Hardware Implementation and Testing of the Module Back-End Electronics for the LOFT Mission

Diplomarbeit
von
Michael Gschwender

Insitut für Astronomie und Astrophysik Tübingen
Abteilung Hochenergieastrophysik
Eberhard-Karls-Universität Tübingen

Juli 2014

The time will come when diligent research over long periods will bring to light things that now lie hidden. A single life time, even though entirely devoted to research, would not be enough for the investigation of so vast a subject. . . . And so this knowledge will be unfolded through long successive ages. There will come a time when our descendants will be amazed that we did not know things that are so plain to them. . . . Many discoveries are reserved for ages still to come, when memory of us will have been effaced. Our universe is a sorry little affair unless it has in it something for every age to investigate . . . . Nature does not reveal her mysteries once and for all.

— Lucius Annaeus Seneca (c. 4 BC – AD 65), naturales quaestiones

Hiermit versichere ich, dass ich die vorliegende Arbeit entsprechend §18 Abs. (7) der Diplom-Prüfungsordnung Physik vom 10. August 2002 selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Tübingen, den 29. Juli 2014

Abstract
The Large Observer For X-ray Timing (LOFT) is a proposed mission in the Cosmic Vision program of the European Space Agency (ESA). LOFT consists of two main instruments, the Large Area Detector (LAD) and the Wide Field Monitor (WFM), and is committed to the exploration of dense matter and strong-field gravity. As such its target sources include neutron stars and black holes.
The LAD consists of modules, holding the detectors and a dedicated read-out electronics. At the end of each panel, a Module Back-End Electronics (MBEE) is placed. This MBEE handles the data from a module and transmits them to another electronic higher in the hierarchy. This thesis describes the existing MBEE Prototype developed at IAAT and its implementation in VHDL[1].

Deutsche Zusammenfassung
LOFT (Large Observer For X-ray Timing) ist ein Missionskandidat innerhalb des Cosmic Vision Programm der Europäischen Raumfahrtagentur ESA. Es besteht aus zwei Hauptinstrumenten, dem sogenannten Large Area Detector (LAD) und dem Wide Field Monitor (WFM). Deren Aufgabengebiet umfasst die Erforschung von dichter Materie, wie sie zum Beispiel in Neutronen Sternen vorkommt und extrem starker Gravitation wie man sie um Schwarze Löcher findet.
Der LAD ist modular aufgebaut, wobei sich auf jedem Modul die Detetektoren und ihre jeweiligen Auslese-Elektroniken befinden. Zustandig für solch ein Modul ist die Module Back-End Electronics (MBEE), welche für die Kommunikation mit den Detetektoren und das Auswerten ihrer Daten konzipiert ist. Ein Prototyp für diese MBEE wurde bereits am IAAT entwickelt und Ziel dieser Diplomarbeit war die Implementeriung der nötigen Prozesse der MBEE mithilfe der Programmiersprache VHDL[2].

[1]Very High Speed Integrated Circuit Hardware Description Language: A hardware description language
[2]Very High Speed Integrated Circuit Hardware Description Language: Eine Hardware Programmiersprache.

Contents

Chapter 1

Introduction

Astronomy has accompanied mankind through the ages, and as technology developed, so did astronomy. But astronomy has always been more, the stars have always been there. And with no information, with no possibility to understand, mankind could only gaze in wonder. There was something to learn from the sky, something seemingly directed at us, because it showed us the years, the seasons, when the herds would wander, how to navigate on see and land. Through all times we took these messages, from signals of the gods in the antique, to comets predicting fires in Newton's age. It fueled our imagination and it influenced our picture of ourselves, by showing us our ever changing place in the universe, up to present time. Finally, no bystander anymore, explorations can be made, and the messages still help us to understand more than just the stars.

Today astronomy covers all of the electromagnetic spectrum, and reaches from the radio to the high energy gamma ray band (Figure 1.1). There are observatories all around the globe and satellite missions in orbit around the Earth, and even a few traveling our solar system.

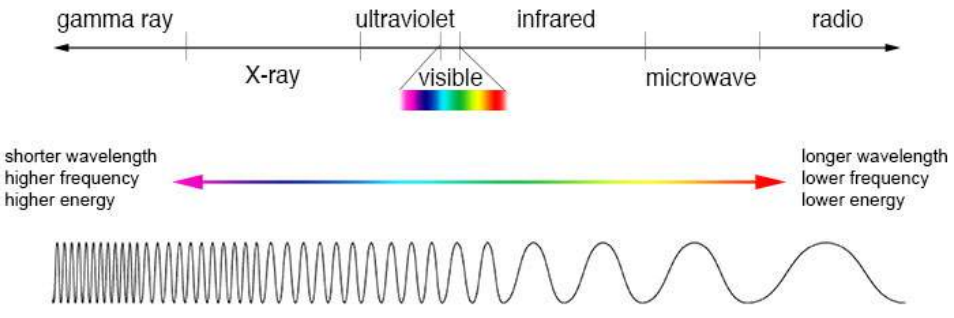


Figure 1.1: The electromagnetic spectrum of light. Typical wavelengths for visible light lie between 390 - 700 nm. X-rays have wavelengths from 0.01 nm up to 10 nm and cover the energy range from 100 eV up to 100 keV. Image: NASA

The following chapters present the field of X-ray astronomy by introducing the early

historical discoveries, while providing explanations as they are needed. This particular approach is taken from Seward and Charles (2010).

1.1 A fundamental discovery

In Würzburg 1895, Wilhelm Röntgen, when experimenting with electrical discharge tubes, noticed that they seemed to be the source of a yet unknown radiation. This radiation would pass through paper and even wood, but seemed to be stopped by heavy metals. Röntgen named this form of electromagnetic radiation X-rays.
The dominant interaction for X-rays, with energies from 0.1 - 10 keV, with matter is the photoelectric effect (Longair, 2011). Here an X-ray photon with the energy E_{ph}, gets absorbed by an atom, which ejects an electron with the energy $E_{el} = E_{ph} - E_B$, where E_B is the binding energy of the electron. This absorption process is most effective for low X-ray photon energies and materials with high Z[1]. A beam of X-rays that passed through a material, will have fewer photons, but each individual photon still has the same energy. The intensity of the transmitted beam weakens exponentially depending on the thickness and the absorption coefficient (Henke et al., 1993) of the material.
For higher energies, i.e for silicon at ≈ 30 keV, the probability for Compton scattering increases. The energy of the incoming X-ray is distributed via inelastic scattering to an electron and the scattered photon. For photon energies above 1.02 MeV, all the energy can be used to from an electron-positron pair. This process is known as pair production and is more important for the detection of γ-rays.

The first applications in medicine were obvious, since Röntgen himself already showed an X-ray image of his wife's hand, where the X-rays would pass through the flesh and are absorbed by the bones. Astronomical applications, however, suffered from several problems. First of all, the flux of Röntgens X-rays was much higher than that from astronomical sources, furthermore, nearly all the X-rays are absorbed by the Earth's atmosphere, as shown in Figure 1.2.

To detect, for example 3 keV X-rays from an extraterrestrial source, one would have to go above 99 Percent of the Earth's atmosphere (Seward and Charles, 2010, p. 2). It took science almost 60 years from Röntgens discovery, to slowly make its way to higher altitudes, using balloons and rocket experiments.

[1] The atomic number of an element, equal to its number of protons

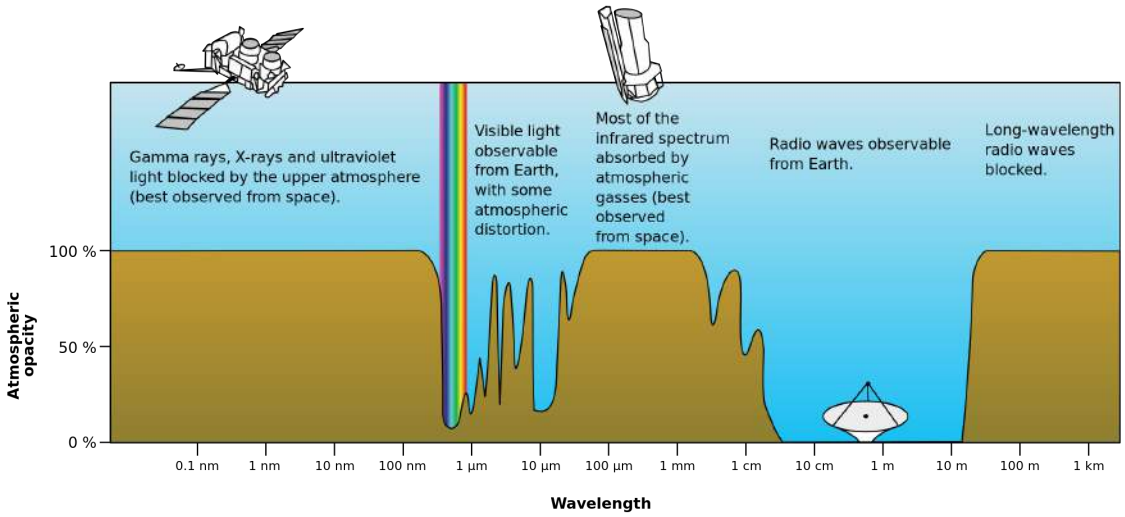


Figure 1.2: The opacity of the atmosphere as a function of wavelength. While visible light and radio waves can penetrate the atmosphere to be observed from the surface, X-rays are absorbed already at high altitudes. Image: NASA

1.2 Early X-ray astronomy

In 1949 a detector placed on a rocket launched by the U.S. Naval Research Laboratory (NRL), was able to detect the first X-rays from the Sun. Under the direction of Herbert Friedman, the NRL used modified German V2 (Figure 1.3) rockets to reveal the Sun as a UV and X-ray source (Friedman et al., 1951).
The optical spectrum of the Sun, can be best described as the spectrum of a black body[2], with the temperature of 6000 K. The maximum intensity of the spectrum can be derived through Planck's law, while the temperature of the black body can be obtained using Wien's approximation. To generate X-rays through black body emission, the object's temperature on the surface has to be around 10^6 K. This means that the X-rays observed from the Sun can not come from the same region as the optical photons or they have to be generated by another process.

X-rays, created by black body emission, are considered thermal X-rays, due to their generating process. Another example for thermal X-rays, are X-rays generated by bremsstrahlung. When an electron passes a positive charged particle, it emits radiation due to acceleration. In hot ionized gas (i.e a Plasma) the velocity distribution of the electrons follows a maxwellian distribution, resulting in a continuum of radiation which is solely dependent on the temperature. For a temperature of above 1 million degrees, the maximum of this distribution is in the X-ray regime. This is one responsible process for the generation of the solar X-rays, which happens in the corona, the Suns outer layer.

In the corona, also magnetic fields are present which provide another means to gen-

[2] An idealized body in thermodynamical equilibrium that absorbs all incoming radiation while reflecting none.

erate X-rays photons. When charged particles travel through a magnetic field, their velocity vector changes due to the Lorentz force. They will then emit synchrotron radiation, which is sometimes also called magnetic bremsstrahlung. The properties of this radiation depend on the energy of the particle, its direction of motion and the strength of the magnetic field. In astrophysical settings, the charged particles are often relativistic electrons with isotropic velocities. The energy spectrum of the electrons can be described as a power law, and the spectrum of the resulting synchrotron radiation can be described as a power law as well, with the constant A and the spectral index γ: $I(E) = AE^{\gamma}$.
Another process can lead to a power law in the observed spectrum. When the energy of electrons exceeds the energy of photons, i.e cosmic microwave background photons can gain energy from the electrons through collisions (Inverse Compton scattering). These collisions can happen repeatedly, and provide a cooling mechanism for the plasma. This also leads to a power law with an exponential cutoff in the observed spectrum and is called comptonisation.

Figure 1.3: Before the launch of a V2 rocket at White Sands missile range, New Mexico, in 1946. Image: U.S. Naval Research Laboratory

The Sun is not considered a strong X-ray source. Assuming a similar ratio of X-ray to optical photons as in the Sun, an object at galactic distances would not have been detectable during the time of these early rocket experiments. This is the reason why a search for galactic X-ray sources was not given a high priority.

1.3 The first galactic source

In 1962 the first galactic X-rays source (Figure 1.4) was detected. This was accomplished by a group at American Science and Engineering (AS&E) led by Riccardo Giacconi. The group searched for X-rays from the Moon. They expected those X-rays to be produced when the surface of the moon would get hit by the energetic solar wind[3]. This would provide information about the lunar surface, which was of interest due to America's plan to send the first human on the Moon. Although the first mission was a failure (the doors for the instrument couldn't be opened) the second mission launched from White Sands New Mexico in June 1962, was quite a success, because it did detect the first powerful galactic X-ray source (Giacconi et al., 1962).

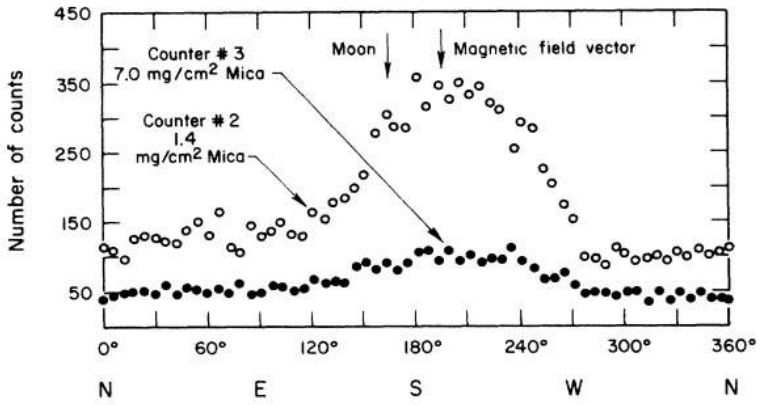


Figure 1.4: The number of counts acquired from two Geiger counters versus the azimuth angle. For 350s, in a 6° interval, the counts where accumulated. Counter 1 was not functioning, but counter 2 and 3 both show a peak at 195°. Also the count does never reach zero, implicating a diffuse X-ray background. Image: Giacconi et al. (1962)

The source, known to be located in the Scorpious constellation, became known as Sco X-1. Not only was it clear that it was a galactic source, but also that is was one that behaved in unexpected ways. If the ratio of X-rays to optical, would be be the same as in the case of the Sun, then the source should be as bright as the full Moon in the sky (in the optical). A location was given then by Sandage et al. (1966). It showed an O-star as the optical counterpart. This was interesting for the scientific community, because suddenly there was an object, which obviously generated large amounts of energy, yet was barely visible in the optical.
For a better understanding, a small excursion through the end stages of stellar evolution is needed, followed by a closer look from today's perspective at Sco X-1.
Depending on their initial mass (and their metallicity), stars differ in their evolution and their end stages. Stars with lower mass like your sun become white dwarfs, heavier ones neutron stars and the heaviest could end up as black holes.

[3]A stream of charged particles ejected from the Sun

1.3.1 White dwarfs

In 1914 H. Russel noticed a very small star (40 Eridani B.), which was located below the main sequence in the Hertzsprung-Russell diagram (Russell, 1914). He named the star, because of the small size, a white dwarf. A similar object was found in the companion of Sirius, Sirius B. Here it was possible, due to the binary nature of the system, to provide an estimation for the radius. With the mass already known, a mean density for Sirius B could be derived by Adams (1915) to be around $5 \cdot 10^4$ g/cm^3. This high density made clear that these stars were quite different from ordinary stars and formed a new class of stars.

White dwarfs mark the end stages of stellar evolution for intermediate and low-mass stars. Stars with initial masses of $10 \pm 2M_\odot$ are believed to form white dwarfs (Ritossa et al., 1996). For example a star with $8M_\odot$ will most likely form a white dwarf with $\approx 1M_\odot$, while our Sun would end up as a white dwarf with about $0.5M_\odot$ (Althaus et al., 2010).
The mass distribution of white dwarfs is narrow (Kepler et al., 2007), ranging from $\approx 0.2M_\odot$ to $1.33M_\odot$ (Figure 1.5).

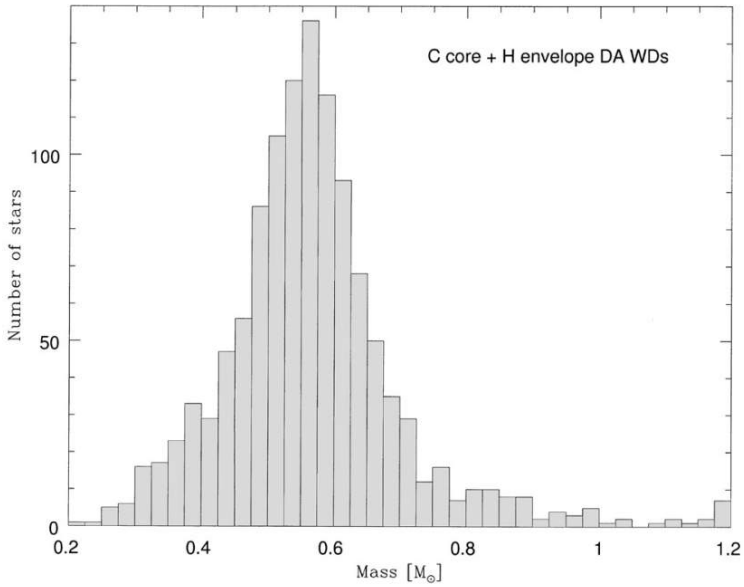


Figure 1.5: The mass distribution of 1175 white dwarfs, with $T_{eff} \geq 12000K$. This data was gathered from the Sloan Digital Sky Survey, showing only nonmagnetic DA white dwarfs (white dwarfs with a core of carbon and a hydrogen envelope). The peak mass is at $M = 0.562M_\odot$ and the individual bins have a width of $0.025M_\odot$. Image: Madej et al. (2004).

A complete degenerate electron gas is the source of pressure in a white dwarf, thus making it a stable object. By combining quantum mechanics and the concept of relativity, Chandrasekhar was able to describe the Equation of State (EOS) in a white dwarf and find an upper mass limit for these objects (Chandrasekhar, 1931). This limit of $1.4M_\odot$ for non rotating white dwarfs, is now known as the Chandrasekhar

limit.

Figure 1.6: NGC 7293, the Helix Nebula, a planetary nebula with a white dwarf in the center. This image was taken with NASA's Galaxy Evolution Explorer (GALEX) that operates in the UV. At the end stages of stellar evolution an intermediate mass star sheds most of its outer layers, leaving a hot core that forms the white dwarf. Image: NASA/JPL-Caltech

1.3.2 Neutron Stars

While testing newly wired antennas, Jocelyn Bell noticed a signal from a source in the sky. A signal, with a precise and stable period of 1.337s (Hewish et al., 1968), which came from a radio Pulsar. A fast rotating (up to ≈ 700 Hz), highly magnetized ($\approx 10^9 - 10^{12}$ G), very compact object (typical radius of $\approx 10 - 14$ km) with strong gravity, that emits electromagnetic radiation (Thorsett and Chakrabarty, 1999). This radiation can only be observed when the emission is pointed towards Earth (Figure 1.7) and shows itself as pulsations, which can be as precise as an atomic clock. What Hewish and Bell discovered was the first neutron star.
Neutron stars existed as a theoretical concept since Oppenheimer and Volkoff (Oppenheimer and Volkoff, 1939) proposed them as the end stages of stellar evolution. In late 1968 Gold concluded that a fast rotating Neutron star could be the explanation for the discovery of Bell and Hewish (Gold, 1968).

Stars with an initial mass of $\geq 10 M_\odot$ end their life with a supernova explosion, while the core collapses (Heger et al., 2003). The pressure in the core becomes so high, that electrons and protons are squeezed together, combining them to neutrons. The degeneracy pressure of the neutrons can stop the collapse, forming then a stable neutron star. The typical neutron star has a mass of $\approx 1 - 2 M_\odot$, a radius of $\approx 10 - 14$ km. This makes neutron stars the most compact stars in the universe, with densities of $\approx 10^{15}$g cm^{-3}, which is three times the normal nuclear density (Y. Potekhin, 2010).

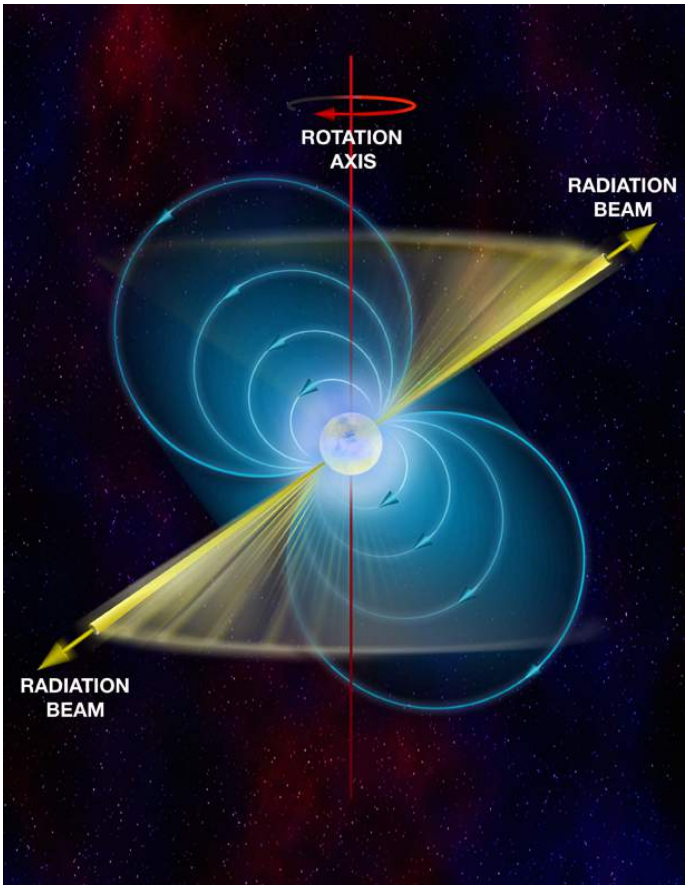


Figure 1.7: In the strong magnetosphere of the Pulsar, electrons are accelerated and they emit electromagnetic radiation, focused into a narrow beam on the magnetic poles of the Pulsar. The beam can be detected when it sweeps through the direction of the Earth not unlike a lighthouse. Image: Bill Saxton, NRAO/AUI/NSF

Neutron stars that show pulsations are often classified by the primary power source of these pulsations (Harding, 2013). Pulsars that generate their energy primarily through their rotation are called rotation-powered Pulsars. If such an isolated neutron star has a very strong magnetic field, it is referred to as a Magnetar. Neutron stars in binary systems (accretion powered) are explained in the next chapter.

The actual composition of neutron stars is complicated and also to this date unknown. Theories, some proposing strange quark matter, try to derive an EOS, making predictions of the mass and radii. These predictions then can be checked through observation and used to constrain, or rule out certain theories and their respective EOS (Lattimer and Prakash, 2007). This is one of the science goals of the LOFT mission, the Large Observer for X-ray Timing, which is the core topic of this thesis. The search for the correct EOS is not only important for astrophysics, but also impacts other fields of physics, like particle or nuclear physics. This is often the case in modern astrophysics, because the objects it deals with provide unique opportunities to study matter in states than can not be reproduced on Earth. For an object like a neutron star, this is especially the case, thus making it a perfect

Figure 1.8: An X-ray Image of Cassiopeia A by NuSTAR (Bhalerao and NuSTAR Team, 2012). Cassiopeia A is a supernova remnant, about 300 years old, the gas is still about 50 million degrees hot. The blue tones indicate high energy X-rays, the red and green colors low energy X-rays. Image: NASA/JPL-Caltech/DSS

laboratory for questions concerning dense matter.

1.3.3 Black Holes

General relativity predicts that light passing near a massive object gets bent toward that object, because a massive body curves the space-time around it. For a very high mass and density, this curvature forms a circle, preventing the light from escape. Such an object is called a black hole. Schwarzschild used general relativity to obtain the space-time around a black hole (Schwarzschild, 1916). This Schwarzschild region is given by the radius R_s (Schwarzschild-Radius), where the speed of light equals the escape velocity. $R_s = \frac{GM}{c^2}$ (Hughes, 2005). No information from within that region, also called the event horizon, can be transmitted beyond it.

Considering general relativity combined with the knowledge of how matter behaves up to nuclear densities, it can be shown that there is a limit for the mass of relativistic compact objects. This limit, known as the Oppenheimer-Volkoff limit, seems to be around $\approx 3M_\odot$ (Narayan, 2005). If the remaining core of a star, after a supernova explosion, exceeds that limit, it will collapse to a black hole. For these stellar mass black holes, there are currently about 20 promising candidates, located in binary systems (McClintock and Remillard, 2006). The observable parameters for a black hole, are its mass, its angular momentum (the conserved spin from the progenitor star) and its electric charge. Because there is no reason to believe that a black hole should have an electric charge, the significant properties are the mass and the spin. This means objects with masses higher than $\approx 3M_\odot$, located in accreting binaries

(see next chapter), make for promising black hole candidates.

Figure 1.9: The spiral galaxy NGC7742 containing a bright AGN in the center. This is an optical image by the Hubble Space Telescope. The spiral galaxy is located at a distance of 72 million light years in the Pegasus constellation. The yellow core, which has a diameter of $\approx$ 3000 light years, is suspected to contain a super massive black hole. Image: The Hubble Heritage Team (AURA/STScI/NASA)

In the center of our own galaxy lies a dark very massive object, this can be derived from the effects it has on the motion of gas and stars close to it. This object, with a mass of $3.7 \pm 1.5 \cdot 10^6 M_\odot$ (Schödel et al., 2002), is most likely a super-massive black hole and they seem to be common in the center of many galaxies (see: Figure 1.9). The black hole in our own galaxy is currently not active, but some in other galaxies are quite active (Active Galatic Nuclei (AGN)), emitting over nearly all of the electromagnetic band.

1.4 A closer look at Sco X-1

After its discovery in 1962, Sco X-1 was a hot topic for debate, because it was not clear where and how the X-rays from that source were generated. It took a few years to eventually, through the discovery of similar systems, develop theories and finally uncover the nature of Sco X-1. In 1967 a soviet astrophysicist by the name of Iosif Samuilovich Shklovsky suggested that the source of the observed X-rays could be an accreting neutron star (Shklovsky, 1967). A step in the right direction, but still missing observational evidence. Eventually, in 1971 with the launch of UHURA

and the discovery of Cen X-3 the mystery got lifted (section 1.5). Today X-ray astronomers would classify Sco X-1 as a Low-mass X-ray binary (LMXB). A short description of these classes of X-ray binary (XRB) systems follows.

A High-mass X-ray binary (HMXB) consists of a massive early type [4] star and a compact object, in the form of a neutron star or a black hole. A powerful stellar wind ejects material from the star in all directions. As the compact companion moves through this cloud, it creates a shock front and an accretion disc (Figure 1.10). X-rays are generated in these shock fronts and where matter is accreted onto the compact object. Because the flux of such X-rays is highly dependent on the properties of the stellar wind, observation of these X-rays give a deep insight in these systems. Also these classes of early type stars have high optical luminosities and are therefore easy to detect.

