

Calibration and Regulation of BrainScaleS' PowerIt Subsystems

Patrick Nisblé

August 21, 2018

This Thesis will address Calibration and Regulation of the in BrainScaleS used power delivery Board, PowerIt. Its circuitry will be examined and calibrated, for use in BrainScaleS' monitoring and a method of regulating the 1.8V Output Terminal will be implemented.

This thesis also contains the changes done to the PowerIt Firmware while working on these Problems.

Diese Arbeit befasst sich mit der Kalibration und Regulation der PowerIt Komponenten des BrainScale Systems. Die Schaltungen werden untersucht und kalibriert, damit sie in der Systemüberwachung von Nutzen sein können.

Ausserdem enthält diese Arbeit die Änderungen, die dabei auf die PowerIt Firmware angewendet werden.

Contents

1	Introduction	3
1.1	The BrainScale System	3
1.2	About the PowerIt Subsystem	4
1.3	Contents on Detail	5
2	Theory	6
2.1	Hardware Component Behavior	6
2.2	ADC Calibration	11
2.3	1.8V Output Regulation	11
3	Experiments	17
3.1	Characterization	17
3.2	1.8V Regulation	23
3.3	Pitfalls	31
4	Results	32
4.1	Firmware	32
4.2	Calibration	33
4.3	Regulation	36
5	Outlook	37

1 Introduction

1.1 The BrainScale System

The BrainScale Wafer System [1], developed and used in the electronic visions Group at Heidelberg University is a neuromorphic hardware implementation.

For this thesis the following core components are of importance:

- mixed-signal ASICs, named HICANNs, structured in packs of 8 into Reticles
- Control Units for Reticles, short CURE Boards
- analog breakout Boards, AnaB for short
- and power supply, called PowerIt.

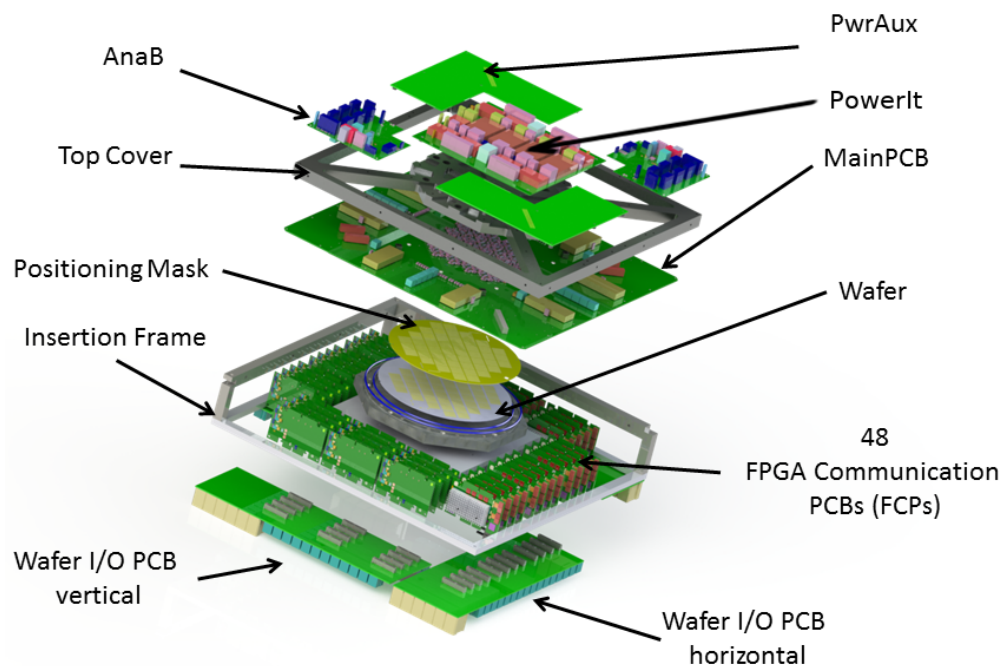


Figure 1.1: The BrainScaleS wafer-scale hardware system, marked are the main components comprising a single wafer system. [2]

1.2 About the PowerIt Subsystem

The main subject of this thesis is the PowerIt board (Figure 1.2). It functions as power supply inside of the WaferScale System (Figure 1.1). In which it is providing the Wafer with 1.8V and the FPGAs with 9.6V. Its maximum rated power draw is 2kW. [3]

