Internship report

Optimisation of the Automated Calibration Routine of the ANANAS Board

Heidelberg University, Electronic Vision(s) Group
Supervisor: José Luis Montes Marrero

Carl Clemens von Randow

February 2024

This report describes the optimisation of the automated calibration routine of the **Ana**log **N**etwork **A**ttached **S**ampling board (ANANAS). The handling and storing of the data has been restructured, plotting functionalities have been added for the data and results, as well as for intermediate calculation values. Validation scripts were written in order to analyse the quality of the calibration and assess its accuracy. Furthermore, some debugging efforts concerning the ground shift and the terminating resistors have been undertaken.

Contents

1 Objective

The goal of this internship was to analyse and verify the quality of the calibration parameters obtained in the automated calibration routine of the ANANAS board. In order to be able to do that, a few issues needed to be resolved. On the one hand, the ground shift was not working correctly for all boards. On the other hand, reliability issues for the measurements of the terminating resistors were found.

Once the verification of the calibration is completed, the further objective will be to incorporate the ANANAS board into the wafer-scale system, in order to enhance the analog readout capabilities, which are currently limited to 12 channels per module.

2 Introduction

The BrainScaleS-1 system is a wafer-scale neuromorphic hardware system developed by the Electronic Vision(s) Group. At its heart is a silicon wafer made up of 384 *High Input Count Analog Neural Network chips* (HICANNs) [6], each containing 512 neurons and up to 114688 synapses. Additionally, there are multiple auxiliary printed circuit boards for different purposes, e.g. for communication, power distribution or the readout of analog traces. Currently, there can be 12 traces recorded simultaneously per wafer.

The **Ana**log **N**etwork **A**ttached **S**ampling board (ANANAS) [1] was designed in order to increase this number by a factor of 7 to 84 concurrent recordings (two ANANAS boards are used for one wafer), whilst also increasing the quality of the recorded traces. To this end, each board has two different types of analog-to-digital converters (ADCs): A high precision 20 bit lowspeed ADC (lsADC) [3] with a sampling rate of 2.325 kHz, and three 12 bit highspeed ADCs (hsADCs) [2] with a sampling rate of 31.25 MHz. Three hsADCs are necessary because they only have 16 channels each, whereas the lsADC has 64 channels and therefore more than enough for the 48 data channels accessible to each board. The hsADCs are measuring their input differentially against a *common mode voltage*, which needs to be shifted to be in the centre of the desired input range. This is achieved by operating it on a shifted ground level.

The work carried out in this internship extends work done by Simon Rosenkranz [4] and Jakob Sawatzki [5].

3 Experimental Setup

The experimental setup can be seen in Figure 1. On the right hand side, the ANANAS board is shown, with its three hsADCs and one lsADC, as well as its Spartan FPGA-based board (FlySpi). The input voltage is generated by the sourcemeter and connected to the ANANAS board via a 50 Ω series resistance and the analog input test adapter. The user controls everything from the host PC, which is connected to the front end *Helvetica*, which controls the *HBPHost*s (in this case number 6) via the group's *Skynet* subnet. The *HBPHost* then controls the sourcemeter through a USB connection, as well as controls and communicates with the ANANAS board through Ethernet, via the group's *wafernet* subnet. The recorded data is stored in the FPGA's RAM during the measurement process and then read out at the end of the experiment via the Ethernet connection.

