Dissertation
submitted to the
Combined Faculties for the Natural Sciences and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Put forward by
Dipl.-Inf. Tim Armbruster
born in Mannheim, Germany
Oral examination: May 3rd, 2013

SPADIC - a Self-Triggered Detector Readout ASIC with Multi-Channel Amplification and Digitization

Referees: Prof. Dr. Peter Fischer
Prof. Dr. Johanna Stachel

Zusammenfassung

Ziel dieser Dissertation war die Entwicklung eines Mehrkanal-Mischsignal-ASICs für die Auslese von Detektoren. Die vorliegende Arbeit beschreibt den gesamten Entwicklungsprozess, welcher mit der unspezifischen Aufgabenstellung des Entwurfs eines Auslesechips für einen der Teildetektoren von CBM/FAIR beginnt und der mit der neusten und tatsächlich realisierten system-on-a-chip Lösung mit dem Namen SPADIC endet, deren Haupteinsatzzweck die Auslese des zukünftigen CBM-TRD ist. Diese Arbeit umfasst den von Grunde auf begonnenen Entwicklungsprozess von insgesamt 6 ASIC Prototypen und 10 PCB Ausleseaufbauten sowie die Entwicklung diverser Software- und Firmwarekomponenten, die Charakterisierung der entworfenen ASICs, die Entwicklung des SPADIC-Konzeptes, den Entwurf der SPADIC Webseite und nicht zuletzt die Zusammenarbeit mit den TRD-Physikern, mit dem erreichten Ziel mithilfe verschiedener SPADICs Kammer-Prototypen während CERN-Testbeams oder im Labor auszulesen. Neben den Beschreibungen der wichtigsten Chipdetails und den dazugehörigen theoretischen Analysen, enthält diese Abhandlung auch eine allgemeine, auf einem Niveau für Ingenieure gehaltene, Einführung in die Varianten von Detektoren und in die Detektorphysik, mit dem Ziel diese technische Arbeit in ihren physikalischen Kontext einzubetten. Das effektive Resultat dieser Dissertation ist der selbstgetriggerte 32-Kanal-Ladungspuls-Verstärker und Digitalisierungs-Chip SPADIC 1.0.

Summary

The intention of this dissertation was the development of a multi-channel mixed-signal detector readout ASIC. This paper describes the whole design process that starts with the vague requirement for a readout chip for some CBM/FAIR sub-detector and that ends with the latest and actually realized system on a chip solution called SPADIC, which is mainly intended to read out the future CBM-TRD. This work comprises the design from scratch of 6 ASIC prototypes and 10 PCB readout setups as well as the development of various software and firmware components, the characterization of the designed ASICs, the development of the SPADIC concept, the design of the SPADIC website, and not at least the collaboration with the TRD physicists with the achieved goal to read out signals of chamber prototypes using different SPADICs during CERN beam-times or in the laboratory. Besides the descriptions of the most important chip details and the corresponding theoretical analyses, an overall introduction into detectors and detector physics – written on a level for engineers – is given in this paper in order to embed this technical work into its physical context. The effective output of this dissertation is the self-triggered 32-channel charge pulse amplification and digitization chip SPADIC 1.0.

Contents

CHAPTER 1

Introduction

Powered by technology all fields of natural science have developed explosively during the last few hundred years. And similarly, driven by the increasing knowledge in natural science nearly all areas of technology have inconceivably evolved. The cliché of a genius sitting alone (and usually lonely as well) in his laboratory while changing the world with his discoveries derived from structurally simple but pioneering experiments should probably have been outdated long ago. Instead, the corresponding modern stereotype would be international collaborations or enterprises building extremely expensive and complex high technology machines that, after months or years of building, operation, and analysis, eventually lead to seemingly smaller improvements of very abstract formulas or theories. But of course the scientific reality does not deserve to be simplified in either way, even though a grain of truth is probably contained in both pictures. However, one can say for sure that technology is not only a modern science of its own, but also a key parameter for success in nearly all fields of modern research.

An outstanding example of the necessary merge of natural science and technology are the various existing and future particle accelerator physics experiments that are performed to answer some of the most fundamental questions of nature. One could say that well founded understanding only reaches as far as the latest technologies can carry. From an abstract point of view particle accelerators are macroscopic high technology machines searching for natural reality on the smallest even thinkable scale. And in fact the developing and building of modern accelerator facilities and experiments probably takes at least as much time, know-how, manpower, and money as the later operation and research itself. And, moreover, this imbalance between technological input and physical output will probably increase even further in the future. But because the value of new fundamental physical output is generally not assessable and its benefit for mankind might be priceless, the increasing effort is already scientifically justified.

Modern particle accelerator experiments, which, so to speak, are digital cameras scaled to the size of houses – including a respectively scaled complexity and an unchanged density of

integration –, comprise of numberless sub-systems, each ideally being both an independent machine of its own and an indispensable element of the whole construction. Especially the required focus on the local and the global context at the same time makes the design of each single component as well as the building of the complete experiment a very difficult and elaborate task. The design of a part of the experiment can be descriptively compared to the cutting of a brick for constructing a bridge between physics and technology.

Now, building such a brick has actually been the purpose of this work. To be more specific, the challenge was to design an autonomous readout system on a single silicon chip, which is able to sense and process small electrical signals gathered by dedicated particle detectors operated within an accelerator experiment. In the present case the main application of the chip is intended to be the readout of the transition radiation sub-detector (TRD) that will be part of the compressed baryonic matter experiment (CBM) at the future accelerator facility for anti-proton and ion research (FAIR) nearby GSI Helmholtzzentrum für Schwerionenforschung in Darmstadt (Germany).

The present paper pursues two main objectives: On the one hand the global context of the actually developed readout chips is explained, and on the other hand all essential ideas, details, and results of the realized prototypes are summarized. Whereas the former requires to coarsely introduce into some of the physical basics, the latter leads to a technical discussion of all crucial chip aspects.

The introduction to the physical context – which has been written on a level that should be suitable for engineers – comprises the two chapters "2. Detectors and Detector Physics" and "3. The CBM Experiment". Whereas chapter 2 basically is a short and general outline of detectors and detector physics and has no direct reference to the actual task (although it is frequently referenced), chapter 3 introduces the concrete physical context of the CBM experiment, which indirectly causes the various requirements and constraints for the desired readout chip.

The other chapters in contrast focus on all technical aspects of the chip development and the achieved results, and thus on the productive content of this work. First, the general chip concept is explained (chapter "4. The Chip Concept"), then, all technical aspects of the analog components (chapter "5. The Analog Part") as well as all essential issues of the digital components (chapter "6. The Digital Part") are given. Afterwards, most details and results that have been gathered with the various readout systems based on the several meanwhile realized chip iterations are discussed (chapter "7. The Readout Systems").

Finally, chapter "8. Summary and Outlook" closes this document.

CHAPTER 2

Detectors and Detector Physics

This chapter summarizes selected parts of modern knowledge in physics and instrumentation that are essential to know for everybody working in the field of detector and electronics development for high energy experiments. The following overview makes no claim to be complete, but instead tries to lead the reader quickly through the most important topics. It is intended for non-physicists (e.g. engineers) who are interested in becoming more acquainted with the background of detector physics. It is explicitly aspired not to spend too much time with details on formula, which are best described in modern text books anyway, but to rather give the reader a brief outline of the basic principles and most important interrelations.

If not noted otherwise, the subsequent description is mainly based on the sources [34], [20], [57] and [19].

2.1. Effects of Particles Crossing Matter

In the field of accelerator physics and detector development in particular, the multiple physics effects occurring if fast moving particles (also called projectiles in the following context) cross matter are extremely important. Although this might be obvious for the experienced reader, it is worth mentioning that these effects already make up a very large part of knowledge that is required for a basic understanding of how the various types of detectors work and what purpose they can be used for.

A common property of all relevant effects – although they are quite different in detail – is that the incident particle loses a certain fraction or even all its energy each time the effect occurs. The multiple types of effects hereby fundamentally depend on the particle energy, the particle type, and the characteristics of the crossed matter (material, state, geometry, ...), and each of them is of course based on one of the three fundamental interactions:

electromagnetic, strong or weak (gravitation plays a minor role on this scale[1]). In most cases the loss of energy of the projectile can be exploited by sensors or detectors, which usually collect or amplify (or both) the lost energies and in doing so are able to transfer the microscopic information to a macroscopic scale.

2.1.1. Energy Loss of Heavy Charged Particles

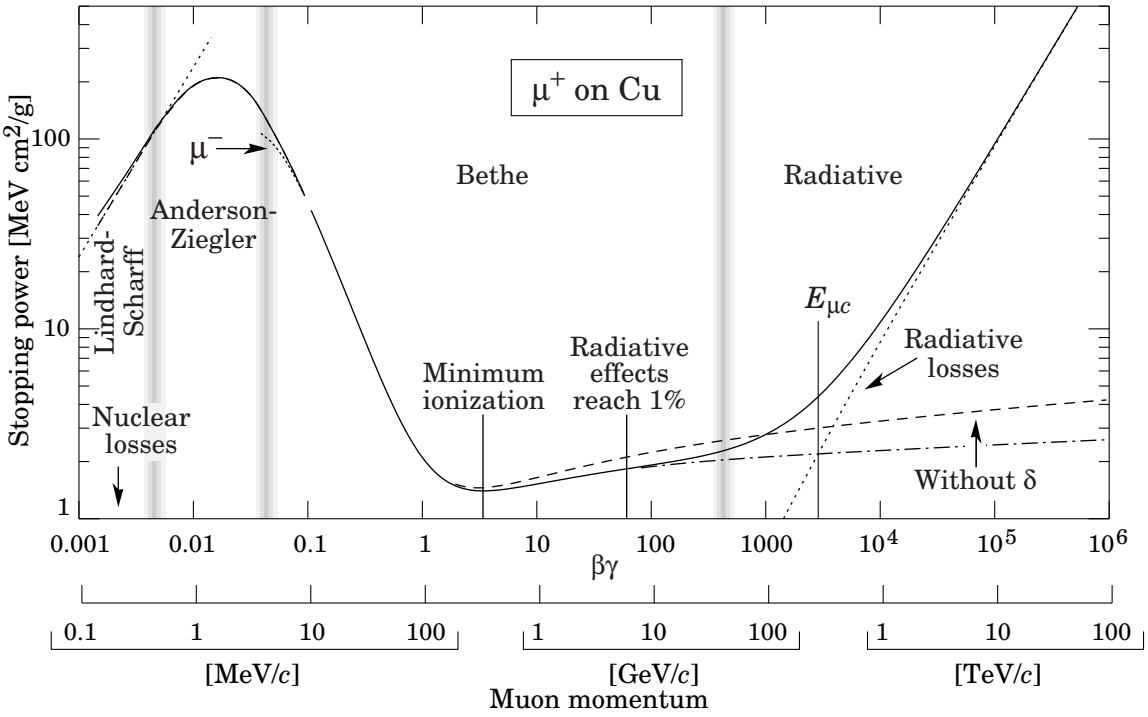


Figure 2.1.: The commonly used plot of stopping power (mean rate of energy loss) as a function of $\beta\gamma$ (or momentum) of positive muons traversing copper. The solid line, which depicts the total stopping power over fairly a huge range of energy, crosses several boundaries (vertical gray lines) that separate different regions of physical interpretation [34].

Figure 2.1 shows the mean loss of energy of a muon (which stands exemplarily for a charged particle of medium mass) in dependency on the muon momentum (or $\beta\gamma$ with Lorentz factor γ and ratio β of velocity v to the speed of light c) over many orders of magnitude. Formally, the mean loss off energy is usually expressed as $-\langle\frac{\mathrm{d}E}{\mathrm{d}x}\rangle$ describing the mean energy loss of the incident particle per traversed path $\mathrm{d}x$. The effective mean energy loss, shown in the graph as a solid line, is actually the sum of very different contributions due to very different effects that all occur in different regions of energy.

[1]The strength of gravitation is in the order $10^{25} - -10^{38}$ times smaller compared to the three other fundamental interactions.

2.1.1.1. Medium Energy Range ($0.1 \leq \beta\gamma \leq 1000$)

The probably most important energy range (because it both spreads over several orders of magnitude and includes energies that are most common in nearly all practical detector applications) of roughly $0.1 \leq \beta\gamma \leq 1000$ (for intermediate-Z materials) can be described very accurately by the popular **Bethe formula** (or Bethe-Bloch formula) [16]. The physical model underlying the Bethe formula is that of a charged, heavy and moderately relativistic particle ionizing atoms along its straight trajectory through a volume of matter. The assumption made here is that the charged projectile electromagnetically knocks out bound electrons of their atoms while it just slightly changes its momentum and direction. The amount of energy transferred from the charged particle to a bound electron in such a collision directly depends on the distance between projectile and electron[1] and must always be equal to or larger than the specific excitation energy of the passed atom. The Bethe formula is derived by calculating the mean energy transfer from the projectile to the medium after multiple collisions in a medium with evenly distributed atoms and directly depends on the average electron density of the medium.

The energy range of the Bethe equation can be further separated into three regions: first, for roughly $\beta\gamma \leq 1$, where the velocity $v = \beta c$ still significantly increases with the energy, the stopping power falls proportionally to $\beta^{-\frac{5}{3}}$. Second, at $\beta\gamma \approx 3.0 - -3.5$, a minimum is reached (projectiles within the energy range of this relatively broad minimum are called **minimum ionizing particles**, or **MIPs**). And third, going further to larger values of $\beta\gamma$, the stopping power again slightly increases due to a relativistic extend of the electric fields, now logarithmically with β, the so-called **relativistic rise**.

Whereas the basic formula of Bethe is very accurate at low and medium energies (starting from $\beta\gamma \geq 0.1$), the stopping power differs considerably from actually measured values at higher energies ($\beta\gamma \approx 5$), where the relativistic rise is already contributing intensively. In this energy region, the so-called **density effect** begins to play an important role, which is based on the observation that the traversed media starts to become polarized. The density effect effectively causes the relativistic extend of the electric fields to saturate, and hence stops the logarithmic rise, until eventually the so-called **Fermi plateau** is reached. The Bethe formula with (solid line) and without (dashed line, "without δ") density correction is sketched in graph 2.1.

A key parameter of the Bethe formula, which includes the respective characteristics of the media and the projectile, is the **mean excitation energy** I. Because at the present state of the art no overall theory exists (except for simple atomic gases) and one rather relies on Lorentz transformed and hence relativistically extended measurements, "the determination of the mean excitation energy is the principal non-trivial task when evaluating the Bethe formula" [62].

For practical reasons, it is very important to emphasize that the Bethe formula only describes the mean energy loss, but tells little about the actual energy loss probability distribution. That distribution becomes particularly interesting though, if one considers a single

[1] This formally only works when adding the very innovative and far-reaching assumption (already recognized by Bohr, solved later by Bethe) that the possible energy transfer is limited to both minimal and maximal values. A classical derivation can be found for instance in [19].

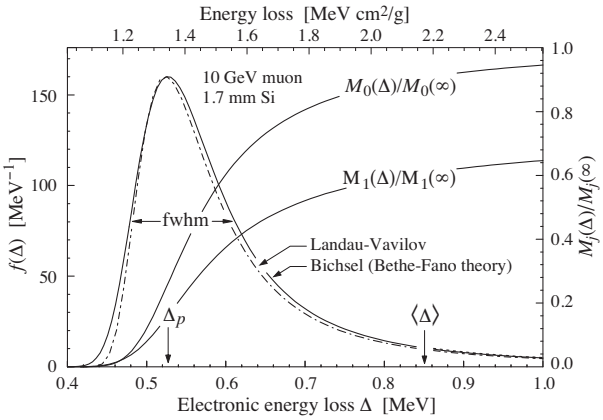


Figure 2.2.: Landau distribution of 10 GeV muons traversing thick silicon of 1.7 mm [34].

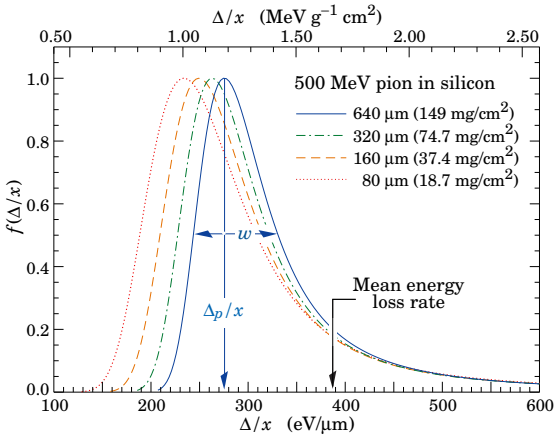


Figure 2.3.: Distributions similar to Landau of 500 MeV pions traversing thin silicon [34].

particle crossing a single piece of matter. Fortunately, this case is well covered by the **Landau distribution** [44] (or Landau-Vavilov distribution) shown exemplarily in plot 2.2. The not Gaussian but rather skewed shape of the Landau distribution results from so-called **delta electrons**, which are knocked-out electrons that carry a significant amount of kinetic energy (much larger than the typical excitation energy) as the result of an exceptionally intense interaction. Although the relative number of produced delta electrons is rather small, they heavily contribute to the actual energy loss, and consequently the mean energy loss (Bethe formula) significantly differs from the most probable energy loss. For very thin absorbers the Landau formula fails to predict the energy loss precisely and rather underestimates the full width at half maximum (FWHM) [14] [18]. As an example of the energy loss in a thin absorber, energy loss probability distributions of Si detectors with different thicknesses are shown in Fig. 2.3.

With each electromagnetic interaction, the charged projectile does not only lose a fraction of energy, but also gets slightly deflected. In doing so it perpetually changes its direction – but usually by small angles only. This effect, called **multiple (Coulomb) scattering**, roughly has a Gaussian angular distribution for smaller angles, and can be well described similar to Rutherford scattering for larger angles [17]. Since multiple scattering effectively blurs the particle track, it potentially affects the position resolution of detectors.

The Bethe formula directly tells that charged heavy particles above $\beta\gamma \approx 1$ lose much less energy (at average) while traversing matter than particles with lower energies (see again Fig. 2.1, logarithmic scale). Therefore one can easily imagine that a projectile typically almost abruptly stops in a medium, if its kinetic energy falls below a certain value. For that reason, and even though the energy loss is a mere statistical process, heavy charged projectiles of the same type and energy usually stick at a very predictable depth. At the same time, in the vicinity of that characteristic depth, the main fraction of energy is deposited. Hence, if one plots the deposited energy as a function of depth, one finds a clear maximum – the **Bragg peak** – nearby the average depth of penetration.

2.1.1.2. Low Energy Range ($\beta\gamma \leq 0.1$)

At very low energies, shell corrections must be considered in order to make the Bethe formula still accurate down to $\beta = 0.05$ (e.g. Barkas, Bloch or Born corrections), but at even lower values ($0.01 \leq \beta \leq 0.05$) no satisfactory theory yet exists. Here one relies on fitting formulas of measured data (e.g. Andersen Ziegler or Lindhard [8] [46]). The low-energy part of Fig. 2.1 was adapted accordingly.

2.1.1.3. High Energy Range ($\beta\gamma \geq 1000$)

At ultra high energies and starting with the lightest charged particles at first (e.g. at several hundred GeV for muons traversing iron) so-called **radiative losses**, which are mainly due to the two effects bremsstrahlung and pair production (see also 2.1.2.1 and 2.1.2.2) start to contribute to the total $-\langle\frac{dE}{dx}\rangle$ and eventually even dominate. Although different definitions exist, the amount of energy at which the loss due to radiative effects roughly equals the loss due to ionization (Bethe formula) is named **critical energy**. In contrast to ionization, the cross-sections of radiative effects at very large energies are typically small, show hard spectra and have large energy fluctuations. Therefore the energy loss can not be any longer described as an uniform and continuous process. Figure 2.1 demonstrates how radiative losses of muons at ultra high energies clearly dominate the stopping power.

At energies in the order of 100 GeV, also hadronic structure effects begin to be energetically possible. But normally their contribution to the total energy loss stays negligible below the loss due to ionization, at least until radiative effects become dominant anyway [41].

2.1.2. Interactions of Photons and Electrons Traversing Matter

Even though it is a charged particle, the previous summary of energy loss fails to describe the behavior of the extremely lightweight electron (and also the positron) accurately. And, moreover, another very important particle, which can also lose fractions or even all its energy electromagnetically, although it carries no charge at all, it even has not yet been mentioned: the photon.

Hence, subsequently the effects leading to energy loss of electrons, positrons and photons are summarized in more detail.

2.1.2.1. Electrons and Positrons

The total loss of energy of electrons and protons is dominated by the two processes ionization and bremsstrahlung.

At lower energies, similar to heavier charged particles, the total stopping power of electrons and positrons is mostly due to ionization. It however (and in contrast to heavier projectiles) takes place a little differently in detail, due to the kinematics, the spins of the electrons, and the fundamental principle of identity between incident and ionized electron. In the Bethe formula, this difference can be expressed by using proper mean excitation energies. Large sets of tables with measured mean excitation energies for electrons and positrons, and for different media were published (e.g. [62] or [63]). Besides ionization

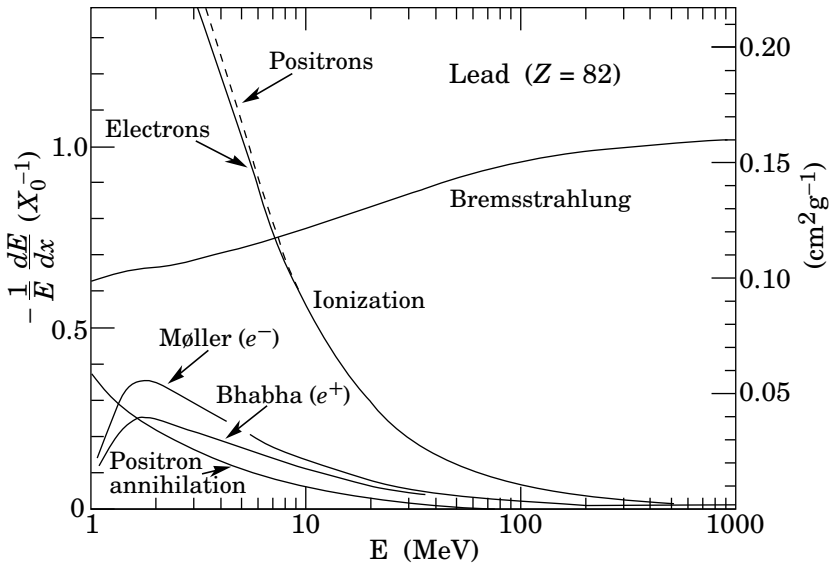


Figure 2.4.: Fractional energy loss per radiation length in lead as a function of electron (or positron) energy [34].

(and still at lower energies) also some minor effects add slightly to the total $-\langle\frac{dE}{dx}\rangle$ of electrons and positrons. Those are basically Møller scattering, Bhabha scattering and positron annihilation.

Due to their small masses electrons and positrons – much earlier than heavier charged particles (as mentioned earlier) – start to emit **bremsstrahlung** when accelerated or decelerated[1]. The latter is the case, if an electron (or positron) passes trough matter and in doing so penetrates the Coulomb field of the medium. With increasing energies, the energy loss due to bremsstrahlung increases and eventually catches up with the decreasing energy loss due to ionization. This happens at a critical energy of about 500 MeV/Z (with Z the atomic number of the material).

Typically the energy loss of electrons and positrons is expressed in units of **radiation length** X_0, normally measured in $g\ cm^{-2}$. One radiation length can be roughly described as the amount of matter traversed by a high-energy electron (or positron) until, due to bremsstrahlung, only $\frac{1}{e}$ of its primary energy remains. Using that convention, Figure 2.4 summarizes the different types of fractional energy losses of electrons and positrons exemplarily traversing lead.

2.1.2.2. Photons

The electromagnetic energy loss of photons in the presence of matter must be considered in three different regions of energy:

At lower energies, the **photoelectric effect** dominates, which causes the photon to disappear completely when ionizing an atom. Therefore quantitative treatments of the effect are

[1] The amount of radiated energy via bremsstrahlung is proportional to $\frac{1}{m^2}$ (with m the rest mass of the projectile).

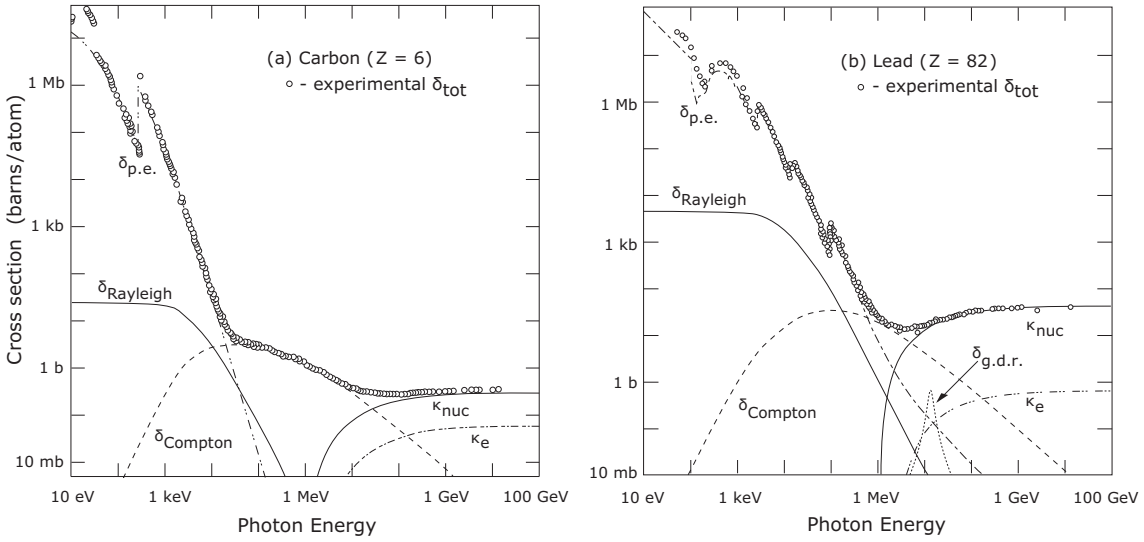


Figure 2.5.: Total cross-section as the result of different electromagnetic effects of photons in light carbon (left) and heavy lead (right). The plot was taken from [34], but the layout rearranged.

ambiguous to describe via a mean energy loss, and are therefore also practically expressed in units of radiation length X_0. The photoelectric cross-section is proportional to $\frac{Z^5}{E^3}$ and therefore typically plays a role only at photon energies somewhere below 1 MeV. Also at lower energies **Rayleigh scattering** occurs, but is due to its small cross-section (compared to the photoelectric effect) of little account in this context.

Within the energy range between 100 keV and 10 MeV, **Compton scattering** takes place and eventually exceeds the decreasing photoelectric cross-section. It is characterized by the fact that the photon is not absorbed when ionizing an atom, but that it rather loses only a fraction of its initial energy (which corresponds to a sudden increase of wavelength). Additionally the photon is deflected at a characteristic angle. More specific, the amount of energy transferred during an interaction increases with the angle of deflection Θ ($E_{\text{trans}} \sim (1 - \cos\Theta)$) and becomes maximal if the photon is completely reflected (at an angle of 180°). Because this maximum is absolute, energy spectra of the ionized electrons show a very sharp edge at some maximum energy, the so-called **Compton edge**.

At very high energies **pair production** becomes the dominant interaction process of photons. In the presence of a nuclear or an electron field, the photon can convert into an electron-positron pair, which starts to be possible at roughly 1 MeV ($\approx 2m_e$, with m_e electron rest mass) and exceeds the Compton scattering cross-section at roughly 10 MeV. Similar to the radiation length in the context of electrons, which can be interpreted regarding bremsstrahlung (compare 2.1.2.1), the radiation length X_0 can be interpreted as $\frac{7}{9}$ of the mean free path of a photon for pair production. Because the Feynman diagrams of bremsstrahlung and pair production are variants of another, the pair production cross-section is very closely related to the cross-section of bremsstrahlung.

As a summary, figure 2.5 shows the different electromagnetic cross-sections of photons traversing light (carbon) and heavy (lead) elements over a large range of energy.

2.1.2.3. Electromagnetic Shower

Because at higher energies on the one hand bremsstrahlung of electrons and positrons leads to the generation of photons, and on the other hand pair-production of photons forms new electron-positron pairs, high-energy electrons, positrons or photons entering matter can each cause an alternating series of both effects, the so-called **electromagnetic shower**. Due to the multiplication of particles with each interaction, the electromagnetic shower evolves as an increasing tree, with the initial projectile as the root.

An electromagnetic shower propagates trough the medium until the remaining energy fractions of the produced particles do not further allow for bremsstrahlung or pair-production, and henceforward the remaining energy is deposited via effects occurring at medium or low energies (ionization, Compton scattering, etc.). The shower depth is usually also expressed in terms of radiation length X_0 and rises logarithmically with the initial energy, while the constant cylindric expansion radius (Molière radius) of the electromagnetic shower can be relatively accurately predicted (it directly depends on the atomic number of the material).

2.1.3. Other Important Aspects

The latter summary was only focused on effects based on electromagnetic interactions, which usually dominate the energy loss of charged particles though. In contrast, this section briefly lists some selected aspects that are also exceedingly relevant for the understanding of detectors, but that do not (or not exclusively) relate to electromagnetic effects.

2.1.3.1. Hadronic Shower

For hadrons passing through very dense and/or thick matter inelastic reactions due to strong interactions become likely. At sufficiently high energies they can cause the projectile to "split" into new hadrons, which then might repeat the process as long as their fractional energy allows for further inelastic processes. That way a whole series of inelastic reactions can evolve – a **hadronic shower**. During a hadronic shower newly produced particles are mostly pions and nucleons. Especially due to the frequent production of uncharged pions π^0 (the average fraction is roughly $0.1 \log(E)$, E in GeV [20]), which predominantly decay electromagnetically into two gammas, also electromagnetic showers frequently occur within a hadronic shower and can carry a significant amount of the initial energy.

Compared to electromagnetic showers, hadronic showers are much more complex (much more different interaction processes are possible), show larger fluctuations of energy and geometry, and are typically more expanded. But similar to electromagnetic showers, they are usually expressed in terms of the average interaction length λ_I. Because an analytic calculation of hadronic showers is extremely complicated, in practice they are mostly simulated with Monte Carlo programs.

2.1.3.2. Cherenkov Radiation

If a charged particle traverses a medium with a velocity v faster than the phase velocity of light in that medium ($v_{\text{light}} = \frac{c}{n}$, n index of fraction), it can polarize the molecules of

the medium and in doing so cause the emission of **Cherenkov light**. Typically very few Cherenkov photons ($\mathcal{O}(10)$) per traversed meter of material are emitted. The continuous frequency spectrum of Cherenkov light comprises even parts of the visible spectrum [67] (causes for instance the blue glow of water around reactors). The Cherenkov photons are emitted at an angle $\Theta_c = \arccos(\frac{v_{\text{light}}}{v})$ and therefore form a Cherenkov wavefront (similar to the Mach cone). For a given material the emission of Cherenkov light starts to be possible at a threshold velocity $\beta_{\text{th}} = \frac{1}{n}$.

2.1.3.3. Transition Radiation

Relativistic charged particles at slightly higher energies than required for the production of Cherenkov photons are likely to lose small fractions of energy by emitting photons whenever crossing a boundary between two materials with different refractive indexes. The intensity of the emitted photons is roughly proportional to the particle energy (or its Lorentz factor γ respectively). The typical light yield lies at roughly 1 % per boundary crossing and the spectrum of this so-called **transition radiation** is spread around soft x-rays (2 to 40 keV). Because transition radiation is emitted at the characteristic angle $1/\gamma$, it is rather collimated along the direction of the incident particle. In order to increase the light yield by maximizing the number of material boundaries, structures like stacks of foils or blocks of foam are commonly used in front of detectors exploiting the occurrence of transition radiation.

2.2. Detectors for High Energy Physics

After the brief summary of the most important physical effects effectively leading to some kind of energy deposition in the penetrated materials, this section now gives a short outline of how these effects are exploited technologically in order to build detectors for modern high energy physics (HEP) experiments. But prior to this, a short introduction to the overall requirements of HEP detectors as well as to the challenges involved is given.

2.2.1. Requirements of Detectors for Modern Accelerator Experiments

The abstract goal of a modern detector system of an accelerator experiment is to record as completely as possible the outcome of numberless collisions, of which later on typically only very few containing a particular piece of significant information are selected. The required accuracy of position and time resolution complexly depends on countless parameters, but fundamentally must be good enough to allow for statistically significant physical conclusions. The fact that in modern HEP experiments the required accuracy, the complexity of single collisions (due to increasing energies), and the average interaction rates (due to increasing luminosities) continuously increase (or are increased on purpose because cross-sections of interesting effects become smaller and smaller), pushes both detector and accelerator designs more and more to the very edge of technology.

For each (relevant) particle evolving from a collision, the recorded data must provide either directly or indirectly the basic information, where the particle has gone (**particle tracking**), how much energy it has carried (particle momentum), and what type of particle actually has been observed (**particle identification** or **PID**). To reach that goal, very different kinds of detectors dedicated to very specific types of measurements are operated. The quality of measurement of most (sub-)detectors can be roughly expressed in terms of **energy/momentum**, **spacial** and **time** resolution.

Whereas the particle track and energy can be measured more or less directly, the particle ID is mostly derived from different (at the best redundant) measurements and/or observations. For instance to calculate the invariant mass of a particle, one usually measures directly or indirectly both its momentum and energy and often additionally considers the particle type probabilities extracted from dedicated PID sub-detectors. To name but a few examples of important observations, the deflection in a magnetic field allows for conclusions on the electric charge, the observation of transition radiation or Cherenkov light leads to the identification of electrons and the analysis of showers helps to distinguish hadrons from photons or electrons.

Because the number of known particles is huge, one could easily conclude a detector system must be able to directly recognize and identify hundreds of different kinds of particles. But in fact, most particles have such extremely small lifetimes, that they barely travel much farer than some 100 µm from their point of production and hence usually never even reach a detector (here one relies on indirect evidence). And of the few remaining (more or less) stable particles (which are something below 30), some are extremely hard to detect at all (neutrinos, again indirect evidence), and others simply occur only very rarely. As a consequence basically 8 particles remain, which a common detector system must be able to directly observe and identify ($p^{\pm}$, n, $\pi^{\pm}$, $K^{\pm}$, K^0, $e^{\pm}$, $\mu^{\pm}$, γ) [59]. All other particles are normally observed only indirectly, by carefully searching the recorded collisions (or events) for secondary vertexes[1] and missing momenta.

2.2.2. Short Outline of Modern Detectors

The following part quickly summarizes the classes of modern detectors, which are operated in many of today's high energy accelerator experiments.

2.2.2.1. Gaseous Detectors

In **gaseous detectors** ionization via charged particles or the deposition of energy by photons, lead to the production of ion-electron pairs along the particle track in the gas volume. In most detectors, an externally applied electric field purposely forces the ion-electron pairs to separate further and to drift towards an array of readout electrodes. Because the drifting electrons and ions are subject to a permanent alternation between acceleration (due to the

[1] By extrapolating tracks back to the origin, so-called secondary vertexes, the position where the decay of a short-lived particle took place, can be spotted. Because short-lived particles typically move nearly at speed of light c, secondary vertexes are separated by distances in the order of $c\tau$ from the primary vertex (with mean lifetime τ of the decaying particle).

electric field) and deceleration (due to elastic collisions with gas molecules), they move at a certain mean velocity, as long as the electric field stays uniform. Typical drift velocities of electrons are three orders of magnitude higher than of ions.

The drifting charges directly induce small currents into the readout electrodes, which could be already detected directly with proper electronics in principle. But for practical applications an additional charge multiplication is beneficial in order to increase the induced signals to reasonable magnitudes. The charge amplification is usually done by additionally applying proper and very strong electric field configurations nearby the readout electrodes, which are able to further accelerate the ions or electrons until secondary ionization becomes possible. In general the mean free path of ionization in gas decreases exponentially with the field strength as soon as a certain gas-dependent threshold is crossed. Hence at sufficiently strong fields, whole avalanches of charges are produces.

If enough energy of the projectile was transferred during an electromagnetic interaction in the first place, the produced ions or electrons might have enough energy to do secondary ionization even in the drift field and before slowing down to their mean drift velocity. For this reason the total amount of moving charges produced before reaching the region of amplification is usually both due to primary and secondary ionization[1].

Assuming perfect electronics and a proper granularity of the electrodes (compare A.1.1), the best achievable spacial resolution of a gaseous detector in general is intrinsically limited by statistical effects: Whereas the single charge clouds shortly after ionization are relatively sharply bound, they are blurred due to diffusion during their drift towards the amplification or readout area. The extension due to diffusion, both in longitudinal and transverse direction, scales with the square root of the drift distance and can be additionally affected by the Lorentz force in a magnetic field. Fortunately this circumstance can also be exploited beneficially: parallel magnetic and electric fields can be used to strongly reduce the development of the transverse diffusion. Or as an alternative, also a detector configuration with higher drift velocities and hence shorter mean collection times can help to reduce the impact of diffusion.

Because the total charge collected in a detector stage gives hints about the energy loss, which allows for a certain amount of particle identification (Bethe formula, e.g. electron/-pion separation) and also for momentum calculations, also a good energy resolution is desirable in general. Fundamentally, the achievable energy resolution is limited by the collection efficiency, the quality of the electronics, as well as by statistical effects (e.g. Landau distribution, compare 2.1.1). Whereas the statistical variation of the number of collected charges stays rather small in thick gas volumes, thin detectors suffer significantly from it. Therefore, in most applications using thin detectors several layers are placed consecutively (which of course is also done to allow for tracking). To get rid of particularly intensive hits (Landau tail) that effectively increase the gap between most probable and mean energy loss, a common strategy of more-layer detector systems is to calculate the mean energy loss only of the n smallest signals of a recognized track.

Nowadays a large variety of gaseous detectors is being built and/or operated. The various detector types strongly differ in terms of gas mixture, field geometries, type of electrodes,

[1] This depends on the type of gas. The total charge generated by secondary ionization typically is several times larger than that generated by primary, e.g. Xe at NTP: 41 e/cm primary, 271 e/cm secondary [34]

granularity of readout nodes, mode of operation, and size, but all exploit ionization, high-field charge multiplication and the collection of charge via drift fields. Popular examples of gaseous detectors using thin wires to create strong radial electric fields in their perimeter are the **multi wire proportional chamber** (MWPC, see also 2.2.2.5), the **drift chamber** or the **time projection chamber** (TPC). For time of flight measurements **resistive plate chambers** (RPC) are commonly used, which benefit from the quick response of a very thin active gas layer sandwiched between two extremely low-resistivity plates connected to high voltage. Detector types using solid and (lithographically) structured readout and amplification layers instead of wire structures are for instance **gas electron multipliers** (GEM) or **Micromegas**. The latter types in particular are presently subject to intense investigations and have been shown to provide significantly better energy resolutions and high rate capabilities – in comparison to rather classical concepts based on wire structures. However, they still suffer from several problems such as very limited radiation hardness or bad reliability.

2.2.2.2. Semiconductor Detectors

Semiconductor detectors, made in most cases of silicon or germanium, are widely used as tracking detectors or energy spectrometers. Indirectly but strongly driven by the semiconductor industry, partitioned semiconductor devices allow for both very high spatial and energy resolutions. Whereas for tracking detectors small material budgets (thin detectors) are beneficial in order to reduce multiple Coulomb scattering as much as possible, energy spectrometers typically go for a maximized stopping power (thick detectors, high-Z materials). Due to the very high integration scale (typical strip geometries $\mathcal{O}(cm)$ x $\mathcal{O}(100\mu m)$, typical pixel diameters $\mathcal{O}(10\mu m)$) and the requirement of ultra low-power and low-noise readout electronics, semiconductor detectors are comparatively expensive and maximal cover total active areas not larger than a few m^2 – or even much less.

The ionization of semiconductors due to charged particles or photons leads to the generation of electron-hole pairs. Due to very small excitation energies (e.g. roughly 3.6 eV for the generation of an electron-hole pair in Si at room temperature), relatively large signals are produced even in thin devices. A MIP for instance deposes 22 ke in 300 µm Si on average. Although the ratio of generated electrons to deposited energy is good, further electronic amplification is obligatory. In order to efficiently collect the deposited charge, the bulk material is normally applied to an electric field, forcing the electrons and/or the holes to drift to their respective readout nodes. To overcome the problem of limited bulk resistances of practically available materials (leading to very high bias currents), properly doped semiconductors are mostly reversely biased until the bulk is fully depleted. Typical collection times in Si are moderate, for instance a 300 µm fully-depleted Si detector roughly requires 10 ns for electrons or 25 ns for holes.

In general one distinguishes between **(semi-)monolithic** and **hybrid detectors**. The former type has at least parts of the active readout electronics integrated directly into the absorption material (e.g. DEPFET or MAPS), whereas the sensing material of hybrid detectors is isolated from the readout electronics. Monolithic detectors allow for much smaller material budgets and potentially for much simpler geometries, but are usually more limited (of course in dependency on the technology used) by many practical problems like

for instance a reduced space for electronic components, parasitic effects (e.g. cross-talk or the back-gate effect), the need of costly and complex non-standard fabrication steps, bad radiation hardness, or long charge collection times.

The exposure of semiconductor detectors to radiation reduces their maximum lifetime, both due to bulk damage and surface charge build-up. Bulk defects caused by atomic displacements mainly lead to increased leakage currents, the generation of space charges, or the emergence of charge carrier traps, and complexly depend on type, intensity and energy of the irradiation. In contrast, surface charge build-up, which can cause significant surface leakage currents, only depends on the total amount of absorbed energy and is independent of the exact particle type. Various techniques are known to reduce radiation damage or its impact, very important are for instance a proper choice of technology (e.g. material, layer thicknesses or doping profiles), geometries (e.g. strip pitch), and the overall design strategy (e.g. distance to the beam-pipe or additional shielding). Moreover, in many cases self-annealing (primarily at high temperatures but to some extend also at room-temperature) can take place, which potentially reduces or even removes completely previous radiation defects.

2.2.2.3. Photon Detectors

Because, for example, semiconductor or gaseous detectors are also able to detect photons in general, the subsequent brief summary of **photon detectors** is not disjunct as to the other detector sections, but rather focuses on sensors and details of sensors dedicated to the detection of photons.

Typically one distinguishes between vacuum or gaseous photon detectors (detectable energy range somewhat exceeding the range of visible light, including the important frequencies of Cherenkov and scintillation light) and solid-state photon detectors (e.g. Si is nontransparent from visible light to hard x-ray or even soft gamma-rays)[1]. Important examples of vacuum photon detectors are the **photo multiplier tube** (PMT) or the **microchannel plate** (MCP), examples of solid-state detectors are the **charge-coupled device** (CCD), the **active pixel sensor** (APS) or the **avalanche photodiode** (APD).

The first and most important step of each photon detector, which mainly differentiates photon detectors from other detector types, is the conversion of the incident photons into primary electron-hole/ion pairs or photo-electrons (compare 2.1.2.2). Afterwards and "as usual", the primary charge is amplified, collected and measured. The mean charge signal generated in certain device can be calculated as the product of QE (the **quantum efficiency**, number of primary charges per incident photon), CE (the **collection efficiency**, fraction of primary charges which actually reach the amplification region), and G (the amplification gain). Naturally the factor $QE \cdot CE \cdot G$ is subject to statistical fluctuations, which together with the given noise level of the readout electronics determines the best achievable energy resolution and hence the quality of conclusions on the number of primary photons. The mean number of photons per measurement cycle differs strongly from application to application. Arrays of APDs for instance provide single photon sensibility, whereas CCD chips usually integrate high luminosities over relatively long periods.

[1] For the detection of very high energy photons, or gammas, see also section 2.2.2.7

2.2.2.4. Cherenkov Detectors

The emission of Cherenkov light in a well defined angle as soon as charged particles travel above a certain threshold velocity (see 2.1.3.2) is frequently exploited to build detectors dedicated for particle separation (and sometimes also for tracking or particle counting). Very important is for instance the separation of electrons from pions. **Cherenkov detectors** normally consist of large volumes of dedicated radiator materials (for instance gas), carefully chosen with respect to many crucial parameters (e.g. refractive index, radiation length of Cherenkov light or radiation tolerance), and an arrangement of photon detectors (e.g. PMTs).

The number of photons emitted per traversing particle in the radiator volume is typically very small ($\mathcal{O}(10)$), why Cherenkov detectors rely on both very good quantum efficiencies of the photon detectors and very little background light in the setup. Detectors that only decide whether the velocity of the traversing particle was above a certain threshold or not (by measuring if there was Cherenkov radiation or not) are called **threshold detectors**. In contrast, Cherenkov detectors also measuring the velocity-dependent polar angle of the emitted light in order to evaluate momenta, are called **imaging detectors**. Since common photon detectors are not able to provide information about the polar angle of the Cherenkov photons directly, image detectors usually make use of focusing devices (e.g. lenses or mirrors). The very popular **ring imaging cherenkov detector** (RICH) for instance spherically maps all photons emitted on a certain track (note the Cherenkov wavefront, compare 2.1.3.2), to a ring on a two-dimensional photon detector plane. That way, due to the direct geometric dependency of the ring radius on the initial Cherenkov angle, some additional momentum resolution can be extracted and/or a probability for the respective particle type set.

2.2.2.5. Transition Radiation Detectors

Because this work is particularly related to the transition radiation detector, this section is slightly more detailed.

The general goal of a transition radiation detector (TRD) is to measure the transition radiation (TR) that is emitted with a certain probability each time a charged particle crosses the boundary between two materials (see 2.1.3.3). Very similar to Cherenkov detectors, the presence or quasi absence of transition radiation can lead to a threshold decision whether a charged particle lies within a certain range of velocity or not. Different however is the fact that also slower particles are able to transmit TR in general (there is no threshold for TR as for Cherenkov light), although the related probabilities are rather small. Similar to Cherenkov detectors, TRDs are able to separate fast and lightweight electrons from slower but heavier pions. Practically this starts to be possible at momenta of roughly 1 GeV/c.

For the detection of TR photons in principle all types of photon detectors sensitive to soft x-rays can be used. But due to the fact that the TR photons are rather collimated along the particle track, the TR sensor is usually traversed by both the TR photons and the charged particle. Whereas on the one hand one can also benefit from the additional ionization signal in the detector, this geometric restriction on the other hand can increase the material budget and lead to the difficult task of finding a rather small TR signal within

a large ionization signal. The latter considerations and moreover the high costs of large detector areas, are two very important reasons, why by far most of today's TRDs are based on gaseous multi-wire proportional chambers (MWPC). But for example also straw tubes are sometimes used, and moreover completely new concepts such as for instance TRDs based on DEPFET sensors were proposed [37].

The MWPC, which belongs to the group of gaseous detectors, is composed essentially of a plane of thin, parallel and equally spread (anode) wires, placed in-between two metal cathode planes. The evolving volume in-between the cathode planes is filled with a gas mixture matching the requirements of the respective experiment. By applying high electric potentials between anode and cathodes, a roughly parallel and homogeneous electric field develops across the entire gas volume in a well designed MWPC, which only and purposely in the closer surrounding of the anode wires goes over to a radial field. The parallel field typically forces the ionized electrons to drift towards the wires, until they are accelerated and effectively amplified in the radial (and hence increasingly strong) electric field (compare 2.2.2.1). Basically, the ionized electrons are either generated by charged particles traversing the gas volume or by TR photons which generate small charge clouds by primary and secondary ionization shortly after entering the entrance window. By replacing the cathode plane with a cathode wire grid and adding a second cathode plane at a considerable distance, some MWPCs purposely expand the otherwise very short drift distances to significant lengths ($\mathcal{O}(cm)$). To readout the signals, the electronics can be either connected to the different anode wires, or the cathode plane nearby the wire plane can be segmented into readout pads.

As an example design a sketch of the ALICE TRD is shown in Fig. 2.6, which is based on a MWPC with a relatively long drift region.

MWPCs usually operate in proportional mode, where electric field strengths and geometries are chosen such that the gain stays constant and independent of the magnitude of the primary charge. The charge avalanche, which is generated when the drifting electrons reach the amplification region in the vicinity of the anode wires (see also 2.2.2.1), initially contains both ions and electrons. But since the electrons nearly immediately reach the anode wire, mainly the motion of the back-drifting induce the electric signals into the readout electrodes (anode wires or cathode pads). The induced signal current is typically characterized by a quickly evolving peak that is followed by a relatively slowly decreasing tail caused by the far back-drifting ions (the **ion-tail**). Although signals of chambers having a larger drift region additionally show a plateau in-between the peak and the tail (due to the arrival of a long chain of ionized electrons instead of only one single electron cloud).

The reachable spacial resolution of a MWPC depends on the pad granularity, statistic effects (diffusion), and the quality of the electronics. A brief discussion is given in the attachment A.1.1.

As mentioned earlier, in TRDs based on MWPCs, the radiator material is placed in front of the gas volume and both the charged particle and the TR photons enter the chamber nearly at the same spot on the entrance window and from the same direction. The gas mixture is usually chosen to have a good TR photon acceptance and mostly contains a small fraction of quenching gas, which is used to slow down too fast electrons and effectively helps to decrease the average collection time (e.g. very common: 85 % Xe and 15 % CO_2). Besides

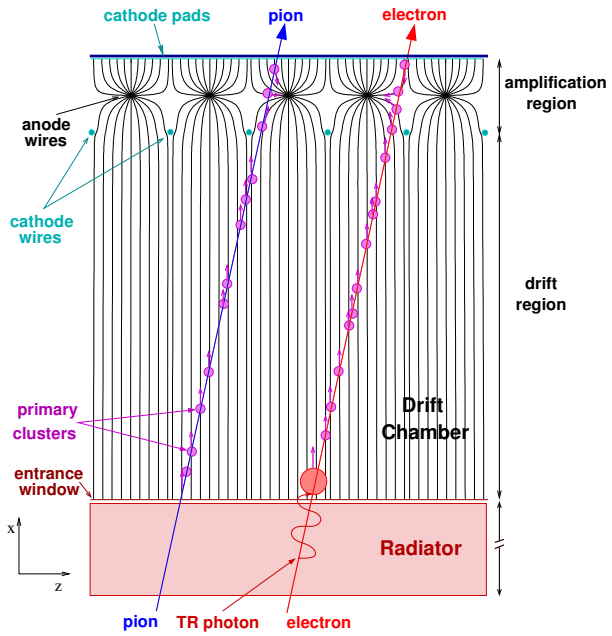


Figure 2.6.: Design concept of the ALICE TRD. On top, the amplification region between two cathode layers (pads and wires) is separated from the long drift region in the middle. The drift region ends at the entrance window on the bottom, onto which the TR radiator is mounted. Exemplarily, ionized tracks of an electron and a pion entering the chamber from below are shown. For illustration, the lighter electron emits a TR photon, which shortly after entering the gas ionizes a small charge cloud (and disappears). Graphic from [32].

the proper gas mixture, the detector thickness is a very crucial design parameter. On the one hand the thickness must be large enough to assure that most TR photons are absorbed, but on the other hand thinner detectors benefit from faster collection times, better high rate stability, easier handling/building, and lower material budgets.

Practically, MWPC-TRDs can deliver information about both the particle momentum (particle id) and the position of its trajectory. If the detector, as mentioned earlier, is optimized for the separation of electrons from pions, it does not only benefit from the additional signal amplitude due to TR photons, which occurs predominantly for the much faster electrons, but also from the fact that the mean ionization loss of electrons is slightly higher than that of pions (given the same energy). Typical numbers in the context of TRD electron/pion separation are pion suppression factors of about 100 at an electron efficiency of roughly 90 % [28]. Tracking information is normally gathered using TRDs arranged in several layers, each delivering a 1D (strips) or 2D (pads) hit position. In doing so, reachable track resolutions are $\mathcal{O}(100\,\mu\text{m})$.

As an example of a signal of a MWPC-TRD with a relatively long drift region, Fig. 2.7 shows mean current signals as a function of (drift) time that are seen by the readout electrode. Shown are signal shapes caused by electrons (with (red) and without (green) radiator) and by pions (blue). The signals were measured with an ALICE TRD prototype (corresponding to the one shown earlier in Fig. 2.6). The average signals have a sharp

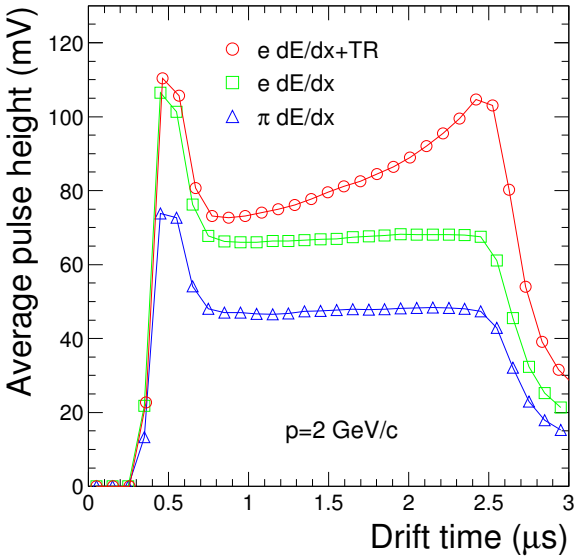


Figure 2.7.: Average signal height at the readout cathode as a function of (drift) time of electrons (with and without TR) and pions measured with an ALICE TRD prototype [9].

maximum after roughly 0.5 μs, which correlates to the quickly gathered charges that were ionized directly in the amplification region. Then, in the absence of TR, the drifting electrons that were generated along the track, induce a constant signal with a length (time) proportional to the length of the drift region, which leads to the earlier mentioned plateau. The TR photons, which are likely to be absorbed quickly after entering the gas volume and hence tend to arrive delayed, add an increasing component to the electron signal (red). The plot demonstrates nicely how by either integrating the signals or by analyzing the respective signal shapes, the TR component can significantly add to the quality of the extracted particle probabilities and hence allow for a good electron/pion separation.

2.2.2.6. Scintillators

Excited molecules of certain materials (e.g. due to ionization) can release small fractions of their energy by emitting short light pulses. Detectors exploiting this mechanism are called **organic** or **inorganic scintillators**. Because the wavelength of the emitted "scintillation light" normally lies in the UV region, it cannot be detected directly with common photon detectors (e.g. Si is transparent to UV). Therefore wavelength shifters (fluorescent materials), which effectively increase the UV wavelengths to more convenient lengths, are either added to (inorganic crystals are doped) or properly mounted besides the scintillation material. A very important design parameter of a scintillator is its overall efficiency, which consists of the collection and transport efficiency of both the scintillator and wavelength shifter material, and the quantum efficiency of the photon detector.

