1 Readout Chip Characterization

The design of the two HGCAL endcaps includes a total of about 30 000 sensors containing about 6 000 000 Si channels that each needs to be read out. Therefore, an amount of about 100 000 HGCROCs must be produced, mounted and tested [?, ?]. The final chip and hexaboard testing will take place at six mechanical assembly centers (MACs). By testing a significant amount of chips (223 so far) individually, on PCBs and on PCBs attached to sensors data for characterization has been taken.

Characterization of electronics devices (i.e. chips) refers to determining statistically characteristic values for key properties. By measuring these properties together with the impact on the performance of the devices, a quality control framework can be established accepting only devices within a certain range of characteristic values. Additionally, it enables a post-production tuning of these properties, per chip, in order to bring them into the accepted range. To create such a framework is the overarching task towards which the work in this chapter is directed.

1.1 Process variation

Process variation describes the variation of physical properties such as length, width and oxide thickness of semiconductor devices [?]. This variation causes measurable differences in performance between components that have been produced in the same way (e.g. same batch). Although digital circuits are affected as well, the effect on analog components is more significant due to their operation in a continuous range [?].

The HGCROC has the task to amplify naturally weak occurring signals caused by particles interacting with sensor material to an accurate and measurable one. To achieve this goal the ROC includes for each read-out channel several analog components that are termed *the analog front-end*. The components of the analog front-end together with the calibration circuit (which is shared amongst channels) can be seen in Fig. 1. Process variation, the requirement of precision and the amount of analog components inside the ROC motivate the characterization efforts.

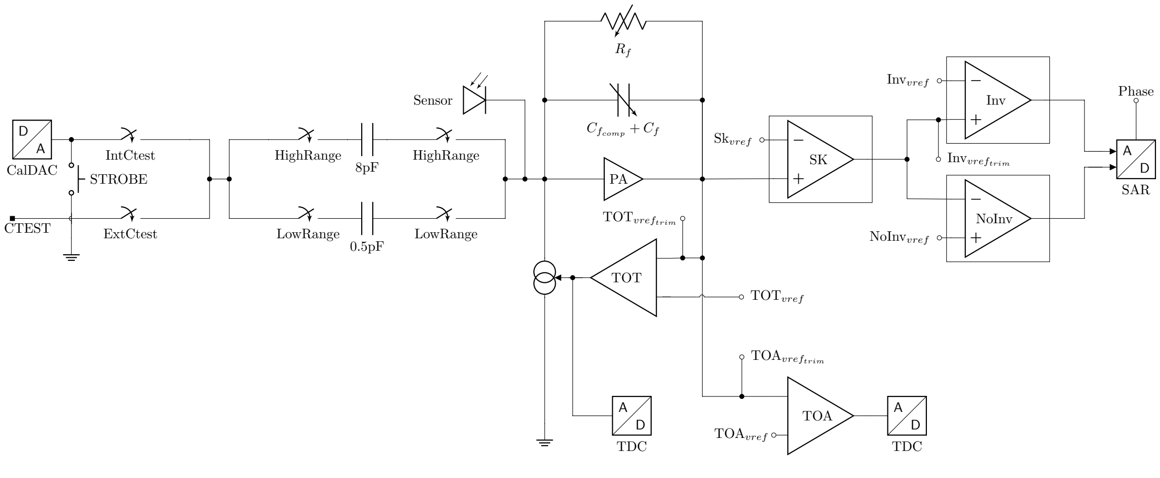


Figure 1: Analog front-end of HGCROC

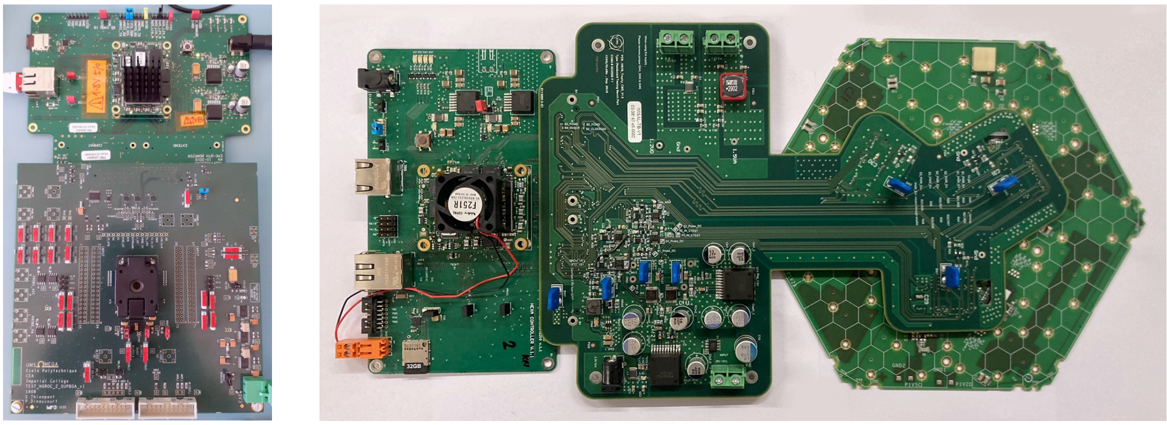
1.2 Strategy

Besides taking data for characterization, another major focus lies on the accumulation and transfer of knowledge attained from online analysis (at the time of testing) to the chip designers. As the HGCAL project foresees three HGCROC generations to obtain the final readout chip, the generated knowledge from testing has a direct impact on design decisions of the subsequent generation. At the time of writing the second generation of HGCROCs was subject to tests. The online analysis is performed to investigate certain chip components that are of immediate interest and allow a direct evaluation about the functioning of a chip. In general, single chip, hexaboard and module tests form three distinct stages of testing to assure quality along the different production steps.

Additionally, the chip designers built several digital-to-analog converters (DACs) on critical analog components into the ROC in order to allow for post-production tuning. Some of these handles can be set per half ROC (global) while others are set per channel (local). The DACs can be programmed via specific configuration commands that usually change the voltage supplied to a specific circuit element. The handles postfixed by *trim* in Fig. 1 represent local handles while all others are global.