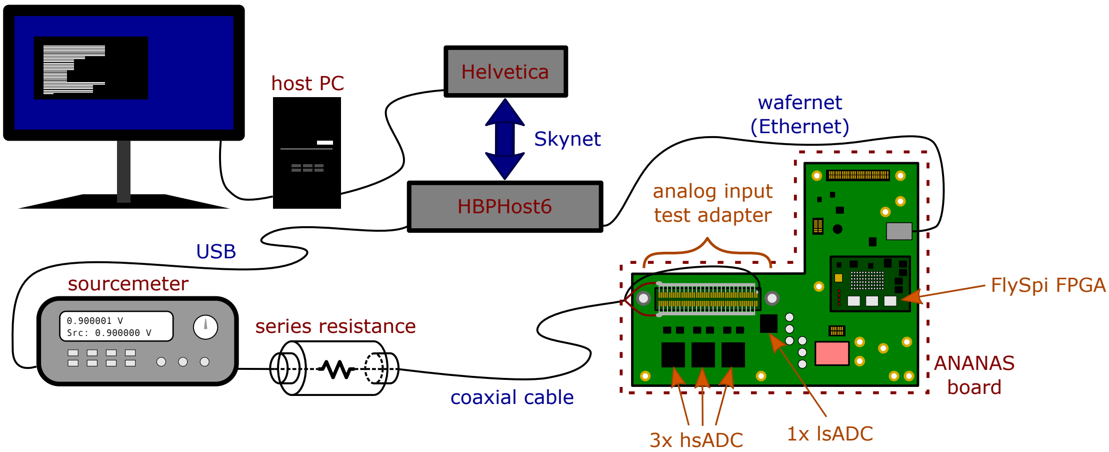

Figure 1: A sketch of the setup, which was created by Simon Rosenkranz and taken from his BSc. Thesis [4, p. 17] and slightly modified.

4 Optimisation of the Calibration Routine

4.1 Restructuring of the Lowspeed Data

The main difference introduced by the restructuring is the way the data is stored. Instead of using nested python lists without any names, the data is now stored in nested python dictionaries, which contain expressive names for the keys. This improves the readability of the data structures and the ease of accessing the correct data. It also improves debugging, because now the data can be handled more easily, and will also be saved to a file. The latter allows reusing the data for later runs, and would also have previously been possible, but much more complicated.

The first application of this was debugging the failing of the calibration routine when run for more than 50 steps. Ultimately, this was found to be caused by the saturation of the lsADC, or more precisely the improper cutting off of the saturated steps.

4.2 Restructuring of the Highspeed Data

The restructuring of the highspeed data was more complex than for the lowspeed data.

The way the data is stored has been changed in a very similar manner to the lowspeed data restructuring, e.g. using `dict` s instead of `list` s, providing the already mentioned advantages, especially accessing the correct data. Previously, the access was complicated by the fact that there needs to be one channel reserved for tracking the common mode voltage, which lead to the fact that there would be a "jump" in the channel numbers, which was not reflected in the list indices. Take for example the second channel (`ch_1`) as the tracking channel: This would mean one would not receive any data for this channel, so the data would be for `ch_0` , `ch_2` , `ch_3` ,...; but the list indices would of course be running from 0 to 6, meaning the data for `ch_2` would have to be accessed by list index 1, and so on. Using the dictionary, the data will simply be accessed by the string `ch_2` as key.

An issue was found, where the calibration routine was failing when trying to use 80 voltage steps or more. The problem was linked to the retrigger count, which is stored as a 9-bit integer, which means its maximum value is 511. Seven channels per trigger group were swept, which means that for 80 steps per channel the retrigger count exceeded its maximum value. This was solved by creating a new `anacoma` object for every channel as opposed to every trigger group, which now allows the sweep of every channel to have up to 512 steps.

Additionally, the base structure of the calibration routine has been improved. Previously, one channel was chosen to track the common mode, then the measurement for all trigger groups was executed and then the calculation of all the data was performed. This resulted in a full set of calibration values for only seven out of eight channels per trigger group, obtained with a manually selected tracking channel. But since the board has eight channels and the common mode voltage correction requires calibration data, one is interested in calibration values for all channels of the board. The easiest way for this would be to run the routine again with a different tracking channel, for example by introducing a `for` loop. Then one would have obtained calibration values for the missing channel, but at the cost of duplicating the baseline measurement, as well as obtained six additional sets of calibration values for the rest of the channels, which are not strictly necessary, and therefore adding possibly unnecessary overhead. One could either proceed by simply discarding those values, or instead by calculating the average of the two sets of values. The first option is not desirable, since it discards useful data, and therefore the overall quality of the calibration would suffer from not using them. However, the second option would result in only six channels having calibration values that consist of an average of two sets of calibration values, whereas the other two channels would only consist of one set. In which case one could argue that the calibrations of those two channels are somehow inferior in quality when compared to the others, for example by being less robust against outliers or measurement issues.