Most organic scintillators are made of plastic, although liquid scintillators – due to their good ratio of volume to costs – are also in use. In general, fast decay times and very quick

response times make organic scintillators very applicable for time measurements or as trigger devices at relatively low costs (reachable time resolutions lie significantly below 1 ns). Moreover, and because plastic scintillators can be easily formed (e.g. into fibers) and handled, organic scintillators also qualify for tracking applications. Especially fibers comprising scintillator and wavelength-shifter materials are commonly used. Arranged in bundles, they can be read out conveniently at the wire-endings, where the light is automatically led due to total internal reflexion.

Inorganic crystal scintillators in contrast have much higher densities but typically also much higher decay and response times. They are used in applications where high stopping power or high efficiency of electron/photon absorption is required (e.g. electromagnetic calorimetry or gamma-ray detection).

2.2.2.7. Calorimeters

The main purpose of **electromagnetic** (ECAL) or **hadronic calorimeters** (HCAL) is to evaluate the total particle energy by causing and measuring electromagnetic (see 2.1.2.3) or hadronic showers (see 2.1.3.1). In many presently operated high-energy experiments, a combination of both (normally an ECAL in front of an HCAL) is used, which allows for energy measurements of photons and electrons, as well as hadrons. In the latter case, it is not unlikely and in some way even intended, that some of the hadronic showers already start in the electromagnetic colorimeter. The thickness of an ECAL typically lies within the range of $15-30$ radiation lengths X_0, that of HCALs mostly within the range of $5-8$ interaction lengths λ_I [34]. Two major variants of colorimeters exist, first the **sampling calorimeter**, which sandwiches absorber material and active material (e.g. scintillator or semiconductor detectors), and second the **homogeneous calorimeter**, which is solely made of active material (e.g. a scintillator crystal).

The energy resolution of electromagnetic calorimeters depends basically on three factors, the signal statistics (e.g. shower fluctuations), the quality of the setup (e.g. non-uniformities or accuracy of calibration) and the overall performance of the electronics (mainly noise). In general, the best energy resolutions can be reached with homogeneous absorbers, which for instance allow for better signal statistics or more uniformity. The achievable spatial resolution is determined by the effective Molière radii of the showers and by the transverse readout granularity of the detector layer(s). In particular, the ability of some ECALs to additionally measure the direction of the incident particle can help to get some tracking information even of high energy photons, to which most conventional tracking detectors (e.g. thin Si strips) are nearly transparent.

Compared to ECALs, the design of hadronic detectors is – at least theoretically – much more complicated. That is basically due to the fact that hadronic showers develop much more complexly (secondary electromagnetic showers, significant energy leakage, etc.). One normally tries to built **compensating HCALs**, which are designed so that the hadronic detection efficiency h just equals the electromagnetic detection efficiency e. That way and as the naming indicates, the usually strongly fluctuating differences between electromagnetic and hadronic fractions in the shower are compensated. Because only sampling HCALs can

be properly adjusted to have $h/e \approx 1$, so far no homogeneous HCALs are being built or operated.

CHAPTER 3

The CBM Experiment

The main goal of the planned fixed-target accelerator experiment CBM (compressed baryonic matter) at FAIR/GSI (facility for antiproton and ion research at GSI Helmholtzzentrum für Schwerionenforschung [3]) is to investigate extremely hot and very dense baryonic matter, and in particular to further explore the QCD (quantum chromodynamics) phase diagram, sketched in Fig. 3.1.

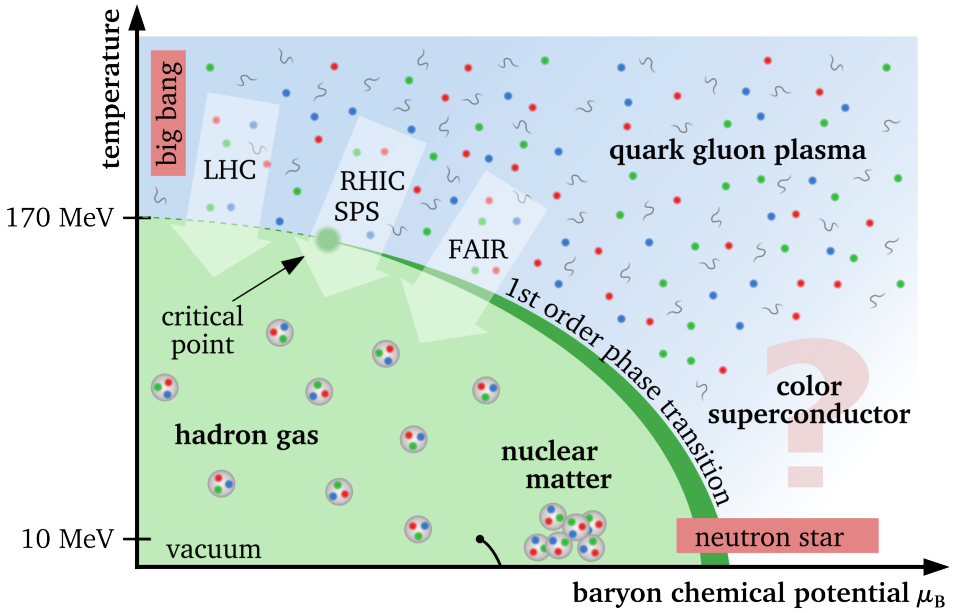


Figure 3.1.: Simplified view of the QCD phase diagram. Most details shown are either only roughly understood or even mere predictions. The diagram is not to scale and the numbers are only approximately valid.

The CBM experiment is one of several planned experiments that shall be operated at FAIR/GSI starting with a first beam not before 2018 [33]. Up to now most parts of CBM, such as the detectors and the electronics, are in a late R&D (research and development) phase. Therefore nearly all sub-components are at least roughly drafted, many prototypes have been developed and tested, and most technical design reports of the different sub-detectors (including their respective electronics) are presently being written. Nevertheless, the technical, physical and mechanical developments on all levels of hierarchy will go on for years to come.

This chapter first shortly describes the FAIR accelerator, then summarizes the physical goals of CBM, and finally deals briefly with the CBM detector and its components.

3.1. The FAIR Accelerator

Figure 3.2.: Left: model of the planned FAIR extension together with the already existing GSI facilities. Right: aerial photograph from August 2012, showing the very first stage of construction. Both photos are taken from GSI [3].

The first stage of construction of the FAIR accelerator near GSI has just begun, as shown in Fig. 3.2. The FAIR accelerator will have to compete with the (mostly already existing) heavy-ion accelerators RHIC (relativistic heavy ion collider) at BNL (brookhaven national laboratory, Brookhaven, USA [1]), SPS (super proton synchrotron) / LHC (large hadron collider) at CERN (european organization for nuclear research, Geneva, Switzerland [2]) and NICA (nuclotron-based ion collider facility) at JINR (joint institute for nuclear research, Dubna, Russia [4]). FAIR, compared to the other accelerators, pushes not that strongly towards ultra-high energies, but rather focuses on high-intensity and high-precision beams at moderate energy. The maximum beam energy will be about 20 times larger than the already available energy at GSI, although one aims for intensities being higher by several orders of magnitude depending on energy and projectile species [33]. In order to get extremely high-precision beams, various sophisticated beam manipulation methods (e.g. stochastic and electron cooling) will be realized.

The characteristics of FAIR allow to explore the QCD phase diagram in an area of medium temperature but at relatively high densities, whereas the other accelerators investigate at higher temperatures but lower net-densities (compare Fig. 3.1). FAIR is designed to deliver high-precision beams over the full "nucleon range" from protons/antiprotons to uranium.

As sketched in Fig. 3.3 (blue), the existing accelerator facility at GSI comprises of UNILAC (universal linear accelerator) and SIS18 (Schwerionen-Synchrotron, 18 Tm bending power, 216 m circumference), which will serve as an injector to FAIR. The heart of FAIR (Fig. 3.3, red) will be two equally large synchrotron rings SIS100 (100 Tm bending power) and SIS300 (300 Tm), both sharing a tunnel with a circumference of about 1100 m. SIS100, which will be available roughly in 2018, is designed to deliver for instance intense pulsed (5×10^{11} ions per pulse) uranium beams at 1 AGeV and proton beams (4×10^{13} protons per pulse) at 29 GeV, whereas SIS300, which shall be completed later, will allow for uranium beams up to 35 AGeV (1.5×10^{10} ions per spill) [33].

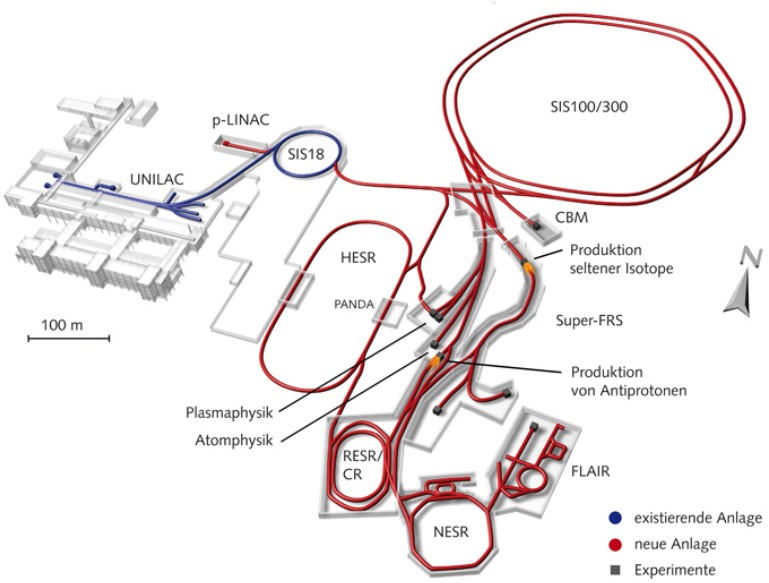


Figure 3.3.: Overview of the FAIR accelerator, components, and experiments [3].

Followed by the SIS rings, a huge variety of cooler-storage rings and physics experiments are going to be operated (also see Fig. 3.3). Besides the two central experiments on QCD, CBM and PANDA (antiproton annihilation at Darmstadt), various smaller experiments will be performed, which are separated into the two groups APPA (atomic, plasma physics and applications) and NUSTAR (nuclear structure, astrophysics and reactions).

3.2. Brief Outline of the CBM Physics

As mentioned earlier, the abstract goal of CBM is to explore the QCD phase diagram (Fig. 3.1), which is basically a map of hadronic matter in dependency on temperature and baryon chemical potential (measure of net baryon density). Many parts of the diagram are to

this date either only roughly known or even merely predicted, but numerical calculations of quantum chromodynamics discretized on a space-time grid (lattice QCD), various theoretical approaches, and not at least first experimental data (for instance from SPS or RHIC) lead to several more or less concrete entries in the diagram.

The most important landmark that one assumes to be at very high temperatures and/or net baryon densities, is a phase transition from "conventional" (confined by strong interactions) baryon matter (hadron gas or nuclear matter) to deconfined matter, the so-called quark gluon plasma (QGP). And indeed, after the initial announcement of a "new state of matter" that has been discovered at SPS/CERN [39], the experiments both at LHC/CERN and RHIC/BNL now frequently announce potential evidence for ultra-hot QGP (e.g. latest news from the websites [5] [6], results will be published in "Physical Review Letters" and at "Quark Matter").

Derived from QCD, QGP is expected to be a soup of quarks and gluons, where quarks (and gluons) are nearly freed from their strong attraction for one another[1]. Although the exact properties are still unknown, one expects QGP to behave very similar to an ideal fluid. The state of quasi liberation of quarks from strong interactions is usually called "asymptotic freedom", since quarks barely interact strongly as long as they stay close to each other (in analogy with two donkeys bound together with a loose string). Lattice QCD calculations, which can be best performed for zero net baryon densities for reasons of numerical complexity, predict a smooth transition (a cross-over) at small net baryon densities but high temperatures from hadronic to partonic matter (QGP). The critical cross-over temperature is predicted to lie within the range from 150 to 190 MeV (170 MeV was chosen for Fig. 3.1). Then, at higher densities the estimated transition temperature decreases slightly until eventually, at a yet unknown position, a critical point is (or might be) reached. The critical point marks the end of the smooth cross-over and the beginning of an abrupt first-order phase transition, which is assumed to separate QGP from conventional matter at higher densities (also sketched in Fig. 3.1).

Looking further into the QCD phase diagram, some areas stand out in particular. First, there is the area of low densities but ultra-high temperatures, which is presently being explored with heavy-ion beams at LHC. In this area, matter is in a similar condition as it has been shortly (up to some microseconds) after the big bang (or at least that is what is assumed). Hence exploring the area of low net baryon densities and ultra-hot matter descriptively means to investigate in the physical situation of the early universe, when the fireball after the big bang predominantly consisted of QGP until the expansion was sufficient large to allow for a "cooling-down" to hadronic matter. In contrast, in the area of very low temperatures but high densities, conditions are equal to those in the inner core of compact stars. And moreover, still at low temperatures, but at ultra high densities, even more unknown phenomena are predicted to occur, for example color superconductivity. For the latter reasons, exploring the QCD phase diagram is commonly put onto a level with studying neutron stars, the fundamental nature of matter and not at least the development of the early universe.

[1] The strong force still acts of course, even though the coupling constant is strongly reduced at small distances and/or high momenta.

In order to produce the very extreme QCD conditions in the laboratory, one totally relies on heavy-ion collisions. But even if those desired conditions can be created in principle, they only remain stable for very short time periods (for about 1×10^{-22} s [22]). Therefore and in order to be able to even define and also measure statistic figures such as temperature or pressure of the dynamically created "new state of matter", both a critical number of particles undergoing the reaction and a sufficient local equilibrium are necessary. The former requirement explains, why only heavy-ion beam collisions are able to create QGP, whereas for instance direct proton-proton collisions are not.

The particle soup (or the fireball) that is produced if two relativistic and hence Lorentz-contracted heavy-ions at a sufficient center of mass energy centrally collide[1], quickly passes through different phases, which are "high-density" (potentially including the transition to QGP), "expansion" (increase of volume, decrease of temperature and density, QGP eventually hadronizes again, then further expansion of hadron gas), and "freeze-out" (chemical freeze-out: no more inelastic scattering, the relative multiplicity of the respective hadrons is fixed – thermal freeze-out: no more elastic scattering, the momentum distribution settles). That means in particular, that exploring a nucleus-nucleus collision is not a simple static measurement, but requires to observe the complex time dynamics of the reaction. Since in the different phases various effects with typical probabilities lead to the emergence of many kinds of particles at characteristic multiplicities, the recording of certain particles and particle spectra is an effective method to probe the dynamics of the collision.

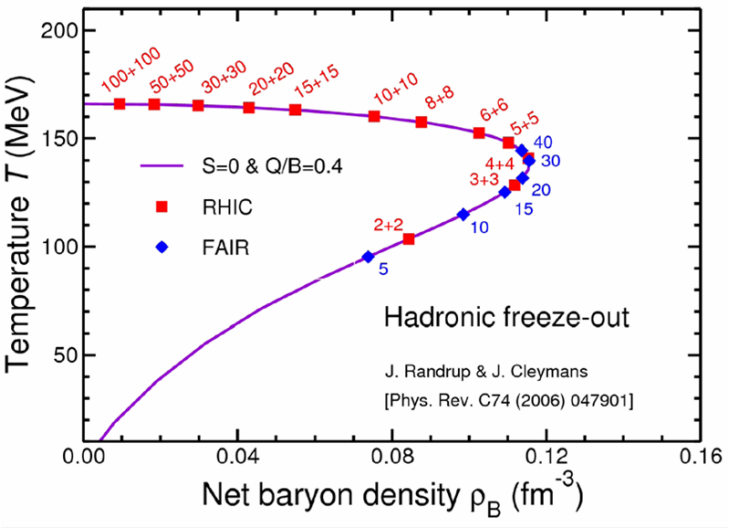


Figure 3.4.: Hadronic freeze-out line in the ρ_B - T (net baryon density - temperature) plane, which is based on experimental data from [27]. The expected freeze-out conditions at the RHIC collider (red) and the FAIR fixed-target facility (blue) for different beam energies (in AGeV) are also shown. Figure from [36].

The question, how the different areas of the QCD phase diagram can be explored in nucleus-nucleus collisions where basically only the particle type and the beam energy can

[1] Nucleons of a certain heavy-ion that undergo a collision are called participants, whereas the rest is called spectators. Measuring the spectators is an import method to diagnose various properties of the collision zone.

be varied, is roughly answered in Fig. 3.4, which shows various expected hadronic freeze-out points in the ρ_B - T plane for different realistic beam energies (note: the baryon chemical potential μ_B depends non-linearly on the net baryon density ρ_B): While the temperature of the final states (at freeze-out) monotonically increases with the collision energy, the correlated net baryon density initially increases, peaks at some medium beam energy and then nearly returns back to zero for very high collision energies. The decrease of ρ_B after the maximum is the result of the increasing nuclear transparency [58]. Due to this argumentation ultra-high-energy facilities like RHIC or LHC effectively investigate matter at high temperatures but at low(er) net baryon densities, whereas the energy range of FAIR mainly spreads around the high-density peak at little lower temperatures (blue dots in Fig. 3.4).

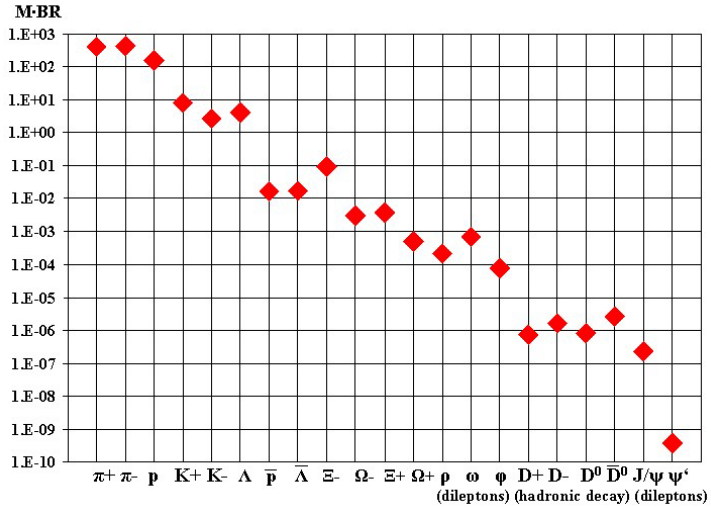


Figure 3.5.: Calculated particle multiplicities (M) times branching ratio (BR) of central Au+Au collisions at 25 AGeV. For the vector mesons ($\rho, \omega, \phi, J/\psi, \psi'$) the decay into lepton pairs was assumed, for D mesons the hadronic decay into kaons and pions [64].

In the CBM experiment, a big part of the physical agenda is to look at particles emitted in the earlier and dense phase of the collision – in contrast to the heavy-ion experiments at Dubna, Brookhaven or Geneva, which are due to luminosity limitations forced to predominantly investigate in the late and dilute phases [36]. Observables that are potentially feasible to give insight in the earlier phase of the fireball are for instance particles containing charm quarks, vector mesons or multi-strange hyperons. But exactly those particles occur extremely rarely, what is quantified in Fig. 3.5. The figure exemplarily shows the calculated product of particle multiplicities and branching ratios of central Au+Au collisions at 25 AGeV. It becomes apparent, that most particles of exceeding interest for CBM (vector and D mesons), lie in the area of 1×10^{-6} and below. For this reason and in order to allow for statistically significant results derived from those rare probes, CBM does not only require ultra high beam luminosities, which FAIR will be able to provide, but demands at the same time an ultra fast and radiation hard detector system, which effectively chal-

lenges both detector and electronics designers. Latest estimations expect reaction rates up to 10 MHz, while up to 1000 charged particles per collision are produced [36].

Moreover and besides the high-speed operation, the CBM detector is designed to allow for a comprehensive scan of various observables at very different beam energies. CBM will be the first experiment to measure the just mentioned rare probes within the FAIR energy range, but intends to look at bulk observables as well. In general, the main focus of the experiment is set on the phase boundaries or transition regions of the QCD phase diagram at higher densities. And in addition to the search for direct evidence of QGP, one intends to look for fluctuations or sudden changes of collective flow, (invariant mass) spectra, multiplicities, momenta, and yields in different QCD phase regions, where potential phase boundaries are expected.

3.3. The CBM Detector

Because the FAIR facility will begin to operate with a modularized start version based on SIS100 first [68], also a reduced version of the CBM detector will be built initially, which nonetheless shall already allow for a significant physics program at SIS100 energies and beam characteristics. Directly in front of the CBM detector and in order to measure electron-positron pairs at medium SIS100 energies (up to 8 AGeV), the already existing HADES detector [29] will be installed, thus sharing the same cave. Then later on, and synchronously to the availability of SIS300, the completion of the full CBM detector system is scheduled.

Subsequently the full CBM setup is further described. Most numbers are from the CBM physics book [36].

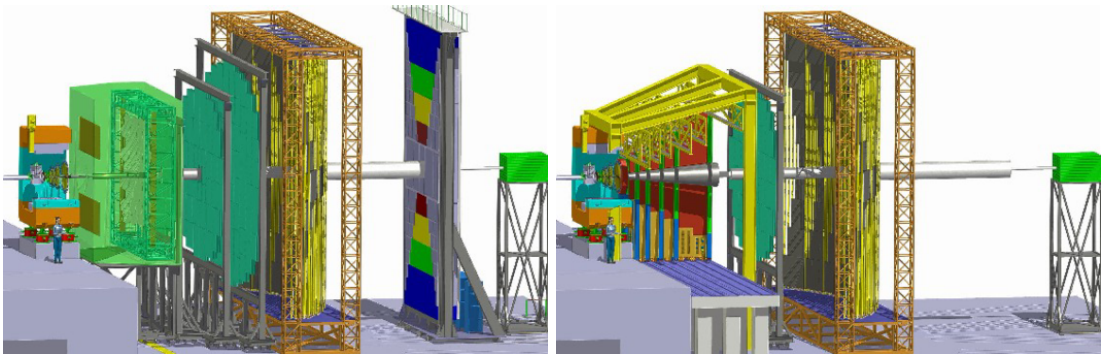


Figure 3.6.: The two different (final) CBM detector configurations. Left: electron setup using the RICH detector, right: muon setup with the MuCh system. Figures from [64].

As shown in Fig. 3.6, the final CBM system will allow to choose between two different setups – one focusing on electron identification, the other on muon detection.

The heart of both setups will be the micro-vertex detector (MVD) in combination with the silicon tracking system (STS, see 2.2.2.2). Both MVD and STS are installed inside a

large dipole magnet. The MVD will be made of two layers of thinned Monolithic Active Pixel Sensors (MAPS) located 5 and 10 cm downstream the target and will be used for the detection of secondary vertexes of D mesons (one expects vertex resolutions of 50 - 100 μm for D mesons). The about 8 double-sided silicon micro-strip layers of the STS will allow for good track reconstruction and momentum determination even at the high CBM multiplicities, and will be mounted 30 - 100 cm behind the target (typical hit resolutions of 25 μm shall be achieved). Typical challenges for the MVD/STS designers are small material budgets, high hit rates, low noise and power dissipation of the readout electronics, and sufficient radiation hardness (especially of the MAPS).

In the electron setup a ring imaging Cherenkov detector (RICH, see 2.2.2.4) will be installed behind the STS, mainly to allow for electron identification and pion suppression (typical factors are within the range 500 - 1000 for particle momenta below 8 GeV/c). In particular due to pair conversion of gamma rays (mainly in the bulk material in front of the RICH), one expects about 100 rings to occur per Au+Au collision at 25 AGeV (roughly 20 photons per ring on average). Detector optimizations include a proper gas mixture, the size and geometry, the mirrors, the readout electronics and not at least the ring-finding algorithm.

For the muon setup the RICH is exchanged with the muon chamber system (MuCh), which shall search for low-momentum muons in the intense hadronic background. In order to track the low-momentum muons undergoing large multiple scattering, 6 hadron absorbers (iron plates of thicknesses 20, 20, 20, 30, 35, and 100 cm) are each interlaced with 3-layer highly granulated gas electron multipliers (GEM, see 2.2.2.1). The high granularity of the GEMs will allow for effective tracking in the environment of high particle densities, while the number of traversed layers will indirectly give information about the muon momenta. Due to the good hadron absorption, the implementation of muon triggers will become possible (e.g. muons from J/ψ decays at beam energies above 15 AGeV have to traverse all 6 absorbers).

In both setups, either after the RICH or the MuCh system, a transition radiation detector (TRD, see 2.2.2.5) will be installed. The full TRD version used in the electron setup will consist of 3 stations (each having 3–4 layers) arranged in rectangular readout pads (sizes between 1 and 12 cm^2). The whole TRD will cover a total active area of about 1100 m^2. The main purposes of the TRD are particle tracking and (additional) electron identification (experts forecast a pion suppression factor of above 100 at 90 % electron efficiency). To handle the high particle rates (e.g. 100 kHz/cm^2 for 10 MHz minimum bias Ar+Ar collisions), very thin multi-wire proportional chambers (MWPC) with no or only small drift regions are being developed, although GEMs are considered as a fall-back solution.

A time of flight (TOF) wall made of resistive plate chambers (RPC, see 2.2.2.1) will be mounted at a distance of 10 m from the target and used in both setups. The TOF wall must be able to measure charged particles with time resolutions down to 80 ps. Again, one of the main challenge for the designers is to make the TOF work properly at the high CBM particle rates (up to 20 kHz/cm^2).

Only the electron setup will use a “shashlik” electromagnetic calorimeter (ECAL, see 2.2.2.7) made of 140 lead/scintillator (1 mm/1 mm thickness) layers to measure direct

photons or photons from neutral meson (π^0, η) decays. The CBM ECAL will be similar to the ECALs operated in the HERA-B (DESY), PHENIX (BNL) and LHCb (CERN) experiments.

A projectile spectator detector (PSD) will complete both CBM setups. It will determine the centrality of the collision and the inclination of the reaction plane by counting the number of nucleons that did not directly undergo the ion-ion collision (the spectators).

3.4. The Data Acquisition System

The CBM challenge to handle very high reaction rates up to 10 MHz does of course not only affect the mere detector design, but also sets hard requirements for nearly all components of the data acquisition system (DAQ). In CBM the DAQ is designed to be free-running, which practically means that no global (first-level) trigger signal is available.

A global trigger signal in conventional systems effectively coordinates the selection and tagging of interesting information units and indirectly helps to synchronize the recorded data. In contrast, the strategy to have a completely free-running system has significant consequences for nearly all leaves and nodes of the DAQ network tree, and demands various "unusual" design techniques. In conventional systems, the recorded data of a certain event is temporarily and locally (nearby or directly in the front-end electronics) stored, until some global trigger decision either leads to a rejection of the respective records or forces all systems to tag (e.g. with some event ID) and to further pass the stored information. Moreover, in many cases the global trigger signal even directly enforces the recording of data in the first place. For instance it can tell a rather passive readout component to even check for new information (e.g. some integrated charge), in analogy to a photographer pressing the trigger in order to tell the camera to readout the pixel sensor.

In contrast, the absence of a global readout command in a free-running environment requires complete autonomy of nearly all components, and particularly of the front-end electronics. That has at least two major consequences: first, most readout components must independently take decisions, for example if certain data shall be recorded (e.g. the readout electronics must be "self-triggered"), transferred (e.g. intermediate nodes might need to drop packages if the network is busy), or processed. And second, the whole DAQ faces continuous data streams rather than synchronized bunches of packages, which demands complicated and effective techniques, such as stream-merging (e.g. hits first must be locally tagged and later globally combined to events), high-throughput online data processing (e.g. feature extraction: interpretation of recorded raw data to reduce the message size), or high-speed data transfer and storage. In particular, the DAQ tree requires a well-balanced hierarchy of elastic buffers and available data bandwidth.

As drafted in the CBM DAQ data flow diagram (Fig. 3.7), the CBM DAQ network tree continuously pushes the generated data from the front-end electronics (left hand side of the figure) through several data processing and combiner boards towards a processor farm called the "green cube" (right hand side). In the green cube, which will be located about 700 m away from the CBM experimental cave in a dedicated building, the asynchronous data streams from the sub-detectors are eventually combined to whole events (online). Then the first level event selector (FLES) directly chooses only the most promising events and thus initializes the final storage of the recorded data to a hard disk.

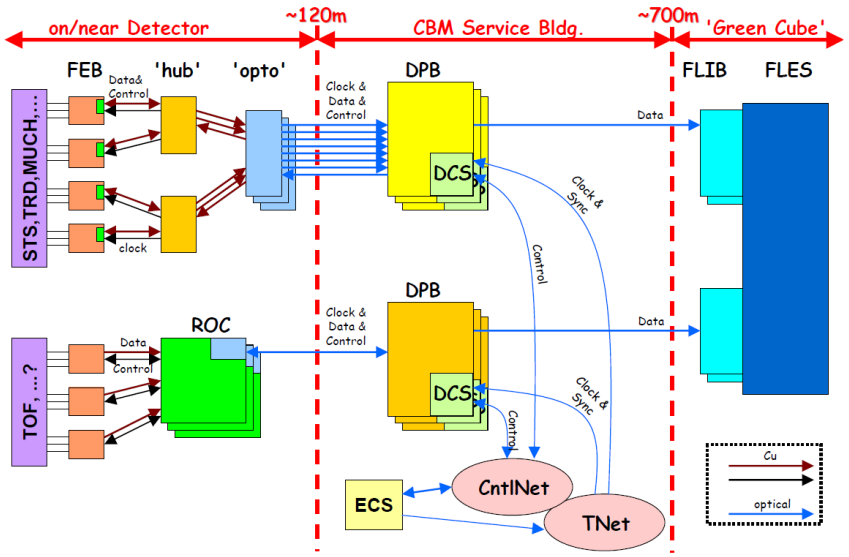


Figure 3.7.: CBM DAQ data flow diagram conceptually showing the DAQ tree with the front-end ASICs on the left hand side, several data processing and combining nodes in the middle, and the computer farm "green cube" on the right hand side. Figure taken from [51].

All over the DAQ network, the unified protocol "CBMnet" will be used to transport data streams and control messages (in both directions), and moreover to distribute synchronization telegrams, the so-called deterministic latency messages (DLM). Since the latter requires synchronously running clocks in all network leaves and nodes, another important concept of CBMnet is to propagate a central clock – node by node – to all leaves of the network tree, either directly or indirectly hidden in the data (clock data recovery). A more detailed description of the CBMnet protocol is given in section 6.4.3.

As indicated in Fig. 3.7, the interconnections nearby the network leaves and therefore close to the beam line will be realized using copper wires, whereas after some meters the data streams are combined and transferred over optical links. Although copper wires compared to optical fibers only have medium bandwidths and are practically limited to lengths of some meters (at the required throughput and using custom drivers, such as for instance LVDS output cells on an ASIC), they generally offer more flexibility, easier handling and higher granularity, and, most important, allow to place the otherwise required, very radiation-sensitive, and bulky opto-couplers (e.g. VCSELS) at some more distance to the beam line.

The challenge of the free-running CBM DAQ system in general is the partitioning and coordination of the whole network and all its components on various levels of hierarchy in the high-rate, high-radiation, and low-space environment of the experiment, while both the details of all sub-detectors on the one hand and the requirements of the computer farm on the other hand must be carefully considered. For instance the network bandwidths must be properly balanced while the different scenarios of the CBM detector must be considered, various mechanical and special constraints require an elaborate granularity, numberless components must be either radiation tolerant enough or shielded, placed or designed prop-

erly, and the required processing tasks must be identified and properly distributed over the available network nodes. Present DAQ developments cover the writing of new firmware, the building of new FPGA boards, the design of a digital network ASICs (the HUB chip), the evaluation of availability and price both of technologies and components, the selection of proper connection standards and connectors, the simulation of countless aspects, the improvement and testing of synchronization strategies, and the strategic planning of software mechanisms, routing methods, buffer sizes, et cetera – just to name some examples.

CHAPTER 4

The Chip Concept

After the general introduction into the physical context of this work, the goal of this chapter is to introduce the reader into the overall concept of the “system-on-a-chip” readout ASIC named SPADIC. Therefore, the SPADIC architecture and moreover the different considerations that have finally led to the present design and its principle of operation are summarized subsequently.

Due to the rather abstract character of this chapter most technical details are given elsewhere in this thesis (corresponding references are given).

4.1. The Abstract Task

The abstract task of a readout chain of a certain detector application is quickly described: the electrical signals from dedicated electrodes must be sensed, certain parameters of the signals (e.g. their amplitudes) must be extracted and finally the gathered information must be transported in a more convenient format to some data sink (e.g. hard disk). In doing so, the weak electric signals must be amplified, eventually converted into some digital representation, and usually combined with various kinds of meta data (e.g. time-stamps).

Figure 4.1.: The logo of the SPADIC readout ASIC.

The goal of this work was the development of a readout chip, which on the one hand would be able to perform each of the just mentioned steps on the same die, and which on the other hand would properly match the environmental conditions of the CBM physics experiment in general and the free-running CBM DAQ system (3.4) in particular. Moreover, and after an initial uncertainty where the path should actually lead, it was decided that the main application of the readout chip should be the CBM TRD sub-detector (3.3). The resulting chip design was eventually named **SPADIC**, which is an abbreviation for **S**elf-triggered **P**ulse **A**mplification and **D**igitization as**IC**. Although the chip characteristics has been adapted to CBM and the specific TRD requirements, the actual design concept and the various available features potentially make SPADIC an interesting option for many other readout applications as well.

Moreover, the output of this work is not only the mere ASIC design, but includes for instance the initial development of the readout concept, various hardware setups, software and firmware solutions, gathered technical know-how, countless measurements, practical detector readout experience, and not at least the establishment of the SPADIC label (the logo is shown in Fig. 4.1) and of the practically very important project website [11].

4.2. Constraints and Requirements

Before the SPADIC architecture can be further described, it is very important to understand all the requirements a TRD readout ASIC for CBM must generally fulfill in order to satisfy all the needs coming from the TRD physics, from the CBM detector setup as a whole, and from the DAQ system in particular. Subsequently the physical requirements, the beneficial features and the additional constraints are briefly discussed and summarized. For a more fundamental discussion on most ASIC design parameters see chapters 5 and 6.

4.2.1. Physical Requirements

This section summarizes what and how accurately must be measured with SPADIC.

4.2.1.1. Signal Amplitude

In detectors where the integral of the generated signal is proportional (or at least directly related) to the deposited energy of the incident particle, the amplitude is usually of high interest (or at least the mere information that it has crossed a certain threshold – binary readout). Because that is in in particular also true for the CBM TRD sub-detector (based on MWPCs working in proportional mode), the required **amplitude resolution** of the SPADIC is a central design parameter, which was and still is frequently discussed. An equivalent ADC value of 8 bit[1] so far seems to be a good compromise between the benefit for the TRD and the costs of the electronic. The TRD experts expect the average signal charge to be in the order of 19 fC, whereas the total input range (which must be accordingly mapped to the ADC output range) has been set to 75 fC so far.

[1] This refers to the equivalent ADC resolution, which is actually derived from both the amplifier noise and the effective ADC resolution. See also section 5.3.1.

For the TRD detector (as well as many other detector types in general), there are basically three major parameters that benefit from an increasing amplitude resolution: the energy resolution, the spacial resolution and the input range. Whereas for increasing the energy or the spacial resolution the equivalent ADC step height must be decreased (while fixing the total ADC output range), for increasing the ADC output range the total number of equivalent ADC bits must be increased (while fixing the step height). Unfortunately, the costs of a better resolution of the electronics are high: empirical investigations show for instance, that one can expect the power requirements of an A/D conversion at least to double with each additional bit of resolution [70] and theory proves it to even quadruple, if the only limitation is thermal noise [49]. And comparably, the power dissipation of the analog front-end roughly quadruples if the S/N ratio is doubled [66].

Whereas the theoretically reachable energy resolution or the output range is directly proportional (when adjusting the other system parameters properly) to the equivalent ADC resolution or the S/N ratio of the system, finding out about the dependency of the spacial resolution on the electrical system resolution is not as obvious. As an orientation, the argumentation given in the attachment (A.1.1) estimates the lower limit of the reachable spacial resolution to be also proportional to the amplitude resolution, as long as all other boundary conditions are ideal. Hence doubling the amplitude resolution helps at least theoretically to double either the energy and the spacial resolution or the input range – but for the costs of roughly four times higher power consumption, increased chip space, increased data rates (more bits per sample, this can lead to significant follow-up costs) and not to forget a much higher design effort and complexity.

When discussing the effective ADC resolution or the noise requirements, it is also very important to recognize that the final energy and spacial resolution of the TRD can also be modified by adjusting different detector parameters in the first place. For example, to a certain limit, the spatial resolution can be doubled by cutting the TRD pads into halves (for the costs of basically doubling everything, for instance the number of channels, the data rates or the power consumption), or the balance between energy resolution and input range can be modified (again only to a certain limit) by increasing the gas gain.

4.2.1.2. Arrival Time

Even though in most cases it leads to no direct physical conclusions (except for time of flight or drift time measurements), an extraction of the arrival time of the electric signals is essential for the complete reconstruction of collision events. That is especially true for the free-running CBM experiment, where no global trigger delivers time-markers, and where the collision rates are very high. Hence in order to be able to separate events, the time granularity of a certain sub-system must only be slightly smaller than the expected average period between two events. But due to statistical fluctuations of the event rates and moreover due to the uncorrelated occurrence of background signals, much smaller granularities are beneficial in many applications (e.g. due to the small number of photons per ring, RICH detectors suffer significantly from background photons and hence usually benefit from good time accuracies).

For the CBM sub-detectors, the absolute upper limit of the required **time resolution** of all detectors is roughly given by the maximum mean event rate of 10 MHz and therefore is about 100 ns. Because no really hard arguments for an even better time resolution of the TRD detector other than a better separation of single pulses (**double hit resolution**) exist, and because the complexity of the electronics rapidly increases for very small resolutions, already a relatively moderate value is expected to be sufficient, although no definitive value has been specified so far by the CBM TRD developers. As shown in 5.3.2, the latest SPADIC version currently achieves time resolutions < 1 ns in the laboratory, whereas the effective resolution in a full system under real conditions will probably be slightly worse.

4.2.1.3. Signal Shape

If one takes a closer look at the temporal evolution of the current signals in the TRD electrodes and does not only consider its integral, one can notify certain sub-structures which potentially contain additional physical information. As shown in 2.2.2.5, the TR component of the average signal measured with the ALICE TRD clearly shows up as a second peak at the end of the induced signal. In that case, an exact measurement of the whole signal shape would obviously allow for a more reliable electron/pion separation (even though also a slightly higher integrated amplitude would be observed). But because in contrast to the ALICE TRD the CBM TRD detectors are designed to have much shorter or even no drift regions, the current signals collapse to some 100 ns. As a direct consequence, the TR component is not any more clearly separated from the initial signal maximum (which is caused by ionization directly in the amplification region). Nevertheless, the CBM TRD experts still hope to benefit from a **complete record** of the signal, which the SPADIC chip allows for.

It is important to note here, that the sub-signal granularity of the recorded pulses is fundamentally limited by the **shaping time** constant of the front-end electronics, which can initially be chosen more or less arbitrarily, but normally affects such a variety of crucial design parameters (e.g. the maximal possible sampling rate or the reachable noise at a fixed power limit), that each later change might become very costly and potentially also requires the adaption of many other ASIC parts. Both is especially true for SPADIC 1.0, where the shaping time has been set to 80 ns so far (compare 5.1.3.2).

4.2.2. Additional Features and Constraints

This section lists beneficial features and essential constraints that lie beyond the most basic functionalities of the TRD readout chip.

As mentioned several times before, a very fundamental requirement of most components of the CBM experiment is the free-running mode of operation. There are two primary consequences for the readout ASICs: first a self-organizational synchronization strategy must be developed and second a lot of global responsibility and intelligence is shifted to the local chip context. In particular, that has a strong impact on two issues: first the chips must autonomously detect and select hits and should hereby avoid death-times as much as possible (for instance, that requires a **self-triggered** hit detection or a fast/continuous reset strategy of the preamplifier feedback). And second, because the front-end ASICs are effectively the

ultimate source of all data streams, they must not only provide and properly respond to several basic **synchronization mechanisms** (e.g. injection of sync markers, data formats adapted to re-synchronization, etc.), but should also be able to autonomously react to various unforeseen events (e.g. buffer overflows, single-event upsets, loss of synchronization, etc.), which are not so frequent in systems where the global trigger provides something like a general rhythm.

An important set of design parameters can be defined by selecting all SPADIC parameters that directly depend on the given detector (or TRD) electrode geometries (and which are consistently taken into account by the detector designers). These are the **number of channels** per chip, the **maximum hit rate** each channel must be able to handle, and the maximum **load capacitance** the preamplifiers have to fight against. Whereas a small pad size reduces the average hit rate per channel and the capacitive load (which also depends significantly on the capacities of cables and connectors though), a larger pad size relaxes the effectively required channel density. The CBM TRD pads are optimized to deliver a sufficient spatial resolution on the one hand, but to reduce the average hit rate per channel and the total number of required channels for the whole detector on the other hand. The latest design iteration of the CBM TRD has pad sizes of 1 to 12 cm^2, which adds up for the SPADIC to a maximum average hit rate per channel of 100 kHz. Moreover, 32 channels per chip seems to be convenient and one expects maximal load capacities (including wires and connector) of about 40 pF. According to the latest estimations a total of roughly 25 000 SPADIC chips will be required to read out the whole TRD.

A beneficial feature which is rather specific for the TRD is the so-called **ion-tail cancellation**. An ion-tail cancellation filter (see 6.2.3) can be used to remove slow signal components of the TRD charge signals, which occur due to the long-lasting influence of back-drifting avalanche ions in MWPCs (see 2.2.2.5). The cancellation of ion-tails effectively helps to reduce pile-up effects, indirectly stabilizes the baseline, and helps to distinguish the single signal contributions of a recorded double or multi hit.

The feature of **neighbor readout** is an effective method to allow for higher thresholds by force-triggering neighboring channels, if they have not recognized a certain weak signal component by themselves. In the CBM TRD, as well as in many other detectors, the overall geometry is chosen such that the mean signal charge spreads over several electrodes. This can be exploited to improve the spacial resolution (discussed in appendix A.1.1). A properly configured neighbor relationship allows to set the trigger threshold such that only the charges gathered by the middle pads within an affected group are directly detected, whereas the charges induced into the border pads are read out only indirectly and due to the enforced readout by the neighbors. That scheme allows to set the detection threshold relatively high and hence helps to significantly reduce the noise hit rates, from which in particular large detector systems suffer strongly.

Moreover, a long list of smaller but important features and constraints exists, which can be summarized casually as “the requisite things”. The most crucial of them are **spark protection** (protection of the sensitive preamplifiers inputs, see 5.1.2.4), **radiation tolerance** (see 5.5 and 6.6), **low power** (a relatively moderate constraint in the case of the CBM TRD, where the spacing is rather relaxed), **low noise** (see 5.1.2.2), **testability** (the importance of test-features, back-doors and accessibility to key signals can be easily underestimated,

see 6.5.3, 5.1.3.4 or 5.2.3.5), **reliability** (e.g. shut-down of broken channels, reliable reset strategy or status reports, see for instance 6.3), **compatibility** (as mentioned before the ASIC must fit into the CBM DAQ concept), and **flexibility** (actually a general goal of the SPADIC design).

4.2.3. Summary Table: Constraints and Features

Parameter	Specified or expected value
Channels / chip	32
Power limit / channel	50 mW (rough estimation)
Energy resolution	8 bit
Input capacity	max. 40 pF, typ. 20 pF
Required shaping time	80 ns
Typical input charge	120 ke = 19.2 fC
Input range	75 fC
Maximum hit rate per channel	100 kHz
Front-end polarity	positive
Required features	self-triggered readout
	synchronization strategies
	compatibility with CBM DAQ
	spark protection
	moderate radiation tolerance
Beneficial features	ion-tail cancellation
	neighbor readout
	complete signal recording

Table 4.1.: Summary of the essential constraints and features of the SPADIC chip. Parameters that are not explicitly listed here are either not very important or depend indirectly (e.g. the preamplifier noise should stay significantly below an LSB of the effective ADC resolution, or the sampling frequency only must be high enough). More and updated details can always be found on the website [11].

4.3. The SPADIC Architecture

After the short discussion of what the SPADIC should be able to do and why, this section focuses on the actually realized architecture and gives some arguments why things have been designed that way.

4.3.1. Important Design Decisions

At the beginning of this work neither a chip concept existed nor had it been even clear yet for which CBM sub-detector(s) the new readout chip should be designed. Indeed, the CBM silicon tracker (STS) was initially considered as the main chip application. It was also

completely vague, which of the initially participating groups should design which part(s) of the ASIC – that by the time has been called CBM-XYTER. But after an initial phase of defining and distributing working packages, and after actually no concrete results from either side, the CBM-XYTER designer community eventually collapsed – and thus the CBM-XYTER alias SPADIC became the main purpose of this work.

At that time, the CBM DAQ system was only coarsely sketched and no concrete concepts or even specifications had yet existed. Hence, since both most details of the DAQ and the actual detector application were unclear, it was neither known what signals the readout chip would get (signal range, time-scales, etc.) nor was defined which data should be extracted and how it should be transported.

Before it eventually turned out that the main application would be the TRD sub-detector, some first chip prototypes with variants of preamplifiers and respective test-setups were designed and characterized (SPADIC 0.0x, see further below). It was decided early that the UMC 180 nm technology should be used, mainly due to two crucial arguments: First, a good intrinsic radiation-tolerance of the process was required, which argued against a smaller scale integration (also compare 5.5). And second the existence of some previously (to the beginning of this work) designed and tested building blocks (basically a forerunner of the pipeline ADC later actually used in the SPADIC, io-pads and different current DACs) made the UMC 180 nm technology particularly attractive.

An early and central design decision was whether to multiplex some few fast ADCs between the analog channels or to provide one slow ADC for each channel instead. In principle both options were directly available (the home-made pipeline ADC mentioned before and a fast switched capacity ADC that was developed in a pilot study for CBM [50]). At the end it was decided to have one slower ADC in each channel, because the gain of flexibility and scalability for the relatively low price of the very small-size and low-power pipeline ADC (see 5.2) considerably prevailed the argument of a potentially better utilization (and all corresponding benefits) of the fast ADC solution. That decision, moreover, also arose the possibility to record complete pulses and in addition to that provided the opportunity of continuous signal processing – two features that eventually became characteristic for SPADIC.

Roughly at that point in time it became evident that the main application of the readout chip would be the CBM TRD and hence new concrete constraints as well as beneficial features came up. That quickly led to the important question, what mechanisms should actually be provided, and more crucial, whether they should be realized in the analog (before the ADC) or the digital (behind the ADC) domain. Again and not at least for the gain of flexibility, the decision was made to digitize the analog signals as early as possible, hence directly after amplification/shaping. That, however, had the direct consequence for the self-triggered ASIC, that the digital part would face continuous data streams and that the trigger logic and hit building mechanisms had to be shifted to the digital domain. Moreover, the resulting dominance of the digital part meant an additional challenge for the mixed-signal multi-channel SPADIC, which carries both ultra sensible analog parts and a huge digital block with several clock domains on the same die.

Moreover, the present SPADIC architecture also stepwise developed from many smaller, although sometimes crucial design decisions, numberless practical results, and evolving

physical requirements. Those are not further listed here, but are mostly described in the subsequent sections and chapters.

4.3.2. Data Flow Through the Chip

The coarse SPADIC architecture (of the latest version 1.0) can probably be best described by "following" the incoming charge signals step by step through the whole processing chain – although that way some also interesting parts are passed over. Hence subsequently, the data flow through the chip is shown.

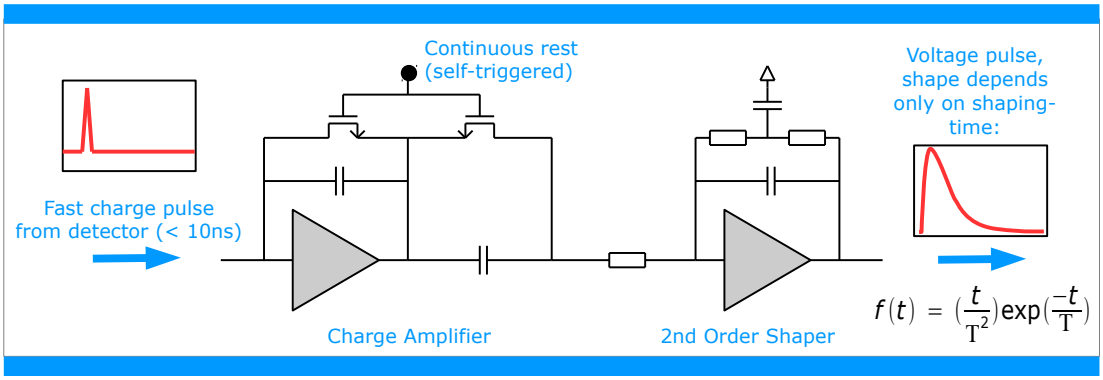


Figure 4.2.: Part 1, front-end: from detector to amplifier output.

Part 1, front-end (figure 4.2, details in section 5.1): After passing the spark protection diodes in the input pads (not shown), the short current pulses from the detector electrodes charge the feedback capacity of the preamplifier, which immediately but very slowly starts to discharge again (continuous reset). Thus a proportional voltage step at the preamplifier output is produced (which again very slowly decreases due to the continuous discharge). The voltage step is then shaped with a characteristic time constant (shaping time τ so far 80 ns) due to the special feedback structure of the second amplifier, until a proportional pulse ($t/\tau^2 \cdot exp(-t/\tau)$) emerges at the shaper output.

The latest SPADIC iteration actually has two parallel implementations of the single-ended preamplifier/shaper front-end, one for each charge polarity. Both circuits nearly have the same structure, parameter set and functionality. Even though the CBM TRD electrodes deliver positive charges, the second front-end has been added to increase the number of potential applications (e.g. readout of negative charges from PMTs used in the CBM RICH).

Part 2, pipeline ADC (figure 4.3, details in section 5.2). After shaping, the voltage pulse is converted into a current (via a serial resistor) and fed into the current-mode pipeline ADC. An additional current source at the ADC input (not shown) allows to trim the input baseline of each channel individually. From the input current, the pipeline ADC continuously produces 8 bit (effective) words at a nominal sampling period of 40 ns.

Part 3, stream processing (figure 4.4, details in sections 6.2 and 6.3). The continuous digital stream from the ADC first passes an IIR filter (4 stages), which can be configured to perform for instance the following tasks: ion-tail cancellation, baseline correction/stabilization, undershoot correction, signal inversion (required particularly for the negative

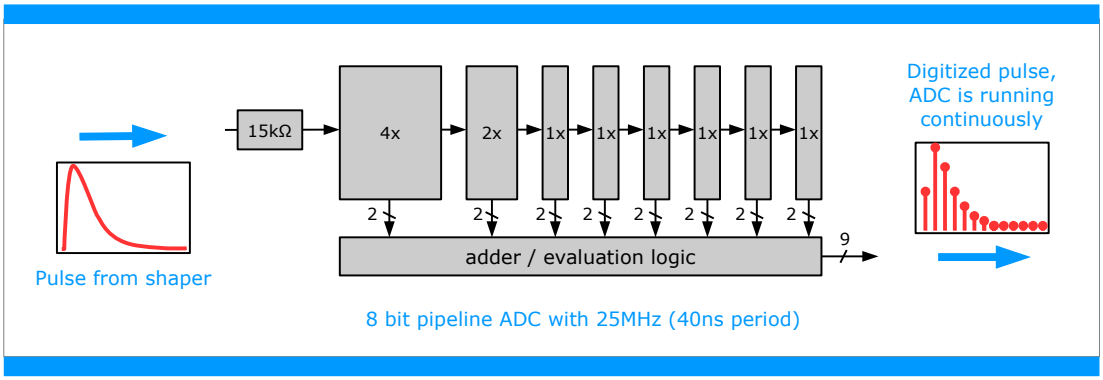


Figure 4.3.: Part 2, pipeline ADC: through the pipeline ADC.

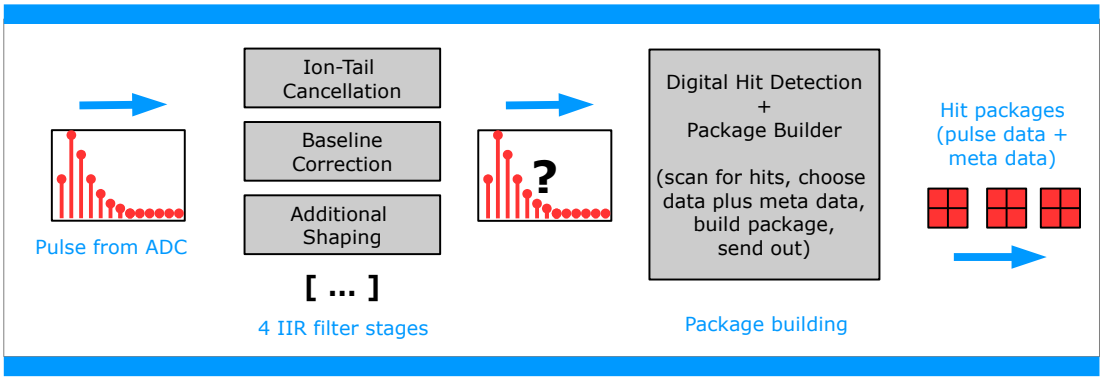


Figure 4.4.: Part 3, stream processing: from the ADC output to end of the channel output FIFO.

front-end), or additional shaping. The overall function of the IIR filter can abstractly be summarized as being a flexible adaption mechanism to the non-ideal characteristics of both the detector and the analog front-end.

Then the subsequent hit detection and message building logic continuously searches the output stream coming from the IIR filter for pulses (hits), optionally informs neighbors (neighbor logic), selects interesting values of the stream, combines them with lots of meta data (e.g. time-stamp, status bits or channel-id), generates hit messages (made of 16 bit-words), and temporarily stores them in the channel output FIFO.

Part 4, inter-channel data transfer (figure 4.5, details in section 6.4). Eventually, the messages stored in the channel output FIFOs of 16 channels are combined to one single message stream (together with epoch messages waiting in a dedicated 17th epoch channel, not shown). The arbiter hereby assures that the resulting message stream is ordered in terms of the initial time-stamp corresponding to and stored in the messages.