1.2.1 Hardware

Due to the three testing stages, two different test systems have been developed. For single chip tests the single-chip characterization (ZIF socket) board, shown in Fig. 2a is used. The central piece of the single-chip board is the black ZIF (zero insertion force) socket sledge that allows to insert a ROC and to press the balls onto metal contacts that establish an electrical connection. Furthermore, the board includes two connectors (at the bottom of the figure) that breakout specific pins of the ROC to external devices.



(a) ZIF-Socket board.

(b) Trophyboard with hexaboard.

Figure 2: Test boards attached to hexacontroller.

For hexaboard and module tests the test system depicted in Fig. 2b is used. The hexaboard is attached via three connectors (not seen here) to an intermediate board referred to as *Trophyboard*, which allows to read out the three (six) chips mounted on low density (high density) hexaboards. The Trophyboard houses six ADCs for reading out the same pins that the ZIF socket board provided by the two break-out connectors. Both

Trophyboard and single-chip board are in turn connected to a control board called *hexacontroller*. The hexacontroller uses a Trenz TE0820 System on Module integrating a Xilinx Zynq UltraScale+ FPGA and processing unit. The processing unit runs the Linux distribution CentOS while the FPGA is programmed via a custom-designed firmware to enable high-speed communication to the testing boards. The FPGA configures the ROCs, controls the data acquisition, collects the data from the ROCs and makes them accessible to the OS.

1.2.2 Software

The HGCROC provides two independent interfaces for chip configuration and data acquisition. To configure the parameters of the analog front-end (and the digital part of the ROC) so called *Slow Control* messages are send via the inter-integrated-circuit (I^2C) bus to the chip. For further information on the I^2C bus and protocol the reader may refer to [**?**]. For setting up and initializing a data acquisition so called *Fast Commands* can be sent via the Advanced eXtensible Interface [**?**] (AXI). The ROCs are continually sending data (consisting of a bit pattern that represents an idle status) through a high-speed connection (eLink) with 1.28 Gb/s. Only after sending an L1A signal to the ROC, the FPGA firmware expects readout data on the eLinks after some latency and starts the acquisition. Due to the separation and independence of the two interfaces software development was carried out in parallel. A schematic overview of the software setup can be seen in Fig. 3. Measurement, data storage and online analysis is triggered by an external client computer via ethernet.

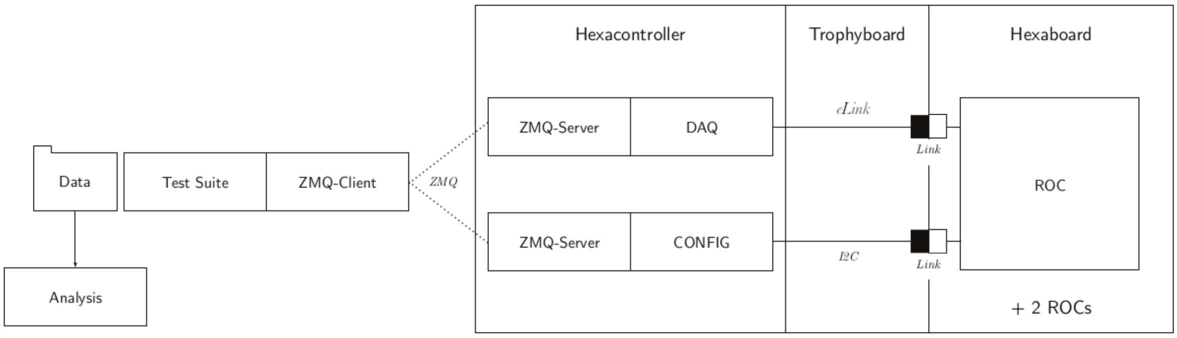
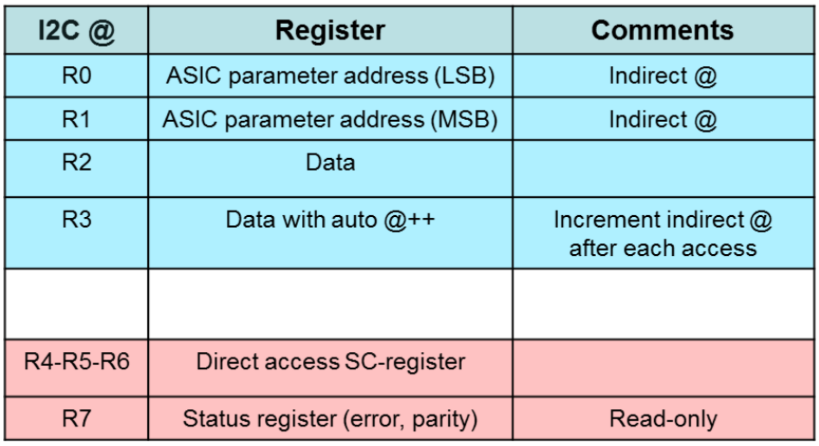


Figure 3: Software architecture of test systems. The client computer (left) sends configuration files to the hexacontorller server (right), which configures the ROCs and sends the data back to the client which stores it for later analysis.

In order to characterize and tune components of the analog front-end internal DACs need to be set. The ROC provides eight I^{2}C addresses of which four (R0 to R3) grant access to setting the values of the DACs. As seen in Fig. 4 writing into registers R0 and R1 form a 16-bit address pointer that references a specific register in HGCROC memory. After the pointer is set, writing the desired value byte to R2 fills the referenced register with the specified value. Similarly, writing the value byte to R3 writes to the same location and increments the address pointer by one after the write, such that another write to R3 would fill the register at the successive address. Thus, taking data for characterization consists of successive configuration and data acquisition.



I2C @	Register	Comments
R0	ASIC parameter address (LSB)	Indirect @
R1	ASIC parameter address (MSB)	Indirect @
R2	Data	
R3	Data with auto @++	Increment indirect @ after each access
R4-R5-R6	Direct access SC-register	
R7	Status register (error, parity)	Read-only

Figure 4: Table of I^2C registers and their function [?].

Software development efforts were directed by three overarching requirements:

1. Remote operation
2. Compatibility across all testing systems
3. Performance and fault tolerance

As mentioned before, one of the author's main contributions was the development of software for configuration. Therefore, the next sections give a brief overview of the software engineering decisions made in order to fullfill the above mentioned requirements.