Instead, the routine was restructured in such a way that now all the measurements are executed, and only afterwards all calibration values will be calculated. This way, one gets a set of calibration values for every channel, which has been obtained by taking into account all seven possible common mode tracking channels and determining the average. The voltage sweep has to be executed for every channel separately, due to the shorted inputs on the analog input test adapter, but the board measures and returns the data for all eight channels regardless. Previously, all data except for the driven channel and the tracking channel was discarded as it was considered not to be of interest. But using it now increases stability against outliers or measurement issues and comes at no additional cost in terms of recording time. On the contrary: If one had wanted to get a full set of calibration values that uses information from every possible tracking channel using the previous program structure, it would have taken roughly seven times as long, requiring additional measurements due to the discarded data. And even then it would have produced separate files and not done any averaging.

The new structure also allowed another improvement. For the calculation of the channel specific offset for the hsADC it is necessary to calculate the mean offset value of all channels from one hsADC, i.e. two trigger groups (16 channels).[1] Previously, only 14 channels were used in this calculation, because one channel of each trigger group was used for the common mode tracking and therefore no calibration values were available. Now the measurements for all channels are completed before starting the analysis, so that calibration values are present for every channel when this calculation is executed. This inaccuracy could be corrected easily with the new structure.

4.3 Other Improvements

Some additional improvements have been introduced. On the one hand, intermediate calibration data is saved to disk now, which makes it easier to identify and solve problems and mistakes by providing further insight into the calculations. On the other hand, the plotting functionalities have been extended, which gives further insight and contributes to validating the calibration results. In addition, three more specific improvements will be shown in the following:

4.3.1 Additional Parser Arguments

The following additional parser arguments have been added to the calibration routine:

- `use-ls-calib`: An optional location of lowspeed calibration files that will be used if the argument is given. Otherwise a new measurement will be performed (which is the default).
- `use-hs-calib`: This is the same as `use-ls-calib`, except for highspeed.
- `plot`: An optional flag that enables plotting.
- `gain`: The gain value to be used (default 19 dB).
- `ground-shift`: The ground shift to be used (default $-1.06\,\mathrm{V}$).
- `loglevel`: The loglevel to be used (default TRACE).

These additional arguments allow for a simpler and more controlled usage of the routine. One example would be parameter sweeps, although one might prefer to do them in a quicker way, for example with a script that only executes a single measurement, which is described in more detail in Subsection 5.2. The first two arguments enable

[1] cf. "Step 5: Acquisition of hsADC-Internal Offset Values" in Section 3.3.2 in [4, p. 45]

reusing data from previous runs, which means those measurements do not have to be executed again, e.g. when only changing some highspeed parameter. Additionally, the parser arguments are saved to a file, in order to quickly be able to see which parameters were used for a specific run.

4.3.2 Ground Shift Measurement

During the internship some problems with the ground shift for the hsADC were faced, which led to the development of the script `probe_ground_shift.py`. The ground shift is set by a DAC on the board and can either be measured with a multimeter directly at the board or derived from a measurement of the `COMMON_MODE` channel of the lsADC. This script executes a sweep from $-0.4\,\mathrm{V}$ to $-1.9\,\mathrm{V}$ ground shift, which makes the lsADC channel go into saturation for both ends of the sweeping range. The desired ground shift value for operation is $-1.06\,\mathrm{V}$. By executing the sweep, the script now not only verifies the ground shift to be at the operation value, but also the ability of the board to set other ground shift values. It then plots the result and offers the possibility of including multimeter measurements in the plots, if there have been any.