The merged and sorted message streams are finally fed into a transport and synchronization logic block implementing the CBMnet protocol. The CBMnet protocol is intended to be the overall CBM DAQ protocol and will be operated throughout the DAQ network. The protocol is able to distribute data, control, and synchronization messages. In the CBMnet

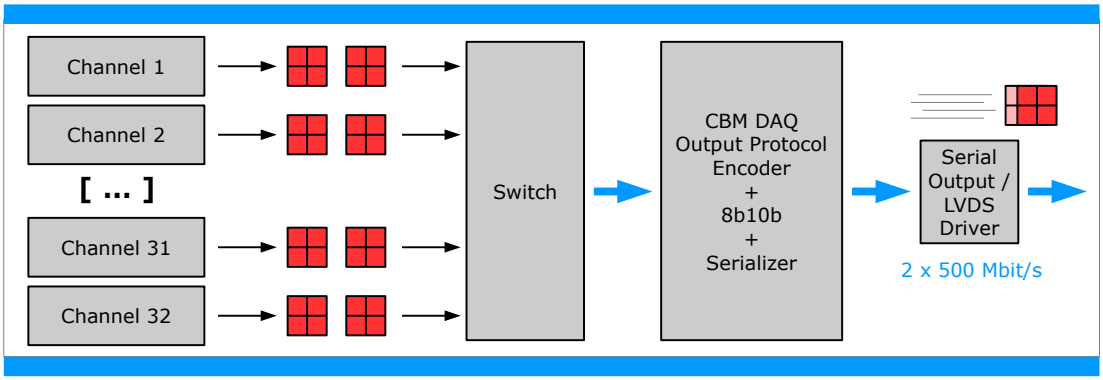


Figure 4.5.: Part 4, inter-channel data transfer: from the channel output FIFOs to the output serializers.

logic the message streams are 8b/10b encoded and cut into transport packages. Finally, the packages are serialized and sent out of the chip (over two 500 Mbit/s DDR LVDS links).

4.4. A Brief Chip and Setup Summary

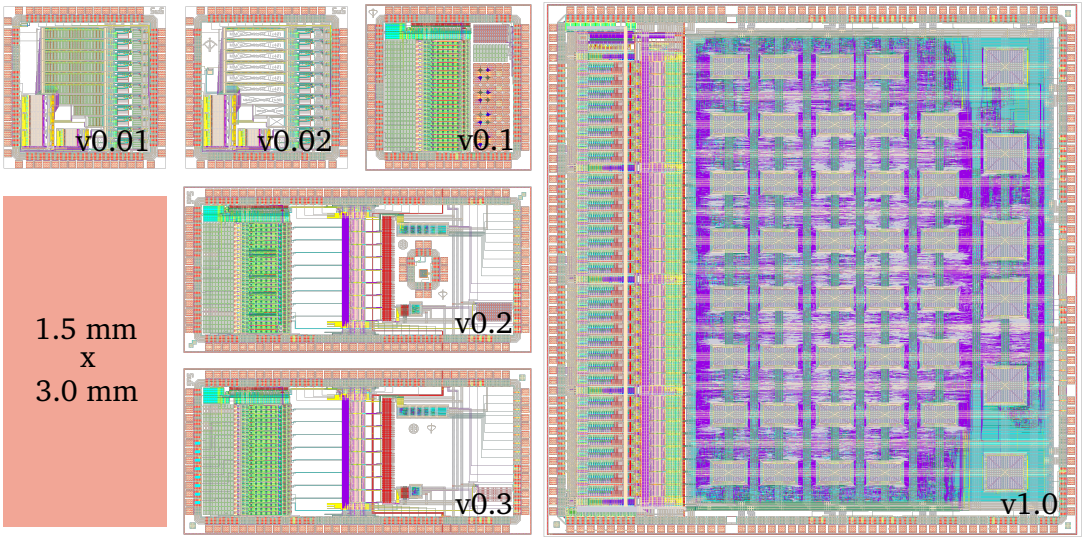
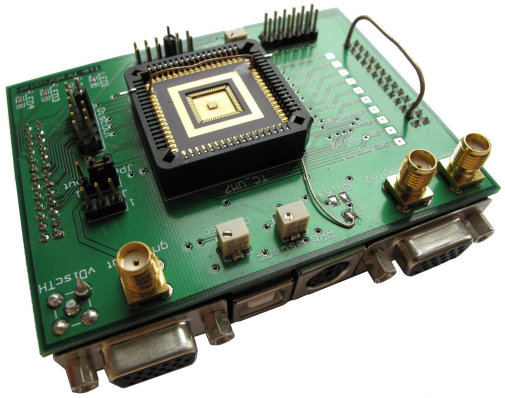


Figure 4.6.: Layouts of all 6 SPADIC versions.

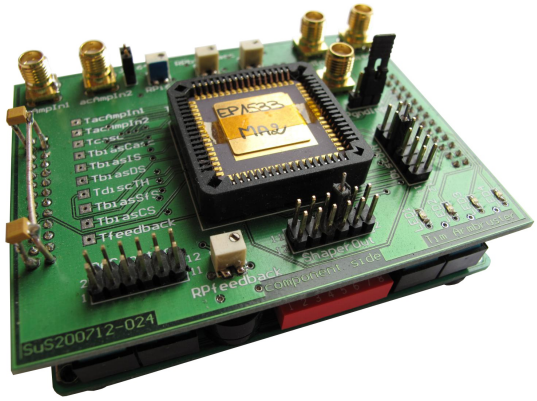
In the course of this work a total of 6 ASICs and 10 setups (plus software and firmware) was developed and tested (the ASICS are shown in Fig. 4.6)[1]. To keep things simple, this document only describes certain details of the last two chip iterations SPADIC 0.3 and 1.0 – except for this section, which at least gives a visual summary of the complete hardware

[1]Actually, the first ASIC, that was both called SPADIC and had a version number, was SPADIC 0.3. The earlier chips were named retrospectively and initially called TC_UM7 (v0.01), TC_UM9 (v0.02), TR1 (v0.1) and TR2 (v0.2).

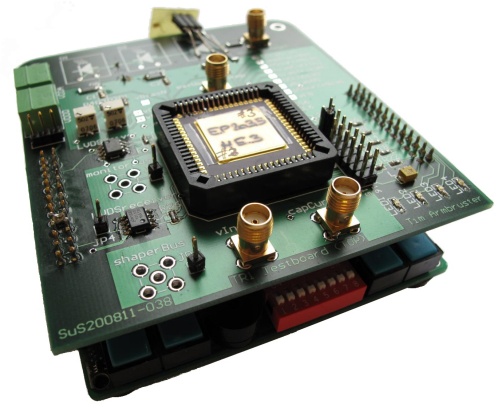
output. Hence subsequently photos of all important ASIC/setup combinations, which were built, operated, and characterized, are shown. For comprehensive information about the two latest setups of SPADIC 0.3 and 1.0 see chapter 7.



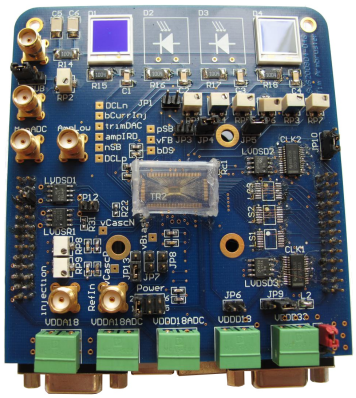
SPADIC 0.01: 1.5 x 1.5 mm^2. 10 preamplifier/shaper channels with discriminators. Design parameters and layouts of the channels were varied to test for their impact on noise. 4 current DACs (12 bit) used for bias generation.
Setup: front-end PCB connected to homemade multi-purpose FPGA board "Uxibo" [35]. Connected to PC via USB.



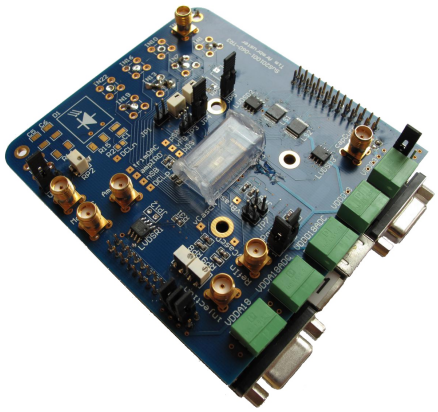
SPADIC 0.02: 1.5 x 1.5 mm^2. Architecture very similar to v0.01 (10 preamplifier/shaper channels each with discriminator). More parameter variations. Slightly improved layouts. Again mainly designed to address noise performance. Additional calibration structures.
Setup: front-end PCB plugged onto Uxibo.



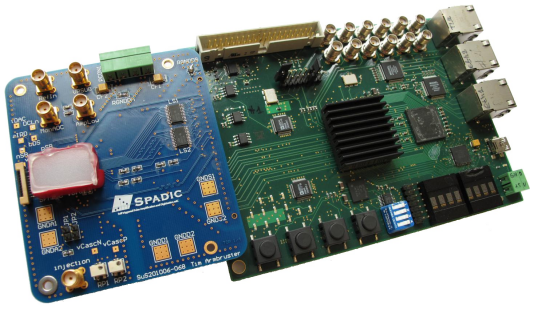
SPADIC 0.1: 1.5 x 1.5 mm^2. 26 preamplifier/shaper channels with discriminators. Completely new and compact layout. Shaper with new feedback scheme. Advanced test injection cell. First LVDS outputs. 14 current DACs (7 bit) for bias generation. Bump bonding test structures.
Setup: PCB with some additional logic ICs. Si photo-diodes. Readout with Uxibo.



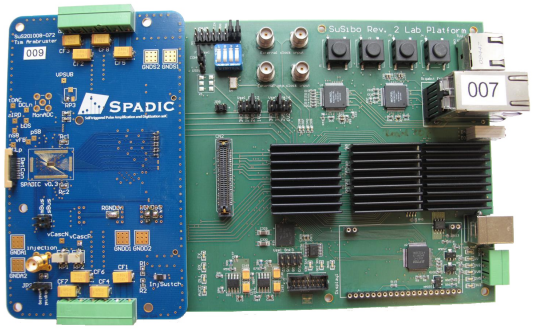
SPADIC 0.2: 3.2 x 1.5 mm^2. Front-end similar to v0.1 (26 CSA channels with discriminators), but now with digital back-end and 8 pipeline ADCs. Two synthesized digital blocks (home-made standard cell library). Compact shift-register buffer matrix. Test structures.
Setup: various level-shifter and logic ICs on PCB. Si photo-diodes. Readout with Uxibo. Many separate power connectors.



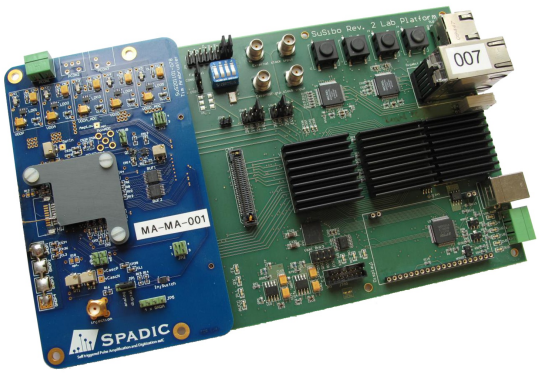
SPADIC 0.3 [13]: 3.2 x 1.5 mm^2. Similar to v0.2 but dedicated and optimized for a first TRD beam-time at CERN. 8 complete CSA/ADC channels with spark protected input pads (still 26 CSA channels). Smaller bug-fixes.
Setup 1: PCB for lab-tests only. Several level-shifters and logic ICs. Last setup using the Uxibo. Still too many different power connectors.



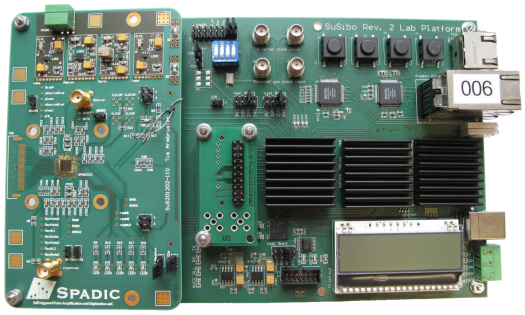
SPADIC 0.3, Setup 2: new PCB to connect to the new home-made "Susibo 1.0" prototype (multi-purpose Xilinx Virtex 5 FPGA board). Main reasons for the new board: Uxibo setup relatively noisy (if PCBs are stacked), Uxibo system clock too slow, FPGA size very limited (data buffering), and improper signal levels.



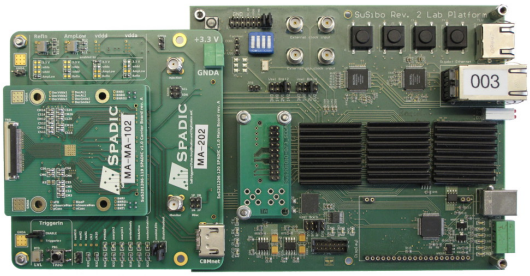
SPADIC 0.3, Setup 3: first TRD test-beam setup. Adapted to connect next board iteration Susibo 2.0 (smaller bug fixes). Adjusted to the form factor of the TRD chambers. Extra large ground pads. Reduced number of required power cables. Board to board power chain – actually never used due to ground loop problems. 8 setups were operated successfully at CERN end 2010.



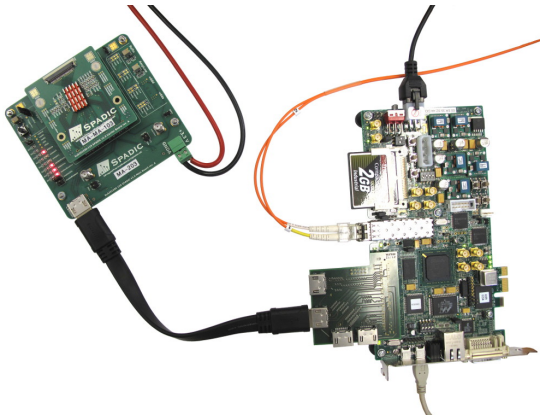
SPADIC 0.3, Setup 4: problems with pickup noise when connecting chambers were successfully addressed (e.g. more compact die footprint, separation of power planes, avoided common return paths, better separation of power domains, or massive use of voltage regulators). Only one power cable per front-end PCB required. 12 setups were operated at CERN end 2011.



SPADIC 1.0 [12]: 4.96 x 4.96 mm^2. First SPADIC iteration with nearly all required features. 32 complete mixed-signal channels (CSA, ADC, IIR, hit detection, message building and buffering). Dedicated output protocol CBMnet 2.0.
Setup 1: first setup for lab-tests. Conceptually also ready for a first test-beam. Readout with Susibo 2.1 via USB.



SPADIC 1.0, Setup 2, Option 1: readout with Susibo 2.1. Communication via CBMnet protocol possible for the first time. Stand-alone setup variant mainly for the laboratory. Die on a separate carrier-PCB for more flexibility and easier assembly/bonding.



SPADIC 1.0, Setup 2, Option 2: dedicated for the beam-time at CERN end 2012. CBMnet link via HDMI to Xilinx Spartan 6 SP605 evaluation board. From there via fiber optics to Xilinx Virtex 5 PCIe development board (not shown). First setup directly integrated into a CBM DAQ sub-system.

CHAPTER 5

The Analog Part

This chapter comprises all analog components and topics that are related to the SPADIC design. The focus is set on a relatively high level of detail and includes theory as well as results from simulations and measurements.

Since the latest version SPADIC 1.0 does not only provide the largest feature set, but also contains most of the important analog components from the previous chip versions (though mostly in a modified version), all subsequent descriptions relate to SPADIC 1.0, as long as not noted otherwise.

5.1. Charge Sensitive Amplifier

As shown in chapter 2, almost all kinds of particle detectors produce very small and short electrical output signals as a response to particle interactions. These extremely weak signals must usually be amplified before they can be further processed and analyzed. Therefore the first building block of a readout chain normally is a dedicated amplifier highly optimized for its respective application. In general the amplifiers can be designed either to sense voltages (high input impedance) or currents (low input impedance). In applications like the TRD, where a small and short charge pulse is induced into the cathode pad, current integrators – in this context also called charge sensitive amplifiers (CSA) – are the by far most frequently used variant of current sensitive (pre-)amplifiers.

It is common practice to connect the preamplifier output to a chain of low-pass filters, which are used to cut out high frequency regions where the noise power density exceeds the signal power density. In doing so the effective signal to noise (S/N) ratio can be significantly improved. Since a bandwidth limitation also significantly changes the time characteristics of the amplified signal, the chain of low-pass filters is usually called shaper.

5.1.1. General Principle

The most basic knowledge required to understand the principle of operation of a CSA is summarized in this section.

5.1.1.1. Preamplifier

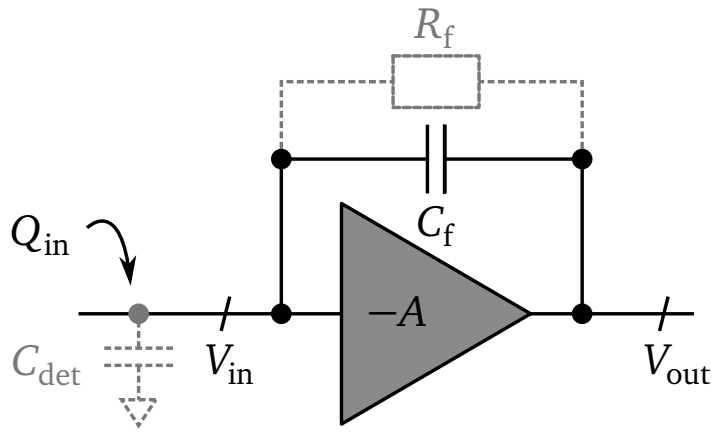


Figure 5.1.: Simplified schematic of a typical preamplifier (integrator) with capacitive feedback C_f, reset resistor R_f and load capacity C_{det}.

As sketched in Fig. 5.1 the most basic charge preamplifier only consists of a feedback capacity C_f and an inverting (voltage) amplifier with gain $-A$. In the absence of parasitic capacities, the input charge Q_{in} must completely flow onto the feedback capacity, hence $Q_{in} = (V_{in} - V_{out}) \cdot C_f$. That, together with $V_{out} = -AV_{in}$, leads to the effective input capacity

$$C_{in} = \frac{Q_{in}}{V_{in}} = \frac{(V_{in} - V_{out})C_f}{V_{in}} = (1+A)C_f, \tag{5.1}$$

which obviously benefits[1] from a high amplifier gain (Miller effect). The input capacity normally competes with the parasitic detector (or load) capacity C_{det}, which represents the effective capacitance seen from the input node to ground, if the DC potential is hold but the preamplifier feedback removed. Because the total input charge is divided between C_{in} and C_{det}, the amplifier only senses the reduced signal charge

$$Q_s = Q_{in}\frac{C_{in}}{C_{in} + C_{det}} = \frac{Q_{in}}{1 + C_{det} \,/\, (1+A)C_f}. \tag{5.2}$$

For that simple reason, the achievable S/N ratio suffers from large parasitic (or detector) capacities. Moreover, the equation demonstrates again why high-gain amplifiers are beneficial.

The voltage signal at the preamplifier output calculates as

[1] A high input capacity corresponds to a low input impedance for fast signals, which is just required here.

$$V_{out} = (V_{out} - V_{in}) + V_{in} = \frac{Q_s}{C_f} - \frac{V_{out}}{-A} \Rightarrow V_{out} = \frac{Q_s}{C_f}\frac{A}{1+A} \approx \frac{Q_s}{C_f}. \tag{5.3}$$

Therefore, if the amplifier gain A is sufficiently large ($A \gg 1$), the preamplifier responses to the effectively deposited input charge Q_s with a proportional voltage output step.

To prevent an overflow the charge on the feedback capacity must eventually be removed again. This can be done either instantaneously (via a switch in the feedback) or continuously. Switches are usually used if a cyclic readout strategy is intended or if the distance between two events is known to be relatively large. But in the free-running CBM experiment, where the incoming hits are Poisson distributed, a continuous reset strategy is certainly the better solution. A continuous reset in principle allows for zero death-times, even though pile-up can occur.

There are various techniques to realize a continuous feedback. The usual options can be divided into passive (e.g. a resistor) and active (e.g. constant current, Krummenacher scheme [43] or ICON cell [25]) circuits, which address different design goals. A constant discharge for instance can be used to extract the signal amplitude indirectly by measuring the time of discharge (time over threshold, TOF) or the ICON cell allows to build preamplifier/shapers with very large time constants. But because the TRD has no exceptional requirements in this regard, the SPADIC preamplifier uses a simple solution: a feedback resistor R_f[1] in parallel to C_f. R_f is also depicted in Fig. 5.1 (gray).

The idealized transfer function of the SPADIC preamplifier (or its Laplace transform) can be derived from equation 5.3 by substituting the capacitive feedback $1/C_f$ with the frequency depend impedance $Z_f = (1/sC_f) \parallel R_f$. This leads to

$$H_{preamp}(s) = \frac{V_{out}}{Q_s} = \frac{1}{sC_f} \parallel R_f = \frac{R_f}{1+s\tau_f}, \tag{5.4}$$

with $\tau_f = R_f C_f$ the characteristic time constant of the feedback. This can be easily transformed to the time domain (assuming Q_s to be a perfectly short impulse, or a Dirac delta respectively):

$$V_{out}(t) = Q_s \, \mathscr{L}^{-1}\{H_{preamp}(s)\}(t) = \frac{Q_s}{C_f} e^{-t/\tau_f}. \tag{5.5}$$

This equation shows that the feedback resistor exponentially discharges C_f – with a rate directly defined by the time constant τ_f. $V_{out}(t)$ is plotted for different values of τ_f in Fig. 5.2. Typically but not necessarily, τ_f is chosen to be much larger than the time constant of the subsequent shaper (see further below).

5.1.1.2. Shaper

As mentioned before, a shaper is usually connected to the preamplifier in order to remove predominantly noisy frequency components and thus to improve the effective S/N ratio. Whereas on the one hand a low cut-off frequency (or a large shaping time τ_s) can help to

[1] Actually the resistor is realized with a MOSFET, see 5.1.3.2

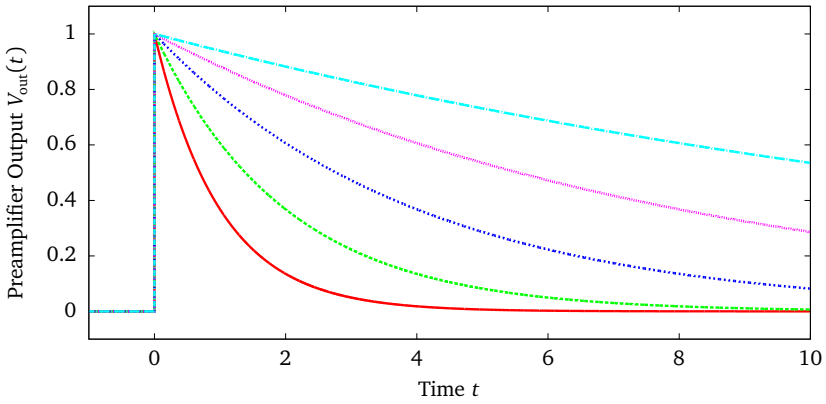


Figure 5.2.: Ideal output signal (impulse response) of the preamplifier sketched in Fig. 5.1 for different time-constants with arbitrary units.

reduce the amplitude noise (actually a minimum might exist, see noise section 5.1.2.2), the achievable time resolution on the other hand gets worse (see time section 5.1.2.3).

Normally, the shaper consists of one high-pass (HP) stage followed by N low-pass (LP) stages, which is commonly expressed as "CR-RC^N" shaper. But this term can be slightly confusing: if one speaks of a shaper of the order N one normally refers to a "CR-RC^N shaper reacting to a unit step pulse $\mathscr{L}\{\text{step}(t)\}(s) = \frac{1}{s}$ (which is basically the preamplifier output). But in this case the effective transfer function of the whole system (preamplifier and shaper) is

$$\begin{aligned} H_{\text{PRE-CR-RC}^N}(s) &= \frac{1}{s}\, H_{\text{HP}}(s)\, (H_{\text{LP}}(s))^N = \frac{1}{s}\frac{s\tau_{\text{HP}}}{1+s\tau_{\text{HP}}}\left(\frac{1}{1+s\tau_{\text{LP}}}\right)^N \\ &= \frac{\tau_{\text{HP}}}{1+s\tau_{\text{HP}}}\left(\frac{1}{1+s\tau_{\text{LP}}}\right)^N \\ &\hat{=} \frac{\tau_{\text{s}}}{(1+s\tau_{\text{s}})^{N+1}} \quad \text{, if} \quad \tau_{\text{HP}} = \tau_{\text{LP}} = \tau_{\text{s}}, \end{aligned} \tag{5.6}$$

and has obviously $N+1$ poles and not only N, especially for the manifest choice $\tau_{\text{HP}} = \tau_{\text{LP}} = \tau_{\text{s}}$.

Using the latter equation, the actual output as a function of time of a CR-RC^N shaper (with $\tau_{\text{HP}} = \tau_{\text{LP}}$) can be calculated (a more detailed derivation can be found for instance here [60]):

$$V_{\text{out}}(t) = \frac{Q_{\text{s}}}{C_{\text{f}}}\mathscr{L}^{-1}\left\{\frac{\tau_{\text{s}}}{(1+s\tau_{\text{s}})^{N+1}}\right\}(t) = \frac{Q_{\text{s}}}{C_{\text{f}}N!}\left(\frac{t}{\tau_{\text{s}}}\right)^N \mathrm{e}^{-t/\tau_{\text{s}}}, \tag{5.7}$$

with peaking time $t_{\text{peak}} = N\tau_{\text{s}}$ and maximum $V_{\text{max}} = \frac{Q_{\text{s}}}{C_{\text{f}}N!}(\frac{N}{\mathrm{e}})^N$. This result is visualized in Fig. 5.3, showing the step responses of different CR-RC^N shapers and the corresponding frequency spectra. The left plot shows how the pulses become smaller and broader, if the

shaping order is increased but the shaping time stays fixed. That behavior is explained by the Bode plot on the right hand side, which demonstrates how the effective cut-off frequency decreases while the cut at the same time becomes sharper and sharper.

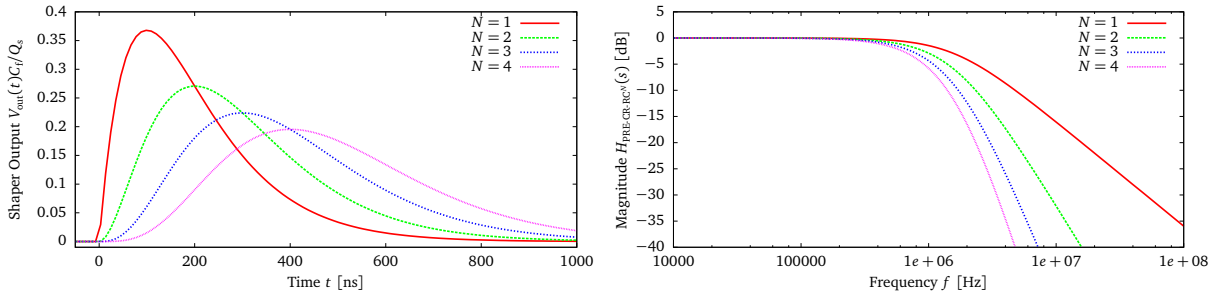


Figure 5.3.: Shaper step response (left) and magnitude of the transfer function (right) of CR-RC^N shapers with different orders but the fixed shaping time $\tau_s = 100\,\text{ns}$.

For analytic reasons it is useful to normalize the transfer function both to the peaking time and the maximum. In doing so the equations becomes pretty simple:

$$V_{\text{out}}^{\text{norm}}(t) = \frac{V_{\text{out}}(t \cdot t_{\text{peak}})}{V_{\text{max}}} = \text{e}^N(\ t\text{e}^{-t}\ )^N. \tag{5.8}$$

This result shows more fairly the influence of the shaper order on the pulse shape: as plotted in Fig. 5.4 (left: shaper output, right: corresponding Bode plot), a higher shaper order can help to get more narrow pulses, but at the same time demands a proper downscaling of the shaping time, which can cause significant consequences in general (e.g. higher noise).

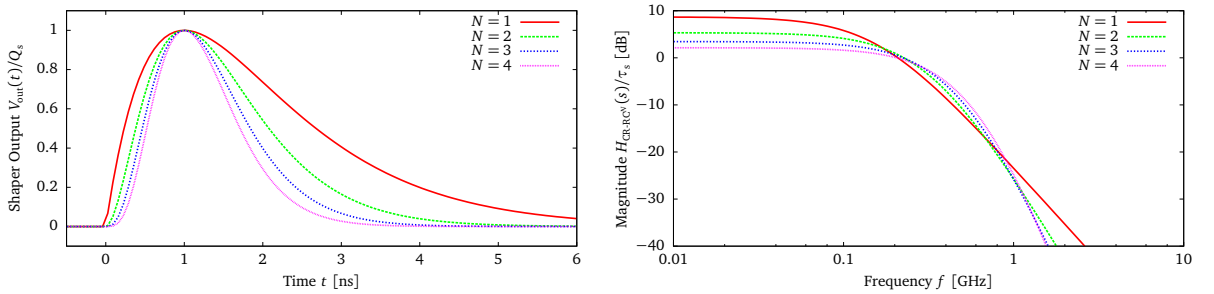


Figure 5.4.: Normalized shaper step response (left) and magnitude of the transfer function (right) of CR-RC^N shapers with different orders N and properly adapted shaping times ($\tau_s = 1, \frac{1}{2}, \frac{1}{3}$, and $\frac{1}{4}$ ns for $N = 1, 2, 3$, and 4).

5.1.2. General Aspects of the Implemented Front-End

A summary of the more general aspects of the front-end implemented in SPADIC 1.0 and 0.3 is given in this section, whereas rather specific details are discussed in section 5.1.3.

5.1.2.1. Transfer Function

A simplified schematic of the preamplifier/shaper circuit implemented in SPADIC 1.0 and 0.3 is given in Fig. 5.5.

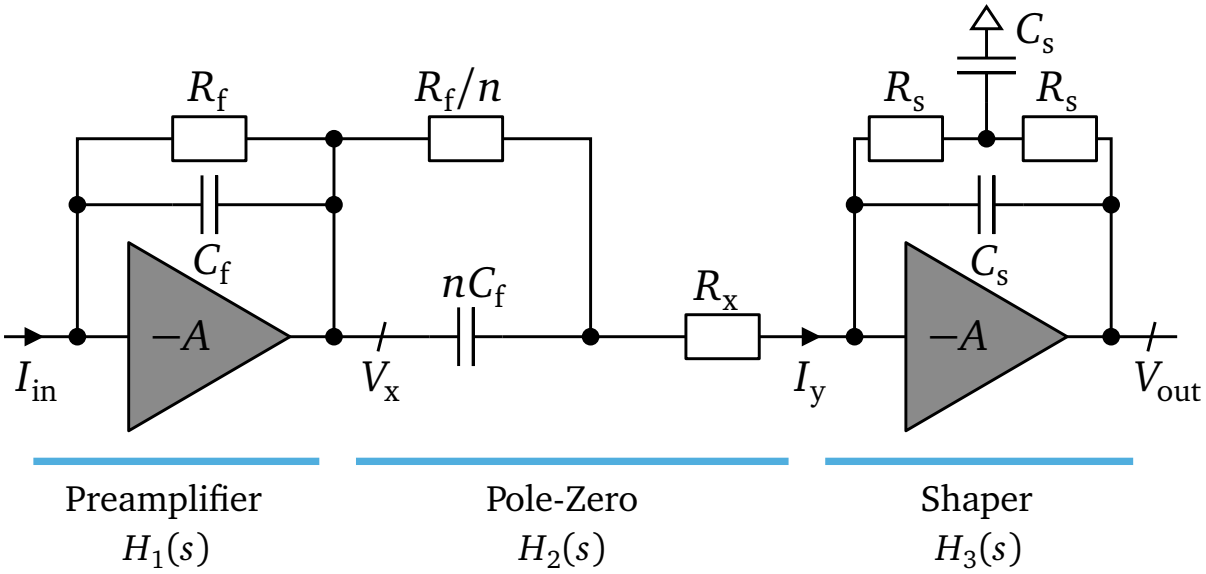


Figure 5.5.: Simplified schematic of the preamplifier/shaper circuit used in SPADIC 1.0 and 0.3.

Similar to the preamplifier transfer function $H_1(s) = \frac{V_x}{I_{in}}$, that has already been calculated (see equation 5.4), $H_2(s)$ can be also easily derived, if one assumes the voltage amplifiers to be ideal (infinite gain and bandwidth):

$$\begin{aligned} H_2(s) = \frac{I_y}{V_x} &= \left(\frac{R_f}{n} \parallel \frac{1}{snC_f} + R_x\right)^{-1} = \frac{n(1+s\tau_f)}{(R_f + nR_x)(1+sC_f(nR_x \parallel R_f))} \\ &\approx \frac{n(1+s\tau_f)}{R_f(1+snC_fR_x)} \quad \text{, if} \quad nR_x \ll R_f. \end{aligned} \tag{5.9}$$

Likewise $H_3(s) = \frac{V_{out}}{I_y}$ can be calculated (with $\tau_s = R_sC_s$) as[1]

$$\begin{aligned} H_3(s) = \frac{V_{out}}{I_y} &= \frac{1}{sC_s} \parallel \left(sC_sR_s^2 + 2R_s\right) \\ &= \frac{2R_s(1+s\tau_s/2)}{(1+s\tau_s)^2}. \end{aligned} \tag{5.10}$$

Finally, the transfer function of the complete preamplifier/shaper circuit becomes

[1] For an easier analysis the T-feedback (or star configuration) can be formally transformed into an equivalent delta configuration, which is not explicitly shown here.

$$
\begin{aligned}
H(s) &= \frac{V_{\text{out}}}{I_{\text{in}}} = H_1(s)\cdot H_2(s)\cdot H_3(s) \\
&\approx \frac{R_\text{f}}{1+s\tau_\text{f}}\cdot\frac{n(1+s\tau_\text{f})}{R_\text{f}(1+snC_\text{f}R_\text{x})}\cdot\frac{2R_\text{s}(1+s\tau_s/2)}{(1+s\tau_\text{s})^2} \quad \text{, if} \quad nR_\text{x}\ll R_\text{f} \\
&= \frac{n}{(1+snC_\text{f}R_\text{x})}\cdot\frac{2R_\text{s}(1+s\tau_s/2)}{(1+s\tau_\text{s})^2} \quad \text{, if} \quad nR_\text{x}\ll R_\text{f} \\
&= \frac{2nR_\text{s}}{(1+s\tau_\text{s})^2} \quad \text{, if} \quad nR_\text{x}\ll R_\text{f} \;\text{ and }\; nC_\text{f}R_\text{x}=\tau_s/2.
\end{aligned}
\tag{5.11}
$$

That means that if the latter two conditions are fulfilled by proper design, the complete analog front-end behaves exactly like a second order low-pass, and moreover, it corresponds to a system consisting of a preamplifier and a simple CR-RC shaper. Therefore the front-end has the impulse response (by comparing equations 5.7 and 5.11)

$$
V_{\text{out}}(t) = Q_\text{s}\,\frac{2nR_s}{\tau_s^2}\;t\;\text{e}^{-t/\tau_\text{s}}, \tag{5.12}
$$

which peaks at τ_s and reaches the maximum $\frac{2n}{\text{e}}\frac{Q_\text{s}}{C_\text{s}}$ (curve $N=1$ in Fig. 5.3).

5.1.2.2. Noise

The effective S/N of a front-end for detector applications becomes a crucial parameter, as soon as a good energy and/or time resolution is required. For practically relevant circuits, exact noise calculations are very difficult or even impossible to achieve, hence one normally relies either on simulations or on mathematical approximations. Whereas simulations can provide reasonably reliable results very quickly, they normally give little insight to the fundamental interrelations. Analytic calculations on the other hand can potentially provide a deep understanding of the circuit, but tend to be inaccurate and unreliable. Therefore a rough calculation in combination with an accurate simulation usually offers the best trade-off between benefit and effort.

To estimate the noise in the particular case of the SPADIC front-end, one can make use of the general approximation for an ideal CR-RC preamplifier/shaper (with $\tau_\text{CR}=\tau_\text{RC}=\tau_\text{s}$) that is derived in several textbooks (e.g. [66] or [60]):

$$
\text{ENC}^2 = \sigma^2_{\text{noise}} = \frac{\text{e}^2}{8}\left(a_n^2\;\tau_\text{s} + b_n^2\;\frac{C_\text{det}^2}{\tau_\text{s}} + c_n^2\;C_\text{det}^2\right), \tag{5.13}
$$

with ENC (equivalent noise charge) the amount of input charge corresponding to $S/N=1$, C_det the total load capacity at the preamplifier input node (detector capacity, gate capacity of the preamplifier, etc.), and a_n^2, b_n^2 and c_n^2 the (quadratic) sums of different noise contributions presented to the input of the preamplifier. More precisely, a_n^2 represents white current noise components (also called parallel noise), b_n^2 white voltage noise components (also

called series noise), and c_n^2 noise components having a 1/f (pink) density spectrum. Corresponding to the formula a minimal ENC for a certain shaping time τ_s exists, but however in the case of SPADIC the shaping time is rather fixed because of different design arguments (compare 4.2.1). Therefore it can not be traded as an additional degree of freedom in the present context. Moreover it becomes evident from the noise equation that a small load capacity is beneficial in general – which is of course no big surprise. Unfortunately, the actual load capacity the SPADIC must be able to handle is rather high (up to 40 pF, see 4.2.2).

In order to use the latter noise formula, the actual noise sources of the implemented SPADIC front-end (Fig. 5.5) must be identified and "mapped" properly to the parameters a_n^2, b_n^2 and c_n^2. This is done subsequently for the most dominating sources:

First, there is the thermal current noise of the feedback resistor R_f, which is already related to the input and hence can be directly written as

$$a_{\mathrm{nf}}^2 = \frac{4kT}{R_f}, \tag{5.14}$$

with k the Boltzmann constant ($8.62 \cdot 10^{-5} eV/K$) and T the temperature (in Kelvin). As the equation implies, increasing the feedback resistor helps to reduce the noise. But unfortunately at the same time the discharge becomes slower, which for instance enhances the risk of an overflow or at least of a shift of the working point towards a suboptimal region. Moreover a higher resistor leads to a higher DC offset between preamplifier input and output, if leakage current is present – a fact, that for instance can effectively limit the internal dynamic range.

Second, the thermal voltage noise of the series input resistors (basically the sum of wire and input protection resistors) contributes significantly as

$$b_{\mathrm{ns}}^2 = 4kTR_s. \tag{5.15}$$

This shows in particular that the size of the series transistor in the input protection is a trade-off between better protection (larger resistor) and lower noise, and that apart from that, series resistances in the path to the preamplifier should be avoided as much as possible (e.g. by using short wire bondings, high quality connectors, thick PCB routing layers, etc.).

Third, there is the very important thermal voltage (series) noise component presented to the preamplifier input. In a well designed front-end, it is dominated by the channel noise in the input transistor of the preamplifier/shaper:

$$b_{\mathrm{nc}}^2 = \gamma_n \frac{4kT}{g_m}, \tag{5.16}$$

with g_m the transconductance of the input transistor and γ_n a noise factor (semi-empirical constant, usually set to $\frac{2}{3}$, but actually varies within the range $0.5 - 1$ [66]) that depends both on the carrier concentration in the channel and the device geometry. The transconductance g_m is a very crucial parameter in this context, since it can be significantly changed by design. Considering an input MOSFET with a fixed drain current, an increase of the transistor size (larger ratio of width W to length L) linearly increases g_m as long as the transistor works in strong inversion ($g_m = \mathrm{const} \cdot \frac{W}{L}(V_{\mathrm{gate\_source}} - V_{\mathrm{threshold}})$). Then eventually the feedback forces the overdrive voltage ($V_{\mathrm{gate\_source}} - V_{\mathrm{threshold}}$) close to zero

or even below. In the latter case the transistor passes on to weak inversion, where g_m further only depends on the temperature and the drain current and hence only an increase of power can further reduce the effective channel noise. For that reasons most CSAs that face large load capacities use very large input transistors, which practically dominate the power consumption of the front-end and usually work at the edge of weak inversion (all is true for the SPADIC CSA).

And finally, the 1/f noise (flicker noise) of the input MOS contributes as

$$c_{\mathrm{nf}}^2 = \frac{K_f}{C_{\mathrm{ox}} W L}, \tag{5.17}$$

with K_f an empirical constant depending on both the device type and the technology, and C_{ox} the gate oxide capacity per area (also technology-dependent). Since K_f tends to be significantly smaller for PMOS transistors than for NMOS transistors, PMOS input transistors are usually preferred in designs where $c_{\mathrm{nf}}^2 C_{\mathrm{det}}^2$ strongly contributes, whereas NMOS are mostly preferred if the contribution $b_{\mathrm{nc}}^2 C_{\mathrm{det}}^2 / \tau_{\mathrm{s}}$ dominates (e.g. due to very small shaping times). The latter results from the fact that NMOS transistors in strong inversion offer a roughly three-times larger transconductance at the same drain current, due to their higher charge carrier mobility. The two inverse front-ends of SPADIC 1.0 actually realize both alternatives, an NMOS was used for the positive and a PMOS for the negative polarity.

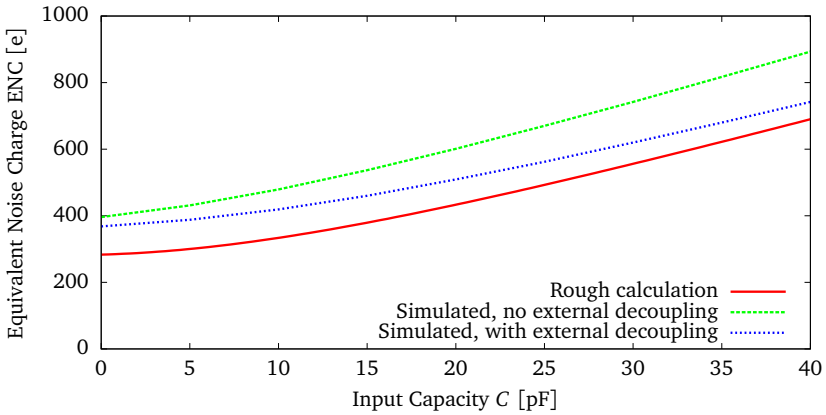


Figure 5.6.: Calculated and simulated equivalent noise charge (ENC) of the positive front-end implemented in SPADIC 1.0 as a function of input capacity.

By using the SPADIC 1.0 design values $R_f = 600k\Omega$, $R_s = 11\Omega$, $g_m = 34mS$ and $W/L = 2900$ together with the technology-dependent parameters $K_f^{NMOS} = 10 \cdot 10^{-25}J$, $C_{ox} = 0.005F/m^2$ and $\gamma_n = 0.66$ at room temperature $T = 300K$, the noise of the implemented positive front-end shown in Fig. 5.6 (red) was calculated. Also shown in the figure are the corresponding simulation results of the positive front-end with (blue) and without (green) external decoupling (roughly 100 nF for decoupling were in each case connected to all crucial bias nodes). It is clearly visible that the rough calculation leads into the right direction but slightly underestimates the simulated noise figure. That is not astonishing, because on the one hand many minor noise sources of the real implementation sum up significantly

and on the other hand the SPADIC CSA is an ideal CR-RC shaper only in a first approximation. In accordance with simulation, for example the series resistor R_x (compare Fig. 5.5) which has not been considered yet for the noise calculation, actually contributes 5 - 10 % to the simulated noise figure.

5.1.2.3. Time Resolution

When measuring the arrival time of a pulse crossing a fixed threshold V_{th} (single measurement), the amplitude noise of the pulse is effectively projected to the time axis. Considering the slope of the pulse at the moment the threshold is crossed, one hence gets the simple but frequently used formula (clock jitter is ignored here and would add quadratically)

$$\sigma_{\text{time}} = \frac{\sigma_{\text{noise}}}{\left.\frac{\mathrm{d}V_{\text{out}}}{\mathrm{d}t}\right|_{t(V_{\text{out}}=V_{\text{th}})}}. \tag{5.18}$$

Therefore for a good time resolution a small rise-time is beneficial. However, as shown later in this chapter (5.3), this formula underestimates the achievable time resolution of the SPADIC, where due to the recording of complete pulses a whole set of measurements is performed. Even though the equation is suitable to deliver a first rough estimation and can hence be used as a good rule of thumb. Using typical SPADIC values one gets for instance (the output voltage was translated into input charge)

$$\begin{gathered}\sigma_{\text{time}} = \frac{800\,\text{e (at 30 pF)}}{\frac{\text{fC}}{80\,\text{ns}}} = 10.25\,\text{ns/fC} \\ \Rightarrow \text{e.g. } 0.51\,\text{ns for a typical input charge of } 20\,\text{fC}.\end{gathered} \tag{5.19}$$

In general, the reachable time resolution additionally also depends on many other factors, such as threshold stability, baseline fluctuations (relative to the threshold) and, which is usually quite significant, the so-called time-walk. Time-walk simply refers to the systematic dependency between pulse amplitude and measured time-offset, which is caused by the fact that two ideal, simultaneously produced but differently scaled pulses never reach a certain voltage level at the same time (with very special exceptions). Therefore in rather classical analog designs additional compensation circuits (e.g. baseline stabilization or time-walk compensation) are commonly used. In the case of SPADIC only some static analog trimming mechanisms were implemented, whereas the overall strategy was either to compensate digitally via the IIR filter (mainly baseline fluctuations) or to further process the pulses off-chip (e.g. fine-granular arrival time derived from a fitted function).

5.1.2.4. Additional Features

As mentioned before, on SPADIC 1.0 two very similar but inverse single-ended CSAs were implemented in parallel in order to provide both polarities and thus to increase flexibility. As sketched in Fig. 5.7, both front-ends are exclusively selectable and share the same input pad, the same ADC, and even the same external decoupling capacitors (or the same

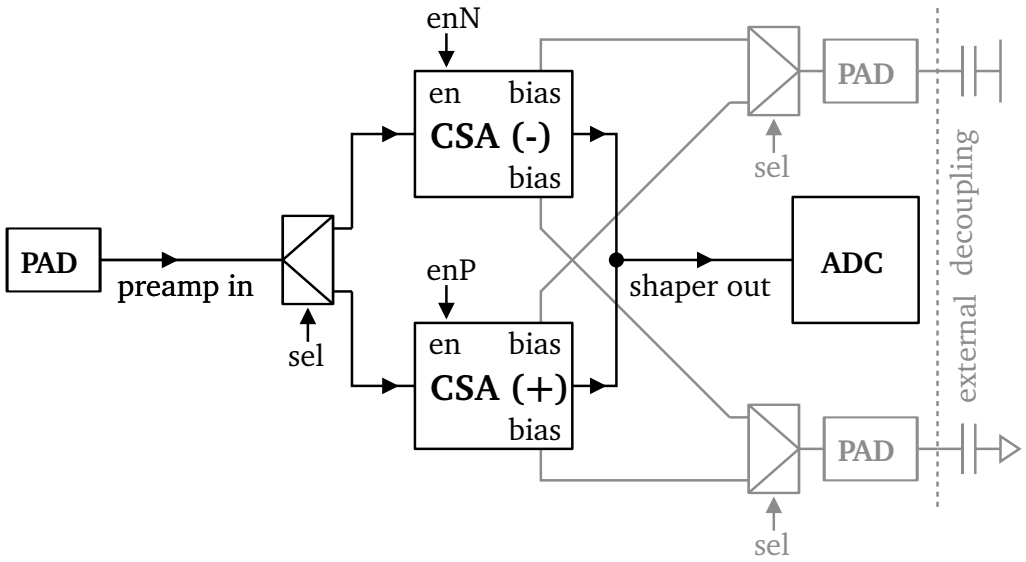


Figure 5.7.: Simplified switching scheme of the two available front-ends of SPADIC 1.0 sharing the same input pad, the same ADC, and the same external decoupling capacitors.

decoupling pads respectively). To allow for that scheme it requires a CSA design that can be completely turned off, because firstly a disabled CSA should not consume power and secondly the CSA outputs are shorted and therefore must not drive both at the same time. Moreover, the power-off feature allows to disable broken or unused analog channels completely. Similar to the enable/disable mechanism itself, the input multiplexer is also very noise critical, since it both adds an additional series resistor and some parasitic capacities directly to the most sensitive node. For that reason the input multiplexer has been realized with a very carefully sized transmission gate, and according to simulation it contributes slightly below 4 % to the total noise.

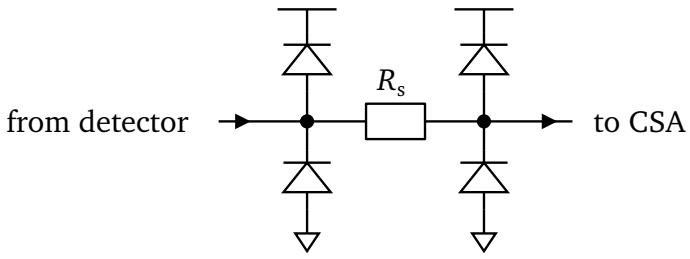


Figure 5.8.: Simplified schematic of the input protection realized in the analog input pads of SPADIC 0.3 and 1.0.

Because the analog inputs of the CSAs are directly connected to detector electrodes (e.g. cathode pads in the case of TRD) which moreover are usually operated in an environment of high electric fields (e.g. amplification and drift fields in the case of TRD), high voltage sparks are likely to hit and damage the very sensitive preamplifier structures. To attenuate

that problem, a simple input protection was integrated in each analog input pad connected to a CSA in SPADIC 0.3 and 1.0 (see Fig. 5.8).

The principle of the input protection is quickly described: If the input voltage significantly rises above the supply voltage or falls below ground potential, the two diodes on the left hand side (Fig. 5.8) provide a DC path for an eventual spark to discharge, while concurrently the series resistor R_s limits the current flowing to the preamplifier input node. Therefore the size of R_s is a trade-off between good ESD protection and additional thermal noise (for the exact noise contribution see 5.1.2.2), whereas the diodes should be mainly designed to have a small on-resistance. The second column of diodes on the right hand side acts similar to the first on the left hand side and also dumps the current if the corresponding node potential rises too high or falls too low. In SPADIC 1.0 the input resistance was set such that it contributes roughly 5 - 10 % (depending on the connected capacitive input load) to the total noise, which corresponds to a design value of 11 Ω.

5.1.3. CSA: Details of Implementation

The subsequent details are mostly related to the implementation of the positive front-end. But, and except for very few details, the shown schematics are conceptually very similar to those of the negative front-end, if one replaces all NMOS with PMOS transistors and vice versa. Moreover, most crucial design parameters of both front-ends, such as shaping time, input/output range and gain are also nearly equal and especially the simulated noise figures are equally large (or small). An exception is the power consumption, which is actually roughly 2–3 times larger in the negative front-end, simply because it was not as thoroughly optimized[1].

5.1.3.1. Unified Amplifier Cell

As it was previously shown in Fig. 5.5, the CSA requires voltage amplifiers to realize the transfer function described in section 5.1.2.1. Because it is necessary to have very similar points of operation of both the preamplifier and the shaper input in order to make the feedback circuitry work properly, it is a good methodology to use instances of the same voltage amplifier both for the preamplifier and the shaper. As mentioned earlier in section 5.1.2.2, the effective S/N ratio is dominated by the input transistor and becomes relatively stable after pre-amplification. For that reason the bias current flowing through the voltage amplifier of the shaper can be significantly smaller than the current flowing through the preamplifier, which, in particular, helps to decrease the total power consumption.

The SPADIC front-ends use the unified voltage amplifier cell shown in Fig. 5.9 to reach the latter goal but without the need to adjust the bias conditions independently: 11 (negative CSA 12) unified voltage amplifier cells were connected in parallel to realize the preamplifier, whereas only 1 cell was used for the shaper. Hence the shaper only consumes 1/11th (negative CSA 1/12th) of the current biasing the preamplifier. More precisely, also the driving part of the amplifier cell (source follower) was designed to be independently scalable

[1] The negative front-end is rather a prove of principle than an optimized design. Even though having a negative CSA is beneficial in general, no concrete application is known so far.

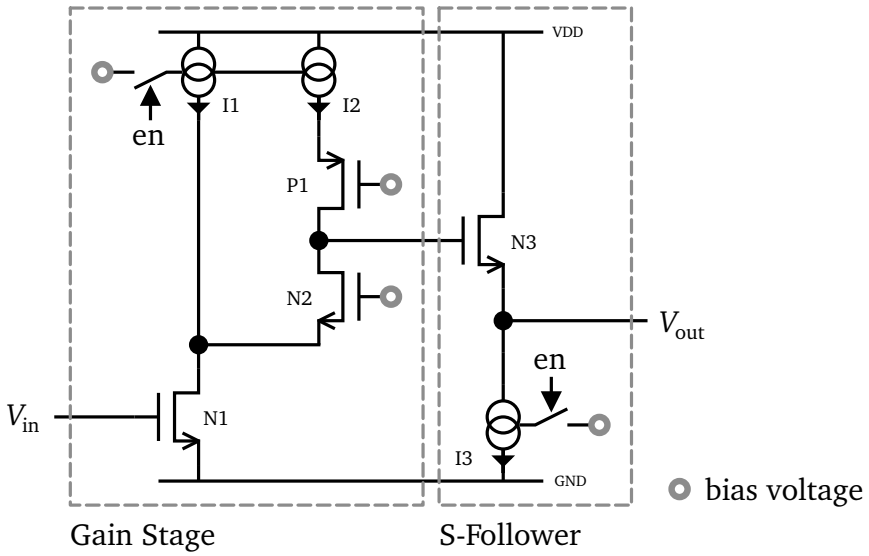


Figure 5.9.: Simplified schematic of the unified voltage amplifier cell used both for the preamplifier and the shaper. On the left hand side the cascoded gain stage is shown, whereas on the right hand side the source follower is sketched.

and was used 11 (negative CSA 12) times in the preamplifier and 5 times in the shaper[1]. The latter was done due to two reasons, first to increase the driving strength of the shaper and second to improve the ability of the source follower to dump the DC offset current flowing out of the subsequently connected current-mode ADC.

Although the shown circuit is quite simple and commonly used for that kind of applications, the details of the implementation and the optimization depend strongly on the given external constraints and the respective technology. Therefore – but without the intention of going too much into detail — at least the most important issues are summarized subsequently.

