2^{14}}{10\ \mathrm{MHz}} \approx 6.71$ s).
The counting of the actual trigger pulses is performed by the 10 bit "lpm_counter2".
A start pulse asynchronously resets the counters (lpm_counter1 and lpm_counter2) which then begin to increment their registers on incoming pulses.
The counting procedure stops when either lpm_counter1 or lpm_counter2 reach their maximum register value, which causes their "cout" output and hence also the "done" output of the rate counter circuit to pull HIGH.

The sizes of the counter registers were chosen such that pulse rates between 25 Hz and 2 kHz can be measured with sufficient precision while limiting the count time to reasonable 6.7 s.

4.5. Computer software

For the more complex tasks such as executing the measurement, fitting the error function to the data points and the overall data handling and management, I developed a computer program supplementing the hardware setup. As much of the data processing algorithms as possible were put into the software. Only the basic "passive" data flow control was designed in hardware-based digital logic (see section 4.4). The application was written in the programming language C on a Windows computer, but I put much emphasis on its portability to Linux operating systems. A helper library provided by Code Mercenaries is used to communicate with the USB-to-SPI converter.
Here I will only very briefly mention the program's basic features. For details I refer the reader to the source code (including annotations) in appendix D.

The control software has the following functionality:

- It manages the data communication with the USB device (using the helper library) and organizes the conversion of data for the transfer protocol (described in section 4.4.1).
- A command line interface enables to manually set/readout values or start the automatic calibration process as well as a rate scan, which are performed in automated subroutines.
- In order to provide a convenient input (and display) of adjustable properties, the software internally handles the conversion from physical values (such as delay in ns, attenuation in % of peak amplitude) to the corresponding binary numbers required to set the DACs and attenuators.
- The fitting of the error function is done by calling the external software "Gnuplot" and passing over the measured rates. The results of the fitting are handed back to the control software.

5. Measurements, results and proof of concept

The prototype of the new Sum-trigger and the analog delay line module described in the previous chapters, were planned, developed, and built with the help of the electronics workshop at the Max-Planck-Institute for Physics in Munich. There, I also performed basic tests using pulse generators, oscilloscopes, and a network analyzer, to design and optimize the CPLD logic schemes and the computer software, including the algorithms for the calibration procedures. In total, one computer control board, one discriminator board and three clipping boards were produced, since the original intention was to make a final test with a patch of 18 pixels, in order to precisely compare the results with the current setup, which uses 18-pixel-patches. Due to a lack of time, caused by delays in the delivery of essential components and minor errors in the manufacturing process, only a small test with a group of eight pixels could be accomplished finally. Nevertheless, all important parameters and the new principle of calibration were studied successfully and a solid functionality test could be done. Since only eight channels were summed up, the trigger threshold can not be directly compared to the current setup, which evaluates the sum of 18 pixels. Though, the final rate scan measurement (section 5.4) allows a rough estimation of the trigger efficiency.

5.1. Installation on-site

The final calibration tests and measurements were carried out on-site at the MAGIC I telescope on the Canary Island La Palma. In the electronics room inside the MAGIC counting house, a free crate was used to install the prototype boards (figure 5.1), which were wired the following manner:

- During the measurements the clipping and discriminator boards were powered via the crate's ± 5 V low-noise voltage source. An external power supply provided 25 V necessary for the delay line tuning.
- For the data transfer all boards were linked to each other via Ethernet cables. The computer control board (USB powered) was connected to a Laptop computer running the control software.
- As camera signal sources, unused outputs of the trigger receiver unit were connected to the test setup, which are arranged in patches of 8 channels per plug.
- All 8 channels of one patch (number 32) were fed into one clipping board, whose 8 outputs were in turn connected to a summing board.

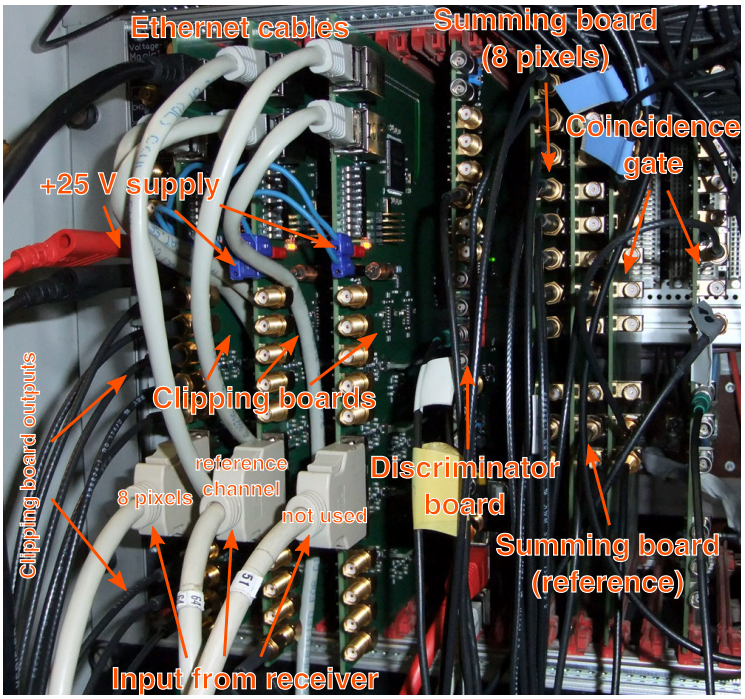


Figure 5.1.: Image of the prototype Sum-trigger installation inside the MAGIC counting house. The many black wires are coaxial cables.

- One output of the summing board was connected to the discriminator board, another output was used for monitoring with an oscilloscope.
- From another free patch, one single channel was used as a reference signal, traversing a second clipping board and dedicated summing stage before it was fed into the discriminator board's reference input.
- The discriminated sum of five arbitrary camera pixels was used as a gate signal to distinguish calibration pulses from background or shower signals (see chapter 2.4). This gate signal was fed into the gate input of the counter unit on the discriminator board.

5.2. Calibration and component tests

The most extensive part of the on-site installation procedure was the optimization of the automated amplitude and delay flatfielding processes. First of all, a stable data transfer between computer and control board had to be established by reducing the amount of read/write attempts per second on the USB interface. As mentioned in section 4.3.3 this workaround is necessary due to the flaws of the USB-to-SPI converter used on the computer control board. As a consequence, to limit the time required for the calibration, the number of measurement points per channel and calibration run had to be as little as possible, while still providing enough data for precise fitting of the error functions. In addition, adequate reference values (discriminator threshold for the amplitude flatfielding and delay of the reference

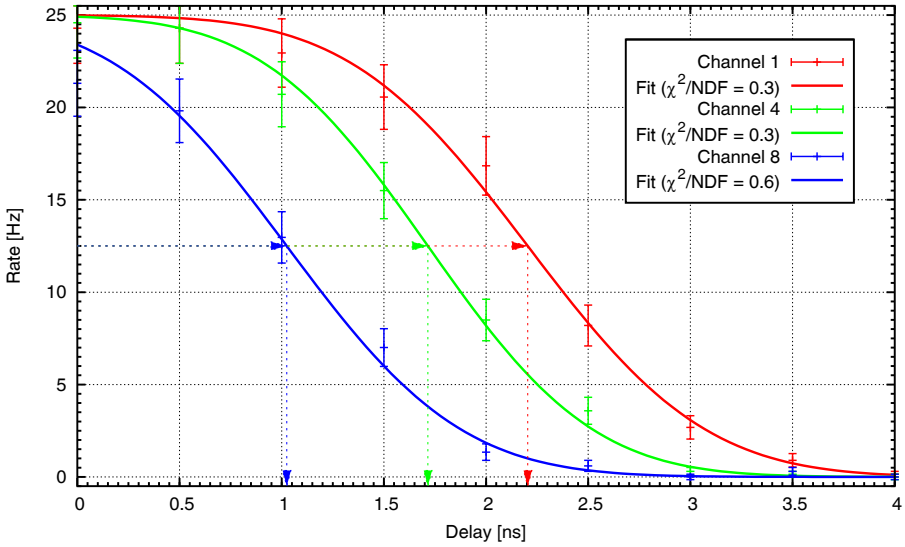


Figure 5.2.: Delay measurement for three arbitrary channels of the test setup. The error function (continuous lines) is fitted to the data points (crosses). The horizontal arrows mark the 50 % rates of the corresponding curves, which determine the optimal value (vertical arrows) for the delay adjustment.

channel for the timing adjustment) had to be found in order to cover the full dynamic range of all 8 used channels and achieve reliable fits. Finally, various checks of the pulse properties after each calibration run lead to a step by step improvement of the parameters.

5.2.1. Precision of the timing measurement

The precision and reliability of the delay adjustment procedure were very promising, already during first tests at the MPI in Munich. As can be seen in figure 5.2, the measured data closely follows the fitted error function shape, indicating that the distribution of the timing jitter is mainly Gaussian. Here, the principle of the measurement procedure theoretically described in section 2.4.2 is excellently working for real data: while tuning through different delay values, the rate of the trigger output is counted. The point of inflection of the error function fitted to the data represents the delay value in the transition region where the measured pulse and reference pulse arrive simultaneously. Comparing several repeated calibration runs on the same channel under identical conditions shows that the optimal delay value can be derived with a precision of ≤ 70 ps, even with only 10 measurements per channel (see figure 5.3). After automatic delay flatfielding the maximal spread of arrival times of calibration pulses on all 8 adjusted channels is ≤ 100 ps (see figure 5.4). This effectively achieved accuracy is much higher than required for the Sum-trigger, hence one might reduce the number of measured points to increase the calibration speed. This in turn enables the option of a fast online trigger calibration, with still sufficient precision. Though not necessary for the MAGIC telescopes, it

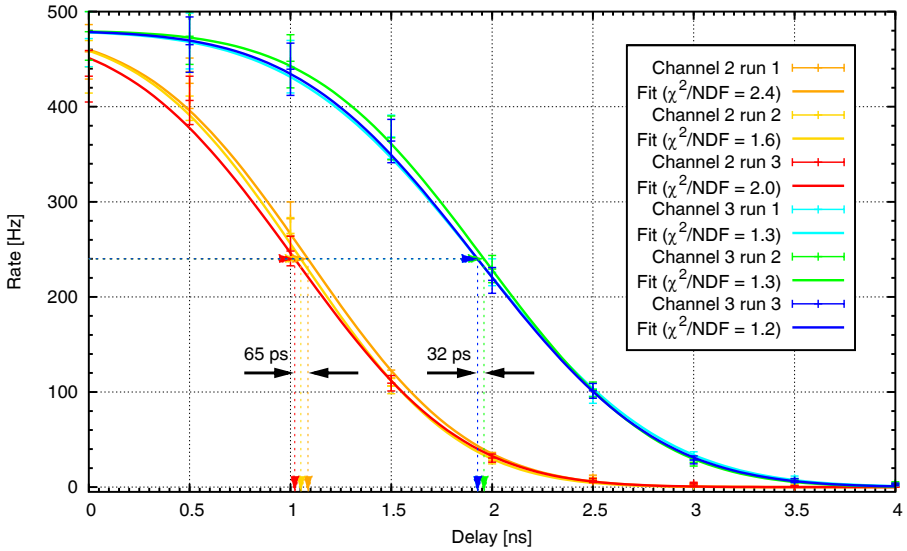


Figure 5.3.: Repeated delay measurements on two channels under identical conditions. Two extreme cases are shown here: the red/orange curves have the largest variance of all measured channels, the blue/green curves the smallest variance. For each channel three measurements are displayed: a medium, the maximum and the minimum delay value derived by the fits. For test purpose, an exceptional high calibration pulse rate of 500 Hz was used. The rate scans were performed within a time window of ≈ 2 h.

might be useful for other multi-channel systems, like CTA (see section 6).

This calibration method turned out to be very reliable: among all my test runs not a single outlier (accidentally improperly measured or falsely set delay) was found. Also, no drift of the delays over at least three days was observed, even after multiple power cycles of the prototype setup.

One evident problem is the rather high variance of pulse widths (up to ≈ 0.8 ns FWHM) among the channels. Since the presented flatfielding procedure is sensitive to the rising edges of the pulses, the peaks of differently wide pulses do not correctly overlap in time (also visible in figure 5.4). However, the differences are small enough not to impair the analog sum of all signals and hence the Sum-trigger efficiency. Moreover, the variances in the pulse widths will be smaller after the planned MAGIC I camera upgrade.

5.2.2. Results of the amplitude flatfielding process

Compared to the unproblematic delay calibration, the tuning of the amplitude flatfielding process was a difficult task. Technical problems of the MAGIC I system caused the calibration pulses to change by up to 10 % in amplitude over a time scale of around 20 minutes, in addition to the large Poissonic fluctuations of about ± 15 %. Though the measurement principle is based on variations in the signal, the gradual changes of the average amplitude can critically disturb a time-limited

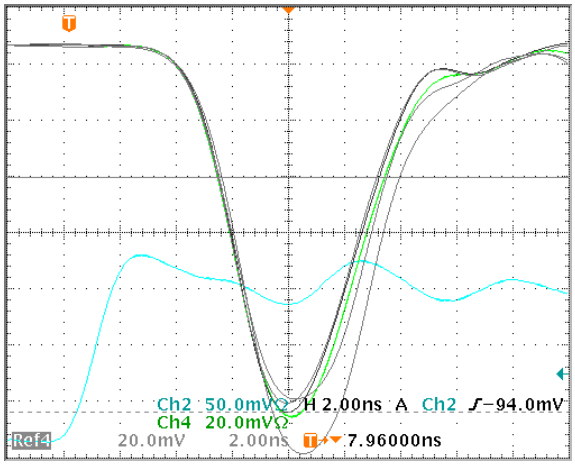


Figure 5.4.: Recording of calibration pulses on five different channels, after an automatic delay adjustment has been performed on them (with a calibration pulse rate of 25 Hz). The pulses were captured at the output of the summing stage, which inverts the signals (hence their vertically inverted depiction). The blue curve is the gate signal on which the oscilloscope was permanently triggering. Actually, the rising edges ("falling" edges on the image) of the pulses have even less time spread than the screenshot might suggest: the oscilloscope slightly shifts the temporarily saved waveforms (gray) when recalling them on the screen. Effectively, the maximal pulse delay error after adjustment is $\lesssim 100$ ps.

calibration run. Due to the reduced USB transfer speed (see chapter 4.3.3) one run takes roughly 15 minutes and thus is in the range of the large superimposed amplitude variations. This substantially reduced the efficiency of the amplitude calibration procedure. The rate measurements revealed that the frequency distribution of the pulse's amplitudes is not precisely Gaussian distributed as one can see in figure 5.5, where for some channels the fit of the error function shows large deviances, in contrast to the delay measurement in figure 5.2.

Due to peculiarities mentioned in section 4.2.4, the amplitude adjustment can only be performed by *attenuating* the signals down to a common level. Unfortunately it was necessary to attenuate the initially high pulses to a rather low reference amplitude (≈ 70 % of the peak voltage), since the absolute variance of the amplitudes is large, which results in very wide error function shapes. The reference voltage (discriminator threshold) had to be set to this low value in order to record an almost full shape of the error function for a successful fit to the data points (see figure 5.5). If a much smaller attenuation (hence larger amplitudes) was chosen, then the measured points covered only a fraction of the error function shape and fits would occasionally fail to converge. Nevertheless, an acceptable trade-off between small attenuation and reliable error function fitting had finally been found. In figure 5.5 rate scans of three representative channels show that the amplitudes among the 8 channels of the used patch vary by ≈ 15 % of the non-attenuated amplitude. Much larger variations may occur when more channels are equalized in gain, requiring even stronger average attenuation. This has to be considered for the final design

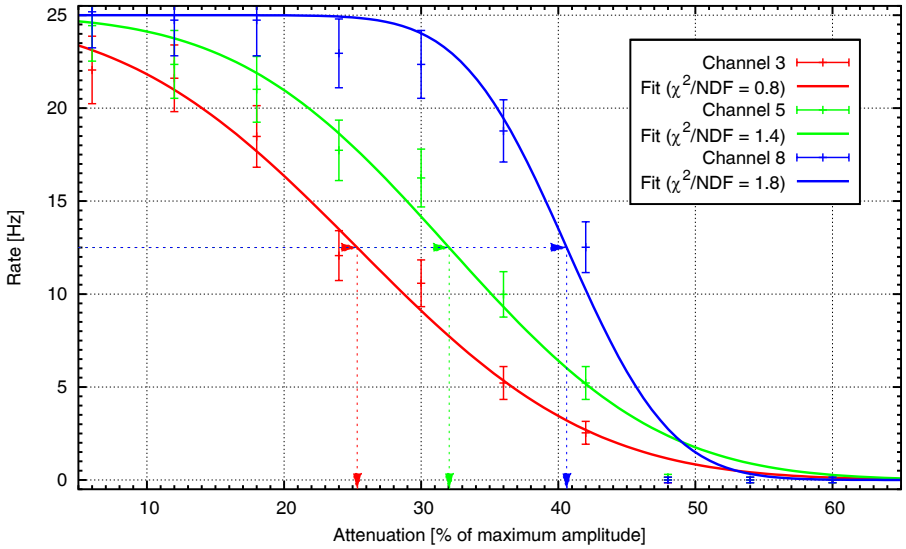


Figure 5.5.: Amplitude measurement for three arbitrary channels of the test setup. The error function (continuous lines) is fitted to the data points (crosses). The horizontal arrows mark the 50 % rates of the corresponding curves, which determine the optimal value (vertical arrows) for the attenuation.

of the new Sum-trigger system. Similarly to the delay measurement, the accuracy of amplitude flatfielding was estimated through repeated rate scans under identical conditions (figure 5.6). Though the variance of the optimal attenuation value per channel is rather small (≈ 3 %), the error in equalizing the amplitudes among all 8 channels can be as large as ≈ 10 % due to changes of the average amplitudes during an ongoing measurement. However, this precision is still sufficient and does not affect the Sum-trigger operation.