Thus the ground shift problems could be traced back to different causes. Some were simply failing or missing hardware components, e.g. the operational amplifier was broken for some boards. This might have been due to some too high currents flowing, which was remedied by replacing a solder bridge jumper with a $20\,\Omega$ resistor and thereby protecting the operational amplifier. A different cause was a USB to JTAG box connected to the FPGA, which is needed to (re)flash the FPGA. But it should not be connected during normal operation of the board, because if it is, then it affects the ground level and the ground shift is no longer working properly.

4.3.3 Terminating Resistors Measurement

Special attention should be paid to the connection from the sourcemeter via series resistance and analog input test adapter to the ANANAS board, as this can affect the measurement of the terminating resistors in a significant manner. In the beginning of this internship this connection was using a distributor box which was unreliable and led to terminating resistor values going as low as $40\,\Omega$. In order to investigate this issue, the script `term.py` has been written. It allows a quick estimation of the terminating resistor value of one channel by making one unterminated and one terminated measurement. It then calculates the terminating resistor by using the measurements and also by using the sourced current and then returns the results. It can also be run in a continuous loop mode. This script helped in tracking the issue to the distributor box and getting correct terminating resistor values in the range of $50\,\Omega$.

5 Results

5.1 Measurement of the Lowspeed Calibration Accuracy

In order to validate the parameters for the lowspeed part of the calibration, the script `lowspeed_single_measurement.py` was written. It executes a voltage sweep for one trigger group and plots the result. Optionally, it also applies some calibration values and plots the difference to the sourced voltage.

The possible parser arguments are the following:

- `ip` : The IP address of the ANANAS board.
- `tg` : Trigger group to be used (default 0).
- `start` : Starting voltage to be used (default 0 V).
- `stop` : Stopping voltage to be used (default 1.8 V).
- `steps` : Voltage steps to be used (default 1).
- `rectime` : Recording time to be used (default 43 ms, i.e. roughly 100 samples).
- `output` : Output directory, by default creating a folder `ls_single_measurement` .
- `calib` : Directory of the calibration to be used, by default no calibration is used.

In Figure 2 the result of applying the calibration can be seen for trigger group 5. The plot is representative for all trigger groups. The maximum deviations are on the order of 0.1 mV around the sourced voltage, which was deemed to be an adequate result. The overall average deviation is 0.04 mV. There seem to be systematic differences, which could suggest a higher order polynomial might be better suited for modelling the ADC, but in both [4] and [5] this was investigated and no benefits were found in light of long term variations (e.g. due to temperature changes). Additionally, it looks like the values for sourcing 0 V are systematically lower than the rest. The cause for this has not really been investigated further because of its small absolute deviation on the order of 0.1 mV.

5.2 Measurement of the Highspeed Calibration Accuracy

The validation of the highspeed part of the calibration is mostly analogous to the lowspeed part. The script `single_measurement.py` was written, which executes a voltage sweep for one channel and plots the result. Optionally, it also applies some calibration values and plots the difference to the sourced voltage. In contrast to the lowspeed part, this script only uses one active channel and not a whole trigger group. This is because the shorted inputs of the analog input test adapter only allow for one closed switch to the highspeed channels at any given time. Otherwise the input would be terminated with additional 50 Ω resistors. Applying the highspeed calibration is also more complex than applying the lowspeed calibration, because one needs an additional