As previously discussed in section 5.1.2.2, the fraction of total electronic noise that can be directly affected with proper design techniques is dominated by the thermal channel and flicker noise of the input transistor N1. Whereas flicker noise tends to decrease for higher gate areas W · L (it also depends on the transistor type as discussed in 5.1.2.2), the thermal channel noise can be reduced by increasing the transconductance g_m. The approach taken for the SPADIC front-end, where high detector capacities are given and small shaping times are required (compare table 4.1), was to choose and fix the maximum bias current I1 + I2 first and afterwards to maximize the transconductance of N1 as much as possible. Practically that means to increase the W/L ratio of N1 working in saturation until it changes from strong ($g_m \sim$ W/L) to weak inversion (g_m independent of W/L). A further increase of W/L would only increase the gate capacity and hence the total load capacity of the input stage.

In order to increase the output resistance of the gain stage, both a straight cascode and a folded cascode are commonly used alternatives. A folded cascode provides the possibility to

[1]Hence the ratio of source follower to gain stage cells is 1 for the preamplifier but 5 for the shaper.

raise the source potential of the input NMOS and hence to lower the power consumption of the gain stage at the same drain current. But the biggest disadvantage of the folded cascode and at the same time the reason why a straight cascode was used in the SPADIC front-end, is the requirement of a second ground (or power) contact, which effectively demands an additional and separated power domain including all crucial and expensive consequences. Furthermore, an advantage of the straight cascode is that the bias current I2 is not "lost" but also flows through the input transistor. In the implemented front-end of SPADIC 1.0 the total bias current was chosen to be about 2 mA – with the ratio I1/I2 set to 10.

Because on the one hand the current sources I1 and I2 effectively tend to have a relatively large transconductance (high W/L ratio even if large L is preferable to increase the output resistance) in order to be able to provide the large bias currents, but on the other hand act like parasitic gain stages with the bias voltages as input signals, they are the second most noise critical devices in the amplifier cell and must be designed carefully. Therefore a very high effort to decouple the bias voltages of I1 and I2 with multiple capacities distributed over various levels of logical and geometrical hierarchy all over the chip and externally was taken. Moreover, the enable switches used to switch off all three current sources I1, I2, and I3 were designed not to mask the decoupling capacitors. Basically they require low on-resistances and very noiseless logic signals (coming for instance from logic cells powered with the same analog supply as the whole CSA).

The source follower at the voltage amplifier output has several tasks: it acts as unity gain buffer, helps to dump the baseline current of the ADC, increases the driver strength, and works as level shifter. Since the DC-feedback between V_{in} and V_{out} forces V_{in} to $V_{\text{threshold}}$ of N1 (which works at the edge of weak inversion), the DC operation point of V_{out} is also $V_{\text{threshold}} = 500\,\text{mV}$. While the input node of the source follower can practically reach positive supply (with some loss of linearity), $V_{\text{threshold}}$ of N3 should be as minimal as possible in order to maximize the upper limit of V_{out}. Therefore a low-threshold NMOS was used for N3, which led to an upper output limit of roughly 1.5 V.

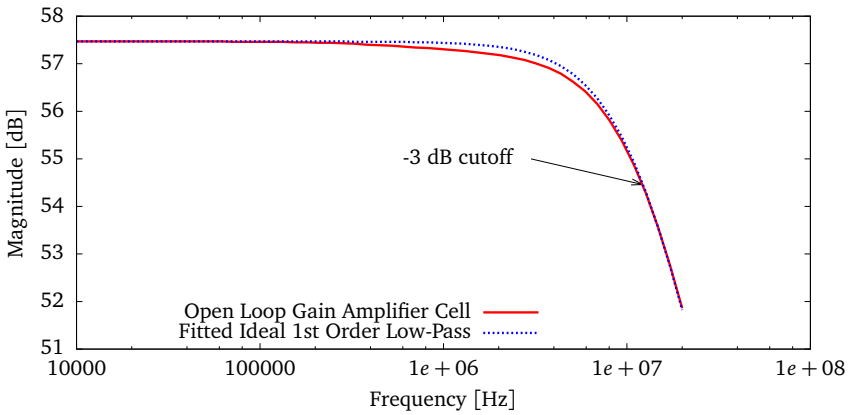


Figure 5.10.: Simulated open loop gain of a unified amplifier cell operated at the closed loop operation point (red curve). For comparison the gain of an ideal low-pass filter with a DC gain of 750 and a bandwidth of 12.2 MHz ($RC = 13\,\text{ns}$) is also shown (blue curve).

The simulated open loop gain of the implemented unified amplifier cell of the positive CSA is shown in Fig. 5.10. The simulated DC gain is 750 and the bandwidth 12.2 MHz ($RC = 13$ ns).

To roughly check the simulated AC result the circuit can be simplified: first, if one assumes that the output resistances of the current sources I1 and I2 (including the cascodes) are very large, and if one further considers that N1 effectively drives into the output resistance seen at the output of the gain stage, the DC component of V_{out}/V_{in} can be derived easily. Since I1, I2 and P1 can be ignored due to the latter assumption, the output resistance becomes

$$r_{out} = r_{N1} + r_{N2} + g_m^{N2}\, r_{N1}\, r_{N2}, \tag{5.20}$$

and hence the DC gain is simply $g_m^{N1}\, r_{out}$. For the AC component, one must consider the dominant pole at the output node of the gain stage, which is $\tau = r_{out}\, c_{out}$, with c_{out} the total capacity seen from the output node to ground. The capacity c_{out} can be approximated as the sum of the gate capacity of N3 c_g^{N3} and the drain capacities of P1 c_d^{P1} and c_d^{N2} N2. Putting everything together, the complete transfer function becomes

$$H(s) = \frac{V_{out}}{V_{in}} = \frac{A_{DC}}{1 + s\tau} = \frac{g_m^{N1}\, r_{out}}{1 + s\, r_{out}\, c_{out}}. \tag{5.21}$$

If one takes the numbers from the simulation ($g_m^{N1} = 3.1$ mS, $g_m^{N2} = 230\,\mu$S, $r_{N1} = 13.75$ kΩ, $r_{N2} = 73$ kΩ, $c_g^{N3} = 13$ fF, $c_d^{P1} = 10$ fF, $c_d^{N2} = 16$ fF), one gets $r_{out} = 317.6$ kΩ and $c_{out} = 39$ fF. That finally leads to the numbers $A_{DC} = 984$ and $\tau = 12.3$ ns, which lies at least within the range of the simulated results – one must not forget the simplifications made and that the source follower was nearly ignored.

5.1.3.2. CSA Circuit and Feedback

To realize the transfer function of the preamplifier/shaper calculated before and shown in Fig. 5.5 various requirements and dependencies have all to be considered at once, which make the design process relatively difficult and practically iterative. Besides the two constraints from equation 5.11 ($nR_x \ll R_f$ and $nC_fR_x = \tau_s/2.$), the circuit must be linear, exploit the full dynamic range, must not introduce additional dominant noise sources, must be stable, should use small components (to save layout space), and must be adjusted to the required input range – just to name to most obvious criteria. Those details are discussed subsequently.

The feedback of the preamplifier requires a very large resistor (the discharge time should be slow compared to the shaping time and the thermal current noise must be kept small, see equation 5.14) and a large capacitor (to keep the negative voltage peak at node V_x relatively small in order not to lose linearity). If one for instance allows for a maximum voltage peak of -150 mV and considers the maximum input charge of 75 fC, the capacity must be at least 500 fF. And moreover, setting the current multiplier n to the reasonable value 10, the sum of all feedback capacitors required the preamplifier becomes already 5.5 pF. Such a capacitor practically covers an area of roughly 75 µm x 75 µm in the used technology, if one

uses metal-metal capacitors. Moreover, if one demands the preamplifier discharge to be at least 5 times slower than the shaping time, the feedback resistor becomes 400 ns/500 fF = 800 kΩ, which again requires a huge chip area, if one uses for instance poly-gate resistors.

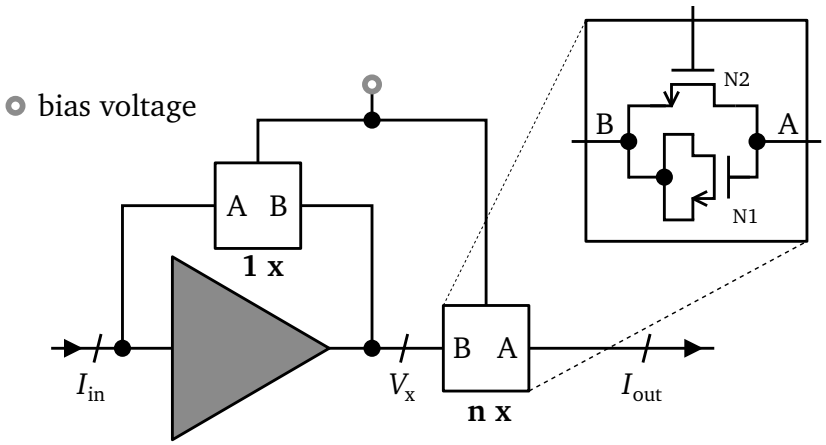


Figure 5.11.: Implemented feedback of the preamplifier using a dedicated feedback cell. The feedback cell uses transistors instead of a capacitor and a resistor in order to considerably save chip area.

To overcome the latter problem of too area-consuming feedback devices, both the capacitor and the resistor were built using the feedback cell shown in Fig. 5.11. The feedback cell (used n times in parallel on the right hand side to realize the current scaling factor n) realizes both resistor and capacitor with NMOS transistors. These cover much less area, but potentially inject a strong non-linearity to the circuit. But fortunately (or in fact intentionally) in this particular case the latter problem solves itself: if one considers that both the input and the output node are virtual grounds at the same DC potential (as mentioned before the shaper makes use of the same voltage amplifier cell), the conditions of both feedback cells are completely symmetric – independent of their respective implementation or area of operation. Therefore, if V_x decreases, the charge stored on N1 in the right feedback cell must be n times the input charge Q_s stored on N1 in the left feedback cell. And similar, the discharge current through N2 of the left feedback cell is always $1/n$-th of current flowing through the right cell. Therefore all potential non-linearities cancel each other and the preamplifier effectively behaves similar to the version using "real" resistors and capacitors. That technique, at least for the feedback resistor, is well known and commonly used (for instance in [38]).

The required parameters of the shaper can be calculated easily. First the feedback capacity C_s must be chosen such that the output range (500 mV to 1.5 V) is fully exploited. To find a good start value one can use equation 5.12 and the actual design values $Q_s^{max} = 75$ fC and $n = 12$, which leads to $C_s = 660$ fF (actual design value 450 fF). And second the desired shaping time of 80 ns directly defines $R_s = 120$ kΩ (actual design value 176 kΩ). Due to the relatively small values and the need of a very linear behavior, both the resistors and the capacitors where implemented using metal-metal and poly structures respectively.

The simulated gain of the whole shaper and, as a reference, the gain of an ideal 2nd order low-pass is shown in Fig. 5.12 (right). For comparison: corresponding to the simplified

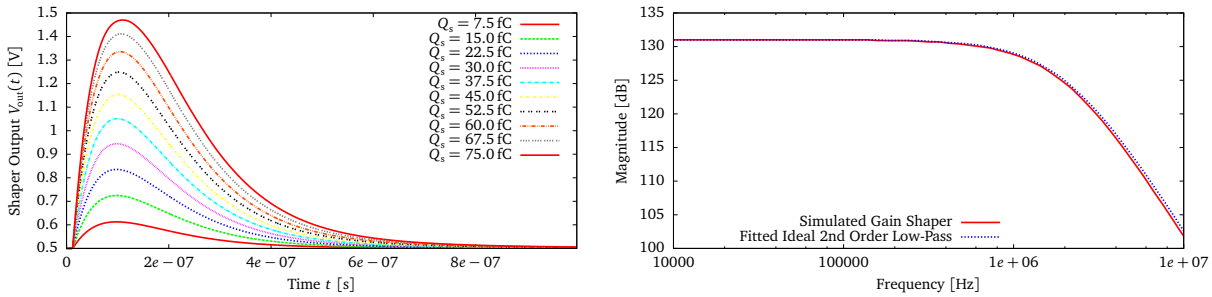


Figure 5.12.: Left: simulated output pulses of the positive CSA for input charges from 7.5 fC to 75 fC. Right: simulated gain of the CSA (red curve). For comparison the gain of an ideal 2nd order low-pass with a DC gain of 131 dB and a bandwidth of 1.99 MHz ($RC = 80$ ns) is also shown (blue curve).

equation 5.11 the calculated DC gain should be $2 \cdot 12 \cdot 176\,\text{k}\Omega \mathrel{\hat{=}} 132\,\text{dB}$ (and is 131 dB). Also shown in Fig. 5.12 (left) are simulated output pulses of the shaper for input charges from 7.5 fC to 75 fC. A small non-linearity at very large pulses is visible, which is – given the main application TRD – completely uncritical though.

5.1.3.3. Bias and Configuration

To save pins, power supplies and external logic all bias voltages of the CSA (as well as of the ADC) are internally generated with 10 7 bit current DACs that are placed in a dedicated bias channel, although all internal bias voltages can also be accessed externally (for monitoring, decoupling, or overwriting – as indicated in Fig. 5.7).

The control bits of the current DACs together with all controls for the switches used to configure the CSAs are chained in a two-phase shift register having a total length of 584 bit (for details see section 5.4.1). The shift register runs from top to bottom, first through all bias structures and later through all 32 channels. Also connected to the configuration chain are the static switches required for the ADCs, the 7 bias DACs of the ADC and the 32 7 bit current DACs used for trimming the baseline between CSAs and ADCs (see also section 5.2.3.2). Therefore, by writing the shift register, the whole analog part is configured at once. Both ends of the register chain are connected to the digital part and can only be written or read that way.

5.1.3.4. Monitoring

For quick analog diagnoses, each of the 32 CSA outputs can be connected to a monitor bus, which allows to directly visualize the analog pulses or the baseline. Due to the limited bandwidth of the bus (long path through the chip, long and thin PCB wire, bonding-wires, etc.), the signals seen via the monitor bus are significantly distorted and hence are not suitable for qualitative measurements (like extracting the rise-time, noise, etc.). Even though the monitor bus is a very important tool for commissioning, quick checks, and debugging.

5.1.3.5. Layout

A very high effort was taken to design all parts of and around the CSAs (including the bias and powering structures) as modularly as possible. All analog layout parts were composed of bricks attached to a coarse grid. No overlapping between any cells of the whole analog part (except for the ADC) occurs (due to perfectly clean cuts). Whereas for larger blocks a modular layout is not an unusual practice, in the present case the modular principle has been excessively extended even to the smallest component (which is also true for the schematics). For instance the power buses (e.g. of vddc and gndc) were built of a unified bus layout brick (using M5, M6, and proper via arrays), which, if properly rotated and mirrored, sums up to a complex, convoluted, easy accessible, and symmetric (considering vddc and gndc) signal/layer structure. Other cells like for instance the unified voltage amplifier or the feedback cells of the CSAs were laid out such that they can be scaled (connected in parallel) by simply putting more of them next to each other.

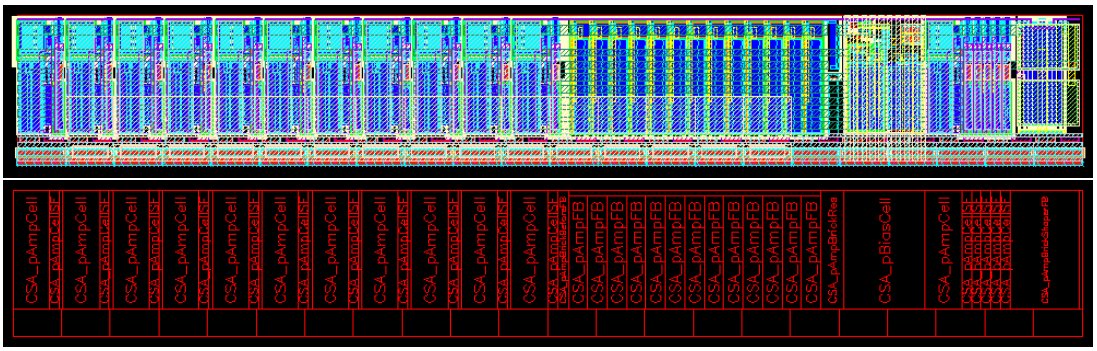


Figure 5.13.: Layout of the positive CSA channel. Both the layout details (top) and the modular structure (bottom) are shown. The whole block has a size of 550 µm x 60 µm.

The high effort of such a modular layout is justified by its advantages: everything is much cleaner, easier to scale or modify, better to understand, and probably, most important, in many details re-usable. Exemplarily the layout and the corresponding module structure of the positive CSA is shown in Fig. 5.13.

The vertical channel pitch of the (two) CSAs was set to 120 µm, simply to match the already given height of the ADC layout. Eventually the CSA channel length came out to be 550 µm. To avoid crosstalk or crosscurrents as much as possible every channel (as well as every ADC) was encapsulated by a guard ring. To relax the channel density and in oder to stabilize the bias and power voltages, each block of 8 CSA/ADC channels was terminated by a dedicated decoupling channel (height 90 µm), which is completely filled with metal-metal and MOS capacitors. On top of all channels (above the first decoupling channel) the bias channel was placed. Because the bias channel also supplies the ADC, it is much longer and spreads over the whole CSA/ADC analog part (length ≈ 1000 µm). For Fig. 5.14 the upper-left part of the analog layout was cut out. The first two horizontal structures belong to the bias channel (DACs and diodes were separated in two parts), followed by the first decoupling channel. Below the decoupling channel lie the first two CSA channels 0 and

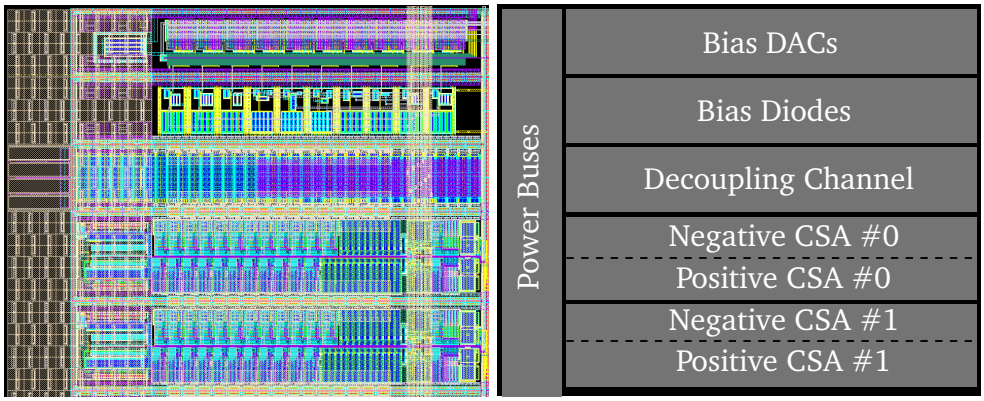


Figure 5.14.: Cut-out of the CSA layout. The first two horizontal structures belong to the bias block, followed by a decoupling channel. Below the first two CSA channels can be spotted (upper half negative CSA, lower half positive CSA).

1 (upper half negative CSA, lower half positive CSA). Moreover, on the left hand side the vertical power lines (made of the earlier mentioned unified bus brick) and on the right side vertical bias lines (yellow) can be vaguely spotted.

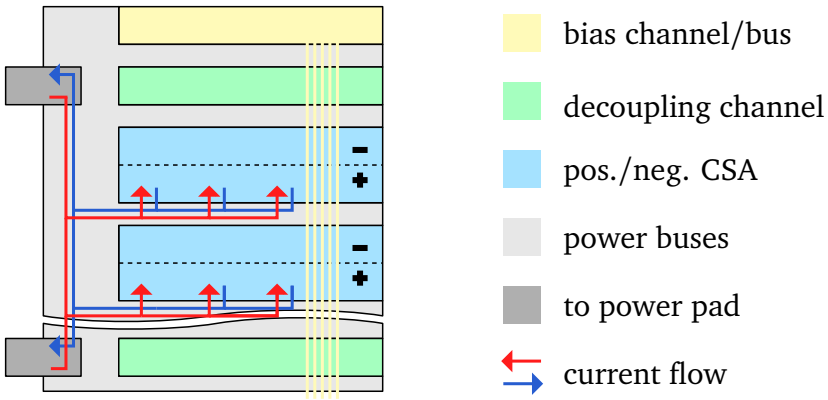


Figure 5.15.: Power and bias scheme of the analog front-end. A finger structure was used to avoid common return paths between the different channels.

A more abstract view of the channel structure is sketched in Fig. 5.15, where also the intended current flows are drawn. The power, which is externally connected on the left hand side, is led via a finger structure to the channels and back. Since the positive and the negative CSAs are never active at the same time, one finger must only supply one CSA (current enters/leaves either from the bottom or from the top). The layout has been designed such that no DC path between two channels exists (of course except for the power bus on the left hand side). The latter detail is very important, because common return paths (or wrongly but usually also called ground loops) can easily lead to non-stabilities, especially if some detector shares its ground with the ASIC. A similarly improved power structure on the latest SPADIC 0.3 PCB could indeed help to significantly reduce oscillations

and the overall sensitivity of the amplifiers to external disturbances (for details see section 7.3.1).

5.1.4. Selected Measurements

The very first result that is at least a proof of principle, is the direct analog measurement of shaper output pulses via the monitor bus. Figure 5.16 shows two of the very first oscilloscope screenshots that were taken with the first SPADIC 1.0 setup iteration. The pulses were generated using the analog test injection. Even though the recordings are not very pretty and the pulses suffer from suboptimal bias conditions and are distorted due to the limited bus bandwidth, the screenshots have a certain historical value. The plot on the left hand side shows a pulse of the positive CSA (baseline at 500 mV), the other plot a pulse of the negative CSA (baseline at about 1.3 V).

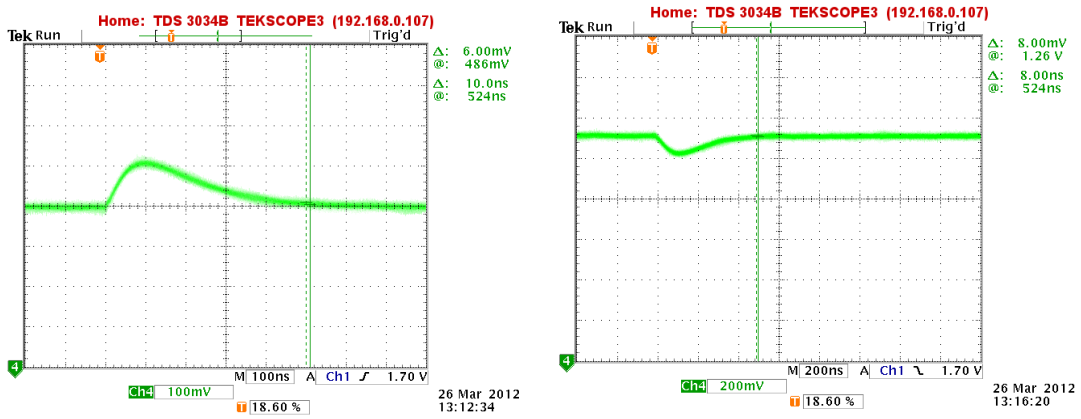


Figure 5.16.: Oscilloscope screenshots of analog shaper pulses recored via the monitor bus. Left: positive CSA, right: negative CSA.

As it was briefly described in section 4.4, on the last 4 SPADIC versions (0.1, 0.2, 0.3, and 1.0) slightly different but conceptually similar versions of the CSA circuit were realized. One of the main goals of the different CSA versions was to investigate the effective preamplifier noise. And indeed in doing so very interesting results, which are summarized subsequently, could be gathered.

In Fig. 5.17 the measured noise values (in ENC) as a function of load capacity of the different channel types on SPADIC 0.2 are shown. All noise measurements were done by evaluating threshold/s-curve scans of test pulses[1], which were injected via a well calibrated on-chip injection capacitor. The 26 CSA channels of SPADIC 0.2 were realized with 3 different types of input NMOS transistors, whereas the remaining channel layout remained identical: *normal* (tri-well, minimal gate length 180 nm), *no-tri-well* (no tri-well, minimal gate length) and *long* (tri-well, gate length 320 nm). Although the different types of input transistors both in simulation and measurement had more or less the same noise offset (at 0 pF input capacity) of about 200 e, the slopes of the noise figures showed large differences.

[1]The SPADIC versions 0.x all had analog discriminators connected to the shaper outputs.

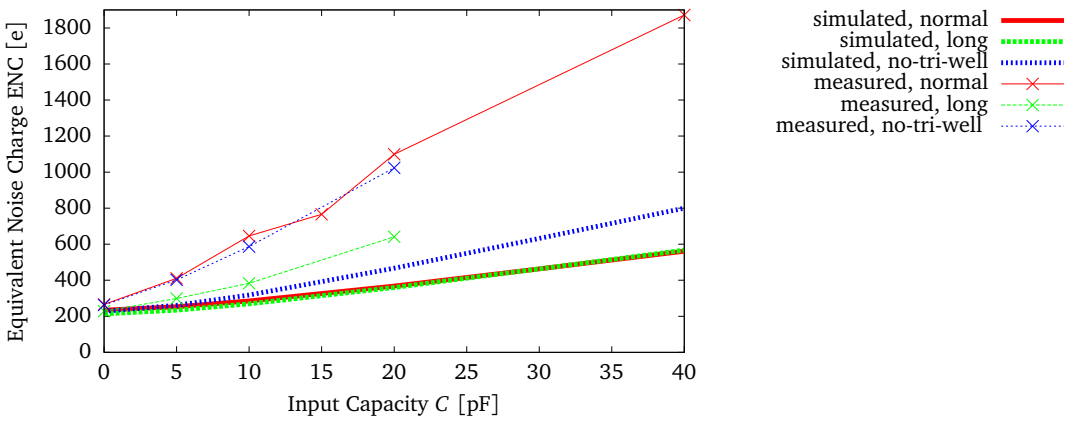


Figure 5.17.: Simulated and measured noise curves of the positive CSA realized on SPADIC 0.2. To evaluate an optimal noise figure, type and size of the input transistor were varied.

Whereas the variation of the input NMOS type had only little impact on the noise figures in the simulation (lower 3 curves in Fig. 5.17), the measured noise of the long NMOS channel was by far smaller than of the other two channel types, even though still larger than predicted by the simulator.

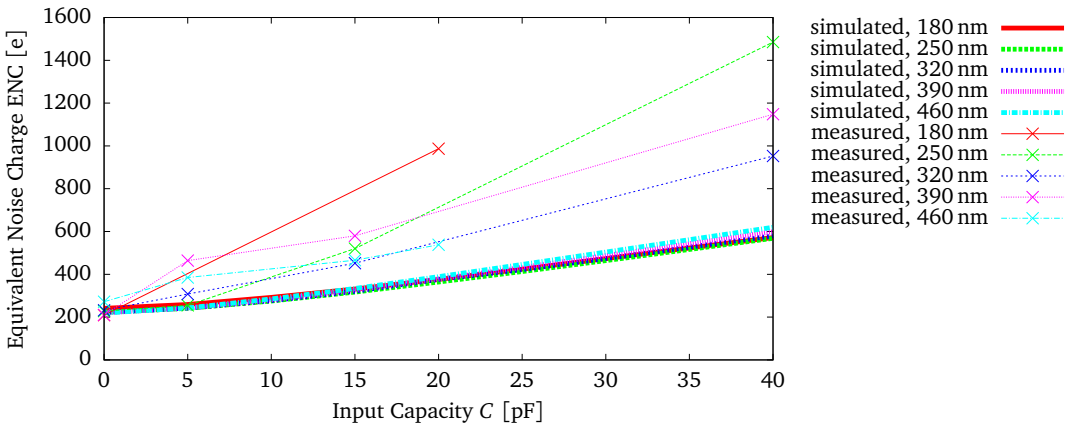


Figure 5.18.: Simulated and measured noise curves of the positive CSA realized on the SPADIC 0.3. To evaluate an optimal geometry, this time the length of the input transistor was varied. The impact on the measured noise figures was huge, while nearly no differences have been predicted by the simulator.

To further investigate the bad noise results of version 0.2, most schematics of the consecutive version SPADIC 0.3 were modified, most parts of the channel layout completely remade and again different channels with systematically scaled input transistors (NMOS with tri-wells and lengths from 180 to 460 nm) realized. The noise results evaluated from SPADIC 0.3 are summarized in Fig. 5.18. Again the offsets of all measured and simulated results were equally at 200 e, whereas all measured slopes strongly differed both from the

simulated values and from each other. Surprisingly, a noise minimum at a gate length of about 320 nm emerged. It was and still is not understood why in contrast to the measured values the simulated noise curves stayed nearly identical for all transistor lengths.

Nevertheless, the best achieved noise results of roughly 200 e + 20 e/pF with the 320 nm input MOS were at least satisfying. For that reason the lengths of the input transistors of SPADIC 1.0 were also set to 320 nm, where hence similar noise results are expected but have not yet been measured[1].

5.1.5. Summary Table: CSA

Both CSAs	
Order of shaper	2nd
Order of complete CSA	1st (CR-RC)
Shaping time	80 ns
Input range	75 fC = 468.1 ke
Size of layout (per CSA)	440 µm x 60 µm
Positive CSA	
Charge polarity	positive
Peaking time (0 to 100 %)	87 ns (@ 100 ke input)
Input type	NMOS
Power consumption	3.8 mW
Number of amplifier cells	12
Noise	387 e + 11 e/pF (@ 100 ke input, simulated)
Negative CSA	
Charge polarity	negative
Peaking time (0 to 100 %)	97 ns (@ 100 ke input)
Input type	PMOS
Power consumption	10 mW (not optimized yet)
Number of amplifier cells	13
Noise	439 e + 11 e/pF (@ 100 ke input, simulated)

5.2. Algorithmic Pipeline ADC

According to the SPADIC architecture the output pulses of the CSA shall be digitized as soon as possible with an individual ADC in each channel. And indeed, in SPADIC 0.2, 0.3, and 1.0 very similar versions of a home-made current-mode pipeline ADC are implemented and operated[2].

[1] An analog discriminator is not anymore available in SPADIC 1.0, which would allow for direct s-curve scans. Although a rough estimation could be evaluated by measuring and comparing the sigmas (standard deviations) of digital ADC baseline variations with and without a CSA connected.

[2] Pipeline ADCs are sometimes also named algorithmic pipeline ADCs, which refers to the iterative processing scheme the ADC is based on.

This section now gives some theoretical background, analyzes details of the implemented design, and shows some results from simulation and measurement.

5.2.1. General Principle

This section summarizes in general the most important aspects of the algorithmic ADC principle.

5.2.1.1. Formalization

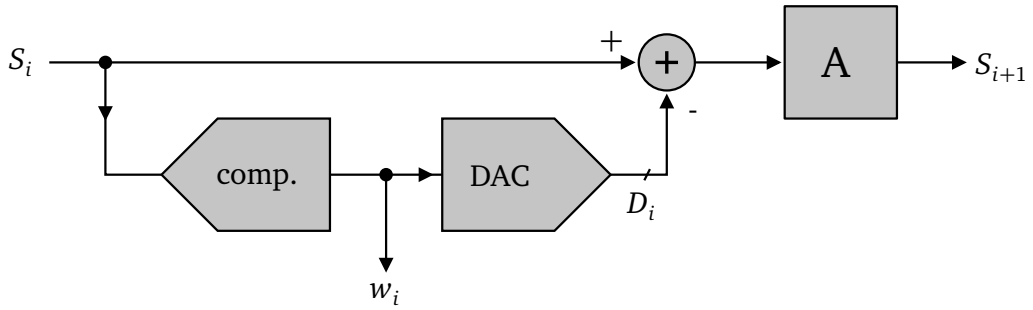


Figure 5.19.: Flow diagram of a generalized stage used within an algorithmic ADC. The input signal S_i gets coarsely digitized in steps of w_i (lower path), then the analog fraction D_i corresponding to the digitized fraction w_i is subtracted from S_i (the adder), and finally the residue is amplified with the constant gain A, which leads to the output signal S_{i+1}.

The flow diagram of a single precessing stage of an algorithmic ADC is sketched in Fig. 5.19. It is assumed here that the signals S_i, D_i and S_{i+1} are continuous and limited to $[-S_R, S_R]$. The comparator in the lower path converts S_i into a numerical value $w_i \in \mathbb{Q}$. If the comparator has a total of $T-1$ thresholds, the T different w_i can be set without loss of generality to

$$w_i \in \left\{ -\frac{T-1}{2}, -\frac{T-1}{2}+1, \ldots, \frac{T-1}{2}-1, \frac{T-1}{2} \right\}. \tag{5.22}$$

In general, the DAC produces the equidistant analog values $D_i = w_i B$, with $B \in \mathbb{R}$ some fixed step height. Considering the adder and the multiplier (with constant gain $A \in \mathbb{R}$), the output signal of the i-th stage S_{i+1} is simply $S_{i+1} = A\,(S_i - D_i)$. If one wants to fully exploit both the input and the output range $[-S_R, S_R]$, one can demand that the highest (or lowest) possible input signal S_R $(-S_R)$ is mapped to S_R $(-S_R)$. Then B can be calculated as

$$\begin{aligned} &(\max\{S_i\} - B\max\{w_i\})\,A = \max\{S_{i+1}\} \\ &\Rightarrow \left(S_R - B\frac{T-1}{2}\right)A = S_R \\ &\Rightarrow B = \frac{2S_R}{A}\frac{A-1}{T-1}, \end{aligned} \tag{5.23}$$

which leads to a good definition of D_i:

$$D_i = \frac{2S_R}{A}\frac{A-1}{T-1}w_i. \tag{5.24}$$

By nesting the iterative equation of a single stage N times, the initial signal S_0 after passing N stages can be expressed as

$$S_N = A(\ldots A(A(S_0 - D_0) - D_1) - \ldots) = A^N S_0 - \sum_{i=0}^{N-1} D_i A^{N-i} \tag{5.25}$$

$$\Rightarrow \quad S_0 = \frac{S_N}{A^N} + \sum_{i=0}^{N-1}\frac{D_i}{A^i} = \underbrace{\frac{S_N}{A^N}}_{\text{analog term}} \underbrace{+ 2S_R\frac{A-1}{T-1}\sum_{i=0}^{N-1}\frac{w_i}{A^{i+1}}}_{\text{digital term}}. \tag{5.26}$$

This important equation shows the basic principle of the algorithmic ADC: the initial signal is step by step fragmented into a digital and an analog component. While the digital information is effectively stored in the numerical values w_i, some of the initial analog information remains hidden in the final analog output signal S_N of the last stage $N-1$. Therefore increasing the ADC resolution practically means decreasing the analog term, which can be done by incrementing A or N.

In order to bring the final ADC result into a convenient format, the sum in the digital term must be computed eventually. To avoid the generally very complex division by A^{i+1}, it is extremely advantageous to set A to 2^k with $k \in \mathbb{N}$ ($k = 0$ is not possible since for the analog term to decrease A must be strictly larger than 1). Due to the fact that w_i can always be converted to some kind of binary representation, the division becomes only a simple shift operation. Apart from that one would come to the same conclusion, if one considered that a multiplication by 2^k is usually much easier to realize than a multiplication by an arbitrary factor.

In general, the algorithmic methodology is intrinsically able to correct digital errors. For instance a wrong decision of the comparator in stage i_0 (e.g. due to a noisy threshold) would lead to the wrong digital result $w'_{i_0} = w_{i_0} + w_{\text{err}}$. That would further cause the wrong output signal $S'_{i_0+1} = A(S_{i_0} - D_{i_0} - \frac{2S_R}{A}\frac{A-1}{T-1}w_{err})$. Corresponding to equation 5.25 the analog error would then propagate to the last stage like

$$S'_N = A^N S_0 - \sum_{i=0}^{N-1}\left(D_i A^{N-i}\right) - A^{N-i_0}\frac{2S_R}{A}\frac{A-1}{T-1}w_{\text{err}} = S_N - A^{N-i_0-1}2S_R\frac{A-1}{T-1}w_{\text{err}}. \tag{5.27}$$

But at the same time the digital term would be changed by $+\frac{2S_R}{A^{i_0+1}}\frac{A-1}{T-1}w_{\text{err}}$ (equation 5.26). With the latter results (and equation 5.26), the equivalent input signal of the first stage S_0' including the impact of the digital error becomes

$$\begin{aligned} S_0' &= \frac{S_N'}{A^N} + 2S_R\frac{A-1}{T-1}\sum_{i=0}^{N-1}\left(\frac{w_i}{A^{i+1}}\right) + \frac{2S_R}{A^{i_0+1}}\frac{A-1}{T-1}w_{\text{err}} \\ &= \frac{S_N}{A^N} + 2S_R\frac{A-1}{T-1}\sum_{i=0}^{N-1}\left(\frac{w_i}{A^{i+1}}\right) \underbrace{-\frac{A^{N-i_0-1}}{A^N}2S_R\frac{A-1}{T-1}w_{\text{err}} + \frac{2S_R}{A^{i_0+1}}\frac{A-1}{T-1}w_{\text{err}}}_{=0 \quad \text{(compensation)}} \\ &= S_0 \end{aligned} \tag{5.28}$$

and hence stays unaffected. An incorrect digital decision therefore only causes a shift of information from the digital term to the analog term, and – in a manner of speaking – only postpones the digitization, if stage i_0 is followed by a sufficient number of correctly working stages to compensate.

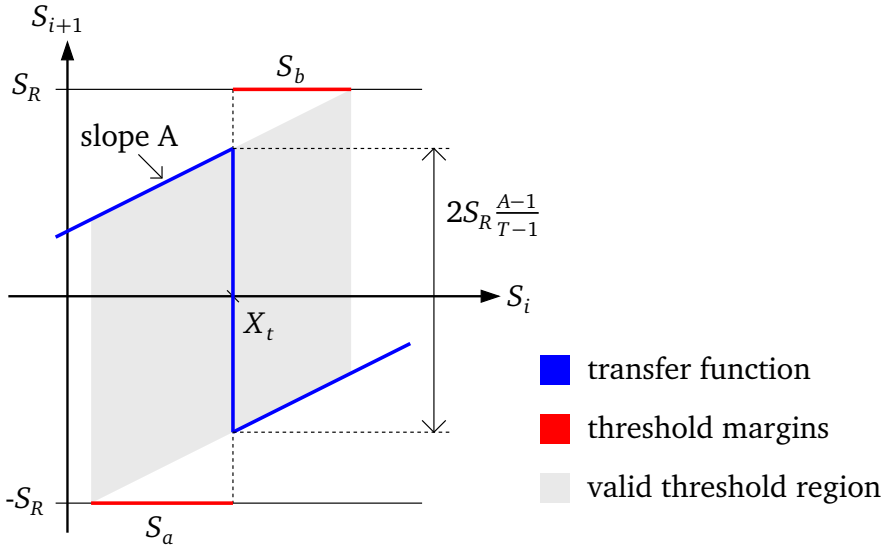


Figure 5.20.: Cut-out of the transfer function of a single stage of an algorithmic ADC. As long as a wrong threshold stays within the gray area, the resulting incorrect decision can theoretically be compensated by the subsequent stages.

The principle of error correction is limited in real circuits due to the limited analog signal range ($[-S_R, S_R]$ in the present case). If one plots the transfer function of a stage around the t-th threshold (reminder: the comparator has $T-1$ thresholds) adjusted to position X_t, one gets Fig. 5.20. From equation 5.23 and as shown in the figure it becomes evident that if the input signal S_i crosses the threshold, the output signal S_{i+1} skips by

$$\Delta S_{i+1} = -2S_R\frac{A-1}{T-1}. \tag{5.29}$$

In general and as just discussed, the algorithmic mechanism is feasible to correct digital errors, as long as the output signal S_{i+1} stays in $[-S_R, S_R]$. But if a wrong threshold causes

the output signal to leave the region $[X_t - S_a, X_t + S_b]$ (Fig. 5.20, gray region), the output will saturate and an uncorrectable error will develop (the analog information is irrevocably lost). For the balanced solution $S_a = S_b$, the range $[X_t - S_a, X_t + S_a]$ can be geometrically calculated as

$$\begin{aligned} AS_a &= \left(2S_R - 2S_R\frac{A-1}{T-1}\right)\frac{1}{2} \\ \Rightarrow S_a &= \frac{S_R}{A}\left(1 - \frac{A-1}{T-1}\right). \end{aligned} \tag{5.30}$$

That gives an interesting insight in the impact of A: when defining the gain, one must consider the trade-off between a better resolution of a fixed number of stages (higher A reduces the analog term more quickly) and a larger threshold margin (smaller A increases S_a). And because S_a obviously must be larger or equal to 0, the latter equation sets the upper limit $A \leq T$. Altogether A is strongly limited to

$$1 \leq A \leq T. \tag{5.31}$$

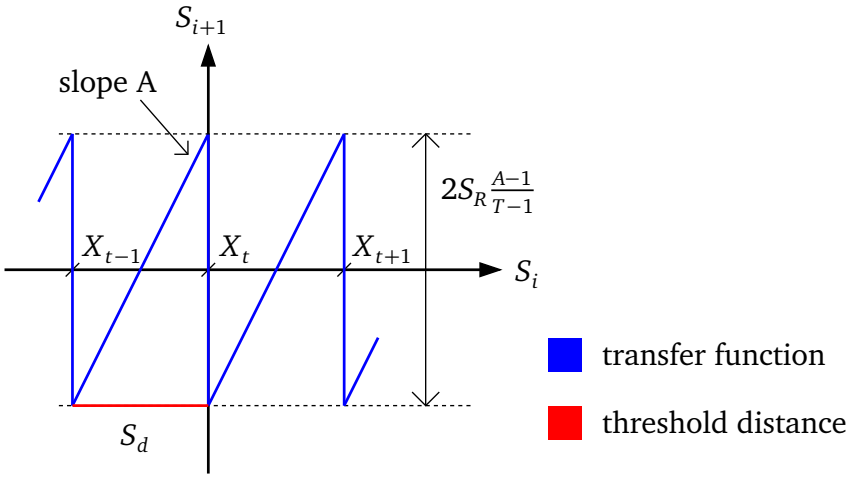


Figure 5.21.: Sketched positions of the comparator thresholds. Due to the constrained symmetry of both the signal range and the threshold margins, they must be evenly and symmetrically spread around the origin. Hence the constant distances S_d only depends on A, S_R and T.

Moreover, the symmetric postulation $S_a = S_b$ together with the earlier definition of D_i, implicitly defines the ideal positions X_t of all $T-1$ thresholds. Again due to simple geometric reasons and as sketched in Fig. 5.21, the constant distance S_d between the ideal positions of two neighboring thresholds must be

$$S_d = \frac{2S_R}{A}\frac{A-1}{T-1}. \tag{5.32}$$

And because the absolute positions X_t (due to the symmetry of the input and the output range) must be spread evenly around the origin, they finally can be calculated as

$$\begin{aligned} X_t &= \frac{2S_R}{A}\frac{A-1}{T-1}t \\ \text{with } \; t &\in \left\{-\frac{T-2}{2}, -\frac{T-2}{2}+1, \ldots, \frac{T-2}{2}-1, \frac{T-2}{2}\right\}. \end{aligned} \tag{5.33}$$

Not yet clearly formalized is the resolution in bit of the ADC in the absence of noise. In principle, and assuming that no digital errors occur, the analog term $\frac{S_N}{A^N}$ from equation 5.26 represents the remaining analog uncertainty (or the quantization error). In architectures that have a finalizing ADC stage with resolution r_{fin} behind N pipeline stages (e.g. a flash ADC), the total resolution in bit r_{eff} is simply

$$r_{\text{eff}}^{\text{close}} = \log_2(A^N) + r_{\text{fin}}. \tag{5.34}$$

If a pipeline ADCs has no dedicated finalizing stage, the last comparator effectively works as such. But in general – as just discussed – the resolution of the final comparator stage generally depends on the input signal, since the thresholds are not evenly distributed over the whole input range. The resolution becomes

$$r_{\text{eff}}^{\text{open}} \approx \log_2(A^{N-1}) + log_2(T) = log_2(TA^{N-1}). \tag{5.35}$$

For a very common ADC configuration, which at the same time was used to implement the SPADIC pipeline ADC, see further below (section 5.2.2.1).

5.2.1.2. Architectural Aspects

The iterative methodology that generates S_N and produces the N different w_i from the input signal S_0 can be realized (at least) in two fundamentally different ways. One can either process the analog signal N times with the same stage by "shortening" the input and the output node, or one can connect N stages in series. The first variant is named *cyclic*, the second for obvious reasons *pipeline*. From a theoretical point of view, the main difference between both is the trade-off between throughput and costs. One can coarsely say that a pipeline ADC has N times the throughput of a cyclic ADC, but consumes N times the power and the size.

More specific, a pipeline ADC requires a more complex digital evaluation logic and can suffer from a mismatch between the different stages, but at the same time can be adjusted to provide a comparatively better noise performance: if one considers the (uncorrelated) noise component σ_i introduced by stage i and appearing at the output of the i-th stage, the input referred noise of N stages with gain A is

$$\sigma_{\text{in}}^2 = \sum_{i=0}^{N-1} \left(\frac{\sigma_i}{A^i}\right)^2. \tag{5.36}$$

A worst-case assumption is that the noise of the i-th stage σ_i^{eff} scales as $\sigma_i^{\text{eff}} = \sigma_i/\sqrt{u_i}$ with u_i the scaling factor (number of parallel stages) of the i-th stage[1]. If one demands that each stage contributes the same amount of noise (note that $\sigma_i = \sigma_{i+1}$), the scaling factors must be set such that

$$\begin{aligned} \left(\frac{\sigma_i^{\text{eff}}}{A^i}\right)^2 &= \left(\frac{\sigma_{i+1}^{\text{eff}}}{A^{i+1}}\right)^2 \\ \Rightarrow \frac{\sigma_i \frac{1}{\sqrt{u_i}}}{A^i} &= \frac{\sigma_{i+1}\frac{1}{\sqrt{u_{i+1}}}}{A^{i+1}} \\ \Rightarrow u_i &= A^2 u_{i+1}. \end{aligned} \tag{5.37}$$

If one for instance sets the gain to $A = 2$ and the number of stages to $N = 3$, the scaling 16-4-1 will cause all stages to contribute equally. In comparison, a cyclic ADC would require a scaling factor of 7 of its sole stage in order provide the same noise performance[2]. Practically, the scaling factors are set less aggressively, which means that normally only some of the first stages are scaled (e.g. 9-3-1-1-1).

A completely different aspect concerning the concrete implementation is whether the ADC refers to current or voltage signals. As it becomes evident from the transfer function (compare again Fig. 5.19), the most crucial operation within the algorithmic procedure is the multiplication by A (or 2^k respectively) – whereas also the adder and the comparators must be designed carefully. Whereas voltage-mode ADCs are commonly based on switched capacitor circuits to perform the multiplication and profit from simple voltage comparators, current-mode ADCs use current storage cells to realize the multiplication and the subtraction, but mostly depend on complicated current comparators. A comprehensive comparison of both modes is complicated in general, because the various differences are only on a very low level of technical details. But if one counts publications, voltage-mode pipeline ADCs are clearly more popular – which is true at least for no obvious reasons. And moreover, the term “pipeline ADC” always refers to a “voltage-mode pipeline ADC”, while the term “current-mode” is normally explicitly added to the title of a corresponding publication.

5.2.1.3. Digital Evaluation Logic

As said before, in order to evaluate the final ADC result, some proper logic must compute the sum (equation 5.26)

$$S_{dig} = 2S_R \frac{A-1}{T-1} \sum_{i=0}^{N-1} \frac{w_i}{A^{i+1}}. \tag{5.38}$$

Numerically or technically, the division becomes a simple bit-shift operation, if the digital comparator results w_i are given in (or converted to) some binary representation and

[1] This approximation is for instance true if one only considers only thermal noise.

[2] Only in the particular case $A = 2$ it makes nearly no difference if one uses three 7-scaled cyclic ADCs or one 16-4-1 pipeline ADC.

at the same time the gain A is set to 2^k. Moreover, the effectively simple adder can be realized straight forward and usually stays comparatively small in terms of active area (and complexity). But in contrast, the number of required flip-flops can quickly get significantly large. The main reason for that is the fact, that the T different w_i from the N different stages/iterations do not only appear temporarily delayed, but are unfortunately produced "MSB-first". Hence normally a significant number of flip-flops to delay the w_i properly is required, which due to their large number and layout size actually dominate the evaluation logic.

In a cyclic ADC, the number of flip-flops to realize the delay is roughly proportional to the effective binary width of the w_i, which is simply $\lceil \log_2 T \rceil$, to the number of clock cycles c each stage requires (assuming only one clock domain), and to the number of stages N:

$$n_{\mathrm{FF}}^{\mathrm{cyclic}} \approx \lceil \log_2 T \rceil cN. \tag{5.39}$$

In a pipeline ADC, each w_i must be delayed by $c(N-1-i)$ cycles to correct the stage delay (demanding $\lceil \log_2 T \rceil c(N-1-i)$ flip-flops). Additionally ki flip-flops are necessary to realize the delay representing the division/shift. Hence the total number of flip-flops that is required is roughly

$$\begin{aligned} n_{\mathrm{FF}}^{\mathrm{pipeline}} &= \sum_{i=0}^{N-1} \lceil \log_2 T \rceil c(N-1-i) + ki = (\lceil \log_2 T \rceil c + k) \frac{N(N-1)}{2} \\ &\approx \frac{\lceil \log_2 T \rceil c + k}{2} N^2. \end{aligned} \tag{5.40}$$

Therefore the costs of the evaluation logic (at least in terms of size) scale linearly with the number of iterations N for a cyclic ADC but quadratically for a pipeline design. That is not really surprising if one considers that one pipeline ADC roughly corresponds to N cyclic ADCs.

5.2.2. General Aspects of the Implemented ADC

In the subsequent sections the most important aspects of the actually implemented ADC design are discussed. The current-mode pipeline ADCs operated in SPADIC 0.2, 0.3, and 1.0 are a modified versions of an already existing current-mode *cyclic* ADC [55].

5.2.2.1. 1.5 Bit Stage

As most published pipeline ADCs, the SPADIC ADC uses so-called 1.5 bit stages to introduce some threshold margin and thus to gain some digital error tolerance (as explained earlier in section 5.2.1.1). The naming simply refers to a comparator with two thresholds (compare Fig. 5.19), which produces three different digital values w_i ($T = 3$). Actually the number 1.5 bit is wrongly derived from the effective number of bits generated by the comparator. Strictly speaking, the correct value would be $\log_2(3)$, which is why the naming "1.6 bit stage" would indeed be more appropriate, but is practically unused though.

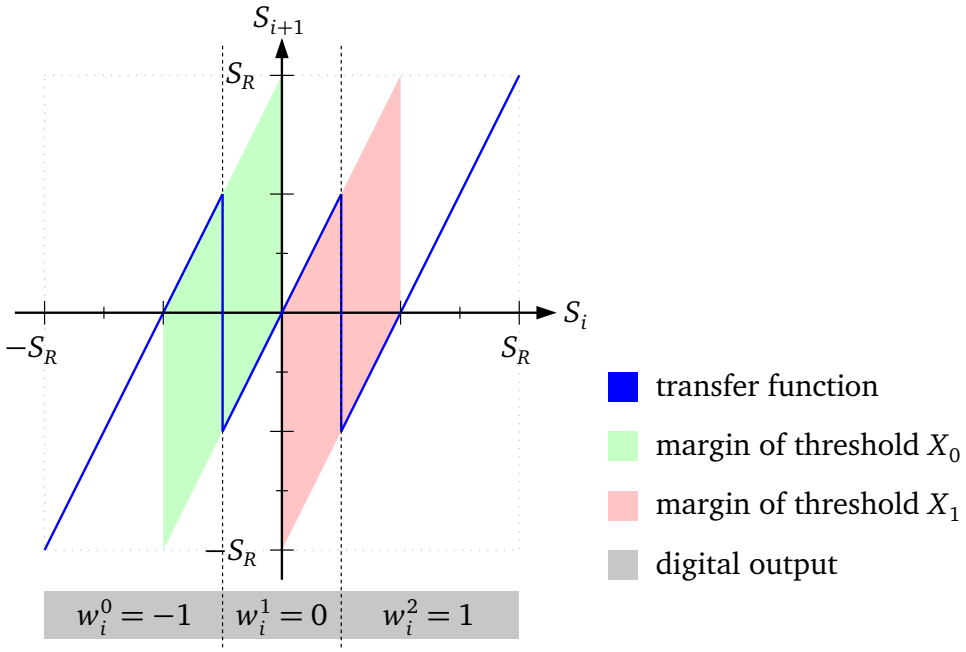


Figure 5.22.: The ideal transfer function of a 1.5 bit stage. The threshold margins of both thresholds are colored red and green. In the gray box the digital outputs w_i are assigned to their respective input range.

Using the formalization gathered further above in section 5.2.1.1, the 1.5 bit stage produces $T = 3$ different digital values w_i. Due to the constraint in equation 5.31, the gain A must be limited to $]1,3]$, but to simplify the division ($A = 2^k$), in this particular case the obvious (and only) choice is $A = 2$. According to equation 5.33 the two ideal thresholds must be set to $\pm S_R/4$, corresponding to an output step of S_R (equation 5.29), and each have a margin of $\pm S_R/4$ (equation 5.30). The digital outputs are (equation 5.22) $w_i^1 = -1$, $w_i^2 = 0$ and $w_i^3 = 1$ and with equation 5.35 the maximal resolution (ignoring noise and the not-equidistant threshold distribution) of the $N = 8$ implemented stages is $log_2(3 \cdot 2^7) = 8.58$ bit (9 bit are produced by the evaluation logic). The corresponding transfer function is shown in Fig. 5.22.

5.2.2.2. Processing Scheme

The SPADIC current-mode pipeline ADC is based on a novel type of current-mode storage cell (for details see section 5.2.3.1). In general, a storage cell has the three modes of operation write, read, and disconnected. In write-state the input current I_{in} is sensed and stored, in read-state the cell replicates the earlier stored value $I_{\text{out}} = -I_{\text{in}}$, and in disconnected-state the cell does nothing – although an eventually earlier written currents remains stored. The novel storage cell used to implement the algorithmic principle of the SPADIC ADCs is moreover able to add or subtract an offset current (required for the DAC in the algorithmic stage) and in particular offers the possibility to connect a (voltage!) comparator.