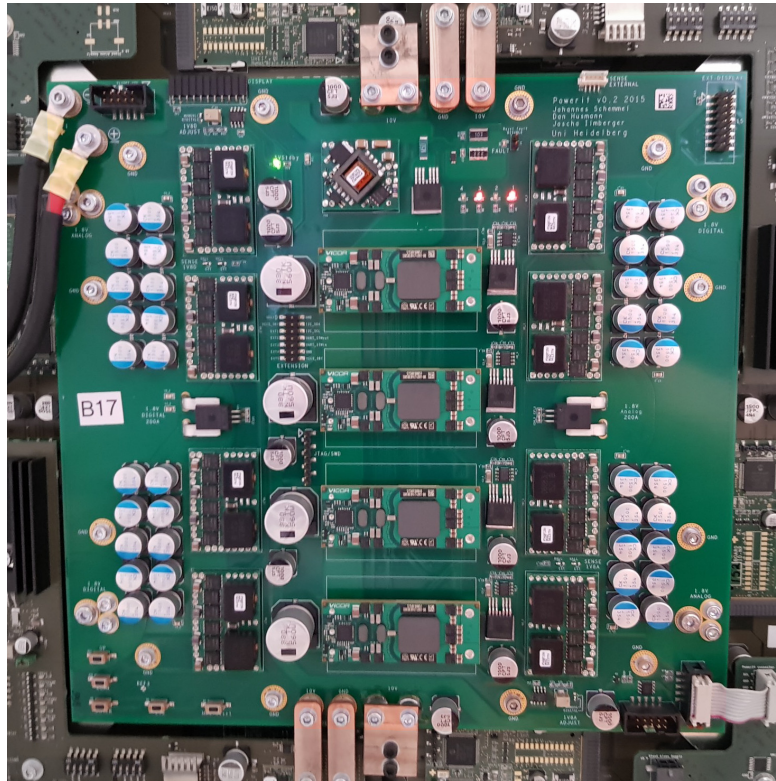


Figure 1.2: PowerIt Board, top view, receiving 48V as input (top left) and outputting 9.6V (top and bottom) as well as 1.8V (analog: top left, bottom right; digital: top right, bottom left)

The Brain of these PowerIt Boards is a STM32 Chip¹ which runs a custom Firmware based on ChibiOS [5]. The PowerIt, while providing 9.6V and 1.8V, also is able to measure the following Values:

- input Voltage and current
- 1.8V output voltage and current
- and the 9.6V output Voltage

Which can then be used inside the Firmware.

¹STM32F405RGT [4]

1.3 Contents on Detail

- Upgrade the PowerIt Firmware to be able to calibrate all on board measurements, voltages and currents.

This requires the Firmware to handle calibration changes on the fly, (each PowerIt requires its own configuration, which needs to be changed after the Firmware is running).

- Provide a communication interface for changing those parameters at runtime.

It is required to have write access to all coefficients (defining the degree of polynomial is done at compile time).

- Provide access to more parameters within the PowerIt while unifying the protocol used.

Collect all writable, readable and static parameters in a single interface using PItCOMM v2, creating a mapping for reference to those values.

- Calibrate the onboard measurement circuits, using a database containing uniquely mappable values.

Characterizing the circuits, providing a default for fallback and a Database file, readable by the BrainScaleS Monitoring System and the Person calibrating each Board.

- Provide a regulation mechanism for stable, modifiable output at its endpoints.

The connection between PowerIt and HICANNs can be seen as a non zero Resistance, which requires the 1.8V output to regulate based on experimental data.

2 Theory

This Chapter will be discussing the fundamental principles used in the experiments. These will contain simplified circuits and their respective Equations as well as component behavior as specified in their respective datasheets by their Manufacturer

2.1 Hardware Component Behavior

Before discussing the experimental results it needs to be clear what circuitry is used in these experiments and what behavior we expect. Keeping in mind, that these are theoretical values and will most likely not be exactly the same as those found in actual hardware, as all values given will always be within some error as defined by their manufacturer.

Each of the three voltage regimes that will be observed on the PowerIt Board, 48V 9.6V and 1.8V, has a voltage- and in the cases of 48V and 1.8V also a current-measurement circuit. Additionally we have a temperature sensor built into the STM32 Chip.

2.1.1 48V Input Voltage

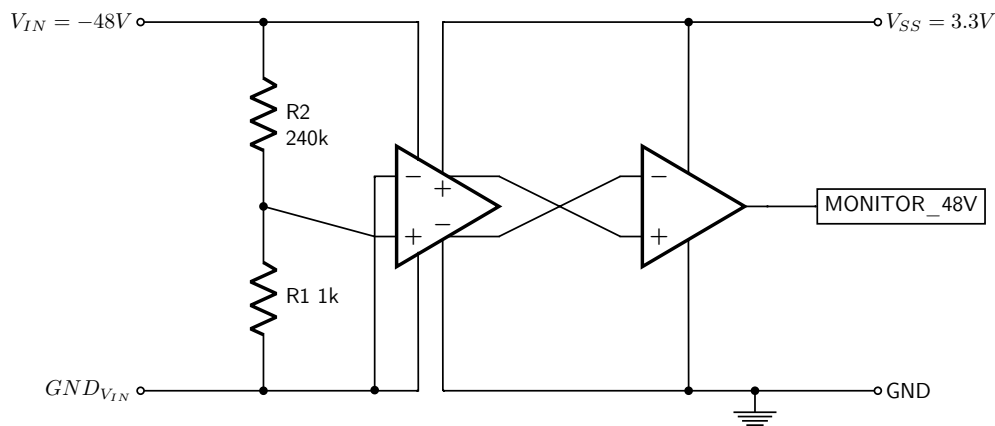


Figure 2.1: Circuit for measuring the 48V input Voltage, consisting of input potential (left), two resistors as voltage divider, one fully differential isolation amplifier (left), one operational Amplifier (right), output voltage as well as the connection to the STM32-Chips input pin (right)

The circuits for measuring input voltage and current are the most complex, because for Voltage measurement the circuit needs to

- divide our input voltage into a usable potential range
- decouple the input (48V) from signal potential (3.3V)
- amplify the voltage, to be in the STM32-Chips Voltage range of 0–3.3V

The already implemented Circuit can be seen in Figure 2.1. It consists of a 1:240 Voltage Divider, a full differential isolation amplifier taking in the roughly 200mV (nominal voltage range), and amplifying it by a factor of 8 ($r_{\text{diffOpAmp}}$ [6]). It is also decoupling the input and output voltages, so our 48V and 3.3V circuit parts are electricly insulated. The remaining operational amplifier provides difference to single ended Conversion with an amplification or 1.1 (r_{OpAmp})