Remote operation is facilitated via a TCP socket established by the embedded networking library ZeroMQ (ZMQ) [?]. The hexacontroller running the configuration software acts as server while the remote computer initializing tests is the client (Fig. 3). Control flow between client and server is handled via the request/reply pattern. The client sends a request for configuration which the server handles by starting a new thread for each ROC to configure. When each thread finishes successfully, the server sends a reply signaling that it is ready for configuration again. If an error occurs during the handling of the request, the client is notified with an error code in the reply.

The ROC understands a configuration as a sequence of bytes written to certain registers. However, in order to create a more user-friendly interface to configuration the software includes a conversion table of human-readable parameter names to register addresses, referred to as the *Register Map*. This map enables the user to write structured documents as YAML-files which are then sent to the server. YAML is a human-readable data-serialization language commonly used for configuration files. As certain parameters span more than one register while others occupy only a fraction of a register the conversion routine was crafted in a generalized way considering all edge cases. In case a register's value cannot be inferred from the information available at a given time, the value was read from the ROC in order to not overwrite information.

Compatibility with all test systems is assured by an automatic detection routine, scanning all available I^{2}C busses for responding ROCs. As depicted in Fig. 4 one chip will show up as eight consecutive I^{2}C addresses on a bus. The amount of ROCs found on the busses determines the platform and thereby the test system that is used: One ROC for single-chip, two ROCs for scintillator tileboards, three ROCs for low density hexaboards and six chips for high density hexaboards. All test boards are designed in a way that the first of the eight consecutive I^{2}C addresses has a unique value. This allows for example in hexaboard systems to detect and correct for the state of rotation. This is accomplished by defining a mapping of *correct* rotation between ROC address and bus ID which in turn depends on the pin assignment defined in the firmware and thereby does not change. However, since the configuration software is independent of the software for data acquisition, the latter needs to be notified about the state of rotation as well in order to establish the correct mapping to the readout links.

The requirement of **performance** has been achieved by avoiding redundant I^{2}C transactions and by caching. The most significant performance boost in terms of redundant bus transactions came by saving the most recent written address and to reuse it in subsequent read/write operation. Many operations use partially overlapping or identical registers R0 and R1 which can be exploited by partially (*lazily*) updating the state of these registers. This mechanism was further enhanced by sorting the operations of a configuration request by similar addresses before writing and using the burst-write register R3 for truly consecutive addresses. The results of these enhancements on I^{2}C transactions can be seen in table 1. Another bottleneck was discovered in converting between parameter and register information. Since the register map used for conversion contains about 2500 entries, continuous table look-ups turned out to consume a significant amount of processing time. By closer inspection of commonly used configuration files and their respective registers, it became clear that only a small subset of registers occur often. By implementing a cache of the conversion, the total runtime for configuration was reduced by a factor of about two.

Table 1: Amount of bus read/write operations for different configuration files. The init.yaml file contains the initial configuration of the ROCs, while all other yaml-files are used to configure the ROC for certain tests. Because these files are executed with each run, a full test requires configuring the ROC with the almost identical configuration file several times in a row. The amount of successive configurations for testing puposes is indicated in paranthesis behind each configuration file.

Config file (configurations)	**Regular R/W**	**Enhanced R/W**	**Speedup**
init.yaml (1×)	708	155	4.6×
scan_pedDAC.yaml (32×)	16512	5250	3.1×
scan_calibdac.yaml (41×)	984	249	4×
config_samplingscan.yaml (16×)	16224	397	41×
scan_vrefinv.yaml (30×)	1440	361	4×
deconfig_samplingscan.yaml (1×)	1008	104	9.7×

In terms of **fault tolerance** another type of cache was developed, termed *Write cache* (Fig. 5). Besides storing all address-value pairs that have been read or written since initialization, the cache avoids redundant write operations. If an error in one of the I^{2}C transactions occurs, a reset of the I^{2}C state machine and the ROC registers is issued before the chip is reconfigured to the last known state saved in the Write cache. After this procedure, the last configuration sent from the client is repeated.

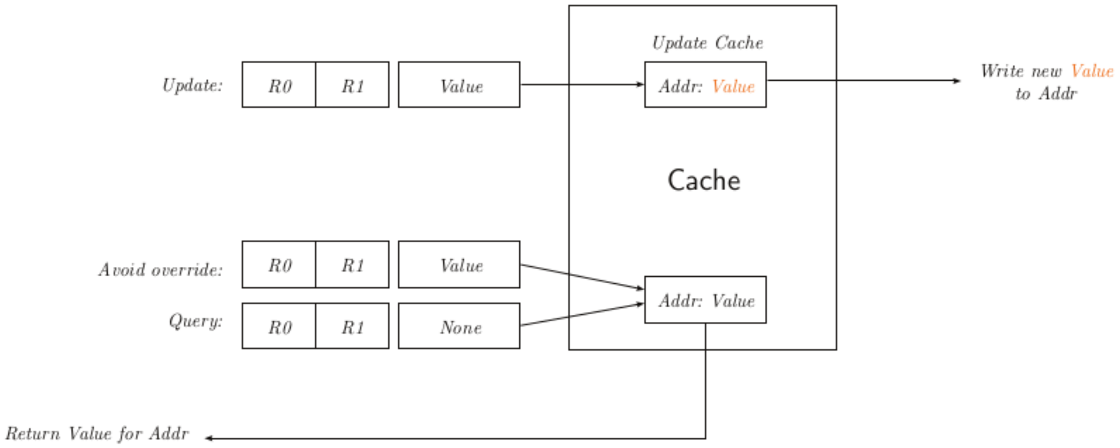


Figure 5: Write cache saves state and avoids redundant writes. It is written to or querried via an address–value pair. The address consists of a tuple (R0,R1) and the value is a byte to be written to the specified ROC register.

1.3 Tests and results

In order to characterize the different components of the ROC's analog front-end, several procedures have been developed and tested. Besides generating characterization data, these procedures include online analysis to give fast feedback which proved to be advantageous in understanding the general behavior of the components as well as the interdependence between components of the analog front-end. The procedures can be controlled via a client computer that communicates with the software suite described in section 1.2.2. The procedures are planned to be used in the final chip, hexaboard, and module tests in the MACs. As the focus of this thesis lies on the implementation of procedures, analysis will only be discussed briefly.