The algorithmic processing scheme that was used for the SPADIC ADC can be divided into 4-phases and is sketched in Fig. 5.23. Shown are 6 (of the 8) 1.5 bit stages of which

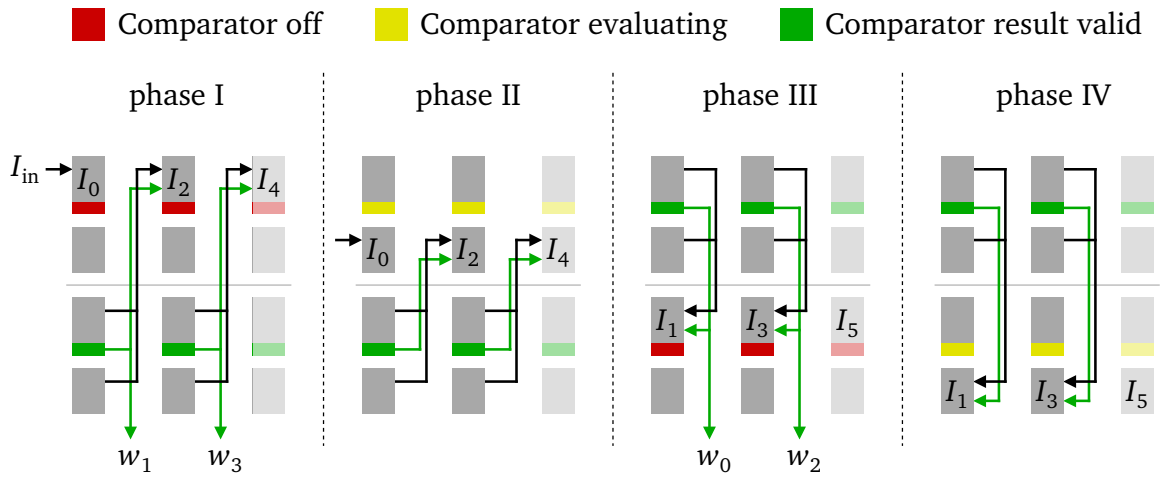


Figure 5.23.: The 4-phase processing scheme of the pipeline ADC. Shown are the first 6 1.5 bit stages, each consisting of two current storage cells (gray) and one comparator (red: off, yellow: evaluating, green: finished). The multiplication by $A = 2$ is done by copying the current twice, the offset correction (DAC) is realized with additional current sources in the storage cells (not shown) and depends on the comparator result of the previous stage. According to the formalization developed earlier, the currents written to the i-th stage are named I_i and the produced digital outputs w_i.

each consists of two current storage cells, one comparator, and some current sources in the storage cells (for the DAC, not shown). In phases I and II, the input current $I_{in} = I_0$ is written both to the first and the second memory cell. During phase II the comparator connected to the first memory cell evaluates the previously stored current I_0 and finishes just at the beginning of phase III. During phase III and IV, the initial current I_0 is produced twice by shorting the first two memory cells and setting them to read-state. Simultaneously, the comparator result of the first memory cell dictates how the offset is corrected (formal $-2w_0D_0$). Hence the current written to the third and forth memory cells in phase III and IV is $I_1 = 2(I_0 - w_0D_0)$ – just as desired. Of course, the same scheme is repeated in the subsequent stages, as also shown in the figure.

5.2.2.3. Stage Scaling

In order to equalize the single noise contributions of the different stages, the previously described technique of stage-scaling (equation 5.37) was partly used for the SPADIC pipeline ADC. According to the simulation a rather small scaling factor of 2 was predicted to be a good choice, so the actually realized scaling of the $N = 8$ stages is 4-2-1-1-1-1-1-1. Therefore a total of 24 current memory cells and a total of 16 comparators (two thresholds per stage, one comparator per threshold) are used for each ADC.

5.2.3. Details of Implementation

Subsequently further details and concrete calculations of the implemented ADC are summarized. Although some aspects, especially those of the memory storage cell, are better and more accurately described in the publication mentioned earlier (the very similar cyclic ADC [55]).

5.2.3.1. Current Storage Cell

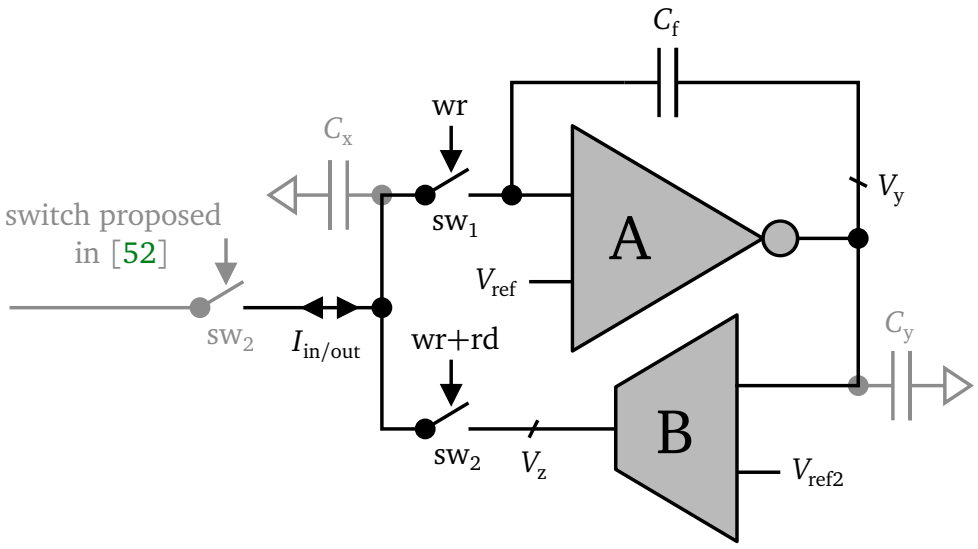


Figure 5.24.: Simplified schematic of the current storage cell used in the pipeline ADC.

The current storage cell used in the ADC is very similar to the zero-voltage switching cell proposed in [52], except for the position of the lower switch sw_2 (which is explained later). A simplified schematic of the cell is shown in Fig. 5.24. The idea is quickly described: to write the cell, both switches sw_1 and sw_2 must be closed. Then the input current can charge (or discharge) the feedback capacity C_f of the integrator and cause the internal voltage V_y to increase (decrease). In parallel the transconductor in the lower path reconverts the voltage V_y into a current that becomes subtracted from the input current until both are exactly equal – or in other words – until the input current flows completely into the transconductor. As soon as this state is reached, the charge on C_f and hence the voltage V_y remains stable. To bring the cell into read-state, sw_2 must be opened and since V_y stays constant, the transconductor still delivers the previously stored current. To disconnect the cell both switches must be opened (and the stored current temporarily dumped), while the stored current (or charge respectively) stays unchanged.

A very important detail of the cell is the constant and regulated potential of the input/output node both during read- and write-state. In write-state the closed loop forces the input node of the integrator to stay at V_{ref} (virtual ground) and since no current flows through sw_1 as soon as the cell has settled. The same is true for the input/output node of the cell. At the same time the voltage V_z stabilizes at V_{ref} minus the drop-off voltage over sw_2. If the cell

changes to read-state (by opening sw_1), the input/output potential of the cell remains at V_{ref}, which it now externally forced by another storage cell entering write-state. Due to the equal and constant potentials on both sides of switch sw_1, the charge that is injected onto C_f when the switch is opened, is independent of the input signal and only adds a constant offset. That offset effectively causes a constant shift of the digital results but does not affect the resolution or the linearity of the ADC.

Moreover, another important detail and actually the reason why the switch sw_2 was moved with respect to the initial proposal in [52], is the good stability of V_z after sw_1 was opened. Since both the input potential and the voltage drop over sw_2 stays the same (still the same current is flowing through sw_2), the potential V_z does not change either when going from write-state to read-state. That detail reduces the general need for a high output resistance of the transconductor significantly.

Very beneficial is moreover the existence of the internal voltage node. Here a common voltage-mode comparator can be connected to concurrently evaluate the stored current (since $V_y \propto I_{in}$), and hence no complicated and destructive current comparator is required.

The achievable resolution of the ADC (and therefore implicitly the reasonable number of stages) is limited by the relative current error that adds each time the current is copied. It is derived in [55] and given as

$$\frac{\mathrm{Var}(\Delta I_{\mathrm{out}})}{I_{\mathrm{max}}^2} = \frac{(C_x^2 + 16(C_f + C_x)C_y)kT\gamma n}{(C_f + C_x)C_y C_f V_{\mathrm{y\_swing}}^2}, \tag{5.41}$$

with the parasitic capacitors C_x and C_y as sketched in Fig. 5.24 (gray), k the Boltzmann constant, T the absolute temperature, $n \approx 1.5$ the slope factor, $\gamma = 2/3$, and $V_{\mathrm{y\_swing}}$ the maximum voltage swing of node V_y.

The characteristic settling time constant of the memory cell is also derived in [55]:

$$\tau_s = \frac{C_f}{B}, \tag{5.42}$$

with B the gain (transconductance) of the transconductor, although the actual settling time of the current cell is about three times larger ($3\tau_s$).

The current storage cell was designed such that the nominal sampling period ($\hat{=}$ 4 settling times or $12\tau_s$) is 40 ns ($\hat{=}$ 25 MS/s). Moreover, the relative current error finally led to an effective resolution of about 8 bit (8 stages, 9 bit output).

5.2.3.2. Interface Between ADC and CSA

The connection between shaper and ADC has to be adjusted carefully. On the one hand, the two CSA outputs produce a voltage signal (pos.: 0.5 to 1.5 V, neg.: 1.2 to 0.3 V) and on the other hand the ADC input expects a current signal (± 32 µA, input potential 900 mV). The simple solution (see Fig. 5.25) used in the SPADIC is to convert the signal with a series resistor (15 kΩ)[1], which maps the output swing of roughly 1 V to 66 µA. In order to be able to adjust the baselines properly (neg.: 0.5 V → -32 µA, pos.: 1.2 V → +32 µA), two current

[1] The thermal noise introduced by the resistor can be neglected here.

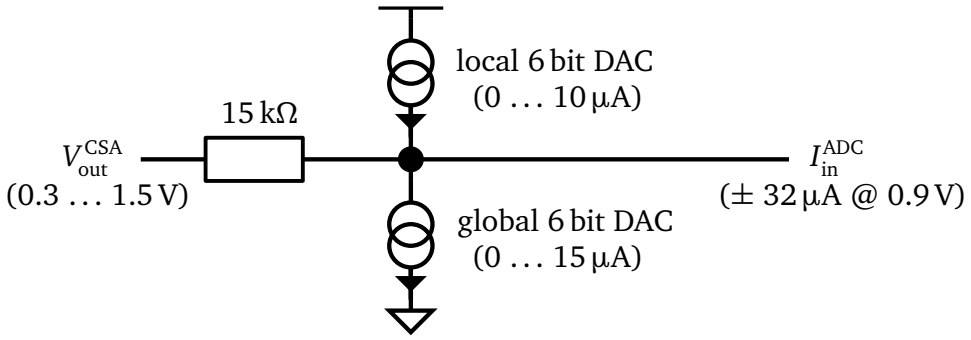


Figure 5.25.: Schematic of the interface cell between CSA and ADC.

sources were added to the ADC input. Whereas the lower source can be configured globally, the upper source is controlled by a local trim DAC to allow for local offset corrections.

5.2.3.3. Radiation Tolerance

Not yet mentioned has been the fact that the whole pipeline ADC was designed to be radiation tolerant (for details see also [55]). To cut potential leakage paths due to ionized oxides, guard rings and "round" NMOS transistors were used throughout the design. Moreover, due to their smaller sensibility to radiation-induced effects, PMOS transistors were preferred over NMOS transistors wherever possible. And in fact, since the current-mode design offers constant DC operation points at most internal nodes, most of the NMOS transistors could be avoided in the first place. See section 5.5 for a more general discussion on analog radiation tolerance and the techniques just described.

5.2.3.4. Bias and Configuration

The ADC uses the same bias block and configuration registers as the CSA (see section 5.1.3.3). The total of 12 internally generated bias voltages of the ADC share only 7 analog pads (for external decoupling and monitoring), simply due to the limited number of available pins. Not at least for that reason (and similar to the CSA), all bias voltages are decoupled internally wherever possible. The 7 available analog decoupling pads are each internally connected to the output of an analog 3 : 1 multiplexer. That way the 12 bias signals could be properly distributed over 21 multiplexed output wires. But due to that limitation, not all combinations of 7 signals can be decoupled (or monitored) at the same time, though a high effort was taken to allow at least for the (probably) most important combinations. Moreover, each three signals sharing one multiplexer or one analog pad were selected such that all have the same decoupling target (vdda, gnda, etc.). That way the external decoupling circuit is suitable for every possible combination.

Besides the main power (vdda) and ground (gnda) nets (internally separated from the CSA), the ADC requires two additional power signals (the source RefIn $\approx 0.9\,V$ and the sink AmpLow $\approx 0.3\,V$). In the present ADC design that is probably one of the largest drawbacks, since RefIn and AmpLow both require a separated power supply and cannot, due to their

intermediate potentials, be simply connected to vddc and gndc (the power nets of the CSA) outside the chip.

5.2.3.5. Monitoring and Test Signal Injection

Similar to the monitor bus of the shaper, a bus for monitoring and signal injection can be connected to any ADC input (only one ADC should be connected at once). In fact the simple bus it a very important tool for ADC diagnoses and measurements. Most important in this context is that the bus allows to inject a well-defined current into the ADC, which allows both for static and dynamic (limited to lower frequencies due to the low-pass characteristic of the bus) measurements.

5.2.3.6. Layout

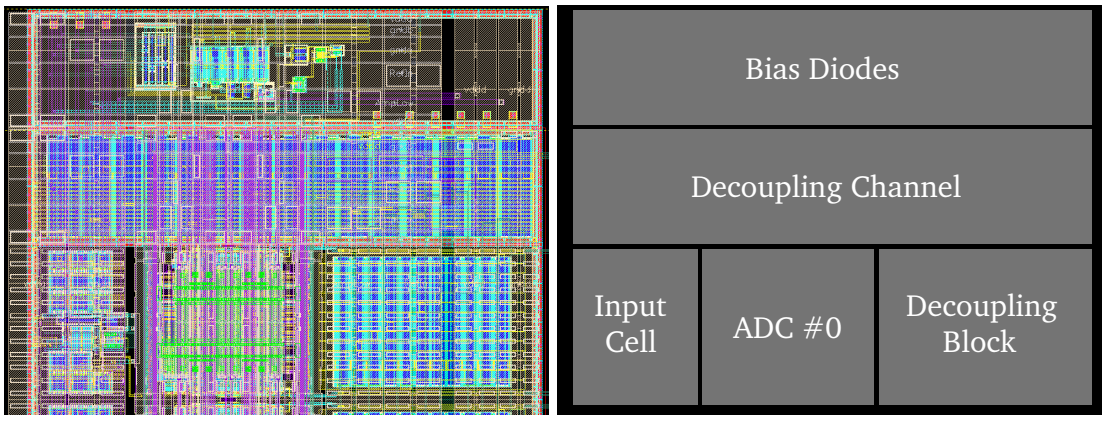


Figure 5.26.: Cut-out (400 µm x 300 µm) of the upper ADC layout. Shown are the first ADC (including input cell and decoupling block), some of the bias diodes, and a part of the decoupling channel.

A cut-out of the ADC layout of SPADIC 1.0 is shown in Fig. 5.26. The ADC core occupies only 130 µm x 120 µm (excluding the digital evaluation logic). It is encapsulated by several bias structures, which actually dominate the total area of the whole ADC layout in this particular design. Similar to the CSA, each block of 8 ADCs is framed by a dedicated decoupling channel and flanked by an additional decoupling block, which simply exploits the free space under the wide bias and power buses on the right hand side (in contrast to the CSA the ADC power pads are only placed on top and on bottom of the die, hence the buses span over a distance of roughly 5 mm). Also similar to the CSA, all layout blocks are individually surrounded by guard rings and are designed as modularly as possible (the ADC layout was already given, although most layout parts had to be adjusted). Of course the vertical ADC channel granularity matches exactly the CSA channel granularity (including the bias and decoupling channels) – and in particular no additionally routing is required, if the layouts of CSA and ADC are properly placed next to each other.

5.2.4. Selected Measurements

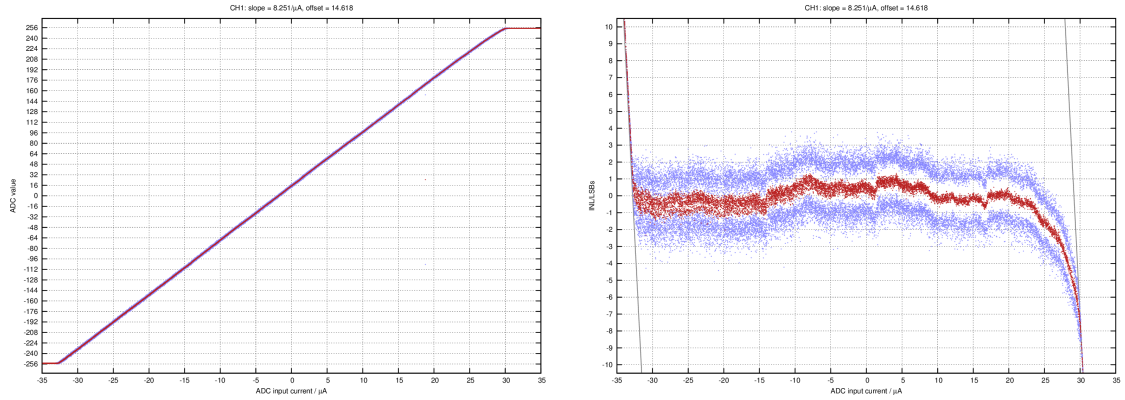


Figure 5.27.: ADC Measurement with static input currents. The ADC was operated at 20 MS/s, both plots relate to 9 bit output values.

The result of a static ADC measurement is shown in Fig. 5.27. For the measurement the whole digital and most analog parts of SPADIC 1.0 have been operated. The plot on the left hand side shows the 9 bit output values as a function of the (static) input current (range ± 32 µA), whereas the plot on the right hand side is the corresponding INL (deviation from a linear fit). Some hundred samples were recorded for each plotted point, red points are mean values and purple points visualize the standard deviation (one sigma). The input currents were set at random — that is not simply stepwise increased – and the whole measurement took several hours. Therefore the plot includes equally spread long-term variations (baseline drift, etc.) as well. It is important to note that the measurement has not yet been made at optimal bias conditions, but already nearly meets the expectations gained from the simulation.

5.2.5. Summary Table: ADC

Parameter	**Value**
Sampling rate	25 MS/s
Power consumption	≈ 5.3 mW (decoder 0.3 mW)
Resolution, nominal	9 bit
Resolution, effective	≈ 8 bit
Size of analog layout (core)	140 x 120 µm^2
Area digital decoder	8995 µm^2 ≙ 75 x 120 µm^2
Total required area	25 795 µm^2 ≙ 215 x 120 µm^2
Actually used area	58 195 µm^2 (power, input cell and decoupling)
Characteristics	rad-tolerant layout
	current-mode
	2's complement output

5.3. System Performance: CSA + ADC

So far the analog building blocks CSA and ADC have been discussed separately. But in fact many important questions can only be answered properly, if one analyzes both together and especially considers their dynamics. Important questions are for instance: What is the effective amplitude resolution of the recorded pulses considering electronic noise, the quantization error, and multi-sampling? How fast should or must be sampled given the CSA characteristics? How many samples per pulse have to be recorded and what samples should be selected? What effective time resolution can be achieved? Et cetera.

5.3.1. Effective Amplitude Resolution

To quickly deliver some answers an important assumption is made now: for the following calculations the single sampling errors (quantization or noise) Δa_i of the different ADC samples a_i are taken to be uncorrelated. That assumption is very realistic for large sampling periods, but becomes increasingly wrong if the sampling rate increases. In general, the smallest possible sampling frequency before information gets lost is defined by two times the system bandwidth (Nyquist-Shannon sampling theorem [65]). In the present case, the required sampling rate can be roughly estimated as two times the bandwidth of the CSA: $f_{\text{sampling}} \approx 2 f_{\text{shaper}} = 1/(\pi\tau_s)$, with τ_s the shaping time. For the previous reasons, the following conclusions are only valid within the range of moderate sampling periods up to about $\pi\tau_s$ ($\approx 250\,\text{ns}$ in SPADIC 1.0).

The signal to noise ratio is commonly defined as $S/N = \mu/\sigma$, with σ the standard deviation of the relative noise error (or more general of some statistical error) and μ the mean signal amplitude (at the position of interest). Hence a single-sample ADC that samples an analog signal with the mean signal shape function $\mu(t)$ at the time T_0 has a signal to noise ratio of

$$S/N_{\text{single sample}} = \frac{\mu(T_0)}{\sigma}. \tag{5.43}$$

If more than one sample is taken, the S/N contributions of the different samples add quadratically (uncorrelated sampling errors have been assumed). That directly leads to a formula for the effective S/N after N samples have been taken (with t_0 some time offset and T_{ADC} the sampling period)

$$(S/N_{\text{eff}})^2 = \sum_{i=0}^{N-1} \left(\frac{\mu(iT_{\text{ADC}} + t_0)}{\sigma} \right)^2 = \frac{1}{\sigma^2} \sum_{i=0}^{N-1} \mu(iT_{\text{ADC}} + t_0)^2. \tag{5.44}$$

The equation directly tells that only those samples effectively contribute to S/N_{eff} which are on average larger than zero ($\mu(t) > 0$). That leads to at least three important conclusions. First, samples of the baseline (obviously) can neither improve the resolution nor impair it. Second, if the number of samples is limited (which is always the case), one should simply take the values having the largest average (if the pulse shape is known well

enough and assuming that the baseline is either known or stable). Third, the best achievable resolution can be easily evaluated mathematically by assuming an unlimited number of samples (if the pulse length is finite).

The theoretical pulse shape produced by the CSAs of all SPADICs designed so far is simply $\mu(t) = Ae\,(t/\tau_s)e^{-t/\tau_s}$ for $t > 0$ (equation 5.7 for $N = 1$ and pulse amplitude A normalized to $[0,1]$) or $\mu(t) = 0$ else. Thus the latter formula for a SPADIC pulse becomes

$$S/N_{\text{eff}} = \frac{Ae}{\sigma\tau_s}\left(\sum_{i=0}^{N-1}(iT_{\text{ADC}} + t_0)^2 e^{-2(iT_{\text{ADC}}+t_0)/\tau_s}\right)^{0.5}. \tag{5.45}$$

For a numerical evaluation of this formula the noise component σ must be better understood and properly scaled. It is assumed here that the noise is only introduced by the CSA, whereas the ADC only adds a quantization error. The relative noise of the CSA scaled to the normalized input range $[0,1]$ is $\sigma_{\text{CSA}} = \sigma_{\text{ENC}}/\max\{Q_{\text{in}}\}$ and the absolute quantization error of an ideal ADC with n bit going from 0 to $2^n - 1$ is $1/\sqrt{12}$[1]. Properly scaled, this leads to the relative quantization error $\sigma_{\text{ADC}} = 1/(\sqrt{12}(2^n - 1))$ and the standard deviation of the total noise becomes

$$\sigma = \sqrt{\frac{\sigma_{\text{ENC}}^2}{\max\{Q_{\text{in}}\}^2} + \frac{1}{12(2^n-1)^2}}. \tag{5.46}$$

Moreover, it is beneficial to reconvert the effective S/N_{eff} into an equivalent ADC resolution (in bit) and thus to bring the results into a more intuitive format. This can be done by using the simple formula

$$n_{\text{eq}} = n + \log_2\left(\frac{S/N_{\text{eff}}}{S/N_{\text{ADC}}}\right) = n + \log_2\left(S/N_{\text{eff}} \cdot \sigma_{\text{ADC}}\right), \tag{5.47}$$

where S/N_{ADC} is simply the initial S/N of the ideal ADC. Note that the resulting equivalent ADC resolution n_{eq} can be better as well as worse than the initial ADC resolution n – in dependency on the achieved S/N_{eff}, which can get better due to oversampling or worse due to noise.

For the subsequent calculations typical design parameters from SPADIC 1.0 are taken. They are $\tau_s = 80\,\text{ns}$, $\sigma_{\text{ENC}} = 800\,\text{e}$ (at 30 pF), $\max\{Q_{\text{in}}\} = 75\,\text{fC}$, and $n = 8$.

With equation 5.46 the noise component can be directly calculated as $\sigma \approx 2.05 \times 10^{-3}$. If only the amplitude A of the pulse was sampled, the effective S/N_{eff} would be $A/\sigma = A \cdot 487.8$ (equation 5.45 for $N = 1$ and $t_0 = \tau_s$), which corresponds to an equivalent ADC resolution of 7.14 bit. In that case the total system resolution was (mainly) limited by the noise of the CSA – and not by the ADC resolution.

In strong contrast to one single sample, the best achievable equivalent ADC resolution n_{eq} can be evaluated considering an unlimited number of samples. The optimal n_{eq} (using the SPADIC 1.0 parameters) as a function of the sampling period T_{ADC} is shown in Fig. 5.28 (for $A = 1$). The plot on the left hand side shows n_{eq} for various sampling offsets t_0 (normalized to $t_0' = t_0/T_{\text{ADC}}$, red). The green line visualizes the earlier calculated effective

[1] That is simply the standard deviation of a box distribution with length 1 ($\hat{=}$ 1 LSB).

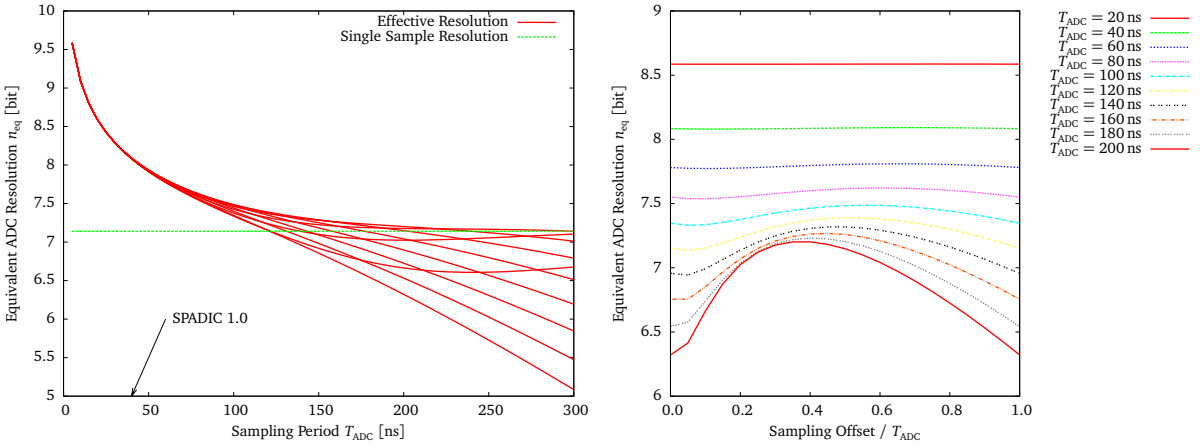


Figure 5.28.: Left: best achievable equivalent resolution n_{eq} of the SPADIC system as a function of the sampling period T_{ADC}, plotted for different sampling offsets t_0. Right: the equivalent resolution as a function of the sampling offset, plotted for various sampling periods.

resolution, if only the amplitude of the pulse is sampled. It becomes evident that decreasing the sampling period helps to increase the effective resolution – as predicted. As mentioned before though, this statement is only true to a certain limit, because eventually the single noise contributions start to correlate (that was not considered in the plot). According to the plotted result the equivalent resolution of the SPADIC 1.0 system for $T_{ADC} = 40\,\text{ns}$ (and $\sigma_{ENC} = 800\,\text{e}$ at 30 pF) is 8.08 bit. The plot on the right hand side also shows the equivalent resolution, but this time as a function of sampling offset t_0. Thus the graph shows how different arrival times of the pulse (with respect to the sampling clock) affect the achievable resolution. One can estimate that the strong sensitivity to the sampling offset vanishes for sampling periods of about 100 ns and below.

An open question still is how many samples should be chosen. As said before, the formula for S/N_{eff} implicates directly that the contribution to the effective S/N is proportional to the mean signal height of the pulse at the sample position. Hence simply the N largest samples should be selected. Since the SPADIC pulses have short rise-times but relatively long tails, a perfect approach is to simply take the first N samples – except for $N = 1$, in which case one should take the second value instead of the first. Using the parameter set of SPADIC 1.0 again, Fig. 5.29 was calculated. Shown is how the equivalent ADC resolution depends on the first N samples. It is clearly visible that already 6 samples are sufficient both for a resolution close to the optimum and good insensitivity to the arrival time of the pulse. But in order to monitor the baseline some additional samples might be required.

5.3.2. Effective Timing Resolution

Very similar to the effective amplitude resolution, the effective time resolution can be calculated. As explained in section 5.1.2.3, the standard deviation of the achievable time resolution in the presence of noise is (ignoring clock jitter)

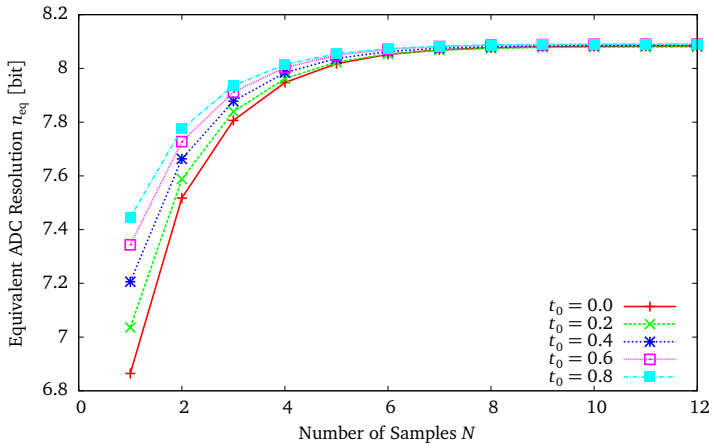


Figure 5.29.: The equivalent ADC resolution as a function of the number of samples. For the calculation simply the first N samples of the pulse were selected (which is actually very close to the optimum in the case of the SPADIC pulse shape, even for small values of N).

$$\sigma_{\text{time}} = \frac{\sigma_{\text{noise}}}{\left.\frac{\mathrm{d}\mu(t)}{\mathrm{d}t}\right|_{t_s}}, \tag{5.48}$$

with t_s the sampling time (or any time in-between two samples respectively) and $\mu(t)$ again the mean signal shape. Since for N samples one gets $N-1$ slopes, the effective time resolution if several samples are taken becomes (with similar considerations as before)

$$\begin{aligned}
\sigma_{\text{time_eff}} &= \left(\sum_{i=0}^{N-2} \left(\frac{\sigma}{\left.\frac{\mathrm{d}\mu(t)}{\mathrm{d}t}\right|_{iT_{\text{ADC}}+t_0}} \right)^{-2} \right)^{-0.5} \\
&= \sigma \left(\sum_{i=0}^{N-2} \left(\frac{\mu((i+1)T_{\text{ADC}}+t_0) - \mu(iT_{\text{ADC}}+t_0)}{T_{\text{ADC}}} \right)^2 \right)^{-0.5} \\
&= \sigma T_{\text{ADC}} \left(\sum_{i=0}^{N-2} \left[\mu((i+1)T_{\text{ADC}}+t_0) - \mu(iT_{\text{ADC}}+t_0) \right]^2 \right)^{-0.5}.
\end{aligned} \tag{5.49}$$

Inserting the SPADIC pulse shape leads to

$$\begin{aligned}
&\sigma_{\text{time_eff}} = \\
&\frac{\sigma T_{\text{ADC}} \tau_s}{A} \, \mathrm{e}^{\left(\frac{T_{\text{ADC}}+t_0}{\tau_s}-1\right)} \left(\sum_{i=0}^{N-2} \left[((i+1)T_{\text{ADC}}+t_0)\mathrm{e}^{\frac{-iT_{\text{ADC}}}{\tau_s}} - (iT_{\text{ADC}}+t_0)\mathrm{e}^{\frac{(1-i)T_{\text{ADC}}}{\tau_s}} \right]^2 \right)^{-0.5}.
\end{aligned} \tag{5.50}$$

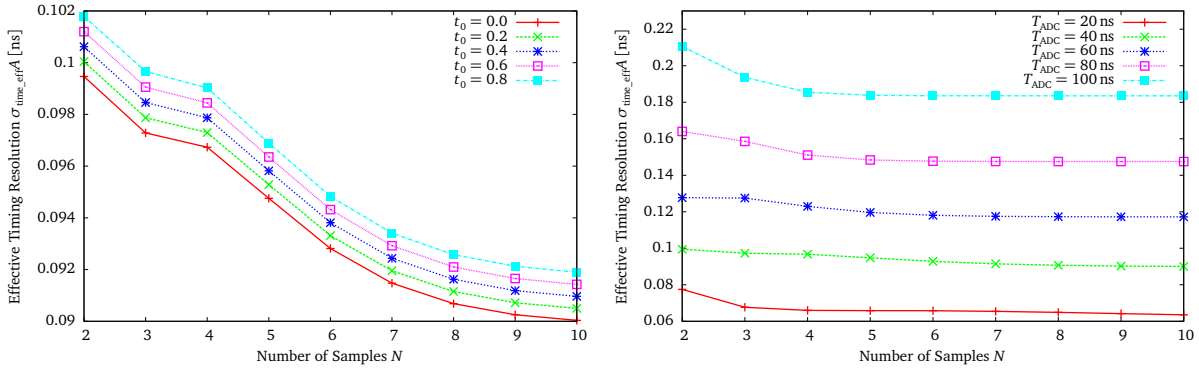


Figure 5.30.: The effective time resolution $A{\cdot}\sigma_{\text{time\_eff}}$ as a function of the number of samples. The best achievable time resolution mostly depends on the sampling period T_{ADC} and most of the time information is contained in the first two samples.

In both plots of Fig. 5.30 $A \cdot \sigma_{\text{time\_eff}}$ as a function of the number of samples N is plotted. The same values as before were used for τ_s and σ. For the plot on the left hand side T_{ADC} was set to 40 ns while the sampling offset t_0 was varied. The resulting curves are in principle magnified versions of the green curve in the graph on the right hand side. Hence the plot on the right hand side shows $A \cdot \sigma_{\text{time\_eff}}$ on a larger scale but for different sampling periods T_{ADC}. It becomes evident how – similar to the effective amplitude resolution – the effective time resolution gets better, if the number of samples is increased, although the effect is very small. The reason for the latter is the asymmetric shape of the CSA pulses, which have very short rise-times but only slowly decreasing tails. Therefore most of the time information is contained in the first two samples, whereas the subsequent samples contribute only sparsely[1]. Hence as long as the first two samples are selected, the choice of the number of samples should only depend on the effective amplitude resolution.

According to the latter calculations the at least theoretically best achievable time resolution of SPADIC 1.0 for a typical signal pulse of $A = 0.26$ ($\hat{=}$ 20 fC/75 fC) is about 350 ps (excluding clock jitter).

5.4. Analog Building Blocks

Besides the two main parts CSA and ADC various secondary but also important analog building blocks were designed for or at least adapted to the different SPADIC versions. Therefore short descriptions of selected secondary analog building blocks are given in this section.

5.4.1. Analog Shift Register

The analog shift register used for the DAC configurations as well as for most of the static analog settings (enable and selection switches) is based on the two-phase register cell

[1] That is by the way an argument against higher order shaping.

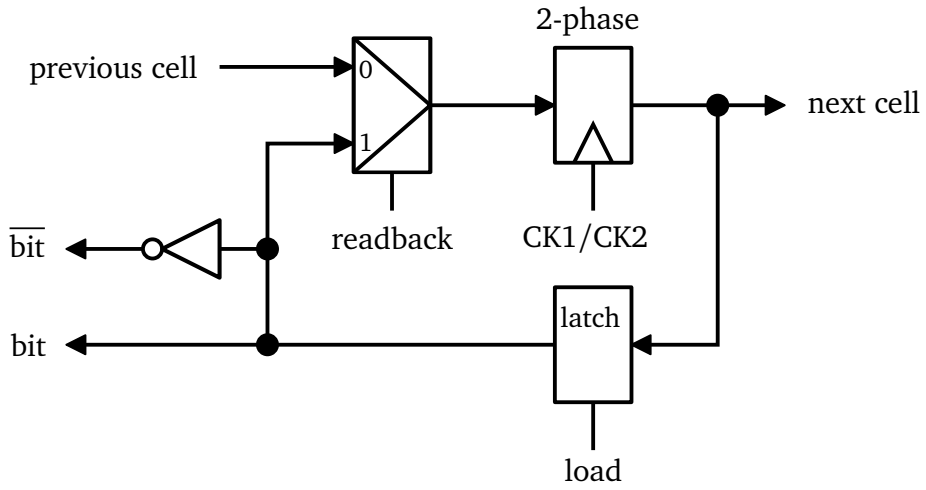


Figure 5.31.: Schematic of the analog two-phase shift register cell used to realize the analog configuration register.

sketched in Fig. 5.31. The register cell was not designed directly for the SPADIC design but was already available and is used in other projects too. Its functionality is simple: to write a chain of cells (*readback* = 0), the configuration is serially shifted in by properly applying the two non-overlapping clock pulses *CK1* and *CK2*. Then the complete bit vector is stored at once, if a single *load* pulse triggers the latch in the lower path. To serially read back the configuration stored before the multiplexer must be toggled (*readback* = 1) for one *CK1*/*CK2* cycle. Afterwards the data can be shifted out again. The two-clock scheme is used here to avoid setup- and hold-time problems, which are difficult to avoid and laboriously to detect in manually designed and routed analog blocks (especially the analog delays and a realistic analog clock-tree behavior are difficult to extract and simulate).

Especially in SPADIC 1.0 much care was taken to properly drive the very long global wires *CK1*, *CK2*, *load*, and *readback* (wire lengths > 5 mm, extracted load per wire ≈ 2 pF). Therefore a dedicated driver cell has been designed, which is able to handle at least a serial shift-speed of 25 Mbit/s.

The whole analog shift register of SPADIC 1.0 was connected to dedicated control logic blocks realized in the digital (synthesized) part. The analog configuration therefore is only accessible via the digital interface. More specific, the write and read logic of the analog shift register was logically mapped to a certain address of the register file that is otherwise used to store the complete configuration of the digital part.

An important future improvement of the analog register cell would be to make sure that the cell is in a well defined state directly after it has been powered. So far the initial state of the cell is arbitrary, which leads to an unpredictable status of the whole analog part after power-on. In particular, the initial current consumption changes with each power cycle. A very simple solution would be to introduce some global analog reset signal, which would be released after the power supply has settled.

5.4.2. Standard Cell Library

For the large digital part of SPADIC 1.0 a home-made standard cell library was used. Although standard cells are applied in the digital domain, designing them is predominantly an analog process — which is why the used library is described briefly in this "analog chapter".

The so-called universal core library (UCL) was designed for the 180 nm UMC technology in order to have a general purpose standard cell library for different applications, of which the SPADIC was and is a very important one. Although some experience was gathered before with similar libraries (e.g. a radiation-tolerant library for the same technology), designing the UCL meant a step into a new territory and to handle various minor and major unforeseen problems. Starting with only a handful of absolutely necessary cells, the latest UCL version has roughly 60 cells and is still growing. The library is effectively the common result of many designers working on different projects but eventually committing some kind of contribution to the shared design (e.g. a new cell, a script change, or a bug-fix). For instance some dedicated mixed logic gates and a half-adder particularly beneficial for the SPADIC-IIR filter design were added eventually.

Of course the number of cells in the UCL is fairly small compared to commercial standard cell libraries, which normally provide at least several hundreds of cells. But the possibility to have unrestricted access to its own library also has crucial advantages: the usage is completely free, everything can be analyzed and evaluated on a very high level of detail, crucial parts can be simulated in the analog domain, individual adjustments can be made, and even DRC and LVS checks on transistor level become possible. But besides the obvious advantages, the UCL provides the unusual feature of separated substrate contacts. More specific, the substrate of the UCL is completely isolated from the p-doped chip bulk, because all UCL logic cells are embedded in n-wells, whereas NMOS transistors in turn are placed in p-doped tri-wells. Moreover, the bulk contacts of the n-wells and tri-wells are routed separately from the power connections (vdd and gnd), which additionally allows for a better isolation between bulk material and digital signal transitions. Especially in mixed-signal designs such as the SPADIC, where the analog circuitry is extremely sensitive and at the same time the digital part dominates (at least the power and area consumption), a good separation between the digital and the analog domain is extremely important.

If one has access to modern ASIC design tools, the main task when developing a standard cell library comes down to the production of schematics and matching layouts (and maybe some HDL code) – although countless smaller technicalities, parameter settings and constraints must be considered or set properly. The rest then can be done with the help of a proper library characterization tool, which can be everything from a home-made collection of scripts to a dedicated commercial program. And indeed, both of the latter "extreme" approaches have been made: a forerunner of the UCL library was designed using home-made scripts (again controlling commercial tools like for example the analog simulator), whereas the UCL library itself was realized with a professional library characterization tool.

In order to make a standard cell library usable, basically three types of information must be extracted (by the tools or by hand) from the layouts and the schematics: first, a whole set of analog information must be gathered, such as timing tables listing simulated values of all possible timing arcs (as a function of input transition times, load capacities, environmental parameters, etc.) or simulated values of the static and dynamic power consumption.

Second, an abstract geometric cell description (for instance of pin positions, cell sizes, or forbidden areas) must be generated. And third, some kind of of a functional description (behavioral description in HDL, logic equations, etc.) must be produced. The extracted output is usually stored in some set of files, which is later read in again by the semi-custom design tools (see section 6.7).

In the case of the UCL, the simulated analog characteristics is stored in a library file (LIB file), which basically contains all simulated characteristics (timing, power, capacities, etc.), whereas all geometrical information required for place and route is given in a layout description file (LEF file). Additionally, a manually prepared set of HDL (Verilog) files allows for normal and post-place-and-route simulations.

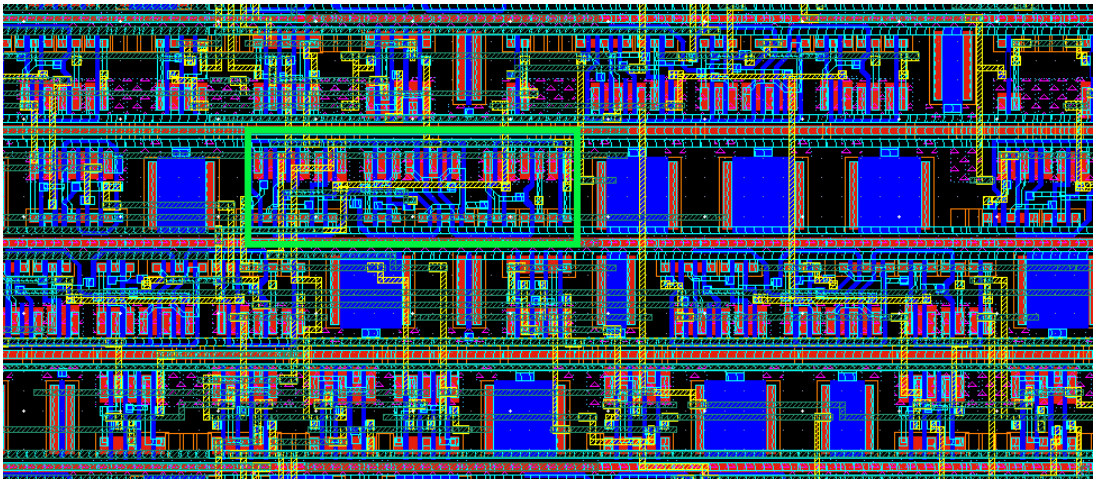


Figure 5.32.: Cut-out of the digital part showing some of the UCL standard cells. Marked is a D flip-flop (with reset).

To give at least a small insight in the layout of the UCL design[1], Fig. 5.32 shows a small cut-out of the digital part of SPADIC 1.0. In the layout a D flip-flop is marked, its size is 14.4 µm x 5.76 µm. It was made sure that all cells are on the same coarse grid and that no short or open can occur whenever any two kinds of cells are placed next to each other. With very few exceptions, most of the internal routing was done either on the polysilicon or the first metal layer (6 metal layers are available in the present technology). The pins are placed also on a coarse grid exceptionally using the second metal layer[2]. The layout makes use of small layout bricks to avoid redundancies as much as possible (that was indeed the inspiration for the modular SPADIC layout). Because both the isolated n-wells and tri-wells require their own substrate contacts and because both are purposely not shorted to power or ground, four (instead of two) horizontal power lines are to be routed and connected, which has two smaller drawbacks: on the one hand, the (semi-)automatic power routing is more complex and complicated (see section 6.7) and on the other hand the vertical

[1] The schematics are straight forward and are hence not further shown. Moreover, on the one hand the non-disclosure agreement with UMC disallows to show layout details and on the other hand since the UCL is a rather secondary issue here it shall not be too much expanded.

[2] The definition of a dedicated pin position on some grid is not really required by modern routing tools which can connect everywhere, but at least helps to keep things clean and organized.

expansion of the cells and hence the total area per cell is slightly larger then it would be without separated substrate nets.

5.4.3. IO Cells

Similar to the standard cells, all IO cells used in any SPADIC version were no commercial IP cells but home-made. Even though all already available IO pads for the 180 nm technology were at least slightly modified to better adjust for the actual SPADIC 1.0 requirements. In doing so, the major and simple change was an increase of the pad pitch from 80 µm to 95 µm, which was done to relax the requirements for the wire-bonding and the PCB spacing. The new pitch was chosen such that the most dense and noise critical side where the detector is connected, just provides enough pads for all 32 CSA inputs plus a sufficient number of power and ground pins. That way a number of 49 pads per edge or 196 pads per chip was defined. Besides the minor layout and pitch adjustments two important new IO pads were added to the library: the spark protected input pad which has been shown earlier in this chapter and a bidirectional open-drain pad which is required for the bidirectional SDA data bus of the I^2C slow-control interface.

According to the strict power separation scheme of SPADIC 1.0 (guard rings between all blocks and channels, several separated power domains, carefully separated digital and analog parts, etc.), the power rings generally passing all IO pads were properly cut. That way not only the power and ground domains but also the IO protection nets could be isolated. In doing so, 4 different ring sections were created: one section for the CSAs (mainly left side of the die), two sections for the ADC (one on the upper and one on the lower left side), and one large section for the digital part (the upper right, the right, and the lower right edges).

5.5. Analog Radiation Tolerance

Radiation tolerance in general is both an important design aspect and a complicated field of research. And in fact, radiation tolerance is the only design goal, which has not been explicitly set for SPADIC 1.0 so far and will therefore become one of the main issues of the next and maybe final iteration. Nevertheless, radiation tolerance was not completely ignored in SPADIC 1.0 – for instance in the case of the pipeline ADC (section 5.2.3.3). See section 6.6 for details on the digital radiation tolerance.

For the analog part total ionizing dose effects (TID) are by far more relevant than single event upsets (SEU) or single event transients (SET), which instead play a major role in the digital part. In the present case (UMC 180 nm technology) two TID effects dominate: the ionization of the field oxide can lead to leakage currents, whereas the ionization of the gate oxide can cause threshold shifts. Both effects are dominantly caused by trapped holes at the junction between silicon-dioxide and silicon bulk. In general, PMOS transistors of deep-submicron processes are much less affected by radiation-induced effects than NMOS transistors [10]. The reason for that is that the ionized holes in the dioxides (electrons tunnel out eventually) in the case of PMOS (and in contrast to NMOS) transistors normally travel away from the junction between the silicon-dioxide and the silicon bulk and hence

towards a region where they have very little influence on the threshold of the (parasitic) transistor. The orientation of holes in the PMOS dioxide away from the critical junction simply comes from the fact that the PMOS gate normally has a negative potential with respect to the channel-substrate (the transistor bulk).

Common analog design techniques to improve the analog radiation tolerance of a design are to prefer PMOS transistors over NMOS transistors and to avoid potential leakage current paths. The latter can be done by introducing guard rings and by using NMOS transistors with "round" or "encapsulated" gates (one electrode is encapsulated by the poly-silicon gate). Encapsulated NMOS transistors allow to cut direct DC paths between two n-doped contacts having different potentials, some p-doped bulk in-between, and field oxide above (a parasitic NMOS structure).

In the CBM experiment the total ionization dose the TRD electronics close to the beam line will have to face (over the whole CBM live time) is expected to stay below 100 krad. Fortunately, the used UMC 180 nm technology was measured to have very good intrinsic self-annealing properties. Investigations showed for instance a total self-annealing of all total ionizing dose effects in a dedicated test ASIC (after some weeks, at room temperatures) that was shortly exposed to a total dose of 2.4 Mrad [47]. Even though there might exist certain low rate effects, which have not been measured so far.

However, the probably very good intrinsic radiation tolerance of the process is actually one of the main reasons why it was chosen in the first place. As a consequence, the radiation tolerance of the analog part will probably or hopefully be no big issue for the SPADIC.

CHAPTER 6

The Digital Part

This chapter summarizes most details of the digital processing concept that was realized in the latest version SPADIC 1.0 (unless otherwise noted). Similar to the previous chapter, the focus is set on a relatively high level of technical details – for a conceptual overview go back to section 4.3.2 where an abstract summary of the overall concept is given.

6.1. ADC Interface

The first synthesized block behind the analog parts in each channel is the ADC interface. It has two tasks: on the one hand it converts the redundant raw data stream continuously coming out of the ADC into 2's complement (in agreement with equation 5.26) and on the other hand it delivers synchronous control signals, which the ADC effectively uses as its system clock[1].

As previously discussed in section 5.2.1.3, the evaluation logic basically consists of an adder in combination with a proper delay matrix. Whereas the adder is rather small, the delay matrix in the present case requires about 100 flip flops (equation 5.40).

Due to internal reasons of the ADC implementation the details of the internal raw data interface are complicated. This comes from the fact that the partial signals emerging from the ADC are time-multiplexed, inverted and delayed in pairs, and, moreover, must be properly converted before the subsequent adder can handle them correctly. For that reason the complete ADC design including the ADC interface is an example application of a mixed-signal simulation (see also section 6.7) which consists of both a complex analog and a complicated digital part. And indeed all ADCs of all SPADIC versions have been verified that way.

[1]The logic sequence controlling all internal ADC switches in order to generate the algorithmic rhythm, also uses digital logic gates. But the whole sequence logic of the ADC was designed manually and completely laid out by hand. Hence to avoid setup- and hold-time problems, it is clocked by non-overlapping sync signals instead of a "normal" clock signal.

A very practical issue worth noting here is the placement of the pins of the non-overlapping sync (ADC clock) signals: In order to guarantee for a correct passage of the signals from the manually designed sequence logic to the synthesized digital evaluation block, the timing constraints of the synthesis between digital clock and ADC sync signals must be carefully set. In order to make sure that the distance from the constrained sync signal pins to the corresponding ADCs is kept short, the global sync signals have each been defined 4 times (by naming them in 4 different ways) and the pin positions have been properly distributed. That way each of the 4 signal groups could be placed in the middle of a block of 8 analog channels and thus no path longer than roughly 4 channel heights (480 µm) from a constrained sync pin to an ADC exist on the chip.

However, if one takes the ADC interface as a black box, it simply delivers a continuous 9 bit data stream in 2's complement with a word (sampling) frequency of 25 MS/s.

6.2. Digital Filter

In each SPADIC channel a digital filter is placed between ADC and hit detection logic. Its main purposes are to cancel certain characteristics of the connected detector and to linearly adapt the CSA pulse shapes to the requirements of the hit detection logic. The most important examples are ion-tail cancellation (cutting out the slow signal tails caused by drifting ions in MWPCs, see 2.2.2.5), baseline stabilization (the removal of low frequency oscillations and slow baseline drifts), and pulse-scaling or baseline-shifting (e.g. the pulses of the negative CSA must be inverted before they reach the hit detector).

In general, the two most obvious options to realize a linear and flexible data preprocessing unit were either a solution based on a microcontroller or a digital filter. Although the first option would certainly have provided the highest flexibility, the relatively simple and short list of tasks in the present case never really justified the high costs and the complexity of that option. Hence the only question was whether to use an FIR (finite impulse response) or an IIR (infinite impulse response) filter. Even though both types are feasible for the task in principle, the IIR filter has been chosen without long hesitation, because it acts similarly to an analog filter (the handling is more intuitive, FIR filter have no analog pendants in general), it is much more compact (FIR filters have no feedback and require a lot more stages, for instance if low-frequency components shall be processed), it adds much less delay into the data stream, and its parameter settings are much easier to find and to adjust (FIR filter settings require very complex calculations). However, the biggest drawbacks of IIR filters are certain non-linear effects (e.g. non-linear phase shifts) and potential instabilities (a very good introduction can be found here: [54]).

The IIR filter implemented in each channel of SPADIC 1.0 is predominantly the result of a diploma thesis that was completed in 2011 [42]. Because a comprehensive summary of the theoretical background and the implementation details are already given there, subsequently only the basic ideas and most important issues of the implemented IIR filter are summarized.

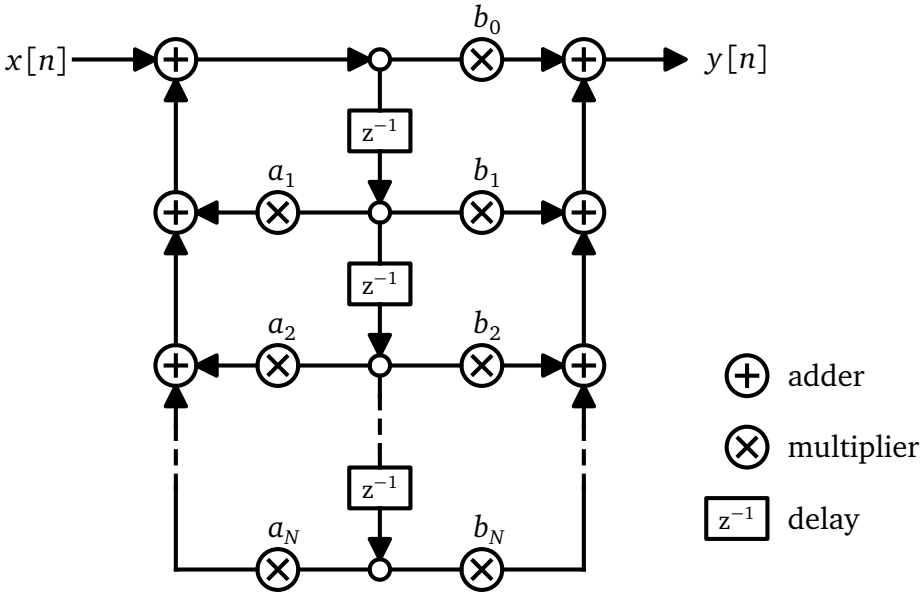


Figure 6.1.: Data flow graph of an IIR filter in direct form II of order N.