This circuit results in the following equation for calculating the input voltage from a pin voltage:

$$V_{48V \text{ in}} \cdot \frac{R_1}{R_1 + R_2} \cdot r_{\text{diffOpAmp}} \cdot r_{\text{OpAmp}} = V_{\text{MONITOR_48V}}$$

$$\Leftrightarrow \frac{V_{\text{MONITOR_48V}}}{r_{\text{diffOpAmp}} \cdot r_{\text{OpAmp}}} \cdot \frac{R_1 + R_2}{R_1} = V_{48V \text{ in}} \quad (2.1)$$

and the extremes, when assuming $48V \pm 10\%$ are

$$V_{\text{MONITOR_48V, min}} = 43.2 \text{ V} \cdot \frac{1}{240 + 1} \cdot 8 \cdot 1.1 = 1.5774 \text{ V} \quad (2.2)$$

$$V_{\text{MONITOR_48V, max}} = 52.8 \text{ V} \cdot \frac{1}{240 + 1} \cdot 8 \cdot 1.1 = 1.9280 \text{ V} \quad (2.3)$$

2.1.2 48V Input Current

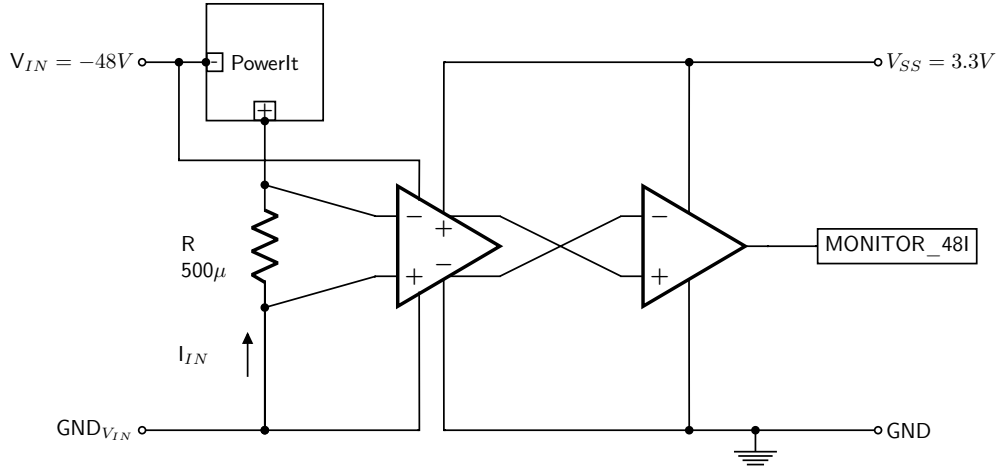


Figure 2.2: Circuit for measuring the 48V input Current, consisting of the powerit Input Circuit, one shunt-resistor, one full diff isolating Amplifier, one operational amplifier, output potential, as well as the connection to the STM32-Chip input pin

The circuit has to satisfy the following constraints:

1. use a shunt resistor, with minimal heat dissipation
2. still providing a good resolution also within the STM32-Chips Specifications

To accomplish that, the circuit is measuring the voltage over a small ($500\mu\Omega$) shunt Resistor, while a current is flowing, resulting in a proportionality, by Ohms Law. The obtained Voltage is then decoupled and amplified by a factor of 8, as well as converted from a difference to single ended Voltage, with a amplification factor of 1.1.

Here we use the same Amplifiers as in subsection 2.1.1 and so we can use the following equation for our input current:

$$I_{48V\ IN} \cdot R_{shunt} \cdot r_{diffOpAmp} \cdot r_{OpAmp} = V_{48I\ pin}$$

$$\Leftrightarrow \frac{V_{48I\ pin}}{R_{shunt}} \cdot \frac{1}{r_{diffOpAmp} \cdot r_{OpAmp}} = I_{48V\ IN} \quad (2.4)$$

And applying the extrema of 0A and 41.7A, a voltage range from 0V to:

$$41.7\ A \cdot 500\ \mu\Omega \cdot 8 \cdot 1.1 = 0.1833\ V \quad (2.5)$$

can be observed.

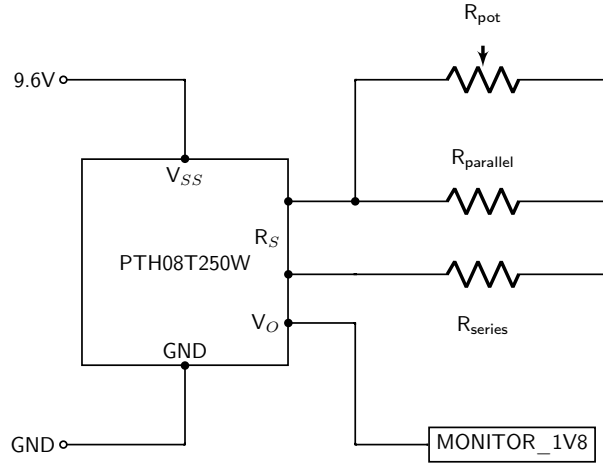
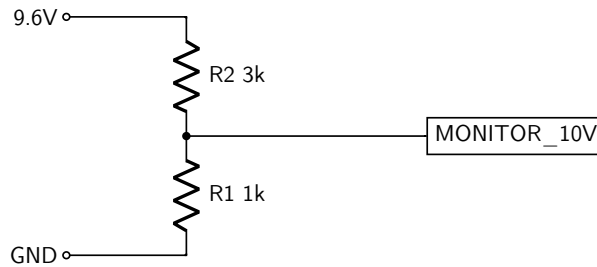


Figure 2.3: 1.8V supply circuit, featuring a DC-DC Converter, a resistor chain, supply voltage (left) and resulting voltage (right)

2.1.3 9.6V Output Voltage

This Circuit consists of a simple 1:3 Voltage Divider.