5.2.3. Performance of the clipping amplifier

The clipping amplifier LMH6553 was never examined before its application in the Sum-trigger prototype and hence was the main uncertainty concerning the analog performance of the prototype. Intrinsically being a fully differential device, it had to be wired in an uncommon way to be applied in the completely single-ended circuit of the Sum-trigger (as mentioned in chapter 4.2.5). Measurements at the output of the clipping amplifier revealed that its bandwidth is sufficiently high enough not to affect the overall Sum-trigger bandwidth: the pulses' shapes, in particular their FWHM, are not altered when passing through the amplifier. In case of high clipping voltages, the clipped signal is perfectly congruent with the unclipped pulse below the clipping level and shows a rather flat plateau (figure 5.7 (a)). When stronger clipping is applied an apparent distortion of the signal occurs (figure 5.7 (b)), which becomes critical for clipping levels below ≈ 25 % of the unclipped pulse amplitude (figure 5.7 (c)). Such a strongly distorted signal also showed very large fluctuations (≈ 50 % in peak amplitude) and thus will not properly contribute to the analog sum.

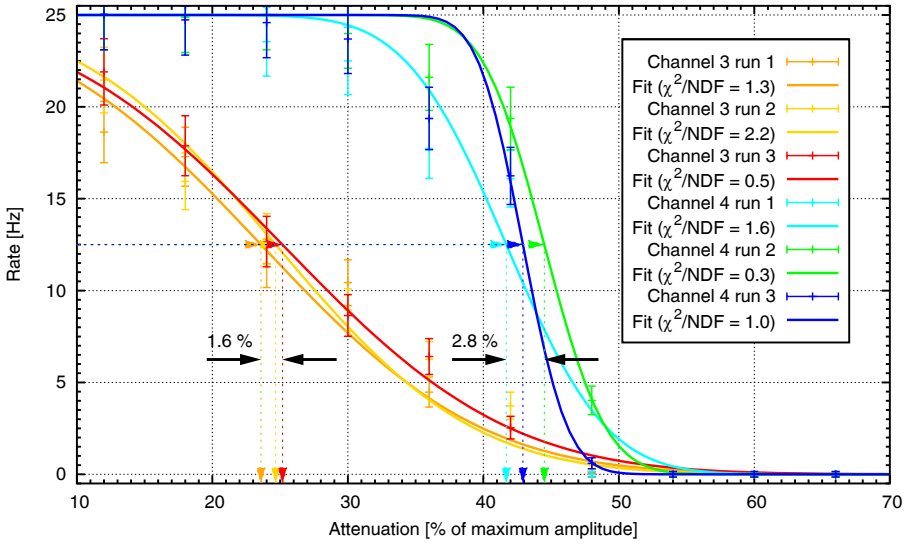
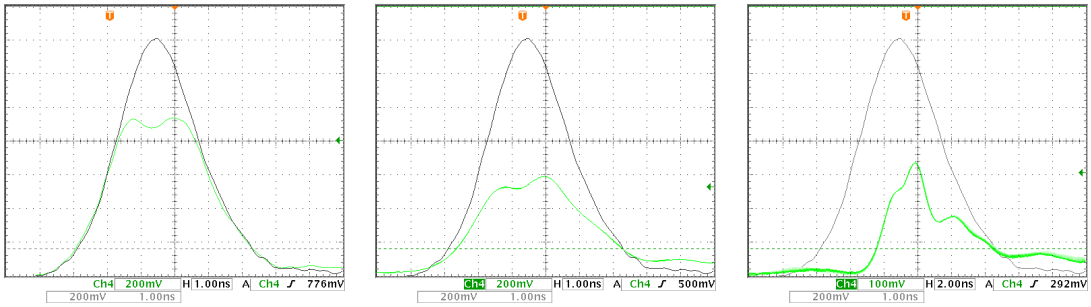


Figure 5.6.: Repeated amplitude measurements on two channels under identical conditions. The channel represented by the red/orange curves has a lower gain than the one of the blue/green scans and thus needs to be attenuated less. For each channel three measurements are displayed: a medium, the maximum and the minimum attenuation value derived by the fits. The rate scans were performed within a time window of ≈ 2 h.

In the amplifier's data sheet no lower limit of the clipping level was specified and such a distortion should not appear. A possible explanation for this behavior are parasitic currents inside the amplifier caused by the improper single-ended usage of the amplifier. Further testing of the device is required to clarify its applicability in the new Sum-trigger circuit. Due to a design error in the prototype clipping board circuit the amplification of the analog signal up to the input of the clipping amplifier is too small, evoking the need of very low clipping levels. For the clipping adjusted to 6 Phe, a very small corresponding value of 360 mV was necessary (see following section), for which the clipping amplifier's misfeature causes an unusual fast drop of the rates for large pulses (higher numbers of photoelectrons) in the clipped patch rate scan (figure 5.10): the higher the incoming signal amplitudes, the stronger are the distortions and amplitude fluctuations caused by the clipping and hence the less efficient is each channel's contribution to the analog sum, thus yielding smaller rates. A possible solution of this problem is a different order of the amplifying components along the signal path such that no clipping below 500 mV is necessary.

Deriving the conversion factor from mV to Phe

As described in section 2.4.7 the conversion from the directly measurable signal amplitudes to the actual number of photoelectrons that trigger the readout had to be derived, in order to set the clipping levels in units of Phe. Due to lack of time, the originally planned cross-calibration between clipping level and discriminator threshold by clipping level rate scans was not performed. Instead, all relevant pulse properties were manually determined with an oscilloscope at one output of the



(a) Clipped at 980 mV (b) Clipped at 620 mV (c) Clipped at 330 mV

Figure 5.7.: Recordings of calibration pulses at the output of the clipping amplifier. The gray bell-shape is the unclipped signal, whereas the green pulse is the clipped one. For small clipping (a) the clipped signal has a very flat plateau and is mostly congruent with the unclipped pulse. In case of stronger clipping (b and c) the pulse shape becomes more distorted, especially for clipping levels below ≈ 25 % of the unclipped amplitude (c). Mind the different scale (100 mV/division) of the green pulse in (c).

clipping board and the conversion factor was deduced by the following procedure:

1. Clipped voltage amplitudes for a series of DAC values were measured and a linear function was derived, to calculate the DAC value directly in terms of the required clipping voltage (figure 4.6).
2. Delays and gains of all 8 channels were equalized by the automatic calibration.
3. Averaging the FWHM of various afterpulse waveforms revealed a value of $\mathrm{FWHM}_{afterpulse} \approx 2.6$ ns (see also section 5.3).
4. Likewise, the average FWHM of calibration pulses was obtained, yielding $\mathrm{FWHM}_{calibPulse} \approx 4.9$ ns.
5. The average peak voltage of calibration pulses was found at $U_{calibPulse} \approx 1.54$ V.
6. From the telescope's automatic online calibration the average number of Phe per calibration pulse was deduced: $N_{phe_{calibPulse}} = 49$.
7. Evaluating equation 2.1 results in a conversion factor $C \approx 59$ mV/Phe at the output of the clipping board for unclipped signals.

Consequently, to cut the signals at a level of 6 Phe, the clipping voltage

$$U_{clip} = C \cdot N_{phe_{calibPulse}}$$

(see equation 2.2) had to be adjusted to ≈ 360 mV on each channel, which equivalents a decimal DAC value of 112 (70HEX), according to the formula derived in figure 4.6. This was manually set in the control software.

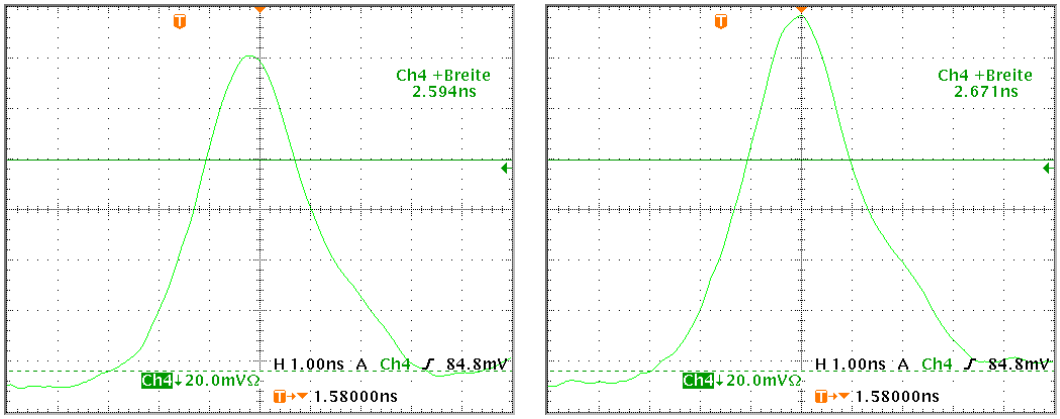


Figure 5.8.: Snapshots of afterpulses at the output of the summing stage. The actual FWHM of ≈ 2.6 ns was derived by calculating the average FWHM of 10 single afterpulse snapshots (not averaged by the oscilloscope).

5.3. Bandwidth of the analog signal path

As pointed out in chapter 2 the bandwidth of the analog path of the Sum-trigger is a very crucial parameter. Monte Carlo simulations carried out during the development of the first generation Sum-trigger system [4] showed that afterpulses should have a FWHM between 2.5 and 3.0 ns at the discriminator input, for optimal separation of γ-ray induced showers from NSB. This corresponds to a bandwidth of 230 to 360 MHz (see following calculation).

The estimate of the bandwidth was performed using PMT afterpulses as input signals, since they represent the shortest pulses coming out of the receiver board. As displayed in figure 2.2 (a), the FWHM of averaged afterpulses at the output of the receiver was measured to be 2.1 ns. Two snapshots of afterpulses having passed the prototype clipping board and a summing stage are shown in figure 5.8, with an average FWHM of ≈ 2.6 ns. Single pulse snapshots were recorded, because a waveform averaged by the oscilloscope would include various pulses from different origins (NSB, calibration, and large cosmic events), superimposing the pure afterpulse and thus showing a slightly deviating FWHM.

With the formula presented in section 3.4, the overall bandwidth of a transmission line can be derived by comparing the signal rise times of a Gaussian pulse before and after passing the line. Since the FWHM of recorded pulses can be measured with higher precision, the relation between the conversion factor from the FWHM of a Gaussian pulse[1] and its rise time derived in the following. Starting from the Gauss formula

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

and the relation

[1]Even though the pulses in figure 5.8 are slightly distorted at their trailing edge, for the measurement of the FWHM they can be assumed as Gaussian shaped.

$$\mathrm{FWHM} = 2\sqrt{2\ln 2}\sigma$$

calculating the difference of the x_2 value at 90 % of the Gauss peak $G(0)$ and x_1 at 10 % of $G(0)$ yields

$$t_{rise} = x_2 - x_1 = \mathrm{FWHM} \cdot \underbrace{\sqrt{-\frac{1}{4\ln 2}}\left(\sqrt{\ln(0.9)} - \sqrt{\ln(0.1)}\right)}_{\approx 0.72}$$

. Thus the transmission line cut-off frequency can be expressed by the following formula:

$$f_{limit}\,[\mathrm{GHz}] \approx \frac{0.49}{\mathrm{FWHM}\,[\mathrm{ns}]} \qquad \mathrm{FWHM} = \sqrt{\mathrm{FWHM}^2_{output} - \mathrm{FWHM}^2_{input}}$$

(cf. equation 3.4). Evaluating the equation with the measured values (2.1 ns input, 2.6 ns output) results in an overall bandwidth of ≈ 320 MHz for the full analog signal path of the Sum-trigger prototype. The delay line of the channel that was used to measure the pulse widths, was tuned to a delay of 2.32 ns, which is a medium value (≈ 3 ns). Hence, for channels that require smaller delays even higher cutoff frequencies are possible (see section 3.4).
Among the 8 channels the prototype setup was comprised of, the settings of the delay lines ranged from 1.17 ns to 2.49 ns, whereas 6 channels were adjusted to less than 2.27 ns and thus can achieve higher bandwidths than 320 MHz. This result fulfills the required specifications for the new Sum-trigger system.

5.4. Final trigger patch rate scan

After successful equalization of gains and timings of all 8 channels and adjusting the clipping level per channel to 6 Phe, several rate scans were performed during regular observation with MAGIC. First, rates of each single pixel were measured without clipping, while the other 7 channels were turned off. Then, rates with the sum of all 8 clipped channels were recorded.

The unclipped single pixel rate scans (figure 5.9) serve as a cross check if the conversion factor from pulse amplitudes to the number of photoelectrons was derived correctly. Within errors, the prototype rate scans of figure 5.9 are in agreement with the measurements from the current Sum-trigger and Monte Carlo studies shown in figure 2.2 (b). Especially the position of the kink, origination from the transition from NSB to afterpulses, coincides in both measurements at a discriminator threshold of 6 to 6.5 Phe. That implies that the correct conversion factor was found.

As described in section 1.1 the cumulative cosmic ray energy spectrum follows a power law with $\Gamma = -1.75$ for the MAGIC telescope's energy range. Thus, the particle rate R is a function of the energy threshold E_{th}:

$$R \propto E_{th}^{-1.75}$$

For two differential energy thresholds the ratio of rates is

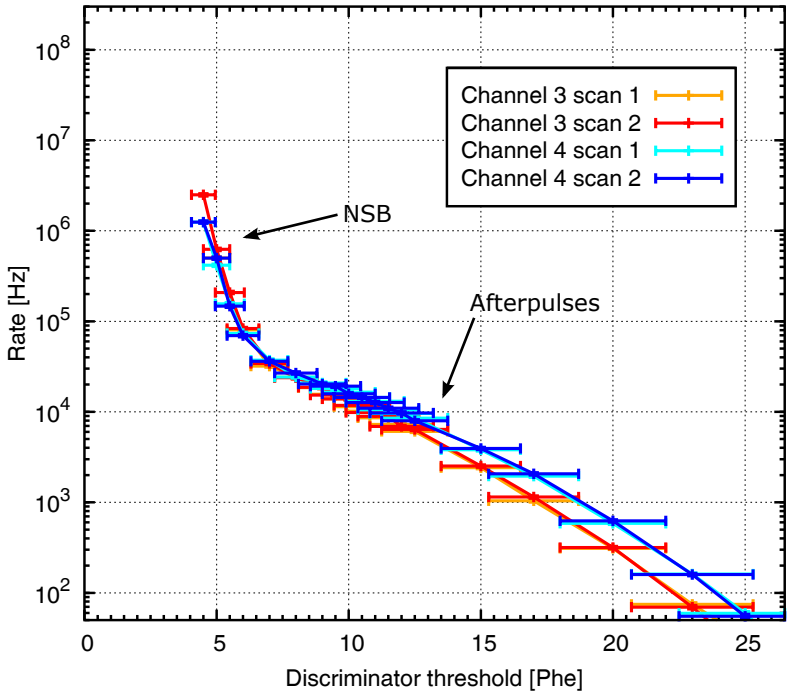


Figure 5.9.: Single pixel rates of two channels without clipping, recorded with the prototype Sum-trigger.