1.3.1 Power consumption

Measuring power consumption gives quick insight into the functioning of a chip. After powering on, a list of default parameters is loaded from permanent memory into the digital and analog parts. In this state a ROC (on a hexaboard) consumes about 2.9 W. Chips that showed a drastic deviation from this value (by about 0.3 W to 0.5 W) did not respond to I^2C communication and were therefore declared malfunctioning - in total 4 of the 223 tested chips behaved like this. Characterization of power consumption allows to reject chips in an early testing stage which is advantageous when, at a later point in time, thousands of chips need to be tested.

For single-chip and multi-chip systems there are two different ways to measure power consumption:

- After connecting the single-chip characterization board (ZIF socket board) to a power supply unit, a constant voltage of 3.3V is being supplied and the drawn current is measured via the connected power supply. The power consumption is calculated by $P = U \cdot I$, which includes the current drawn by the board itself.

- For multi-chip test systems the current drawn is read out via six current sensing amplifiers (INA250A2 [?]) positioned on the interfacing Trophyboard. Each sector[1] of the hexaboard is connected to two current sensing amplifiers for measuring the consumption of the analog and digital parts separately. The INA250A2 is designed with a gain of $G = 500\,\mathrm{mV/A}$ and supplied with a voltage of $V_{sup,dig} = 1.2\,\mathrm{V}$ for the digital part and $V_{sup,ana} = 1.5\,\mathrm{V}$ for the analog part. The output signal of the sensing amplifiers is subsequently converted into a digital signal by ADCs [?] which are read out by the hexacontroller via I^2C. Power consumption is calculated from the ADC output V_{INA} (automatically converted by the Linux I^2C ADC Kernel driver) as $P = V_{sup} \cdot V_{INA}/G$. The total power consumption of the chip is the sum of consumptions from the digital and analog parts. Typically $P_{ana} \approx 2.5\,\mathrm{W}$ and $P_{dig} \approx 0.5\,\mathrm{W}$ for low density hexaboards (one chip per sector) and twice these amounts for high density hexaboards (two chips per sector) after power-on.

1.3.2 Probe points

The ROC is designed to allow reading out several direct current (DC) probe points located at critical components in the circuitry. Characterizing these DC levels across chips (and their halves) helps the chip designers better understand the impact that process variation has on the chip design. Due to different architectures for single- and multi-chip systems there are two different ways to measure the probe points:

- Single-chip boards expose two ports (one per half) that provide two pins each (ProbeDC1 and ProbeDC2) for an external device to read from. The probe points must first be configured and can then be read out, one at a time by a connected digital multimeter via GPIB.

- Hexaboards break-out the ROC's ProbeDC pin per sector to the Trophyboard. The Trophyboard includes one ProbeDC ADC per sector, accessible to the hexacontroller via I^2C. Since the halves cannot be read out in parallel as in the case of single-chip boards, care must be taken when configuring the chip. For low density hexaboards the probe points must be read out sequentially first for one half, then deactivated and afterwards read out for the other half. For high density hexaboards (two chips per sector) this procedure must be done sequentially for both chips of a sector.

The results of measuring the probe points on different chips can be seen in Fig. 6. The variation of some points is stronger than for others. Especially the TDCs *VD_*TDC** and the reference voltage of the pre-amplifier *Vref_pa*, *VrefCf* and *Vcp* seem to be particularly effected by process variation.

Another probe point called $CTEST$ is used to characterize the internal charge injection DAC termed $CalDAC$. $CalDAC$ is used in most of the characterization tests and has

[1] A third of a hexaboard is defined as a sector. Low density modules mount one ROC per sector, high density modules mount two ROCs per sector.

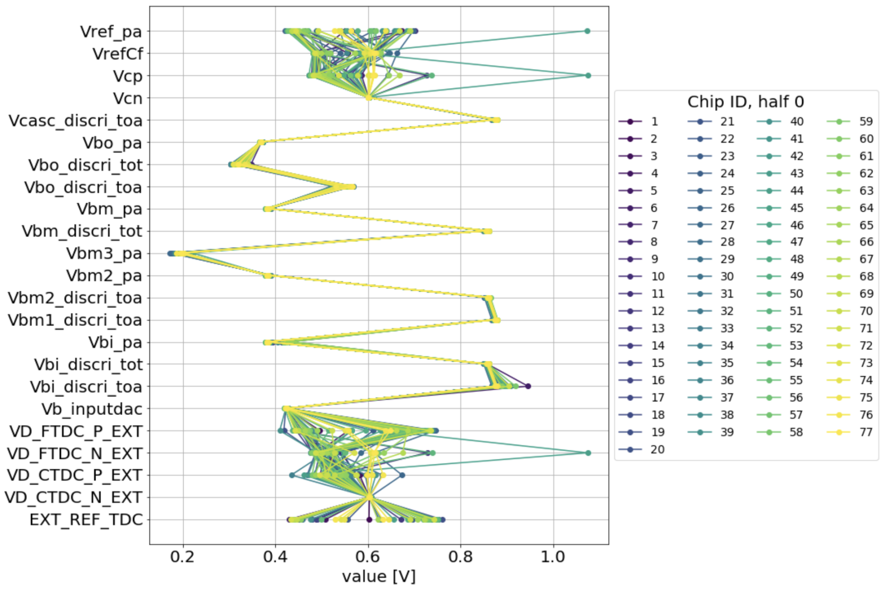


Figure 6: ProbeDC measurements for one half across chips [?].

thereby an implicit impact on the measurements. Therefore it makes sense to characterize $CalDAC$ as well by measuring voltages that are produced after setting particular DAC values. Voltage as a function of the DAC values are expected to increase linearly with a spread in slope and offset between chips as result of process variation. The results in Fig. 8 show a more coherent behavior for lower $CalDAC$ values. The offset at $CalDAC = 0$ is almost identical between chips, however a dispersion in slope becomes apparent when looking at larger $CalDAC$ values.

To accomplish the measurement of $CalDAC$, the switches $IntCtest$ and $ExtCtest$ must be closed, while switches $HighRange$ and $LowRange$ are open (Fig. 7). By triggering the $STROBE$ (Fast Command) signal, $CTEST$ can be read out either via an external device over the $CTEST$ pin (for single-chip) or via the $IntCtest$ ADC on the Trophyboard.