And the following equation can be used to calculate that voltage:

$$V_{10V \text{ PIN}} = \frac{V_{10V \text{ IN}} \cdot R_1}{R_1 + R_2} \quad (2.6)$$

2.1.4 1.8V Output Voltage

To measure this Voltage the output is directly connected to a pin on the STM32-Chip.

Until now the voltages and current could only be measured, now the mechanism for setting a resulting Voltage at the 1.8V Terminals is known. Figure 2.3 is the circuit for generating 1.8V. It consists of a power module and a resulting resistance between two pins, defined by R_{series} , R_{parallel} and R_{pot} . The Resistances job is to set the output to a given voltage of around 1.8V, but we can vary R_{pot} , because this resistance is set via a digital potentiometer.

The in Figure 2.3 used 1.8V Converters have a characteristic formula [7], and the in this circuit used Potentiometer is a linear 10k Ω Rheostat resulting in the following equations:

$$R_{potentiometer} = P_{val} \frac{10k\Omega}{256} \quad (2.7)$$

$$R_{SET} = 1 / \left(\frac{1}{R_{potentiometer}} + \frac{1}{R_{parallel}} \right) + R_{series} \\ = \frac{R_{potentiometer} \cdot R_{parallel}}{R_{potentiometer} + R_{parallel}} + R_{series} \quad (2.8)$$

$$V_{MONITOR_1V8} = \frac{30.1k\Omega}{R_{SET} + 6.49k\Omega} \cdot 0.7V + 0.7V \quad (2.9)$$

Equation 2.9 is visualized in Figure 2.3. Note that this Equation is proportional to $\frac{1}{x}$

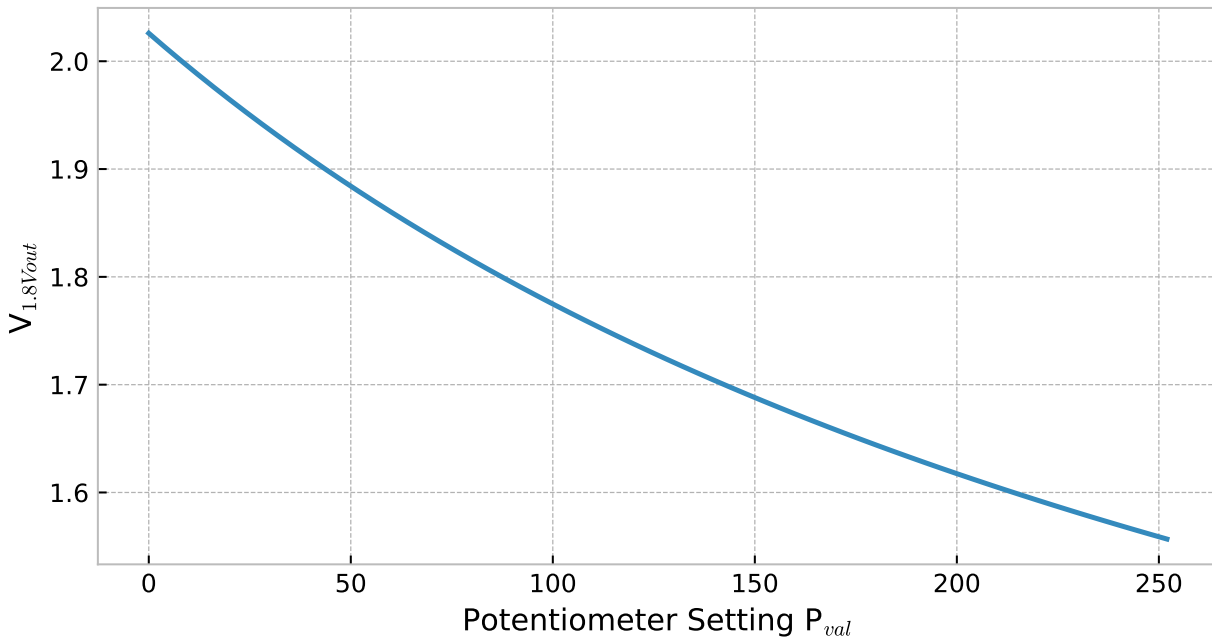


Figure 2.4: Expected behavior of our output voltage by setting the potentiometer

2.1.5 1.8V Output Current

The circuit for measuring the outgoing current over 1.8V, consists of a current sensing IC, which is Hall sensor based. Each connection (digital and analog) has this ic in series to its load.

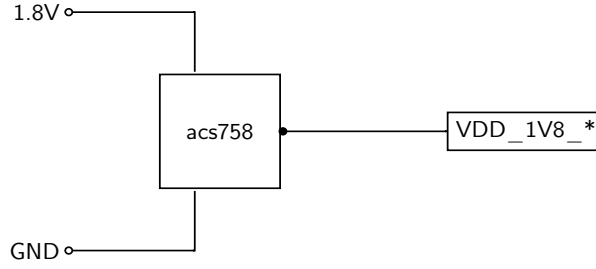


Figure 2.5: 1.8V current sensing circuit, featuring a acs758, hall sensor based current sensing IC, input voltage (left) and output voltage (right)

The IC is rated for a maximum constant current draw of 100A, and has the following behavior:

$$I_{1.8V, in} \cdot 0.004 \text{ V A}^{-1} + 0.12 \text{ V} = V_{\text{MONITOR_1I8_}*} \quad (2.10)$$

which gives for the extremes:

$$0 \text{ A} \cdot 0.004 \text{ V A}^{-1} + 0.12 \text{ V} = 0.12 \text{ V} \quad (2.11)$$

$$100 \text{ A} \cdot 0.004 \text{ V A}^{-1} + 0.12 \text{ V} = 0.52 \text{ V} \quad (2.12)$$

2.2 ADC Calibration

As mentioned beforehand, the actual hardware will differ in behavior from its theoretical counterpart. Therefore it is possible to say that all signals with a significant difference of behavior ($\approx 5\%$) will need to be corrected.