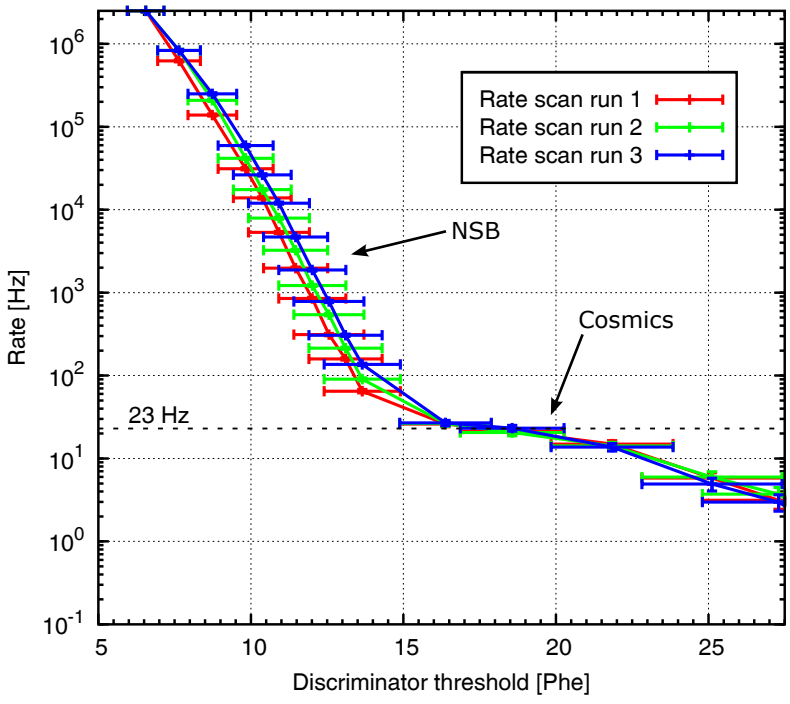


Figure 5.10.: Rate scan with the patch of 8 pixels (clipping set to 6 Phe). This is the final measurement performed with the calibrated prototype. At the threshold a rate of 23 Hz is achieved.

$$\frac{R_1}{R_2} \propto \left(\frac{E_2}{E_1}\right)^{-1.75}$$

. This means that a lower threshold results in a higher event rate, giving a practical criteria to optimize the trigger settings. It was used for the optimization of the current Sum-trigger system [4]. Applied to a patch of 18 pixels and with optimized settings, the current Sum-trigger provides a rate of ≈ 50 Hz for cosmic events.

Since the prototype was using a patch of only 8 pixels, its properties can not be directly compared to the current, thoroughly analyzed Sum-trigger. Just a very rough estimation of the trigger performance is possible, if we assume that the cosmic event rate scales linear with the area covered by a patch (strictly speaking, this is not correct!). Accordingly, the prototype should yield a rate of ≈ 23 Hz. This matches the measurements with the sum of 8 channels, as depicted in figure 5.10. Still, detailed Monte Carlo simulations are required to classify this result.

6. Conclusion & Outlook

The general concept of the Sum-trigger was successfully proved with a prototype installation in early 2008 (M. Rissi et al. [1]), which enabled the first ground-based detection of the pulsed γ-radiation originating from the Crab pulsar above 25 GeV with a high significance [15]. This observation allowed to reject a fundamental model predicting the high energy γ-ray production close to the central neutron star, the so-called *polar cap model*. In addition, the results gave evidence that the alternative leading model, the *outer gap model*, is too simple to describe the characteristics of the detected γ-radiation [4].
Unquestionable, the prototype Sum-trigger significantly improved the already outstanding sensitivity of the MAGIC I telescope in the lower energy ranges of 20 to 100 GeV. While the first prototype demonstrated a significant advancement, it also showed that further improvements and upgrades are required to make its operation more stable and easier to handle.

In my thesis I developed, built and tested a new prototype based on the successful first-generation Sum-trigger setup. The goal was to design a new computer-based automatic calibration of the trigger and its parameters, in order to eradicate the time consuming manual tuning required to maintain the initial Sum-trigger. Included was the development of a novel analog delay line to adjust the signal transit times of the camera sensors, and the improvement of the bandwidth of the circuit's analog part. For the suppression of false triggers from large afterpulses generated in photomultipliers, a new clipping amplifier was implemented.

In tests, performed in the electronics laboratory at the MPI in Munich and on-site at the MAGIC I telescope on La Palma, I found that the prototype's circuit and calibration procedure entirely fulfilled or even surpassed the required specifications. The minimal bandwidth of the analog signal path exceeds 300 MHz, which only slightly widens the PMT pulses, just by the right amount for optimal trigger performance. Adjusting the arrival times of signals from all trigger channels can be performed with a precision better than 100 ps over a full delay range of 6 ns. For the amplitude flatfielding an accuracy of at least ± 5 % could be achieved, even with the very unstable calibration pulses of the MAGIC I system. I could also prove that the novel concept of calibration, exploiting the intrinsic signal jitter in timing and amplitude, works remarkably precise and reliable, especially for the equalization of signal delays which is a crucial prerequisite for the efficiency of the Sum-trigger.
The new clipping amplifier principally showed a good performance but generated some distortions when large signals were clipped at low levels. A reason might be the unconventional use of the amplifier in asymmetric mode. Hence, this previously untested component has to be analyzed in more detail.

Besides the usage in the Sum-trigger circuit, the excellently performing new analog delay line can be applied in many other fields where a precise temporal adjustment of analog pulses in the order of several nanoseconds is required.

Currently, the final version of the new Sum-trigger is in development, based on the design and results of the prototype created in this thesis. The final setup will essentially feature the following enhancements:

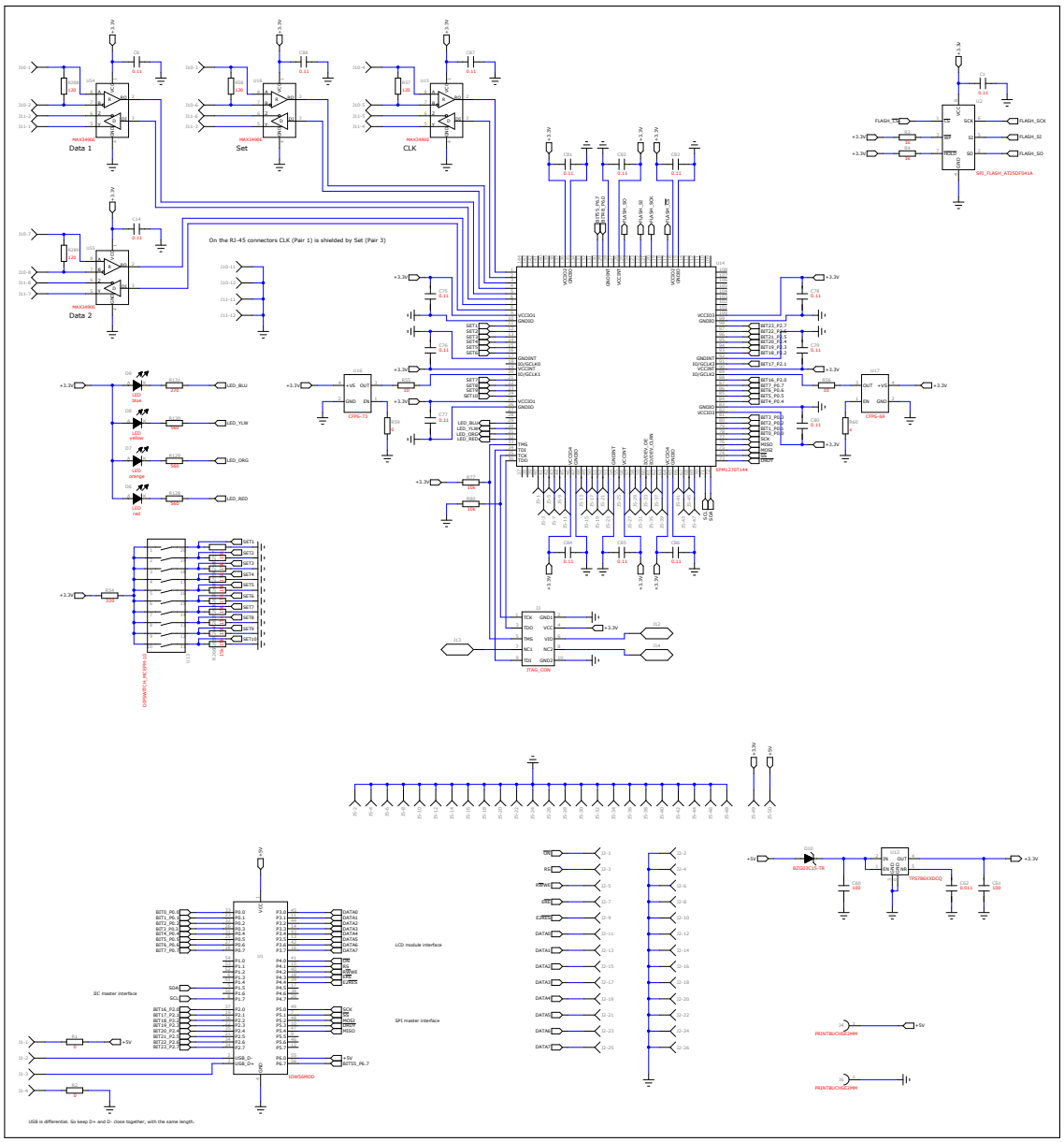
- A much larger trigger region of 512 pixels, covering the same area as the MAGIC II standard trigger (figure 2.5 (b))
- New circular-shaped overlapping trigger patches with a size of 19 pixels each, whose design was optimized in Monte Carlo studies
- One completely passive "backplane" circuit board, routing all signals from the clipping boards to the summing stages according to the arrangement of the trigger patches
- A twin of the final Sum-trigger will be installed in MAGIC II, making both telescopes capable of covering the energy range above 25 GeV

The new Sum-trigger will preferentially be used for a high statistics study of the Crab pulsar and very likely to look for the Geminga pulsar in a next step. Other important targets are the search for GRBs and high redshift AGNs. Due to absorption of ultra high energy γ-rays by the cosmic extragalactic background light (EBL), high redshift AGNs can only be studied from ground using Cherenkov telescopes with a threshold below 50 GeV. If at least part of the spectrum should be recorded, then a threshold close to 20 GeV is necessary. Another important reason to push the threshold down to well below the EBL absorption energy is the observation of GRBs, which are frequently detected at very high redshift (the most distant GRB has been observed at a redshift of 6). Obviously, the new Sum-trigger will allow to extend the spectral measurements of most of the standard γ-ray emitting sources to significantly below 100 GeV and thus provide a good overlap with measurements of the FERMI satellite. In the long run the Sum-trigger can be considered as a possible trigger system for the 23 m large size telescopes (LST) of the future Cherenkov telescope array (CTA) project. With a Sum-trigger installation it will enable to lower the energy threshold even further down to 10 to 15 GeV.

A. Schematics of electronic circuits

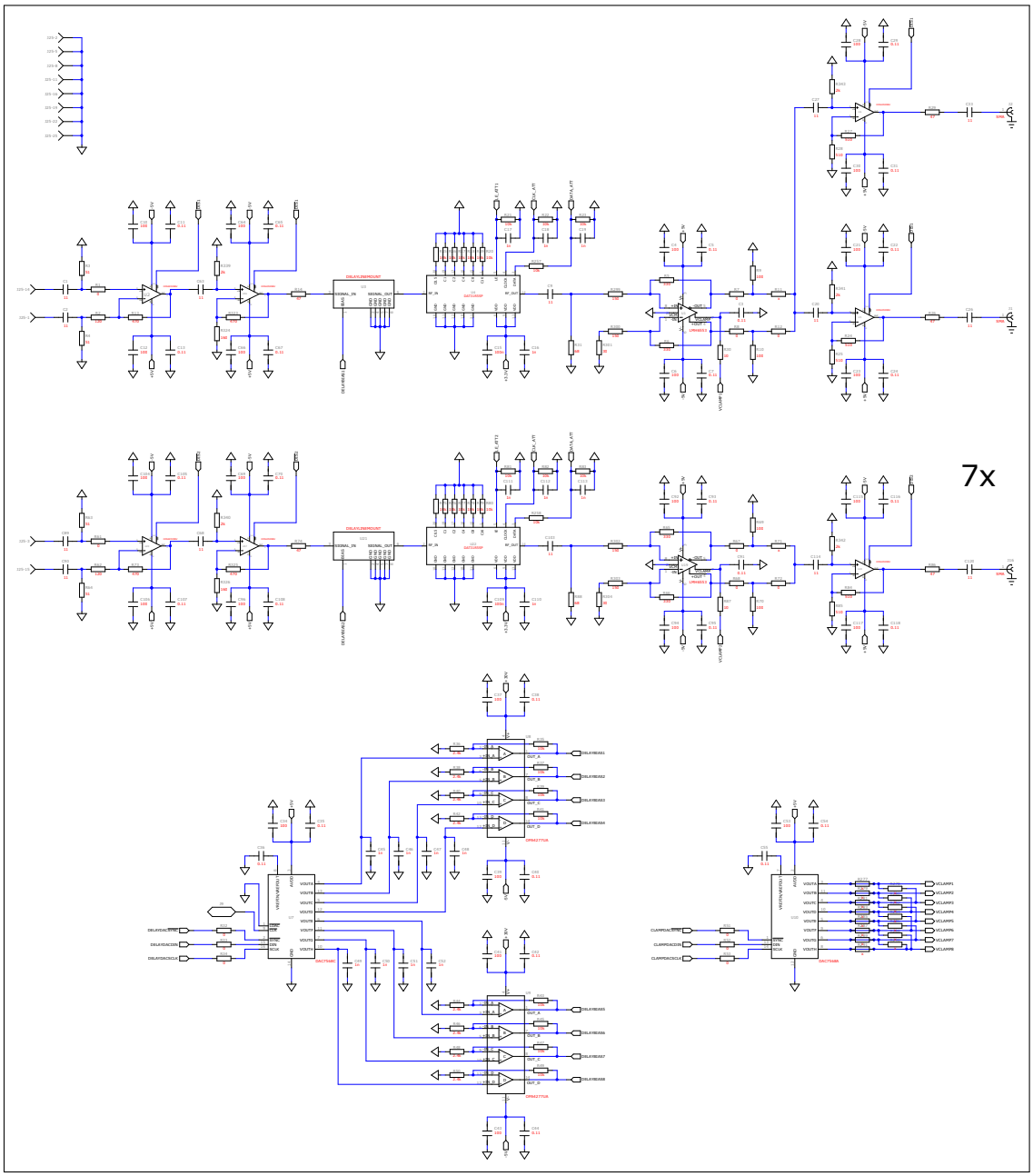
A.1. Computer control board

Schematic of the computer control board. The big square in the middle is the CPLD. Bottom left is the USB-to-SPI converter and top left the four RS-485 transceivers.