6.2.1. Structure of IIR Filters

The data flow graph of an IIR filter in direct form II of order N is shown in Fig. 6.1. The filter stepwise converts the discrete input sequence $x[n]$ into an output sequence $y[n]$. The delay elements z^{-1} define the end of a discrete processing or iteration step, or in other words each z^{-1} delays the input word w_{in} by one cycle (very similar to a flip-flop, formally $w_{\text{out}}[n] = z^{-1} \cdot w_{\text{in}}[n] = w_{\text{in}}[n-1]$). Moreover, the multipliers scale the input value by a constant factor c ($w_{\text{out}}[n] = c \cdot w_{\text{in}}[n]$) and the adder simply adds two input values ($w_{\text{out}}[n] = w^1_{\text{in}}[n] + w^2_{\text{in}}[n]$). Although the direct form II is one of numberless possible IIR filter structures, it is a very important one. The reason simply is that it corresponds to a very practically relevant system function (z-transform, for a derivation see [54]):

$$H(z) = \frac{y[n]}{x[n]} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}. \tag{6.1}$$

This rational function has up to N zeros and N poles (or less if the respective coefficients are set to zero) and is generally able to "carry" complex conjugated pairs (if $N \geq 2$). The formal flexibility of the transfer function further increases, if M IIR filter stages in direct form II are connected in series:

$$H(z) = H_1 \cdot H_2 \cdot \ldots \cdot H_M = \prod_{l=1}^{M} \frac{\sum_{k=0}^{N_l} b_{kl} z^{-k}}{1 - \sum_{k=1}^{N_l} a_{kl} z^{-k}}, \tag{6.2}$$

which obviously leads to an ambiguous representation mixed of parts in factorization and parts in partial fraction decomposition, but visualizes exemplarily how the order of the l-th stage N_l and the number of stages M affects formally the rational system function. As a consequence, the effective number of poles (and zeros) only depends on the sum $\sum_{l=1}^{M} N_l$,

but not on the segmentation of M and N_l. Hence the same system function can be realized either with a large number of low-order stages or a small number of high-order stages[1]. But one should not forget that complex conjugated pairs require $N_l \geq 2$, since the terms of different stages can not be combined such, that they build complex conjugated pairs (see the lemma in appendix A.2). Consequently, no formal flexibility is lost, if one sets $N_l = 2 \; \forall \; l$. In that case the last equation can be rewritten much more clearly as

$$H(z) = \prod_{l=1}^{M} b_{0l} \frac{(1 - x_{0l} z^{-1})(1 - x_{1l} z^{-1})}{(1 - y_{0l} z^{-1})(1 - y_{1l} z^{-1})}, \tag{6.3}$$

with poles y_{0l}, y_{1l} and zeros x_{0l}, x_{1l}.

Nevertheless, the IIR filter implemented in SPADIC 1.0 completely does without complex pole pairs ($N_l = 1 \; \forall \; l$), due to a simple reason: as shown in [42] and also discussed briefly further below, the principle of ion-tail cancellation (the main application here) does not require complex conjugated pairs – and so do the other tasks listed earlier. The relinquishment of complex conjugated pairs on the one hand helps to shorten the critical data path in a single filter stage and on the other hand reduces the analytical complexity and thus makes the parameter search and handling easier. Removing the space for complex conjugated pairs the latter equation simplifies to

$$H(z) = \prod_{l=1}^{M} \frac{b_{0l} + b_{1l} z^{-1}}{1 - a_{1l} z^{-1}}, \tag{6.4}$$

with poles a_{1l} and zeros $-\frac{b_{1l}}{b_{0l}}$ – which can be directly set via the initial coefficients of the data flow graph.

6.2.2. The Analog Pendant to an IIR Filter

Because a function in (unilateral) z-transform is basically the Laplace-transform (s-transform) of an ideally sampled signal, an ideal transition can be made with the so-called bilinear transform $z \rightarrow e^{sT}$ (with $1/T$ the sampling frequency). Mathematically, the bilinear transform maps the unit cycle of the z-plane ($e^{i\omega}$ for ω from 0 to 2π) to the imaginary axis ($i\omega$ for ω from 0 to ∞) of the s-plane. But for practical reasons it is mostly beneficial to use the first-order approximation $z \rightarrow \frac{1+sT/2}{1-sT/2}$ instead. That way equation 6.4 becomes

$$H(s) = \prod_{l=1}^{M} \frac{2(b_{0l} + b_{1l}) + sT(b_{0l} - b_{1l})}{2(1 - a_{1l}) + sT(1 + a_{1l})}. \tag{6.5}$$

This analog interpretation of the filter is particularly helpful, if one wants to figure out how the sampling period T and the time constants τ correlate. For instance, to make the IIR filter behave like a simple first order high-pass, one can set $M = 1$. Then in order to get

[1] It is important to note that even though this is true mathematically, real implementations suffer from limited resolutions and ranges and hence actually do depend on the respective filter structure. That is briefly discussed further below, but to simplify the matter ignored here.

rid of the number in the nominator the equation $b_{01} = -b_{b11}$ must be fulfilled. And since $\tau = 2T\,b_{01}$, thus all b_{k1} are defined (with similar considerations one gets $\tau = \frac{T}{2}\frac{1+a_{11}}{1-a_{11}}$).

6.2.3. Principle of Ion-Tail Cancellation

As calculated in [42] the ion drift back from the amplification region of a MWPC induces a slowly decreasing tail $i(t) \propto 1/t$, frequently called the ion-tail. In high-rate applications such as CBM it is very likely that the rest signal caused by a previous hit is still significantly high, when a new hit occurs. Therefore pile-up (the overlapping of two or more pulses) becomes a common issue. In a linear system (like SPADIC) pile-up is theoretically no problem, since one can always mathematically completely distinguish the pulses from each other as long as they have been completely recorded. But in practice, pile-up might lead to internal overflows, can bring the internal hit detection logic seriously in trouble, and in general shifting the processing complexity to later stages is a lazy approach. In some regard the purpose of the IIR filter implemented in SPADIC 1.0 is redundant, since SPADIC 1.0 is able to record complete pulses and has a comparator mode that does not depend on a fixed comparator threshold. Nevertheless, on the one hand adding flexibility is one of the main maxims of the SPADIC concept and on the other hand the ion-tail cancellation effectively helps to reduce the required number of samples per recorded pulse and the subsequent processing complexity. And moreover, the ion-tail cancellation filter effectively acts as a digital baseline stabilizer, which simplifies the subsequent interpretation of the data and particularly makes the extraction of certain pulse characteristics easier.

The principle of an ion-tail filter is rather simple: first, one assumes that the initial current signal of the detector $i(t) \propto 1/t$ can be approximated with a finite sum of exponential components:

$$i(t) \approx \sum_{i=1}^{L} w_i \mathrm{e}^{-t/\tau_i}, \tag{6.6}$$

with different time constants τ_i and weightings w_i. And second, the IIR filter is configured such, that it removes a certain number of slower components $w_i \mathrm{e}^{-t/\tau_i}$. That way the tail of the generated output pulse decreases much faster.

Before a formal discussion can be started it is important to emphasize that the actual characteristics of the shaper, which acts in-between the initial ion-signal and the IIR filter, explicitly plays no role in this context. This comes from the fact that the positions of CSA and IIR filter (theoretically) can be swapped without effecting the transfer function of the system or the resulting signal shapes – which is a fundamental principle of linear time-invariant systems in general. Consequently the existence of the CSA in the readout chain is completely ignored subsequently.

It will now be briefly shown, that each first-order IIR filter stage that corresponds to a single factor in equation 6.4 can be used to cancel exactly one component of the approximated ion-tail (equation 6.6). The argumentation is based on the detailed discussion given in [42], although a less abstract formalism is used.

The z-transform of a sampled ion-tail component $w_i \mathrm{e}^{-nT/\tau_i}$ (with sampling period T) is

$$I(z) = \mathcal{Z}\{w_i e^{-nT/\tau_i}\} = \mathcal{Z}\{w_i p_i^n\} = w_i \frac{z}{z - p_i}, \tag{6.7}$$

with constant $p_i = e^{-T/\tau_i}$. Without loss of generality a single filter component (equation 6.4) can formally be further simplified by ignoring the constant factor b_{0l} (which is only an unimportant scaling factor) and renaming the variables ($b_l = b_{1l}/b_{0l}, a_l = a_{1l}$):

$$H(z) = \frac{z + b_l}{z - a_l}. \tag{6.8}$$

In that case the digital result of one component passing a single filter stage can be calculated as

$$\begin{aligned} i(n) &= \mathcal{Z}^{-1}\left\{ w_i \frac{z}{z - p_i} \frac{z + b_l}{z - a_l} \right\} \\ &= w_i \left(\frac{a_l + b_l}{a_l - p_i} a_l^n + \frac{p_i + b_l}{p_i - a_l} p_i^n \right) \\ &= w_i \left(\left(1 - \frac{p_i + b_l}{p_i - a_l} \right) a_l^n + \frac{p_i + b_l}{p_i - a_l} p_i^n \right). \end{aligned} \tag{6.9}$$

That important result clearly tells that the filter effectively scales the initial component p_i^n by $\frac{p_i + b_l}{p_i - a_l}$, but at the same time introduces a new exponential component $a_l^n = e^{-nT/\tau_l}$ (with the corresponding time constant $\tau_l = -T/ln(a_l)$).

Finding a feasible set of filter coefficients which effectively removes undesired exponential components is complicated in general, because several possible strategies exist. The methodology used in [42] is simple and clever though: the first step is to choose the b_l such that the l-th filter stage completely cancels the i-th component, hence $b_l = -p_l \ \forall l$. That way each undesired initial component p_i is canceled eventually. Then proper values of a_l can be found by forcing the sum of all newly introduced components a_l^n in stage l to zero (or in other words by restraining their occurrence in the first place):

$$\begin{aligned} &\sum_{i=1}^{L} w_i \frac{a_l + b_l}{a_l - p_i} a_l^n \stackrel{!}{=} 0 \\ &\Rightarrow \sum_{i=1}^{L} \frac{w_i}{a_l - p_i} \stackrel{!}{=} 0 \quad (a_l,\ b_l, \text{ and } a_l^n \text{ are independent of } i). \end{aligned} \tag{6.10}$$

The latter equation has exactly $L - 1$ solutions for a_l and is equally valid for each of the M stages.

But the discussion is not yet complete: As shown in [42], it is beneficial not to use the found a_l at random, but to sort them (and also the p_l) like $p_l > a_l > p_{l+1}$. That way, stage l does three things at the same time: first it further suppresses exponential components with time constants larger than $\tau_- = -T/ln(p_l)$ (slower components), second it exactly cancels p_l, and third it amplifies components with time constants smaller than $\tau_+ = -T/ln(a_l)$

(faster components). Only the range τ_- to τ_+ is problematic in theory, since components within that range are inverted, what can potentially lead to undershoots[1].

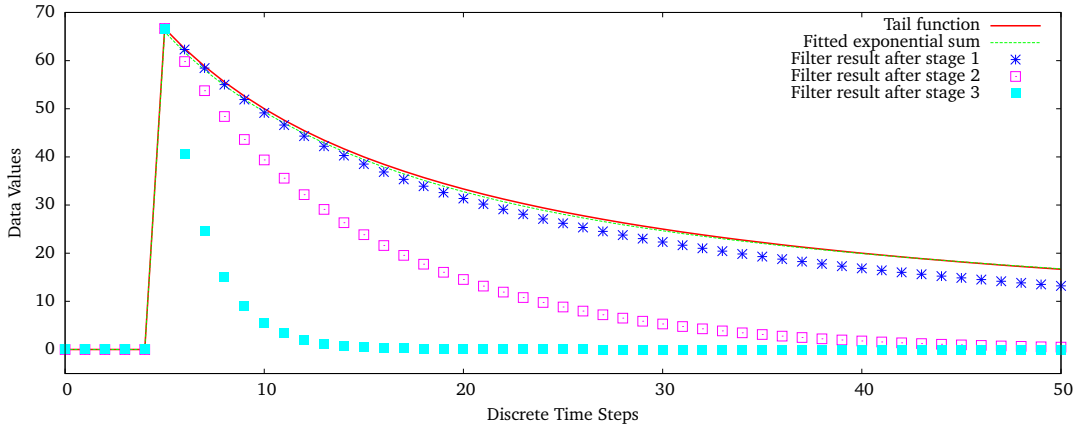


Figure 6.2.: Example: The initial ion-tail function, a proper fit, and the filter results after 1, 2 and 3 stages.

As a simple example it is assumed that the function $f = \frac{1000}{t+10}$ represents a typical ion-tail (plotted in Fig. 6.2, red curve). In order to calculate the ion-tail filter result after an IIR filter with 3 first-order stages, a fit with 4 exponential components had to be made first (green curve). In the present case the fit parameters $w_1 = 7, w_2 = 26, w_3 = 31, w_4 = 2, \tau_1 = 250\,\text{ns}, \tau_2 = 50\,\text{ns}, \tau_3 = 10\,\text{ns}$ and $\tau_4 = 2\,\text{ns}$ were chosen. That values and the method just described then directly led to the filter coefficients $a_1 = 0.993, a_2 = 0.943, a_3 = 0.617, b_1 = -0.996, b_2 = -0.980$ and $b_3 = -0.905$. The corresponding filter results after 1 (blue), 2 (pink), and 3 (light blue) stages are also plotted in Fig. 6.2. It can be clearly seen how the slower ion-tail components are canceled step by step until only the fastest tail component remains.

6.2.4. Internal Resolution

The filter structure yet analyzed was discrete-time but not discrete-value. But of course, the internal resolution of real filters is limited. That alone would be no issue, since the incoming digital data values already have a limited resolution that the filter in principle can easily adopt. But if one considers that adders and multipliers usually produce output words being wider than the initial input words, it becomes clear that the intermediate results must be properly cut or rounded eventually (especially if the filter has a feedback). Therefore – very similar to analog circuits that suffer from analog noise sources – a digital filter typically introduces quantization noise sources that limit the achievable accuracy. The main difference to analog noise sources is the fact though, that the digital quantization error is time-invariant, perfectly predictable, and more easily manageable.

In general, to minimize the quantization noise of an IIR filter one can adjust the structure of the data flow graph (which is one important reason why various filter structures for the

[1] Practically that effect turned out to be rather small – at least in the tested cases.

same transfer function can be found in most relevant text books) or increase the internal resolution. In the present case, due to the very simple filter structure of the SPADIC IIR filter, only the internal resolution was adjusted properly – or, to be more specific, both the resolution of the data values passing the filter and the resolution of the filter coefficients were investigated independently.

The approach that has been taken for the SPADIC 1.0 IIR filter was to set the internal resolution such that the emerging uncertainty due to the internal quantization stayed significantly below the already existing uncertainty due to CSA and ADC noise. With the help of the IIR filter emulator, which has been developed in [42], the internal resolution of all intermediate nodes was finally set to 16 bit, whereas the filter coefficients were fixed to 6 bit. Moreover, the filter was designed to finally cut the output values and thus to make their resolution match the initial input resolution of 9 bit (set by the ADC). The latter makes the implemented IIR filter a black box that can be easily put into any 9 bit data stream in 2's complement.

6.2.5. Realization of the Multipliers

The realization of a filter structure as just discussed obviously requires three types of building blocks: adders, multipliers, and delays. Because delays are easy to build (with flip-flops) and the adder is actually a building block that is also required within a multiplier, the task of developing an IIR filter with standard cell logic basically comes down to the development of proper multipliers.

Most briefly summarized, a digital multiplier calculating the product of an m-bit and an n-bit value can be roughly sub-divided into three parts: first, a partial product generation matrix builds $m \cdot n$ partial products (each bit of the m-bit word times each bit of the m-bit word – logical AND), second, the partial products are combined in a partial product reduction tree (PPRT) until only two words remain, and third, the two intermediate words are added in a final adder stage. The most design considerations deal with the optimization of the partial reduction tree, where the most operational steps have to be performed. For instance a very basic but commonly used method is to build so-called Wallace-trees [71], which parallelize the PPRT as much as possible ($\mathcal{O}(\log m)$ or $\mathcal{O}(\log n)$ steps are required), or so-called Dadda-trees [30], which in contrast to Wallace-trees require only a minimum number of logic cells (mainly half-adders or full-adders), but usually have a higher number of reduction steps. Nevertheless, also design techniques to reduce the number of partial products in the first place are in use (e.g. Booth-encoding [21]). But in fact those techniques only tend to shift the complexity from the partial product matrix into the PPRT without really changing the total design effort [69].

Two approaches were initially taken in [42] when designing the SPADIC IIR filter: The first was to manually build the multiplier (and also the adder), which eventually led to the usage of the so-called three dimensional minimization algorithm (TDM) [53] that generates a PPRT very similar to a Dadda-tree. The output of the TDM – basically a netlist of logic gates (mainly half- and full-adders) – was mapped then to the home-made standard cell library UCL (see section 5.4.2), to which (exactly for that purpose) a full-adder and half-adder have been added before. The second approach, that was taken after all important

details of multiplier design techniques have been understood properly, was fairly simple: The formal command $c = a \cdot b$ was just directly and properly translated into HDL code and the handling completely left to the synthesis tool (First Encounter from Cadence, also using the UCL standard cell library). More specific, the used Verilog code was (simplified):

```
1  //signal definitions
2  input wire signed [WIDTH-1:0] a, b;
3  output wire signed [WIDTH-1:0] c;
4  output wire signed [(2*WIDTH)-1:0] temp;
5
6  //multiplication and quantization
7  assign temp = ((a * b) >>> (WIDTH-1));
8
9  //select the proper bits
10 assign c = temp[WIDTH-1:0];
```

Note that it is necessary to interpret at least one input value not as an integer but as a decimal number (e.g. b as a number between $\frac{-1}{WIDTH-1}$ to $\frac{1}{WIDTH-1}$) to make sure that the quantization of the output signal effectively cuts decimal places instead of dividing the final result.

Then, an evaluation of both approaches was made by comparing the designs after synthesis. The observation was that both approaches were equally fast (without any time-constraints set), but the manually built TDC design required a slightly larger number of logic gates and hence more chip area (e.g. a multiplication 16x16, Verilog/TDC: transition time 5.6/5.3 ns, area 19 500/25 300 μm^2). Accordingly the challenge between home-made multiplier and commercial tool was – not really surprisingly – won by the synthesis tool. And because not only the result of the commercial tool was slightly better, but also the flexibility of simple HDL code is higher and less error-prone in general, the IIR filter of SPADIC 1.0 was finally realized taking the second approach – and thus only with a handful of Verilog files.

The nominal clock frequency of the IIR filter was matched to the 25 MS/s of the ADC (hence set to 25 MHz). Luckily, the synthesis even without any set time-constraints produced a filter stage that was already fast enough (for instance, the single product of a 6 bit coefficient and a 16 bit value takes less than 5 ns). Therefore no manual optimizations needed to be done.

6.2.6. Other Design Aspects

The number of filter stages was chosen to be 4 (or $3\frac{1}{2}$, compare [42]). Actually, the value was set rather arbitrarily, since on the one hand no specification how well the ion-tail cancellation must perform was given, and on the other hand the real shape of the ion-pulses was not exactly known. Accordingly, the IIR filter implemented in SPADIC 1.0 is primary a prove of principle and a test design. It will probably be bypassed for the first detector measurements, but might become very relevant though, as soon as the actual shape of the ion-pulses have been measured more precisely. However, considering the required

moderate amplitude resolution of about 8 bit for the TRD, the 4 available 1st order IIR stages will most certainly be sufficient in the end.

A very important consideration in general is whether the filter coefficients should be configurable for each channel individually or only globally[1]. In the present implementation, a complete set of coefficients has about 60 bit and so far can only be set globally. That solution was chosen simply to reduce the total number of registers. But of course an independent parameter set in each channel would allow to adapt the respective channels more flexibly to local variations caused by the analog channels or the connected detector, and therefore could be very beneficial in principle. This issue should be risen again as soon as the next SPADIC iteration is going to be built, and then the decision should be based on statistical and experimental experience gathered in the meantime.

6.3. Hit Detector and Message Builder

As just shown, the output of the ADCs or the IIR filters are respectively continuous streams of data values in 2's complement. If the SPADIC was an externally triggered system, the readout strategy would most certainly be to cut out a fixed sequence of the data stream whenever the global trigger signal arrives, or to dump the stream otherwise. In contrast, the free-running approach intended for the SPADIC chip makes it necessary to actively and continuously search the data stream for interesting information, and, if required, to extract and sent it out of the chip. In the present case, interesting information comes in the form of short pulses rising far above (or below) an otherwise stable baseline. An important detail of the given application is the Poisson statistics of the particle hits, which, since the system has no death-times (the SPADIC CSAs have a continuous reset), directly leads to a Poisson distributed occurrence of hit pulses. Hence in particular, on the one hand the hit detection and extraction logic must be able to handle even pulses overlapping each other (multi-hits) and on the other hand must provide enough internal buffers to avoid the loss of data as much as possible. But, however, also the ever possible event of a full buffer must be handled properly.

The subsequent section summarizes the concept of the hit detection and message builder logic implemented in SPADIC 1.0, that was designed to conveniently handle the situation just described.

6.3.1. Overall Block Diagram

A simplified block diagram of the hit detection and message builder logic is shown in Fig. 6.3. To gain several clock cycles in-between two incoming data values (25 MS/s), most logic parts are clocked with 125 MHz (as indicated in the figure, dark gray blocks). However, the data input and message output interface both run at the initial clock frequency of 25 MHz (light gray blocks). The synchronous clock crossing is realized with a shift register at the input and a FIFO at the output, and is globally coordinated with a dedicated sync signal.

[1] Despite the fact that also in the analog part individual adjustments are possible.

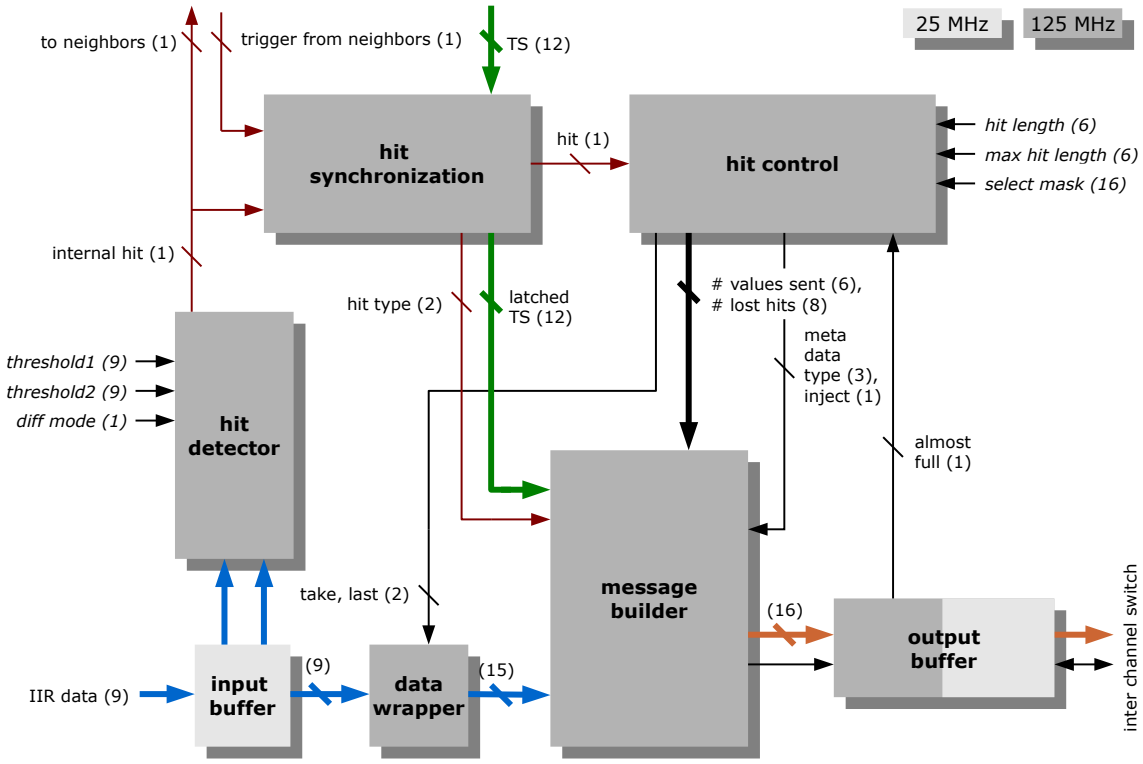


Figure 6.3.: Simplified block diagram of the hit detection and message builder logic. Blue: data flow of IIR/ADC values, red: trigger/hit signals, green: global time-stamp, orange: generated message words.

The overall functionality can be briefly described as follows: The 9 bit data stream from the IIR filter (blue arcs) is temporarily stored in a shift register (*input buffer*) and simultaneously monitored by a comparator (*hit detector*). The output of the shift register is connected to a *data wrapper* that packs the (selected) 9 bit values into a 15 bit format, which is required later. If the comparator logic detects the threshold-crossing of a value stored in the input buffer, an internal hit signal is generated and optionally also sent to selected neighbor channels. The re-synchronized internal hit signal (*hit synchronization*) then eventually triggers the generation of a new message. The generation of a new message is handled by the main FSM (*hit control*), which coordinates the proper multiplexing of raw data values (wrapped IIR/ADC values) and meta data words (time-stamp, status bits, etc.). Each of the created messages consists of several 16 bit words each representing either some raw or some meta data (orange arcs). Finally, after a hit has been detected and during the corresponding message is being generated, the whole message (word by word) is stored in the *output buffer*, where it is eventually read out and further arbitrated later.

6.3.2. Hit Detection and (Neighbor-)Trigger Concept

The hit detector provides two thresholds (in 2's complement) and two modes. In "normal mode" two data values ($x[n]$ and $x[n-1]$) are evaluated at once. If both values are

greater than their respective threshold, the internal hit signal is produced. In "differential mode" the comparator works similarly, but is fed with the differentiated input values instead ($x[n] - x[n-1]$ and $x[n-1] - x[n-2]$). The differential mode is independent of absolute values or the baseline and therefore allows to detect double- or multi-hits much more efficiently. Moreover, a low-frequency stabilization of the baseline becomes unnecessary in principle (even though that feature is available due to the IIR filter). In the subsequent section, the threshold is called "virtual", if the comparator is assumed to be in differential mode.

The usage of two instead of one thresholds allows both in normal and differential mode to set more conservative trigger conditions. For instance, the comparator can be configured to ignore single runaway values[1] or to demand a minimum slope. However, one threshold can of course be disabled completely, simply by setting the other threshold to the most negative digital value.

After a pulse has been recognized by the hit detector, the generated internal hit signal is sent to all (selected) neighbor channels as well as directly to the internal hit synchronization logic. That way the neighbor readout scheme introduced earlier is realized (see section 4.2.2). The idea of the present implementation (SPADIC 1.0) is that each channel listens to hit signals of all selected/configured channels – and also to its own internal hit signal, of course. Due to the availability of a large neighbor matrix that is configurable at will, nearly every neighbor relationship can be programmed in SPADIC 1.0. For example, a typical configuration would be to trigger some of the direct neighbors (e.g. channel 6 triggers the channels 4,5,7, and 8) or, if the pad layout is interleaved, to trigger only even (or odd) channels (e.g. channel 6 triggers the channels 2,4,8, and 10) – or another example would be a complete readout scheme (e.g. channel 6 triggers channels 0–5 and 7–15).

The configuration matrix in SPADIC 1.0 was realized for a block of only 16 channels. Hence for the 32 channels two matrices (for the channels 0–15 and 16–31) were used, each requiring a huge configuration register of 484 bit and more than $16 \cdot 16$ AND and $16 \cdot 16$ OR gates (plus buffers).

To moreover allow for an exchange of hit signals in-between two channel blocks, an additional mechanism was implemented that provides 3 bidirectional hit connections to the "previous" matrix and 3 to the "next" matrix. That way selected hit signals can be transported in-between the two internal channel groups (0–15 and 16–31), but also in-between two neighboring ASICs. To be more specific, to physically propagate the hit signals in-between chips, on the upper and on the lower chip edge three input and three output hit signal connections (LVDS) were placed and routed (24 pads, also used for various other purposes). And of course, the 12 inter-chip signals are also included in the neighbor matrix and can be flexibly exploited. For instance, one can configure channel 0 of chip B to be externally triggered by the channels 30 and 31 of chip A, while at the same time the channels 0 and 1 of chip B trigger channel 31 of chip A. In the next SPADIC iteration, the neighbor network could be significantly simplified, if the actually required neighbor relationships for the CBM-TRD were exactly known and defined.

Because the signal propagation through the neighbor matrix can take some time and especially because the delay between two ASICs cannot be predicted in general, the internal

[1] Which should not occur under normal conditions though.

and the external (coming from the neighbors) hit signals have to be re-synchronized within each channel. That is exactly the reason why the hit synchronization (Fig. 6.3) has been introduced. The implementation of the hit synchronization is rather simple. It basically waits for all hit signals (generated with the fast 125 MHz clock) that occur within one slow cycle (25 MHz) and generates a synchronous trigger signal going to the main FSM (hit control). Moreover, the hit synchronization latches the extracted hit type (internal, external, or both), which then is stored as meta data in the respective hit message.

6.3.3. (Multi-)Hit Handling, Selection Mask, and Time-Stamp

Before the implementation of the message building mechanism of SPADIC 1.0 is further described, it is self-evident to summarize how normal and multi-hit pulses should be handled in theory.

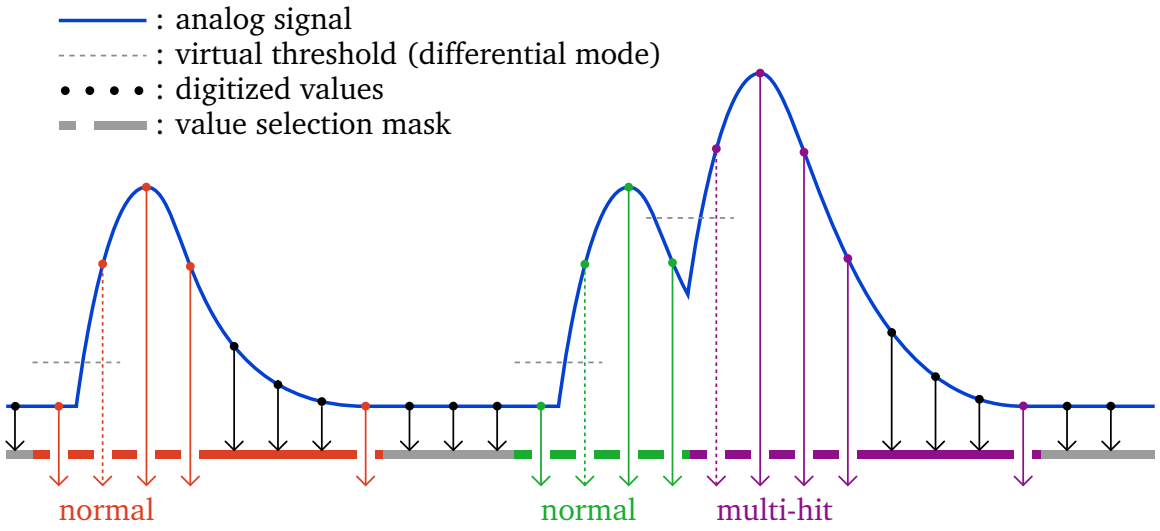


Figure 6.4.: The figure shows how normal and multi-hit pulses are handled by the selection logic. A trigger and hence a new message is produced, each time the (virtual) threshold is crossed or a neighbor sends an external hit signal (not shown). If a multi-hit is detected, the hit selection mask is reset.

As an example, a typical pulse sequence is sketched in Fig. 6.4. The sequence contains a single pulse that is followed by a double pulse. As indicated in the figure, a normal hit message is generated as soon as the single pulse is recognized by the hit detector. The normal hit message then contains a sequence of data values, which previously has been configured via a selection mask. In the example, the selection mask was set to *1111_0001* (the actual selection mask of SPADIC 1.0 has 32 entries). In this case all of the 8 possible values except for the 5th, the 6th, and the 7th are stored in the hit message.

After the single pulse, the double pulse at first also triggers the generation of a new normal hit message. But due to the second overlapping pulse (that can be recognized because of the virtual threshold) the generation of the normal hit message is interrupted eventually. Then, the main FSM handles the finalization of the normal message (the abort due to a

multi-hit is stored as meta data in the message) and immediately starts the generation of a new multi-hit message (which equals a normal message but for a multi-hit flag). At the same time the selection mask is reset and thus again only the most interesting values of the overlapping pulse are stored in the multi-hit message.

The exact same selection and generation mechanism is started, if the pulse is not recognized directly by the hit detector, but if instead a neighbor channel sends a trigger signal. Hence for the internal logic it makes absolutely no difference whether the hit signal was produced internally or externally – except for the stored hit type in the meta data. The generation of the previous multi-hit message for instance could have been caused by some neighbor channel as well, even if the virtual threshold was not crossed internally. In that case the same multi-hit message would have been produced, but containing the hit-type information “extern” instead of “intern”.

Besides the proper selection of data values, the recording of a corresponding time-stamp is a very fundamental task of the hit logic. In order to assign the recorded data values to the arrival time of the pulse, a unique time-stamp must be added to each generated message. Therefore a global (and previously synchronized) counter is latched each time a new hit is recognized and the latched value is added as meta data in front of the hit message. In the present case, the time-stamp has a word-clock granularity (25 MHz), which corresponds to a coarse time-resolution of 40 ns. But indeed, the exact arrival time of the pulse can be interpolated much more accurately later (see section 5.3.1). Figure 6.4 also indicates how the recorded data values are correlated to the latched time-stamp (dashed vertical arrows).

6.3.4. Lost Hits

As described before, the generated messages of each channel are stored in an output buffer until they are read out again by the subsequent instance. The output buffers in SPADIC 1.0 are limited to 64 message words (a 16 bit), which – in dependency on the respective configuration of the selection mask – suffices for about 10 normal hit messages. Hence, considering the Poisson statistics of the incoming pulses, it is very likely that the output buffer is still full when a new pulse arrives or – more complicated – becomes full while a message is being created. Consequently, the hit logic must be feasible to handle several abort events properly.

The idea implemented in SPADIC 1.0 to handle pulses arriving while the buffer is already full, is to increment a lost-hits counter whenever a new pulse occurs that could not be recorded. Then, as soon as the buffer again provides enough free space, a special 3-word buffer-overflow message containing the number of meanwhile lost hits is generated – and the lost-hits counter is reset. That way, respective hits are not completely lost, but at least counted, which is an essential information for a reliable event analysis.

If, in contrast, the message generation must be interrupted because of the buffer becoming full, a proper abort word telling the exact reason for the interruption (there are various cases) is attached to the end of the message. In doing so, it is of course always guaranteed by the logic that even an aborted message at least contains the most fundamental information (channel/chip ID and time-stamp) and, moreover, that there is still enough space for the abort information itself left in the output buffer.

Besides the event of a full output buffer, the hit logic must also handle a full ordering FIFO (which has not yet been introduced, compare section 6.4.1). But because the implemented mechanism is very similar to that of the event of a full output buffer, details are skipped here.

6.3.5. Meta Data, Message Types, and Message Format

As partially mentioned before, besides the raw data values each message contains a bunch of necessary or at least beneficial meta data. The most important meta values are the time-stamp, the hit type, the channel/chip ID, the stop type (normal end of message, interrupted message, etc.), and the number of stored raw data values.

Because of the various abort situations, the configurable selection mask, the different possible message types, and the multi-hit logic, the size (number of 16 bit words) of the single messages usually heavily fluctuates. Unfortunately, that fact does not only require additional flexibility of most subsequent stages on and off the chip (for instance the arbiter or the buffers), but also can complicate the firmware and software developments.

To make the single messages robust and the stream handling convenient, the SPADIC message words are nearly context-free, meaning that each 16 bit word has a unique preamble and that thus every message word can be interpreted independently (except for the so-called continuation word, see the table further below). In detail, there are various reasons for that definition, for instance:

1. Because of the various possible compositions of raw data and meta data words, the unique preamble effectively helps to keep the interpretation and the handling of the message streams flexible and simple (e.g. if one wants to sort the messages by their channel-IDs, count all internal hits of a certain channel, monitor time-stamps, or search for error markers).

2. The generation of messages gets much easier in the first place, because no context must be kept in mind. Moreover, a message can be interrupted anytime by simply adding a proper stop word. And, very important, especially since the beginning and the end of a message within a message stream can be found simply by monitoring the preambles, arbitration mechanisms do not need to know the actual message length.

3. Messages can be very flexibly recomposed in any subsequent processing stage without the requirement of redefining the message structure (e.g. messages of a certain channel can be combined, multi-hit message can be built, or special markers inserted).

4. The single messages as well as the message streams globally become very insensitive to single bit-flips (e.g. due to single event radiation effects). That means, that even if a preamble is corrupted, the respective message or at least the message stream can always be re-synchronized simply by looking at some of the previous and next message words. That way a simple “context monitor” could flexibly search for local inconsistencies and either fix or mark them. And even if complete data words are lost or doubled, the redundant message word definition always allows to simply recover the context of the SPADIC message stream.

The table below briefly lists all defined 16 bit message words that can occur within a message generated in SPADIC 1.0.

Preamble	Payload	Description
1000	.... gggg gggg cccc	Start of message
1001	.... tttt tttt tttt	Time-stamp
1010	.... dddd dddd dddd	Begin of raw data
1011	.... nnnn nnhh -sss	End of message
1100	.... --- bbbb bbbb	Buffer overflow
1101	.... eeee eeee eeee	Epoch marker
1110	.... --- --- ---	Reserved/unused
1111	.... iiii aaaa aaaa	General information
0	.xxx xxxx xxxx xxxx	Continuation

Table 6.1.: Message word definition used in SPADIC 1.0. g: group ID, c: channel ID, t: time-stamp, d: raw data, n: number of samples, h: hit type, s: stop type, b: number of lost hits, e: epoch counter, x: content depends on previous word (e.g. additional raw data), i: info type, a: content depends on info type.

For a more comprehensive description of the message format, the different types of meta data, and the possible message compositions see the latest documentation on the SPADIC website [11] – a complete explanation would lead too far here. Nevertheless, at least an example of a normal hit message is subsequently given:

Data words	Interpreted content
1000 0000 0111 0010	Start of message - group: 7, channel: 2
1001 0011 0010 1010	Time-stamp - counter value: 810
1010 1111 1110 1000	Begin of raw data - sample 1, sample 2
0000 0101 1111 1100	Continuing raw data - sample 2, sample 3
0000 0000 1000 0011	Continuing raw data - sample 4, sample 5
0111 0000 0000 0000	Continuing raw data - sample 5, unused
1011 0001 0101 0000	End of message - ADC samples: 5, internally triggered, unused, normal stop

Table 6.2.: Example: a normal hit message that consists of seven 16 bit data words. The important 4 bit message word preambles are colored red (note that, as the only exception, the continuation preamble has only 1 bit).

As the example shows, each word of a message is interpreted by first looking at the preamble (red text) and then on the remaining bits following thereafter (blue and green text). The example particularly demonstrates how the continuation word (preamble 0) is used and how it can be exploited in certain cases to slightly reduce the preamble overhead. Moreover, the example shows how the selected 9 bit ADC sample values have been folded by the data wrapper to better exploit the given 12/15 bit word format (see section 6.3.7 for more details on the implementation). Note that the stored number of ADC samples (5 in

this example) is not a redundant information, since the unused bits after sample number 5 could as well contain an additional sample number 6.

6.3.6. Hit Control and Message Builder

The allocation of tasks between the hit control and the message builder is simple: the hit control generates all major control signals that are required to handle the full message building process, whereas the message builder passively arranges and multiplexes the respective raw data and meta data words and leads them to the output buffer. The hit control hence actively manages the message generation, whereas the message builder only takes orders.

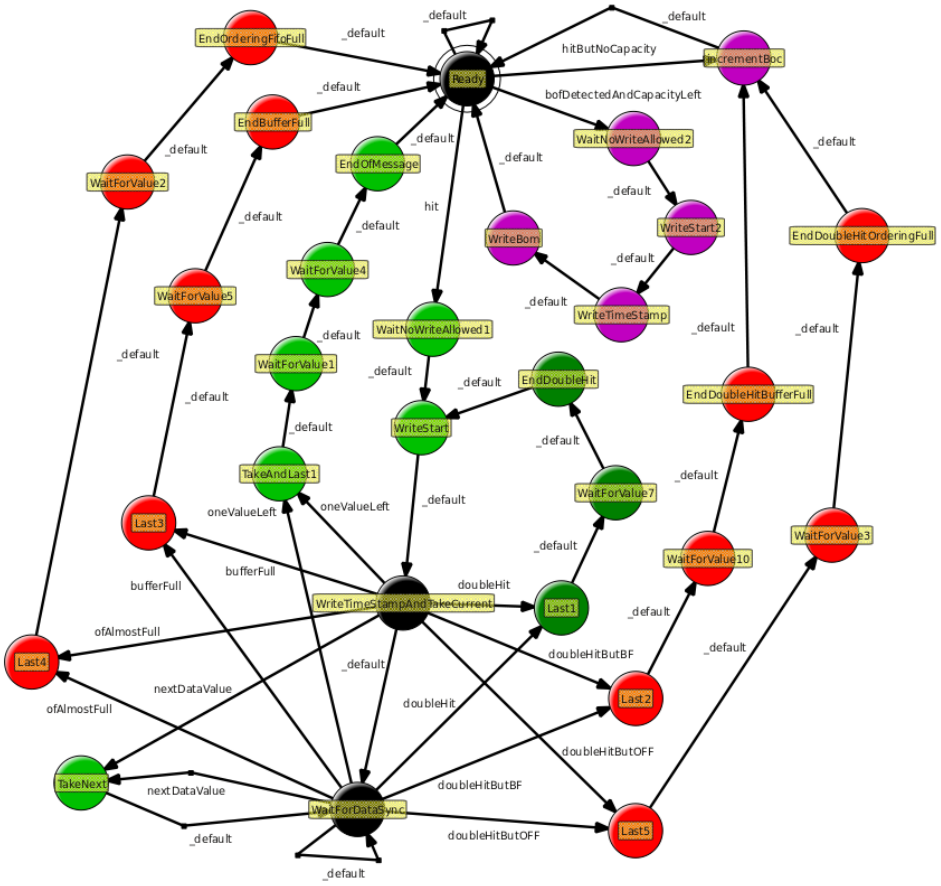


Figure 6.5.: State diagram of the finite state machine (FSM) used the hit control.

In Fig. 6.5 a state diagram of the FSM (finite state machine) used in the hit control is shown. Only if the FSM is in a black state, a new hit signal can occur. Because the hit signals are synchronous to the 25 MHz clock (which is assured by the hit synchronization) whereas the FSM is clocked with 125 MHz, no path longer than five states between two black states can be found in the FSM. And for the same reason, the FSM can only idle in black states. Light green states mark paths that are passed through during the generation

of a normal message, while the dark green path processes a multi-hit. In contrast, red paths each handle a certain abort situation, while the pink states are used to realize the lost-hits mechanism described earlier.

As an example, the path through the FSM that has been taken during the generation of the example message shown in the previous section is now briefly described: Starting in the black idle state on the top (*Ready*), the recognition of an incoming pulse causes the state pointer of the FSM to travel to the black state on the bottom (*WaitForDataSync*). During the passage, the start of message word (*WriteStart*) as well as the time-stamp (*WriteTimeStamp...*) word is written to the output buffer. Then, while further incoming raw data values are selected (selection mask), wrapped (data wrapper), and shifted to the output buffer (message builder), the FSM stays in the small loop on the bottom and only switches between the two states *WaitForSync* and *TakeNext*. If finally no further incoming raw data value shall be recorded, the state pointer travels back (using the other light green path) to the initial state *Ready*. Eventually, on the way back to the top of the FSM, the end of message word is written to the output buffer (*EndOfMessage*).

An important detail of the FSM are the various wait states. They are required to wait for the data wrapper, which can require up to three 125 MHz clock cycles to finalize a raw data message word — as it is described in the next section.

6.3.7. Data Wrapper

As it has been visualized in the example of a normal message (table 6.2), the task of the data wrapper is to properly rearrange the 9 bit data values into words of 12/15 bit. Two requirements make the implementation rather complicated: first, the fact that the first output word must have a length of 12 instead of 15 bit (due to the longer preamble of the first data word). And second, any 9 bit input word might be the last that should be taken (the decision is made by the hit control) and thus can force the immediate completion of the internal wrapping procedure (that is exactly the reason for the wait states in the hit control FSM, which have just been mentioned in the previous section).

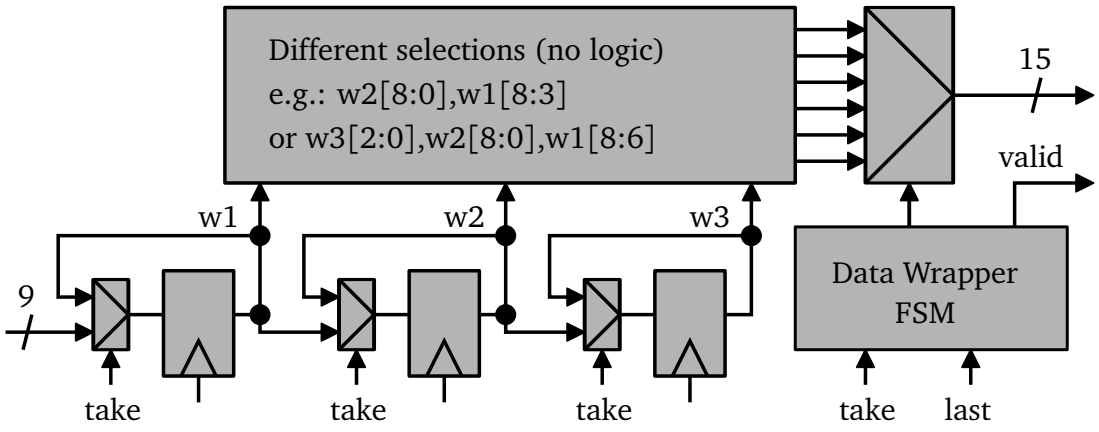


Figure 6.6.: Simplified block diagram of the data wrapper realized in SPADIC 1.0.

The interface between the hit control and the data wrapper only requires the two signals *take* and *last* (inputs of the data wrapper), which are synchronous to the 125 MHz. If one thinks of the data wrapper as a black box, it temporarily stores the incoming 9 bit input word each time *take* is set and eventually (the delay depends on the internal state) generates either a 12 (first word) or a 15 bit output word. Moreover, if the *last* signal is set by the hit control, the internal memory of the wrapper is flushed and the output words which are not yet finished are generated as quickly as possible.

The structure of the implemented data wrapper is sketched in Fig. 6.6. The idea is to shift the 9 bit input words (coming from the left hand side) into a sufficiently large shift register each time *take* is set. At the same time a FSM, which has all the intelligence, controls a multiplexer that switches between the required selections of the temporarily stored data values. Moreover, the FSM tells the subsequent stage when a new 15 bit output word is available (or is *valid*).

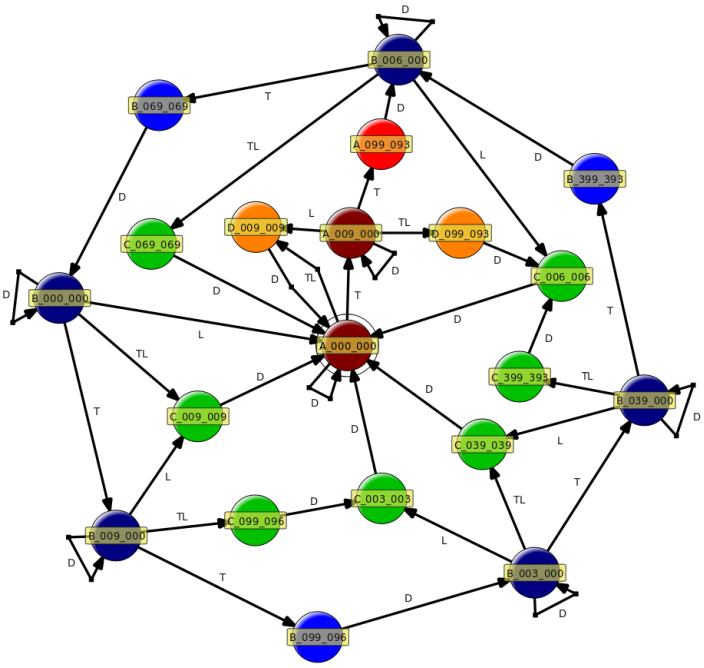


Figure 6.7.: The FSM of the data wrapper realized in SPADIC 1.0.

Even though in the present case the number of required multiplexer inputs is rather small (8 input combinations in the present case) and the required shift register can be very short (three stages), the generation of a (minimal) FSM is rather complicated. For that reason a formalism together with a simple algorithm has been developed, that can be used to build a minimal data wrapper FSMs for any combination of input and output word widths (the input width must be smaller than the output width though). A detailed explanation of the algorithm and an example is given in the appendix (A.3). The developed formalism is very important in the present case, because if either the ADC resolution or the message definition is changed in a subsequent SPADIC version, at least the data wrapper FSM, the shift register length, and the multiplexer combinations have to be adapted properly. The

minimal FSM that is implemented in the data wrapper in SPADIC 1.0 is shown in Fig. 6.7. It was designed with the help of the data wrapper algorithm.

In order to demonstrate the aesthetics of the FSM, a short "round trip" is taken here: The state in the middle of the FSM is the reset or initial state. A transition can only relate to 4 different cases: T (*take*), L (*last*), TL (both), and D (default). As long as only T (but no L) occurs, the FSM pointer travels from the center state along the red path to the top of the FSM and then, circle after circle, along the outer (light and dark) blue path. If eventually an L (or a TL) occurs, the pointer is forced to take some green path back the center (and thus to finalize the procedure). Moreover, dark colors mark states in which the FSM waits for the next occurrence of T, L, or TL, whereas lightly colored states are states that are immediately (on default) passed. Because the light states are internally required, it must be made sure by design (or by the hit control, wait states), that T, L, or TL can only occur if the FSM is in a dark state.

6.4. Message Transport, Synchronization, and Epoch Channel

The previously described hit logic or, to be more specific, the output buffers define the end of a channel. Consequently, the focus now leaves the local channel context. In this section it is explained how the temporarily stored messages are further handled and how they finally find their way out of the chip. Moreover, various global aspects like for instance the synchronization mechanism or the generation of epoch markers are explained.

A detail which becomes important in this context should be emphasized again: The 32 channels of SPADIC 1.0 are grouped into two 16-channel blocks (also called a channel group) and therefore most of the logic parts are implemented twice (similar to the neighbor matrix discussed earlier). From an abstract point of view, each SPADIC 1.0 channel group in principle behaves (except for some shared global parts) like a separate chip with only 16 channels. In particular, each channel group produces its own and physically separated message stream. For that reason many of the subsequent building blocks refer to 16 instead of 32 channels.

6.4.1. Channel Message Switch

The purpose of the channel message switch is to arbitrate and transport the messages stored in the output buffers of the 16 (+1, see next section) channels to one single message stream buffer (again a FIFO). In general, various methods to perform such a task are known, very popular for instance is the so-called token-ring network (e.g. implemented in the n-XYTER chip [23]), which basically realizes the Round-Robin scheduling algorithm. Even though the token-ring network has a very simple structure, is statistically very fair (no channel is favored), and can be effectively built, it has one major drawback: the initial arriving order (time-stamp) of the arbitrated messages is lost and a subsequent sorting becomes necessary.

In order to overcome the latter problem, the channel switch realized in SPADIC 1.0 takes a different approach (Fig. 6.8): The idea is to use an "ordering FIFO" that stores the

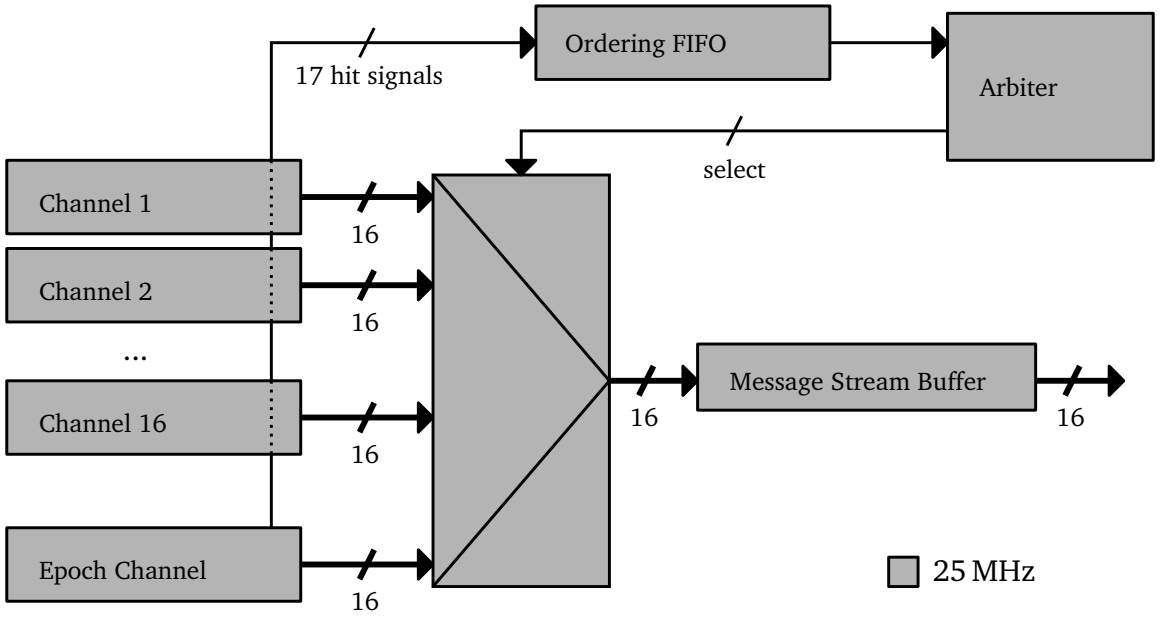


Figure 6.8.: Simplified block diagram of the channel switch based on an ordering FIFO.

channel number each time a new message is generated and is later read out by the arbiter to reconstruct the initial order of arrival. That way, the produced output message stream is guaranteed to have a preserved time-stamp order. In detail, the implementation is simple: The ordering FIFO has an input word length that equals the number of channels (one hot, 16 (+1) in the present case). Then, each time one or more hits are generated within a certain time-slot (one 25 MHz clock cycle), a (17 bit) vector is written into the FIFO, with each 1 representing a new message that is waiting in the corresponding output buffer. Then, everything the arbiter has to do is to repeat the following loop: read a new word from the ordering FIFO (if it is not empty) and arbitrate (in any order) one message of every marked channel.