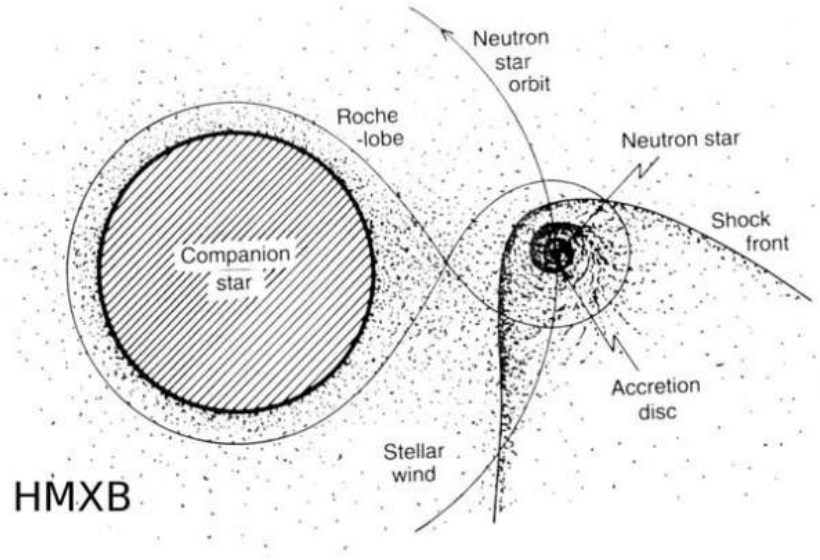


Figure 1.10: Schematic view of an HMXB. The companion star ejects matter through a strong stellar wind; as the compact object orbiting the star ploughs through this wind it forms a comet-shaped shock front. X-rays are generated within this shock front. Image: EXOSAT Observatory, ESA

If the companion of the compact object is not a massive star that emits material through a stellar wind, such a system is then referred to as a Low-mass X-ray binary (LMXB). Note that the distinguishing factor of these two is the companion star and its means to transfer mass to the compact object. In these LMXBs the star has expanded to a degree that its material on the surface is no longer gravitationally bound to it (Roche-Lobe-Overflow). Matter then flows through the inner Lagrangian point[5] shaping the star often into a drop like form, whilst building an

[4] Historical term for stars, of the O, B and A stellar type
[5] The point where the gravitational pull equals the centripetal force

accretion disc around the compact object (Figure 1.11). In the case of Sco X-1 the compact object is a neutron star, with a mass $\approx 1.4 M_{\odot}$. The donor is a faint star with a mass of $\approx 0.42 M_{\odot}$, as found by Steeghs and Casares (2002).

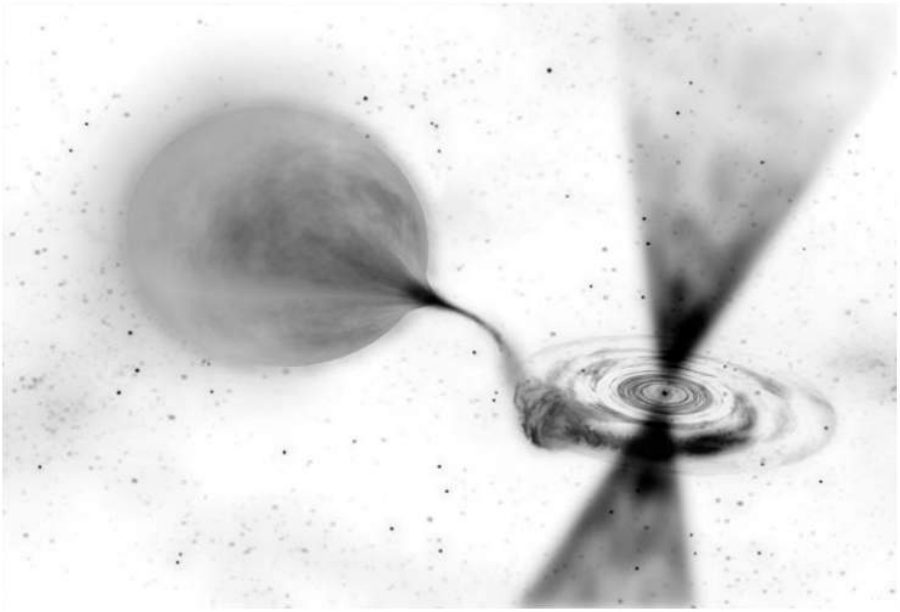


Figure 1.11: The basic geometry of LMXBs is shown in this drawing. On the left is the companion star which ejects matter to the compact object. Due to the angular momentum, the transferred mass forms an accretion disk. Some LMXBs show evidence for jets, shown here as streams of ejected matter perpendicular to the accretion disk. Image: Dana Berry (SkyWorks Digital/NASA-GSFC)

Considering that nearly 50% of the stars in our galaxy are in a binary system, one would think that X-ray binaries are a common sight. This seems not to be the case, the number of confirmed X-ray binaries is in the range of a few hundred (Liu et al., 2006), indicating that their formation and evolution is a very delicate process. To obtain the precise location and history of a particular X-ray binary can be challenging but also very interesting. In the case of Sco X-1 this has been done by Mirabel and Rodrigues (2003).

1.5 A question of Timing

Although a few more interesting sources were discovered in the following years, the big picture of the X-ray sky was missing. This changed in December 1970 with the launch of the UHURU satellite. Its mission was to perform the first survey of the X-ray sky. It proved to be successful not only in its primary task, but also in providing the important observations for specific sources. This was especially useful for a source named Cen-X3. During the years 1967 to 1968 this source was observed by one group, but not seen at all by another group. This source was then observed by UHURU in 1971 (Schreier et al., 1972). It revealed that the X-ray flux coming from that source, had a periodicity of 4.8 seconds, which indicated a pulsar. Also

slight variations of this periodicity could be identified as the Doppler shift, therefore the neutron star was in an orbiting motion. Combined with the fact, that the flux of the X-rays would disappear, each two days for roughly eleven hours (indicating an eclipsing binary), this lead to the identification of a system containing a fast rotating neutron star, orbiting a companion star which is accreting matter to the poles of the pulsar.
This was a tremendous success, because in this case, only by observing the X-rays and their timing properties, the whole system could be understood. But these early X-ray timing studies were only the beginning with many more to follow. Timing studies in particular provide the opportunity to see deep into a system, revealing the properties of the observed matter.
In the following decades, various satellites and telescopes launched, revealing even more of the X-ray sky and its sources. Also the contribution of Riccardo Giacconi was not forgotten, earning him the Nobel prize for his groundbreaking work in X-ray astronomy in 2002.
Timing studies continue to be an important tool in understanding the properties of these sources, and LOFT is one proposed mission in this frame. Which is why the LOFT mission is discussed in the next chapter.

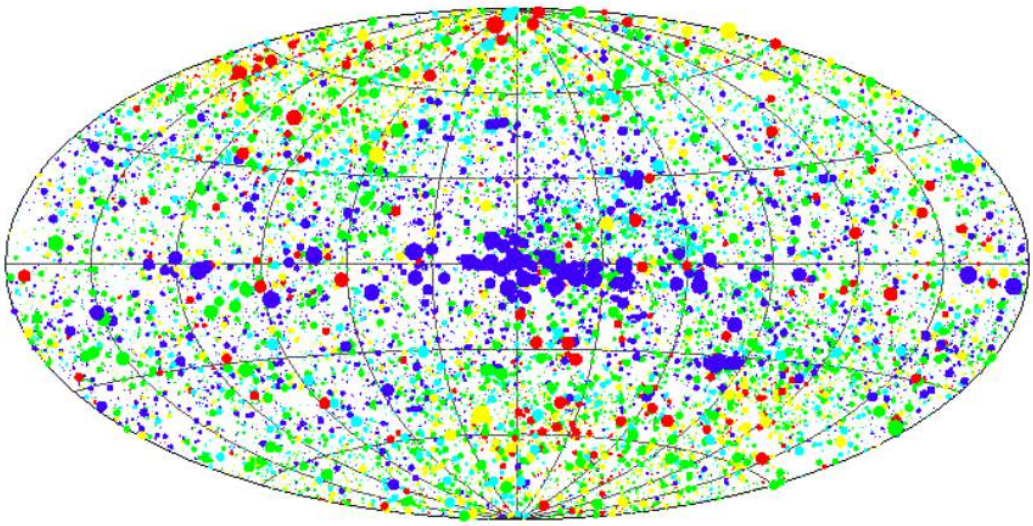


Figure 1.12: The present X-ray sky containing many hundreds of thousands of sources. This image contains 18811 sources and covers the energy range from 0.1 to 2.4 keV. The size of the points scales with the logarithm of the count rate, while the colors represent the hardness. More information on ROSAT and the image above can be found in Voges et al. (1999)

Chapter 2

The LOFT Mission

As discussed in the last chapter, variability in X-ray sources is not only quite common, but does also provide unique opportunities to explore them. In its core, the LOFT mission is designed as an X-ray timing observatory.

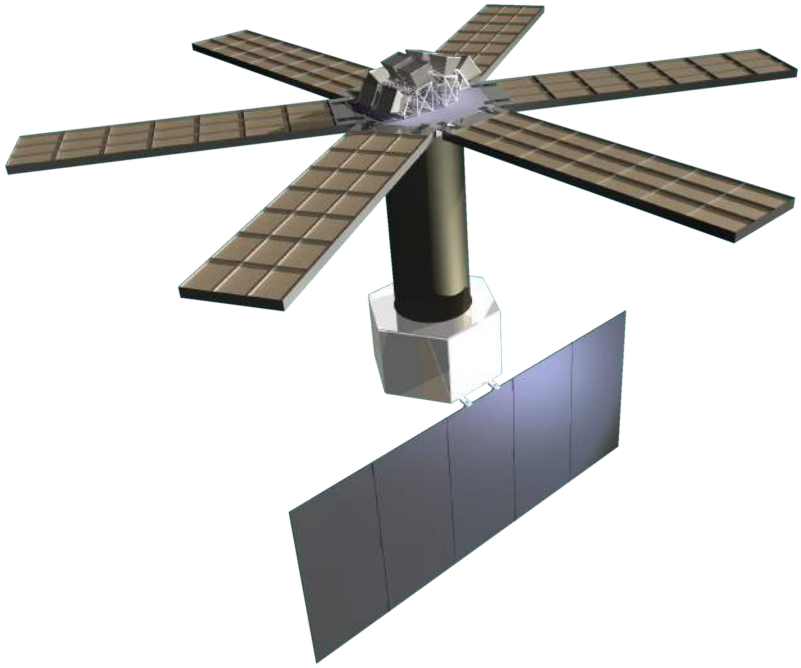


Figure 2.1: A LOFT model, showing the six panels of the Large Area Detector (LAD) and the Wide Field Monitor (WFM) with its cameras on top. Image: LOFT collaboration

2.1 Overview

For something as complex as space missions, obviously long term planning is required. It is not uncommon for a mission to take 20 years from proposal to launch. The European Space Agency (ESA) calls its long term planning, in the period between 2015 - 2025, the Cosmic Vision Program. The program gives a general

direction which topics the future missions should address. In the period from 2015 to 2025, ESA will launch several missions, and for each launch there are several competitors. The process of selecting the final mission for a launch is a long one, including science and industry studies. LOFT was such a competitor for the third call for medium mission (M3).

LOFT was competing with four other missions until the final selection stage. Although well received by the scientific community, and considered an overall robust design, it was unfortunately not selected. ESA decided to adopt the PLATO[1] mission which is committed to the search of potential terrestrial exoplanets, orbiting solar type stars. LOFT will now compete for the M4 launch slot, scheduled in 2025.

The following sections will discuss the two key science questions of LOFT and, then explain the techniques used to achieve them. They will introduce LOFT's modular structure and the essential technology. The LOFT yellow book (Barret et al., 2013), developed by the LOFT consortium for ESA as an Assessment Study Report, contains a summary of the available information on the LOFT mission. If not stated otherwise information, technical notes and descriptions in this chapter are taken from that document.

2.2 Science goals

Under the general topic of matter under extreme conditions, LOFT addresses two specific questions. They are namely:

1. What is the equation of state in neutron stars?
2. Does matter orbiting close to the event horizon of a black hole follow the predictions of general relativity?

The core structure of a neutron star begins with the outer crust, containing ions and low density electrons. In the inner crust, and the outer layer, the ions gradually become more neutron rich. In the center the density is higher than in an atomic nuclei, and the actual composition is unknown. The question for this composition, or more precise the EOS is one unsolved problem of modern physics and of interest for nuclear physics, astrophysics and quantum chromodynamics.

As part of the effort to better understand gravitation, LOFT will measure its direct effects on the motion of matter in the strong field gravitational regime. This is done best by directly observing as close to the black hole as possible, within a few Schwarzschild radii. LOFT will measure the gravitational effects near neutron stars and black holes in accreting X-ray binaries. There, the X-ray emitting plasma in the accretion disk behaves like a test fluid, enabling to probe the strong curved space-time near the compact object.
To obtain the spins of stellar mass black holes is also of importance, because the spin

[1]Planetary Transits and Oscillations of stars

carries information about the formation of the black hole. The extraction of angular momentum, or accretion, is not enough to significantly influence the mass of a black hole or its spin in a binary system. Therefore, potential measurements of mass and spin of black holes in X-ray binaries, reflect the initial mass and spin at the black hole formation (King and Kolb, 1999). LOFT will also focus on super-massive black holes. The mechanism of accretion and the growth of these giants is of particular interest, due to their influence on galaxy evolution and even feedback effects they have on their surroundings (Volonteri and Bellovary, 2012).

2.2.1 EOS in neutron stars

The maximum mass a neutron star can have depends on how the EOS behaves at densities higher than the nuclear densities, while the maximum radius depends on the behavior of the EOS at lower densities. Therefore, by obtaining neutron star masses and radii, the EOS can be reconstructed. Also, depending on a specific EOS and the matter it proposes, a limit for the maximum mass or the maximum radius can be derived. This means, that by measuring high masses and radii, certain EOS can already be ruled out (see: Figure 2.2). Assuming the knowledge of a neutron stars radius, with a precision of 1 km or better, the neutron stars pressure can be determined in the range of $1 - 2\ n_S$ [2] (Lattimer, 2001).
If three neutron stars would be observed with a 5% uncertainty, it would be possible to discriminate between proposed EOS with a 3σ confidence level (Özel and Psaltis, 2009).
LOFT will use three different techniques in order to constrain the neutron star masses and radii. These techniques evolve around different systems of neutron stars (Magnetars, Bursters) and enable constraints which are independent from each other. These techniques will be discussed in the following subsections.

[2]The nuclear saturation density : $2,74 \cdot 10^{14}$ g cm^{-3}

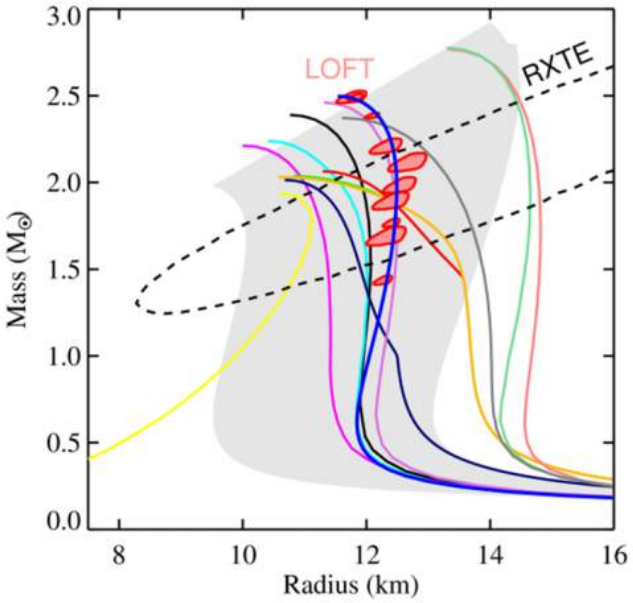


Figure 2.2: The mass-radius relation of neutron stars, shown for various EOS and the possible constraints that LOFT can contribute. Shown as colored lines are different EOS models. The red error regimes show possible LOFT mass-radii measurements, which would be able to discriminate between EOS models. Image: Anna Watts / LOFT Assessment Study Report

2.2.1.1 Pulse Profile Modeling

As photons travel through the curved space-time of a pulsating neutron star (Figure 2.3), information about the mass and the radius of the star is stored in its pulse profile.
It is possible to obtain the information about mass and radius by modeling the shape of these pulse profiles. How to apply this technique is showed in detail for the LOFT case by Lo et al. (2013).

The pulsations can have different origins. A hot spot can occur when material falls on the surface of the neutron star along the magnetic field lines. Thermonuclear explosions can occur, from unstable burning of hydrogen or helium, leading to burst oscillations. Especially these burst oscillations are well understood (Suleimanov et al., 2012) and LOFT will focus on 24 already known X-ray Bursters and model their pulse profiles.

2.2.1.2 Spin measurements

The fastest spin a neutron star can sustain, is limited by the frequency where mass-shredding occurs, thus the neutron star can not be stable any more. This maximum frequency can be described by an empirical formula, as given by Lattimer and Prakash (2004), where M is the mass, R the radius, and C a constant.

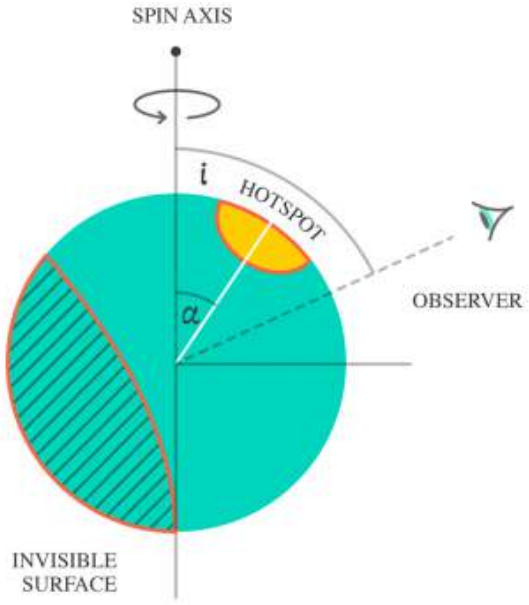


Figure 2.3: Emission from a hot spot on the surface of a neutron star generates a pulse profile. This hot spot is not aligned with the rotational pole, and will therefore create modulations, in the flux that can be observed because of the neutron stars rotation. Due to relativistic light bending, light from behind the neutron star will be bent around it, leading to part of the back of the neutron star being visible. This effect depends on the compactness of the object (M/R). Image: Anna Watts / LOFT Assessment Study Report

$$f_{max}(M) \approx C(\frac{M}{M_\odot})^{\frac{1}{2}}(\frac{R}{10km})^{-\frac{3}{2}}$$

This constant C does depend on the EOS. For most EOS proposing hadronic[3] matter C is $\approx$ 1.08 kHz. For more exotic EOS it can get up to C $\approx$ 1.15 kHz. (Haensel et al., 2009)

The fastest spin discovered so far is at 0.716 kHz by Hessels et al. (2006), which does not rule out any EOS (so far), but spins higher than 1 kHz would put constraints on the EOS.
Understanding these high spins, would also provide further insights in the evolution of XRB's. To explain the formation of fast millisecond radio pulsars, one proposed explanation is that they spin up due to accretion on the pulsar, in Low Mass X-ray binaries. This is currently a topic of active research and debate.

2.2.1.3 Astroseismology

Highly magnetized neutron stars, known as Magnetars, can produce so called Giant Flares. It is believed that these flares originate from instabilities due to fracturing of

[3]Hadrons are a family of particles, consisting of quarks, e.g. protons and neutrons

the Magnetar's crust. Quasi-period oscillations[4] (QPOs) were discovered in such a giant flare from a Magnetar, by Strohmayer and Watts (2005). These quasi-periodic variations in the X-ray flux, typically in the range of milliseconds, are believed to be some sort of seismic vibration, can constrain the magnetic field strength and ultimately, with the use of seismic oscillation models, the EOS. More information on the topic of QPOs in X-ray binaries can be found in van der Klis (2000).

Unfortunately, these giant flares are a rare occurrence (approx. once every 10 years). With its sensitivity LOFT could use so called Intermediate Flares, which are unpredictable but occur more often. Simulations from the LOFT team have shown, assuming similar frequencies and amplitudes as in giant flares, LOFT would be able to detect QPOs in intermediate flares. Expressions for the frequency of the fundamental oscillation can then be derived, by using relativistic oscillation models. From their the mass-radius relation can be calculated and used to constrain the EOS, as shown by Samuelsson and Andersson (2007).

2.3 Strong field gravity

As discussed in the previous chapter, the mass of a black hole and its spin is of particular interest in understanding not only black hole properties but also their evolution (Volonteri and Bellovary, 2012). Beyond that, verifying general relativity in the strong-field regime, as well as discriminating between competing models, that focus on the dynamics of matter near a black hole, is all part of LOFT's strong-field gravity topic.
To achieve this, LOFT uses two basic techniques, targeted at stellar mass black holes in XRBs and super massive black holes, which are briefly shown in the next two subchapters. These techniques make use of the relativistically broadened Fe-Kα line, as well as time variability on millisecond scales, in particular quasi-periodic oscillations.

2.3.1 Relativistic iron line

The fluorescent iron line (6, 4 - 6, 9 keV in the X-ray band) is a very strong line in the X-ray spectrum of AGNs and Seyfert galaxies and is an excellent tool in studying the geometry and the accretion flow near black holes.
Gas orbiting close to the event horizon of a black hole, gets accelerated to relativistic velocities. Here, also effects from general relativity influence the behavior of the accretion disk. Transverse Doppler shift, relativistic beaming and gravitational redshift result in a broadening of the observed Fe line (see: Figure 2.4).

With LOFTs spectral resolution of ≈200 eV, this relativistic iron line could be observed and resolved for galactic black holes. From there the basic properties of the black hole could be derived. With the use of black hole and accretion models, overlapped with the accretion disk, the observed iron line could be fittet and the

[4]Oscillations detectable in the lightcurve, typically in the range of 100 Hz - 1 kHz.

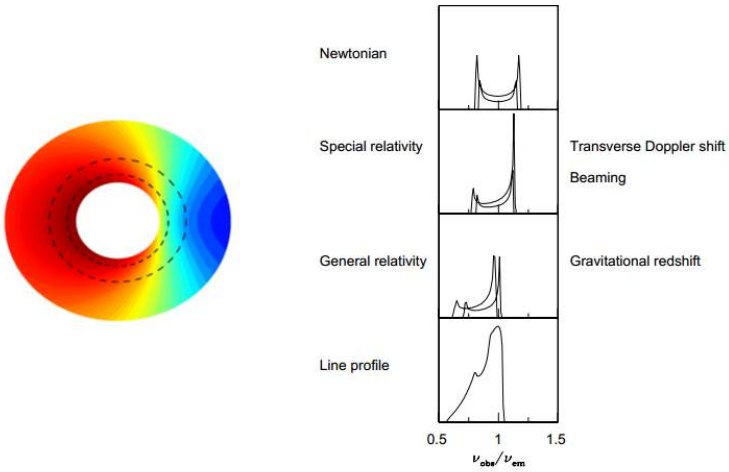


Figure 2.4: Each radius, in a non-relativistic disk, shows a symmetric line profile. The two peaks correspond to emissions from the approaching, blue shifted side, and the receding, red shifted side. This leads to the first line profile (Newtonian). In a disk with relativistic velocities, various relativistic effects broaden the line and lead to the last line profile. Turbulence in the disk could also lead to a broadened line, if the disc is thick enough. This is currently a field of study, where magneto-hydrodynamic disk simulations play a key role. Image based on: Fabian et al. (2000)

parameters for mass and spin obtained.
An example of how this technique is applied can be found in Tanaka et al. (1995).

2.3.2 QPOs

Quasi-period oscillations are associated with the motion of plasma, orbiting very close to a black hole. For these QPOs there are models, based on global oscillations within the disk, that suggest different orbiting motions for this plasma. They link the frequencies of these QPOs to radial, vertical and azimuthal relativistic orbits, near the event horizon. In the Relativistic Precession Model (RPM) (Stella et al., 1999), the observed frequencies vary and are correlated with the variable inner disk radius. While in the Epicyclic Resonance Model (Abramowicz and Kluźniak, 2001) the frequency is believed to be constant. LOFT would be able to discriminate between these models, by measuring the epicylic motion in high frequency QPOs from black holes in X-ray binaries, which up to now was not possible.

2.4 Observatory Science

With its wide field of view (WFM: 3.2 steradian) and its good spectral resolution (LAD: $\approx$ 200 eV), LOFT is well equipped to study various objects and phenomena. The LAD enables timing studies of a variety of astrophysical processes, such as accretion or gravitational collapse. The WFM, will catch bursts and other unexpected

events, by monitoring hundreds of known sources. In Table 2.1 the observatory science themes, the associated objects and the instruments monitoring them are shown.

Table 2.1: The observatory science themes and associated objects.

	Accretion physics	Strong B-field	Cosmic explosions
Accreting white dwarfs	LAD, WFM	LAD	LAD, WFM
Low Mass X-ray binaries	LAD, WFM	LAD, WFM	
LMXB (X-ray Bursters)	LAD, WFM		LAD, WFM
High Mass X-ray binaries	LAD, WFM	LAD, WFM	
Isolated neutron stars	LAD	LAD	
Magnetars		LAD, WFM	LAD, WFM
Stars		LAD, WFM	
Nearby Galaxies	LAD, WFM	LAD, WFM	LAD
Tidal disruption events	LAD, WFM		
Bright AGN (Seyferts, Blazers)	LAD, WFM		
Gamma-ray burts	LAD, WFM		WFM

2.5 Main Instruments

The LAD and the WFM are described in detail below. Because they use the same detectors, these detectors and their working mechanism are discussed after the instruments.

2.5.1 Large Area Detector

The LAD is a 10 m^2 class instrument, with good spectral resolution ($\approx$ 200 eV). The effective area of the LAD compared to previous missions can be seen in Figure 2.5. The requirements as well as the expected performance are summarized in Table 2.2

Table 2.2: Requirements and anticipated performance of the Large Area Detector

Item	Requirement	Anticipated performance
Effective area	3.8 m^2 @ 2 keV 7.6 m^2 @ 5 keV 9.5 m^2 @ 8 keV 0.95 m^2 @ 30 keV	4.4 m^2 9.0 m^2 9.8 m^2 1.3 m^2
Energy resolution	240 eV @ 6 keV	180 eV @ 6 keV
Absolute time accuracy	2 μs	1 μs
Dead time	$< 1\%$ @ 1 Crab	$< 0.7\%$ @ 1 Crab
Max Flux	$>$ 500 mCrab	650 mCrab

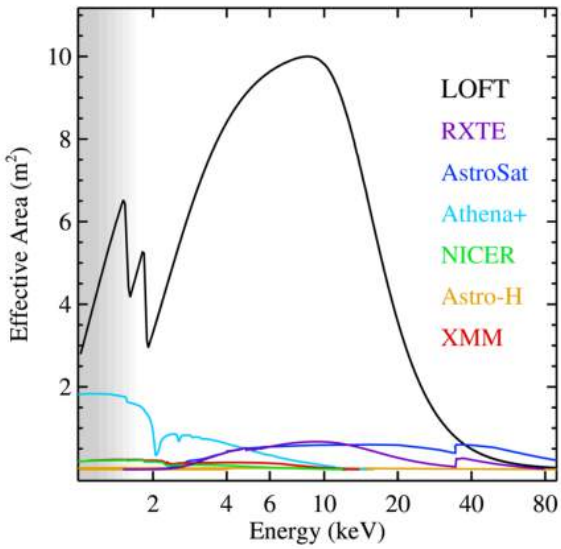


Figure 2.5: A comparison of the effective area, as a function of energy. The LOFT mission (in black), compared to previous missions. Image: Marco Feroci / LOFT Assessment Study Report

2.5.1.1 LAD hierarchical design

The LAD follows a modular design, with 4 x 4 detectors and 4 x 4 collimators per Module (including their electronics). These Modules are mounted on deployable Panels (Figure 2.6).
A detailed description of the LAD electronics design follows in section 3.1.

2.5.1.2 LAD Module

The structure of a LAD Module, from top to bottom (see: Figure 2.7):

- The Collimator tray, containing 16 Micro Cappilary Plates (MCP), one for each Silicon Drift Detector (SDD). In order to reflect sunlight and to keep the temperature constant at below $-10°$, an aluminum film is placed on top of each MCP.
- A thin Kapton film is placed between the collimators and the detector tray, to filter out UV-light.
- 16 SDDs and the Front-End Electronics (FEE) form a LAD Module. Each SDD is read-out by 224 anodes, and each FEE holds 14 ASICs, which handle the A/D conversion (see: subsection 2.5.5).
- The ASICs and the detectors are controlled and read out by the MBEE, the Module Back-End Electronics, the electronics mounted on the back of each Module, which is the main part of this thesis (see: Figure 2.8).