To calibrate these readouts we need to employ some simple actions.

2.2.1 serial ADC readout

The measurements will be done by the STM32-Chip, which uses 12bit ADCs. A single ADC will be switching between all connected pins. This Behavior can be problematic in regards to measuring accurately. The timing used to measure a single pin can be programmatically set from 3 up to 480 clock ticks¹

2.3 1.8V Output Regulation

For regulating the Output the method used is a relatively simple one. The voltage, wanted at the output terminal, will be calculated and then a corresponding potentiometer setting can be extracted, which changes the voltage produced (see Figure 2.3). The second part is already done beforehand and then available as a lookup table² to the firmware. On the other

¹this clock is the internal adc clock, with a frequency of 159 Hz

²mV-Scale, from 1.549V to 2.022V

hand, to calculate the voltage to output, it is necessary to classify the connections between the Powerlts output terminals and the reticle.

2.3.1 Power Wafer

To test the 1.8V Regulation the so called Power Wafer is going to be used, it can be controlled similar to a in BrainScales used, functional, Wafer module. But it is fundamentally different, as it cannot be used for computation, but only to test for voltages and currents. Its internals behave like ohmic resistors, which provides us with a maximum power draw per Reticle of what is possible inside a usable wafer module.

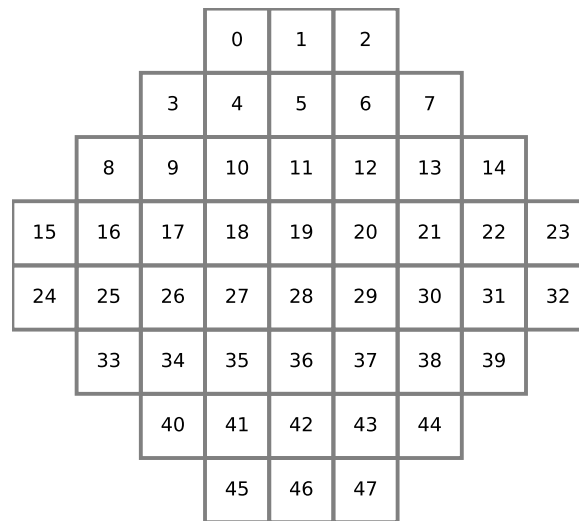


Figure 2.6: reticle diagram of a wafer in BrainScaleS. All 48 Reticles are shown

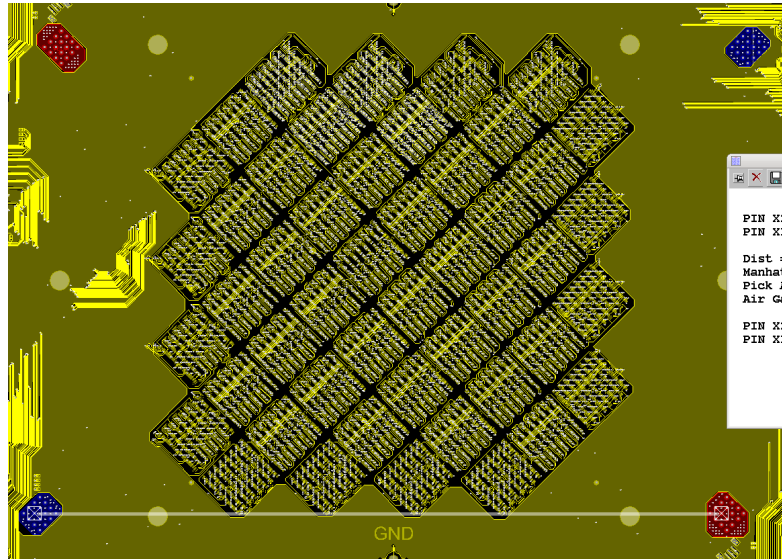


Figure 2.7: part of the mainpcb on which a wafer is placed, visible are the 48 Reticles and two terminals each for 1.8V Digital (blue) and Analog (red), which correspond to the output terminals on a PowerIt

It has the same layout as its system counterparts and each of the 48 Reticles can be accessed, digitally as well as electrically. Each Reticle is connected to its corresponding CURE Board. Those can read voltages of each reticle, but not at the same position as an AnaB. They read right before R_1 in Figure 2.8.

Another specialization of the Power wafer is, that all reticles analog and digital 1.8V lines are connected directly to pins on the Analog Readout Boards [8]. There it is possible to measure a voltage, which is the one after the load resistors in Figure 2.8

2.3.2 Simple Wafer Resistance Model (SWRM)

The circuit in Figure 2.8 can be used, as a first step, to describe the connections powering Reticles inside a wafer.