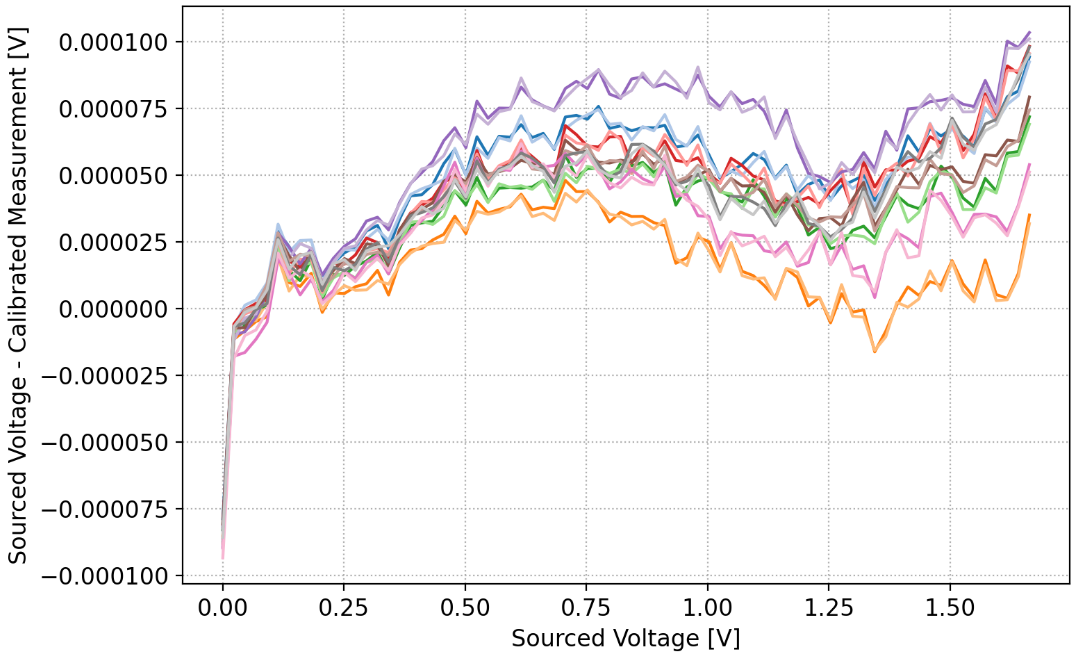

Figure 2: Difference between the sourced voltage and the calibrated lowspeed measurements for trigger group 5 (8 channels with 2 integrators each result in 16 traces). The data above 1.67 V has been cut away due to the ADC going into saturation.

channel to track the common mode voltage due to operating on a shifted ground. In this case, each of the seven other channels will be used separately as correction channel to calibrate the measurement and then the average will be taken.

The possible parser arguments are the following:

- `ip` : The IP address of the ANANAS board.
- `tg` : Trigger group to be used (default 0).
- `ch` : Channel to be used as actively driven channel (default 0).
- `start` : Starting voltage to be used (default 0 V).
- `stop` : Stopping voltage to be used (default 1.8 V).
- `steps` : Voltage steps to be used (default 1).
- `rectime` : Recording time to be used (default 3.2 µs , i.e. roughly 100 samples).
- `gain` : The gain value to be used (default 19 dB).

- `ground-shift` : The ground shift to be used (default $-1.06\,\mathrm{V}$).
- `output` : Output directory, by default creating a folder `single_measurement` .
- `run-number` : Run number (default `None`).
- `calib` : Directory of the calibration to be used, by default no calibration is used.
- `use-data` : Data to be used instead of a measurement, by default no data is used.

The result of applying the calibration to all channels can be seen in Figure 3. The deviations start out around 12 mV for a sourced voltage of 0 V and then most tend to get better as the sourced voltage is increased. The majority is in the range of 4 mV to 8 mV, but some stay on the order of 12 mV. The overall mean of all data points is 9.1 mV deviation. The deviations for every channel seem to follow a linear curve. This would indicate that a linear fit function is well suited, but the parameters are not optimal.