The input logic before the ordering FIFO can be easily built: it is sufficient to write the 16+1 internal hit signals of all channels directly to the FIFO every time the vector is not null (large OR). And also the arbiter and the logic reading the FIFO are not very complicated in detail. Therefore the dominant costs of the ordering FIFO approach is the ordering FIFO itself. If one wants to make sure that the ordering FIFO cannot become full, its depth must be equal or larger than the maximum number of messages that can be stored in the output buffers at once. In SPADIC 1.0 that number is roughly 170 for normal messages, but depends of course on the stored message types, on the selected number of ADC values, et cetera. But even if the latter condition is not perfectly fulfilled, the probability of a full ordering FIFO can stay rather small: a dedicated study, that has compared the data flow structures of n-XYTER and SPADIC, came to the conclusion that even an ordering FIFO having only 80 % of the maximum number of messages that can be stored in the output buffers, effectively leads to a fraction of lost messages due to a full ordering FIFO far below $1‰$ [15]. Therefore, the implemented ordering FIFO of SPADIC 1.0 only has a depth of

128 (≙ 75 %, conservative estimation). In return, the event of a full ordering FIFO must be properly handled by the hit control FSMs (as discussed earlier).

A very crucial problem for the implemented channel message switch are bit-flips in the output buffers (or to be more specific in the message preambles) or in the ordering FIFO. As briefly discussed in section 6.6, bit-flips due to single event effects (SEE) become likely in high-radiation environments such as CBM. And especially the SRAMs that have been used to realize the FIFOs of SPADIC 1.0 are potentially vulnerable. In general, one can identify four major cases which can become a serious problem for the arbiter:

1. Bit-flip in the preamble of an end of message word.
2. Bit-flip in the preamble of a message word, falsely making it an end of message word.
3. Bit-flip from 1 to 0 in the ordering FIFO.
4. Bit-flip from 0 to 1 in the ordering FIFO.

The first and second cases are problematic, because the arbiter has no information of the actual message length and hence totally relies on the preambles (or, to be more specific, on the detection of end of message words). The third case leads to the skipping of messages, which, as a direct consequence, causes the corruption of the message order. And the forth case either causes a deadlock, if no message is available but the arbiter tries to arbitrate, or again a corrupted message order, if some message is arbitrated too early.

A first attempt to solve the latter situations was made in SPADIC 1.0. The realized but certainly not perfect solution basically has two features: first, timeouts were introduced to force the arbiter to continue operation, if a deadlock is recognized, and second, all remaining messages in the output buffers are shifted out, if the ordering FIFO stays empty for too long. That way, the implemented solution at least guarantees two things: First, the arbitration will never stop and second, if for some time no new hit occurs, all remaining messages are eventually arbitrated – but in that case the message order is not preserved. Even though a more conventional method that should be implemented in the next SPADIC iteration instead would be protecting the FIFO words with some additional ECC (error-correction code) bits. Thus the radiation tolerance of all buffers could be significantly increased – but without the need to adjust any of the various logic parts.

However – and that is at least conceptually an important detail –, whenever the arbiter starts to handle a timeout or to flush the output buffers, a dedicated info message is added to the data stream and thus the information that an unusual event has occurred is propagated. Therefore, by monitoring the data stream (or more exactly the message word preambles) any takeover of the corruption logic can be recognized efficiently and a potential corruption of the message stream can be predicted.

6.4.2. Epoch Channel

The length of the time-stamp is a trade-off between longer time intervals and less data overhead. Because the 12 bit time-stamps of SPADIC 1.0 (see the message word definition

in table 6.1) is synchronous to the 25 MHz clock, it takes only 163.8 µs before the time-stamp counter wraps around. Such a short time period alone would make a system-wide temporal separation of all recorded messages practically impossible. But contrariwise, a further increase of the time-stamp would enlarge the size of each single hit message (an additional 16 bit message word would be required in the present case) and thus significantly raise the total data traffic (by roughly 10 to 15 % here).

To overcome that problem SPADIC 1.0 makes use of the well-known technique of injecting additional time markers (so-called epoch markers) into the (temporarily ordered) message stream. The general idea is to add a short epoch marker message each time the 12 bit time-stamp wraps around. In doing so, the effective time period is significantly increased, whereas the additional data traffic stays fairly small[1]. The epoch message defined in SPADIC 1.0 has a 12 bit epoch number, which effectively increases the time periods to 24 bit or 0.671 s.

Even though the principle of epoch markers is very simple in theory, the implementation of a proper epoch marker injection can be rather complicated in general. Intuitively, one might want to place the injection mechanism behind the channel switch, because at that position all messages are already combined and temporarily ordered. But in fact there is absolutely no possibility to find the proper injection positions in the merged message stream, since in fact the global temporal context is already lost (otherwise the time-stamps were not necessary at all). Therefore the only feasible position for the epoch marker injection is either besides or in front of the channel switch.

However, the actual implementation of the epoch marker injection circuit operated in SPADIC 1.0 benefits strongly from the ordering FIFO approach the channel switch is based on: As mentioned before, the whole trick is to add a dedicated 17th “epoch channel” in parallel to the 16 normal channels and to connect it to the ordering FIFO in the same way as the 16 normal channels. In doing so, the only task the epoch channel has to perform is to generate an epoch message (and to inject it into the output buffer) each time the time-stamp counter wraps around. Due to the fact that the ordering FIFO preserves the temporal order of the messages, the generated epoch messages are thus automatically injected at the correct positions and no further modification of any other logic part is required.

As a direct consequence the whole implementation of the epoch channel comes down to the development of an extremely simple FSM, which basically controls a multiplexer writing into the output buffer. More concrete, all the implemented epoch channel does is injecting a two-word message into the output buffer each time the time-stamp counter wraps around[2]. In doing so, each of the such generated epoch messages contains besides

[1] If the hit rate is high, the additional traffic due to the epoch marker messages can be neglected, and if the hit rate is low, the additional traffic is high compared to the data traffic, but at the same time plenty of bandwidth is available.

[2] This note might be important for people actually working with SPADIC 1.0: even though the statement is correct from an abstract perspective, the actual technical details of the event of a wrap-around are slightly different and do not only depend on the time-stamp counter, but also on the DLM synchronization mechanism described in the next section. As a consequence, the epoch markers are not produced by default, but require the CBMnet (also next section) to work properly. In fact the epoch channels do not only monitor the time-stamp wrap around, but also wait for the synchronous arrival of a certain DLM. This scheme is currently under discussion and will probably be changed in the next SPADIC iteration.

the 12 bit epoch number – similar to a normal hit message – also the source address of the corresponding 17-channel block (the group ID).

6.4.3. CBMnet

The so-called CBMnet is a transport and synchronization protocol intended to manage the data and control traffic between most parts of the CBM DAQ system (section 3.4), and moreover to provide a mechanism for the system-wide (time) synchronization. The fundamental principle of CBMnet has initially been developed in the context of a dissertation [45], whereas the developments are still ongoing. CBMnet will be integrated in nearly all DAQ nodes, both as firmware in FPGAs and as semi-custom design in ASICs, and even a dedicated CBMnet data aggregation chip is currently in the planning stage (working title "HUB chip").

With SPADIC being a leave of the final network tree, it is required (or at least desired) to make SPADIC 1.0 directly "speak" CBMnet. For that reason, the latest available CBMnet version 2.0 was not only integrated into SPADIC 1.0, but was indeed made the main communication interface (a fall-back test interface is also available though). In this regard SPADIC 1.0 is a pioneer among the CBM ASICS, since it is the first ASIC carrying a CBMnet block adapted to silicon and therefore serves as a reference for the other ASICs. For instance, the so-called "STS-XYTER" (for the readout of the CBM STS, first results expected beginning/middle of 2013, no publications yet) that is architecturally very similar to the n-XYTER [23], will have nearly the same CBMnet back end as SPADIC 1.0 – and, by the way, also uses the UCL standard-cell library (section 5.4.2) and some of the SPADIC IO cells.

6.4.3.1. Principle of Operation

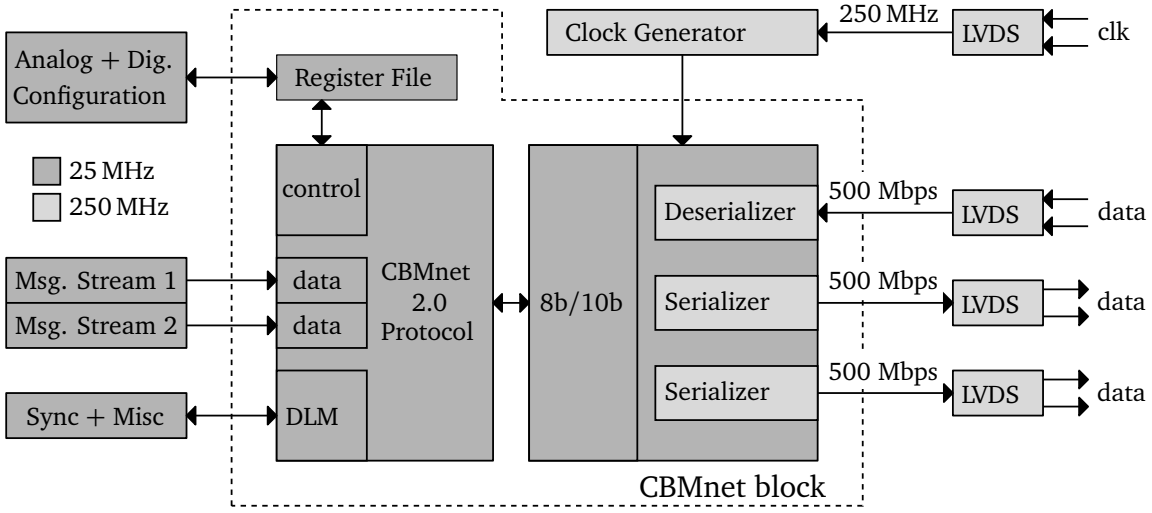


Figure 6.9.: Simplified block diagram of the CBMnet block and its periphery.

A simplified block diagram of the CBMnet block and its periphery, as it was realized in SPADIC 1.0, is shown in Fig. 6.9. From a user's point of view, CBMnet provides simple

interfaces for control, data, and synchronization packages (left hand side of Fig. 6.9), while all important tasks like for example link management, error correction, or (de-)coding are handled internally and thus stay transparent. For communication with the outer world, the physical interface of the CBMnet block of SPADIC 1.0 requires only four LVDS pairs (right hand side of Fig. 6.9) – two inputs (clock and serial data/control) and two outputs (twice serial data/control).

The incoming LVDS clock has a nominal frequency of 250 MHz. It is directly fed into a clock generator that creates all internal (single-ended) clocks (250 MHz, 125 MHz, 25 MHz, and the ADC sync signals). That scheme in particular means, that all logic parts of the ASIC are synchronous to the incoming CBMnet clock or, in other words, that nowhere a local clock context exists[1]. That detail is particularly important for the system-wide synchronization mechanism which is explained further below.

All three available package types (data, control, and synchronization) are transported over the same serial LVDS data connections, which run at 500 Mbit/s DDR (double data rate). Serializers and deserializers clocked with 250 MHz are operated to translate between the serial LVDS data streams and an internal 20 bit interface that runs only at 25 MHz. If one subtracts from the internal 20 bit interface the 8b/10b (de-)coding overhead, one directly comes to a raw data width of 16 bit (25 MHz), which is available at the control and data interfaces.

6.4.3.2. Link Initialization

Before any user data can be sent or received, the CBMnet block runs a routine to initialize the link, which is only briefly discussed here (strongly simplified). In the first step, the send and receive delays of the IO-pads in the subsequent stage connected to SPADIC (an FPGA or later maybe the HUB chip) are properly adjusted (phase adjustment on the 125 MHz level) in order to assure a correct sampling of both sides. The SPADIC IO-cells itself do not have adjustable delays (so far) and hence totally rely on the subsequent DAQ node. Due to the usage of well-known initialization characters, the phase adjustment can be done automatically and dynamically. Then, during the second step, the barrel shifters behind the serializers/deserializers (on both sides) move their windows of interest until familiar 8b/10b characters are found. Finally, after the barrel shifter positions have settled, the CBMnet protocol blocks of both sides perform a simple handshake and eventually pass over to normal operation mode. To announce the latter, a ready signal (link active) is set, telling that the user interfaces are now ready to operate.

It is important to note, that at least the first two steps just described are in fact no direct part of the CBMnet, but an internal initialization sequence of the implemented serializers. And in fact, in other CBMnet network nodes, which for instance use commercial FPGA transceivers, the low-level part of the initialization sequence can be very different.

[1] The same is true for all other components within the whole CBM network (or at least for those using CBMnet). Therefore, in the end, all frequencies of the CBM DAQ will be derived from one central clock. That methodology assures that the whole system is clocked synchronously.

6.4.3.3. Retransmission

An important feature introduced with the CBMnet version 2.0, which shall only be briefly noted here though, is the retransmission mechanism. The idea is simple: The sender holds a copy of each package that has been sent until an acknowledge (ACK) comes back from the receiver. Instead, in case the receiver recognizes the corruption of a package (each package has some CRC bits attached), a not acknowledge (NACK) is sent back, telling the sender to repeat. The implemented output buffer can store 256 package words (640 B) and therefore at least 7 packages.

An alternative to retransmission would have been to implement a forward-error correction (FEC) instead. But because one expects the error rates in the given application to stay rather small, the retransmission mechanism promises a better utilization of the link – but for the costs of the relatively large output buffer and occasionally increased delays.

6.4.3.4. Data and Control Interfaces

Both the data and the control interfaces are technically very similar, but have different purposes. The data interface is intended to predominantly transport actually measured data, such as for instance the message streams generated by SPADIC 1.0, whereas the control interface is basically used to send and receive different kinds of "slow control" messages. Although many ideas or possibilities of feasible control messages exist, no binding control message specification has been written by the CBM community so far. In particular, that means that no reliable list of mandatory commands exists, which for instance the front-end ASICs must understand and implement. For that reason, the two (one for each direction) CBMnet control interfaces of SPADIC 1.0 are solely connected to a large register file, which is used for the digital (directly) and the analog (indirectly) configuration. Consequently, the only purpose of the control interface of SPADIC 1.0 so far is to read and write the configuration.

Both the control and the data interface operate at 25 MHz and transfer 16 bit words. Two unidirectional data interfaces are available in SPADIC 1.0, each taking the message stream of one 17-channel group[1]. Whenever the data or the control interface is ready to take new data, the user can at once inject a complete package with any size between 4 words (64 bit) and 32 words (64 B). To keep the utilization as high as possible, in SPADIC 1.0 the generated message streams are simply cut into pieces of 32 message words (while disregarding the respective message borders), as long as enough message words are stored in the output buffer (behind the channel switch). Otherwise, the stream is cut – so to say arbitrarily – into smaller pieces. Then, at the receiving end, the CBMnet packages (arriving in the same order as they have been generated) are concatenated again and thus the initial message stream is rebuilt. If the data interface is ready to take the next package, but the number of stored message words in the output buffer contains less than 4 words (the minimal package size), the missing package spaces are filled with empty message words (which can be removed any time later). That scheme had to be introduced to assure that

[1] The first LVDS data output transports the data from the first user data interface as well as both the outgoing control messages and the synchronization messages, whereas the second LVDS data output is only used to transport the data messages from the second user data interface.

even remaining message parts or smaller messages (e.g. an epoch message) are transferred in time.

Additionally to the payload a 4 B (two words) header (start of package and source address) and a 4 B (two words) trailer (CRC and end of message) are added to each package. In particular, the user must take care by himself that the source address (1 word) is put in front of each new package. Therefore – without the additional 8b/10b overhead – the CBMnet protocol overhead is 12.5 % (best case, 64 B packages only), but can rise to 100 % (worst case, 4 B packages only).

6.4.3.5. Synchronization Interface

The synchronization interface gives access to one of the probably most central features of CBMnet: the deterministic latency messages (DLM). A DLM is a single word 4 bit message, which is guaranteed to need a constant (and measurable) number of clock cycles when traveling from one node in the CBMnet network to another. Technically, that means that the DLMs are implemented to bypass all elastic buffers and wait states within the network.

In order to measure the respective path latencies, a dedicated DLM (DLM0) can be looped between any two network nodes and thus the number of required clock cycles of any path can be counted. Moreover, using the DLM mechanism, it is intended to align all path delays within the DAQ network tree: to adapt a certain sub-tree, the latency from the sub-root to all sub-leaves is initially measured. Afterwards, all individual path delays are properly adjusted, until a DLM sent by the sub-root exactly reaches all sub-leaves at once. By repeating that routine on all levels of hierarchy (starting from the network root), eventually all network leaves can be reached simultaneously (as well as all intermediate nodes that are on the same level of hierarchy). Moreover, all path delays are well known then.

In a completely initialized, measured, and adjusted network, the DLMs can be used to send different kinds of synchronization signals to all nodes. Whereas DLMs are broadcast on default, one can mask certain sub-trees by setting proper registers of the intermediate nodes beforehand (via control messages). Thus DLM broadcasts can be limited to single branches or in principle even to single nodes or leaves.

Until now 5 of the 16 different available DLM types were globally defined. The remaining DLMs are either reserved or can be used for user specific purposes:

Type	**Official Purpose**	**SPADIC 1.0 Implementation**
DLM0	Delay measurement	Looped back within 1 clk cycle
DLM1	Periodic time counter reset	Resets time-stamp, triggers epoch
DLM2	Set new counter value	Updates epoch from register
DLM3–7	Reserved	Unused
DLM8	Start DAQ	Enable trigger mechanism
DLM9	Stop DAQ	Disable trigger mechanism
DLM10	User DLM	External trigger (to LVDS output)
DLM11	User DLM	Internal force trigger (digital)
DLM12	User DLM	Internal force trigger (analog)
DLM13–15	User DLM	Unused

As shown in the table above, the mandatory DLMs 1 and 2 are used to synchronize time-stamps or epoch counters. In the present SPADIC implementation, the arrival of DLM1 resets the time-stamp counter to 0 and at the same time forces the generation of a new epoch marker. If (in the current implementation) the DLM1 arrives asynchronously to the internal time-stamp wrap around instead – which normally indicates some kind of problem –, a special epoch marker instead of a normal epoch marker is sent. The special epoch marker contains the deviation (number of cycles) between the internal time-stamp wrap around and the arrival time of DLM1. The occurrence of DLM2 tells the epoch counter not to simply increase the epoch the next time the time-stamp wraps around (or the next time a DLM1 arrives), but to take the value of the next epoch from a dedicated register instead (which can be set properly before the DLM2 is sent).

DLM8 and DLM9 are used to tell the readout leaves to start or to stop data recording. That explicitly does not affect older data messages that are already traveling through the network. In SPADIC 1.0 DLM8 and DLM9 simply enable or disable the hit detector mechanism. Therefore, if the readout is disabled, no new hit messages are generated – but epoch markers are still produced (there are other ways to stop the epoch mechanism as well). Before the design of the next SPADIC iteration is started, the latter scheme should be discussed again.

The freely definable user DLMs 10–12 are used in SPADIC 1.0 to initiate the generation of different internal and external trigger signals. Via the internal digital trigger (DLM11) one can force selected (via a trigger mask) hit logic blocks to generate a new message. The internal analog trigger (DLM12) toggles a CMOS inverter that is AC coupled to the analog input of channel 31. Moreover, in order to be able to trigger some external device via LVDS (e.g. some analog pulse generator), a certain LVDS output pad can be toggled using DLM10. On the second SPADIC 1.0 PCB, the external LVDS trigger signal is connected to an analog pulse generator, which injects a voltage step into a wire being placed on a PCB layer directly underneath the CSA input fan-out (and thus couples a certain amount of charge into all CSAs at once).

6.4.3.6. Register File

The implemented register file is provided together with the CBMnet logic block (compare again Fig. 6.9) and is adapted to the SPADIC 1.0 requirements. As mentioned earlier, it is directly connected to the CBMnet control interface and thus can be remotely accessed (read and written).

The register file directly stores the whole digital chip configuration (e.g. enable/disable channels, trigger thresholds, or neighbor relationship) and, moreover, gives access to the analog shift register configuration (see section 5.4.1). To be more specific, the analog bit vector must be stepwise written to a dedicated register file address. Internally, the bit vector is converted into a serial bit-stream and is then properly shifted into the analog register chain. Similarly, by reading from another dedicated register file address several times in a row, the analog configuration can be read back again.

A detailed documentation of the complete register file can be found on the SPADIC website [11].

6.5. Other Design Aspects

Other important digital design aspects that did not fit into the previous structure are listed below.

6.5.1. Some General Numbers

The digital part of SPADIC 1.0 extends roughly 75 % of the active die area (compare Fig. 6.10) and has a total size of 3.5 mm x 4.5 mm. The total memory of all 44 SRAM blocks is 4.4 kB, whereas the register file additionally stores 1270 bit (distributed). Excluding the SRAM blocks, the digital part consists of 2.09×10^6 transistors and is routed with a total wire length of 14.39 m.

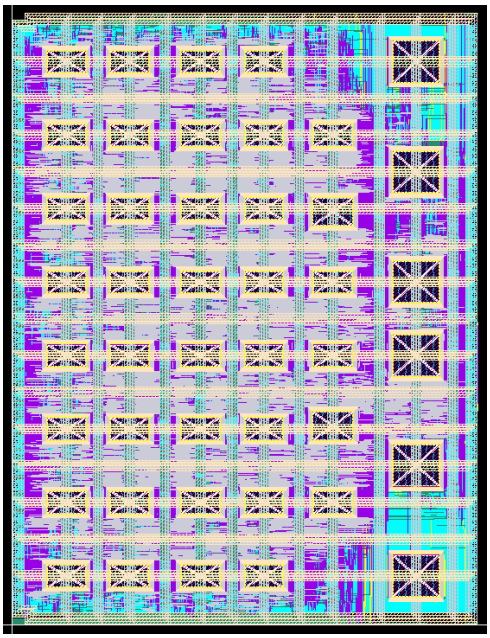


Figure 6.10.: Layout of the whole digital part of SPADIC 1.0. The cut-out has a size of 3.5 mm x 4.5 mm. The black boxes are place holders for the 44 SRAM blocks. The 6 larger SRAMS on the right hand side are used within the CBMnet, basically to buffer outgoing packages for a potential retransmission, whereas the remaining SRAMS are used for the 34 channel output buffers, the 2 ordering FIFOs, and the 2 channel switch buffers.

6.5.2. Performance and Results

The maximum hit rate the chip can handle (at the nominal frequency 250 MHz) can be easily calculated: The output rate of both serial LVDS links is 1 Gbit/s, which drops down to 800 Mbit/s after the 8b/10b decoding. The protocol overhead of CBMnet is 12.5 %, leading to a maximum (best case) message stream data throughput of 700 Mbit/s or a

maximum message word rate of 43.75 Mw/s (the average epoch marker word rate is only 12.2 kw/s and can be ignored here). With the conservative assumption that an average hit message consists of 10 message words and considering the total of 32 channels, the theoretical maximum hit rate per channel becomes 136.7 khits/s. But that maximum can only be reached with a sufficiently even distribution of the incoming hits over all channels (otherwise data loss in the buffers would occur). And of course, the hit rate decreases accordingly, if the neighbor trigger mechanism is enabled.

So far SPADIC 1.0 has been successfully tested up to 200 MHz[1]. The proper functionality of all digital blocks could be shown, although some minor bugs, which are all due to logical reasons, were discovered. But because the problems that have been found so far are technically, operationally, and conceptually uncritical, they are not further discussed here. The total power consumption of the digital part (with the de-/serializers running but the CSAs and ADCs off) at 200 MHz is 594 mW (static 144 mW, dynamic 450 mW)[2]. At 250 MHz, the predicted power consumption (by the CAD tools) is 682 mW (static 90 mW, dynamic 592 mW), which – if one scales the dynamic part properly – leads to a predicted power consumption at 200 MHz of 563 mW. Fortunately, both numbers match well enough.

6.5.3. Fall-back Solutions

As a part of the fall-back plan a very simple test data output interface is added in SPADIC 1.0 in order to be able to bypass the CBMnet. The alternative interface allows to directly read out one of the two message streams (selectable) of the channel groups. To reduce the number of pins, the message stream (16 bit at 25 MHz) is partly serialized (4 bit at 125 MHz) and sent out via 4 LVDS output pairs. And indeed, the test data output has been excessively used for the first measurements, although, unfortunately, a small logic bug sets some limits to both the readout speed and the general reliability of the interface (which is actually a good example why the reliable message word definition can be very helpful).

Similar to the test data output, a digital test data input interface is available. Again to decrease the number of pins, a simple double data rate (DDR) receiver is operated to translate the input words coming in on 4 LVDS pairs (4 bit at 25 MHz, DDR) into the internal raw data format behind the ADC (9 bit at 25 MHz, 2's complement, LSB fixed to 0). The test data input can be used to overwrite the ADC raw data stream of channel 0 and thus to reliably test the complete digital channel logic. That way, nearly all important channel features could be tested. The IIR filter for instance has been exactly verified, the trigger and hit building logic has been completely checked, or the neighbor trigger mechanism has been launched.

Moreover – also as a CBMnet emergency bypass – an I^2C back door to access the register file is implemented. Unfortunately, the dedicated I^2C pad (open drain IO-pad) that has been designed for that particular purpose, has turned out to have an internally inverted SDA signal. That means in particular that the I^2C pad pulls down the SDA bus, if the ASIC wants to listen, or releases the bus, if the ASIC intends to write (which is just the opposite of what one would like to have). Even though, this problem could be partly solved by

[1] The present FPGA setup is limited to 200 MHz.

[2] Preliminary numbers.

adding an external pull-up PNP transistor forcing the external SDA bus to the respective signal levels defined by the I^2C master. That way the bidirectional SDA bus has effectively been made unidirectional. As a direct consequence, via I^2C the register file can be only written but not read – which, however, is possible via the CBMnet, which therefore can be interpreted as the fall-back solution of the fall-back solution.

6.5.4. Future Improvements

Even though the mixed-signal system SPADIC 1.0 has already proved the whole concept to work properly, the list of new beneficial improvements, new ideas, and minor bug-fixes is already long. A complete listing would be inappropriate here, which is why only some few selected aspects are given subsequently.

One of the major challenges on the way towards a final SPADIC iteration will probably be to guarantee for a sufficient radiation tolerance of the whole design. Whereas the analog part, as briefly discussed in section 5.5, is expected to be already sufficiently radiation tolerant, the digital part and especially the used SRAM cells most certainly need some additional protection (see section 6.6 below). But before further changes to improve the radiation tolerance are implemented, some accurate irradiation measurements of SPADIC 1.0 should be performed and evaluated. Only a well-founded understanding of the weak spots, the exact statistics, and the specifics of the technology can lead to a design that is both radiation tolerant and effective.

A comparatively easy but also important topic might be to properly rewrite some of the message definitions. For instance, the two-word epoch marker message could be composed of an epoch word and a continuation word as well (instead of a start of message word and an epoch word). That would allow for larger epochs (19 bit instead of 12 bit) without any loss of functionality.

Furthermore, at some point a concrete guideline has to be defined, telling how the various devices within the CBM DAQ network shall react to certain events – for example how warnings or errors shall be propagated. The chosen methodology of SPADIC 1.0 to inject info and error markers directly into the message stream might not be the best solution in the end – especially since other CBM ASICs most probably will not provide such a methodology. Another possibility for instance would be to write dedicated status registers whenever interesting or problematic situations occur. But then a global poll strategy of the DAQ system has to be defined. And moreover, one should consider using the CBMnet control message interface for other purposes than only accessing the register file.

Another minor but also interesting feature will be to enable clock gating. In SPADIC 1.0 all digital channels can be separately disabled (as well as all analog channels). However, this so far is only realized on a logical level, meaning that the clock signals still reach the registers and thus some dynamic power is still consumed. But in fact the whole HDL syntax has already been written such that enabling clock gating requires only to set a certain parameter in the synthesis scripts (compare section 6.7) – and even a dedicated clock gating cell has been added to the UCL standard cell library. The only reason why clock gating has not yet been introduced in SPADIC 1.0, was to remove one rather risky entry from the already long list of new features.

A potentially very beneficial but in detail probably complicated feature that could be shifted to the SPADIC chip is the feature extraction. The purpose of the feature extraction in general, which is intended to be so far operated on an FPGA elsewhere in the DAQ network behind the SPADIC, is to extract the exact pulse amplitude and an interpolated fine time-stamp from the recorded raw data values (see sections 5.3.1 and 5.3.2). In general, shifting the feature extraction from some FPGA directly to the SPADIC would reduce the message size, but would also limit the overall flexibility. For the latter reason, if an on-chip feature extraction should be operated, the extraction algorithm must be perfectly well settled, excessively tested with real data, and rather simple.

6.6. Digital Radiation Tolerance

As mentioned earlier (see also section 5.5), the main problems of the digital part are (or might be) single event upsets (SEU) or single event transients (SET). In general, a bit-flip can cause all kinds of undesired effects from wrong data words to invalid logic states and deadlocks, or even to the damage of hardware components (e.g. if due to a wrong configuration two driver cells are shorted). So far, the digital part of SPADIC 1.0 is not explicitly designed to handle SEU/SETs, but makes at least some first attempts (compare for instance the channel message switch 6.4.1).

Making a good prediction of realistic SEU/SET cross-sections of flip-flops or SRAM cells is not trivial, especially since the statistic numbers strongly depend on the details of the layout and the circuit, on the modes of operation, on the used technology, and on the type of radiation. The study of the "GRISU" test ASIC, that has been designed in the same technology as the SPADIC, for instance predicts relatively large SEU cross-sections of about $\mathcal{O}(1 \times 10^{-14}\,\mathrm{cm}^2/\mathrm{bit})$ for protons[1] hitting "normal" flip-flops [47]. But however, for a more exact and reliable estimation of how the SPADIC building blocks (e.g. the home-made standard cell library (UCL) or the used commercial SRAMs) will respond to irradiation, comprehensive measurements will have to be performed – as mentioned earlier.

Without going to much into detail here, in general, the radiation tolerance of digital circuits can be increased either by using physically improved libraries (e.g. higher capacities of storage nodes can help to increase the critical linear energy transfer levels) or by introducing certain redundancy techniques. Examples of the latter are the triplication of critical logic parts in combination with the adding of voters to the outputs, or the appending of some error-correction code (ECC) to the words being stored in larger memory blocks.

6.7. Tooling

Using sophisticated CAD tools can be everything between a scientific challenge and a indispensable necessity – it most certainly depends on the available experience and expertise, the complexity of the task, but also on the personal taste. Most parts of this work are actually focused on the scientific or at least the essential elements rather than on handling and

[1] The cross-sections for heavy-ions are within the range $\mathcal{O}(1 \times 10^{-10}\,\mathrm{cm}^2/\mathrm{bit})$.

tooling details – which however often dominate in the daily business of engineers. That is especially true for many digital development steps, where high-level tooling is the only effective way to handle large and complex designs.

During the SPADIC developments, tooling has of course always been a big issue (also relating to the analog part), but two topics probably dominated the tooling complexity: the mixed-signal simulation and the semi-custom design flow. Both had in common that little practical expertise has been available at the beginning of this work, whereas at the same time both applications have been crucial for the building, the testing, and the verification of various central parts of SPADIC. Consequently, an initial part of this work has been to learn how to use those two (and other similar) tools and, in particular, how to develop or improve proper script collections. The latter is very important, since scripts play a major role in the context of tooling in general. They flexibly allow to almost control all software components and once written serve as a recipe for the typically complex design processes.

But the topic of tooling should not be too much expanded here, instead subsequently only some short comments on mixed-signal simulation and semi-custom design flow are given. However, it is important to note, that the lengths of the subsequent comments yet stay in no relation to the actually associated amount of engineering work. In fact, using and adjusting the scripts, running the tools, gathering expertise, and interpreting the results was a big and intensive part of this work.

Mixed-signal transient simulation, hence the temporal simulation of digital logic based on HDL in parallel to analog circuits based on physical models, requires two different simulator worlds to interact properly. So-called interface elements, which can be adjusted at will, are used to translate the continuous analog signals into discrete digital logic levels and vice versa. Because both simulator types are internally event-driven, they both must listen to events that are propagated via the interface elements. As a direct consequence, the internal time-steps have to stay synchronized, which usually slows down the whole process and makes the whole simulation only as fast as the slowest component. Mixed-mode transient simulations for example were excessively used for the ADC design, where the analog parts have to properly interact with the digital evaluation logic and the control part. At the beginning of this work running the mixed-signal simulation was a real challenge, although today all that is required to start a new mixed-signal simulation is to properly adapt some few script files. The reason for that is not only the meanwhile gathered experience and the availability of script collections, but also the significantly improved usability of the CAD tools and the available documentation.

The semi-custom design flow, hence the whole work flow from the initial HDL code to the finally placed and routed standard-cell design, is likewise the mixed-signal simulation a sophisticated application. But in contrast to mixed-signal simulation, performing a semi-custom design flow does practically not only require some initial run-script modifications, but rather a complete and interactive script management. To be something more specific, for example a detailed floor-planning, numberless signal and timing constraints, many technology and library adjustments, countless optimization steps and parameters, or various layout guidelines must be set. An indeed, in order to synthesize the large digital part of SPADIC 1.0, an already available script collection has been completely restructured, adapted to the specific design needs, and cleaned up such, that it eventually could be run

autonomously without a single user interaction (the final SPADIC 1.0 run took more than 5.5 hours on the fastest available server). In particular, a big challenge has been the embedding of the 44 SRAM blocks, which did not only make a manual placement (and a lot of time consuming trial and error steps) necessary, but also demanded numberless script adjustments at completely different places. The final SPADIC 1.0 script collection is designed such that it can now be easily adjusted to the respective needs of the next SPADIC iteration(s).

CHAPTER 7

The Readout Systems

This chapter briefly describes the two most important SPADIC readout systems of the ten different setups (for the 6 different SPADIC iterations) that have been built so far (for an overview see section 4.4): First, the latest SPADIC 0.3 8-channel readout system, which was and still is frequently used by different detector groups, for instance to read out CBM-TRD prototypes during beam-times or in the laboratory [24] [7]. And, second, the latest SPADIC 1.0 32-channel readout system, which shall soon replace the well-established SPADIC 0.3 setup.

Besides the mere description of the two setups, some important front-end PCB design aspects are summarized at the end of this chapter.

7.1. The Latest SPADIC 0.3 Readout System

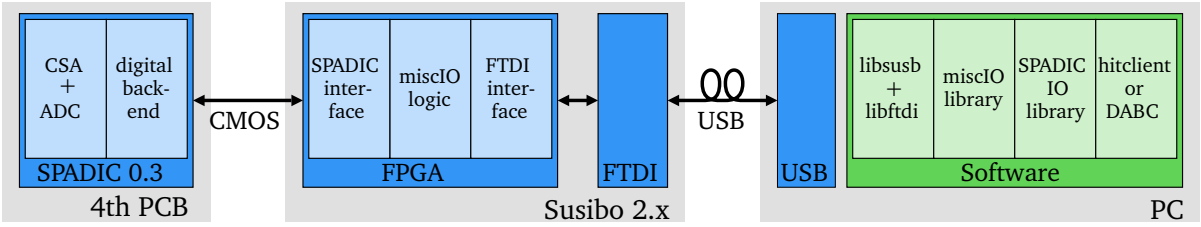


Figure 7.1.: Block diagram of the latest SPADIC 0.3 readout system.

As sketched in Fig. 7.1, the latest SPADIC 0.3 readout system [13] consists of the ASIC itself, of the forth iteration of the front-end PCB onto which the die is directly wire-bonded, of the home-made general purpose FPGA readout board called “Susibo” that is connected via USB to a PC, of the latest SPADIC FPGA firmware, of the SPADIC IO software library (based on libftdi and libusb), and finally of either the stand-alone SPADIC analysis software “hitclient” or the CBM/GSI readout framework “DABC/Go4” [61].

It is very important to note here that the previous versions of the front-end PCBs (especially the third SPADIC 0.3 setup which was the first connected to a detector) had very crucial problems with oscillations or pickup noise, as soon as some TRD prototype or even only a TRD pad plane was connected and grounded. But a complete redesign of the PCB (as further described in section 7.3.1) finally led to the very stable forth iteration, which is subsequently shown.

7.1.1. Digital Back End of SPADIC 0.3

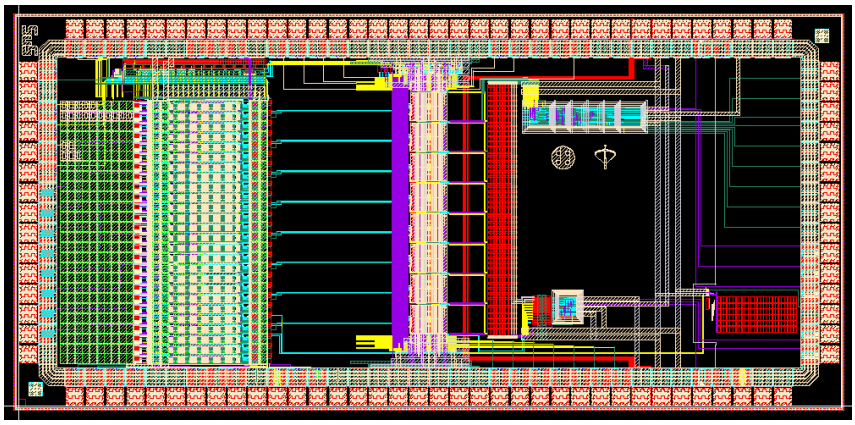


Figure 7.2.: Layout of SPADIC 0.3. The die has a size of 1.5 mm x 3.0 mm but is not completely filled. On the left hand side are the 26 CSAs channels, followed by a column of 8 ADCs in the middle. The shift register matrix is placed next to the ADCs (red), which is controlled by a standard cell block (bottom right) and read out via another digital block (top right).

As mentioned before, SPADIC 0.3 (compare Fig. 7.2) has 26 analog channels (CSAs with positive polarity followed by an analog discriminator) of which only 8 are connected both to an analog input pad and a pipeline ADC (25 MS/s, 8 bit effective). Therefore, the setup is only an 8-channel readout system. Although the analog channels (CSAs and ADCs) are conceptually very similar to the analog part of SPADIC 1.0[1], the digital part of SPADIC 0.3 is very simple and can not be compared with the one of SPADIC 1.0. In detail, to read out SPADIC 0.3 the 8 raw data streams coming from the 8 ADCs (the values are still in redundant signed binary 16 bit representation at this point) are continuously shifted into a large shift register matrix (total size is roughly 6 kbit, storage nodes similar to DRAM cells, laid out by hand, total layout size 90 µm x 1 mm), which temporarily stores the last 45 ADC values of each stream. If an externally or internally generated (by a discriminator) analog trigger signal occurs, the whole shift register matrix freezes, switches into a serial readout mode, and shifts all 360 ($8 \cdot 45$) ADC values out and further through the digital output logic. The output logic converts the incoming 16 bit raw data values into 9 bit 2's complement and further passes the result out of the chip (single ended CMOS output).

[1] But for numberless minor schematic adjustments, two CSAs instead of one, a completely redesigned layout, et cetera.

The setup can be either operated in a force-trigger or in a self-triggered mode. The latter is done by looping back the output signal of one selected discriminator to the force-trigger input. Therefore the SPADIC 0.3 in self-triggered mode can only monitor pulses of one single channel at a time. Moreover, due to the readout methodology that has just been described, each single readout causes long dead-times (roughly 16 µs) and a huge data overhead, since each time the chip is triggered a whole data package (3240 bit) is shifted out.

7.1.2. FPGA Firmware

The firmware running on the FPGA basically performs three tasks:

1. The firmware has to transport and translate control messages from the PC to the ASIC (or to the firmware itself) and vice versa. That way, mostly the analog configuration (SPADIC 0.3 has no digital configuration) is set, but also different status and monitor registers of the firmware itself can be read or written.

2. Moreover, the firmware must continuously listen to the SPADIC 0.3 data output interface and immediately transport all occurring data packages (360 values a 9 bit) to the PC.

3. And third, the firmware must handle the two different trigger modes: Either the output signal of a selected analog discriminator must be sampled and properly looped back to the force trigger input of the chip (the self-triggered mode mentioned before) or the so-called SYNCH LVDS trigger signal (coming from some GSI trigger unit) must be (over-)sampled (no reference clock is available), properly interpreted[1], and finally passed to the force-trigger input of SPADIC 0.3. The latter mechanism is required to synchronize several SPADIC boards as well as other front-end setups (e.g. during beam-times).

To manage the low level control and data message transport, the firmware uses the so-called “miscIO” building block (in combination with a corresponding software library), which was initially written for SPADIC 0.3. MiscIO is of such a general nature though, that it is not only used in the latest SPADIC 1.0 setup as well, but also in several other prototype projects that are based on the Susibo FPGA board. For that reason, the miscIO logic and the relating software library are further described in a dedicated section (7.3.2).

7.1.3. Software

The two available user programs, the hitclient and the DABC/Go4 plugin, include the SPADIC IO library. Whereas the low level FTDI communication is handled by the miscIO library (section 7.3.2) and the underlying system libraries libusb and libftdi, the SPADIC IO library provides various high-level control methods, such as connect/disconnect a certain Susibo

[1] The fist falling edge sets the trigger time, then an event ID and some check-bits are coded on the 1-wire LVDS signal. The extracted information is attached to the produced 360 value packages.

(having a certain serial number), write the SPADIC configuration, or access status registers. Moreover, the library provides thread-save access to an internal FIFO memory, where all new arriving data packages are automatically and autonomously (a dedicated readout thread runs in the background) buffered.

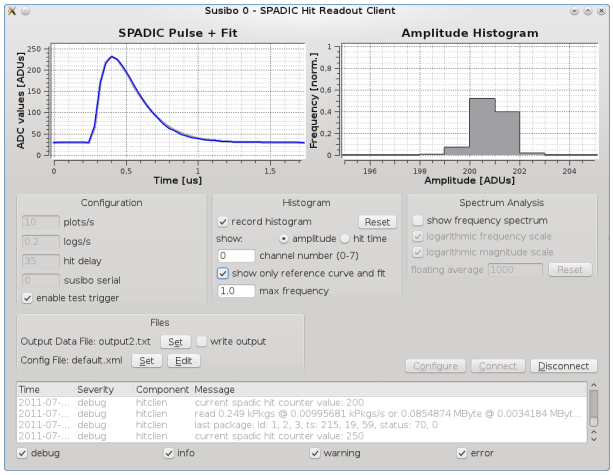


Figure 7.3.: Screenshot of the hitclient while monitoring SPADIC 0.3. The blue pulse in the upper left window is a live signal digitized and recorded in channel 0 (oscilloscope-like behavior). The upper right window shows an amplitude histogram extracted from the online pulse fits. Some videos showing the hitclient in action are available on the website [11].

The main purpose of the hitclient – a screenshot is shown in Fig. 7.3 – is to have a stand-alone and easy-to-use analysis and configuration tool. The main features are online monitoring of the recorded pulses (oscilloscope-like behavior), the online generation and recording of amplitude or time histograms (extracted from pulse fits), the SPADIC 0.3 configuration GUI, the possibility to stream the recorded data to a previously defined file, and the online frequency spectrum plotter (uses FFT, e.g. to analyze oscillations). The hitclient is intended for smaller applications, such as lab setups or quick diagnoses (e.g. in the cave during beam-times with the laptop), and provides a simple but reliable access a single setup. In fact, the hitclient is frequently used by the TRD (and other) groups to read out prototypes in the laboratory.

The DABC/Go4 software, compare Fig. 7.4, is a large DAQ framework developed at GSI to readout and coordinate several sub-systems simultaneously. Whereas the DABC contains all the DAQ libraries and tools (e.g. network protocols, data structures, synchronization methods, or module drivers), Go4 is basically an online monitoring and data analysis GUI. In order to access the SPADIC setup (via USB), the DABC/Go4 also includes the SPADIC IO library and accesses the available high-level functions. Because DABC/Go4 only supports the reading (and the interpretation) of SPADIC 0.3 data streams so far, the configuration of the setups must be carried out using the hitclient beforehand. DABC/Go4 is not only designed to read out the front-end setups, but also to record and store whole events, to synchronize whole detector systems, to provide numberless libraries for data analysis, and

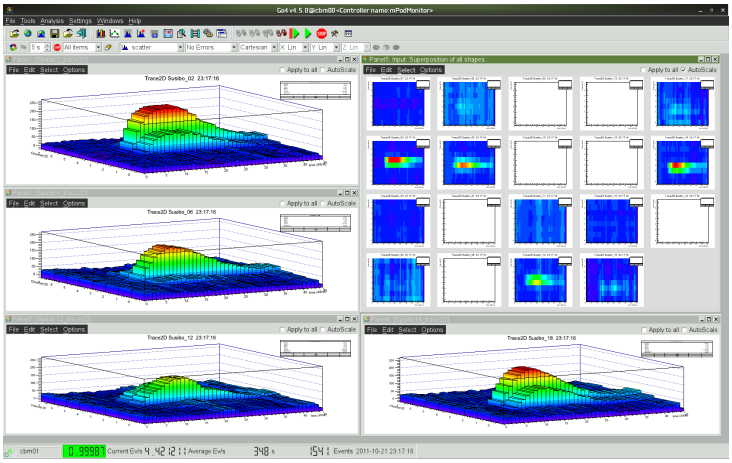


Figure 7.4.: Screenshot of the DABC/Go4 framework. One can see the recordings (3D diagrams) of a single particle via 4 different SPADIC 0.3 setups (or 4 different detector layers) that have been produced during the CERN beam-time in 2011. Each 3D diagram shows the recorded pulses as a function of time and as a function of the channel number. One can clearly see that the induced charge signal is mostly spread over several neighbor pads/channels. In the upper right edge the same event but in 2D representation is shown for all 12 setups that have been operated in parallel.

to properly store large amounts of data – especially during beam-times. Hence DABC/Go4 has a completely different field of operation than the simple stand-alone hitclient.

7.1.4. Selected Results

The latest SPADIC 0.3 setup is mainly used by the CBM-TRD groups from Münster and Frankfurt, but also several other groups and even non-CBM members made their first experiences with SPADIC 0.3 and/or are waiting for SPADIC 1.0 to become publicly available (e.g. groups from Dubna, Mainz, and Münster).

Until now the setup has been used during three beam-times at T9 PS/CERN in 2010, 2011, and 2012. The photo in Fig. 7.5 exemplarily shows a part of the beam-line of 2011 (1 - 10 GeV/c negative electron/pion beams). Shown are at least 8 of the 12 SPADIC 0.3 setups connected to different TRD prototypes that have been operated in parallel (the beam went from right to left). A total of 8 chambers have been tested, 4 chambers with 2 setups each (16 channels per chamber) and 4 chambers with only 1 setup each have been attached.

As an example of a result from the laboratory, a Fe-55 spectrum measured by the TRD group from Frankfurt using the SPADIC 0.3 setup connected to a MWPC/TRD prototype, is shown in Fig. 7.6. On the right hand side of the plot is the 5.9 keV iron peak[1], whereas on the left hand side, at 2.9 keV, one clearly sees the so-called argon escape peak.

[1]The resolution here is solely limited by the detector, not by the ASIC.

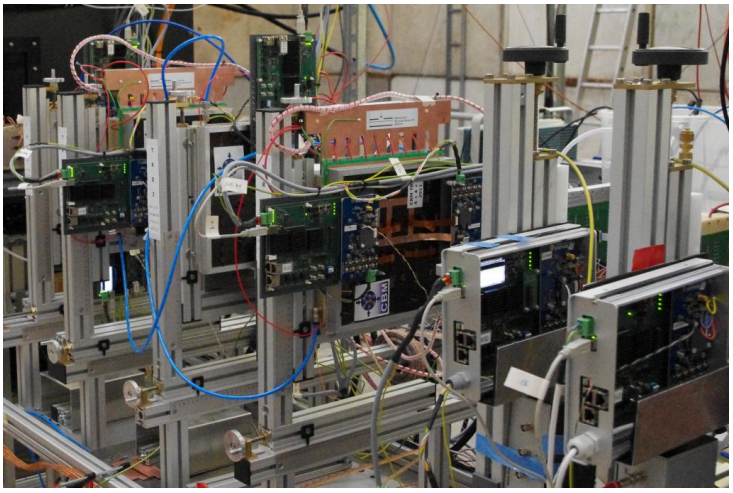


Figure 7.5.: TRD/SPADIC test-beam setup at T9 PS/CERN 2011.

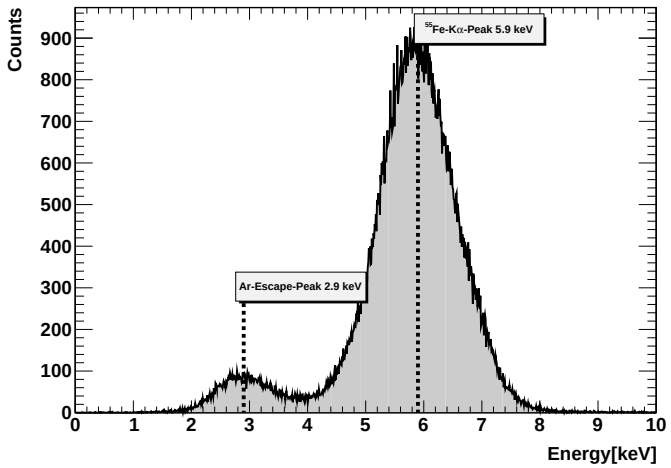


Figure 7.6.: Fe-55 spectrum detected with a CBM TRD prototype and read out with the SPADIC 0.3 setup in the laboratory [31].

7.2. The Latest SPADIC 1.0 Readout System

As shown in Fig. 7.7, the latest SPADIC 1.0 front-end is separated into two parts, a carrier board, which only carries the die that is directly wire-bonded to the PCB as well as some decoupling capacitors, and the main board, which basically provides different power domains, connectors, decoupling, and various helper and test structures. The PCB has been separated into two parts in order to gain flexibility (easier handling during assembly and bonding, reusability of the main board, ...) and to reduce the costs (very small structures and a gold plated surface are only required in the direct periphery of the die).

Presently, two different hardware setups are available: The first is a stand-alone system which is very similar to the SPADIC 0.3 system previously described (front-end PCBs,

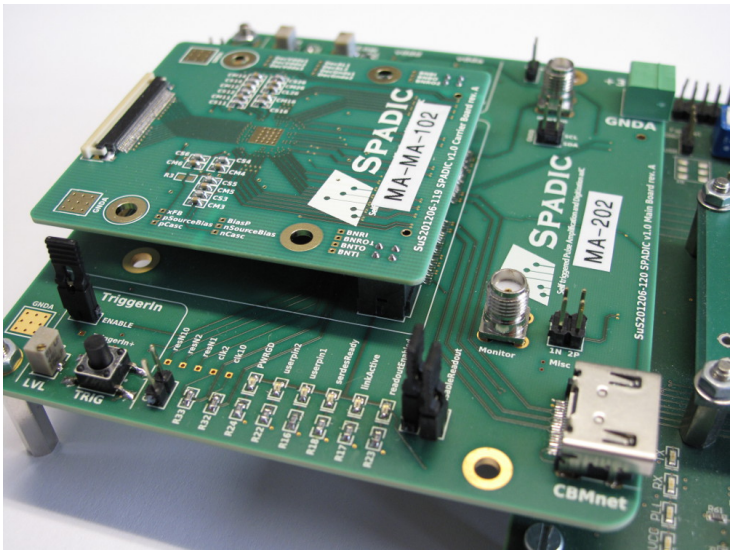


Figure 7.7.: Close up of the latest SPADIC 1.0 front-end PCB.

Susibo, and PC), whereas the second is a complete DAQ system with some intermediate nodes communicating via CBMnet (several front-ends are combinable at once, small network tree with FPGA nodes, partly optical interconnects, PCIe adapter to the PC, etc.). Both setup options are now further described.

7.2.1. Option 1: Stand-Alone Readout System

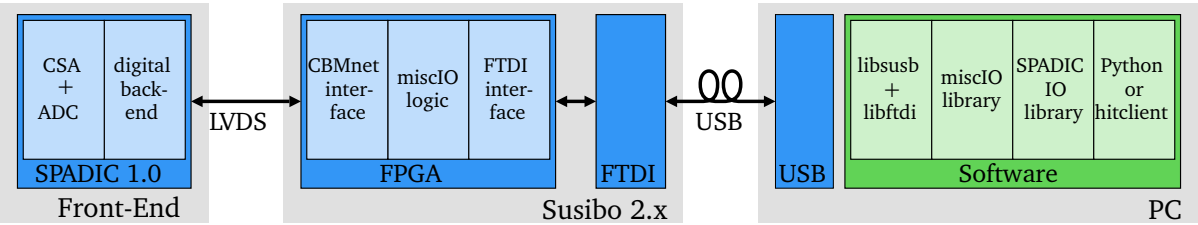


Figure 7.8.: Block diagram of the latest SPADIC 1.0 stand-alone readout system.

As shown in Fig. 7.8, the stand-alone SPADIC 1.0 readout chain is conceptually similar to the latest SPADIC 0.3 readout chain, although it is very different in detail. The two main differences from an abstract point of view are the new protocol interface between ASIC and FPGA (Susibo) and the completely newly implemented end-user software. The communication between ASIC and FPGA can be either done via the test data input and output interfaces in combination with the I^2C register file port (compare section 6.5.3) or exclusively via CBMnet. The second method is preferred, not at least because many features (e.g. several trigger mechanisms, epoch marker production, etc.) can not be used if the test interfaces are operated. Moreover, two different software options are available: first, an interactive Python script collection, which is best suited for chip measurements

and data analysis. And, second, a completely re-written hitclient, which is – similar to its older relative – intended to be a simple but reliable online monitoring, diagnosis, and data storage tool.

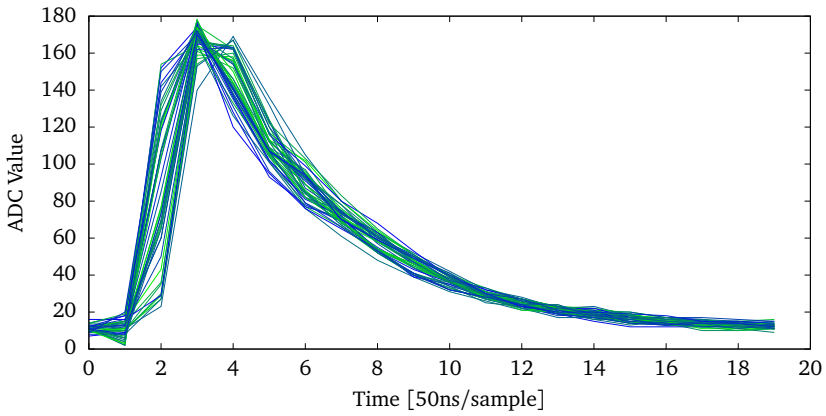


Figure 7.9.: Preliminary measurement with the stand-alone SPADIC 1.0 setup. The graph shows 44 superimposed pulses that have been analogously injected into the CSAs via a injection circuit that is implemented directly on the front-end PCB.