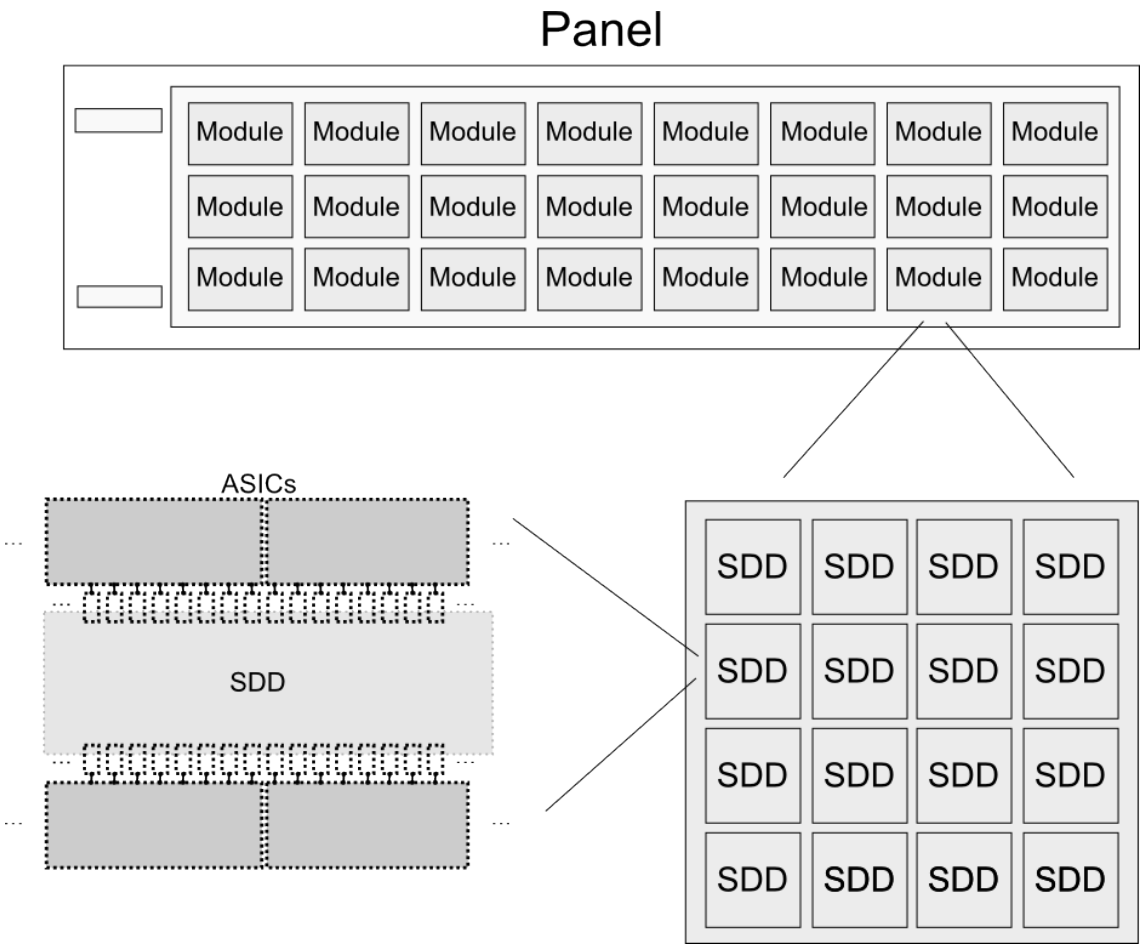


Figure 2.6: The hierarchical design of the Large Area Detector (LAD). On the bottom left: The Silicon Drift Detectors (SDDs), are read out by ASICs. On the bottom right: 16 SDDs, their Electronics and the Collimator are placed on one Module. On the top: The Modules are placed on Panels that can be deployed once in space. The exact number of Modules and Panels depends on the Spacecraft design, as it is discussed in subsection 2.5.2.

2.5.1.3 LAD Panel

The Panels of the LAD are stowed in the launcher and are deployed once LOFT is in orbit. The number of Panels and the grid, on which the Modules are mounted depends on the spacecraft design, as it can be seen in the next chapter. A Panel Back-End Electronics (PBEE) is the responsible Electronic on the Panel and handles the Module Back-End Electronics.

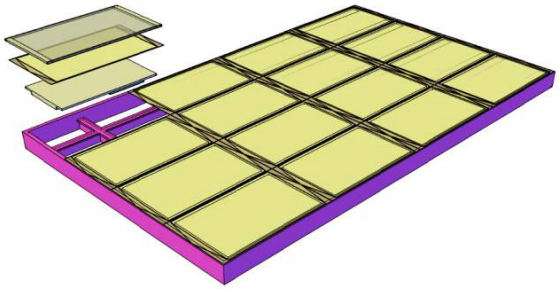


Figure 2.7: The front-side view of a Module. It shows the placement of the collimator, the Silicon Drift Detectors (SDDs) and the Front-End Electronics (FEE). Image: LOFT Assessment Study Report

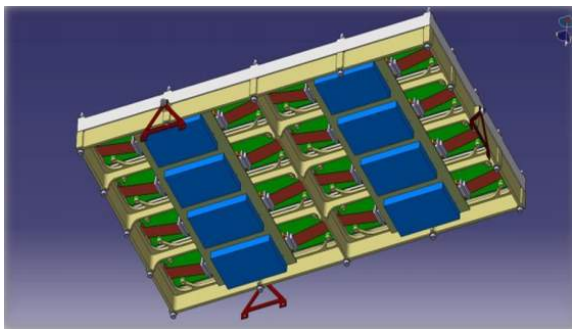


Figure 2.8: The back-side view of a LAD Module, showing the Module Back-End Electronics. Image: LOFT Assessment Study Report

2.5.2 Possible Spacecraft Design

During the assessment phase of LOFT two possible mission designs were derived through industrial contracts. These two spacecraft designs have a different accommodation concept, but their are both consistent with the overall requirements for LOFT.

- A. 5 Panels, containing 25 Modules. 125 Modules in total (see: Figure 2.10)
- B. 2 Panels, containing 62 Modules. 124 modules in total (see: Figure 2.11)

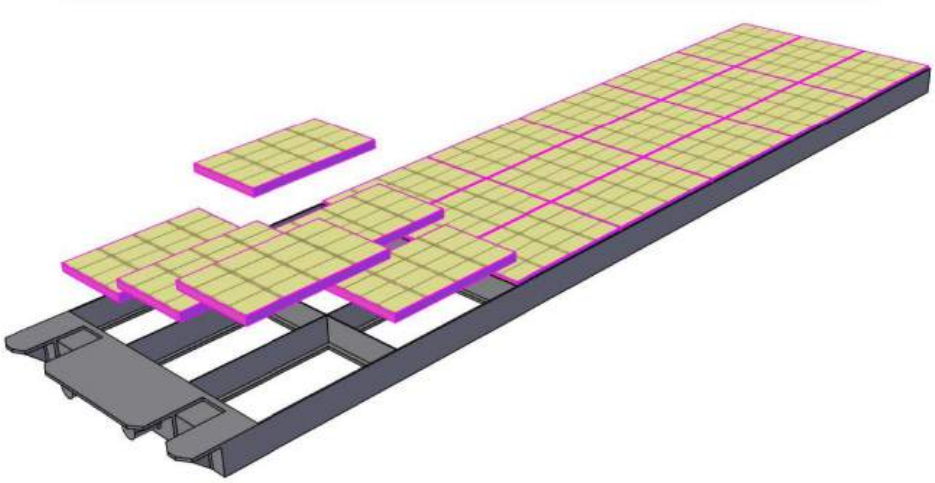


Figure 2.9: A assembled LOFT/LAD panel showing the placed Modules and the connection to the deployment system. Image: LOFT Assessment Study Report

Figure 2.10: View of the LOFT Spacecraft Design concept A. The left shows the deployed configuration while the right shows the design stowed in the launcher. In this Spacecraft design there are five Panels, each containing 25 Modules. Image: LOFT Assessment Study Report

2.5.3 Wide Field Monitor

The Wide Field Monitor (WFM) contains five pairs of coded mask cameras, that are shown in Figure 2.13. These cameras have an effective field of view (per camera pair) of 70° x 70°. The detectors allow only for position reconstruction in one dimension; by placing two pairs of cameras orthogonal to each other, it is possible to obtain the source location in two dimensions. The requirements for the WFM, as well as the expected performance, are summarized in Table 2.3.

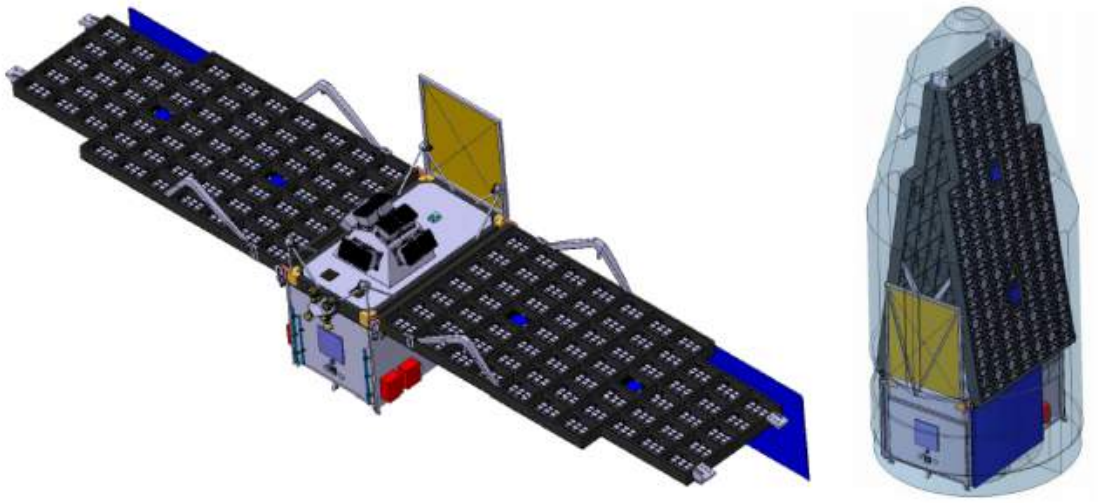


Figure 2.11: View of the LOFT Spacecraft design concept B. On the left: The deployed configuration with two Panels, each containing 62 Modules. On the right: The configuration stowed in the launcher. Image: LOFT Assessment Study Report

Table 2.3: Requirements and anticipated performance of the Wide Field Monitor.

Item	Requirement	Anticipated performance
Energy resolution	500 eV @ 6 keV	$<$ 300 eV @ 6 keV
Location accuracy (2D)	$<$ 1 arcmin	$<$ 1 arcmin
Angular resolution (2D)	$<$ 5 arcmin	$<$ 4.3 arcmin
Absolute time accuracy	2 μs	1 μs
Field of view	3.2 steradian around the lad	5.5 steradian at zero response

A coded mask camera for the WFM from top to bottom (Figure 2.13):

- The coded mask is 0.15 mm thick, made of Tungsten and its open area is 25%. It is important that this mask is stable and flat, in order to obtain high quality images.
- A collimator supports the mask (3 mm thick) and also shields it in order to reduce background. 4 SDDs and ASICs are placed on one detector tray. This is similar to the LAD design, but differs in smaller anode pitch, more channels per ASIC and the size of the SDDs.
- A 25 μm thick Beryllium filter shields the SDDs against micrometeorite impacts. This is necessary due to the WFM's large field of view.
- The WFM uses a Back-End Electronics (BEE), similar to the MBEE in the LAD, although with more computing capability in order to determine photon positions.

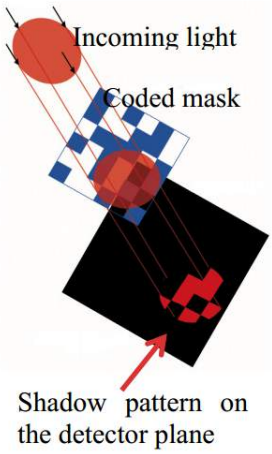


Figure 2.12: The working principle of a coded mask instrument. Photons from a certain direction in the sky project the mask pattern on a detector that is position sensitive. Opposed to focusing optics, where a source is concentrate on a small detector area, in a coded mask instrument the signal is distributed over the detector, and the source can illuminate the detector through the mask. This creates a shadow pattern of the mask on the detector, which is used to mathematically reconstruct the position of the source (Skinner, 1995). Image: B.J. Mattson, L3/NASA/GSFC

2.5.4 Silicon Drift Detectors

The Silicon Drift Detectors (SDDs) are based on the ones used by the ALICE[5] experiment. They operate on the principle that an incoming X-ray photon creates a cloud of electrons, when it enters the detector. Due to the applied voltages, this charge cloud drifts to the readout anodes, widening on the way due to diffusion. The working principle is shown in Figure 2.14. The time for this drift is about 7 μs and the maximum diameter of the charged cloud is around 1 mm, this of course depends on the energy and the location of the original X-ray photon. An interaction near the anodes will have a smaller charge cloud than one happening further away. The possibility for the charge cloud to be collected by multiple anodes, can of curse be altered by the anode pitch. In the case that the charge cloud is collected by more than one anode, the signal to noise gets worse, because the read-out noise of multiple anodes now is added. On the contrary, this enables a two dimensional position reconstruction.
As the LAD is a non imaging instrument, position reconstruction is not needed. Therefore, to optimize the signal to noise ratio of the LAD, an anode pitch of 970 μm was chosen. This results in $\approx$ 40% of the events in the LAD are read-out by one anode (enabling an energy resolution of $\approx$ 200 eV), while $\approx$ 60% will be collected by two anodes (energy resolution of $\approx$ 200 eV). The single and double anode events combined, will meet the overall required energy resolution for the LAD of at least 240 eV.

[5] A Large Ion Collider Experiment: One of seven collider experiments from the Large Hadron Collider (LHC) near Genf, dedicated to study collisions between heavy Ions.

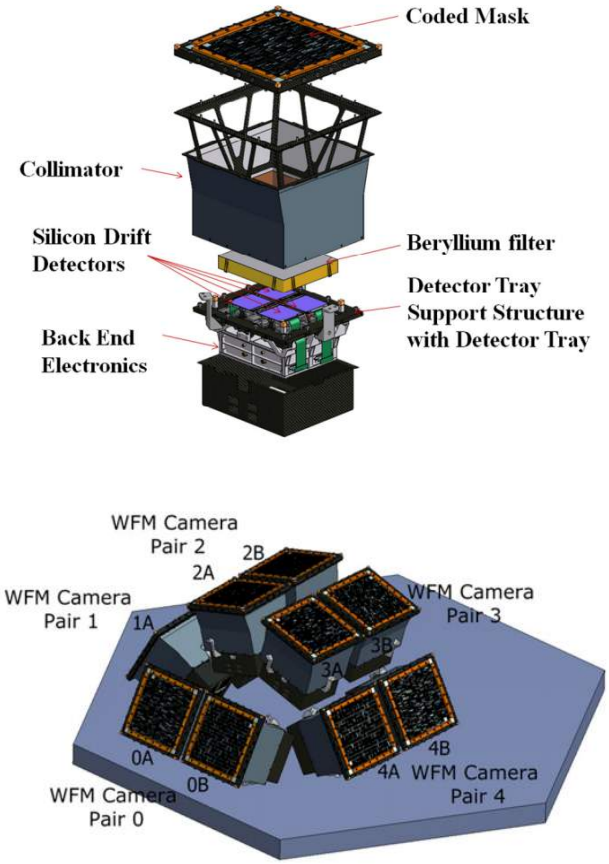


Figure 2.13: Top Image: A coded mask camera. Bottom Image: The five pairs of coded mask cameras of the Wide Field Monitor. Image: Dmitry Karelin / LOFT Assessment Study Report

In the Wide Field Monitor, at the expense of higher read-out noise, a smaller anode pitch of 145 μm was selected in order to achieve a higher accuracy in positional reconstruction.
An SDD prototype can be seen in Figure 2.15 and a comparison of the main parameters of the SDDs in the LAD and the WFM is shown in Table 2.4.

2.5.5 ASICs

The task to read out the anodes from the SDDs is performed by the low-noise ASICs. An ASIC is a chip or a set of electronic circuits, designed for a particular purpose. The LOFT ASICs design is based on the ESA StarX32 ASIC project. These mixed-signal ASICs already provide a low-noise (18-19 e-rms), while having a low-power consumption (500 μW per channel). In May 2012 the development for a LOFT ASIC prototype was started by the Dolphin Integration[6]. This first prototype, called Sirius V1, features 8 channels, one half of them directed at the LAD (Anode pitch of 970 μm), and the other half directed at the WFM (Anode pitch of 145 μm).

[6]www.dolphin.fr

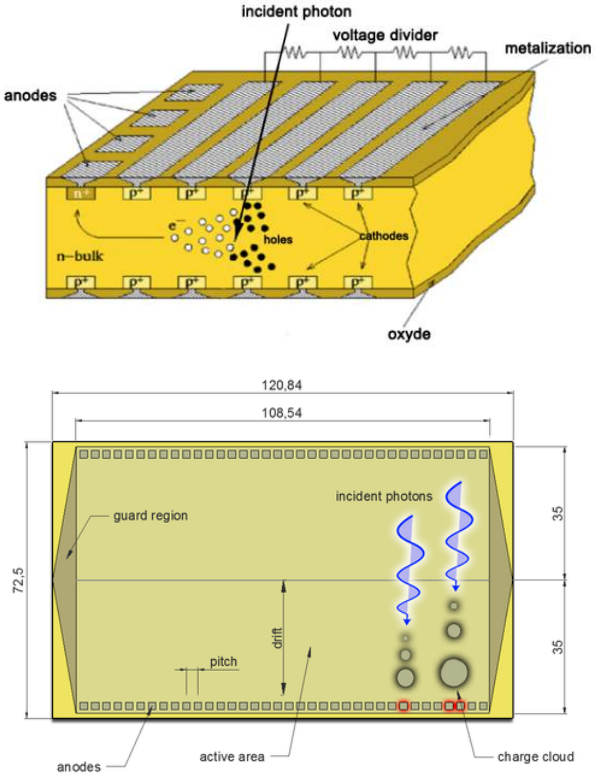


Figure 2.14: The working principle of the Silicon Drift Detector, shown from the side and the top. An incoming photon creates a cloud of electrons, due to the applied voltages they drift to the readout anodes. Image: Rashevsky et al. (2002)

Each individual channel in the Sirius V1 already includes the needed analog features:

- A charge pre-ampifier
- A shaper amplifier with a selectable shaping time between 1 to 4 μs.
- The peak detection and hold

The development of a second prototype, Sirius V2, started in October 2013. This ASIC features 16 channels and includes an built-in A/D converter.

Table 2.5: The key specification of the LOFT ASICs shown for the LAD and the WFM.

Parameter	LAD	WFM
Max power consumption	0.65 mW/channel	0.72 mW/channel
Detector pitch	970 μm	145 μm
Assumed SDD leakage current (EoL)	7 pA	3 pA
Assumed input capacitance	350 fF	80 fF
Noise (rms, Eol) at 7 pA	17 e^-	13 e^-
Number of channels	16	64

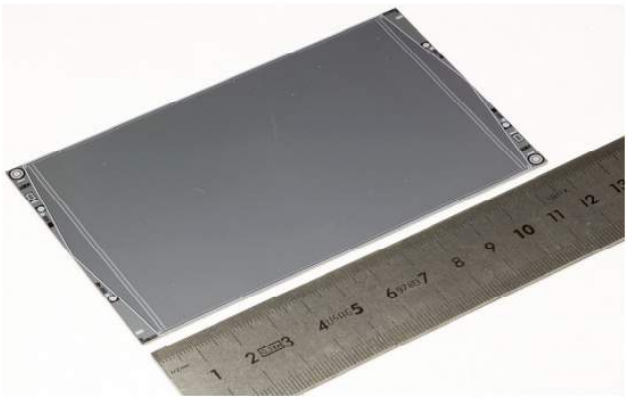


Figure 2.15: A Prototype of the SDD used by the LOFT/LAD. The triangular structures on both sides identify the built-in voltage divider responsible for the drift of the charge cloud. Image: LOFT Assessment Study Report

Table 2.4: The main parameters of the SDDs in the LAD and the WFM.

Parameter	LAD	WFM
Thickness	450 μm	450 μm
Active Area	108.5 mm x 70.0 mm	65.1 mm x 70.0 mm
Anode Pitch	970 μm	145 μm
Drift length	35 mm	35 mm
Drift field	360 V/cm	360 V/cm
Number of anodes	2 x 112	2 x 448
Anode capacitance	350 fF	85 fF

2.5.6 Collimator

The collimator restricts the field of view of the LAD, to prevent source confusion and to shield the detectors. In the current requirement of the collimator size (111.0 mm x 72.5 mm), a single collimator covers the active area of a single SDD. The LOFT collimators are Lead-glass capillary plate collimators and stop X-ray photons up to $\approx$ 30 keV, coming from outside the field of view. Above $\approx$ 40 keV the collimator becomes increasingly transparent, thus resulting in a higher background for this energy. As no timing variations are expected in this background, dealing with it should prove easy. The collimator will have an open fraction of 70 percent and a collimator mass of approximately five kilogram per square meter. The collimators (and a prototype) are manufactured by the Photonis [7] company, which already has experience in similar missions. A prototype of these Lead-glass collimators has been tested at the University of Leicester. A comparison between this prototype developed for the LOFT consortium and the LAD requirements can be seen in Table 2.6.

[7]Photonis (Brive, France, http://www.photonis.com/en/ism/21-microchannel-plate.html)

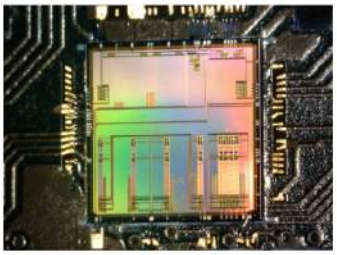


Figure 2.16: The ASIC prototype Sirius ASIC V1. The performance of this prototype is already very close the LOFT requirements, in regards of the required power consumption and the low read-out noise. Image: IRAP / Dolphin Integration

Table 2.6: The specifications of the LOFT collimator compared to a prototype developed for the LOFT consortium.

Parameter	LOFT Requirement	LOFT Prototype
Unit size	111.0 mm x 72.5 mm	50 mm x 50 mm
Thickness	5 mm	6 mm
Aspect ratio	60 : 1	60 : 1
Pore size	83 μm, squared	100 μm, squared
Wall thickness	16 μm	15 μm
Open Area Ratio	70%	60%
Number of Units	2016	2

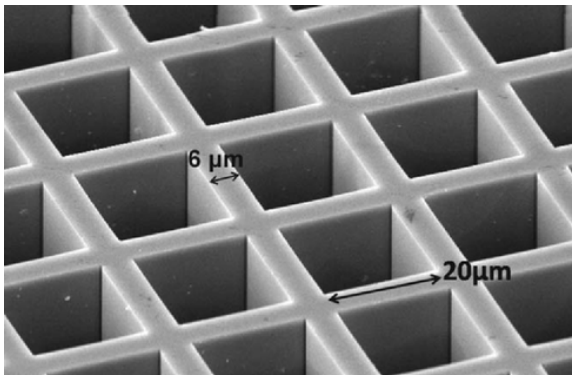


Figure 2.17: An electron microscope image of a MCP collimator for the BepiColombo mission, developed by Photonis. This collimator has a squared pore size of 20 μm and a wall thickness of 6 μm and was part of the mercury imaging X-ray spectrometer (MIXS). Image: Fraser et al. (2010)

Chapter 3

The Module Back-End Electronics for the Large Area Detector

In this chapter the requirements and tasks of the Module Back-End Electronics (MBEE) are shown. The electronics design of the LAD with its components, the proposed operational modes, data processing and the connectors of the MBEE will be explained. Additionally, the MBEE prototype, as developed at IAAT (Uter, 2013) will be presented.

3.1 LAD electronics design

In the electronics design of the LAD, each detector has its own Front-End Electronics (FEE), each Module has its own Module Back-End Electronics (MBEE) and each Panel has its own Panel Back-End Electronics (PBEE). The FEE interfaces the detectors and passes their values to the MBEE. The MBEE processes the data and transmits it to the PBEE, which collects data from all MBEEs on one Panel. The data collected by all PBEEs is passed on to the Detector Handling Unit (DHU), which is part of the Instrument Control Unit (ICU) (see: Figure 3.1). The DHU collects and buffers data from all PBEEs and then passes it to the On-Board Data Handling (OBDH) which is tasked with ground communication. A detailed description of these individual electronic parts, starting with the ICU, is given in the following sections.

The FEE and the MBEE, as well as the PBEE and the MBEE, communicate with a custom interface, while the PBEE transmits its data via a SpaceWire[1] interface to the DHU. More on these interfaces of the MBEE can be found in subsection 3.2.5.

Power will be distributed via 3 different voltages from a dedicated Power Distribution Unit, to the PBEEs and the MBEEs. Lower voltages will be generated at the level on which they are needed (e.g. PBEE, MBEE) by local Power Supply

[1] A communication protocol designed by ESA, which allows transmission rates from 2 -400 Mbits/s and uses a 9-pin Sub-D connector. More information on the SpaceWire interface can be found on "http://spacewire.esa.int".

Units on the PCB boards, as it is shown in Figure 3.2. This minimizes the amount of harness needed over the panel hinges. Also, the high voltage needed for the Silicon Drift Detectors is provided by the power supply of the MBEE (see: subsection 3.2.2)

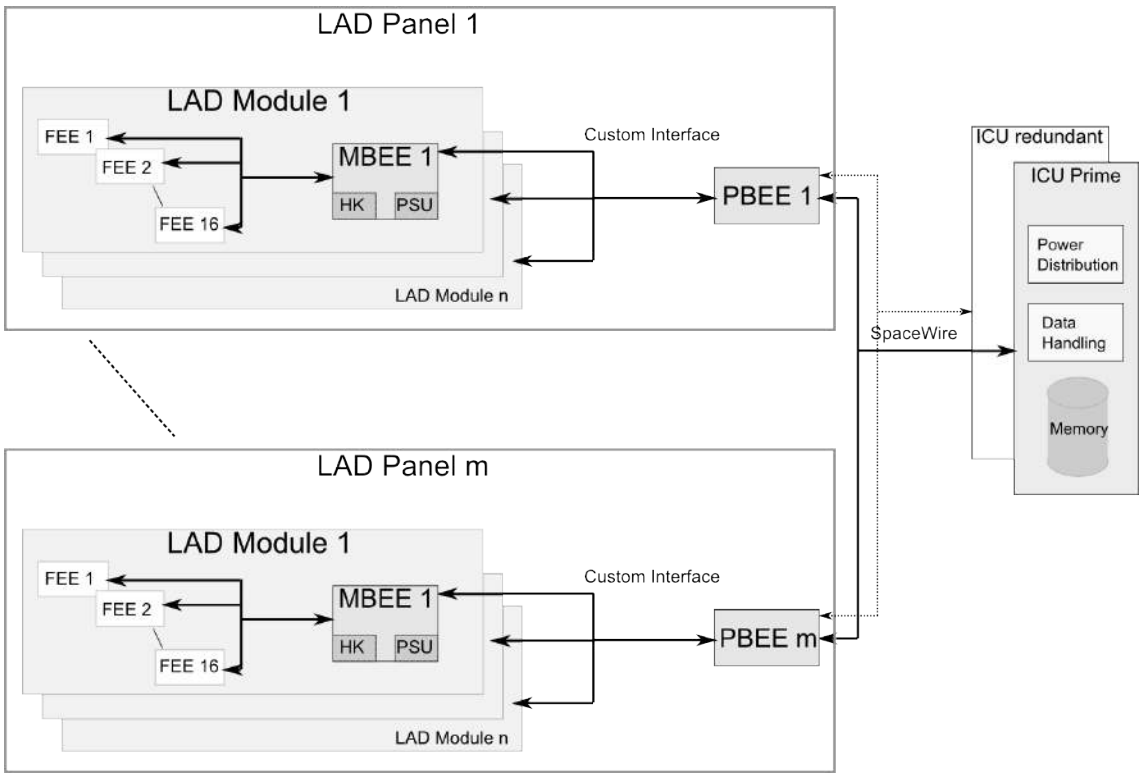


Figure 3.1: A schematic of the LAD electronics design. One Module contains the ASICs which form the Front-End Electronics (FEE) and the Module Back-End Electronics (MBEE). Each Module has its own MBEE, and all MBEEs from one Panel are connected to one Panel Back-End Electronics (PBEE). All PBEEs send their data to one Data Handling Unit.