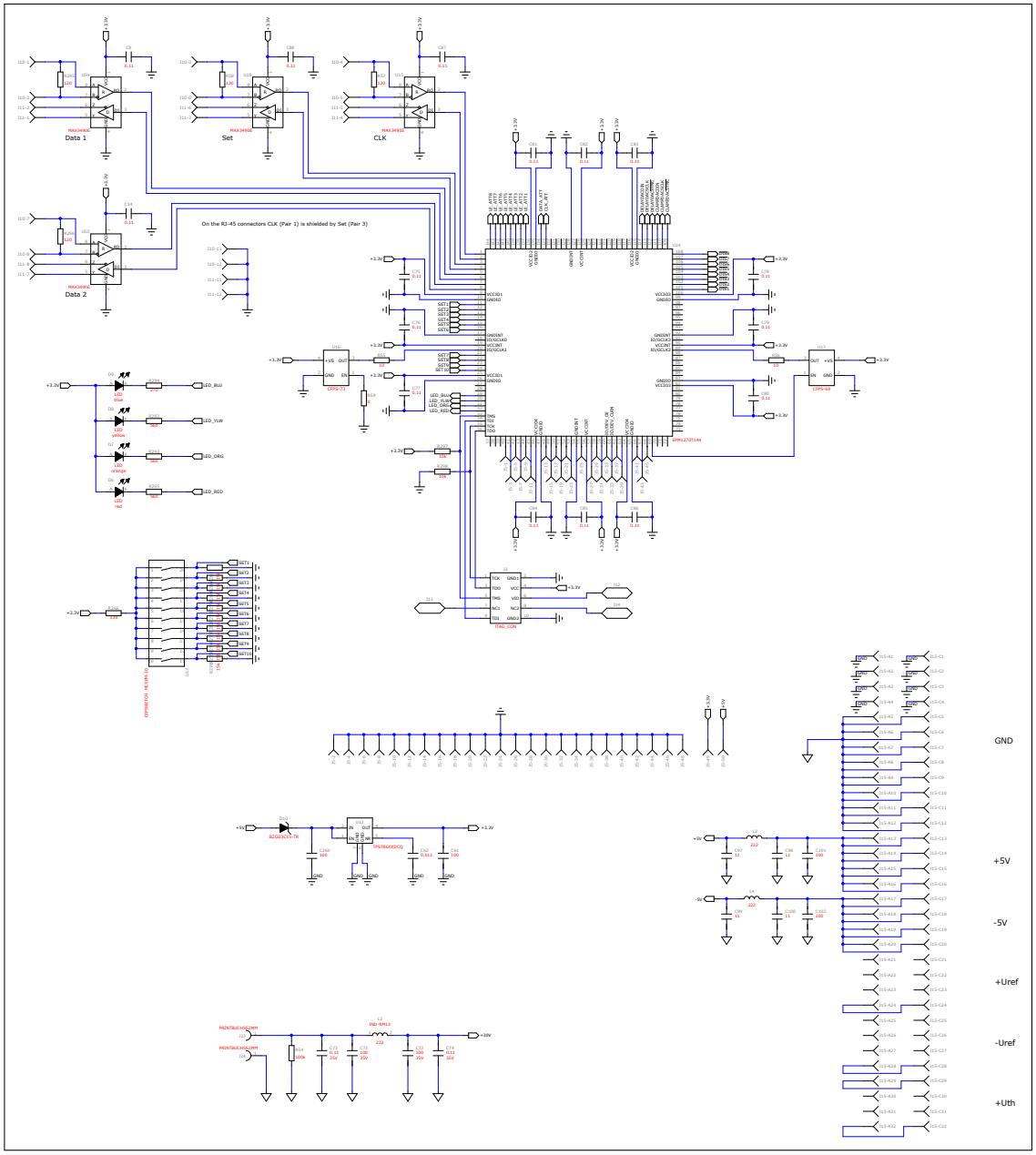


A.2. Clipping board

Schematic of the clipping board. The upper part shows the analog circuit of the eight channels. Only the first channel is split just before the output amplifier. From the other seven channels only one is shown explicitly. Below is the DAC and the two operational amplifiers to adjust the delay lines (left), and another DAC providing the clipping voltage (right).

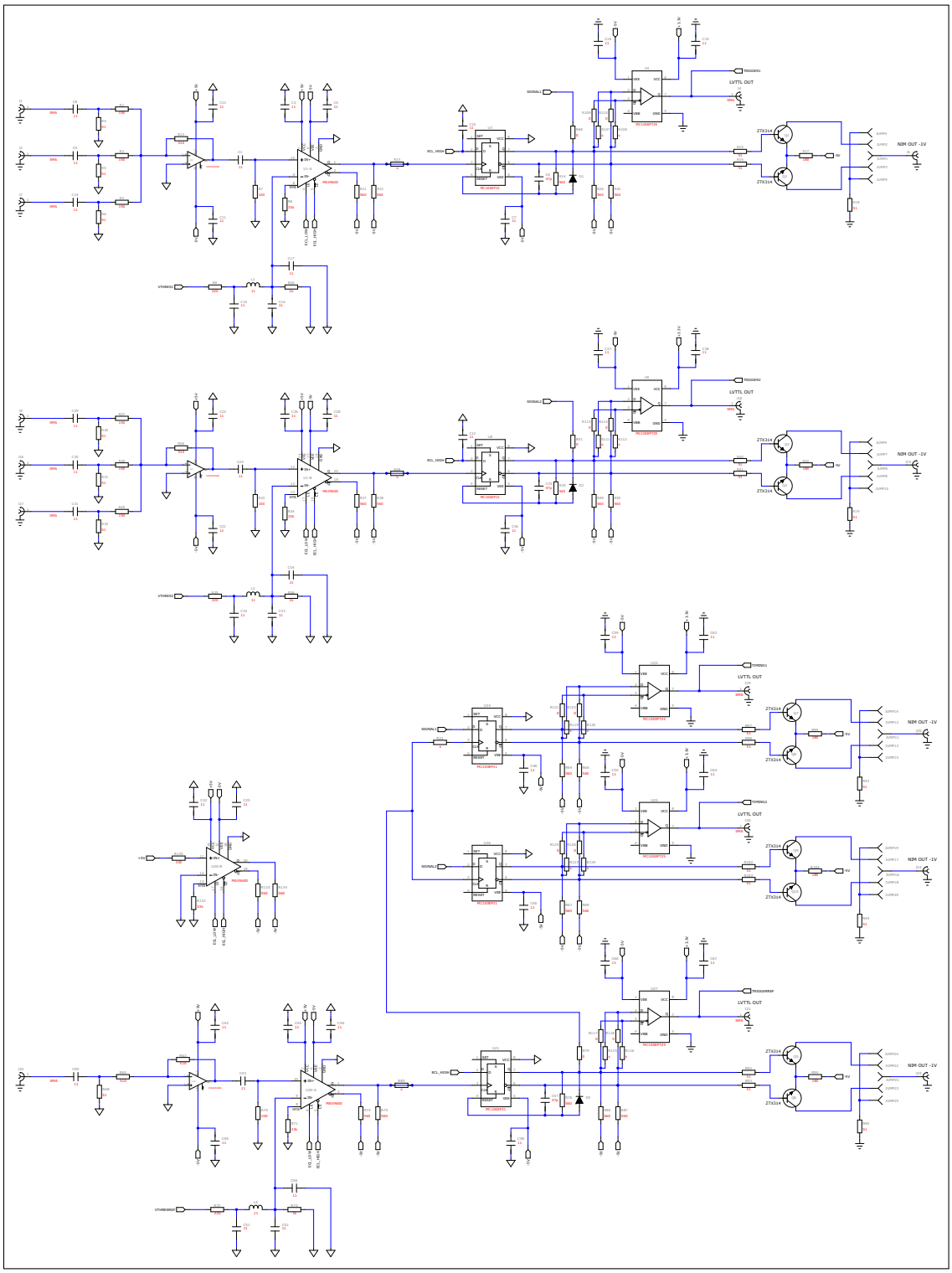


Here, the digital part of the clipping board is displayed, including the CPLD (big square) and the four RS-485 transceivers (top left). Below are the schematics of the voltage supplies and connectors.

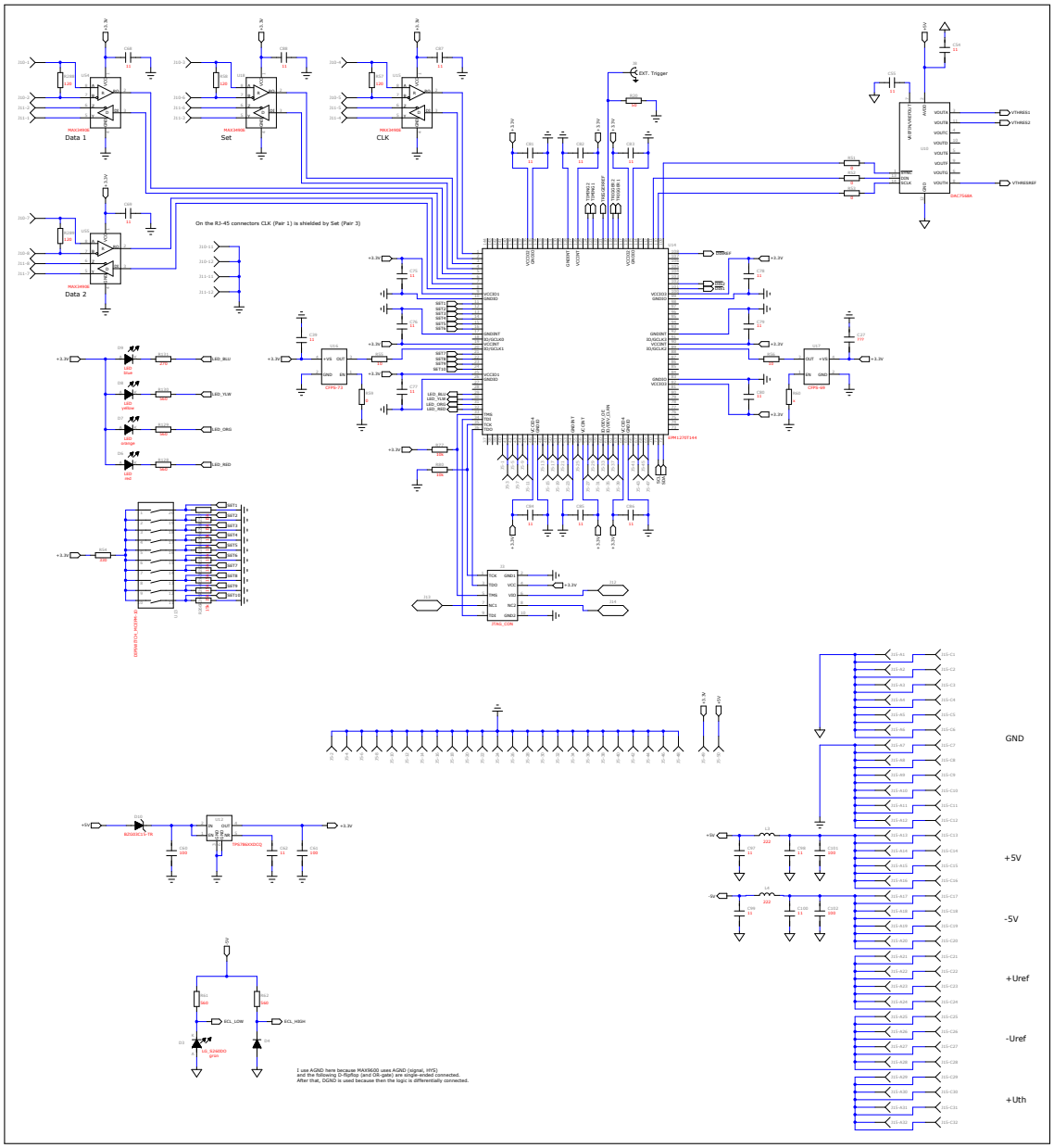


A.3. Discriminator board

Schematic of the discriminator board. The upper two sections show the circuits of the clipping channel one and two (for the prototype only the second one was used) with three summed analog inputs each (left). The larger block below represents the circuit of the reference channel and the timing unit (above).

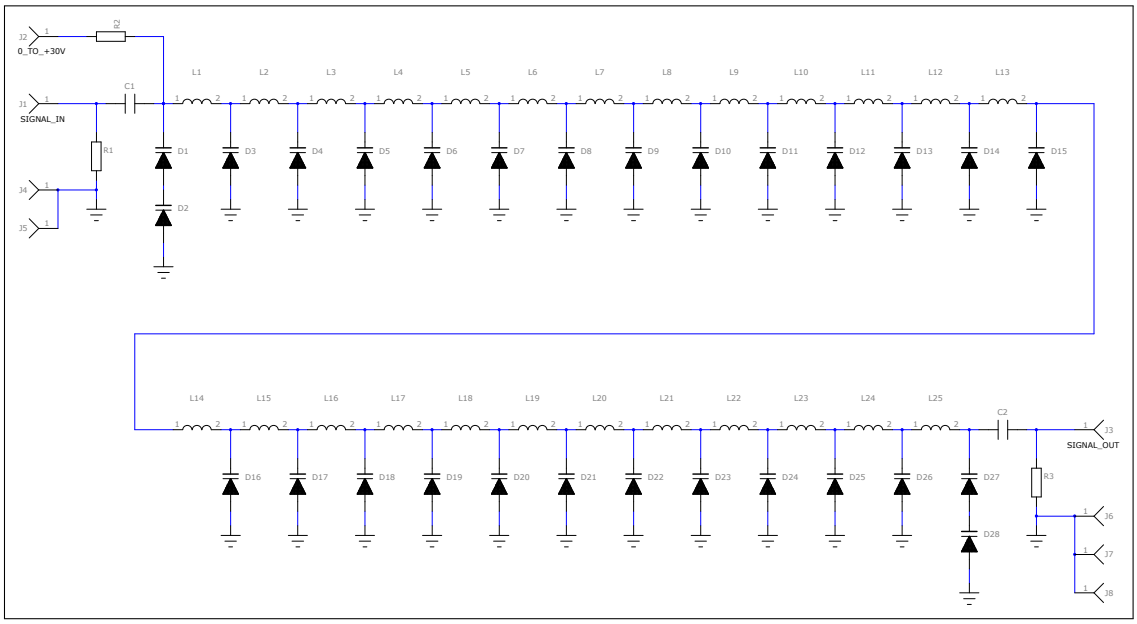


This is the digital section of the discriminator board, very similar to the one of the clipping board. In the top right corner one finds the DAC to set the discriminator thresholds.



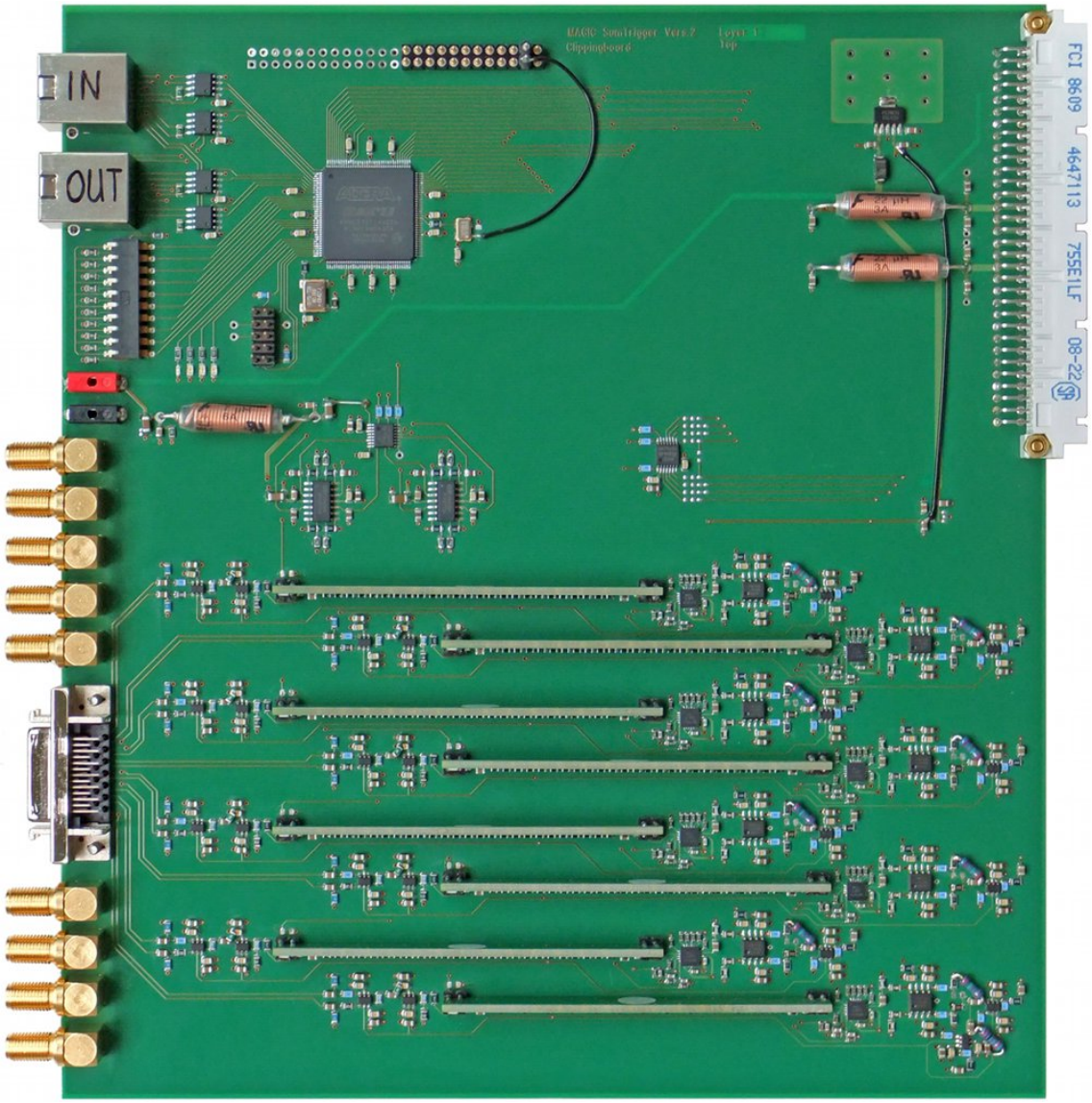
A.4. Delay line

This is the very simple schematic of the delay line prototype module, showing the varactor diodes and inductors.