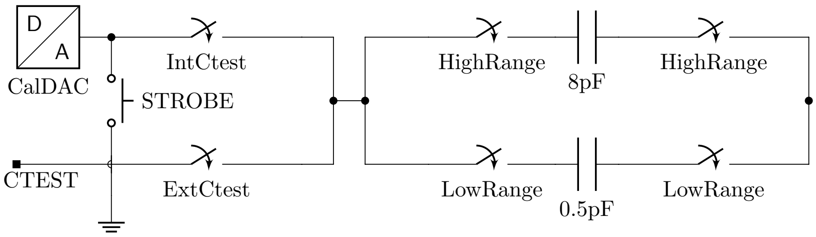


Figure 7: HGCROC calibration circuit. Figure is a zoomed excerpt of Fig. 1.

1.3.3 Pedestals

A pedestal is a DC level that is constant over time. Since real signals are always subject to noise the pedestal is defined as the average (equivalent) DC level of a signal. The baseline pedestal is the DC level that is measured when no charge is injected. It is noteworthy that this level changes when the calibration circuit is connected to a channel. Measuring

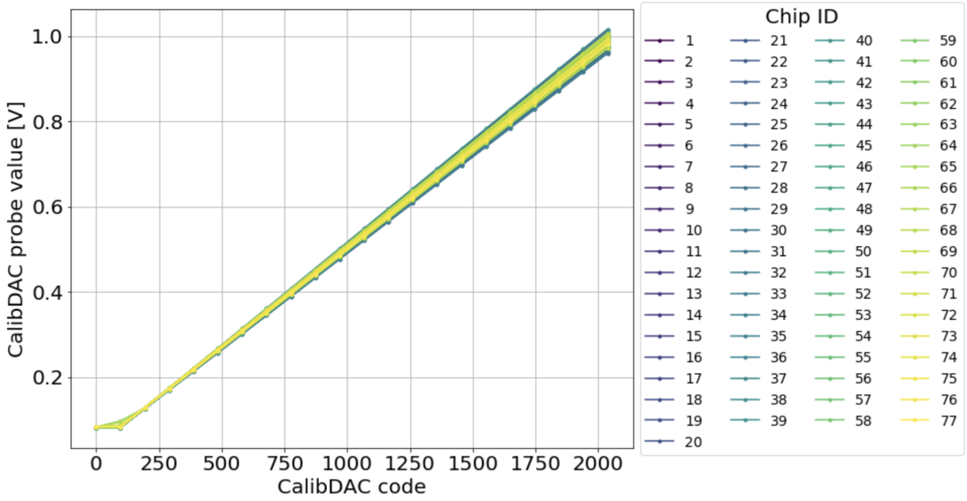
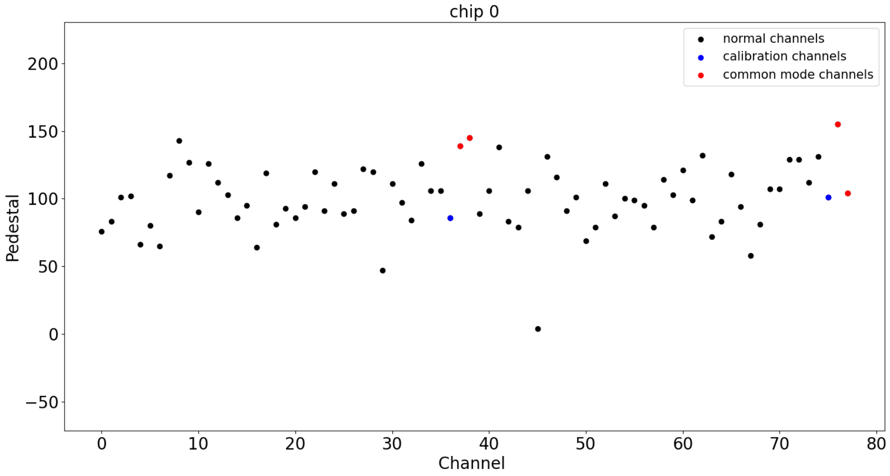


Figure 8: Voltage at $CTEST$ as function of $CalDAC$ value [?].

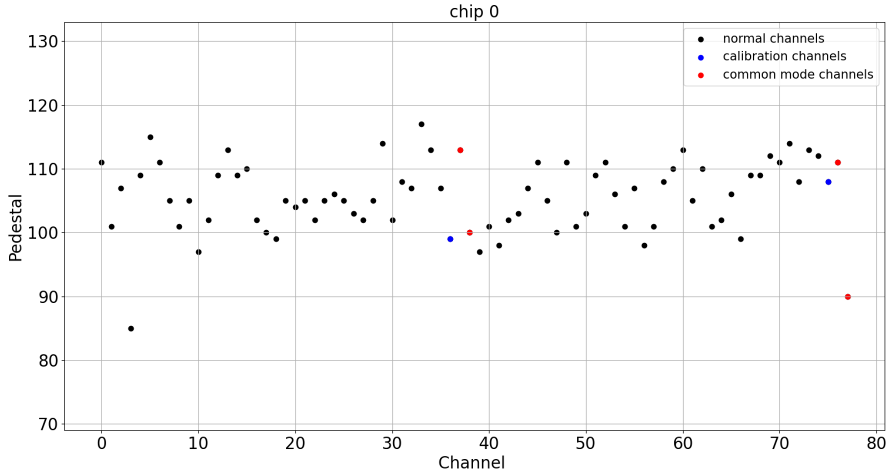
the pedestals is referred to as *pedestal run*. Since the two halves of a chip are physically separated, the pedestals of channels in the halves tend to accumulate around different baseline pedestals. As can be seen in Fig. 9a there is a large inter-channel dispersion of about 100 ADC counts and a relatively high average pedestal of about 150–175 ADC counts with power-on default settings. Both of these outcomes are not desired. Desired would be a low average pedestal, in order to obtain a larger dynamic range, and an as small as possible inter-channel dispersion, in order to obtain a most similar channel response.