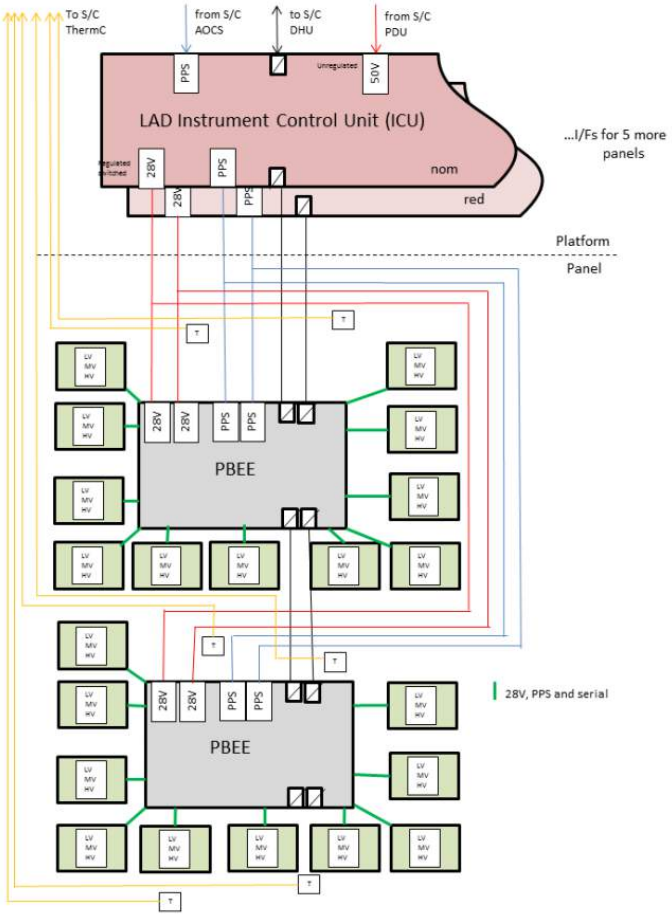


Figure 3.2: A schematic showing the electrical architecture of the LAD electronics design. The power, data and clock lines between the MBEE and the PBEE are shown in green. The schematic corresponds to one Panel. Image: Dave Walton, MSSL/UCL

3.1.1 Instrument Control Unit

The Instrument Control Unit (ICU), is the main control electronics and is the top of the LAD electronics hierarchy. The ICU interfaces the Spacecraft On-board Data Handling system (responsible for LOFT not only the LAD) and is tasked with the control of all the LAD sub-system, via SpaceWire. The ICU box (two boxes for redundancy reasons) contains the Data Handling Unit (DHU), the Power Distribution Unit (PDU) and a mass memory. The main tasks for the ICU are:

- Telecommand execution and distribution
- Accessing the mass memory
- Time distribution
- Processing and compression

- Collection of housekeeping data
- Monitoring the instruments health
- Calibration tasks

3.1.2 Panel Back-End Electronics

The Panel Back-End Electronics (PBEE) handles all MBEEs on one panel. It buffers their data and compresses it, to already prepare it for transmission to the ground. The connection to the (DHU) is realized via a 100 MHz SpaceWire Interface. The PBEE is also tasked with giving commands to the MBEE and setting it to different operational modes. A PBEE prototype was also developed at IAAT (Figure 3.3). It provided a test facility to test the MBEE and PBEE interface and communication, which helped to improve the VHDL design of the MBEE. The main tasks of the PBEE are to

- Interface and communicate with the MBEEs
- Collect and buffer event packets
- Reformat the data depending on observational modes of the instrument
- Transfer data to the DHU
- Collect housekeeping data and create housekeeping packets

3.2 The Module Back-End Electronics

This section describes the MBEE, the main focus of this diploma thesis. The MBEE is proposed as two identical PCB boards (with two identical FPGAs), with the task of processing data from 16 FEEs, and the communication with the PBEE. It also contains a housekeeping logic (HK) and the power supply unit (PSU). In the following pages, a short description of the HK and the PSU is given, followed by the operational modes of the MBEE, its data processing as well as the interfaces to the FEE and the PBEE. In the last subsection of this chapter, the complete flow of a valid event through the MBEE, from the ASICs to the PBEE is explained.

3.2.1 Housekeeping

The tasks of the housekeeping logic are the monitoring of the temperatures, the voltages and currents and to count the valid or invalid events. In order to measure temperature, voltages and currents of the FEE and the MBEE, analog sensors are attached. The data of these sensors are A/D converted and sent automatically as a package to the PBEE in regular time intervals.
The temperature is measured by at least three sensors on different positions, while the voltages and currents are monitored by 32 sensors on the detector module. The

Figure 3.3: The PBEE prototype developed at IAAT. It contains an onboard MBEE simulation unit which has the ability to send 40 bit sized dummy packages. The connectors on both sides can be connected to the IAAT MBEE prototype board. This connection and the communication between the two boards has been implemented and tested in the frame of this diploma thesis. Image: Henning Wende / IAAT

temperature values are needed for the data processing of the MBEE (see: subsection 3.2.4).
The number of valid events and valid triggers, given by the MBEE while or after processing events, is transferred to the housekeeping logic in order to determine the dead-time of the detector. Also it is used as a "health monitor" during the time of operation.

3.2.2 Power supply unit

The FEE needs high (-1300 V for the drift field) and medium (-100 V, for the pull-up cathodes) voltages, which are provided by the power supply unit in the MBEE. The high voltage (HV) is needed for the electric field in the SDDs, while the medium voltage is necessary for the general operational functions. The power supply unit provides the HV for 8 detectors and is attached directly to the MBEE board. The MBEE itself and the corresponding ASICs will need a low voltage power supply of 3.3 V for the ASICs and 5 V for the MBEE, which also will be provided by the power supply unit.

3.2.3 Operational modes

The MBEE will work in different operational modes. These modes and their transitions are shown in Figure 3.4, while each individual mode is described below.

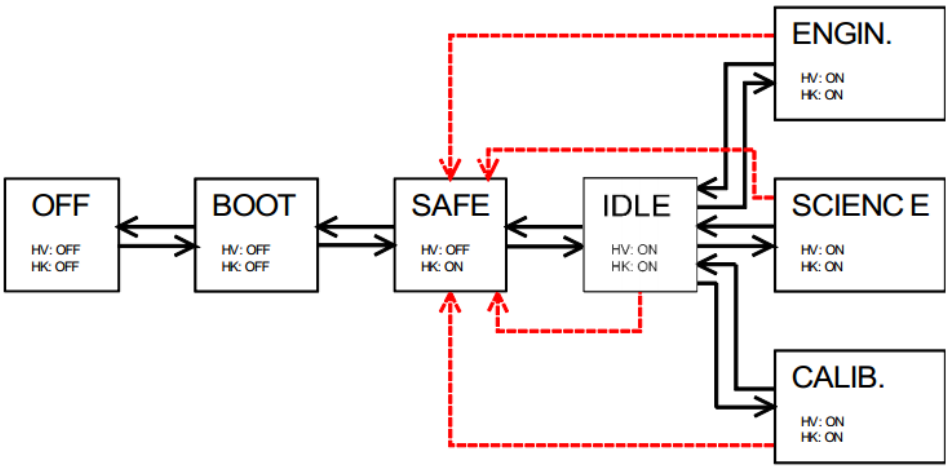


Figure 3.4: Diagram of the operational modes of the MBEE. Black arrows stand for transitions between modes, red arrows indicate emergency shutdowns. HV refers to the high voltage that the power supply unit of the MBEE provides and HK refers to the housekeeping logic. Image: Slawomir Suchy / IAAT

3.2.3.1 Off Mode

All systems, including housekeeping and the high voltage for the detectors, are turned off in this mode. The Off Mode is only for the launch and for the case that the module becomes non-operational or malfunctions due to power-failure or micro-meteor impacts.

3.2.3.2 Boot Mode

In this mode, when the MBEE is turned on, it waits for a stable clock signal and establishes a connection with the PBEE. As part of this boot routine the values for gain (subsubsection 3.2.4.5) and pedestal (subsubsection 3.2.4.3) are loaded in the on-board memory (from the PBEE) and the housekeeping system is turned on. When booted successfully the MBEE goes into the Safe Mode.

3.2.3.3 Safe Mode

This is the mode where the MBEE waits for commands from the PBEE. In the Safe Mode the high voltage is turned off. During operation, when a value of the monitored high voltage exceeds its pre-determined boundary, the MBEE will go to the Safe Mode and turn off the high voltage to prevent damage to the electronics.

3.2.3.4 Idle Mode

In the Idle Mode the MBEE is ready to operate and the high voltage for the detectors is turned on. Data processing is turned off and eventual triggers from the FEE are ignored. Commands can be received in this mode and the MBEE can switch from here to the calibration, science or engineering Mode.

3.2.3.5 Calibration Mode

The FEE-ASICs are provided with a built in capacitor, which allows to induce a calibration charge of adjustable value, on one or on multiple anodes to determine the pedestal values for individual anodes. This on-board energy calibration is done in the calibration mode. The data from the FEE will not be processed by the MBEE, but will be transferred to the PBEE for analysis.

3.2.3.6 Science Mode

This is the standard operational mode for the MBEE. Triggers are accepted from the FEE and data is fully processed, packed and transferred to the PBEE. Event processing in the Science Mode is further discussed in subsection 3.2.6.

3.2.3.7 Engineering Mode

The Engineering Mode allows to bypass individual components of the data processing chain to verify their functionality. In this mode it is possible, to send raw data to the PBEE. This may increase the data rate and should only be used for short times, in order to test individual parts of the data processing chain.

3.2.4 Data processing

This section describes the data processing of the MBEE from one detector half. Here, various corrections to the anode values are applied as depicted in the following subsections.

3.2.4.1 Event processing

When an event is detected by the ASICs, a trigger is sent to the MBEE via the dedicated trigger line. A timetag, based on the 1 MHz clock from the DHU, is instantaneously generated in the MBEE when this trigger arrives. The MBEE then request the trigger map from the ASICs, which is a 128 bit array. For each anode that has triggered, the trigger map contains a 1.

3.2.4.2 Trigger validation

The MBEE checks the trigger map, to see if there are more than three valid entries. A valid event (Figure 3.5) is either a single (one triggered anode) or a split event (two triggered neighboring anodes; can also be split between two ASICs), everything else is not considered a valid event and the MBEE sends a reset signal to the ASICs. For the energy range (e.g. 2 keV to 30 keV) in which the LAD is operating, it is not possible for the charge cloud to spread wide enough to affect three read-out anodes.

If the trigger map is valid, the MBEE requests the values of the anodes of the corresponding ASIC that has triggered. These anode values, 11 bit per anode, are called the charge map.

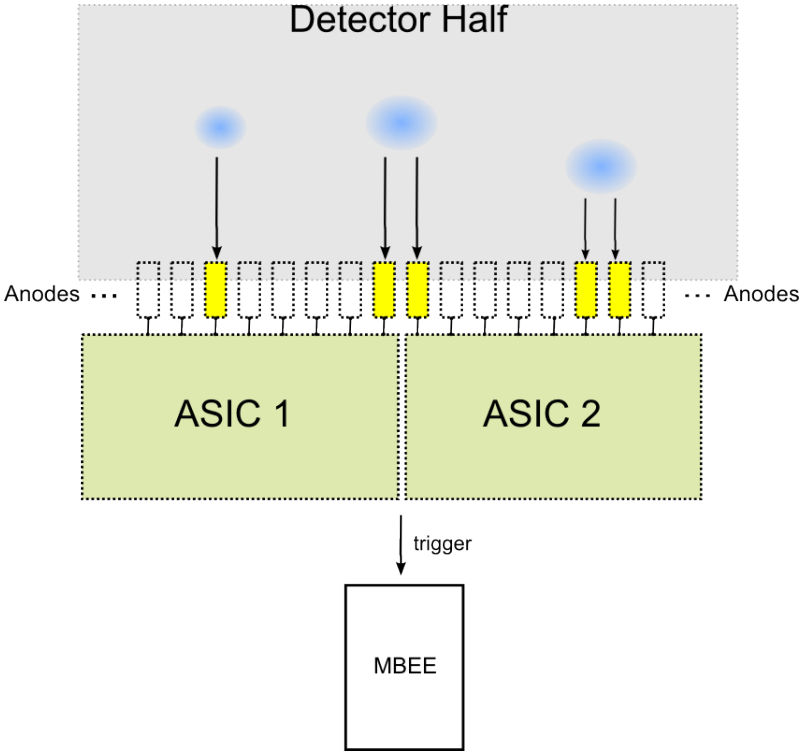


Figure 3.5: Schematic of the possibilities for valid events. On the left, a charge cloud drifts to the anodes and triggers one of them (single anode event). In the middle the charge cloud triggers two anodes that are split between two ASICs (split event). On the right the charge cloud hits and triggers two anodes (double anode event) which is also a valid event. Any other event triggering more than two anodes or triggering two anodes that are not next to each other is considered an invalid event.

3.2.4.3 Pedestal correction

Due to inhomogeneities and structural damage of the detectors, small charges are constantly produced and added to the anode values. These pedestal values vary for each anode and have to be accounted for by subtracting these values from the chargemap. The pedestal values are obtained in the calibration mode and stored in the memory of the MBEE.

3.2.4.4 Common noise correction

The common noise (CN) is a noise component composed of two parts. One being the noise from the detector, the other is the noise introduced by the ASICs. Both effects originate from small fluctuations of the power supply. Thus, the CN will be calculated independently for each ASIC considering only the channels not hit by the charge cloud. A median value will be calculated and subtracted from the chargemap, in order to correct for the common noise.

3.2.4.5 Gain correction

To reconstruct the energy of the event, each of the channel signals of the ASICs is multiplied by an individual gain factor. This gain factor is stored in the MBEE and can be updated from the ground. The gain factor is corresponding to a fixed temperature, but it is in fact temperature dependent, therefore a temperature correction is needed. This is done via the linear correction $E = E \cdot (1 - C \cdot (T - T_0))$ which is applied to the energy value of the event and is not negligible.

3.2.4.6 Energy reconstruction

The energy reconstruction checks first if the event is still valid and has not fallen below a certain threshold, after the applied gain, common noise and pedestal corrections. In this case it then adds up the chargemaps of the triggered anodes and checks that the energy value has not exceeded a higher threshold, in order to exclude unwanted events. In the last step, the energy is scaled in order to generate the final 9 bit sized energy. This is done with the use of a linear function, where 60 eV are stored in one digit (up to an energy of 30 keV; for higher energies up to 80 keV, 2 keV are stored in one digit).

3.2.4.7 Event package

Before the event is passed on to the PBEE, it is packed into a 40 bit sized package. This package consists of a header, a 20 bit timestamp and 9 bit energy value. This will be explained in more detail in section 5.3.

3.2.5 Interfaces

The proposed connections from the MBEE to the PBEE and the FEE are shown in this section.

3.2.5.1 FEE Interface

In the current baseline design, the MBEE and the FEE are connected via a Hyperstac connector. The pin localization is not yet finalized, but 40 pins are foreseen for the LVDS[2] signals, and power supply lines, while 4 pins will be used to transmit the high and medium voltages to the SDDs. The actual structure of a command, that is sent from the MBEE to the FEE, is discussed in subsection 4.3.2.

[2]Low-voltage differential signaling (LVDS) is a technical standard that specifies a communication protocol, where one signal is transmitted over two lines.

Table 3.1: Signals between the FEE and the MBEE.

Signals per FEE	Signal Name	Differential	Direction	Pins on the Connector	Digital/Analog
2	Trigger	yes	FEE to MBEE	4	Digital
2	Hold	yes	MBEE to FEE	4	Digital
2	Reset	yes	MBEE to FEE	2	Digital
2	Calibration	yes	MBEE to FEE	4	Digital
2	CMD	yes	MBEE to FEE	4	Digital
2	CLK	yes	MBEE to FEE	4	Digital
2	DO-E	yes	FEE to MBEE	4	Digital
2	DO-O	yes	FEE to MBEE	4	Digital
1	Temperature	no	FEE to MBEE	2	Analog

3.2.5.2 PBEE Interface

For the connection between the MBEE and the PBEE, a 9-pin Sub-D connector is foreseen (see: Figure 3.6). No SpaceWire Interface could be used, because the RTAX-FPGA of the MBEE, is not suited to integrate the necessary SpaceWire cores along with the processing pipeline. Data signals and the clock signal from the PBEE to the MBEE and vice versa are implemented in the LVDS signal standard. The communication between the MBEE and the PBEE is implemented with a 40 bit custom data package and is discussed in section 5.3.

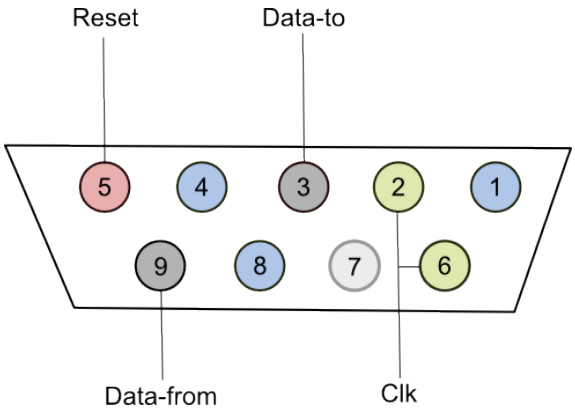


Figure 3.6: The drawing shows the connections from the MBEE to the PBEE. Ground pins are shown in blue, the reset pin in red, the two data pins in grey and the clock pins in green. Pin 7 is unused.

Table 3.2: Signals between the PBEE and the MBEE

Signal Name	Differential	Direction	Pins on the Connector
Clk	yes	PBEE to MBEE	2
Reset	no	PBEE to MBEE	1
Data-to	yes	PBEE to MBEE	2
Data-from	yes	MBEE to PBEE	2

3.2.6 Flow of a valid event

The whole sequence from the arriving trigger to the transfer of the final event package to the PBEE is described in the following.

As shown in Figure 3.7 the procedure is the following:

1. When the accumulated charge of an anode exceeds a certain threshold, the ASIC sends a trigger signal to all other ASICs on the detector half and to the MBEE, via the dedicated trigger line. After a trigger was released, the ASICs are on hold and wait for a command from the MBEE. In addition, the MBEE immediately creates a timetag of the event.

2. The MBEE's detector handling (see: section 5.1) interface is activated by the trigger and requests the triggermap from the ASICs, via a command to the ASICs.

3. The ASICs send their triggermaps to the MBEE. Note that the first two ASICs can send their triggermap even before the last ASIC, e.g. ASIC number seven receives the request to send its triggermap.

4. The MBEE collects the triggermaps from all seven ASICs and validates them. The detector handling interface checks how many anodes have triggered. If there are more than two, the triggermap is invalid and the MBEE will reset the ASICs.

5. If the triggermap is valid, the ID of the triggered ASIC(s) is stored in the MBEE and the chargemap from the corresponding ASIC(s) is requested.

6. The triggered ASIC(s) send(s) all its/their anode values, each as an 11 bit value. When all the anode values are transferred, all ASICs receive the reset command through the reset line from the MBEE and the detector half is active again.

As illustrated in Figure 3.8, the MBEE's functionality can be broken down into three parts. The first part is the interface used for communication with the FEE. The second part is the actual data processing and the third part is the interface to the PBEE. After the steps in Figure 3.7 the MBEE successfully collected the chargemap of an event and starts to process the data according to subsection 3.2.4.

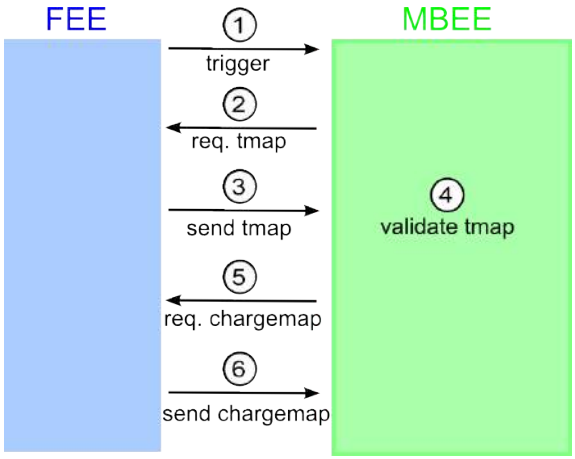


Figure 3.7: General communication steps between the FEE and the MBEE, for a valid event after a successful trigger.

This data processing chain can do this parallel for even and odd ASICs at the same time. In case of a single event, both pipelines get initiated but only one has actual data and in case of a split event between two ASICs both pipelines are working simultaneously.
If the event is still valid after the energy reconstruction it is sent to the PBEE.

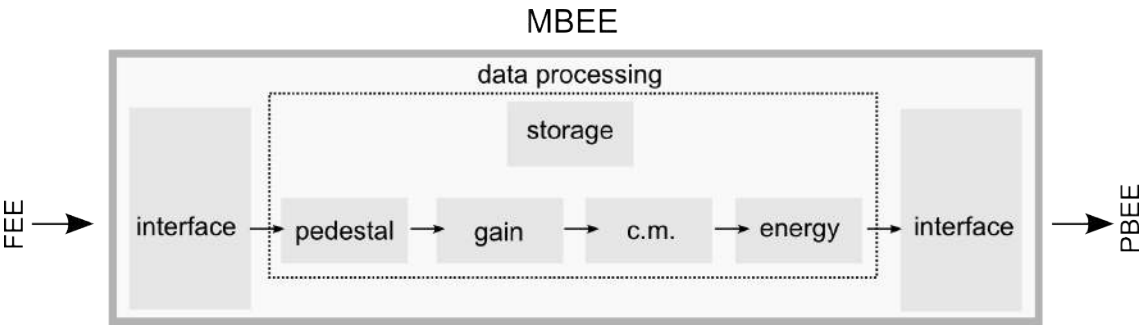


Figure 3.8: Diagram of the data processing flow of the MBEE. The MBEE's functionality can be broken down into three parts. On the left side, the interface for communication with the FEE. In the middle the main data processing, and on the right side the interface for communicating with the PBEE.

3.3 The MBEE hardware prototype

This section briefly describes the MBEE hardware prototype as developed at IAAT (Uter, 2013). The printed circuit board can be seen in Figure 3.10. The board features a USB connection, power supply, FEE-Connection and PBEE-Connection. The final MBEE design includes an RTAX2000 FPGA as the main processing circuit. FPGA, short for Field Programmable Gate Array, is a chip that can be configured by the customer for its purpose. Most FPGAs are flash based; they have a memory chip and their programming can be changed repeatedly. The RTAX2000 is not flash

based, but rather works on an anti-fuse technology. Permanent connections are created in the chip during the configuration, which means it can only be configured once.
In the early stages of development for the MBEE this is obviously not preferred to use (not to mention the high costs). To provide a good development environment which still allows to program for the RTAX2000, an adapter solution was found.
The aldec adapter in use is a flash-based ProASIC FPGA connected to an adapter board that is footprint compatible with the RTAX2000 (Figure 3.9). This way, the adapter can be removed later and the RTAX2000 can take its place without any changes on the prototype board. This aldec adapter is solded to a carrier board, which makes it plug-able. Also, any hardware design is still directed at the RTAX2000, because the aldec adapter simulates an RTAX2000 and code should, in theory, later work the same with the real RTAX2000. On the negative side, because designing and developing is done for an FPGA which is not on the board, designing of the actual ProASIC is much more challenging, than it is compared to other FPGAs. Furthermore, minor differences between the RTAX2000 and the ProASIC like the specific use of internal memory can create further obstacles. In order to configure the hardware prototype one has to do a netlist conversion which converts the code developed for the RTAX2000 to the ProASIC architecture. There is a detailed how-to for this process in the appendix (see: section 7.1), which was also developed as part of this diploma thesis.

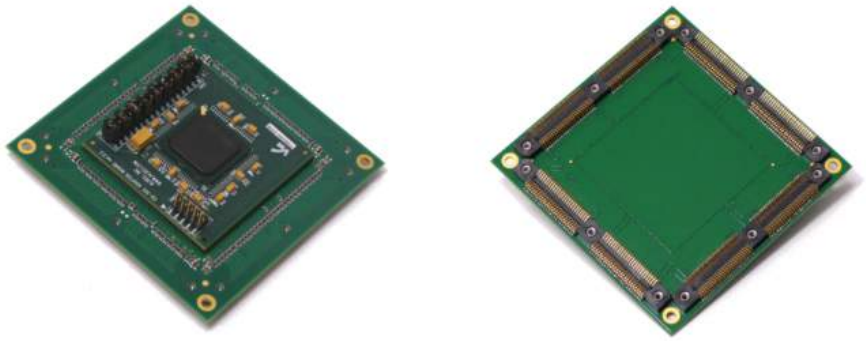


Figure 3.9: Left side: Top view of the carrier board containing the aldec adapter and on it in black the ProASIC FPGA. Right side: Bottom view of the carrier board which can be plugged on the prototype board. Image: Pascal Uter / IAAT

Figure 3.10: The MBEE prototype board. (1) The aldec adapter on the carrier board. (2) The Spartan 3E FPGA, used for the clock distribution and the ASIC simulator which is described in the next chapter. (3) The interface to the PBEE. (4) The interface to the FEE. The other four connectors in the back are just there for symmetrical reason, but have no actual functionality. (5) The power supply for the board. (6) A USB connector. Image: Pascal Uter / IAAT

3.3.1 FEE-ASICs

The Front-End Electronics (FEE) of the LAD is split into two symmetrical rows, each containing 7 ASICs (see: Figure 3.11). The ASICs are daisy chained by a differential trigger line, which is also connected to the MBEE. The trigger can propagate from the first to the last ASIC. The MBEE can force a trigger to all ASICs, which is needed for noise measurements.

Each ASIC has its own hardwired address, so it can be accessed via a command. Also a differential clock signal as well as a reset and hold signal is connected to each ASIC. The data output is divided into even and odd ASICs per FEE side. This separation into even and odd lines, allows for a higher bandwidth when all ASICs transmit their data, or in the case of a split event between two ASICs. Similar to

the MBEE, the FEE has its own operational modes:

- Configuration
  Here, the MBEE transfers the individual configuration registers of the ASICs.
- Science Operation
  The standard operation mode, where events are processed according to subsection 3.2.6.
- Pedestal Measurement
  This mode is used to obtain the baseline signal of each anode. The MBEE puts all ASICs on hold, then commands the A/D conversion for all ASICs, while no trigger has occurred. The obtained values are needed in the pedestal correction of the data processing of the MBEE.
- Calibration
  The MBEE uses the calibration line to generate a voltage inside a specific ASIC (on which anode this voltage is applied can be pre-selected). The exact procedure, for e.g if the ASICs should be put on hold is still under discussion.