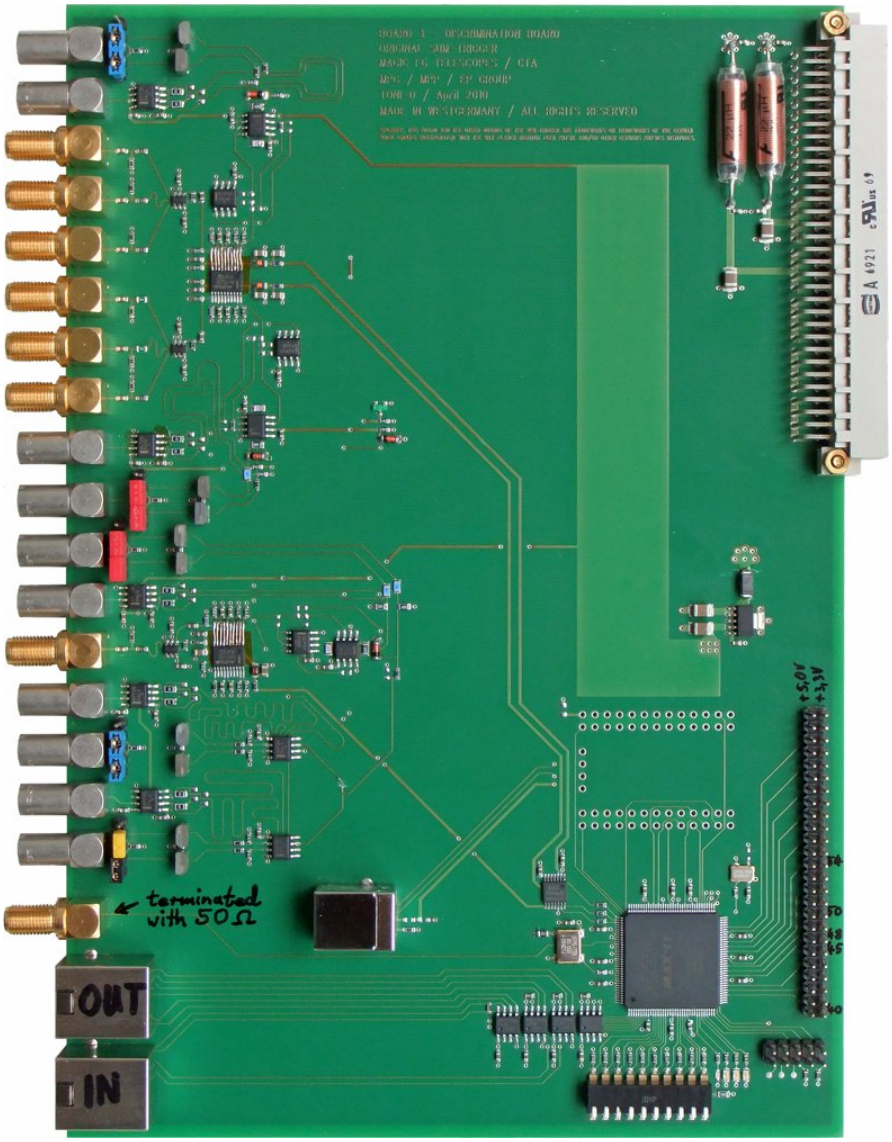


B. Final prototype boards

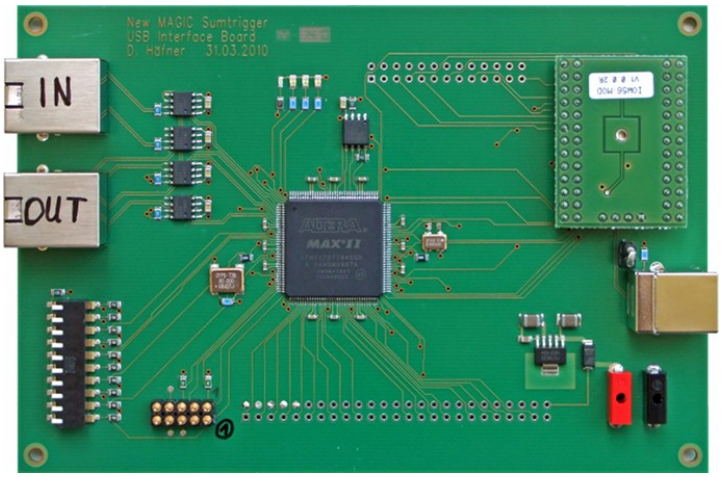
B.1. Clipping board



B.2. Discriminator board

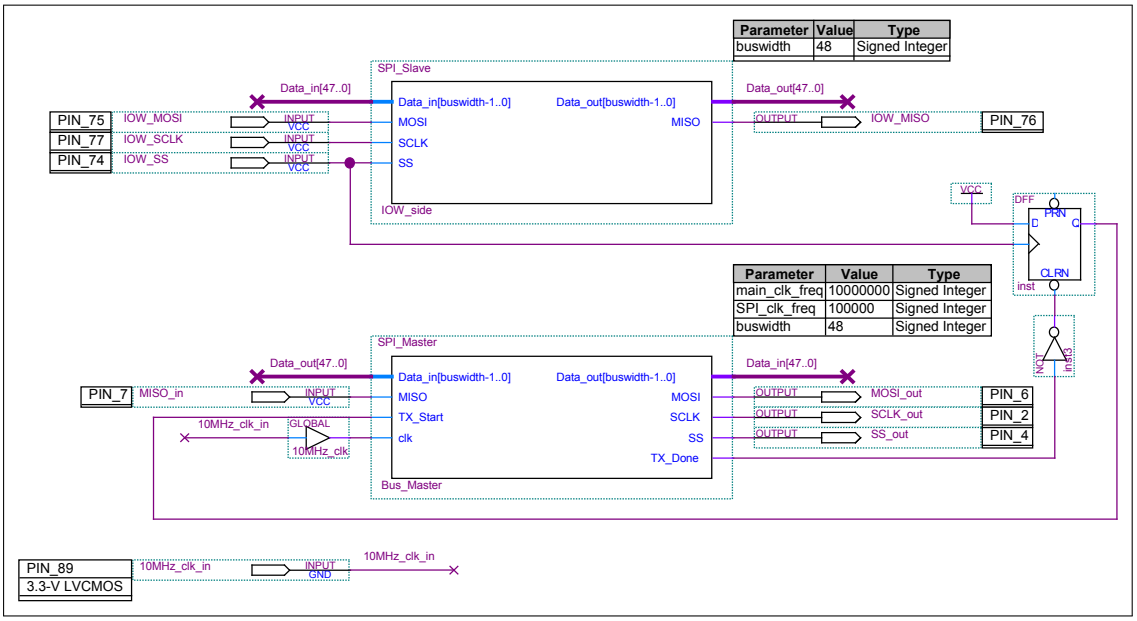


B.3. Computer control board

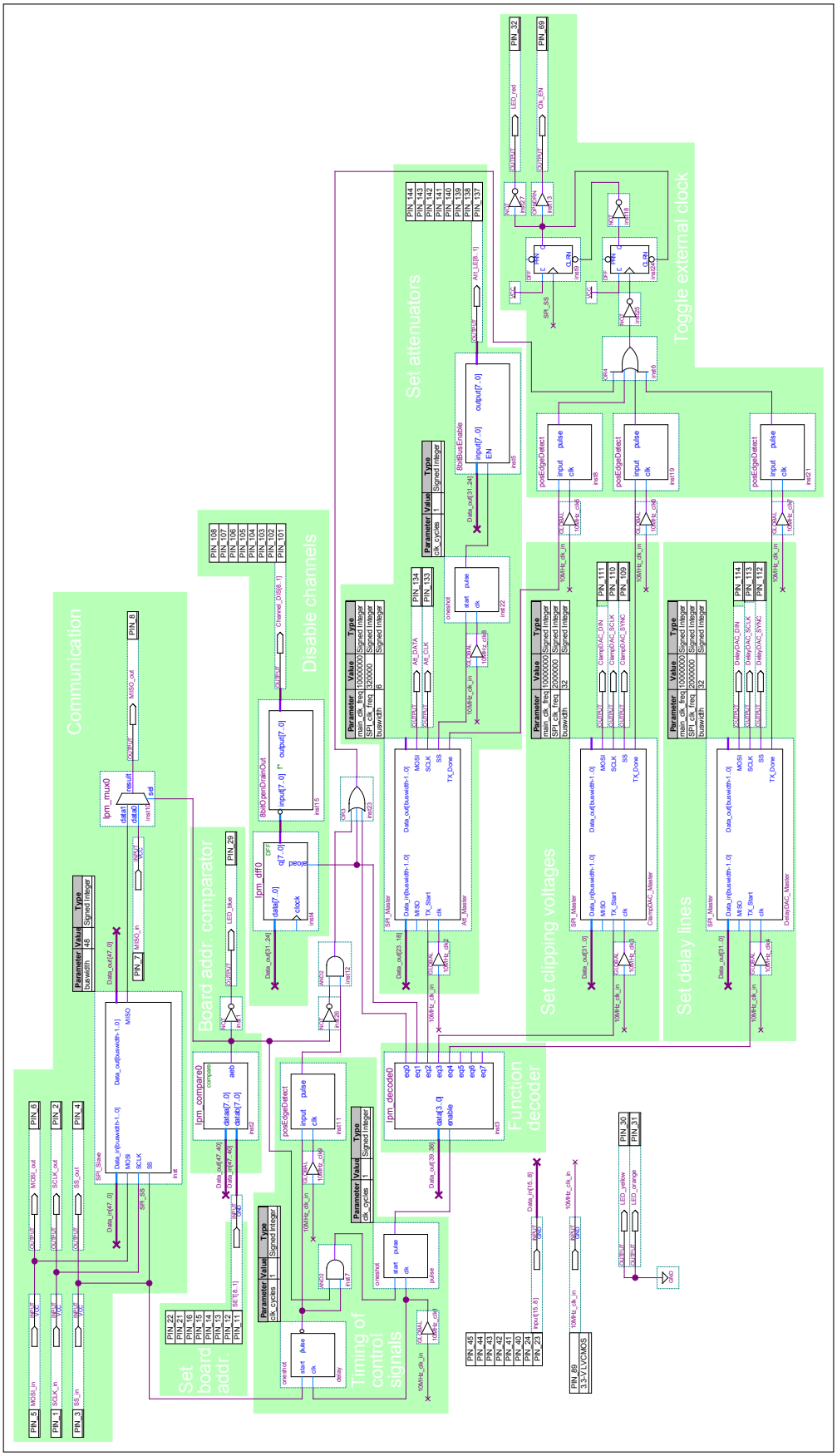


C. CPLD program schematics

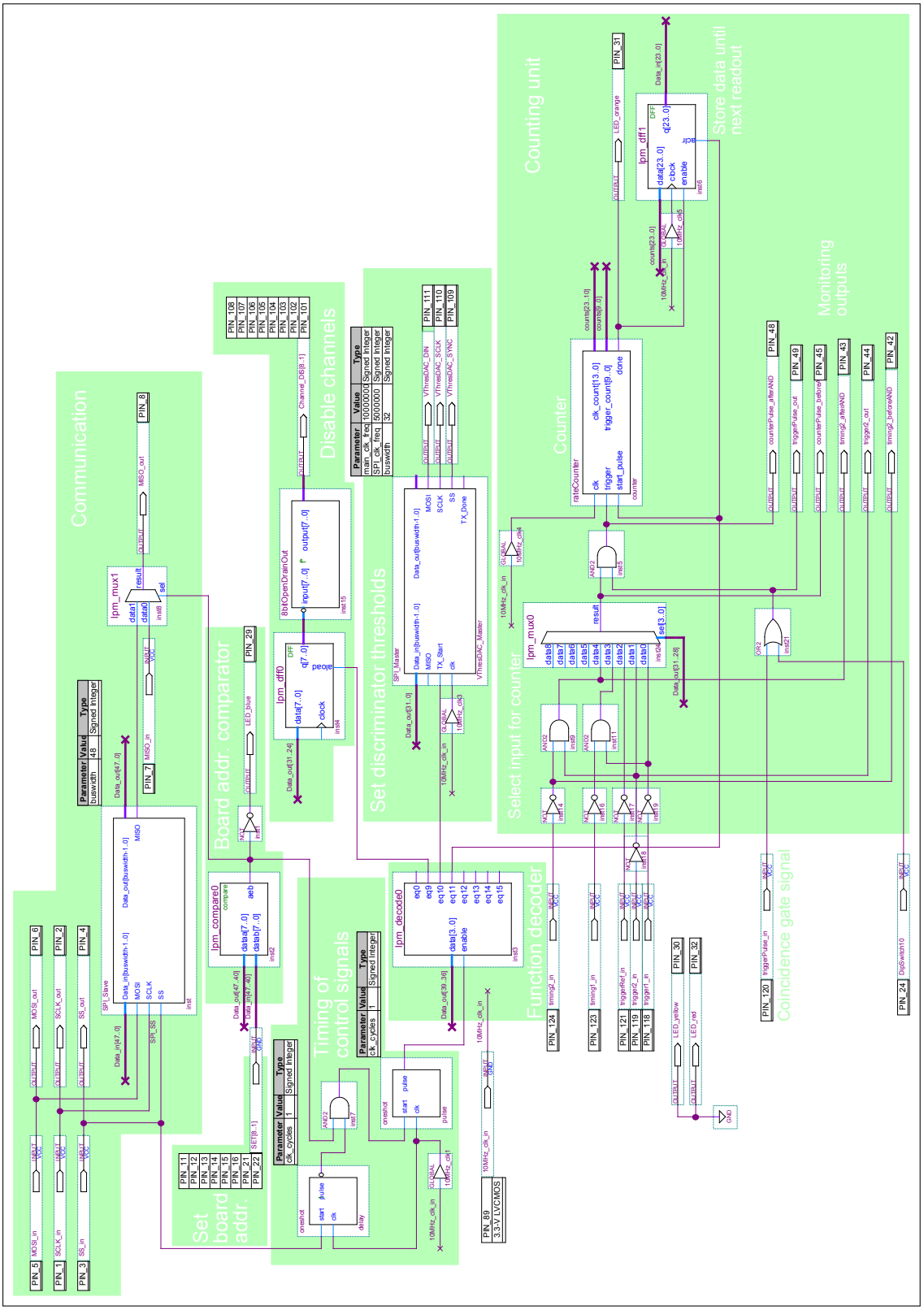
C.1. Computer control board logic



C.2. Clipping board logic

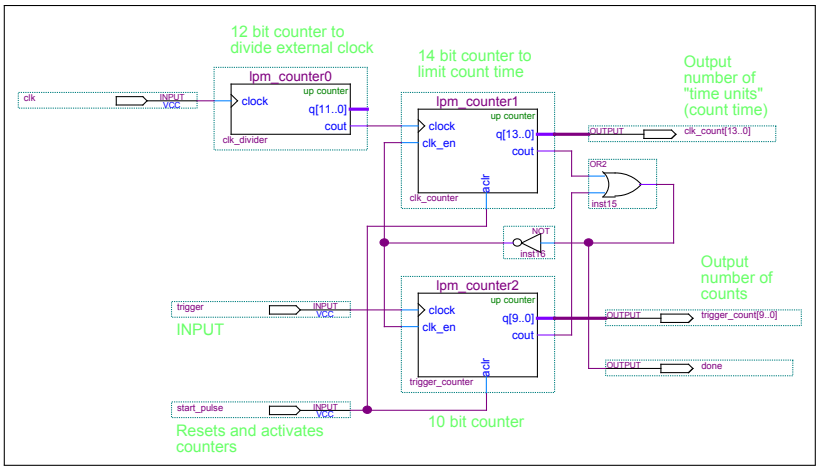


C.3. Discriminator board logic

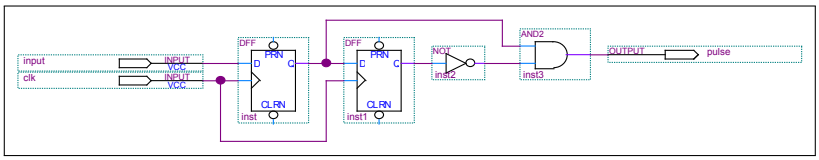


C.4. Schematics of encapsulated logic

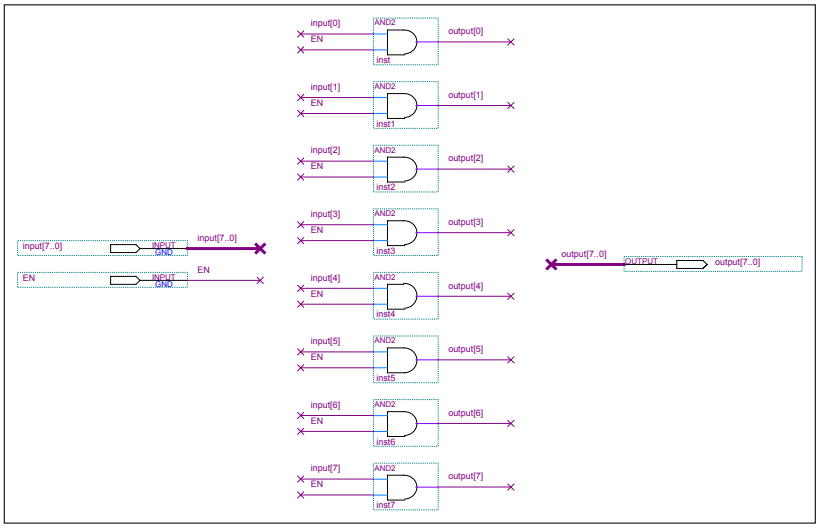
C.4.1. Rate counting logic



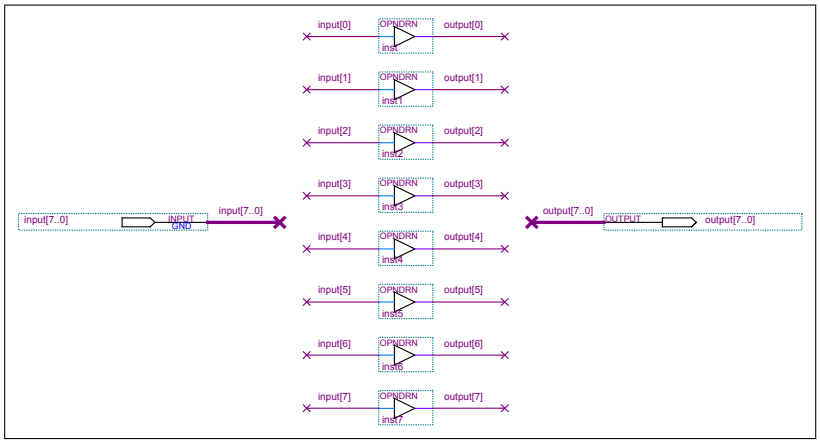
C.4.2. Positive edge detector



C.4.3. 8bit wide bus switch



C.4.4. 8bit wide open drain output



C.5. Source code of VHDL based elements

C.5.1. SPI master

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.NUMERIC_STD.ALL;
4
5  entity SPI_Master is                                          -- SPI-Mode 0: CPOL=0, CPHA=0
6      Generic ( main_clk_freq : integer   := 80000000;          -- in Hz
7                SPI_clk_freq  : integer   := 10000000;          -- in Hz / to calculate the reload value for the clock
                     divider
8                buswidth      : integer   := 32                 -- Number of Bits to transmit
9               );
10     Port ( Data_in  : in  STD_LOGIC_VECTOR (buswidth-1 downto 0); -- Data to be sent to the SPI slave
11            Data_out : out STD_LOGIC_VECTOR (buswidth-1 downto 0); -- Data received from the SPI slave
12            MOSI     : out STD_LOGIC;
13            MISO     : in  STD_LOGIC;
14            SCLK     : out STD_LOGIC;
15            SS       : out STD_LOGIC;
16            TX_Start : in  STD_LOGIC;
17            TX_Done  : out STD_LOGIC;
18            clk      : in  STD_LOGIC
19          );
20 end SPI_Master;
21
22 architecture Behavioral of SPI_Master is
23   signal   delay       : integer range 0 to (main_clk_freq/(2*SPI_clk_freq));
24   constant clock_delay : integer := (main_clk_freq/(2*SPI_clk_freq))-1;
25
26   type   spitx_states is (spi_stx,spi_txactive,spi_etx);
27   signal spitxstate    : spitx_states := spi_stx;
28
29   signal spiclk    : std_logic;
30   signal spiclklast: std_logic;
31
32   signal bitcounter    : integer range 0 to buswidth;          -- if bitcounter = buswidth --> all Bits sent
33   signal tx_reg        : std_logic_vector(buswidth-1 downto 0) := (others=>'0');
34   signal rx_reg        : std_logic_vector(buswidth-1 downto 0) := (others=>'0');
```

```
35
36  begin
37    ------ Management --------
38    process begin
39       wait until rising_edge(CLK);
40       if(delay>0) then delay <= delay-1;
41       else             delay <= clock_delay;
42       end if;
43       spiclklast <= spiclk;
44       case spitxstate is
45         when spi_stx =>
46               SS          <= '1';                          -- slave select disabled
47               TX_Done     <= '0';                          -- Transmission Done disabled ("active HIGH")
48               bitcounter  <= buswidth;
49               spiclk      <= '0';                          -- SPI-Mode 0
50               if(TX_Start = '1') then                      -- Transmission Start is "active HIGH"
51                  spitxstate <= spi_txactive;
52                  SS         <= '0';
53                  delay      <= clock_delay;
54               end if;
55
56         when spi_txactive =>                               -- transfer data from tx_reg
57               if (delay=0) then                            -- shift
58                  spiclk <= not spiclk;
59                  if (bitcounter=0) then                    -- all bits transfered -> deselect
60                     spiclk     <= '0';                     -- SPI-Mode 0
61                     spitxstate <= spi_etx;
62                  end if;
63                  if(spiclk='1') then                       -- SPI-Mode 0
64                     bitcounter <= bitcounter-1;
65                  end if;
66               end if;
67
68         when spi_etx =>
69               SS      <= '1';                              -- disable Slave Select
70               TX_Done <= '1';                              -- enable Transmission Done ("active HIGH")
71               if(TX_Start = '0') then                      -- Handshake: wait until Start-Flag is deleted ("
                       active HIGH")
72                 spitxstate <= spi_stx;
73               end if;
74       end case;
75    end process;
76
77    ---- Receiving shift register -----
78    process begin
79       wait until rising_edge(CLK);
80       if (spiclk='1' and  spiclklast='0') then             -- SPI-Mode 0
81          rx_reg <= rx_reg(rx_reg'left-1 downto 0) & MISO;
82       end if;
83    end process;
84
85    ---- Sending shift register -------
86    process begin
87       wait until rising_edge(CLK);
88       if (spitxstate=spi_stx) then                         -- reset if SS is not active
89          tx_reg <= Data_in;
90       end if;
91       if (spiclk='0' and  spiclklast='1') then             -- SPI-Mode 0
92          tx_reg <= tx_reg(tx_reg'left-1 downto 0) & tx_reg(0);
93       end if;
94    end process;
95
96    SCLK     <= spiclk;
97    MOSI     <= tx_reg(tx_reg'left);
98    Data_out <= rx_reg;
99
100 end Behavioral;
```

C.5.2. SPI slave

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.NUMERIC_STD.ALL;
4
5  entity SPI_Slave is
6      Generic ( buswidth: integer := 32);
7      Port ( Data_in  : in  STD_LOGIC_VECTOR (buswidth-1 downto 0);