To mitigate such large dispersion and offset, the chip designers included the local shaper handle $Inv_{vref,trim}$ and the global shaper handles Inv_{vref} and $Noinv_{vref}$. While the two global handles are discussed in 1.3.4 the local handles allow each channel's pedestal being adjusted in 32 steps. The act of adjusting local pedestals is referred to as *pedestal trimming* and the encapsulating procedure for finding the optimal pedestal per channel is called *pedestal scan* (Fig. 10). Starting from each channel's baseline pedestals and plotting the ADC response as a function of $Inv_{vref,trim}$, a horizontal line can be found that minimizes the pedestal dispersion by using one particular value for $Inv_{vref,trim}$. Instead of solving this problem analytically, the horizontal line is set to begin at the channel with the highest baseline pedestal (at $Inv_{vref,trim} = 0$). This is often a good enough approximation and requires less computation. A chip's channel is defined as *untrimmable* if its function does not cross the horizontal line. In practice, untrimmable channels have not been observed in any of the tested chips.

After the pedestals have been trimmed, a pedestal run shows the effect (Fig. 9b). It is apparent that the inter-channel dispersion has been reduced from 100 ADC counts to about ten. It is desired to reach a similar pedestal level for each channel per half with low inter-channel dispersion. This allows for using simple algorithms – pedestal subtraction, zero suppression, common-mode noise correction, etc – to post-process the sensor data.



(a) Before pedestal trimming.



(b) After pedestal trimming.

Figure 9: Measurement of baseline pedestals.

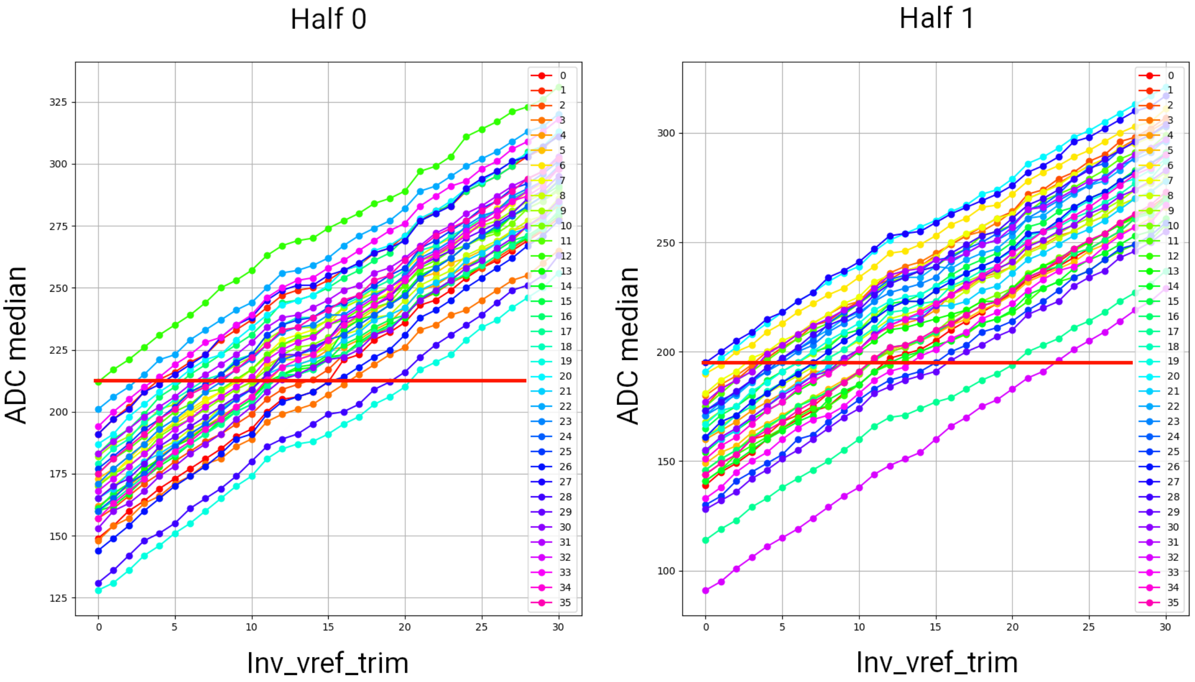


Figure 10: ADC counts as a function of $Inv_{vref,trim}$.

1.3.4 Shapers

The shaper block is composed of three stages: A Sallen-Key shaper, the inverting and non-inverting shapers, and unity gain amplifiers (buffers). While the Sallen-Key shaper and the buffers optimize the signal-to-noise ratio [?] and provide a stable signal for ADCs to sample from, the inverting and non-inverting shapers condition its dynamic range. After the signal leaves the Sallen-Key shaper it is fed to both inverting and non-inverting shapers where it is positively (negatively) amplified and further propagated as a differential input to the ADC. Since the shaper gain is susceptible to process variation, it must be characterized. To tune the gain and mitigate inter-half variation in dynamic range, the chip designers built in two global DACs Inv_{vref} and $NoInv_{vref}$. Together with the local shaper handle $Inv_{vref,trim}$ as mentioned in 1.3.3 and the voltage level V_{PA} provided by the pre-amplifier the magnitude of the output signal $S = |S_- - S_+|$ follows from:

$$S_+ = 3 \cdot (Inv_{vref} - Inv_{vref,trim}) - 2V_{PA} \tag{1}$$

$$S_- = 2V_{PA} - NoInv_{vref} \tag{2}$$

To better understand the behavior of the shaper response in practice one can set all possible combinations of Inv_{vref} and $NoInv_{vref}$ as a function of the differential output signal measured by the ADC. The result of such a scan on baseline pedestals can be seen in Fig. 11a. A preceding pedestal trimming has been performed in order to mitigate the impact of inter-channel dispersion on the median ADC value calculated per half. As shown, a horizontal shift between the two distributions becomes apparent which can be attributed to the physical separation of the two halves inside a chip's package. In order to maximize the dynamic range, a low output signal for baseline pedestals (light blue) is desired which would correspond to different Inv_{vref} and $NoInv_{vref}$ settings for the different halves. To see the upper end of the dynamic range, the same scan is performed but instead of baseline pedestals a charge is injected. The charge is chosen to be so large that it will saturate the ADCs, e.g. $CalDAC = 1000$ using $LowRange$ capacitor. The existence of optimal global shaper settings (where all ADCs will saturate before $CalDAC = 1000$) is assumed. The result is shown in Fig. 11b. The difference between the plots 11a and 11b is the dynamic range (Fig. 11c). Only values of dynamic range larger than 800 ADC counts are plotted to illustrate the existence of a region of optimal global shaper settings which is situated at similar positions for both halves. In order to maximize the dynamic range, a global shaper setting from a central position of this blob is desired.