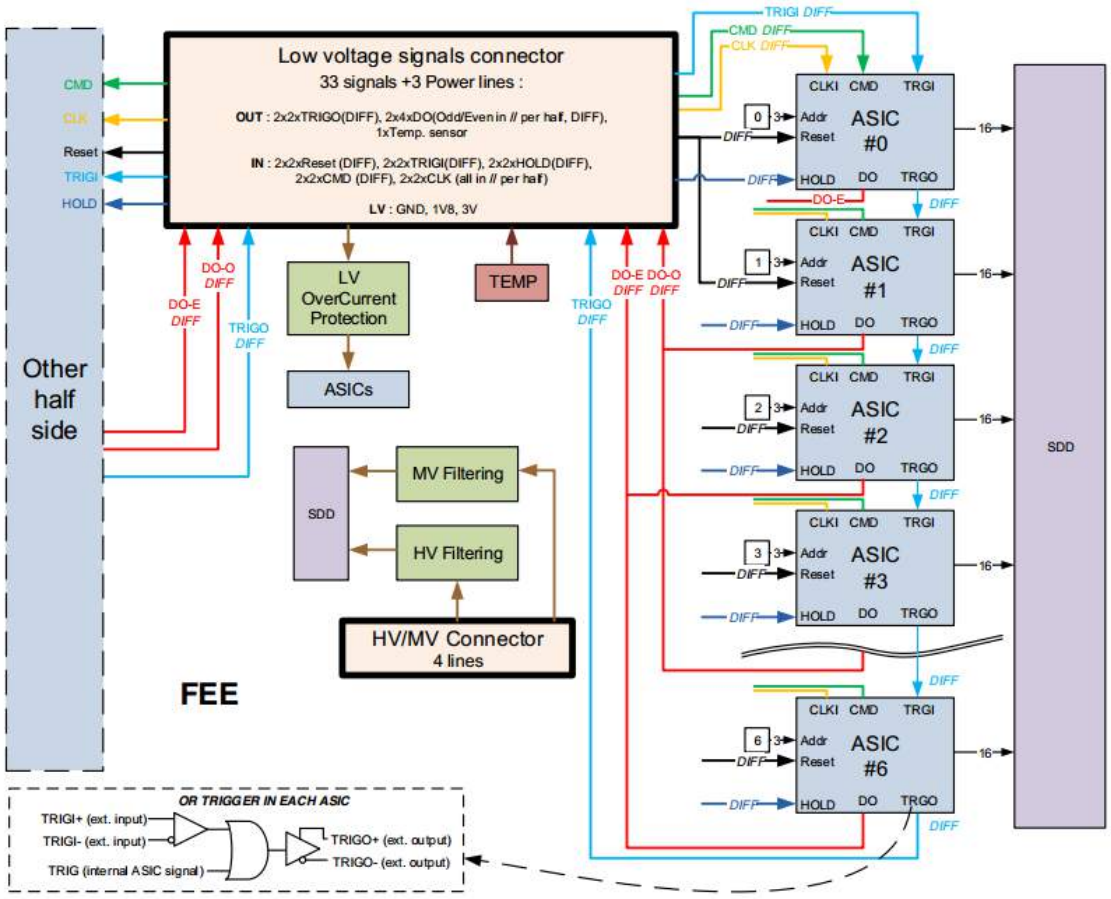


Figure 3.11: Block diagram of the FEE connections to the MBEE, showing the ASICs and their means of connection. Image: Yannick Favre, DPNC/University of Geneva

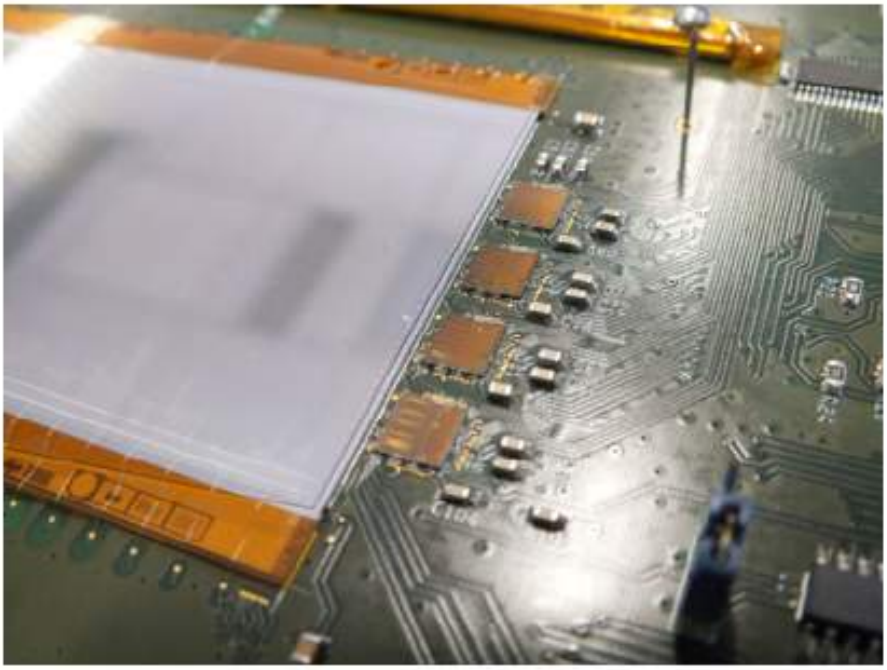


Figure 3.12: ASICs bonded to an SDD prototype. The four quadratic electronic components in the middle of the image are the ASICs. They are bonded here to to the anodes of the detector prototype, in white plane on the left. Image: IRAP / Dolphin Integration

Chapter 4

The VHDL design of the ASIC Simulator

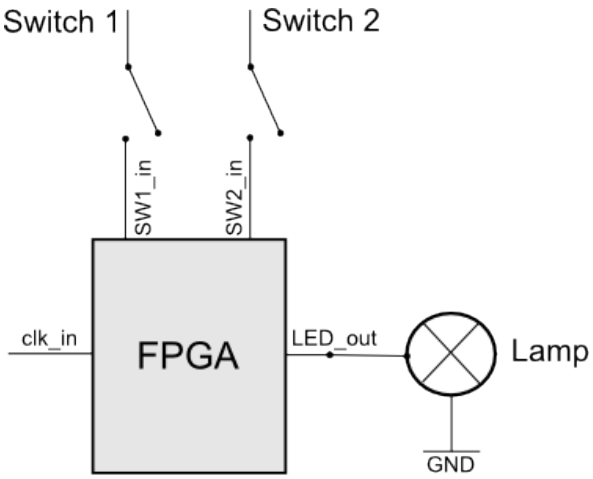
The PBEE, as mentioned earlier was also developed at IAAT, but a working prototype of an ASIC to test with the MBEE was not available. For a realistic testing environment for the MBEE an ASIC is needed. Without the trigger signals, the chargemap the whole communication with the FEE, the MBEE verification would lack some of its most basic test requirements. It is for this reason, that an FPGA based ASIC Simulator in order to work with the MBEE was developed within this thesis.
The Spartan on the MBEE prototype board is not an actual part of the MBEE itself, but it is instead used as a support chip. It was foreseen to harbor the ASIC Simulator, which is described in this chapter, as well as a clock distribution, which also is described below.
The main focus of this and the next chapter is the development of specific hardware designs using the VHDL language. VHDL stands for "Very High Speed Integrated Circuit Hardware Description Language" and is the designing language of choice for hardware development. It is important to state that the target device for the ASIC Simulator is a Spartan3E-FPGA. As the ASIC Simulator is not part of the MBEE it should not run in the RTAX (or the aldec adapter). Designing and configuring of the Spartan3E-FPGA is done with the Xilinx ISE software.

4.1 The basics of VHDL design

In order to understand the VHDL design of the MBEE some basics in VHDL are needed and will be given in this chapter. Those already experienced in designing with VHDL can skip this very basic introduction.

As an example, a simple circuit is used. This circuit consists of two switches, a lamp and an FPGA. The two switches and the lamp are connected to the pins of the FPGA in hardware. Now the FPGA should, via a small hardware design using the VHDL language, turn on the lamp but only when the two switches are

simultaneously switched on.

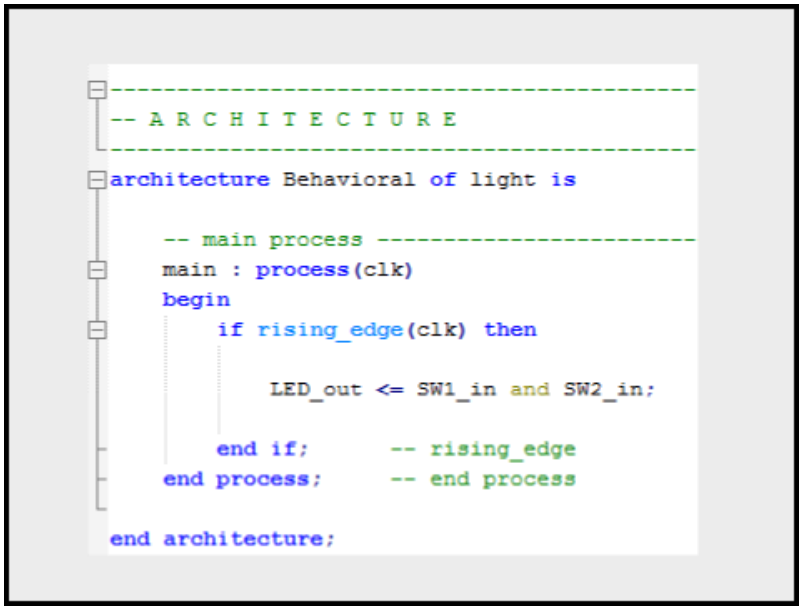



```
-------------------------------------------
-- E N T I T Y
-------------------------------------------
entity light is

    port(
        --IN
        clk_in          : in std_logic;
        SW1_in          : in std_logic;
        SW2_in          : in std_logic;
        --OUT
        LED_out         : out std_logic
    );

end light;
```

Figure 4.1: On the left side, the example containing the two switches and the light is shown. The right side shows the VHDL entity of this example. The signals going in and out are described as the most basic standard logic signals. This means that in this case, these signals are either 1 or 0. Note also that a clock signal (clk_in) is included in the design. Although not needed here for this static problem, normally clock signals are a basic part of VHDL designs.



```
-------------------------------------------
-- A R C H I T E C T U R E
-------------------------------------------
architecture Behavioral of light is

    -- main process ------------------------
    main : process(clk)
    begin
        if rising_edge(clk) then

            LED_out <= SW1_in and SW2_in;

        end if;       -- rising_edge
    end process;      -- end process

end architecture;
```

Figure 4.2: Here the architecture of the VHDL unit of the discussed simple example can be shown.

A typical VHDL design consists of two parts. The first part is the entity and it describes the signals going in and out of the VHDL unit. The second part is the architecture which describes what is done with these signals and how the output signals are produced. In Figure 4.1 it can be seen that the entity basically describes the VHDL unit as an electronic component, while the architecture describes how this electronic component behaves (Figure 4.2).
In case of the discussed example, the two switches are connected via a logic AND. The clock signal in this example is used as a sort of sampling, meaning that the FPGA would check every time the clock signals changes from 0 to 1
(if rising_edge(clk_in)) the two switches and give their combined value to the output LED_out pin. "If rising_edge(clk_in)" creates a register inside the FPGA, which means that the design is now dependent on the rising edge of the clock signal and this kind of edge driven logic is very robust to disturbances. This would achieve the goal set in this simple example that the light only is switched on, when both switches are on.
In this simple example there is only one VHDL unit, and its ports are directly connected to the pins of the FPGA. A realistic project, however, consists of several such units and their subunits (in VHDL also refereed to as components). There are different design approaches and styles for these kind of projects and the resulting VHDL structure. The VHDL design approach of this thesis, consistis of the Toplevel, individual VHDL units and components. The actual VHDL design of the MBEE consist of one Toplevel, managing more than 16 units and their components.

For more complex units it is often a good idea to structure them as finite state machines. A concept often used in designing computer programs and sequential logic. The basic idea is to conceive the wanted design as an abstract machine com-

posed of a finite number of states (Booth, 1967). The machine can only be in one state at a time and a specific signal or a trigger enables the machine to change the states (known as state transition).
Nearly all units in the MBEE VHDL design will have at least two states, one being the actual working state and the other the reset state. In a reset state all the internal values are resetted to their default values. In the following a units behavior will be explained not by locking at its code but rather by first focus on the ports and then try to get its working principles by following the finite state machine.

4.2 Debugging techniques

The Spartan3E-FPGA not only includes the ASIC Simulator (see: section 4.3) and the digital clock manager (see: section 4.4) but it is also used for debugging. Most signals that are important to access in the FPGA design of the MBEE are routed through the board and can be accessed directly for debugging purposes. This way, using ChipScope[1], parts of the MBEE can be monitored in real time. This would otherwise not be possible because the ProASIC does not support a debugging software like ChipScope. On the downside, although allowing for excellent debugging, this also occupies a lot of space and ports. The reason for that is, that all signals one wants to observe with ChipScope will have to be routed through the ProASIC to the Spartan3E-FPGA.

4.3 The ASIC Simulator

The tasks and communication of an actual FEE-ASIC are discussed, followed by the implementation of the ASIC Simulator in VHDL.

4.3.1 Tasks of the ASICs

The purpose of the ASICs is the reading out of the detector anodes and the release of a trigger signal when the anode values exceed a certain threshold. They also send which anodes have triggered, perform an A/D conversion of the anode values and transmit them to the MBEE. In subsection 3.2.6 the procedures of an ASIC in MBEE science mode was shown. A more detailed look, this time the focus is on the ASICs themselves and their exact behavior, is now given in order to simulate the ASICs as exactly as possible.

1. One, or two ASICs in case of a split event, send a trigger signal. Immediately when this trigger signal is send all ASICs are put on hold.

2. All ASICs listen to commands from the MBEE, they receive all commands coming through the even-odd lines in a shift register.

[1]Basically a build in logic analyzer, that allows for the monitoring of previous selected signals or ports inside the FPGA, via an external software

3. When they get the right command (see: Figure 4.3) they start transmitting their trigger map. An even and an odd ASIC can transmit at the same time, but e.g. when two odd ASICs would transmit at the same time, they would render the trigger map useless. This also has to be taken into account in order to build a realistic ASIC Simulator.

4. While the MBEE validates the trigger map, the ASICs wait for commands.

5. The ASIC (or the two) that has triggered is commanded to send his trigger map. Only the triggered ASIC(s) should react to that command.

6. Only the triggered ASIC(s) sends its chargemap to the MBEE.

7. All ASICs are resetted via the dedicated reset line and wait for a trigger again.

4.3.2 Communication with an ASIC

The communication between the MBEE and the FEE is realized via a 20 bit command. This command is sent from the MBEE over the even-odd lines (Figure 3.11 into a shift register of the ASICs. The structure of such a command is shown in Figure 4.3. The ASICs receive this command in a 20 bit sized serial shift register [2]. The ASIC checks constantly to find the header, and further handle the command. In the case of the ASIC Simulator the header contains no information in itself, its purpose is to inform the ASIC of an incoming command. When the ASIC received the header it further checks the command, in order to find if the command is directed at it. This is done by checking the 3 bit ID, ranging from ”000” for ASIC number 1 to ”110” for ASIC number 7. If the ID is correct the ASIC gets ordered to sent the trigger or the chargemap depending on a 3 bit sized sequence (”001” for the triggermap and ”010” for the chargemap).

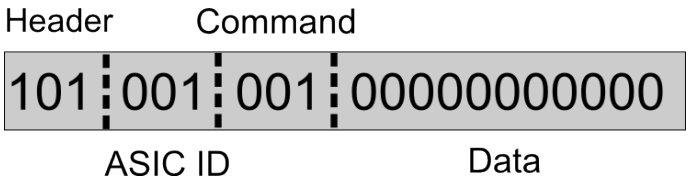


Figure 4.3: The structure of the command between the MBEE and the FEE. This case is showing a command from the MBEE to ASIC number one, sent over the odd line. Starting with the header (101), then the ID of the ASIC the command is directed at, followed by a 3 bit sequence commanding the ASIC, in this case to send his trigger map. The last bits are reserved for sending of actual data, like the chargemap, the triggermap, or other yet unforeseen data exchanges between the MBEE and the FEE.

[2] A shift register is a circuit that shifts data, by one position each clock cycle.

4.3.3 The ASIC Simulator VHDL design

In order to simulate the FEE correctly a working ASIC Simulator is required. This ASIC Simulator can then be used to substitute multiple ASICs and is a sufficient environment to test the communication and the data transfer sequences between the FEE and the MBEE.

4.3.3.1 Structure

The ASIC Simulator or the FEE Simulator consists of two VHDL units. The most important one is the ASIC unit. It simulates the behavior of a single ASIC and tells the ASIC if it should trigger and which values it should send. The goal here was to simulate the behavior of a single ASIC as close as possible and then combine them to make a FEE simulation in hardware possible. The behavior of the ASIC Simulator is shown in Figure 4.4.
The ASIC unit is part of the FEE unit, which initiates the ASICs and connects them in the appropriate way. The FEE in itself has only a structural purpose in grouping the ASICs into even and odd ASICs and propagate the trigger signal. The ASICs all use the same memory where they read out their specific events, as it can be seen in Figure 4.5.

Table 4.1: Signals used by the ASIC Simulator and their purpose.

Name	IO	Size	Purpose
rst	IN	1 bit	A global reset signal, either from the PBEE or a button on the PCB board
clk	IN	1 bit	The 40 MHz clock signal
enable_in	IN	1 bit	The enable signal to start the ASICs via a button on the PCB board
comm_in	IN	1 bit	The command line from the MBEE
ASIC_reset	IN	1 bit	The dedicated reset line from the MBEE
data_out	OUT	1 bit	The data line from the ASICs to the MBEE
trigger_out	OUT	1 bit	The dedicated trigger line to the MBEE
ASIC_diag	OUT	4 bit	The current state of the FSM is encoded in 4 bit array, used for debugging

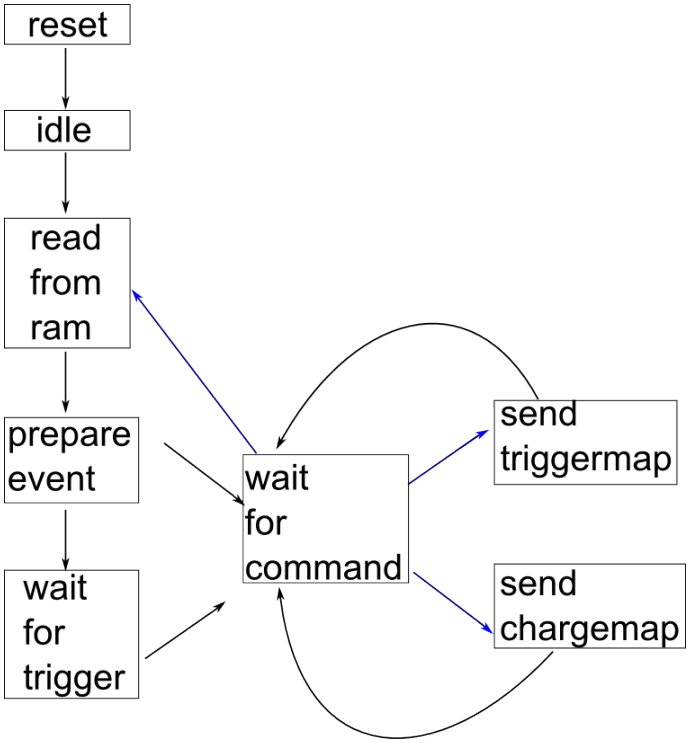


Figure 4.4: A diagram of the finite states of the ASIC Simulator. The black arrows indicate automatic state changes, while the blue arrows show state changes that have to be initiated by a MBEE command. The ASIC Simulator starts in the ”reset” state and then moves in the ”idle” state. In the ”idle” state the ASIC waits to be enabled, which can be either done by pushing a button on the MBEE board or by sending a command to the ASIC, by the MBEE. After that the ASIC acquires its data in the ”read from ram” state, it determines if it should trigger or not during the ”prepare event” state. An ASIC that should trigger waits in the ”wait for trigger state” for a predetermined time and then sends the trigger signal. After the trigger signal all ASICs are in the ”wait for command” state until they revive a command by the MBEE. They can then send their data and get the reset via the dedicated reset line, in that case the ASICs go back to the ”read from ram” state and again acquire new data.

4.3.3.2 Features

Besides from performing the basic digital tasks of an ASIC, the ASIC Simulator has some added features.

- All the various ASIC values, trigger threshold, the anode energy, trigger latency are stored in the RAM and can be changed easily. An easy to use text document can be edited with the wanted values and then a software script automatically performs the needed conversion for the file in order to make it work in the VHDL design (see: Figure 4.6).

- Throughout the whole design of the ASIC unit and the FEE unit the focus has been on developing units that are highly customizable. This means that the number of ASICs, the number of anodes, the size of the trigger and chargemap can be changed. To reduce the triggermap, to less than the 128 bit sized regular

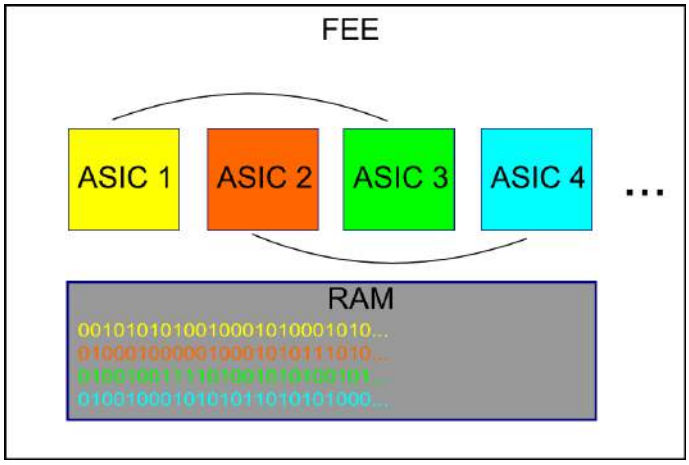


Figure 4.5: The FEE unit contains the ASIC Simulator and RAM. The FEE initiates multiple ASIC units, groups them in even/odd and connects them accordingly. All ASICs access the same Block RAM in the Spartan3E-FPGA, in order to get their data in the "read from ram" state. The first ASIC reads the first RAM line and the second reads the second RAM line and so forth, according to the colors shown in this drawing. After all ASICs returned in the "read from ram" state they shift the address line of the ram via the number of ASICS. If there are for example 4 ASICs then the first time ASIC 1 reads line 1, for the second event ASIC 1 will read line 5. This way the ASICs can continuously provide events for the MBEE. This feature can also be turned off, the same event can occur over and over again, or just once.

one, is especially useful when debugging a design, because if an event is valid can be easily checked manually in a smaller triggermap. The FEE e.g. can be commanded to only send a single event which is especially useful in order to determine its functionality. It also can send events continuously, while mixing some invalid events in between.

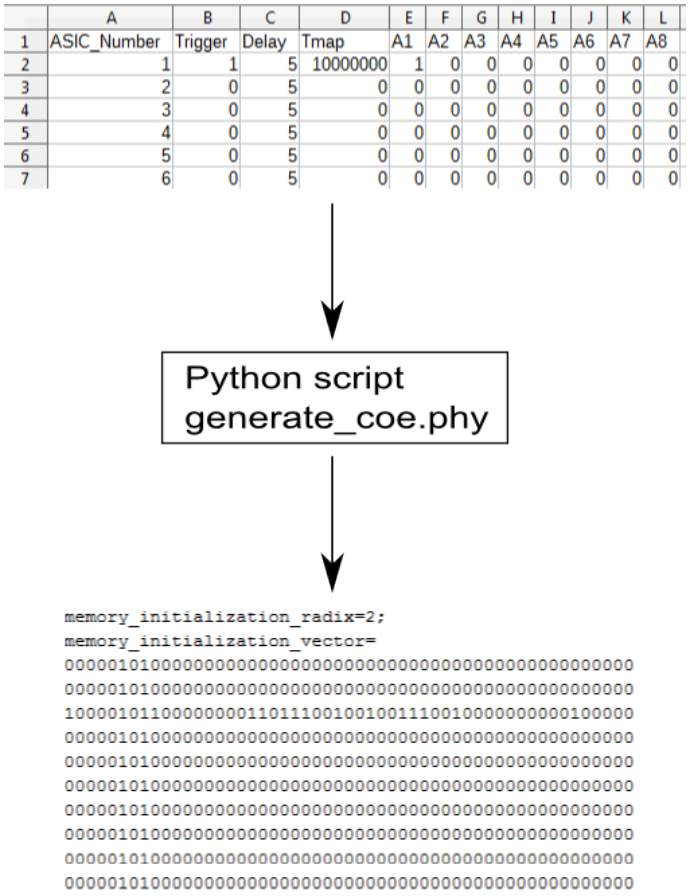


Figure 4.6: In order to fill data in the internal memory of the Spartan3E-FPGA, a ".coe"-file has to be prepared before the chip is programmed. This file can only consist of single lines of binaries. In order to avoid typing hundreds of ones and zeros a more comfortable and efficient solution was found. The data for the ASIC Simulator can be prepared in a table. This file is shown here in the top. The ASIC(s) that should trigger, the trigger latency and every other value can be set here and easily typed in as integers. After that, a software script (realized in the software language python) converts the file into the ".coe"-file (bottom) that can then be stored in the RAM of the Spartan3E-FPGA.

4.3.3.3 Future development

Although the ASIC Simulator is working sufficiently, there are still a few things that could be integrated. This, however, depends on information that has to come during the development of the real ASICs.

- Implement a realistic time in order to simulate the A/D conversion (10 -50 μ s).
- Which IDs do the ASICs actually have, and do they start with zero?
- Implement realistic events and corresponding event rates.

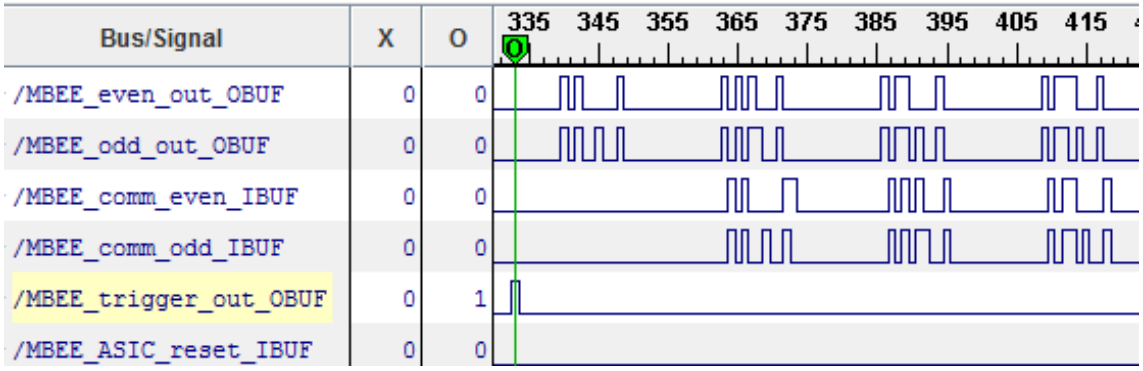


Figure 4.7: A ChipScope image of the working ASIC Simulator. As indicated by the green line, a trigger is sent to the MBEE. The MBEE then sends the command to request the triggermap. This can be seen in the first and second line. In the third and the fourth line the ASICs send their triggermaps via the command structure.

4.4 Clock distribution

The standard clock frequency for the MBEE is 40 MHz, but the interface that connects the MBEE to the PBEE needs a clock of only 5 MHz. The 40 MHz clock signal is generated on the board or it can be given by the PBEE via its connector. A jumper on the MBEE board determines which of the two clock signals will be used. Either way the 40 MHz clock signal is routed through the RTAX-FPGA into the Spartan-FPGA, where a digital clock manager (DCM) buffers and divides the 40 MHz clock signal. The DCM generates a 40 MHz LVDS signal and a 5 MHz LVDS clock signal. When these two clock signals are stable (locked), the two clocks are distributed to the RTAX-FPGA. Should the DCM ever fail to provide a stable clock, then all units will go into their reset modes until there is a stable clock signal again.

4.5 Results

A working hardware ASIC Simulator has been developed and established on the MBEE Prototype board. It has been successfully tested and is now an integral part of the current test setting for the MBEE.
With the use of Chipscope, a debugging solution has been implemented to further improve and check the MBEEs functionality.

4.5.1 Device usage

The device usage describes how many cells of the FPGA will be used by the VHDL design once it is configured. Also, FPGAs have a limited amount of memory, resources and primitives like buffers, or digital clock mangers. This sets limits for the VHDL design and determines if the design can be used to configure the FPGA.
The device usage of the Spartan3E, seen in Table 4.2, contains the VHDL design of the ASIC Simulator, the Clock distribution and the Chipscope Core used for debug-

ging. It can be seen that device usage is not an issue here although a lot of memory is in use.

Table 4.2: Device usage of the Spartan.

Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	1,748	15,360	11%
Number of 4input LUTs	3,054	15,360	19%
Number of occupied Slices	1,736	7,680	22%
Number of RAMB16s	17	64	70%
Number of BUFGMUXs	4	8	50%
Number of DCMs	1	4	25%
Number of BSCANs	1	1	100%

4.5.2 Pin usage

The Pin usage for the Spartan3E is 12%, with the signals for the ASIC Simulator, the clock distribution and a debug port. This leaves room to easily implement much more debug ports on the Spartan3E side, depending on which signals or behavior should be monitored. Not all Pins are wired to the RTAX-FPGA, and furthermore the RTAX-FPGA is more limited in terms of Pins than the Spartan3E is. The Pins currently used by the design contain one debug port, "to_pbee_debug" which allows the monitoring of the packages send from the MBEE to the PBEE, can be found in Table 4.3. The available connections between the two FPGAs on the MBEE Prototype board are listed in the appendix.

Table 4.3: Pins configuration of the Spartan.