8             Data_out : out STD_LOGIC_VECTOR (buswidth-1 downto 0);
9             MOSI     : in  STD_LOGIC;
10            MISO     : out STD_LOGIC;
11            SCLK     : in  STD_LOGIC;
12            SS       : in  STD_LOGIC
13          );
14 end SPI_Slave;
15
16 architecture Behavioral of SPI_Slave is
17 signal dinsr  : STD_LOGIC_VECTOR (buswidth-1 downto 0);
18 signal doutsr : STD_LOGIC_VECTOR (buswidth-1 downto 0);
19 begin
20   -- Parallel inputs --> MISO
21   process (SS, Data_in, SCLK)
22   begin
23      if (SS='1') then
24         dinsr <= Data_in;
25      elsif falling_edge(SCLK) then                          -- after falling edge of SCLK
26         dinsr <= dinsr(dinsr'left-1 downto 0) & '0';        -- MISO is being refreshed
27      end if;
28   end process;
29   MISO <= dinsr(dinsr'left) when SS='0' else '1';           -- Modified: originally "else 'Z'"
30
31   -- MOSI --> Parallel outputs
32   process (SCLK)
33   begin
34      if rising_edge(SCLK) then                              -- with rising edge of SCLK
35         doutsr <= doutsr(doutsr'left-1 downto 0) & MOSI;    -- MOSI is being read
36      end if;
37   end process;
38
39   process (SS)
40   begin
41      if rising_edge(SS) then                                -- Device is being deselected
42         Data_out <= doutsr;                                 -- Output signals are given to the pins
43      end if;
44   end process;
45
46 end Behavioral;
```

C.5.3. One-shot module

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity oneshot is
5      Generic ( clk_cycles : integer range 1 to 3 := 2 );
6      port (
7          start : in  std_logic;
8          clk   : in  std_logic;
9          pulse : out std_logic
10     );
11 end oneshot;
12
13 architecture Behavioral of oneshot is
14 signal count : integer range 0 to 3 := 0;
15 signal run   : std_logic := '0';
16 begin
17     process(start, clk, run, count)
18     begin
19         if rising_edge(start) then
20             run <= '1';
21         end if;
22         if (rising_edge(clk) and run = '1') then
23             count <= count + 1;
24         end if;
25         if (count = clk_cycles) then
26             count <=  0;
27             run   <= '0';
28         end if;
29     end process;
30     pulse <= run;
31
32 end Behavioral;
```

D. Computer software source code

D.1. Computer control software source code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <windows.h>
#include <math.h>
#include <iostream>

#include "iowkit.h"

using namespace std;

typedef struct SPIdata {
    unsigned char boardAddress;
    unsigned char functionCode;
    union {
        unsigned long dword;        // Long is 4 bytes long data word
        unsigned char byte[4];      // Has the same address as "dword" so you can address each byte (=char) of "dword" separately (but in reverse byte order on little endian machines!)
    } data;
    bool valid;
} SPIdata;

const int numDataBytes = 6;         // Number of data bytes (includes board address, function code and 4 bytes of data)
const unsigned short clippingBoards[] = {1};
const unsigned short discriminatorBoards[] = {4};
const unsigned short refChannelBoard = 2;

// Make sp_rep56 a global variable since it is used in several functions
IOWKIT56_SPECIAL_REPORT sp_rep56;

// FUNCTION dwordToBytes converts a 4 byte unsigned long to an array of 4 chars and returns the pointer to it
unsigned char* dwordToBytes (unsigned long dword) {
    // Make return array static otherwise it is gone when the function call ends
    static unsigned char bytes[4];
    bytes[0] = (unsigned char) ((0xFF000000 & dword) >> 24);
    bytes[1] = (unsigned char) ((0x00FF0000 & dword) >> 16);
    bytes[2] = (unsigned char) ((0x0000FF00 & dword) >> 8);
    bytes[3] = (unsigned char) ((0x000000FF & dword));
    return (bytes);
}

// FUNCTION bytesToDword converts an array of 4 chars to one 4 byte unsigned long. Argument is the pointer to the char array
unsigned long bytesToDword (unsigned char* bytes) {
    return ((bytes[0] << 24) | (bytes[1] << 16) | (bytes[2] << 8) | bytes[3]);
}

// FUNCTION shiftData shifts data through the SPI bus and returns the data that was shifted out
SPIdata shiftData(IOWKIT_HANDLE iowHandle, SPIdata inputData) {
    // Set all bits in sp_rep56 to 0
    memset(&sp_rep56, 0, IOWKIT56_SPECIAL_REPORT_SIZE);
    int numBytesRead;
    bool validData = true;
    SPIdata outputData = {0, 0, {0}, false};

    sp_rep56.ReportID = (UCHAR) 0x09;           // SPI-Mode
    sp_rep56.Bytes[0] = (UCHAR) numDataBytes;   // Write (and read) numBytesOfData bytes to the SPI slave
    sp_rep56.Bytes[1] = (UCHAR) 0x00;           // flags (leave 0)

    sp_rep56.Bytes[2] = inputData.boardAddress;
    sp_rep56.Bytes[3] = inputData.functionCode;
    // Be aware: reverse byte order here since data.dword is used for reading/writing!
    sp_rep56.Bytes[4] = inputData.data.byte[3];
    sp_rep56.Bytes[5] = inputData.data.byte[2];
    sp_rep56.Bytes[6] = inputData.data.byte[1];
    sp_rep56.Bytes[7] = inputData.data.byte[0];

    // VERY dirty and ugly but without the sleep the IOWarrior might hang at some point...
    Sleep(800);
    // Send data to the SPI bus and shift data out of the SPI slave
    IowKitWrite(iowHandle, IOW_PIPE_SPECIAL_MODE, (char*) &sp_rep56, IOWKIT56_SPECIAL_REPORT_SIZE);
    // Take the data that was shifted out by the previous IowKitWrite() and write it to &sp_rep56. Nothing is written to the SPI!
    numBytesRead = IowKitRead(iowHandle, IOW_PIPE_SPECIAL_MODE, (char*) &sp_rep56, IOWKIT56_SPECIAL_REPORT_SIZE);