1.3.5 Analog-to-Digital Converters

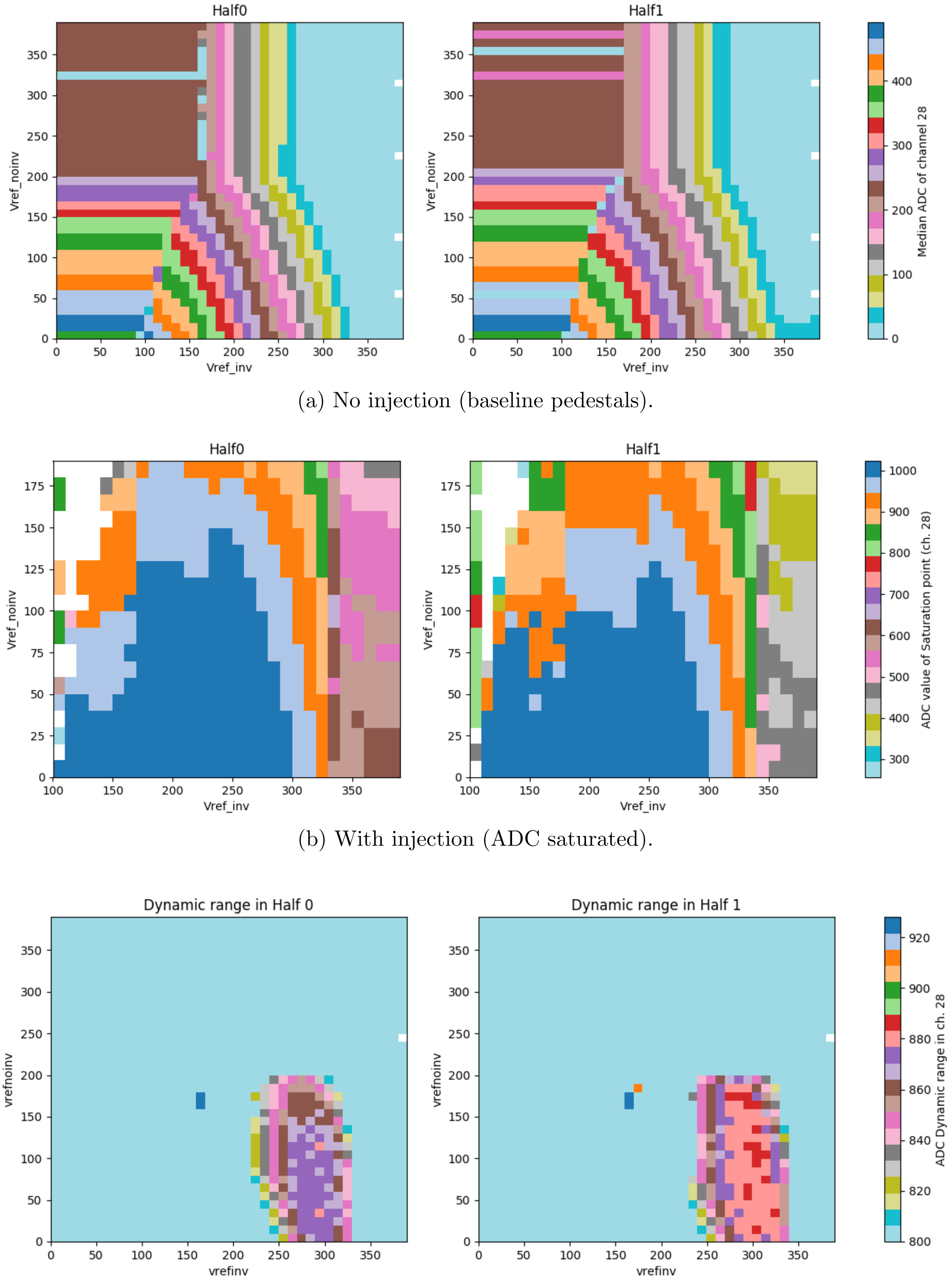
The successive approximation register (SAR) ADC samples the voltage differential supplied by inverting and non-inverting shaper outputs with a 40MHz sampling frequency into a 10-bit digital signal. Due to delay introduced by the preceding circuit the ADC might start converting the analog signal at an unfavorable point in time. To avoid this, the ADC has a phase handle that can offset the start of conversion. The phase parameter is tunable in 16 steps of about 1.5 ns. The 16th step corresponds to one full BX (delay of 25 ns). It is desired that the ADC starts the conversion at the maximum of the input signal. The phase parameter is a global setting. The procedure that detects the ideal phase parameter is called *sampling scan*. In Fig. 12 a phase offset of one, corresponding to 25 ns, would start the conversion at the best timing.

After the optimal phase is set, the ADC's response to input signals can be characterized. By increasing the injected charge, a linear increase in ADC counts is expected, saturating at some point. This test is referred to as *injection scan* (Fig. 13a). The slope of the injection scan and the location of the saturation point is desired to be similar between channels and chips. A non-optimal shaper reference voltage leads to earlier ADC saturation as can be seen in Fig. 13b or no ADC response at all. This is one of the main results from the first testing session of single chips and led to the investigation of dynamic range with respect to the Inv_{vref} and $Noinv_{vref}$ parameters discussed before.

1.3.6 Discriminators

The Time-Over-Threshold (TOT) and Time-Of-Arrival (TOA) discriminators (comparators) each trigger a Time-To-Digital converter (TDC) when the input voltage exceeds a set reference voltage. The TDC used for TOA measurements is activated at the rising edge of the discriminator output to capture the time the signal arrives, while the TDC for TOT measurement triggers on rising and falling edges of the discriminator output to capture the time the comparator was activated, i.e. the time spent over the reference voltage threshold. The TOT measurement allows the measurement of large charges that exceed the dynamic range of the ADC. Therefore, a reference voltage about the amount of charge it takes to saturate the ADC is ideal. For the TOA comparator a reference voltage as low as possible is ideal to record the timing of the smallest amount of input charge without being too low to be triggered by noise. For that reason, the chip designers included global and local DACs on the reference voltage of the discriminators.