IOB Name	Type	Direction	IOB Standard	Drive Strength	Slew Rate	PIN LOC
MBEE_ASIC_reset	IOB	INPUT	LVCMOS25			F4
MBEE_comm_even	IOB	INPUT	LVCMOS25			G5
MBEE_comm_odd	IOB	INPUT	LVCMOS25			J4
MBEE_even_out	IOB	OUTPUT	LVCMOS25	12	SLOW	J3
MBEE_odd_out	IOB	OUTPUT	LVCMOS25	12	SLOW	K5
MBEE_trigger_out	IOB	OUTPUT	LVCMOS25	12	SLOW	E4
button_in	IOB	INPUT	LVCMOS25			D8
clk40_in_n	DIFFS	INPUT	LVDS_25			
clk40_in_p	DIFFM	INPUT	LVDS_25			N8
clk40_locked_out	IOB	OUTPUT	LVTTL	12	SLOW	R11
clk40_out_n	DIFFS	OUTPUT	LVDS_25			
clk40_out_p	DIFFM	OUTPUT	LVDS_25			L2
clk5_dbg_out	IOB	OUTPUT	LVTTL	12	SLOW	T14
clk5_locked_out	IOB	OUTPUT	LVTTL	12	SLOW	R9
clk5_out_n	DIFFS	OUTPUT	LVDS_25			
clk5_out_p	DIFFM	OUTPUT	LVDS_25			J1
led_out	IOB	OUTPUT	LVCMOS25	12	SLOW	N6
master_reset_out	IOB	OUTPUT	LVCMOS25	12	SLOW	M4
reset_from_mbee	IOB	INPUT	LVCMOS25			K4
to_pbee_debug	IOB	INPUT	LVCMOS25			P14
trigger_led	IOB	OUTPUT	LVTTL	12	SLOW	R12

Chapter 5

MBEE design

As already discussed earlier, the MBEE can be grouped in three individual working parts (Figure 5.1). The FEE has already been described in the last chapter (as the ASIC Simulator), this chapter will focus on the detector handling (tasked with communicating with the ASICs), the data processing pipeline (tasked with the processing of the anode values) and the interface with the PBEE. Furthermore, the global structure of the MBEE VHDL implementation will be explained, as well as the individual VHDL units and how their are connected.

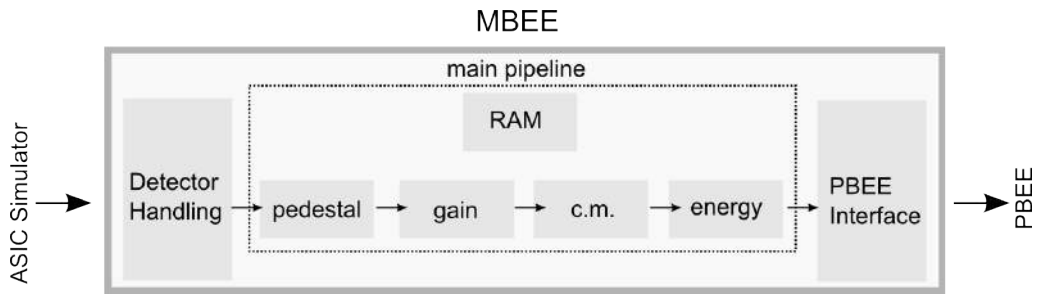


Figure 5.1: Basic drawing of the MBEE's tasks. In this chapter VHDL implementation of the proposed MBEE design will be explained. Starting from the left, the FEE (as the ASIC Simulator) has been shown in the last chapter, the interface to the FEE (the detector handling) will be explained first, followed by the processing pipeline and the interface to the PBEE.

Through the whole MBEE VHDL design, VHDL units were developed on a need-to-know-only basis, meaning no unit should have access to more information than it needs for its main functionality. This sounds obvious but sometimes results in a slightly more complicated design. The reward for this is, that possible changes only affect a few units and have no impact on the overall design of the MBEE. The modular structure of the detector handling unit, discussed next, is a good example for that.

5.1 Detector handling

The detector handling (DH) consists of two components, one being the ASIC communicator (AC) and the other is the trigger evaluator (TE). The detector handling unit commands these two units, and handles the trigger and read out procedure (Figure 5.2). Basically, the ASIC communicator handles everything directed at the ASICs, while the trigger evaluator handles all data coming from the ASICs.

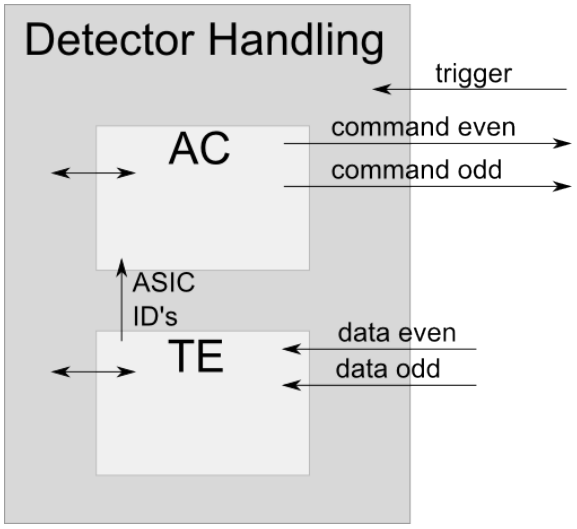


Figure 5.2: Schematic of the structure of the detector handling (DH). The detector handling contains the ASIC communicator (AC) and the trigger evaluator (TE). The connections to the FEE, are shown as black arrows on the right. While the trigger signal is used by the DH itself, the two command lines are driven by the AC, and the two data lines from the FEE end in the TE. The only connection between the AC and the TE is a 3bit signal, containing the ID of the triggered ASICs. A command line from the DH to AC and the TE is implemented, and also several other information from within the two components e.g. the number of triggered ASICs is accessible for the DH.

5.1.1 ASIC communicator

The ASIC communicator (AC) handles all the communication to the FEE. It is tasked with commanding the ASICs, to send their trigger or their chargemap (Figure 5.3). In the ”clear” state, individual counters and signals are reset. Via a command from the detector handling, the AC changes to the ”wait for command” state. From here it can be ordered to either request the trigger or the chargemap. When the AC is ordered to request the triggermap, it begins to send the necessary command to two ASICs simultaneously. In the ”ID tmap” state, the AC verifies at which ASICs it already has sent to. When given the order to get the chargemaps, the AC needs the ID(s) of the triggered ASIC(s). This ID(s) come from the TE and are checked, if they belong to even or odd ASICs in the ”ID chargemap” state. It is also possible for the detector handling to order the AC to send data directly, to all or just one specific ASIC. This is needed due to the requirement of a calibration

mode of the MBEE (see: subsubsection 3.2.3.5), and has already been tested with data from the PBEE, that has been sent to the ASIC Simulator.

Table 5.1: Possible commands from the AC to the ASICs.

Command	Value
Send Triggermap	001
Send Chargemap	010
Enable	111

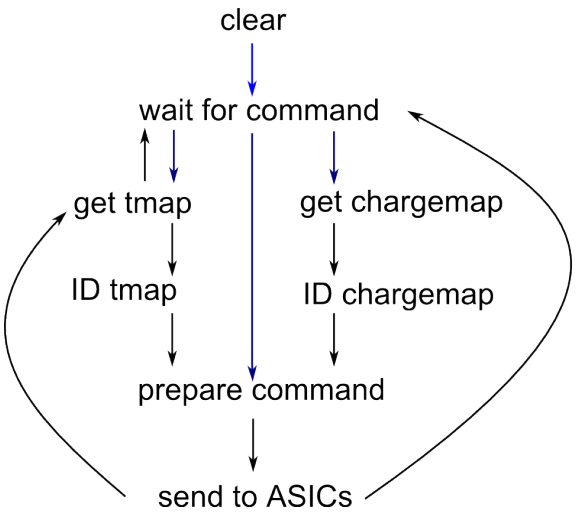


Figure 5.3: A state transition diagram of the ASIC communicator. The blue arrows show commands from the detector handling initiating state changes of the AC, while the black arrows are automatic state progressions.

5.1.2 Trigger evaluator

The trigger evaluator's (TE) main task is to check if a received triggermap is valid, and if so, give the ID(s) of the triggered ASIC(s) to the ASIC communicator. In order to achieve this, the trigger evaluator listens constantly for data coming from the ASICs. When the ASICs send their triggermaps, the trigger evaluator collects them and validates them when it has all triggermaps from all the ASICs (see: Figure 5.4). When an event was valid and the ASIC communicator then has requested the chargemap, it will be collected by the trigger evaluator and handed to the detector handling.

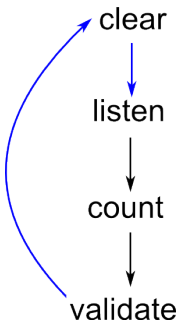


Figure 5.4: The finite state machine of the trigger evaluator. When set in the ”listen” state by the DH, the trigger evaluator checks the data lines from the ASIC, for either their triggermaps or the chargemaps. A detected chargemap will be just handed to the DH. Triggermaps will be collected and then stored as one big array. In the ”count” state this array will be checked and triggered anodes counted. In the ”validate” state the TE checks the triggered anodes and their position in the triggermap, to either reject them as invalid, or if valid, to reconstruct the ID(s) of the triggered ASIC(s). This ID is then given to the AC, as well as the information if the triggermap was valid or not, is given to the DH. When the triggermap is no longer needed the DH will command the TE in the ”clear” state to reset internal values, counters and the TE is ready again to listen for data.

5.1.3 Detector handling

The tasks of the detector handling itself are the handling of triggers from the FEE, creating of a timetag when a trigger arrives, and the collecting of the chargemaps. The chargemaps will then be given to the processing pipeline. These steps can be seen in detail in Figure 5.5.

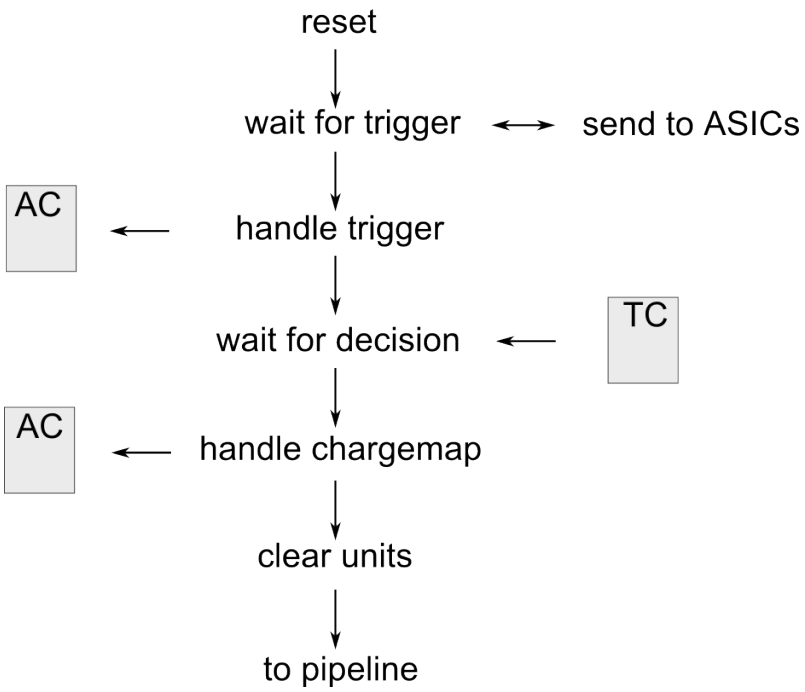


Figure 5.5: Chart of the finite states used by the detector handling for a valid event, in the MBEE science mode. In the "wait for trigger" state, the DH waits for an trigger from the FEE to start the read out procedure, but it can also here receive a command to start sending data to the ASICs. When the PBEE wants to send configuration data to the FEE, it has to wait until the DH is in this state, and is not currently handling a trigger. When a trigger signal arrives, the DH will create a timetag of this event and order the AC to get the triggermaps (done in the "handle trigger" state). The DH will then wait for the ASICs to transmit their triggermaps and also for a decision of the TE, if the event is valid or not. Then the AC is ordered to request the chargemap. When the chargemaps have been transmitted, the DH clears its two components and resets the ASICs ("clear units"). The chargemaps are then passed on to the processing pipeline.

Table 5.2: Signals used by the detector handling and their purpose.

Name	IO	size	Purpose
rst	IN	1 bit	The reset from the MBEE Toplevel
clk	IN	1 bit	The 40 MHz clock signal from the MBEE Toplevel
timestamp_reset	IN	1 bit	A reset signal used in order to generate the timestamp, resets every second
data_from_TL	IN	20 bit	Data from the PBEE that can be send to the ASICs
command_enable	IN	1 bit	Informs the DH, that there is data to send to the ASICs
data_odd	IN	1 bit	The data line from the odd ASICs to the MBEE
data_even	IN	1 bit	The data line from the even ASICs to the MBEE
trigger_in	IN	1 bit	The dedicated trigger line from the ASICs
command_out_odd	OUT	1 bit	The command line to the odd ASICs
command_out_even	OUT	1 bit	The command line to the even ASICs
to_ASIC_reset	OUT	1 bit	The dedicated reset line, to reset the ASICs
idle	OUT	1 bit	Informs the Toplevel, that the DH is able to receive data from the PBEE
solid_send	OUT	1 bit	Informs the Toplevel, that the DH has successfully sent the given data to the ASICs
new_anode_out	OUT	1 bit	A flag set to high when a new anode is revived, needed by the data processing pipeline
ASIC_ID_a_out	OUT	3 bit	The ID of the triggered ASIC
ASIC_ID_b_out	OUT	3 bit	In case of a event split between two ASIC, the ID of the other ASIC is stored in this signal
anode_odd_out	OUT	11 bit	Anode values of odd ASICs
anode_even_out	OUT	11 bit	Anode values of even ASICs
number_of_ASICs_out	OUT	1 bit	The number of ASICs that have triggered
timestamp	OUT	20 bit	The timestamp of the event, created when a trigger is received

5.2 Processing Pipeline

The processing pipeline is tasked to reconstruct the energy from the incoming chargemaps, given by the detector handling. In order to achieve this, several corrections have to be applied before the energy can be obtained. These corrections are done by small VHDL units discussed later in this section. The structure of this processing pipeline is shown in Figure 5.6.

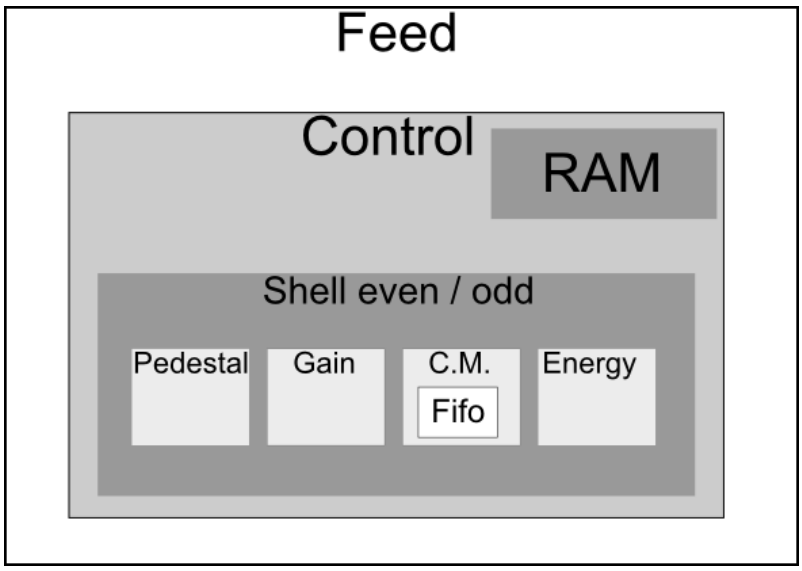


Figure 5.6: Drawing showing the structure of the implemented data processing pipeline of the MBEE. The pipeline Feed unit contains the actual processing pipeline, and all components, and serves as a connector between the detector handling and the pipeline. The detector handling gives charmaps, but the pipeline expects them already to be in a certain format, the pipeline feed unit provides this format. The pipeline control unit contains a Ram and the pipeline shell unit, which contains the VHDL units necessary for the corrections applied to the chargemap. The pipeline control unit generates two versions of the pipeline shell unit, one for even and one for the odd chargemaps. The output from the incoming chargemap is a 8 bit sized energy value, if the event was considered valid by the energy unit after the applied corrections.

In order to successfully apply these correction, it is required to store realistic values in the memory of the FPGA. These values are needed for the gain and the pedestal correction. The target device for the MBEE is, as already mentioned, the RTAX-FPGA, which has a so called protected RAM. This special RAM, suitable for space environment, needs a specific interface and integration. The actual FPGA on the board, the ProASIC-FPGA, however, uses a regular block RAM. For the sole purpose of simulating the design, one could use the protected RAM from the RTAX-FPGA, but this would not work on the actual board. But since the whole design is directed at the RTAX-FPGA, it would not make sense (and also not work) to try and implement the regular block RAM that the ProASIC-FPGA uses. Therefore, in order to test functionality in both simulation and actual hardware, a simple RAM simulator was implemented. This RAM simulator, of course can not really hold

realistic pedestal or gain values, and therefore just assumes a constant. But it can provide the correct timing and procedures for both simulation and actual hardware implementation. This solution was chosen since it is of much more importance to show the whole flow of the design working in hardware, rather than showing the right corrections are applied to the chargemap. In case of designing for the RTAX-FPGA in hardware, this then would be no problem, because only the RAM simulator file would have to be exchanged for the protected RAM and all VHDL units, pipeline structure and procedures would remain intact.

Table 5.3: Signals used by the processing pipeline and their purpose.

Name	IO	size	Purpose
rst	IN	1 bit	The global reset
clk	IN	1 bit	The 40 MHz clock signal
valid_in	IN	1 bit	The valid is high, when a new anode is ready for the pipeline, otherwise always zero
ASIC_ID_a_dh	IN	3 bit	The ID of the triggered ASIC
ASIC_ID_b_dh	IN	3 bit	The ID of the other triggered ASIC, in case of an event split between two ASICs
anode_even_dh	IN	11 bit	The anode values from the even ASICs
anode_odd_dh	IN	11 bit	The anode values from the odd ASICs
energy_out	OUT	9 bit	The energy from the event
energy_valid_out	OUT	1 bit	High, when a new energy is ready, otherwise zero

5.2.1 Pedestal unit

The pedestal unit subtracts a pedestal value from the chargemap (11 bit), and then passes the corrected chargemap on to the gain unit. These pedestal values vary per individual anodes and are stored in the RAM of the MBEE. Since, as stated above, the RAM is not accessible in this case, instead a constant is subtracted from the chargemap. After the subtraction the chargemap could be negative, therefore one bit indicating the sign of the chargemap is added.

5.2.2 Gain unit

The gain unit multiplies the chargemap with a gain factor. This gain factor is stored in the RAM for each individual anode. In order to multiply two arrays in VHDL, a special designed multiplier core is used. This multiplier core expects the chargemap (signed 12 bit) and the gain constant (12 bit) as input and hands out the product of the two. Similar to the pedestal unit, and the RAM issue, the chargemap is multiplied with a constant. The gain unit passes a gain corrected, now 25 bit chargemap to the common mode unit.

5.2.3 Common mode unit

The common mode unit is tasked with subtracting the median values of all anodes from the chargemap. In order to calculate this median value, the common mode unit sorts all incoming anodes and then calculates it by subtracting the median value from all chargemaps. Incoming chargemaps are stored in a FIFO[1], before they are processed and passed on to the energy unit.

5.2.4 Energy unit

The energy unit reconstructs the energy of the event. It sums the chargemaps of anodes that have triggered and are still valid after the applied pedestal, gain and common mode correction. A threshold check for these energy values is currently not included. One reason for that is the fact that due to a non useable RAM the gain and pedestal correction are always wrong and it can not be checked for a ”realistc” energy.

5.3 PBEE Interface

The MBEE always initiates the communication with the PBEE by sending a 40 bit sized package. The structure of this package can be seen in Figure 5.7. The PBEE responds to the package, by sending back an acknowledge package (Figure 5.9). When the PBEE wants to send to the MBEE, the PBEE has to reverse the communication. In order to achieve this, the PBEE can change the mode of the MBEE via a command, setting the MBEE into a receiver state (Figure 5.10).
The structure of the PBEE Interface can be seen in Figure 5.8.

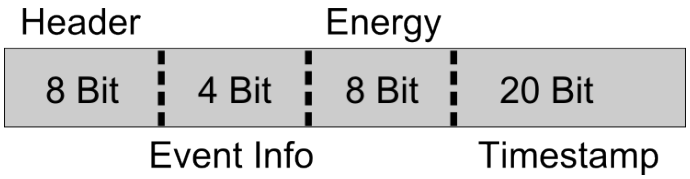


Figure 5.7: The structure of the command package sent from the MBEE to the PBEE. In this case the MBEE sends an event to the the PBEE, containing the Header, event information (e.g. split or single event), the energy and the timestamp when the trigger of the event arrived in the Detector handling. There are currently multiple possible headers, a dummy event header, event header and a error header. Also the actual data in the 4 bit sized event info is not yet decided.

[1]First In First Out (FIFO): A buffer that can store data and hands it out in the order it was inserted

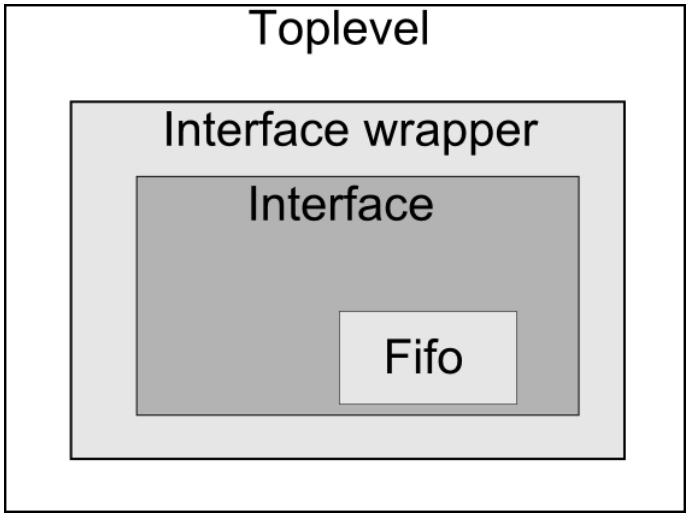


Figure 5.8: Drawing of the structure of the Interface to the PBEE. The Interface is directly managed from the Toplevel. The actual PBEE Interface, is imbedded in the Interface wrapper and contains a FIFO, in which the data is stored before transmitted to the PBEE.

Table 5.4: Signals used by the PBEE Interface and their purpose.

Name	IO	size	Purpose
rst_system	IN	1 bit	The global reset
clk_system	IN	1 bit	The 40 MHz clock signal used by the MBEE
rst_interface	IN	1 bit	The reset for the Interface
data_from_pbee	IN	1 bit	The connection to receive data from the PBEE
enable	IN	1 bit	The Toplevel sets the enable to high, when it wants to send data to the PBEE
data_send	IN	40 bit	The data that gets send to the PBEE; It is set by the Toplevel
data_to_pbee	OUT	1 bit	The connection to send data to the PBEE
transmitting	OUT	1 bit	High when the Interface is sending data to the PBEE
ready	OUT	1 bit	High when the Interface is able to handle new data
send_mode	OUT	1 bit	High, when the MBEE is the sender, and low when it is the receiver
data_receive	OUT	40 bit	The data received from the PBEE

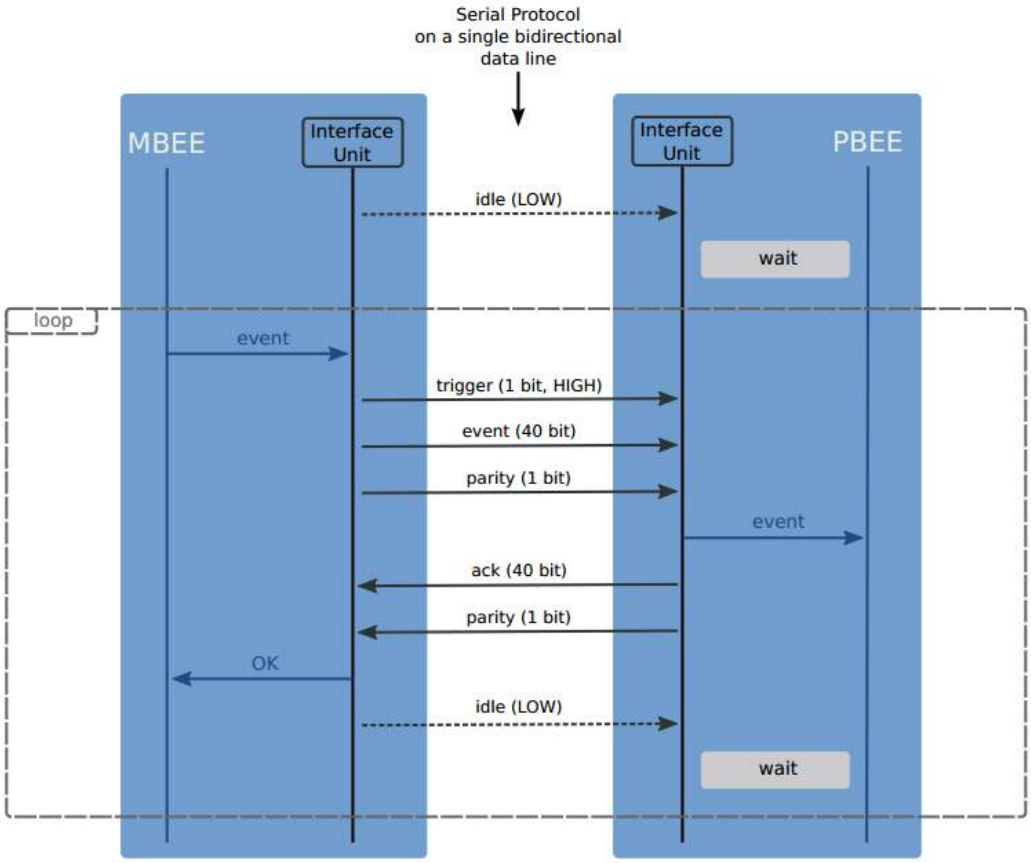


Figure 5.9: The Diagram shows the communication flow between the MBEE on the left and the PBEE on the right. The arrows in black indicate the communication between the two boards, while the blue arrows show the communication within the boards. In this mode, the standard mode of the communication from the MBEE to the PBEE, always the MBEE starts the communication. Before the MBEE sends it 40 bit package, it pulls the data line to low. After a successful transmission from the MBEE to the PBEE, the PBEE checks the parity bit and answers with a 40 bit acknowledge (ack) package. After the MBEE has received the ack it can start transmitting again. Image: Henning Wende / IAAT

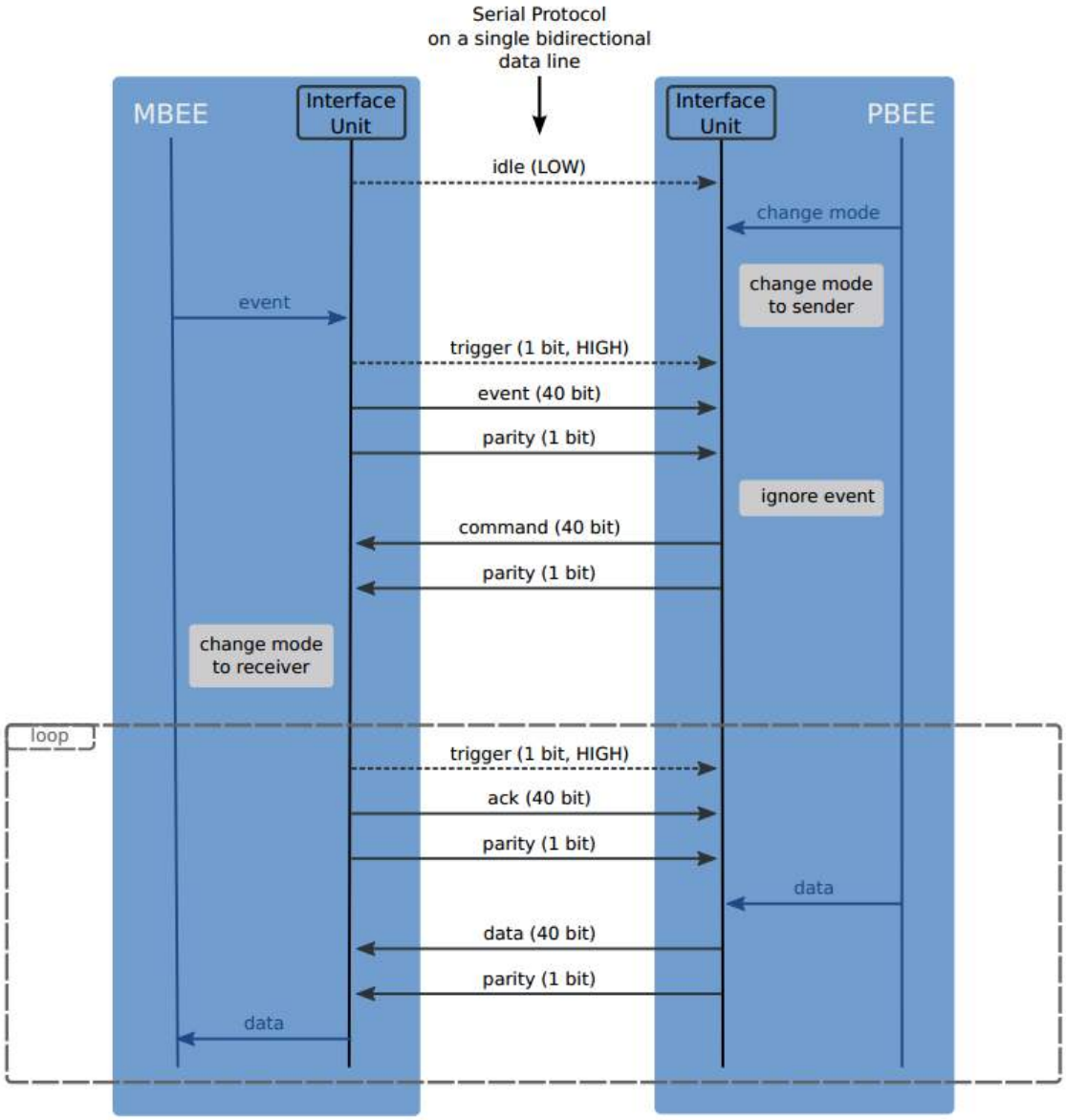


Figure 5.10: The schematic shows the communication flow between the MBEE and the PBEE. The PBEE is able to change the direction of the communication, by sending a command to change the state of the MBEE. The MBEE is then the receiver and instead of sending events to the PBEE it begins its communication by sending the 40 bit ack, and then receives commands or data from the PBEE. Image: Henning Wende / IAAT

5.4 MBEE

The structure of the MBEE VHDL design can be seen in Figure 5.11. The MBEE VHDL toplevel is also designed as a finite state machine. These global modes of the toplevel are oriented on the communication with the PBEE. Basically, the toplevel follows the design of the interface, with its reciver and sender states. Therefore, the toplevel always checks the acknowledge sent from the PBEE, and changes its states if necessary. For most of the time, except when receiving calibration data, the MBEE sends its events to the PBEE. The actual command as seen in Figure 5.7 is implemented in the MBEE toplevel and is highly customizable. Instead of transmitting an event, with its energy and timestamp, the toplevel can e.g. send the raw data as it comes out of the detector handling. This particular design structure enables the MBEE to send basically every signal that is available in the toplevel directly to the PBEE. This was implemented to achieve the goal that the PBEE can, via the acknowledge, order the MBEE to send specific data when needed. Therefore it becomes possible for the MBEE, e.g. to store additional information like errors that may occur and sent them to the PBEE if required.