```

```
74      // Check for valid data in sp_rep56: if board address (byte 1) is 0 there is probably something wrong
75      if (sp_rep56.Bytes[1] == (UCHAR) 0x00) validData = false;
76
77      if (sp_rep56.ReportID == 0x09 && numBytesRead == 64 && validData) {
78          outputData.boardAddress = sp_rep56.Bytes[1];
79          outputData.functionCode = sp_rep56.Bytes[2];
80          // Be aware: reverse byte order here since data.dword is used for reading/writing!
81          outputData.data.byte[3] = sp_rep56.Bytes[3];
82          outputData.data.byte[2] = sp_rep56.Bytes[4];
83          outputData.data.byte[1] = sp_rep56.Bytes[5];
84          outputData.data.byte[0] = sp_rep56.Bytes[6];
85          outputData.valid = true;
86      } else {
87          outputData.valid = false;
88      }
89      return (outputData);
90  }
91
92  // FUNCTION setDAC returns the unsigned long dword necessary to set the channel and value of the DAC7568
93  unsigned long setDAC (unsigned char channel = 0x0F, unsigned short value = 0x0FFF, unsigned char control = 0x03,
        unsigned char feature = 0x00) {
94      return ((control << 24) | ((0x0F & channel) << 20) | ((0x0FFF & value) << 8) | feature);
95  }
96
97  // FUNCTION enableDAC returns the unsigned long dword necessary to enable the internal reference of the DAC7568
98  unsigned long enableDAC (unsigned char control = 0x08, unsigned char feature = 0x01) {
99      return (setDAC(0x00, 0x0000, control, feature));
100 }
101
102 // FUNCTION getDelayDACvalue returns the decimal value for the delay DAC necessary to get the specified delay (in ns)
103 unsigned short getDelayDACvalue (double delayInNS) {
104     // The functon was derived from fitting the varactor formula to the measurement points plus a linear factor for
            the DAC/OpAmp combination
105     // Type cast the double float to a short int with "rounding" (+ 0.5)
106     short result = (short) (((158.952 * (1.04571 * (pow(163.909 / pow(delayInNS + 6.67, 2.0), 1.0 / 0.384566) - 1.0)))
            - 0.875858) + 0.5);
107     // Avoid useless values (negative or bigger than 4095)
108     if (result < 0) result = 0;
109     if (result > 4095) result = 4095;
110     return ((unsigned short) result);
111 }
112
113 // FUNCTION setAttenuation returns the unsigned long dword necessary to set the channel and value of the DAT-31R5-SP+
114 unsigned long setAttenuation (short channel = 0, double value = 0, int linearScale = 0) {
115     // Use channel value 0 to set all channels (achieved with bit pattern of 255)
116     if (channel == 0) channel = 255;
117     // The channels are selected binary by '1's so the power to the basis of 2 gives the correct value
118     // Different selection patterns can be achieved with numbers higher than 8 (e.g. all channels: 255)
119     if (channel < 9 && channel > 0) channel = (short) pow(2.0, (double) (channel - 1));
120     // If linear adjustment of attenuation in percent is needed (100 is maximum attenuation):
121     // The functon was derived from fitting the log dB formula to the measurement points and setting the "zero point"
            to 100
122     if (linearScale) value = -17.3 * log((-value + 100.0) / 100.0);
123     // Avoid useless values (negative or bigger than 63)
124     if (value < 0.0) value = 0.0;
125     if (value > 63.0) value = 63.0;
126     // Type cast the double float 'value' to a short int with "rounding" (+ 0.5)
127     return (((0x00FF & channel) << 24) | ((0x003F & ((short) (value + 0.5))) << 18));
128 }
129
130 // FUNCTION setDefaults sets the default values for all the components (attenuators, DACs) on each board
131 int setDefaults(IOWKIT_HANDLE iowHandle) {
132     int i;
133     int success = 0;
134     SPIdata data = {0, 0, {0}, false};
135
136     // Set default values for clipping boards:
137     for (i = 0; i < (int) (sizeof(clippingBoards) / sizeof(clippingBoards[0])); i++) {
138         data.boardAddress = (unsigned char) clippingBoards[i];
139         // Turn on all channels
140         data.functionCode = 0x10;
141         data.data.dword = 0x00000000;
142         if (!shiftData(iowHandle, data).valid) success += 1;
143         // Set all attenuators to 0 attenuation
144         data.functionCode = 0x20;
145         data.data.dword = 0xff000000;
146         if (!shiftData(iowHandle, data).valid) success += 10;
147         // Turn on internal reference of Clamp-DAC
148         data.functionCode = 0x30;
149         data.data.dword = enableDAC();
150         if (!shiftData(iowHandle, data).valid) success += 100;
151         // Set all Clamp-DAC outputs to maximum value
152         data.data.dword = setDAC(0x0F, 0x0FFF);
153         if (!shiftData(iowHandle, data).valid) success += 1000;
154         // Turn on internal reference of Delay-DAC
155         data.functionCode = 0x40;
156         data.data.dword = enableDAC();
157         if (!shiftData(iowHandle, data).valid) success += 10000;
158         // Disable the /CLR input pin of Delay-DAC
159         data.data.dword = 0x05000003;
160         if (!shiftData(iowHandle, data).valid) success += 100000;
161         // Set all Delay-DAC outputs to maximum value (= minimum delay)
162         data.data.dword = setDAC(0x0F, 0x0FFF);
163         if (!shiftData(iowHandle, data).valid) success += 1000000;
164     }
165
```

```
166     // Set default values for discriminator boards:
167     for (i = 0; i < (int) (sizeof(discriminatorBoards) / sizeof(discriminatorBoards[0])); i++) {
168         data.boardAddress = (unsigned char) discriminatorBoards[i];
169         // Turn on internal reference of Thres-DAC
170         data.functionCode = 0xA0;
171         data.data.dword = enableDAC();
172         if (!shiftData(iowHandle, data).valid) success += 10000000;
173         // Set all Thres-DAC outputs to medium value
174         data.data.dword = setDAC(0x0F, 0x0200);
175         if (!shiftData(iowHandle, data).valid) success += 100000000;
176     }
177     return (success);
178 }
179
180 // FUNCTION switchAllChannels switches all channels on all boards on or off. In case of 'off' one can leave on the
        reference channel
181 bool switchAllChannels(IOWKIT_HANDLE iowHandle, int on = 0, int refChannelOn = 1) {
182     int i;
183     bool success = true;
184     SPIdata data = {0, 0, {0}, false};
185     // Switch all channels on all clipping boards
186     for (i = 0; i < (int) (sizeof(clippingBoards) / sizeof(clippingBoards[0])); i++) {
187         data.boardAddress = (unsigned char) clippingBoards[i];
188         data.functionCode = 0x10;
189         if (!on && (clippingBoards[i] == refChannelBoard) && refChannelOn) {
190             // Leave reference channel turned on
191             data.data.dword = 0xFE000000;
192         } else if (!on) {
193             data.data.dword = 0xFF000000;
194         } else {
195             data.data.dword = 0x00000000;
196         }
197         if (!shiftData(iowHandle, data).valid) success = false;
198     }
199     return (success);
200 }
201
202 // FUNCTION calculateRate returns the number of counts, the duration of the counting process and the resulting
        frequency from the input SPIdata
203 float* calculateRate(SPIdata input) {
204     // Make return array static otherwise it is gone when the function call ends
205     static float result[3];
206     int counterData = 256 * 256 * input.data.byte[2] + 256 * input.data.byte[1] + input.data.byte[0];
207     int triggerCounts = counterData & 0x3FF;
208     int clockCounts = counterData >> 10;
209     float clockFreq = 10000000.0 / 4096.0;
210     float duration = clockCounts / clockFreq;
211     float freq = triggerCounts / duration;
212
213     result[0] = (float) triggerCounts;
214     result[1] = duration;
215     result[2] = freq;
216
217     return (result);
218 }
219
220 // FUNCTION getTimeString returns a formatted string containing the current system time
221 char* getTimeString() {
222     // Make return array static otherwise it is gone when the function call ends
223     static char timeString[21];
224     time_t now = time(NULL);
225     struct tm* timeStamp = localtime(&now);
226     strftime(timeString, sizeof(timeString), "%Y-%m-%d_%Hh%Mm%Ss", timeStamp);
227     return (timeString);
228 }
229
230 // FUNCTION createFileName returns a file name based on the parameters given and the current system time
231 char* createFileName(char* measurement, unsigned short boardNumber, int channel) {
232     // Make return array static otherwise it is gone when the function call ends
233     static char fileName[80];
234     sprintf(fileName, "board%hi_ch%i_%s_%s.dat", boardNumber, channel, measurement, getTimeString());
235     return (fileName);
236 }
237
238 // FUNCTION writeDataToFile writes data to the specified file line by line in append mode. Returns true on success
239 bool writeDataToFile(char* fileName, double column1, double column2, int column3 = -1, char firstCharOfLine = ' ') {
240     bool success = false;
241     FILE *filePointer;
242     filePointer = fopen(fileName, "a");
243     // Check if opening file worked
244     if (filePointer != NULL)
245         // Check if writing data to file works
246         if (fprintf(filePointer, "%c%.8Le\t%.8Le\t%i\n", firstCharOfLine, column1, column2, column3) > 0);
247             success = true;
248     fclose(filePointer);
249     return (success);
250 }
251
252 bool measurement(IOWKIT_HANDLE iowHandle, char mode = 'd', int firstChannel = 1, int lastChannel = 8) {
253     int i, channel;
254     float currentRate = 0, previousRate = 0, maxRate, erfWidth, fitStartValue;
255     double value, startValue, stopValue, step;
256     bool success = true;
257     SPIdata data = {0, 0, {0}, false}, result = {0, 0, {0}, false};
258     char fileName[100], modeName[15], gpCommand[100], valueUnit[3], commentSign;
259     FILE *filePointer;
```

```
260
261     if (mode == 'd') {
262         strcpy(modeName, "delay");
263         strcpy(valueUnit, "ns");
264         startValue = 6.0;
265         stopValue = -0.01;
266         step = 0.5;
267         erfWidth = 1.25;
268         fitStartValue = 0.5;
269
270         printf ("Automatic delay measurement started.\n");
271         rename ("delayValues.dat", createFileName("delayValues", 0, 0));
272
273         // Set discriminator to reasonable value for only one active channel
274         data.boardAddress = (unsigned char) discriminatorBoards[0];
275         data.functionCode = 0xA0;
276         data.data.dword = setDAC(0x0F, 0x01A0);
277         if (!shiftData(iowHandle, data).valid) success = false;
278         // Set reference channel to medium delay
279         data.boardAddress = refChannelBoard;
280         data.functionCode = 0x40;
281         data.data.dword = setDAC(0x00, getDelayDACvalue(2.5));
282         if (!shiftData(iowHandle, data).valid) success = false;
283         // Turn off all channels on all boards but leave reference channel on
284         success = switchAllChannels(iowHandle, 0, 1);
285
286     } else if (mode == 'f') {
287         strcpy(modeName, "flatfielding");
288         strcpy(valueUnit, "%");
289         startValue = 72.0;
290         stopValue = -0.01;
291         step = 6.0;
292         erfWidth = 0.1;
293         fitStartValue = 10.0;
294
295         printf ("Automatic amplitude flatfielding started.\n");
296         rename ("flatfieldValues.dat", createFileName("flatfieldValues", 0, 0));
297
298         // Set discriminator to desired amplitude value
299         data.boardAddress = (unsigned char) discriminatorBoards[0];
300         data.functionCode = 0xA0;
301         data.data.dword = setDAC(0x0F, 0x0370);
302         if (!shiftData(iowHandle, data).valid) success = false;
303         // Turn off all channels on all boards
304         success = switchAllChannels(iowHandle, 0, 0);
305
306     } else {
307         return (false);
308     }
309
310     // Go through all clipping boards
311     for (i = 0; i < (int) (sizeof(clippingBoards) / sizeof(clippingBoards[0])); i++) {
312
313         printf ("Selected board %hi.\n", clippingBoards[i]);
314
315         // Go through all channels on selected board
316         for (channel = (firstChannel - 1); channel < lastChannel; channel++) {
317             maxRate = 0;
318             // Skip reference channel in delay measurement mode
319             if (mode != 'd' || !(clippingBoards[i] == refChannelBoard && channel == 0)) {
320                 strcpy(fileName, createFileName(modeName, clippingBoards[i], channel + 1));
321                 data.boardAddress = (unsigned char) clippingBoards[i];
322
323                 // Turn on channel
324                 data.functionCode = 0x10;
325                 // Move bitpattern 1111 1110 1111 1111 bitwise to the left to turn on desired channel (done by the
                        "0")
326                 data.data.dword = (0xFEFF0000 << channel);
327                 // Leave reference channel on in delay measurement mode
328                 if (mode == 'd' && clippingBoards[i] == refChannelBoard) data.data.dword = (0xFE000000 & data.data.
                        dword);
329                 if (!shiftData(iowHandle, data).valid) success = false;
330
331                 printf ("Measuring channel %i: ", channel + 1);
332
333                 previousRate = 0;
334                 for (value = startValue; value > stopValue; value -= step) {
335                     data.boardAddress = (unsigned char) clippingBoards[i];
336                     if (mode == 'd') {
337                         // Set delay DAC
338                         data.functionCode = 0x40;
339                         data.data.dword = setDAC((0x07 & ((unsigned char) channel)), getDelayDACvalue(value));
340                     } else if (mode == 'f') {
341                         // Set Attenuator
342                         data.functionCode = 0x20;
343                         data.data.dword = setAttenuation((short) (channel + 1), value, 1);
344                     }
345                     if (!shiftData(iowHandle, data).valid) success = false;
346
347                     // Select discriminatorboard. Prototype system has only one
348                     data.boardAddress = (unsigned char) discriminatorBoards[0];
349                     data.functionCode = 0xB0;
350                     if (mode == 'd') {
351                         // Measure timing on discriminator channel 2. Prototype system only uses channel 2
352                         data.data.dword = 0x40000000;
353                     } else if (mode == 'f') {
```

```
354                         // Measure trigger on discriminator channel 2. Prototype system only uses channel 2
355                         data.data.dword = 0x10000000;
356                     }
357                     if (!shiftData(iowHandle, data).valid) success = false;
358 
359                     printf ("%.1f%s", value, valueUnit);
360 
361                     // Start measuring process
362                     // use function code 0 for reading since no function is assigned to it
363                     data.functionCode = 0x00;
364                     // read once to clear result from previous data shift
365                     shiftData(iowHandle, data);
366                     while (calculateRate(result = shiftData(iowHandle, data))[1] == 0.0) {
367                         printf (".");
368                     }
369                     currentRate = calculateRate(result)[2];
370                     // Calculate maximum rate of counts and use it as start value to improve fitting
371                     if (currentRate > maxRate) maxRate = currentRate;
372                     // Comment out data if current rate is more than 2.0Hz lower than previously counted rate to avoid
                            wrong rates after the pulse already "passed"
373                     commentSign = '#';
374                     if (currentRate > (previousRate - 2.0)) {
375                         commentSign = ' ';
376                         previousRate = currentRate;
377                     }
378                     writeDataToFile("measurement.dat", value, currentRate, (int) calculateRate(result)[0], commentSign
                            );
379                 }
380                 printf ("Done\n");
381                 printf ("Calculating optimal value... ");
382                 // On Windows systems use "wgnuplot" instead of "gnuplot". It is much faster!
383                 sprintf(gpCommand, "wgnuplot -e \"a=%f; b=%f; value=%f\" fitErrorFunction.gpt", maxRate / 2, erfWidth,
                        fitStartValue);
384                 system (gpCommand);
385                 rename ("measurement.dat", fileName);
386                 filePointer = fopen ("fitted.value", "r");
387                 if (filePointer != NULL) {
388                     fscanf (filePointer, "%lf", &value);
389                     fclose (filePointer);
390                     remove ("fitted.value");
391                     printf ("Done: %f%s\n", value, valueUnit);
392                     data.boardAddress = (unsigned char) clippingBoards[i];
393                     if (mode == 'd') {
394                         writeDataToFile("delayValues.dat", value, clippingBoards[i], channel + 1);
395                         // Set Delayline to the derived optimal value
396                         data.functionCode = 0x40;
397                         data.data.dword = setDAC((0x07 & ((unsigned char) channel)), getDelayDACvalue(value));
398                     } else if (mode == 'f') {
399                         writeDataToFile("flatfieldValues.dat", value, clippingBoards[i], channel + 1);
400                         // Set Attenuator to the derived optimal value
401                         data.functionCode = 0x20;
402                         data.data.dword = setAttenuation((short) (channel + 1), value, 1);
403                     }
404                     if (!shiftData(iowHandle, data).valid) success = false;
405                 } else {
406                     printf ("Error: return file 'fitted.value' could not be opened.\nMaybe Gnuplot could not fit the
                            function.\n");
407                 }
408             }
409         }
410         // Turn off all channels on the board
411         data.boardAddress = (unsigned char) clippingBoards[i];
412         data.functionCode = 0x10;
413         data.data.dword = 0xFF000000;
414         // Leave reference channel on
415         if (mode == 'd' && clippingBoards[i] == refChannelBoard) data.data.dword = 0xFE000000;
416         if (!shiftData(iowHandle, data).valid) success = false;
417     }
418     success = switchAllChannels(iowHandle, 1);
419     return (success);
420 }
421 
422 bool rateScan(IOWKIT_HANDLE iowHandle) {
423     bool success;
424     int i;
425     SPIdata data = {0, 0, {0}, false}, result = {0, 0, {0}, false};
426     char fileName[100];
427 
428     const float convFactor = 37.02;
429     // Define the measurement points
430     const float phe[] = {4.5, 5, 5.5, 6, 7, 8, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5, 15, 17, 20, 23, 25, 30, 50};
431 
432     // Select discriminatorboard. Prototype system has only one
433     data.boardAddress = (unsigned char) discriminatorBoards[0];
434     strcpy(fileName, createFileName("rateScan", discriminatorBoards[0], 0));
435     printf ("Automatic rate scan started.\n");
436 
437     for (i = 0; i < (int) (sizeof(phe) / sizeof(phe[0])); i++) {
438         // Set value of DAC for comparator voltage
439         data.functionCode = 0xA0;
440         data.data.dword = setDAC(0x01, (unsigned short) ((phe[i] * convFactor) + 0.5));
441         if (!shiftData(iowHandle, data).valid) success = false;
442 
443         data.functionCode = 0xB0;
444         // Measure trigger on discriminator channel 2. Prototype system only uses channel 2
445         data.data.dword = 0x10000000;
```

```
446         if (!shiftData(iowHandle, data).valid) success = false;
447
448         printf ("Discriminator threshold: %.1f%s", phe[i], "PhE");
449
450         // Start measuring process
451         // use function code 0 for reading since no function is assigned to it
452         data.functionCode = 0x00;
453         // read once to clear result from previous data shift
454         shiftData(iowHandle, data);
455         while (calculateRate(result = shiftData(iowHandle, data))[1] == 0.0) {
456             printf (".");
457         }
458         writeDataToFile(fileName, phe[i], calculateRate(result)[2], (int) calculateRate(result)[0], ' ');
459         printf ("Rate: %f%s\n", calculateRate(result)[2], "Hz");
460     }
461     return (success);
462 }
463
464
465 int main(int argc, char* argv[]) {
466
467     IOWKIT_HANDLE iowHandle;
468     //char input[50] = {0};
469     char inputMode;
470     short inputValue = 0;
471     SPIdata data = {0, 0, {0}, false}, result = {0, 0, {0}, false};
472     int returncode, firstChannel, lastChannel;
473     unsigned char oldFunctionCode, channel;
474     double delay = 0, attenuation = 0;
475
476     // Open connected device
477     iowHandle = IowKitOpenDevice();
478
479     while(IowKitGetProductId(iowHandle) != IOWKIT_PID_IOW56) {
480         printf ("IOWarrior 56 could not be found.\n");
481         printf ("Please quit, connect and restart.\n");
482         printf ("q to quit.\n");
483         scanf (" %c", &inputMode);
484         if (inputMode == 'q') exit(0);
485
486         // Try again to open connected device
487         iowHandle = IowKitOpenDevice();
488     }
489
490     // Set the read IO timeout to 1000ms
491     IowKitSetTimeout(iowHandle, 1000);
492
493     // Enable SPI mode
494     // Set all bits in sp_rep56 to 0
495     memset(&sp_rep56, 0, IOWKIT56_SPECIAL_REPORT_SIZE);
496
497     sp_rep56.ReportID = 0x08; // SPI-Mode
498     sp_rep56.Bytes[0] = 0x01; // Enable SPI-Mode
499     sp_rep56.Bytes[1] = 0x00; // MSB first, SPI mode 0
500     sp_rep56.Bytes[2] = 0x01; // 12MBit
501
502     // Set the device to SPI mode. Nothing is written to the SPI bus itself!
503     IowKitWrite(iowHandle, IOW_PIPE_SPECIAL_MODE, (char*) &sp_rep56, IOWKIT56_SPECIAL_REPORT_SIZE);
504
505     // Write and read data
506
507     while (inputMode != 'q') {
508         printf ("\n");
509         printf ("a = board address    (%hi)\n", (short) data.boardAddress);
510         printf ("f = function code    (%hi)\n", ((short) data.functionCode >> 4));
511         printf ("d = data and shift   (%08lx)\n", data.data.dword);
512         printf ("t = set delay [ns]   (%f)\n", delay);
513         printf ("h = set att. [0.5dB] (%f)\n", attenuation);
514         printf ("s = set default values\n");
515         printf ("c = count\n");
516         printf ("r = readout\n");
517         printf ("m = measurements\n");
518         printf ("q = quit\n");
519         printf ("Enter mode: ");
520         scanf (" %c", &inputMode);
521         printf ("\n");
522
523         if (inputMode == 'a') {
524             printf ("Board address (int) [%hi]: ", (short) data.boardAddress);
525             scanf ("%hi", &inputValue);
526             data.boardAddress = (unsigned char) (0x0F & inputValue);
527         } else if (inputMode == 'f') {
528             printf ("Function code (int) [%hi]: ", ((short) data.functionCode >> 4));
529             scanf ("%hi", &inputValue);
530             // Function code is only the first nibble of the second byte, so inputValue needs to be shifted by 4 bits
531             data.functionCode = (unsigned char) ((0x0F & inputValue) << 4);
532         } else if (inputMode == 'd') {
533             printf ("Data (4 byte DWORD) [%08lx]: ", data.data.dword);
534             scanf ("%lx", &data.data.dword);
535             shiftData(iowHandle, data);
536         } else if (inputMode == 't') {
537             printf ("Select channel (1-8; 0 = all) [%hi]: ", inputValue);
538             scanf ("%hi", &inputValue);
539             if (inputValue > 0 && inputValue < 9) {
540                 // Substract 1 (DAC channels are actually addressed from 0 to 7), cast to char and use only last 3
                        bits of second nibble
```

```
541                 channel = (0x07 & ((unsigned char) (inputValue - 1)));
542             } else {
543                 // default: set all channels
544                 channel = 0x0F;
545             }
546             printf ("Set delay for channel %hi in ns [%f]: ", inputValue, delay);
547             scanf ("%lf", &delay);
548             // temporarily save current function code
549             oldFunctionCode = data.functionCode;
550             // use function code 4 to set the delay DACs
551             data.functionCode = 0x40;
552             data.data.dword = setDAC(channel, getDelayDACvalue(delay));
553             shiftData(iowHandle, data);
554             //restore old function code from before
555             data.functionCode = oldFunctionCode;
556         } else if (inputMode == 'h') {
557             printf ("Select channel (1-8; 0 = all) [%hi]: ", inputValue);
558             scanf ("%hi", &inputValue);
559             printf ("Set attenuation for channel %hi in %% [%f]: ", inputValue, attenuation);
560             scanf ("%lf", &attenuation);
561             // temporarily save current function code
562             oldFunctionCode = data.functionCode;
563             // use function code 2 to set the attenuation
564             data.functionCode = 0x20;
565             data.data.dword = setAttenuation(inputValue, attenuation, 1);
566             shiftData(iowHandle, data);
567             //restore old function code from before
568             data.functionCode = oldFunctionCode;
569         } else if (inputMode == 's') {
570             printf ("Default values will be set...\n");
571             returncode = setDefaults(iowHandle);
572             if (returncode == 0) {
573                 printf ("Default values are set.\n");
574             } else {
575                 printf ("Default values could NOT be set properly!\n");
576                 printf ("Error code: %i\n", returncode);
577             }
578         } else if (inputMode == 'c') {
579             printf ("0 = count trigger 1\n");
580             printf ("1 = count trigger 2\n");
581             printf ("2 = count trigger ref\n");
582             printf ("3 = count timing  1\n");
583             printf ("4 = count timing  2\n");
584             printf ("Enter count mode: ");
585             scanf (" %c", &inputMode);
586             data.functionCode = 0xB0;
587             data.data.dword = ((0x0F & inputMode) << 28) | 0x000000;
588             shiftData(iowHandle, data);
589         } else if (inputMode == 'r') {
590             // temporarily save current function code
591             oldFunctionCode = data.functionCode;
592             // use function code 0 for reading since no function is assigned to it
593             data.functionCode = 0x00;
594             // read once to clear result from previous data shift
595             shiftData(iowHandle, data);
596             while (calculateRate(result = shiftData(iowHandle, data))[1] == 0.0) {
597                 printf (".");
598                 Sleep(500);
599             }
600             printf ("\nTrigger counts: %f", calculateRate(result)[0]);
601             printf ("\nDuration [s]:   %f", calculateRate(result)[1]);
602             printf ("\nFrequency [Hz]: %f\n", calculateRate(result)[2]);
603             //restore function code from before counting
604             data.functionCode = oldFunctionCode;
605         } else if (inputMode == 'm') {
606             printf ("d = delay adjustment\n");
607             printf ("f = flatfielding\n");
608             printf ("r = rate scan\n");
609             printf ("Enter measurement mode: ");
610             scanf (" %c", &inputMode);
611             if (inputMode == 'r') {
612                 if (!rateScan(iowHandle)) printf ("Errors occured.\n");
613             } else {
614                 firstChannel = 1;
615                 lastChannel = 8;
616                 printf ("Enter first channel [%i]: ", firstChannel);
617                 scanf ("%i", &firstChannel);
618                 printf ("Enter last channel [%i]: ", lastChannel);
619                 scanf ("%i", &lastChannel);
620                 if (!measurement(iowHandle, inputMode, firstChannel, lastChannel)) printf ("Errors occured.\n");
621             }
622         }
623     }
624
625
626     // Disable SPI mode. Nothing is written to the SPI bus itself!
627     memset(&sp_rep56, 0, IOWKIT56_SPECIAL_REPORT_SIZE);
628     sp_rep56.ReportID = 0x08; // SPI-Mode
629     sp_rep56.Bytes[0] = 0x00; // Disable SPI-Mode
630     IowKitWrite(iowHandle, IOW_PIPE_SPECIAL_MODE, (char*) &sp_rep56, IOWKIT56_SPECIAL_REPORT_SIZE);
631
632     // Close device
633     IowKitCloseDevice(iowHandle);
634
635     return 0;
636 }
```