Since it is desired that the chips' TOA and TOT measurements are uniform for identical input signals, the behavior of the discriminators must be characterized. This is



(a) No injection (baseline pedestals).

(b) With injection (ADC saturated).

(c) The difference between Figs. (a) and (b) yielding the ADC dynamic range.

Figure 11: ADC counts as function of the global shaper parameters.

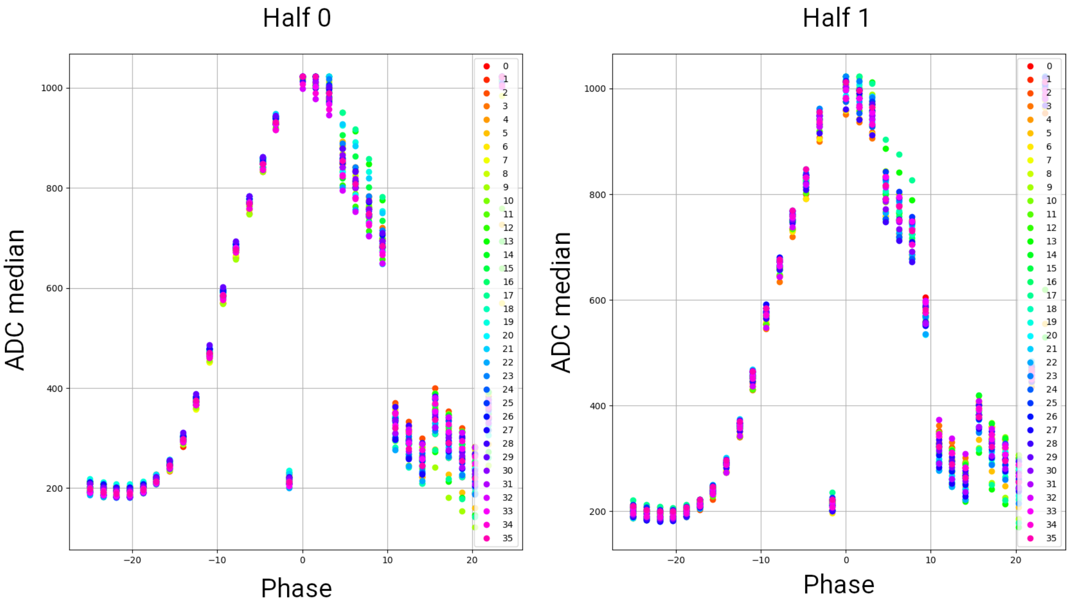
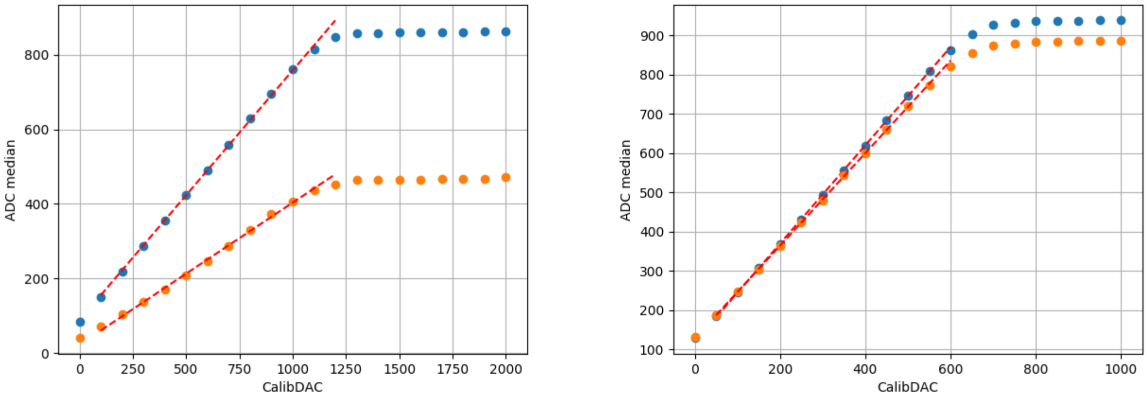


Figure 12: ADC count as a function of Phase parameter for two halves



(a) Non-optimal shaper settings.

(b) Optimal shaper settings.

Figure 13: Injection scans for two halves of a chip. The blue curve corresponds to the first half, orange to the second.

done in two ways. Firstly, by varying the global thresholds TOT_{vref} and TOA_{vref} at a set injected charge and recording the threshold for 50% efficiency (e.g. 50 of 100 events activated the TDC), incrementing the amount of charge and repeating the procedure. This is referred to as *global TOT/TOA threshold scan* or *S-curve*. When plotting the global thresholds at 50% efficiency as a function of the amount of injected charge a linear relation is expected. Secondly, the local pedestals are determined in a similar manner to section 1.3.3 after which a trimming can be applied to mitigate inter-channel dispersion. The first testing session of single chips showed inconclusive results of TOA and TOT measurements. In the hexaboard testing sessions a bug in the PCB design was discovered that prevented the study of the TDCs. At the time of writing, the discriminators and their contribution to noise are under investigation.

1.4 Conclusion

In summary, 223 single chips were tested, of which 186 passed the preliminary selection criteria that were set up and subsequently used in the assembly of 21 LD and the very first 12 HD hexaboards. The testing was critical, especially for the HD hexaboards as only 12 ”good” PCBs existed at that time and it could not be afforded to waste any boards due to non-working chips being mounted. The outcome was that all boards worked fine after mounting. Furthermore, 54 hexaboards (42 LD and 12 HD) have been tested in two testing campaigns. At the time of writing the testing data are being analyzed and compared across chips and platforms which will yield a list of final rejection criteria and tolerances. Besides the produced raw data needed for characterization, online analysis at the time of testing opened new areas of investigation, e.g. optimization of global shaper parameters, interference caused by discriminator and TDC activity and noise studies on hexaboards and modules. Each measurement campaign helped to improve software and firmware - one example for this is the optimization of the configuration software discussed in section 1.2.2. Finally, a recent effort towards usability led to developments of a server-side web application and a client-side graphical user interface program as interfaces to the developed programs, which the author is involved in.