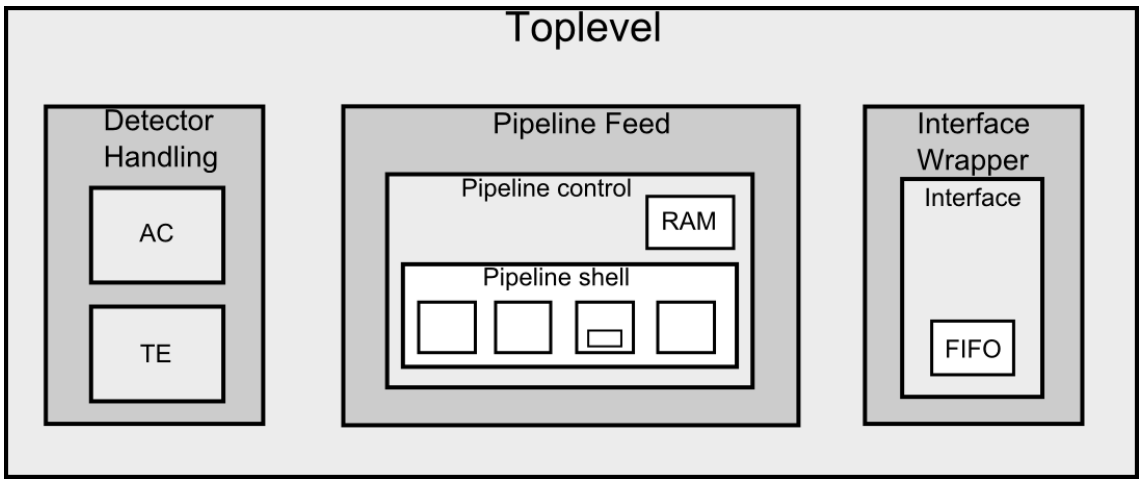


Figure 5.11: The structure of the MBEE VHDL design includes the detector handling, the data processing pipeline and the PBEE Interface. This drawing shows the VHDL implementation of the MBEE's tasks. Every data available to the toplevel, can be given directly to the PBEE via the PBEE Interface and by changing its sender to receiver state, the MBEE can also reverse the communication flow and send data coming from the PBEE to the ASIC Simulator.

5.5 Results

The main results of this thesis are:

- The development and implementation of an ASIC Simulator on the MBEE prototype board.

- The implementation and testing of the communication from the PBEE and MBEE prototype boards.

- A fully working VHDL design for the MBEE.

The ASIC Simulator has been discussed in chapter 4 and the VHDL design of the MBEE in chapter 5. Now a closer look on the timing, possible event rates and the device usage of the RTAX-FPGA, in which the MBEE operates, is given. It will be shown that the timings meet the overall requirements and that the expected event rates can be easily handled. Problems concerning the Pin usage of the RTAX-FPGA will be addressed in (see: subsection 5.5.4) the last subsection of this chapter.

5.5.1 Timing considerations

Table 5.5 shows the timing of the current MBEE VHDL design interacting with the ASIC Simulator. The values have been obtained directly from the MBEE Prototype board, via ChipScope and are not simulated.
It takes 5 clock cycles for the MBEE to react to a trigger and request the triggermap from the ASICs. After 74 clock cycles the MBEE has validated the triggermap, found the triggered ASICs and sends the command to request the chargemap. The whole interaction from an incoming trigger, to handing the chargemaps to the data processing pipeline takes 394 clock cycles. The pipeline itself needs 210 clock cycles in order to process the data and reconstruct the energy of the event. The full process from an incoming trigger to the successful sent of the event to the PBEE takes 900 clock cycles.
It is important that the pipeline can process events faster than the detector handling delivering them, to not bottleneck the design. Also, in the current design, the ASIC Simulator takes no time to make the A/D conversion, therefore making sure that the MBEE can handle the data no matter how fast the ASICs really are. This current design, for one FEE only, easily meets the timing requirements for the MBEE. The resulting event rates are discussed in the next subsection.

Table 5.5: Timing values for the MBEE.

Clock cycles	Time [ns]	Activity
5	125	From incoming trigger to send command
74	1850	Validate triggermap, identify the triggered ASIC, and send the command for the A/D conversion
394	9850	From trigger to give the chargemaps to the pipeline
210	5250	Pipeline applying all the corrections and handing out the energy
900	22500	From incoming trigger to transmit the event successfully to the PBEE

5.5.2 Event rates

A flux of 500 mCrab, from a source in the LADs field of view, would create a count rate of ≈120000 cts/s for the whole LAD (Suchy et al., 2012). This would correspond to 30 cts/s on one detector half. For a 40 Crab source the expected count rate per detector half would be ≈2400 cts/s.
The current design can process a new event every $9,850$ μs per detector half, without taking the time for the A/D conversion of the ASICs into account. This would lead to an upper limit of ≈101522 cts/s per detector half. Assuming a realistic time for the A/D conversion of 10 μs this count rate transfers to 50377 cts/s for one detector half, which is still way beyond the requirements.

5.5.3 Device usage

The device usage for the RTAX-FPGA for one FEE and the MBEE VHDL design uses ≈14% of the logic cells, but needs all the possible clock networks. But as no other clocks are foreseen for the MBEE, this is of no concern. This design uses 36 Pins, with the PBEE communication rooted to the Spartan in order to use ChipScope.

Although the overall usage is quite low, this is misleading. The used logic (≈14%) is for one FEE and the corresponding data processing pipeline. It is most likely possible to fit all 16 FEEs. The design uses no division cores, which often provide challenges due to their high device usage, and most functionality does not change with the number of FEEs (the toplevel or the PBEE interface for example). Therefore, and based on experience with previous projects of the IAAT, device usage should be sufficient. The limiting factor and the reason why it was not possible to implement the full design were problems regarding the PIN usage, which are discussed next.

Table 5.6: Device usage of the RTAX.

Type	Used	Available	Utilization
Sequential (R-cells)	1487	10752	13, 83%
Comb (C-cells)	3092	21504	14, 38%
Logic (R+C-cells)	4579	32256	14, 20%
RAM/FIFO	3	64	4%
IO	36	198	18, 19%
Clock (routed)	4	4	100%
HClock (hardwired)	2	4	50%

5.5.4 Pin usage

The Pin shown in Table 5.7 are the Pins of the RTAX-FPGA before the netlist conversion, where the Pin Number from the RTAX-FPGA gets converted to that corresponding of the ProASIC-FPGA. The Pins shown in Table 5.7, are the PINs

needed for the MBEE designs functionality and the communication with the ASIC Simulator, without any debug ports.

The RTAX-FPGA provides 198 Pins and the Pins used for one FEE are only 20, it should be possible to connect all FEEs foreseen for the MBEE. This is unfortunately not the case. The Pins on the RTAX-FPGA are grouped in banks from zero to three. If, for example, a LVDS signal is rooted to two PINs, the RTAX-FPGA also blocks surrounding Pins. This behavior limits the Pins that can be used freely by a huge margin. In case of bank 1, where the clocks are handled via LVDS and a specific buffer this limits the amount of free pins so dramatically that no FEE could be connected to this bank. When assigning the required clocks, the ASIC Simulator Pins, communication with the PBEE via the Connector and a few debug ports, the Pin usage is almost at its limit. It is yet unclear if this problem gets worse due to the aldec adapter, and the required netlist conversion and how this would translate with a real RTAX-FPGA. It could be possible that the Pins foreseen with the current RTAX2000-352CQFP are simply not enough.

Table 5.7: Pin configuration of the RTAX for basic functionality without debugging.

Port Name	Macro Cell	Bank Name	IO Standard	Slew	Resistor Pull	PIN Number
ASIC_comm_even	OUTBUF	Bank 0	LVTTL	High	None	338
ASIC_comm_odd	OUTBUF	Bank 0	LVTTL	High	None	337
ASIC_data_even	INBUF	Bank 0	LVTTL		None	336
ASIC_data_odd	INBUF	Bank 0	LVTTL		None	335
ASIC_reset	OUTBUF	Bank 0	LVTTL	High	None	342
ASIC_trigger	INBUF	Bank 0	LVTTL		None	343
clk5_in_N	HCLKBUF_LVDS	Bank 1	LVDS		None	299
clk5_in_P	HCLKBUF_LVDS	Bank 1	LVDS		None	300
clk40_in_N	HCLKBUF_LVDS	Bank 1	LVDS		None	305
clk40_in_P	HCLKBUF_LVDS	Bank 1	LVDS		None	306
data_from_pbee_in	INBUF	Bank 3	LVTTL		None	184
data_to_pbee_out	OUTBUF	Bank 3	LVTTL	High	None	183
LED0	OUTBUF	Bank 2	LVTTL	High	None	230
LED1	OUTBUF	Bank 2	LVTTL	High	None	229
LED2	OUTBUF	Bank 2	LVTTL	High	None	224
LED3	OUTBUF	Bank 2	LVTTL	High	None	223
reset_from_DCM	INBUF	Bank 0	LVTTL		None	326
RESET_in	INBUF	Bank 3	LVTTL		Down	181
RESET_onboard	INBUF	Bank 3	LVTTL		Up	180
RESET_out	OUTBUF	Bank 0	LVTTL	High	None	332

Chapter 6

Summary and Outlook

6.1 Summary

The LOFT mission is designed to study timing variability in X-ray sources. It focuses on compact objects and especially on the understanding of the Equation of State in neutron stars. The main Instrument for these timing studies is the Large Area Detector, which has an effective are of $\approx 10\ m^2$. The Large Area Detector follows a modular design, with special electronics mounted on the detectors, the Modules and the Panels.
The Module Back-End Electronics is the electronics mounted on the back of each Module, and handles communication with the Front-End Electronics, the Panel Back-End Electronics and the data processing of the anode values from the detector.
A hardware Prototype of the MBEE board is already available at IAAT. The goal of this thesis was to implement the functionality of the MBEE, via the hardware description language VHDL. In order to provide a good testing environment an ASIC Simulator has been developed. This ASIC Simulator can be easily calibrated and is used to test the communication between the MBEE and the FEE.
The MBEE VHDL design includes a detector handling, focused on communication with the FEE and the validation of the triggermap. A data processing pipeline, which applies the necessary calculations and corrections, in order to reconstruct the energy of an event. An Interface to the PBEE, responsible for the communication and transmitting of the event packages. All parts of the MBEE VHDL design have been tested on the Prototype board, including the communication with the ASIC Simulator (stored in a separate FPGA on the Prototype board), as well as the communication with an PBEE Prototype, also developed at IAAT.
Within the current limitations (such as the non-function RAM; the limitation of available Pins) the MBEE is in good condition and already exceeds what would be necessary at this early project stage.

6.2 Outlook

The MBEE Prototype is already in a good state and would profit the most from the opportunity to test it with an actual FEE Prototype. With the current setup, and its already discussed limitations, further development or code optimization is not a priority. Although, the implementation of e.g. global error states for the MBEE could be a nice feature, it would be far more useful to address the RAM and the Pin issues. This may include the necessity to develop a new MBEE Prototype board. However, with the current advanced state of the MBEE (and the PBEE for that matter), more information from the other electronics component have to be provided at some point, to ensure compatibility and communication standards.
Unfortunately LOFT has not been selected as an M3 Mission, but it will compete again for a launch spot in the next M4 Mission, scheduled for 2025. Given the fact that the LOFT case was well received by the community and the ESA Science comity, LOFT has good chances to be selected as an M4 Mission. LOFT could then provide new insights for neutron stars and black holes, as well as make new discovery's and supply large amounts of data for the X-ray community.

Chapter 7

Appendix

7.1 How to design for the aldec Adapter on the MBEE Prototype board

This Version is from October 2013.
Maybe a few things have changed already, like the Libero or FlashPro Version/Layout. Please keep that in mind.

7.1.1 What is needed

This is just a short description of what you should have and what your software setup should look like.

Libero
You need Libero IDE, choose the latest version. Please don't use Libero SOC because it doesn't support the RTAX. (October 2013).
You need a dongle for that.
You need a valid License for the use of the Libero software.

Netlist Converter
You need the RTAX2A3P Netlist converter. The license for that is valid till 2017, but there is no update support. You have to use the version on the CD.
You need another dongle for that.

Programmer
You need the Programmer for the Aldec Adapter. It will not work with any other Programmer.

Figure 7.1: The Programmer for the aldec adapter

7.1.2 How to install

Libero

- Choose the latest version of Libero IDE, or maybe SOC, if your in the far future and it supports everything and actually not crashes every 2 minutes.
- Get a how-to-install from the microsemi webpage and install for your system exactly like they say.
- Don't forget to set your system environment variable(s) accordingly.
- Make sure to install Simplify Pro and Modelsim. For Modelsim you need both, the ProASIC and the Accelerator packages.

RTAX2A3P

- Install the RTAX2A3P from the software CD,
  by executing the Aldec.RTAX2A3P.Converter.2011.01.RTM.exe.
- Copy the license file from the other CD to your hard drive.
- Set a system environment variable like you did in the Libero installation to the "license.dat" file from the CD. Name it "ALDEC_LICENSE_FILE".
- Insert the dongle, make sure it installs the drivers correctly, then start the RTAX2A3P.
- If everything is correct it looks like Figure 7.2.

RTAX2A3P Patch

- You need the patch, if you want to use the HCLK LVDS buffers, which you definitely want.

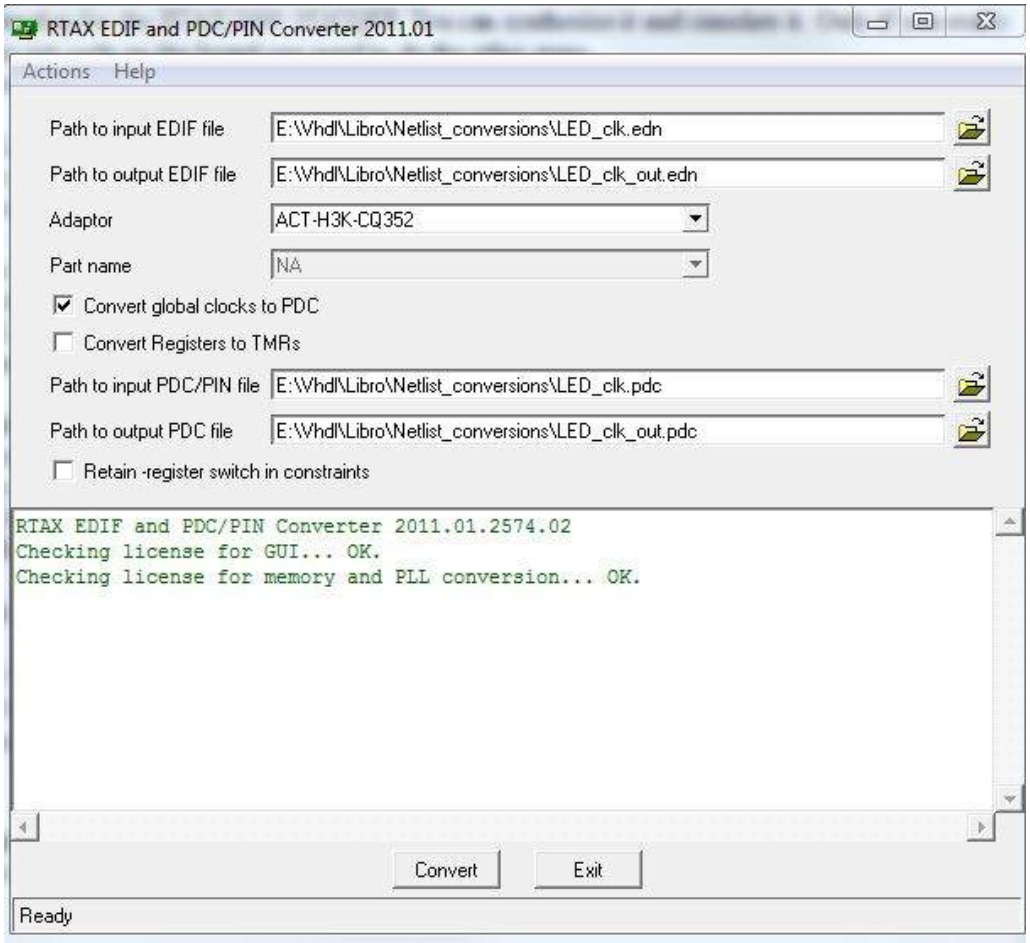


Figure 7.2: The netlist converter looks like this, when set up correctly

- Copy edif files form "patch\cfg\rtax\edifs\" to <Install_dir>\RTAX2A3P.2011.01\cfg\rtax\edifs\.
- Copy proasic3_primitives.cfg file form "patch\cfg\rtax\proasic3_primitives.cfg" to <Install_dir>\RTAX2A3P.2011.01\cfg\rtax\proasic3_primitives.cfg.

7.1.3 The Process

Before you start, make sure that everything is set up correctly. You must have a working Libero, no license errors. And a working RTAX2A3P, including the patch and no GUI errors. Make sure the board works as it should. The board needs 5V to operate. Power it up, the 4 red LEDs should be on, 3 bright 1 barely.
Also confirm that you have the right programmer, it does not work with the Xillinc programmer. In order to get a working clock on the RTAX you have to use a HCLK clock buffer. If you don't now what that is, and how to implement Actel Macros, go read about that first.

7.1.3.1 General work flow

You develop for the RTAX2000 352CQFP. You can synthesize it and simulate it. Only if you really want to get code on the board you need to do the other steps.
So here is what you do:

1. **Libero RTAX project**

   - Open your Libero, start a new project.
   - Select the RTAX2000 352 CQFP under Accelerator.
   - Code
   - When your done synthesize it and then open the place and route designer. Give it the constraints file ".sdc" from the synthesize
   - Click the I/O Attribute editor to open the MultiViewNavigator (see: Figure 7.3), there you can set your pins for the RTAX.
   - Export the ".pdc" file. In the designer |File|Export|Constraint File
   - In order to get a working clock you have to open the timing analyze, under the point SmartTime
   - Then |Actions|Constraints|Clock and add your clock under Clock Sources
     Note: If your clock does not show up, change the pin type from "explicit clocks" to "potential clocks" and assign your clock pins.
   - Set your clock to a frequency of 40 MHz and a corresponding period of 25 ns. Exactly that! Then commit it.

2. **Netlist Converter**

   - Open the Netlist converter and give it the paths to the files ".edn" and ".pdc". The ".edn" file is generated by the synthesize and you can find it in the corresponding folder of your Libero project.
     Note: It is extremely useful to create a separate folder for netlist conversion, there you can copy your files form your project, ".edn" and ".pdc" and set the out path from the converter.

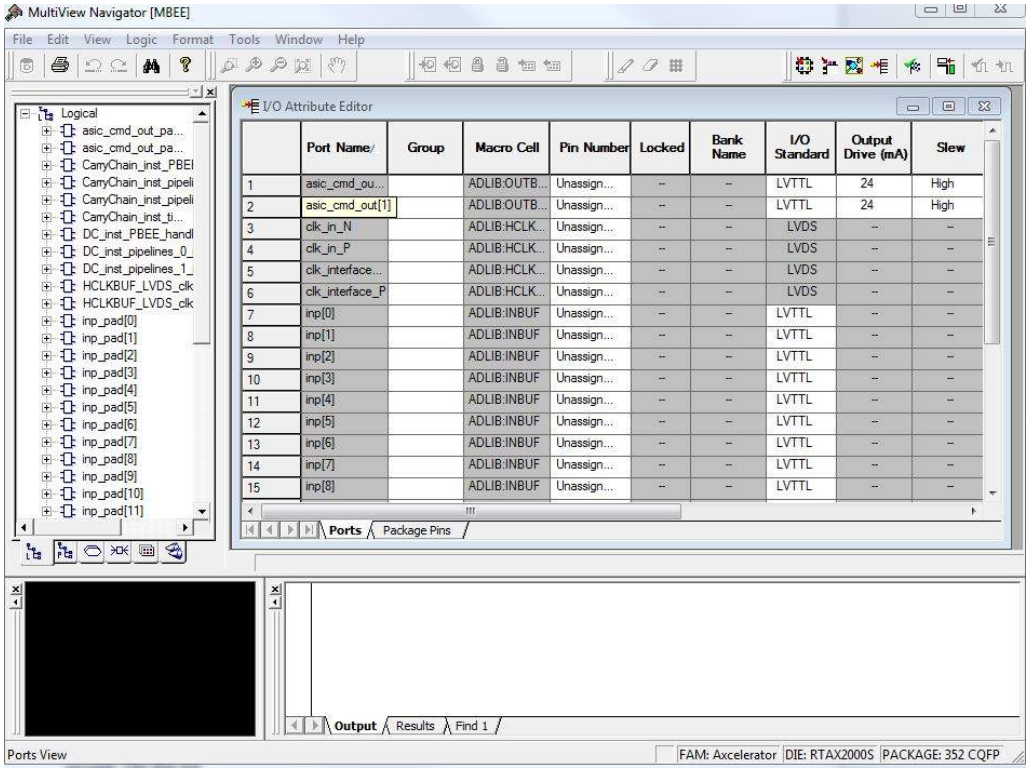


Figure 7.3: The Pins are assigned in the I/O Attribute editor. If the design contains error this window will not open.

- Select the ACT-H3K-CQ352 as the Adapter
- Check the box ”Convert Global Clocks to PDC” then run the Converter.

3. **Libero ProACSIC project**

- Open your Libero, start a new Project
- Select the A3PE3000-FG484 under ProASIC3E
- Import just the 3 files from the converter. The converted ”.edn”, ”.pdc” and also the ”.sdn” file.
- Open the Designer, check your constraints again.
- Check your clock via SmartTime, look at the frequency via |Action|Analysis|Clock Domain| and add your clock there.
- If the summary shows anything other like 40MHz, even if it is 40.1MHz or 39.99MHz, you have to set your clock constraints again to 40MHz, like you did in Point 1.
  Note: The correct clock almost never shows up here, but it is a good idea to set it before the netlist conversion anyway. It can happen sometimes that, when no clock constrain is set before the conversion (and it does not recognize the LVDS buffer), the complete clock disappears and gets replaced by another one.

- Create a programming file ".pdb" with the Designer

4. **FlashPro**

- Open FlashPro, create a new project.
- Enable single device programming
- Target is the ProASIC, A3PE3000-FG484.
- Select your ".pdc" via |configure device| browse|
- Program the board

7.2 Troubleshooting

I got no errors in the process, but the board does not what it should.
The clock, make absolutely sure it reached the constraint of 40 MHz, this cannot be overstated. Look in all protocols you can find, look at the timing analysis.

The clock doesn't show up in the pin section
Look at simplify, check the warnings, if they say anything like "clk_in is unused". Often it is just that it is not recognized as clock pin that drives anything. Implement a counter in the top level or something like that, even if it is completely unused. Most of the time that fixes it.

I get "Error: CMP010" First try everything again with a clean design, implement your VHDL there. Then make sure your clock actually drives stuff in your design, and that you have at least on net. Maybe implement a counter to make sure.
If that does not work, a good option is to write your own ".pdc" file and import it. Google for the syntax or edit an existing ".pdc" file. Once you can open the I/O constraints the error is normally gone. I still have no idea why that happens sometimes.

FlashPro tells me it "Did not exit cleanly" In 9 out of 10 cases everything will be fine, when all connections are plugged again. This is a bug caused by the programmer, mostly resulting in "exit code 10". When pluggin in and out did not solve this, a new FlashPro project has to be created, according to point 4.