As an example of a first measurement with the stand-alone SPADIC 1.0 setup, 44 recorded pulses (at 20 MS/s) are shown in Fig. 7.9. For the measurement a charge pulse has been injected into the preamplifier of one dedicated channel (a pulse step generator is directly implemented on the front-end PCB, a dead-end wire is routed beneath the CSA input wires, the charge is injected through the layer-layer capacity of the PCB). The horizontal scattering of the recorded pulses in the figure simply comes from the fact, that the initial trigger is uncorrelated to the sampling clock (and visualizes why a fit can significantly improve the effective time resolution), whereas the vertical variation (best visible at the tail of the pulses) is due to electronic noise.

Even though the latter result might not seem to be very complicated, the measurement nearly proves the whole mixed-signal ASIC to work properly (except for the CBMnet, which was bypassed in this particular case). And indeed, in order to run the measurement nearly all analog and digital parts have been biased and clocked and hence all crucial chip parts have been operated (CSAs, ADCs, IIR filter, hit logic, message builder, channel arbiter, FIFOs, etc.) – and the same is true for the whole readout system as well as nearly all firmware and software components.

7.2.2. Option 2: CBM DAQ Readout System

The second readout option, which except for the SPADIC front-end PCB is being developed by several CBM groups (from Darmstadt, Frankfurt, and Heidelberg) uses the same SPADIC PCBs (carrier and main board), but no Susibo. Instead, as sketched in Fig. 7.10, the data is transported over an HDMI cable via CBMnet to a commercial FPGA board (Xilinx Spartan-6 FPGA SP605 Evaluation Kit) and from there further via a fiber optics cable (again via CBMnet) to a second FPGA board (Xilinx Virtex-5 LXT PCIe Development Kit). The latter is

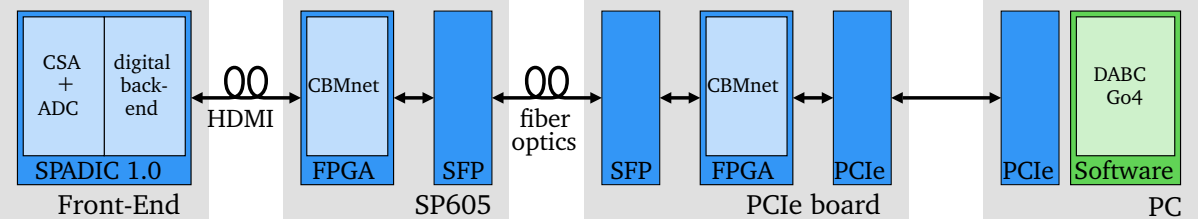


Figure 7.10.: Block diagram of the latest SPADIC 1.0 CBM DAQ readout system.

directly plugged into a PCI express slot of a PC. The high-level software used with this setup is again DABC/Go4, which inherently supports CBMnet, but of course internally requires a dedicated SPADIC message library.

Because each SP605 board provides 4 HDMI connectors and each PCIe board has two optical transceivers (SFPs), it should soon[1] be possible to connect 8 SPADIC 1.0 front-ends via a small CBMnet network tree to one PCIe PC slot.

7.3. Selected System Aspects

Some also important system aspects, that are of a more general nature though, are listed below.

7.3.1. Front-End PCB Design Suggestions

When a MWPC was connected to the 3rd SPADIC 0.3 setup for the first time, an important milestone has been reached, since it was the first time, that any SPADIC setup was used to record and monitor "real" chamber signals. Unfortunately, the high quality of the injected pulses that were recorded in the laboratory without a detector before, decreased significantly as soon as a detector was connected to the preamplifiers[2]. Besides a significantly increased noise component of the recorded pulses, which can not be simply explained with the additional load capacity introduced by the detector, an extremely unstable baseline (low-frequency oscillations) and a very high sensitivity to external disturbing sources was observed. Consequently, a lot effort was made to investigate the problem, especially during the first CERN beam-time in 2010. In fact, the main task was to find a proper grounding and shielding scheme, which would make it possible to record anything at all. Finally, during the beam-time in 2010, a relatively stable setup, that delivered usable but still low quality data, could be built.

To further investigate the problem after the test-beam in 2010, the 4th PCB iteration (the 4th and last SPADIC 0.3 setup) has been designed – with the mere goal to improve the overall readout quality and stability, especially if a detector is attached. In a private meeting with a PCB specialist from GSI [40] several new design techniques could be gathered, of

[1] Proper CBMnet firmware is not yet available.

[2] Later, it became evident that even the connection to a mere detector pad-plane without any further components would cause similar problems.

which most were actually realized in the 4th setup. And indeed, the various implemented changes led to amazing results, as it is exemplarily shown in Fig. 7.11.

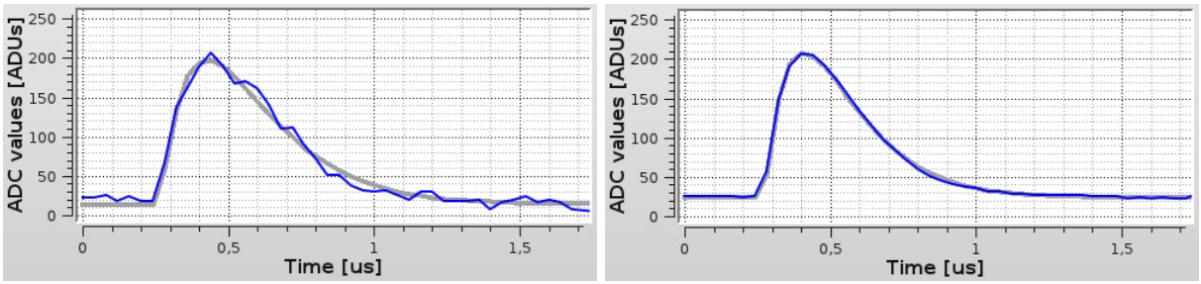


Figure 7.11.: Test pulses recorded with the 3rd SPADIC 0.3 setup (left) and the 4th setup (right) – while a detector is connected. Both pictures are online screenshots of the hitclient. Colored blue are the actually recorded pulses, whereas the fitted pulse functions are gray. Much more impressive than these pictures are the videos available on the SPADIC website [11].

After that success a list of design suggestions was derived from the initial design proposals for the 4th SPADIC 0.3 PCB. Later, those design suggestions were not only considered when designing the different SPADIC 1.0 PCBs, but also when developing the SPADIC 1.0 chip (mainly the analog layout and the pad placement were adjusted). But due to the relatively high effort that would be necessary to further investigate the different suggestions in detail, they have not been individually proved so far.

Subsequently the derived design suggestions are listed.

Suggestion 1 – Central Ground Position: The idea is to have a well-defined position on the PCB where all ground nodes of all power domains run together and where at the same time the ground cable of the detector is connected. The central ground reference is the absolute reference for all signals – practically and in theory – and especially for the preamplifiers in the present case. The existence of a central ground reference should be always kept in mind during all design phases. Because the central ground reference is a certain position on the PCB and not a global net, one should in general rather think of ground (or power) wires that connect ground (or power) electrodes (pins, connectors, ...) than of global ground layers that simply connect everything. In the SPADIC designs undirected ground layers have been avoided as much as possible – especially in noise critical areas (e.g. next to the preamplifier inputs). Moreover, part of the concept is to define a separate ground net for each power domain, and – as said before – to short the different ground domains only at the central ground position.

Technical Motivation: The central ground position helps to avoid “ground loops” or – as it should be correctly called – common return paths. In principle, static as well as dynamic cross-domain currents shall be avoided as much as possible. Isolating the grounds from each other effectively means shielding them from each other.

Suggestion 2 – Voltage Regulators: Each power/ground pair should be powered by a separate voltage regulator. The grounds of the different voltage regulators must not be shorted directly, but eventually run together at the central ground position. Besides the

usual decoupling capacitors, so-called X2Y filters can help to filter the initial (coming from the power supply) voltage signals powering the voltage regulators. Of course, low noise voltage regulators should be favored.

Technical Motivation: The voltage regulators are reliable power supplies (no long cables to some unknown power supply, stable and well-known setup conditions) and help to reduce the number of power cables and connectors (and thus the number of possible disturbing sources). Moreover, the voltage regulators on the PCB actually shield the power and ground domains from each other.

Suggestion 3 – Current Flow Control: The goal is the same as in the case of the central ground position: in general unnecessary or direct paths between foreign nodes shall be avoided. Theoretically – which can not be realized in practice though – the best case would be to have one separate wire for each connection between any power/ground electrode and the respective voltage regulator. Whereas the central ground concept is an overall strategy, current flow control should be practically considered whenever a new device is placed or a new wire is routed. Especially in the periphery of the die it is never a good idea to simply short power, ground, or bias signals only because they have the same potential. Controlling the current flow here means to route the power and ground wires such that they do not simply connect all common power or ground pins at once (e.g. via a power plane), but to lead the expected currents along intended paths (like streets lead the traffic).

Technical Motivation: Again common return paths or cross paths shall be avoided. If for instance the current demand of some power electrode suddenly increases and hence the electrode voltage drops slightly, the deficit current should be taken from some nearby decoupling capacity or directly from the power source and not from any neighboring power electrode.

Suggestion 4 – Proportional Decoupling: Of course, the technique to decouple power and bias nets with different capacitor sizes and hence different time constants is common practice. But because obviously the effective time constant introduced by a certain capacitor is the sum of all associated parasitic time constants (mainly due to the PCB wire resistances and capacities) and the intrinsic time-constant of the capacitor itself (due to the input resistance), the distance between capacitor and respective electrode (as well as the routing details) should also be taken into account. In particular, it is a good strategy to place small capacitors nearby the associated pins, medium capacitors at medium distance, and large capacitors at higher distances (e.g. next to the voltage regulators).

Technical Motivation: Different decoupling time constants allow the decoupling network to react to disturbances on different timescales. Moreover, the proportional decoupling is a simple trick to get a good balance between the size of the capacitor (this time in terms of area not capacity) and the required time constant and therefore serves as an effective placement strategy.

Suggestion 5 – Considering the Fooprint: The goal is simple: the shorter the bonding wires (or the connections in general) the better. But various geometrical design constraints coming from the PCB and the ASIC, technical bonding limits (e.g. maximum angles or equal wire lengths), and potentially a large number of pins might make the actual design geometrically difficult. And moreover, also many electrical constraints usually exist, like for instance the need to separate the power domains, to isolate digital and analog nets,

et cetera – and also the previous suggestions can lead to various additional electrical requirements. Therefore designing the bonding footprint can be a crucial task that should be carefully performed and that, moreover, should be already considered when developing the ASIC (e.g. when the pad positions or the overall layout is set). But because the various details strongly depend on the respective case, no general suggestions are given here. The intention rather is to emphasize that the final footprint is a very important design aspect that should be kept in mind already during the early design phase of the ASIC.

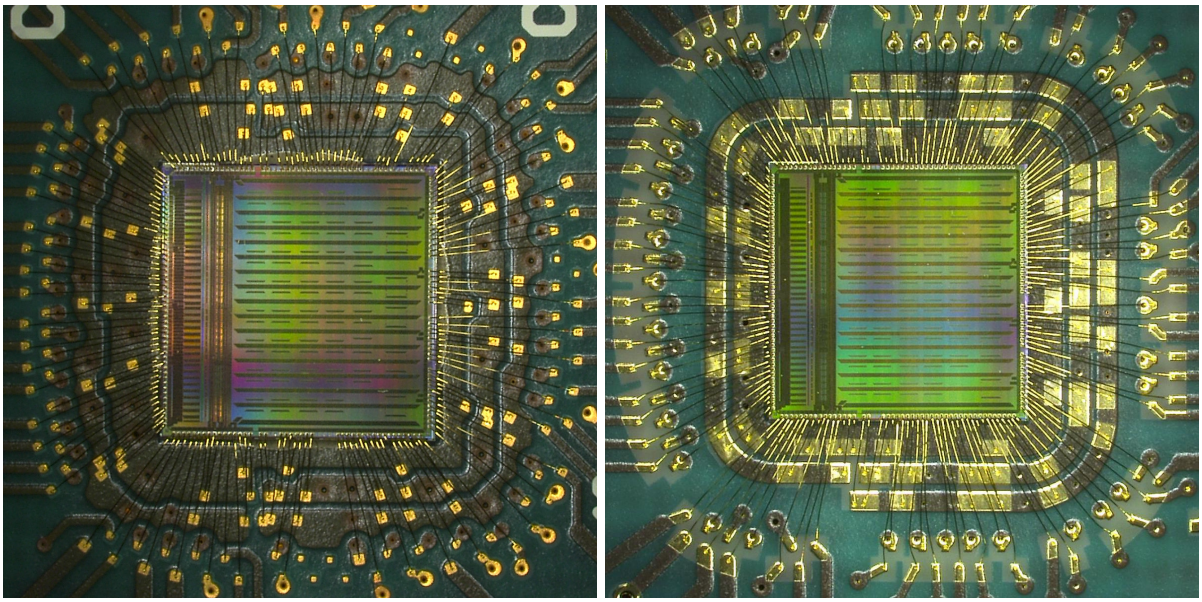


Figure 7.12.: SPADIC 1.0 wire-bonded to the carrier PCB. Left: the first PCB version – the footprint was mainly manually designed. Right: the second PCB, here the footprint was generated nearly automatically (after proper scripts have been written).

Technical Motivation: The goal is to consider all of the previous suggestions also when developing the footprint or the initial pad-placement of the chip. To show how it has been done in the case of SPADIC, the two footprints of the first two SPADIC 1.0 PCBs are shown in Fig. 7.12. As it can be clearly seen, fractions of power rings for the different power and ground domains were routed around the die. The power rings are further encapsulated by two staggered rows of signal pads. That way, the number of long wires is reduced, which effectively simplifies the overall fan-out of the remaining signal wires. Moreover, the power and ground wires could be kept nearly as short as possible, the average wire length stays relatively moderate, and the PCBs can be fabricated with standard design parameters (which significantly saves costs). In the present case it was already assured during the ASIC design phase, that no single analog signal must be routed close to a digital signal: for instance, the sensitive preamplifier inputs are on the left hand side of the die, whereas the fast LVDS IO signals (CBMnet) are on the right hand side of the die – separated as much as possible.

7.3.2. MiscIO

As described earlier, the so-called miscIO firmware (written in Verilog) and the miscIO software (C++ code) have been written to get a simple and easy to use communication interface for lab setups that are based on the Susibo FPGA board. Even though the simple concept is of course transferable to other FPGA boards as well. Since miscIO has been designed to communicate over the FTDI/USB interface, the firmware internally uses an FTDI interface wrapper and the software is based on libusb/libftdi (Unix). Due to the 8 bit granularity of USB, the internal word granularity of the miscIO interfaces was also set to 8 bit.

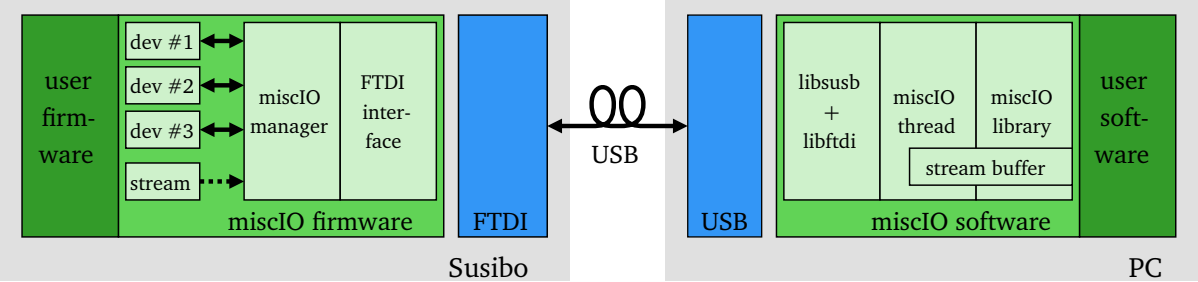


Figure 7.13.: Block diagram of miscIO.

The principle of miscIO is simple (summarized in Fig. 7.13): On the firmware side, the user can define passive device interfaces (here simply called devices), each having a unique 8 bit address and an 8 bit internal word width. Via read and write commands (or better get and push commands) on the software side, the devices can be individually accessed. To be more specific, to write means that a certain number of 8 bit words is automatically transferred from the PC to the addressed device, while to read means that a defined number of 8 bit values is requested from the addressed device and transported back to the PC[1].

Moreover and in addition to the normal devices, a dedicated data stream port is available. It is only unidirectional (from FPGA to PC), but can be operated autonomously. The stream port allows a selected device to actively and autonomously sent complete data packages, which means in particular that no explicit read request of the software side must be started. In order to assure the concurrent operation of the stream port, a read-thread runs independently of any other software commands on the software side. In detail, the thread temporarily stores all arriving stream packages in a thread-safe buffer, which can be accessed by the user at will.

In addition to the three user commands “read from device”, “write to device” and “get new package data”, the miscIO software automatically creates and handles the internal IO-thread, provides convenient connect and disconnect methods (which optionally allow to select a certain Susibo via its respective serial number, for instance if more than one Susibos are connected to the same PC), and provides several helper functions. Moreover, it automatically checks if a compatible Susibo firmware is available on the connected FPGA and manages all kinds of low-level initialization and communication steps.

[1]It must always be assured by the user that the addressed device can actually provide the requested number of values. Otherwise a deadlock will occur.

MiscIO can be used of the shelf, meaning that all the user has to do is to create the required device interfaces, add the miscIO firmware building block to the existing firmware, and include the miscIO library into the C++ project. In doing so, a first communication between PC and FPGA can be quickly and reliably established.

CHAPTER 8

Summary and Outlook

This work comprises the whole journey from the initial and vague requirement of a dedicated readout chip for some CBM sub-detector(s) to a comprehensive and flexible readout system for the TRD sub-detector. The obtained result consists of the full development of the SPADIC readout concept with all its theoretical and technical details and of the realization of 6 SPADIC versions and 10 system prototypes, that, moreover, have eventually all successfully been operated and characterized.

The numberless technical and practical steps that had to be taken for that purpose and the countless lines of code of software, firmware, and hardware, that had to be written are not further summarized here. Instead it shall be emphasized, that the resulting chip design is not simply an arbitrarily chosen solution, but effectively the result of a close and long lasting collaboration with the detector physicists on the one hand and the DAQ engineers on the other hand. In fact, all crucial design decisions that have been made during the years and that have finally led to the present SPADIC design concept, were all based either on intense and perpetual discussions or on well-founded experience, which has been gathered basically from simulations, observations, and measurements. A basic principle of this work has always been to take nothing for granted and to leave nothing to chance.

Being the central result, the whole SPADIC 1.0 prototype does not only prove the overall system concept to be suitable for the CBM-TRD application, but has also shown all used chip components as well as the whole mixed-signal architecture to work properly. Due to the very flexible and versatile architecture, the SPADIC chip can be directly used or at least easily adapted for other applications as well. That is especially true, since all functionalities have been clearly separated and all building blocks have been modularly designed – on all levels of hierarchy.

SPADIC 1.0 is a free-running multi-channel mixed-signal readout system on a chip. It combines high sensitivity amplification with flexible data handling and processing on one single die. It is easily accessible and flexibly adjustable. The oscilloscope-like behavior offers various new opportunities and allows to comfortably monitor and analyze all kinds

of short detector charge pulses. Whereas most of the characteristic numbers rather lie in a moderate range (e.g. the time and amplitude resolution or the power consumption), the main qualities of the SPADIC system are its compactness, its flexibility, and its convenience.

Nevertheless, SPADIC 1.0 is still not the end of the line. As mentioned in the respective chapters before, some uncritical but necessary bugs have to be fixed and some few but important tasks still have to be addressed. Especially an improved single event effect (SEE) tolerance of the digital part and a final analog adjustment of the front-end to the final TRD detector requirements are probably important steps that will have to be taken. And, moreover, already a new list with additional beneficial ideas exists. For all that reasons, at least one iteration after SPADIC 1.0 will have to be designed and tested. Furthermore, one must not forget that even though all crucial parts of SPADIC 1.0 have been successfully operated, the analog part has not been fully characterized yet. The results that are shown in this paper are mostly preliminary and especially the analog details of the CSA and the ADC still have to be extracted more carefully.

Another future issue concerning SPADIC 1.0 is the readout setup. Whereas, as mentioned earlier in this document, first real chamber signals could be read out with the latest available setup, many open questions still remain and a lot of effort will have to be taken to find proper answers. A very crucial task for instance will be to make the setup more reliable and stable. That includes the search for proper power-on cycles, stable grounding schemes, reliable reset strategies, et cetera. Moreover, another important task will probably be to build and operate a PCB that carries several SPADIC 1.0 dies close to each other (pitch $\approx 2\,\mathrm{cm}$). Such a relatively dense setup will most certainly be required to read out the innermost TRD pad arrays.

Because this work comprises the whole development from scratch of the SPADIC chips, the setups, and the concept, and in order to keep things simple, instead of a list of own performances the most important contributions to this work that have been made *by others* are subsequently listed:

- The core parts of the algorithmic ADC were available [55].
- The CBMnet logic and the register file used in SPADIC 1.0 were provided as Verilog code [45].
- The different current DACs already existed within the group and have only been adjusted to the needs of the respective SPADIC versions.
- Even though they have been completely revised for SPADIC 1.0, most of the IO cells were already available within the group.
- The IIR filter was the subject of a dedicated diploma thesis [42].
- The SRAMS used in SPADIC 1.0 are commercial building blocks.
- Most of the wire bonding was done by [26].
- Most of the “SPADIC 0.3, setup 4” and “SPADIC 1.0, setup 2” boards were assembled by the TRD group from Münster (e.g. [24]).

- The latest SPADIC 1.0 setup and most of the SPADIC 1.0 measurements were performed by [48].
- Several components of the numberless SPADIC software and firmware versions for the different setups were written by colleagues (e.g. the FTDI interface, the Logfile C++ library, or the libMemstruct C++ classes).
- The measurements using SPADIC 0.3 or SPADIC 1.0 connected to CBM TRD prototypes were performed by the TRD groups from Frankfurt and Münster (e.g. [24] or [7]).
- The PCB design suggestions (section 7.3.1) were made by [40].
- The development of the different Susibo FPGA board versions is a result of an internal cooperation.
- The digital standard cell library UCL was developed in collaboration with some colleagues.

APPENDIX A

Appendix

A.1. Spacial Resolution of a Sensor Array

The following calculations are based on considerations from [56].

A.1.1. Dependency of Spacial Resolution on Noise

In general, in a perfect system the initial position of a signal that spreads over a well-known sensor array can be exactly reconstructed mathematically by analyzing the gathered signal fractions. But in a realistic setup where different sources introduce uncertainties – for instance electronic noise or statistic fluctuations – the achievable spacial resolution is limited.

To analyze the best achievable spacial resolution for a given noise figure, a sensor array with N sensors placed at the positions $\vec{x}_i$ is now examined. It is assumed, that the i-th sensor has gathered the signal fraction S_i. Hence the total signal S can be easily calculated as $S = \sum_{i=1}^{N} S_i$. Using center of gravity and assuming a noise-free system, the exact signal position $\vec{x}$ can be reconstructed as

$$\vec{x} = \frac{\sum_{i=1}^{N} \vec{x}_i S_i}{\sum_{i=1}^{N} S_i}. \tag{A.1}$$

To simplify the representation without loss of generality, it is further assumed that the original signal S is normalized

$$S = \sum_{i=1}^{N} S_i = 1, \tag{A.2}$$

and moreover that the origin of the coordinate system is located exactly at the center of the sensor array

$$\sum_{i=1}^{N} \vec{x}_i = \vec{0}. \tag{A.3}$$

Considering the latter assumptions, the exactly reconstructed initial signal position becomes

$$\vec{x} = \sum_{i=1}^{N} \vec{x}_i S_i. \tag{A.4}$$

In a non-ideal system each signal fraction S_i is randomly modified by an additional noise component n_i. Hence the reconstructed position (using equations A.2 and A.4) instead calculates as

$$\vec{x}^{\text{noise}} = \frac{\sum_{i=1}^{N} \vec{x}_i (S_i + n_i)}{\sum_{i=1}^{N} (S_i + n_i)} = \frac{\sum_{i=1}^{N} \vec{x}_i S_i + \sum_{i=1}^{N} \vec{x}_i n_i}{\sum_{i=1}^{N} S_i + \sum_{i=1}^{N} n_i} = \frac{\vec{x} + \sum_{i=1}^{N} \vec{x}_i n_i}{1 + \sum_{i=1}^{N} n_i}. \tag{A.5}$$

With the help of the overall correspondence

$$\left(\sum_{k=0}^{\infty} c r^k = c + \sum_{k=0}^{\infty} c r^{k+1} = c + r \sum_{k=0}^{\infty} c r^k \Rightarrow \right) \quad \sum_{k=0}^{\infty} c r^k = \frac{c}{1-r}, \tag{A.6}$$

the denominator in equation A.5 can be expanded. That leads to

$$\begin{aligned} \vec{x}^{\text{noise}} &= \left(\vec{x} + \sum_{i=1}^{N} \vec{x}_i n_i \right) \left(1 - \sum_{i=1}^{N} n_i + \mathcal{O}(n_i^2) \right) \\ &= \vec{x} + \sum_{i=1}^{N} n_i (\vec{x}_i - \vec{x}) + \mathcal{O}(n_i^2). \end{aligned} \tag{A.7}$$

The error of reconstruction can be identified as $\vec{x}^{err} = \vec{x}^{\text{noise}} - \vec{x}$. In general, the corresponding variance can be calculated as

$$\sigma_{err}^2 = E[(\vec{x}^{err})^2] - E[\vec{x}^{err}]^2, \tag{A.8}$$

with E the expectation value.

Because the noise contributions n_i are from geometrically and electrically separated readout channels, they can be assumed to be uncorrelated. Moreover, because the error basically comes from different noise sources, which naturally have an average of zero and a symmetric probability distribution, the expectation value $E[\vec{x}^{err}]^2$ completely disappears. Therefore, the latter equation can be further reduced to

$$
\begin{aligned}
\sigma_{err}^2 &= E[(\vec{x}^{err})^2] = E[(\vec{x}^{\text{noise}} - \vec{x})^2] \\
&= E\left[\left(\sum_{i=1}^{N} n_i(\vec{x}_i - \vec{x}) + \mathcal{O}(n_i^2)\right)^2\right] \\
&= E\left[\left(\sum_{i=1}^{N} n_i(\vec{x}_i - \vec{x})\right)\left(\sum_{j=1}^{N} n_j(\vec{x}_j - \vec{x})\right) + \mathcal{O}(n_i^3)\right] \\
&= E\left[\sum_{i,j=1}^{N} n_i n_j(\vec{x}_i - \vec{x})(\vec{x}_j - \vec{x})\right] + E\left[\mathcal{O}(n_i^3)\right] \\
&= \sum_{i,j=1}^{N} E[n_i n_j] E[(\vec{x}_i - \vec{x})(\vec{x}_j - \vec{x})] + \mathcal{O}(\sigma_{\text{noise}}^3) \quad (n_i,\, n_j \text{ not correlated to } \vec{x}_i,\, \vec{x}_j,\, \vec{x}) \\
&= \sum_{i=1}^{N} E[n_i^2] E[(\vec{x}_i - \vec{x})^2] + \mathcal{O}(\sigma_{\text{noise}}^3) \quad (n_i,\, n_j \text{ uncorrelated} \Rightarrow E[n_i n_j] = 0 \text{ if } i \neq j) \\
&= \sum_{i=1}^{N} \sigma_{\text{noise}}^2 E[(\vec{x}_i^2 - 2\vec{x}_i\vec{x} + \vec{x}^2)] + \mathcal{O}(\sigma_{\text{noise}}^3) \\
&= \sigma_{\text{noise}}^2 \left(\sum_{i=1}^{N} \vec{x}_i^2 + N E[\vec{x}^2]\right) + \mathcal{O}(\sigma_{\text{noise}}^3) \quad \text{(with assumption A.3)}.
\end{aligned}
\tag{A.9}
$$

The result shows, that the best achievable spacial resolution σ_{err} is proportionally limited by the given noise figure σ_{noise}, whereas the constant of proportionality

$$
K = \sqrt{\sum_{i=1}^{N} \vec{x}_i^2 + N E[\vec{x}^2]} \tag{A.10}
$$

only depends on the respective geometry of the sensor array.

A.1.2. Case Study: TRD Strips

Considering TRD strips with a pitch a placed at the positions $p_i = ia$, and moreover assuming the charge distribution $Q^d(x)$ of the initial signal (punctual charge cloud) at d to be a simple box function that spreads over the range b ($Q^d(x) = \frac{1}{b}$ if $x \in [-\frac{b}{2} + d; \frac{b}{2} + d]$ and $Q^d(x) = 0$ else), the signal fraction S_i^d of pad i can be calculated as (assuming $b \geq a$)

$$
S_i^d = \int_{ia-\frac{a}{2}}^{ia+\frac{a}{2}} Q^d(x)\, \mathrm{d}x = \frac{1}{b} \begin{cases} (ia + \frac{a}{2}) - (d - \frac{b}{2}) & \text{if } d \in [ia - \frac{a}{2} + \frac{b}{2}; ia + \frac{a}{2} + \frac{b}{2}[\\ a & \text{if } d \in [ia + \frac{a}{2} - \frac{b}{2}; ia - \frac{a}{2} + \frac{b}{2}[\\ (d + \frac{b}{2}) - (ia - \frac{a}{2}) & \text{if } d \in [ia - \frac{a}{2} - \frac{b}{2}; ia + \frac{a}{2} - \frac{b}{2}] \\ 0 & \text{else} \end{cases}
\tag{A.11}
$$

With the substitution $b = Ma$ ($M \geq 1$), which indicates that the charge is distributed over M pad lengths, the latter equation can be slightly simplified to

$$S_i^d = \frac{1}{M} \begin{cases} \frac{1}{2}(1+M+2i) - \frac{d}{a} & \text{if } d \in [\frac{a}{2}(2i-1+M); \frac{a}{2}(2i+1+M)] \\ 1 & \text{if } d \in [\frac{a}{2}(2i+1-M); \frac{a}{2}(2i-1+M)[\\ \frac{1}{2}(1+M-2i) + \frac{d}{a} & \text{if } d \in [\frac{a}{2}(2i-1-M); \frac{a}{2}(2i+1-M)[\\ 0 & \text{else} \end{cases} \tag{A.12}$$

Because for a given signal at d only $M+1$ consecutive strips $i_0 \dots i_0 + M$ sense a signal fraction larger than zero (the probability to hit only M pads is zero) and moreover because only the two corner strips i_0 and $i_0 + M$ will sense a value unequal to $\frac{1}{M}$, the total gathered (ideal) signal S is always

$$\begin{aligned} S &= \sum_{i=-\infty}^{\infty} S_i^d = S_{i_0}^d + \sum_{i=i_0+1}^{i_0+M-1} S_i^d + S_{i_0+M}^d \\ &= \frac{1}{M}\left(\left(\frac{1}{2}(1+M+2i_0) - \frac{d}{a}\right) + (M-1) + \left(\frac{1}{2}(1+M-2(i_0+M)) + \frac{d}{a}\right)\right) \\ &= 1 \end{aligned} \tag{A.13}$$

and therefore normalized, fulfilling assumption A.2.

By shifting the pad positions to $p_i = a(i-1) - \frac{Ma}{2}$ and using the indexes $i = 1 \cdots M+1$ also assumption A.10 can be fulfilled[1] (due to the symmetry of the positions p_i with respect to the origin), both for odd and even values of M. Hence equation A.10 can be used to calculate the noise scaling factor K of a signal at position d that occurs within the range $[-\frac{a}{2}; \frac{a}{2}]$ (note again that if the charge is spread over Ma pad lengths $M+1$ pads are hit):

$$\begin{aligned} K(M) &\approx \left(\sum_{i=1}^{M+1} (a(i-1) - \frac{Ma}{2})^2 + (M+1)\int_{-\frac{a}{2}}^{\frac{a}{2}} d^2 \, \frac{1}{a} \, \mathrm{d}d\right)^{-\frac{1}{2}} \\ &= \left(\frac{a^2}{4}\sum_{i=0}^{M}(2i-M)^2 + \frac{M+1}{a}\left[\frac{d^3}{3}\right]_{-\frac{a}{2}}^{\frac{a}{2}}\right)^{-\frac{1}{2}} \\ &= \frac{a}{2}\left(\sum_{i=0}^{M}(2i-M)^2 + \frac{M+1}{3}\right)^{-\frac{1}{2}}. \end{aligned} \tag{A.14}$$

$K(M)/a$ is plotted in Fig. A.1. One can see a slight increase of $K(M)/a$, if M rises. That is not surprising, because in the case of a box distribution the reconstruction only depends on the two signal components gathered by the corner pads and, moreover, because an increase of M causes the signal fraction sensed by each individual strip (or the individual S/N fractions of the readout electrodes) to decrease.

[1] A shift of the origin does not affect the normalization.

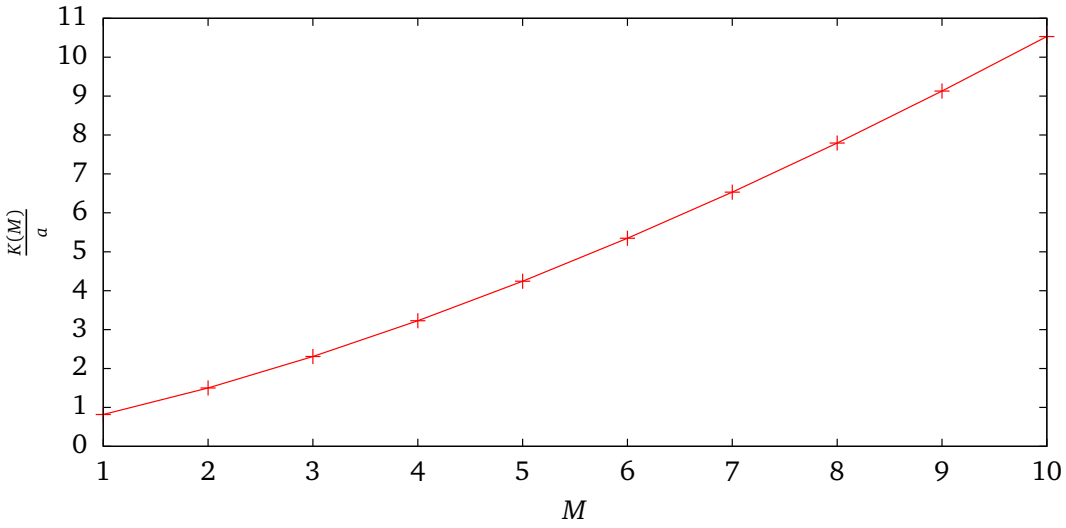


Figure A.1.: Calculated noise scaling factor $K(M)/a$ of strip pads and a box charge distribution spreading over $M+1$ pads. The result is based on equation A.10.

Finally, the best achievable spacial resolution in the present case is

$$\sigma_{err} = K(1)\,\sigma_{\text{noise}} = \frac{a}{2}\left(\sum_{i=0}^{1}(2i-1)^2+\frac{2}{3}\right)^{-\frac{1}{2}}\sigma_{\text{noise}} = \sqrt{\frac{2}{3}}\,a\,\sigma_{\text{noise}}. \tag{A.15}$$

A.2. Lemma

Lemma: It is impossible to find a set of real parameters a and b, such that the equation $(1-az^{-1})(1-bz^{-1})$ contains a complex conjugated pair.

Proof: If it was possible to find a proper parameter set, one would be able to bring the initial equation into the form $1+cz^{-2}$, with c a positive real number depending only on a and b. Rewriting leads to

$$(1-az^{-1})(1-bz^{-1}) = 1-az^{-1}-bz^{-1}+abz^{-2}, \tag{A.16}$$

which obviously demands $a=-b$ in order to get rid of the z^{-1}-terms. But then the equation becomes

$$1-a^2z^{-2}. \tag{A.17}$$

Hence the parameter c must be set to $-a^2$ – which is real but negative $\forall\ a$.

A.3. Data Wrapper Algorithm

To generate the states and the transitions required to build a data wrapper FSM (see section 6.3.7), a simple formalism was developed. The formalism is not very handsome and is just briefly described here, but it is qualified for quickly building a minimal data wrapper FSMs.

Subsequently, it is assumed to have a shift register structure as sketched in section 6.3.7. The words stored in this shift register are called w1, w2, w3, ... (w1 is the word shifted in last).

A state is written as a_b, while a symbolizes the number of valid (or yet unread) bits in the shift register (e.g. a = 93 means 3 bit of w1 and 9 bit of w2 are currently valid or unread), and b symbolizes the number of bits that are being read or selected via the multiplexer (e.g. b = 40 means that the 4 upper bit of w2 stored in the second stage are multiplexed to the output, for instance w2[8:5]). Instead, if a part of a or b is written bold (e.g. **5**00), the part relates to the lower bits instead (w3[4:0] and not w3[8:4] in the previous example). Moreover, if a state must be finalized (the *last* signal has occurred), it is additionally marked (e.g. 123_000′).

Example: Assuming 9 bit input words (w1, w2, ...). Then, the fictive state **3**99_**3**90′ for instance means: valid bits in the shift register are w3[2:0], w2[8:0] and w1[8:0], the multiplexer is set to w3[2:0], w2[8:0] and the state must be finalized, because the *last* signal has occurred sometime before (compare again 6.3.7).

Now, all that is required is a set of simple rules:

1. The initial (reset) state of the FSM is 0_0. One must start with this state.
2. *Take* (T) and *last* (L) can only occur in unmarked states with b = 0. All next states of the transitions T, L, and TL (*take* and *last*) must be calculated, if the present state is unmarked and has b = 0.
3. *Take* appends a new word to a. In the same step, the multiplexer symbolized with b must select as much words as possible as soon as enough bits (which depends on the required output word width) are available (e.g. **3**99_0 + T ⇒ **3**999_**3**960).
4. *Last* marks a state and forces the multiplexer to select as much bits as possible (e.g. **3**9_0 + L ⇒ **3**9_**3**9′).
5. States with b ≠ 0 always (default, D) pass over to a state where the previously multiplexed bits b are subtracted from the initially available bits a. If the initial state with b ≠ 0 is additionally marked, the next state is also marked and again as much bits as possible must be selected, or if not, b of the next state is zero (e.g. 222_121 ⇒ 101_000 or 222_121′ ⇒ 101_101′).
6. Marked states with a = b can be unmarked (e.g. 0_0′ = 0_0 or 1**2**34_1**2**34′ = 1**2**34_1**2**34).

As an explanation and example, a data wrapper FSM translating from 9 bit to 15 bit (which is similar to the implemented FSM but for the special case that the first output word

is only 12 bit) is subsequently derived. One must start with the initial state `0_0` (rule 1) and iteratively elaborate all new evolving states:

	State	Rules	Signal	Next State	
(0)	`0_0`	1,2,3	T	`9_0`	(1)
		1,2,4,6	L	`0_0`	(0)
		1,2,3,4,6	TL	`9_9`	(2)
(1)	`9_0`	2,3	T	`99_96`	(3)
		2,4,6	L	`9_9`	(2)
		2,3,4	TL	`99_96'`	(4)
(2)	`9_9`	5	D	`0_0`	(0)
(3)	`99_96`	5	D	**`3`**`_0`	(5)
(4)	`99_96'`	5,6	D	**`3_3`**	(6)
(5)	**`3`**`_0`	2,3	T	**`3`**`9_00`	(7)
		2,4,6	L	**`3_3`**	(6)
		2,3,4,6	TL	**`3`**`9_`**`3`**`9`	(8)
(6)	**`3_3`**	5	D	`0_0`	(0)
(7)	**`3`**`9_00`	2,3	T	**`3`**`99_`**`3`**`93`	(9)
		2,4	L	**`3`**`9_`**`3`**`9`	(8)
		2,3,4	TL	**`3`**`99_`**`3`**`93'`	(10)
(8)	**`3`**`9_`**`3`**`9`	5	D	`0_0`	(0)
(9)	**`3`**`99_`**`3`**`93`	5	D	**`6`**`_0`	(11)
(10)	**`3`**`99_`**`3`**`93'`	5,6	D	**`6_6`**	(12)
(11)	**`6`**`_0`	2,3	T	**`6`**`9_`**`6`**`9`	(13)
		2,4,6	L	**`6_6`**	(12)
		2,3,4,6	TL	**`6`**`9_`**`6`**`9`	(13)
(12)	**`6_6`**	5	D	`0_0`	(0)
(13)	**`6`**`9_`**`6`**`9`	5	D	`0_0`	(0)

From this table all details of the whole circuit can be extracted: The minimal FSM requires 14 states and 24 transitions (number of lines) – all can be directly taken from the table (note that it must be assured by design that only in states with $\mathtt{b} = 0$ the input signals T or L can occur). Moreover, the input shift register must have 3 stages of which the last must only hold 3 bit. Therefore $9 + 9 + 3$ flip-flops are required (since the maximum `a` is **`3`**`99`). The multiplexer must switch only 7 possible combinations, which are

b	**Word Selection**
`9`	w1[8:0], 6'dx
`96`	w2[8:0], w1[8:3]
`3`	w1[2:0], 12'dx
`3``9`	w2[2:0], w1[8:0], 3'dx
`3``93`	w3[2:0], w2[8:0], w1[8:6]
`6``9`	w2[5:0], w1[8:0]
`6`	w1[5:0], 9'dx

Finally, the valid signal, which indicates that a certain output word is complete, must be set in each state with $\mathtt{b} \neq 0$.

Bibliography

[1] *Brookhaven National Laboratory (BNL), Brookhaven, USA*, http://www.bnl.gov.

[2] *European Organization for Nuclear Research (CERN), Geneva, Switzerland*, http://www.cern.ch.

[3] *Helmholtzzentrum für Schwerionenforschung GmbH (GSI), Darmstadt, Germany*, http://www.gsi.de.

[4] *Joint Institute for Nuclear Research (JINR), Dubna, Russia*, http://www.jinr.ru.

[5] *Hot stuff: CERN physicists create record-breaking sub-atomic soup*, http://blogs.nature.com/news/2012/08/hot-stuff-cern-physicists-create-record-breaking-subatomic-soup.html, 2012.

[6] *'Perfect' Liquid Hot Enough to be Quark Soup*, http://www.bnl.gov/rhic/news2/news.asp?a=1074&t=pr, 2012.

[7] T. Bel P. Dillenseger A. Arend, H. Appelshäuser and M. Hartig, *Test of the Frankfurt CBM TRD prototypes at the CERN-PS*, GSI Scientific Report 2011 (2012), 48.

[8] H.H. Andersen and J.F. Ziegler, *Hydrogen Stopping Powers and Ranges in all Elements*, Stopping and ranges of ions in matter, Pergamon Press, 1977.

[9] A. Andronic, *Electron Identification Performance with ALICE TRD Prototypes*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **522** (2004), no. 1–2, 40 – 44.

[10] G. Anelli, M. Campbell, M. Delmastro, F. Faccio, S. Floria, A. Giraldo, E. H. M. Heijne, P. Jarron, K. C. Kloukinas, A. Marchioro, P. Moreira, and W. Snoeys, *Radiation tolerant VLSI circuits in standard deep submicron CMOS technologies for the LHC experiments: practical design aspects*, IEEE Trans. Nucl. Sci. **46** (1999), no. 6, pt.1, 1690–6.

[11] T. Armbruster, *The SPADIC Project Website*, http://spadic.uni-hd.de, 2012.

[12] T. Armbruster, P. Fischer, M. Krieger, and I. Peric, *Multi-Channel Charge Pulse Amplification, Digitization and Processing ASIC for Detector Applications*, Nuclear Science Symposium Conference Record (NSS/MIC), IEEE, Nov. 2012, not published yet.

[13] T. Armbruster, P. Fischer, and I. Peric, *SPADIC - A self-triggered pulse amplification and digitization ASIC*, Nuclear Science Symposium Conference Record (NSS/MIC), IEEE, Nov. 2010, pp. 1358 –1362.

[14] J.F. Bak, A. Burenkov, J.B.B. Petersen, E. Uggerhøj, S.P. Møller, and P. Siffert, *Large departures from Landau distributions for high-energy particles traversing thin Si and Ge targets*, Nuclear Physics B **288** (1987), no. 0, 681 – 716.

[15] T. Balog, W. F. J Mueller, and C. J. Schmidt, *Comparision of SPADIC and n-XYTER self-triggered front-end chips*, GSI Scientific Report 2011 (2012), 41.

[16] H. Bethe, *Zur Theorie des Durchgangs schneller Korpuskularstrahlen durch Materie*, Ann. Phys. **397** (1930), 325–400.

[17] H. A. Bethe, *Molière's Theory of Multiple Scattering*, Phys. Rev. **89** (1953), 1256–1266.

[18] H. Bichsel, *Straggling in thin silicon detectors*, Rev. Mod. Phys. **60** (1988), 663–699.

[19] J. Bleck-Neuhaus, *Elementare Teilchen: Moderne Physik von den Atomen bis zum Standard-Modell*, Springer-Lehrbuch, Springer, 2010.

[20] R.K. Bock and A. Vasilescu, *The Particle Detector BriefBook*, Accelerator Physics, Springer, 1998.

[21] A.D. Booth, *A signed binary multiplication technique*, Quart. Journ. Mech. and Applied Math. **4** (1951), no. 2, 236–240.

[22] P. Braun-Munziger and J. Stachel, *The quest for the quark-gluon plasma*, Nature **448** (2007), 302 – 309.

[23] A.S. Brogna, S. Buzzetti, W. Dabrowski, T. Fiutowski, B. Gebauer, M. Klein, C.J. Schmidt, H.K. Soltveit, R. Szczygiel, and U. Trunk, *N-XYTER, a CMOS read-out ASIC for high resolution time and amplitude measurements on high rate multi-channel counting mode neutron detectors*, Nuclear Instruments and Methods **568** (2006), no. 1, 301 – 308.

[24] D. Emschermann C. Bergmann, A. Andronic and J. P. Wessels, *Test of Münster CBM TRD prototypes at the CERN PS/T9 beam line*, GSI Scientific Report 2011 (2012), 47.

[25] R.L. Chase, A. Hrisoho, and J.-P. Richer, *8-channel CMOS preamplifier and shaper with adjustable peaking time and automatic pole-zero cancellation*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **409** (1998), no. 1–3, 328 – 331.

[26] Christian Kreidl, ZITI - Heidelberg University, *Wire-Bonding Expert*, 2012.

[27] J. Cleymans, H. Oeschler, K. Redlich, and S. Wheaton, *Comparison of chemical freeze-out criteria in heavy-ion collisions*, Phys. Rev. C **73** (2006), 034905.

[28] ALICE Collaboration, *ALICE Technical Design Report*, (2001).

[29] The HADES Collaboration, *The high-acceptance dielectron spectrometer HADES*, The European Physical Journal A **41** (2009), 243–277.

[30] L. Dadda, *Some schemes for parallel multipliers*, (1965).

[31] P. Dillenseger, *Charakterisierung und Signalanalyse von TRD-Prototypen fuer das CBM-Experiment*, Master thesis, Institut für Kernphysik, Frankfurt University, 2012.

[32] A. Andronic et al., *The Transition Radiation Detector ALICE*, http://www-alice.gsi.de/trd/, 2012.

[33] I. Augustin et al., *BTR_Accelerator and Scientific Infrastructure*, FAIR Baseline Technical Report (2006).

[34] K. Nakamura et al. (Particle Data Group), *2011 Review of Particle Physics*, J. Phys. G **37** (2010), 1+.

[35] Chair for Circuit Design, *Uxibo website, Uxipedia*, http://www.uxibo.de, 2012.

[36] Bengt Friman, Claudia Hoehne, Joern Knoll, Stefan Leupold, Joergen Randrup, Ralf Rapp, and Peter Senger, *The CBM Physics Book - Compressed Baryonic Matter in Laboratory Experiments*, Lecture Notes in Physics, Springer, 2011.

[37] J. Furletova and S. Furletov, *New transition radiation detection technique based on DEPFET silicon pixel matrices*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **628** (2011), no. 1, 309 – 314, VCI 2010 Proceedings.

[38] G. Gramegna, P. O'Connor, P. Rehak, and S. Hart, *CMOS preamplifier for low-capacitance detectors*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **390** (1997), no. 1–2, 241 – 250.

[39] Ulrich W. Heinz and Maurice Jacob, *Evidence for a new state of matter: An Assessment of the results from the CERN lead beam program*, ArXiv Nuclear Theory e-prints (2000).

[40] Ivan Rusanov, GSI, *Design Tips for the Front-End PCB*, private communication, 2011.

[41] J. D. Jackson, *Electromagnetic form factor corrections to collisional energy loss of pions and protons, and spin correction for muons*, Phys. Rev. D **59** (1998), 017301.

[42] M. Krieger, *Entwurf und Simulation eines digitalen Tail-Cancellation-Filters*, Diploma thesis, Lehrstuhl für Schaltungstechnik und Simulation, Universität Heidelberg, 2011.

[43] F. Krummenacher, *Pixel detectors with local intelligence: an IC designer point of view*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **305** (1991), no. 3, 527 – 532.

[44] L.D. Landau, *On the Energy Loss of Fast Particles by Ionization*, J. Phys. USSR **8** (1944), 201.

[45] F. Lemke, *Unified Synchronized Data Acquisition Networks*, Dissertation, Mannheim University, 2012.

[46] J. Lindhard, M. Scharff, and H. E. Schiøtt, Kgl. Danske Videnskab. Selskab, Mat.-Fys. Medd. 33 No. 14. (1963).

[47] S. Lochner and H. Deppe, *Radiation studies on the UMC 180nm CMOS process at GSI*, Radiation and Its Effects on Components and Systems (RADECS), 2009 European Conference on, Sep. 2009, pp. 614 –616.

[48] Michael Krieger, ZITI - Heidelberg University, *SPADIC 1.0 Setup 2 and Measurements*, 2012.

[49] B. Murmann, *A/D converter trends: Power dissipation, scaling and digitally assisted architectures*, Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE, 2008, pp. 105–112.

[50] D. Muthers and R. Tielert, *A 75MS/s Low Power Pipeline ADC with scalable Resolution*, VLSI Design, Automation and Test, 2006 International Symposium on, april 2006, pp. 1 –4.

[51] W. F. J. Müller, *Status DAQ*, Talk from 20th CBM Collaboration Meeting 2012, VECC Kolkata, 2012.

[52] D.G. Nairn, *Zero-voltage switching in switched current circuits*, Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on, vol. 5, May 1994, pp. 289 –292.

[53] V. G. Oklobdzija, D. Villeger, and S. S. Liu, *A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach*, IEEE Trans. Comput. **45** (1996), no. 3, 294–306.

[54] A.V. Oppenheim, R.W. Schafer, and J.R. Buck, *Zeitdiskrete Signalverarbeitung*, Elektrotechnik : Signalverarbeitung, Pearson Studium, 2004.

[55] I. Peric, T. Armbruster, M. Koch, C. Kreidl, and P. Fischer, *DCD - The Multi-Channel Current-Mode ADC Chip for the Readout of DEPFET Pixel Detectors*, Nuclear Science, IEEE Transactions on **57** (2010), no. 2, 743 –753.

[56] Peter Fischer, ZITI - Heidelberg University, *Calculating the Spacial Resolution*, private communication, 2012.

[57] B. Povh, *Particles and Nuclei: An Introduction to the Physical Concepts*, Physics and Astronomy Online Library, Springer, 2004.

[58] J. Randrup and J. Cleymans, *Maximum freeze-out baryon density in nuclear collisions*, Phys. Rev. C **74** (2006), 047901.

[59] W. Riegler, *Detectors (Experimental Physics)*, CERN Summer Student Lecture Programme Course, 2011.

[60] L. Rossi, P. Fischer, T. Rohe, and N. Wermes, *Pixel Detectors: From Fundamentals to Applications*, Particle Acceleration And Detection, Springer, 2006.

[61] J. Adamczewski-Musch S. Linev and P. Zumbruch, *Status of data acquisition software DABC*, GSI Scientific Report 2011 (2012), 48.

[62] S. M. Seltzer and M. J. Berger, *Evaluation of the collision stopping power of elements and compounds for electrons and positrons*, The International Journal of Applied Radiation and Isotopes **33** (1982), no. 11, 1189 – 1218.

[63] ———, *Improved procedure for calculating the collision stopping power of elements and compounds for electrons and positrons*, The International Journal of Applied Radiation and Isotopes **35** (1984), no. 7, 665 – 676.

[64] P. Senger, *The Compressed Baryonic Matter experiment at FAIR*, Nuclear Physics A **862–863** (2011), no. 0, 139 – 145.

[65] C.E. Shannon, *Communication in the Presence of Noise*, Proceedings of the IRE **37** (1949), no. 1, 10 – 21.

[66] H. Spieler, *Semiconductor Detector Systems*, Oxford University Press, 2005.

[67] I. E. Tamm, J. Phys. USSR **1** (1939), 439.

[68] FAIR Joint Core Team [FAIR Joint Core Team], *Green Paper - FAIR Modularized Start Version*, 11 2009, https://www-alt.gsi.de/documents/DOC-2009-Nov-124.html.

[69] D. Villeger and V. G. Oklobdzija, *Evaluation Of Booth Encoding Techniques For Parallel Multiplier Implementation*, Electronics Letters **29** (1993), no. 23, 2016 – 2017.

[70] R.H. Walden, *Analog-to-digital converter survey and analysis*, IEEE Journal on Selected Areas in Communications **17** (1999), 539–550.

[71] C. S. Wallace, *A Suggestion for a Fast Multiplier*, Electronic Computers, IEEE Transactions on **EC-13** (1964), no. 1, 14 –17.

Acknowledgments

It has really been a long time and a lot of people have actually supported me in numberless different ways. Several times I have started to list the names of those who somehow contributed, but even the longest result always seemed incomplete or inaccurate. Therefore I have eventually decided to skip that part completely, but to express at least how thankful I feel towards all who contributed in either way. I must honestly say that cooperation always was the nicest part and nothing ever kept my motivation higher than professional exchange. I sincerely hope that I have never missed to say thank you in the respective situations.