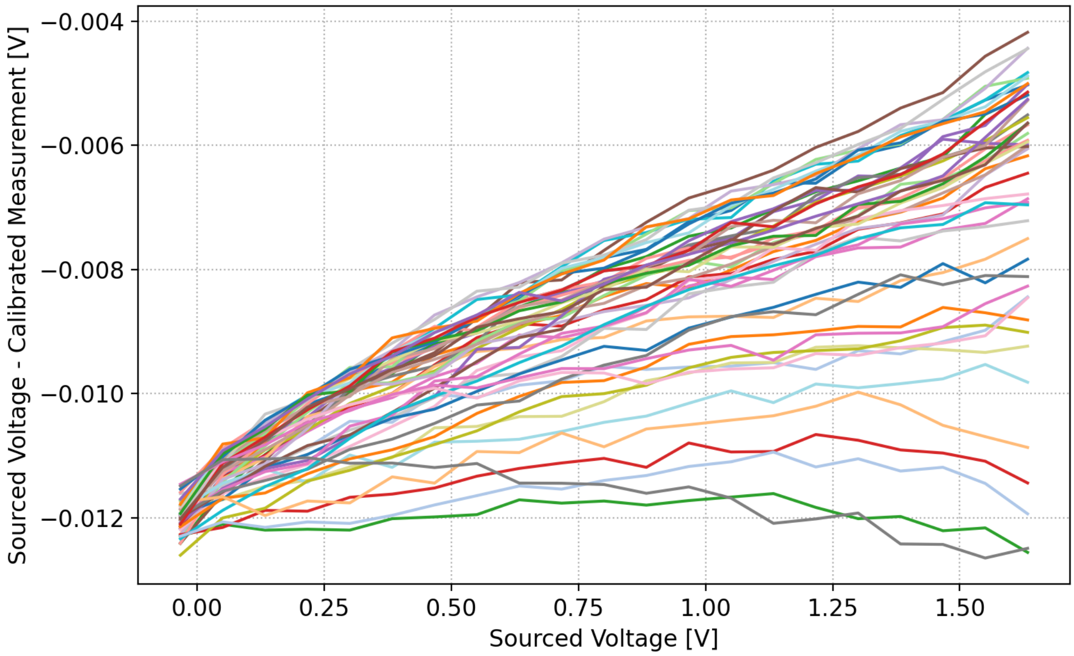

Figure 3: Difference between the sourced voltage and the calibrated highspeed measurements for all 48 channels. Due to saturation, only voltages in the range of $-0.1\,\mathrm{V}$ to $1.7\,\mathrm{V}$ were used.

6 Summary and Outlook

In the course of this internship the automated calibration routine for the ANANAS board was improved by restructuring the data handling and storage, and fixing problems with the ground shift and terminating resistors. Additionally, the routine was extended by plotting functionalities and saving intermediate calculation results. In order to validate the calibration values, scripts were written to execute measurements, apply the calibration and analyse its quality. The obtained results for the lsADC show systematic differences to the actually sourced voltage. But in light of the fact that the deviations are on the order of 0.1 mV, they are perfectly acceptable. The obtained results for the hsADC also show some systematic differences. These deviations on the order of 4 mV to 12 mV can probably be explained by using suboptimal calibration parameters. This could be further investigated, because so far there was no possibility to look into it more closely. If any systematic causes would be found, it might be possible to improve these results.

Another outlook would be the integration of the ANANAS board into the wafer-scale system in order to replace the current analog readout system. This would allow for more analog traces to be recorded simultaneously (84 channels per wafer instead of 12) and increase the quality of the measured data.

References

[1] Joscha Ilmberger. "Development of a digitizer for the BrainScaleS neuromorphic hardware system." Masterarbeit. MA thesis. Universität Heidelberg, 2017.

[2] Texas Instruments. *16 Channel Variable Gain Amplifier (VGA) with Octal High Speed ADC*. AFE5851. Rev. B. May 2010.

[3] Texas Instruments. *64-Channel, Current-Input Analog-to-Digital Converter*. DDC264. Rev. D. Dec. 2016.

[4] Simon Rosenkranz. "Calibration of the Analog HICANN Readout Process." Bachelor's thesis. Universität Heidelberg, 2018.

[5] Jakob Sawatzki. "Validation and Automation of the Calibration Routine of the ANANAS board." Bachelor's thesis. Universität Heidelberg, 2022.

[6] Johannes Schemmel, Johannes Fieres, and Karlheinz Meier. "Wafer-scale integration of analog neural networks." In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. 2008, pp. 431–438.