7.3 Pinlist of the Aldec Adapter

Carrier-Number odd	Pin/Function	connected to		Carrier Number even	Pin/Function	
1001	n/c			1002	VCC_IO6	
1003	IO261PB6F24	J2_2	D6ATP	1004	VCC_IO6	
1005	IO261NB6F24	J2_15	D6ATN	1006	IO258PB6F24	DIAG2
1007	IO262PB6F24	J2_3	D6BTP	1008	IO258NB6F24	DIAG4
1009	IO262NB6F24	J2_16	D6BTN	1010	GND	
1011	IO265PB6F24	J2_5	D6AD1	1012	GND	
1013	IO265NB6F24	J2_18	D6AD2	1014	GND	
1015	IO279PB6F26	J2_6	D6AD3	1016	GND	
1017	IO279NB6F26	J2_19	D6AD4	1018	GND	
1019	IO280PB6F26	J2_7	D6AD5	1020	GND	
1021	IO280NB6F26	J2_20	D6AD6	1022	GND	
1023	IO281PB6F26	J2_8	D6AD7	1024	GND	
1025	IO281NB6F26	J2_21	D6BD1	1026	GND	
1027	IO282PB6F26	J2_9	D6BD2	1028	GND	
1029	IO282NB6F26	J2_22	D6BD3	1030	GND	
1031	IO284PB6F26	J2_10	D6BD4	1032	GND	
1033	IO284NB6F26	J2_23	D6BD5	1034	GND	
1035	IO285PB6F26	J2_11	D6BD6	1036	GND	
1037	IO285NB6F26	J2_24	D6BD7	1038	GND	
1039	IO286PB6F26	J2_12	D6AC	1040	GND	
1041	IO286NB6F26	J2_25	D6BC	1042	GND	
1043	IO287PB6F26		DIAG1	1044	IO296PB6F27	DIAG6
1045	IO287NB6F26		DIAG3	1046	IO296NB6F27	DIAG8
1047	IO294PB6F27		DIAG5	1048	+1V5	
1049	IO294NB6F27		DIAG7	1050	+1V5	
1051	IO300PB7F28	J1_2	D5ATP	1052	+3V3	
1053	IO300NB7F28	J1_15	D5ATN	1054	+3V3	
1055	IO311PB7F29	J1_3	D5BTP	1056	IO303NB7F28	DIAG10
1057	IO311NB7F29	J1_16	D5BTN	1058	IO303PB7F28	DIAG12
1059	IO310PB7F29	J1_5	D5AD1	1060	GND	
1061	IO310NB7F29	J1_18	D5AD2	1062	GND	
1063	IO315PB7F29	J1_6	D5AD3	1064	GND	
1065	IO315NB7F29	J1_19	D5AD4	1066	GND	
1067	IO312PB7F29	J1_7	D5AD5	1068	GND	
1069	IO312NB7F29	J1_20	D5AD6	1070	GND	
1071	IO317PB7F29	J1_8	D5AD7	1072	GND	
1073	IO317NB7F29	J1_21	D5BD1	1074	GND	
1075	IO316PB7F29	J1_9	D5BD2	1076	GND	
1077	IO316NB7F29	J1_22	D5BD3	1078	GND	
1079	IO318PB7F29	J1_10	D5BD4	1080	GND	
1081	IO318NB7F29	J1_23	D5BD5	1082	GND	
1083	IO320PB7F29	J1_11	D5BD6	1084	GND	
1085	IO320NB7F29	J1_24	D5BD7	1086	GND	
1087	IO335PB7F31	J1_12	D5AC	1088	GND	
1089	IO335NB7F31	J1_25	D5BC	1090	GND	
1091	IO334PB7F31		DIAG9	1092	GND	
1093	IO334NB7F31		DIAG11	1094	IO341PB7F31	DIAG14
1095	IO338PB7F31		DIAG13	1096	IO341NB7F31	DIAG16
1097	IO338NB7F31		DIAG15	1098	VCC_IO7	
1099	n/c			1100	VCC_IO7	

Carrier-Number odd	Pin/Function	connected to		Carrier Number even	Pin/Function
2001	GND			2002	VCC_IO0
2003	GND			2004	VCC_IO0
2005	TRST			2006	GND
2007	TMS			2008	GND
2009	TCK			2010	GND
2011	TDI			2012	GND
2013	TDO			2014	GND
2015	GND			2016	GND
2017	IO02PB0F0	LL13	D1AD1	2018	GND
2019	IO01PB0F0	LL14	D1AD2	2020	GND
2021	IO01NB0F0	LL15	D1AD3	2022	GND
2023	IO04PB0F0	LL16	D1AD4	2024	GND
2025	IO04NB0F0	LL18	D1AD5	2026	GND
2027	IO05PB0F0	LL19	D1AD6	2028	GND
2029	IO05NB0F0	LL21	D1AD7	2030	GND
2031	IO08PB0F0	LL22	D1BD1	2032	GND
2033	IO08NB0F0	LL23	D1BD2	2034	GND
2035	IO37PB0F3	LL25	D1BD3	2036	GND
2037	IO37NB0F3	LL26	D1BD4	2038	GND
2039	IO38PB0F3	LR37	D1ATP	2040	GND
2041	IO38NB0F3	LR38	D1ATN	2042	GND
2043	IO41PB0F3/HCLKAP	CLK_P0	CLK_P0	2044	GND
2045	IO41NB0F3/HCLKAN	CLK_N0	CLK_N0	2046	GND
2047	IO42PB0F3/HCLKBP	LR34	D1BTP	2048	+1V5
2049	IO42NB0F3/HCLKBN	LR35	D1BTN	2050	+1V5
2051	IO43PB1F4/HCLKCP	LR29	D2ATP	2052	+3V3
2053	IO43NB1F4/HCLKCN	LR28	D2ATN	2054	+3V3
2055	IO44PB1F4/HCLKDP	LR24	D2BTP	2056	GND
2057	IO44NB1F4/HCLKDN	LR25	D2BTN	2058	GND
2059	IO48PB1F4	LR15	D3ATP	2060	GND
2061	IO48NB1F4	LR16	D3ATN	2062	GND
2063	IO66PB1F6	LR13	D3BTP	2064	GND
2065	IO66NB1F6	LR14	D3BTN	2066	GND
2067	IO68PB1F6	LL27	D1BD5	2068	GND
2069	IO68NB1F6	LL29	D1BD6	2070	GND
2071	IO65PB1F6	LR11	D4ATP	2072	GND
2073	IO65NB1F6	LR12	D4ATN	2074	GND
2075	IO70PB1F6	LR7	D4BTP	2076	GND
2077	IO70NB1F6	LR8	D4BTN	2078	GND
2079	IO71PB1F6	LL30	D1BD7	2080	GND
2081	IO71NB1F6	LL32	D1AC	2082	GND
2083	IO69PB1F6	LL33	D1BC	2084	GND
2085	IO69NB1F6	LL34	D4BD7	2086	GND
2087	IO74PB1F6	LL36	D4AC	2088	GND
2089	IO74NB1F6	LR40	D4BC	2090	GND
2091	IO73PB1F6			2092	GND
2093	IO73NB1F6			2094	GND
2095	n/c			2096	GND
2097	n/c			2098	VCC_IO1

Carrier-Number odd	Pin/Function	connected to		Carrier Number even	Pin/Function
3001	n/c			3002	VCC_IO2
3003	IO89PB2F8	J3_2	D7ATP	3004	VCC_IO2
3005	IO89NB2F8	J3_15	D7ATN	3006	IO87PB2F8
3007	IO88PB2F8	J3_3	D7BTP	3008	IO87NB2F8
3009	IO88NB2F8	J3_16	D7BTN	3010	GND
3011	IO91PB2F8	J3_5	D7AD1	3012	GND
3013	IO91NB2F8	J3_18	D7AD2	3014	GND
3015	IO99PB2F9	J3_6	D7AD3	3016	GND
3017	IO99NB2F9	J3_19	D7AD4	3018	GND
3019	IO100PB2F9	J3_7	D7AD5	3020	GND
3021	IO100NB2F9	J3_20	D7AD6	3022	GND
3023	IO107PB2F10	J3_8	D7AD7	3024	GND
3025	IO107NB2F10	J3_21	D7BD1	3026	GND
3027	IO110PB2F10	J3_9	D7BD2	3028	GND
3029	IO110NB2F10	J3_22	D7BD3	3030	GND
3031	IO111PB2F10	J3_10	D7BD4	3032	GND
3033	IO111NB2F10	J3_23	D7BD5	3034	GND
3035	IO112PB2F10	J3_11	D7BD6	3036	GND
3037	IO112NB2F10	J3_24	D7BD7	3038	GND
3039	IO113PB2F10	J3_12	D7AC	3040	GND
3041	IO113NB2F10	J3_25	D7BC	3042	GND
3043	IO114PB2F10		LED1	3044	IO115PB2F10
3045	IO114NB2F10		LED2	3046	IO115NB2F10
3047	IO117PB2F10		LED3	3048	+1V5
3049	IO117NB2F10		LED4	3050	+1V5
3051	IO129PB3F12	J4_2	D8ATP	3052	+3V3
3053	IO129NB3F12	J4_15	D8ATN	3054	+3V3
3055	IO132PB3F12	J4_3	D8BTP	3056	IO137PB3F12
3057	IO132NB3F12	J4_16	D8BTN	3058	IO137NB3F12
3059	IO142PB3F13	J4_5	D8AD1	3060	IO139PB3F13
3061	IO142NB3F13	J4_18	D8AD2	3062	IO139NB3F13
3063	IO141PB3F13	J4_6	D8AD3	3064	GND
3065	IO141NB3F13	J4_19	D8AD4	3066	GND
3067	IO146PB3F13	J4_7	D8AD5	3068	GND
3069	IO146NB3F13	J4_20	D8AD6	3070	GND
3071	IO145PB3F13	J4_8	D8AD7	3072	GND
3073	IO145NB3F13	J4_21	D8BD1	3074	GND
3075	IO148PB3F13	J4_9	D8BD2	3076	GND
3077	IO148NB3F13	J4_22	D8BD3	3078	GND
3079	IO147PB3F13	J4_10	D8BD4	3080	GND
3081	IO147NB3F13	J4_23	D8BD5	3082	GND
3083	IO149PB3F13	J4_11	D8BD6	3084	GND
3085	IO149NB3F13	J4_24	D8BD7	3086	GND
3087	IO163PB3F15	J4_12	D8AC	3088	GND
3089	IO163NB3F15	J4_25	D8BC	3090	GND
3091	IO161PB3F15		PBEE_DATAP	3092	GND
3093	IO161NB3F15		PBEE_DATAN	3094	IO167PB3F15
3095	IO165PB3F15		PBEE_DATAS	3096	IO167NB3F15
3097	IO165NB3F15		PBEE_RST	3098	VCC_IO3
3099	n/c			3100	VCC_IO3

Carrier-Number odd	Pin/Function	connected to		Carrier Number even	Pin/Function
4001	n/c			4002	VCC_IO4
4003	n/c			4004	VCC_IO4
4005	n/c			4006	GND
4007	IO181PB4F17	RR26	D4BD6	4008	GND
4009	IO181NB4F17	RR25	D4BD5	4010	GND
4011	IO182PB4F17	RR23	D4BD4	4012	GND
4013	IO182NB4F17	RR22	D4BD3	4014	GND
4015	IO183PB4F17	RR21	D4BD2	4016	GND
4017	IO183NB4F17	RR19	D4BD1	4018	GND
4019	IO184PB4F17	RR18	D4AD7	4020	GND
4021	IO184NB4F17	RR16	D4AD6	4022	GND
4023	IO185PB4F17	RR15	D4AD5	4024	GND
4025	IO185NB4F17	RR14	D4AD4	4026	GND
4027	IO190PB4F17	RR13	D4AD3	4028	GND
4029	IO190NB4F17	RL40	D4AD2	4030	GND
4031	IO191PB4F17	RL39	D4AD1	4032	GND
4033	IO191NB4F17	RL38	D3BC	4034	GND
4035	IO192PB4F17	RL37	D3AC	4036	GND
4037	IO192NB4F17	RL35	D3BD7	4038	GND
4039	IO207PB4F19	RL34	D3BD6	4040	GND
4041	IO207NB4F19	RL33	D3BD5	4042	GND
4043	IO212PB4F19/CLKEP	RL32	D3BD4	4044	GND
4045	IO212NB4F19/CLKEN	RL31	D3BD3	4046	GND
4047	IO213PB4F19/CLKFP	RL29	D3BD2	4048	+1V5
4049	IO213NB4F19/CLKFN	RL28	D3BD1	4050	+1V5
4051	IO214PB5F20/CLKGP	RL27	D3AD7	4052	+3V3
4053	IO214NB5F20/CLKGN	RL26	D3AD6	4054	+3V3
4055	IO215PB5F20/CLKHP	RL25	D3AD5	4056	GND
4057	IO215NB5F20/CLKHN	RL24	D3AD4	4058	GND
4059	IO217PB5F20	RL22	D3AD3	4060	GND
4061	IO217NB5F20	RL21	D3AD2	4062	GND
4063	IO237PB5F22	RL20	D3AD1	4064	GND
4065	IO237NB5F22	RL19	D2BC	4066	GND
4067	IO236PB5F22	RL18	D2AC	4068	GND
4069	IO236NB5F22	RL16	D2BD7	4070	GND
4071	IO239PB5F22	RL15	D2BD6	4072	GND
4073	IO239NB5F22	RL14	D2BD5	4074	GND
4075	IO238PB5F22	RL13	D2BD4	4076	GND
4077	IO238NB5F22	RL12	D2BD3	4078	GND
4079	IO240PB5F22	RL11	D2BD2	4080	GND
4081	IO240NB5F22	RL9	D2BD1	4082	GND
4083	IO243PB5F22	RL8	D2AD7	4084	GND
4085	IO243NB5F22	RL7	D2AD6	4086	GND
4087	IO242PB5F22	RL6	D2AD5	4088	GND
4089	IO242NB5F22	RL4	D2AD4	4090	GND
4091	IO244PB5F22	RL3	D2AD3	4092	GND
4093	IO244NB5F22	RL2	D2AD2	4094	GND
4095	IO257NB6F24	RL1	D2AD1	4096	GND
4097	n/c			4098	VCC_IO5
4099	n/c			4100	VCC_IO5

List of Figures

List of Tables

Bibliography

M. A. Abramowicz and W. Kluźniak. A precise determination of black hole spin in GRO J1655-40. A&A, 374:L19–L20, August 2001. doi: 10.1051/0004-6361:20010791. 21

W. S. Adams. The Spectrum of the Companion of Sirius. PASP, 27:236, December 1915. doi: 10.1086/122440. 6

L. G. Althaus, A. H. Córsico, J. Isern, and E. García-Berro. Evolutionary and pulsational properties of white dwarf stars. A&A Rev., 18:471–566, October 2010. doi: 10.1007/s00159-010-0033-1. 6

D. Barret, S. Brandt, J.-W. den Herder, M. Feroci, M. Hernanz, M. Pohl, A. Santangelo, L. Stella, M. van der Klis, and S. Zane. The large observatory for x-ray timing, assessment study report. Technical report, European Space Agency, 12 2013. 16

V. Bhalerao and NuSTAR Team. NuSTAR: Unveiling the Hard X-ray Universe. In *American Astronomical Society Meeting Abstracts 219*, volume 219 of *American Astronomical Society Meeting Abstracts*, page 328.03, January 2012. 9

Taylor L. Booth. Sequential machines and automata theory, 1967. URL `http://opac.inria.fr/record=b1082797`. 52

S. Chandrasekhar. The Maximum Mass of Ideal White Dwarfs. ApJ, 74:81, July 1931. doi: 10.1086/143324. 6

A. C. Fabian, K. Iwasawa, C. S. Reynolds, and A. J. Young. Broad Iron Lines in Active Galactic Nuclei. PASP, 112:1145–1161, September 2000. doi: 10.1086/316610. 21

G. W. Fraser, J. D. Carpenter, D. A. Rothery, J. F. Pearson, A. Martindale, J. Huovelin, J. Treis, M. Anand, M. Anttila, M. Ashcroft, J. Benkoff, P. Bland, A. Bowyer, A. Bradley, J. Bridges, C. Brown, C. Bulloch, E. J. Bunce, U. Christensen, M. Evans, R. Fairbend, M. Feasey, F. Giannini, S. Hermann, M. Hesse, M. Hilchenbach, T. Jorden, K. Joy, M. Kaipiainen, I. Kitchingman, P. Lechner, G. Lutz, A. Malkki, K. Muinonen, J. Näränen, P. Portin, M. Prydderch, J. S. Juan, E. Sclater, E. Schyns, T. J. Stevenson, L. Strüder, M. Syrjasuo, D. Talboys, P. Thomas, C. Whitford, and S. Whitehead. The mercury imaging X-ray spectrometer (MIXS) on bepicolombo. Planet. Space Sci., 58:79–95, January 2010. doi: 10.1016/j.pss.2009.05.004. 32

H. Friedman, S. W. Lichtman, and E. T. Byram. Photon counter measurements of solar x-rays and extreme ultraviolet light. *Phys. Rev.*, 83:1025–1030, Sep 1951. doi: 10.1103/PhysRev.83.1025. URL http://link.aps.org/doi/10.1103/PhysRev.83.1025. 3

R. Giacconi, H. Gursky, F. R. Paolini, and B. B. Rossi. Evidence for x Rays From Sources Outside the Solar System. *Physical Review Letters*, 9:439–443, December 1962. doi: 10.1103/PhysRevLett.9.439. 5

T. Gold. Rotating Neutron Stars as the Origin of the Pulsating Radio Sources. Nature, 218:731–732, May 1968. doi: 10.1038/218731a0. 7

P. Haensel, J. L. Zdunik, M. Bejger, and J. M. Lattimer. Keplerian frequency of uniformly rotating neutron stars and strange stars. A&A, 502:605–610, August 2009. doi: 10.1051/0004-6361/200811605. 19

A. K. Harding. The neutron star zoo. *Frontiers of Physics*, 8:679–692, December 2013. doi: 10.1007/s11467-013-0285-0. 8

A. Heger, C. L. Fryer, S. E. Woosley, N. Langer, and D. H. Hartmann. How Massive Single Stars End Their Life. ApJ, 591:288–300, July 2003. doi: 10.1086/375341. 7

B. L. Henke, E. M. Gullikson, and J. C. Davis. X-Ray Interactions: Photoabsorption, Scattering, Transmission, and Reflection at E = 50-30,000 eV, Z = 1-92. *Atomic Data and Nuclear Data Tables*, 54:181–342, July 1993. doi: 10.1006/adnd.1993.1013. 2

J. W. T. Hessels, S. M. Ransom, I. H. Stairs, P. C. C. Freire, V. M. Kaspi, and F. Camilo. A Radio Pulsar Spinning at 716 Hz. *Science*, 311:1901–1904, March 2006. doi: 10.1126/science.1123430. 19

A. Hewish, S. J. Bell, J. D. H. Pilkington, P. F. Scott, and R. A. Collins. Observation of a Rapidly Pulsating Radio Source. Nature, 217:709–713, February 1968. doi: 10.1038/217709a0. 7

S. A. Hughes. Trust but verify: The case for astrophysical black holes. *ArXiv High Energy Physics - Phenomenology e-prints*, November 2005. 9

S. O. Kepler, S. J. Kleinman, A. Nitta, D. Koester, B. G. Castanheira, O. Giovannini, and L. Althaus. The White Dwarf Mass Distribution. In R. Napiwotzki and M. R. Burleigh, editors, *15th European Workshop on White Dwarfs*, volume 372 of *Astronomical Society of the Pacific Conference Series*, page 35, September 2007. 6

A. R. King and U. Kolb. The evolution of black hole mass and angular momentum. MNRAS, 305:654–660, May 1999. doi: 10.1046/j.1365-8711.1999.02482.x. 17

J. M. Lattimer. How Neutron Star Radius Measurements Can Constrain the Nuclear Equation of State. In *American Astronomical Society Meeting Abstracts*, volume 33 of *Bulletin of the American Astronomical Society*, page 105.01, December 2001. 17

J. M. Lattimer and M. Prakash. The Physics of Neutron Stars. *Science*, 304:536–542, April 2004. doi: 10.1126/science.1090720. 18

J. M. Lattimer and M. Prakash. Neutron star observations: Prognosis for equation of state constraints. Phys. Rep., 442:109–165, April 2007. doi: 10.1016/j.physrep.2007.02.003. 8

Q. Z. Liu, J. van Paradijs, and E. P. J. van den Heuvel. Catalogue of high-mass X-ray binaries in the Galaxy (4th edition). A&A, 455:1165–1168, September 2006. doi: 10.1051/0004-6361:20064987. 12

K. H. Lo, M. C. Miller, S. Bhattacharyya, and F. K. Lamb. Determining Neutron Star Masses and Radii Using Energy-resolved Waveforms of X-Ray Burst Oscillations. ApJ, 776:19, October 2013. doi: 10.1088/0004-637X/776/1/19. 18

M. S. Longair. *High Energy Astrophysics*. February 2011. 2

J. Madej, M. Należyty, and L. G. Althaus. Mass distribution of DA white dwarfs in the First Data Release of the Sloan Digital Sky Survey. A&A, 419:L5–L8, May 2004. doi: 10.1051/0004-6361:20040120. 6

J. E. McClintock and R. A. Remillard. *Black hole binaries*, pages 157–213. April 2006. 9

I. F. Mirabel and I. Rodrigues. The origin of Scorpius X-1. A&A, 398:L25–L28, February 2003. doi: 10.1051/0004-6361:20021767. 12

Ramesh Narayan. Black holes in astrophysics. *New Journal of Physics*, 7(1):199, 2005. URL `http://stacks.iop.org/1367-2630/7/i=1/a=199`. 9

J. R. Oppenheimer and G. M. Volkoff. On Massive Neutron Cores. *Physical Review*, 55:374–381, February 1939. doi: 10.1103/PhysRev.55.374. 7

F. Özel and D. Psaltis. Reconstructing the neutron-star equation of state from astrophysical measurements. Phys. Rev. D, 80(10):103003, November 2009. doi: 10.1103/PhysRevD.80.103003. 17

A. Rashevsky, V. Bonvicini, P. Burger, S. Piano, C. Piemonte, and A. Vacchi. Large area silicon drift detector for the ALICE experiment. *Nuclear Instruments and Methods in Physics Research A*, 485:54–60, June 2002. doi: 10.1016/S0168-9002(02)00531-4. 30

C. Ritossa, E. Garcia-Berro, and I. Iben, Jr. On the Evolution of Stars That Form Electron-degenerate Cores Processed by Carbon Burning. II. Isotope Abundances and Thermal Pulses in a 10 M_{sun} Model with an ONe Core and Applications to

Long-Period Variables, Classical Novae, and Accretion-induced Collapse. ApJ, 460:489, March 1996. doi: 10.1086/176987. 6

H. N. Russell. Relations Between the Spectra and Other Characteristics of the Stars. *Popular Astronomy*, 22:275–294, May 1914. 6

L. Samuelsson and N. Andersson. Neutron star asteroseismology. Axial crust oscillations in the Cowling approximation. MNRAS, 374:256–268, January 2007. doi: 10.1111/j.1365-2966.2006.11147.x. 20

A. Sandage, P. Osmer, R. Giacconi, P. Gorenstein, H. Gursky, J. Waters, H. Bradt, G. Garmire, B. V. Sreekantan, M. Oda, K. Osawa, and J. Jugaku. On the optical identification of SCO X-1. ApJ, 146:316, October 1966. doi: 10.1086/148892. 5

R. Schödel, T. Ott, R. Genzel, R. Hofmann, M. Lehnert, A. Eckart, N. Mouawad, T. Alexander, M. J. Reid, R. Lenzen, M. Hartung, F. Lacombe, D. Rouan, E. Gendron, G. Rousset, A.-M. Lagrange, W. Brandner, N. Ageorges, C. Lidman, A. F. M. Moorwood, J. Spyromilio, N. Hubin, and K. M. Menten. A star in a 15.2-year orbit around the supermassive black hole at the centre of the Milky Way. Nature, 419:694–696, October 2002. doi: 10.1038/nature01121. 10

E. Schreier, R. Levinson, H. Gursky, E. Kellogg, H. Tananbaum, and R. Giacconi. Evidence for the Binary Nature of Centaurus X-3 from UHURU X-Ray Observations. ApJ, 172:L79, March 1972. doi: 10.1086/180896. 12

K. Schwarzschild. On the Gravitational Field of a Mass Point According to Einstein's Theory. *Abh. Konigl. Preuss. Akad. Wissenschaften Jahre 1906,92, Berlin,1907*, 1916:189–196, 1916. 9

F. D. Seward and P. A. Charles. *Exploring the X-ray Universe*. August 2010. 2

I. S. Shklovsky. On the Nature of the Source of X-Ray Emission of SCO XR-1. ApJ, 148:L1, April 1967. doi: 10.1086/180001. 10

G. K. Skinner. Coding (and Decoding) Coded Mask Telescopes. *Experimental Astronomy*, 6:1–7, December 1995. doi: 10.1007/BF00419252. 28

D. Steeghs and J. Casares. The Mass Donor of Scorpius X-1 Revealed. ApJ, 568: 273–278, March 2002. doi: 10.1086/339224. 12

L. Stella, M. Vietri, and S. M. Morsink. Correlations in the Quasi-periodic Oscillation Frequencies of Low-Mass X-Ray Binaries and the Relativistic Precession Model. ApJ, 524:L63–L66, October 1999. doi: 10.1086/312291. 21

T. E. Strohmayer and A. L. Watts. Discovery of Fast X-Ray Oscillations during the 1998 Giant Flare from SGR 1900+14. ApJ, 632:L111–L114, October 2005. doi: 10.1086/497911. 20

S. Suchy, P. Uter, C. Tenzer, A. Santangelo, A. Argan, M. Feroci, T. E. Kennedy, P. J. Smith, D. Walton, S. Zane, J. Portell, and E. García-Berro. The on-board data handling concept for the LOFT large area detector. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 8443 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, September 2012. doi: 10.1117/12.925663. 75

V. Suleimanov, J. Poutanen, and K. Werner. X-ray bursting neutron star atmosphere models using an exact relativistic kinetic equation for Compton scattering. A&A, 545:A120, September 2012. doi: 10.1051/0004-6361/201219480. 18

Y. Tanaka, K. Nandra, A. C. Fabian, H. Inoue, C. Otani, T. Dotani, K. Hayashida, K. Iwasawa, T. Kii, H. Kunieda, F. Makino, and M. Matsuoka. Gravitationally redshifted emission implying an accretion disk and massive black hole in the active galaxy MCG-6-30-15. Nature, 375:659–661, June 1995. doi: 10.1038/375659a0. 21

S. E. Thorsett and D. Chakrabarty. Neutron Star Mass Measurements. I. Radio Pulsars. ApJ, 512:288–299, February 1999. doi: 10.1086/306742. 7

Pascal Uter. Development of the Module Back End Electronics for the Large Observatory For X-ray Timing, 2 2013. 33, 44

M. van der Klis. Millisecond Oscillations in X-ray Binaries. ARA&A, 38:717–760, 2000. doi: 10.1146/annurev.astro.38.1.717. 20

W. Voges, B. Aschenbach, T. Boller, H. Bräuninger, U. Briel, W. Burkert, K. Dennerl, J. Englhauser, R. Gruber, F. Haberl, G. Hartner, G. Hasinger, M. Kürster, E. Pfeffermann, W. Pietsch, P. Predehl, C. Rosso, J. H. M. M. Schmitt, J. Trümper, and H. U. Zimmermann. The ROSAT all-sky survey bright source catalogue. A&A, 349:389–405, September 1999. 13

M. Volonteri and J. Bellovary. Black holes in the early Universe. *Reports on Progress in Physics*, 75(12):124901, December 2012. doi: 10.1088/0034-4885/75/12/124901. 17, 20

A. Y. Potekhin. The physics of neutron stars. *Physics Uspekhi*, 53:1235–1256, December 2010. doi: 10.3367/UFNe.0180.201012c.1279. 7

Chapter 8

Acknowledgments

Without the continuous support of several people this thesis would not have been possible. Interesting and friendly people at the IAAT made this year working on the MBEE a great experience. I like to thank all of you and a few others in particular.

- **Prof. Dr. Santangelo** for his assignment of this thesis (and proofreading) and his always friendly attitude which provided always many laughter's and smiles not only at our group seminars.
- **Dr. Jan-Christoph Tenzer** for his overall excellent monitoring of my work. For his somewhat miraculous ability to remember every little detail related to my work. For his friendly attitude and his always open door. (And for competing in the pushup challenge)
- **Henning Wende** for basically teaching me nearly everything i know about VHDL. For forcing me to use git, forcing me to _underscore and always taking the time to think about problems that occurred on the way and providing solutions for them.
- **Felix Eisenkolb** for coffee, for fun times, for overall support, for discussions, for python, for proofreading, for explaining vim (all the time), for doing pushups. I know who Felix Strübing is!
- **Thomas Schanz** for providing useful tips for my work as well as this thesis.
- **Janina Richter** for supporting me through my studies and for great times.
- **My family and friends**
  I want to thank my family for supporting me through my studies. And i want to thank my many friends for helping me along the way, you know who you are. I had a broken knee, a broken elbow, a broken nose and a heavy lung inflammation, but hell these were the best years of my life. Thank you all very much.