D.2. Gnuplot macro for fitting the error function

```
# The error function with coefficients a, b, value
f(x) = -a * (erf(b * (x - value)) - 1)

# Use the file "measurement.dat" for fitting where the measured data was stored before by the c program
fit f(x) "measurement.dat" using 1:2 via a, b, value

# Output the derived value of "value" to the file "fitted.value"
set print "fitted.value"
print value
```

Bibliography

[1] Michael T. Rissi, Nepomuk Otte, Thomas Schweizer, and Maxim Shayduk. A New Sum Trigger to Provide a Lower Energy Threshold for the MAGIC Telescope. *IEEE Transactions on Nuclear Science*, 56(6):3840–3843, 2009.

[2] Maxim Shayduk. Detection of the Crab Pulsar above 25 GeV with the MAGIC Telescope. *AIP Conference Proceedings*, 1112(1):72–78, 2009.

[3] James W. Cronin, Thomas K. Gaisser, and Simon P. Swordy. Cosmic Rays at the Energy Frontier. *Scientific American*, 276:44, 1997. http://astroparticle.uchicago.edu/siteold/cosmic_ray_spectrum_picture.htm.

[4] Michael T. Rissi. *Detection of Pulsed Very High Energy Gamma-Rays from the Crab Pulsar with the MAGIC telescope using an Analog Sum Trigger*. PhD thesis, Swiss Federal Institute of Technology Zurich, 2009.

[5] Robert M. Wagner. *Measurement of Very High Energy Gamma-Ray Emission from Four Blazars Using the MAGIC Telescope and a Comparative Blazar Study*. PhD thesis, Technische Universität München, 2006.

[6] Malcolm S. Longair. *High Energy Astrophysics*, volume 1. Cambridge University Press, 1992.

[7] Robert M. Wagner. VHE γ-ray Sky Map. http://www.mppmu.mpg.de/~rwagner/sources/, 2010.

[8] Govert Schilling. *Flash! The hunt for the biggest explosions in the universe*. Cambridge University Press, 1st edition, 2002.

[9] Ray W. Klebesadel, Ian B. Strong, and Roy A. Olson. Observations of Gamma-Ray Bursts of Cosmic Origin. *Astrophysical Journal*, 182:L85–L88, jun 1973.

[10] Fabian Schmidt. CORSIKA Shower Images. http://www.ast.leeds.ac.uk/~fs/showerimages.html, 2005.

[11] Marcos López Moya. *Astronomía Gamma con el Telescopio MAGIC: Observaciones de la Nebulosa y Pulsar del Cangrejo*. PhD thesis, Universidad Complutense de Madrid, 2006.

[12] Hendrik Bartko, Florian Goebel, Razmik Mirzoyan, Wendelin Pimpl, and Masahiro Teshima. Tests of a prototype multiplexed fiber-optic ultra-fast FADC data acquisition system for the MAGIC telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 548(3):464–486, 2005.

[13] Razmik Mirzoyan, Juan Cortina, Eckart Lorenz, Manel Martinez, Alexander Ostankov, and David Paneque. Ultrafast FADC Multiplexer. *Nuclear Science, IEEE Transactions on*, 49(5):2473–2476, oct. 2002.

[14] Markus Gaug. *Calibration of the MAGIC Telescope and Observation of Gamma Ray Bursts*. PhD thesis, Universitat Autonoma de Barcelona, 2006.

[15] The MAGIC Collaboration. Observation of Pulsed gamma-Rays Above 25 GeV from the Crab Pulsar with MAGIC. *Science*, 322(5905):1221–1224, 2008.

[16] Ulrich Tietze and Christoph Schenk. *Halbleiter-Schaltungstechnik*. Springer-Verlag, 12th edition, 2002.

[17] Maxim Shayduk, Razmik Mirzoyan, Vinzent Zimmer, Eckart Lorenz, Thomas Schweizer, and Masahiro Teshima. Selection of photomultipliers for the Cherenkov Telescope Array facility. *Nuclear Instruments and Methods in Physics Research A*, 617:422–423, may 2010.

[18] Daniel Mazin. *A study of very high energy γ-ray emission from AGNs and constraints on the extragalactic background light*. PhD thesis, Technische Universität München, 2007.

[19] Acam Messelectronic GmbH. The TDC Cookbook. http://www.acam-usa.com/tdc_cookbook_e.pdf, 2002.

[20] Burle Industries Inc. Photomultiplier Handbook. http://psec.uchicago.edu/links/Photomultiplier_Handbook.pdf, 1998.

[21] Hamamatsu Photonics K.K. Photomultiplier Tubes - Basics and Applications. http://sales.hamamatsu.com/assets/applications/ETD/pmt_handbook_complete.pdf, 2006.

[22] Eric W. Weisstein. Normal Distribution Function. http://mathworld.wolfram.com/NormalDistributionFunction.html, 2010.

[23] Sergei Winitzki. A handy approximation for the error function and its inverse. http://homepages.physik.uni-muenchen.de/~Winitzki/erf-approx.pdf, 2008.

[24] Rhombus Industries. Passive Delay Line Design Considerations. http://www.rhombus-ind.com/dlcat/app1_pas.pdf, 1998.

[25] Rhombus Industries. Electromagnetic Delay Lines. http://www.rhombus-ind.com/app-note/passive.pdf, 1997.

[26] Gregor Schenke. Lecture "Industrieelektronik". http://www.et-inf.fho-emden.de/~elmalab/indelek/download/Ind_5.pdf, 2008.

[27] *Simulation of a Varactor Diode. A Final Exercise for Laboratory of Mathematics*. ICEE, 2003. http://www.ineer.org/events/icee2003/proceedings/pdf/1469.pdf.

[28] Paul S. Linsay. Period doubling and chaotic behavior in a driven anharmonic oscillator. *Physical Review Letters*, 47:1349–1352, nov 1981.

[29] Christian Schulz. *Entwicklung eines neuartigen Niederdruck-Detektors mit hoher Orts- und Flugzeitauflösung zum Nachweis thermischer Neutronen.* PhD thesis, Freie Universität Berlin, 1999. http://www.diss.fu-berlin.de/diss/receive/FUDISS_thesis_000000000275.

[30] Kenneth A. Connor. Artificial (Lumped Element) Transmission Line. http://hibp.ecse.rpi.edu/~connor/education/Fields/lumpline.pdf, 2000.

[31] Lothar Miller. Einfacher SPI-Master / Mode 0. http://www.lothar-miller.de/s9y/categories/45-SPI-Master, 2009.

[32] Lothar Miller. SPI-Slave im CPLD. http://www.lothar-miller.de/s9y/categories/26-SPI-Slave, 2009.

Acknowledgements

First of all I would like to thank Masahiro Teshima and Razmick Mirzoyan for giving me the opportunity to work in the vital and dynamic MAGIC group at the Max-Planck-Institute for Physics, Munich.

My very special thanks go to Thomas Schweizer, the creative mastermind of the Sum-trigger project and the initiator of the improved design. He always provided me with the essential ideas and solutions for any kind of problems that appeared during the development and testing of the prototype. It was a pleasure to work with him.

No less I want to thank Maxim Shayduk for his indispensable support and his efforts to explain me the theoretical background to physical and electronic issues.

I am deeply grateful to Eckart Lorenz for his patient help and support and for sacrificing his time to read through and to correct my thesis until the very end.

Similarly I am indebted to Tobias Jogler for his careful revision of my manuscript.

Furthermore I want to acknowledge Christian Kiesling for supervising my thesis and thus giving me the opportunity to work on this great subject.

I like to express my appreciation to Ronald Maier and Antonius Dettlaff of the MPI electronics workshop for their suggestions to the electronic circuit design and for the layout of the clipping and discriminator boards.
Likewise I owe many thanks to Hermann Wenninger, Christin Fesenmeir and Michael Reitmeier for the diligent assembly of all my printed circuit boards and 30 delay line modules, as well as Markus Fras, David Fink, and Olaf Reimann for their patience to answer my many questions about digital and analog electronics.

My heartfelt thanks go to the great colleagues and friends in the MAGIC Munich group, who made this one year of my life an unforgettable experience.

I owe my deepest gratitude to my family and friends, especially my parents and my love Anke, who have endured me in the demanding time of writing this thesis and supported me in every way they could. And my grandparents, who enriched my life and always believed in me.