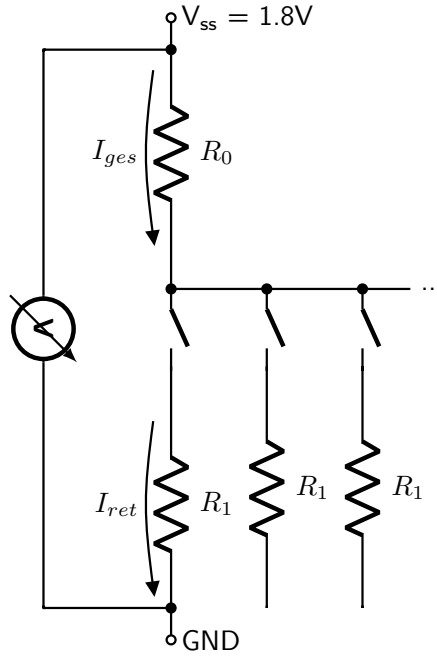


Figure 2.8: model of the to measure resistances and their currents, R_0 describes the resistance of a connection between the PowerIt Output and up to the FET (depicted as switch), while R_1 is a Resistance between FET and Reticles. The measurement is done between Output Terminals on the PowerIt and pins on a Analog readout board

SWRM allows for two fixed resistance values and their respective currents. The current flowing through R_1 will be either 0 or a constant current I_{ret} . The current through R_0 will change depending on the number of reticles that are powered n_{ret}

$$I_{ges} = n_{ret} \cdot I_{ret} \quad (2.13)$$

Therefore the voltage dip V_{dip} as measured by a Voltmeter (see Figure 2.8) can be described with Equation 2.14

$$\begin{aligned} V_{dip} &= V_{R_1} + V_{R_0} \\ &= R_1 \cdot I_{ret} + R_0 \cdot I_{ges} \\ &= I_{ret} \cdot (R_1 + R_0 \cdot n_{ret}) \end{aligned} \quad (2.14)$$

Combining Equations 2.7, 2.8, and 2.9, we gather Equation 2.15. This equation is a reversed function of the one given by the 1.8V supplying IC (see Figure 2.3)

$$P_{\text{val}} = \frac{R_{\text{par}} \left[\left(\frac{0.7V \cdot 30.1k\Omega}{V_O - 0.7V} - 6.49k\Omega \right) - R_{\text{ser}} \right]}{R_{\text{par}} + \left(\frac{0.7V \cdot 30.1k\Omega}{V_O - 0.7V} - 6.49k\Omega \right) - R_{\text{ser}}} \cdot \frac{256}{10k\Omega} \quad (2.15)$$

inside the code used for Regulation, Equation 2.15 will be used to create a lookup table, while Equation 2.17 will be used at runtime, for which Equation 2.14 and 2.16 are needed.

$$V_{\text{dip}} = V_O - V_{\text{off}} \quad (2.16)$$

$$\Rightarrow V_O = I_{\text{ret}} \cdot (R_1 + R_0 \cdot n_{\text{ret}}) + V_{\text{off}} \quad (2.17)$$

2.3.3 Distance Wafer Resistance Model

Although the through SWRM gained functions are useful for determining a theoretical Regulation procedure, it is still not near the realworld scenario. In a wafer, the distance between reticles and voltage connector (see Figure 2.7) are resulting in additional resistance, proportionally.

Therefore we adapt the DWRM after Circuit 2.9 in which each different Distance requires additional Resistors.

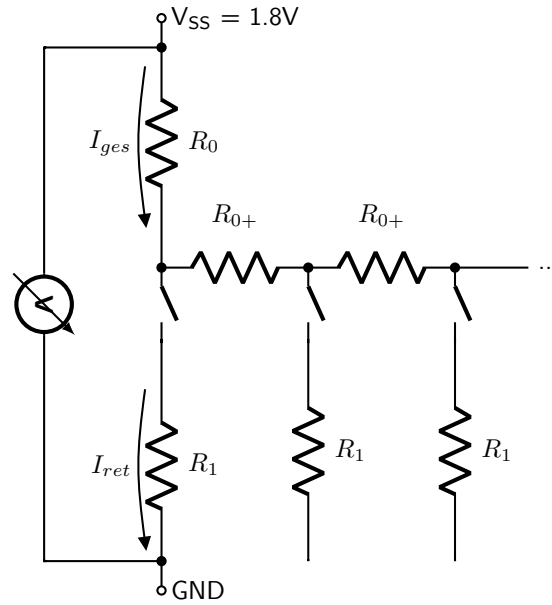


Figure 2.9: retpow2

With this model the voltage is now expected to change depending on the reticles distance instead of being the same. The distances inside a wafer are visualized in Figure 2.10

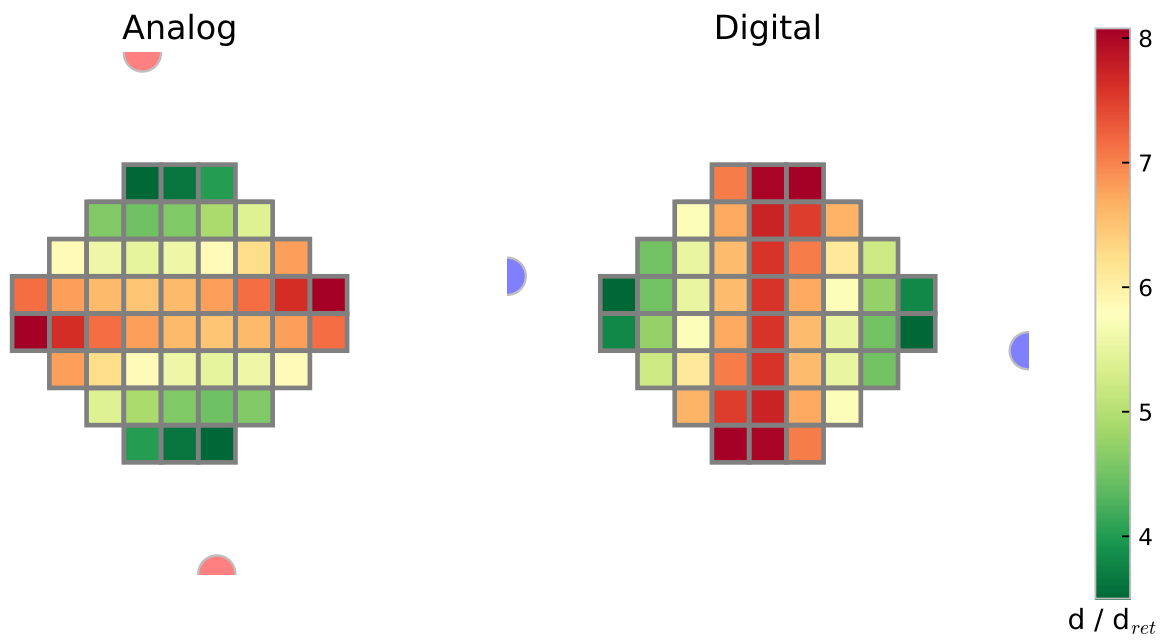


Figure 2.10: Distances of reticles to the nearest voltage supplying connection for DWRM, distance is in reticle-side length

3 Experiments

Now that the theoretical model is complete, experiments can be done to start checking that model and get results to use for in system components.

3.1 Characterization

The first experiments to run are the characterization of hardware behavior. These will then result in a PowerIt Calibration, which later then can be used as basis for creating a regulation method.

3.1.1 sampling time

First up was selecting an optimal number of cycles for which the adc will probe a to it connected pin, like described in subsection 2.2.1.

In this case the uncalibrated measurement of input voltage was taken as example, and repeated with each of the possible 8 settings.

To be able to compare a reference voltage measurement was taken with an external Voltmeter¹. The resulting errors, from a set Voltage, can be seen in figures 3.1

Figure 3.1 contains the absolute error of the measured voltage compared to the theoretical, set input voltages. therefore the reference measurements (yellow), taken with an external Voltmeter, are not at 0. Also shown are the calculated gain errors, in case of all 8 settings.

Important to note is the relative error in only the 0th case, here the `cycleTime`-Setting was set to 0 and therefore the smallest available sampling time of 3 Ticks. This excludes 0 a possible value to use. All other measurements are within error margin of each other, and because a smaller time frame is preferred, the best value to use is 1, resulting in a measure time of 15 Ticks.

3.1.2 Voltages

Now that a sample time is chosen, it is possible to proceed with the voltage calibration measurements. Note, that measurements are expected to be less accurate, the more components are contained in their respective measurement circuit. Because small errors will accumulate and in e.g. the case of 48V's be amplified by a factor of 8.

¹Keithley K2100

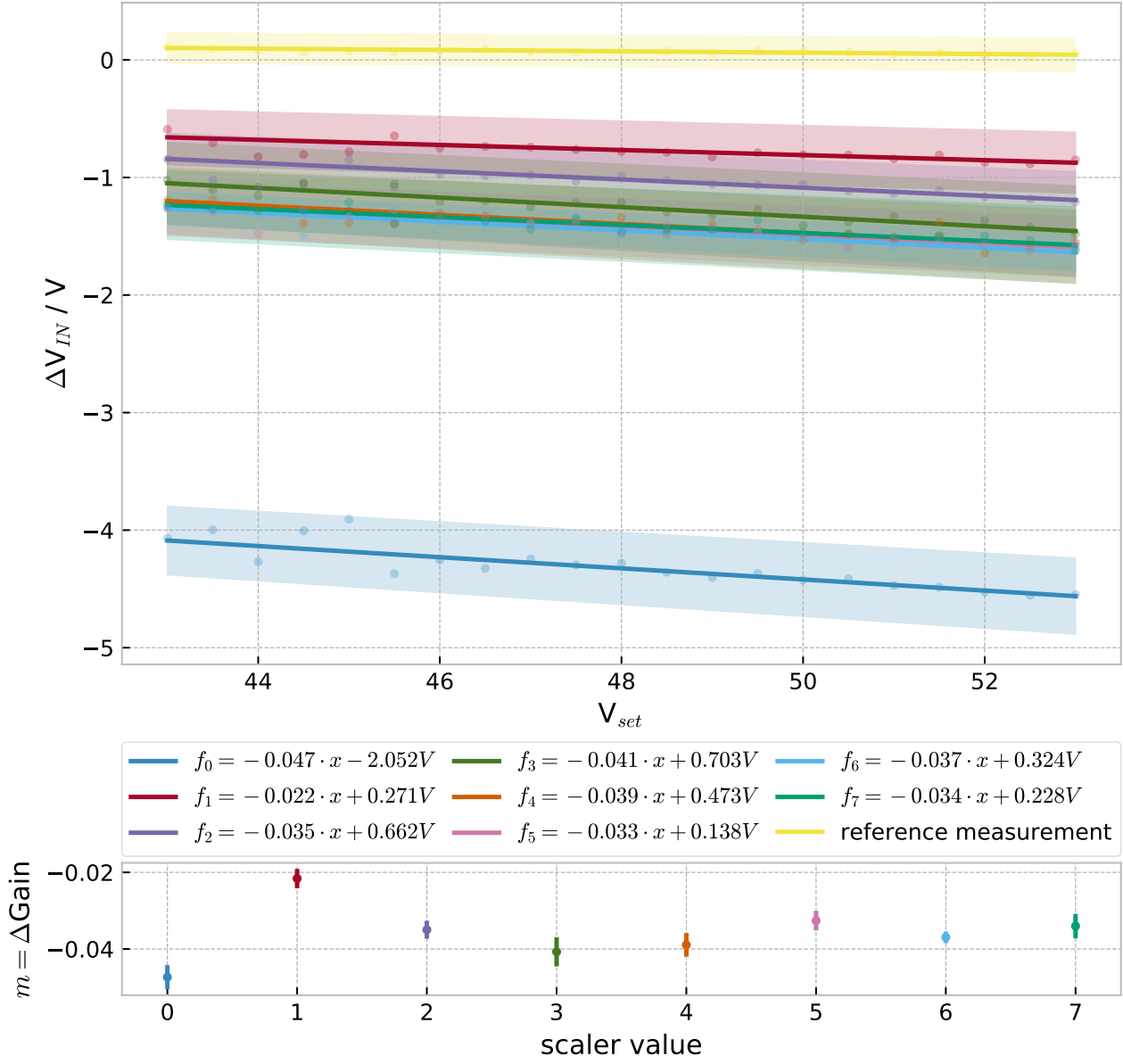


Figure 3.1: up: input difference from set voltage vs set voltage for different possible scaler values; down: gain error of the linear fitted curves vs set scaler value (May 29th 2018, $\approx 32^\circ\text{C}$)

48V Input

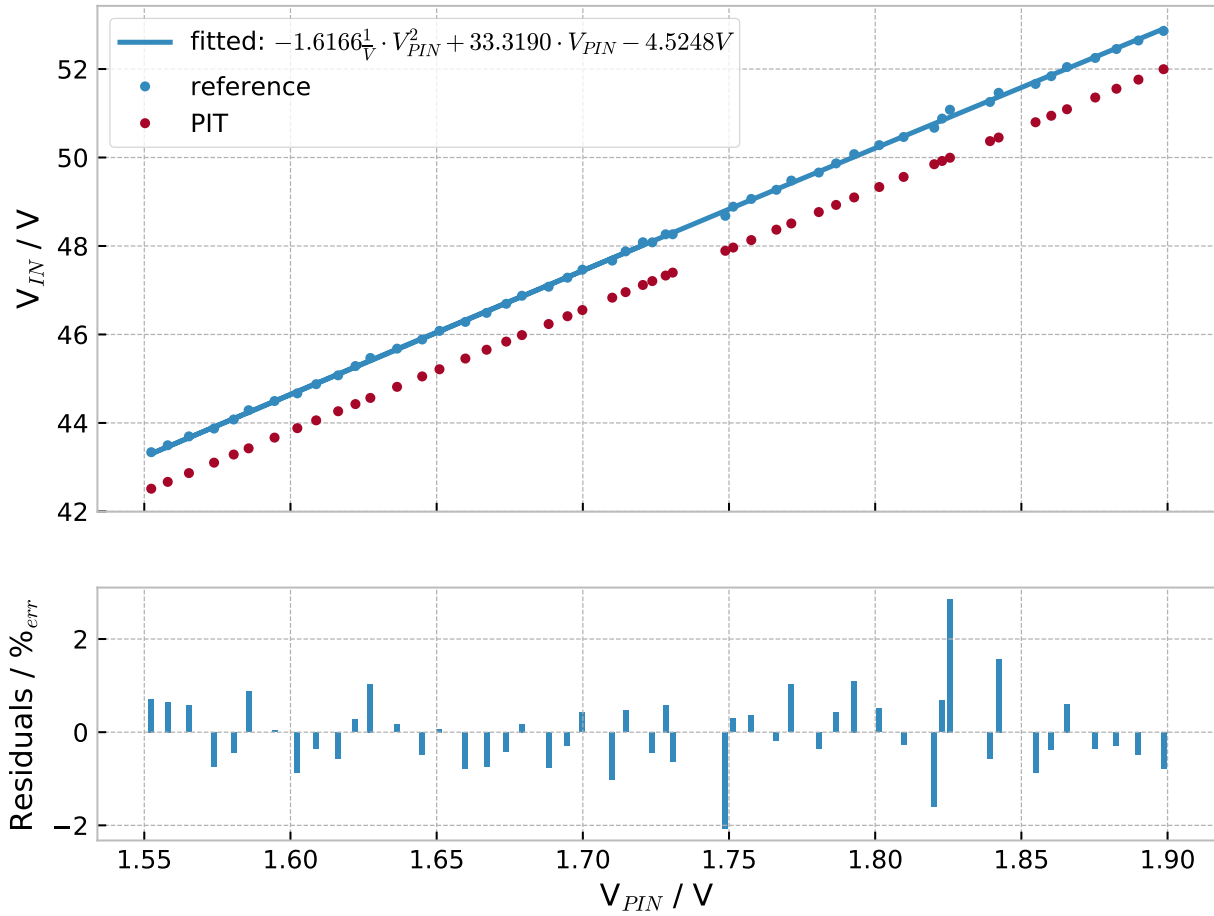


Figure 3.2: Calibration of input voltage, plotted are a external measurement and internal values, vs the recalculated pin voltage based on the internal value and used default function (default coefficients see Figure 4.2)

When looking at calibrating the input voltage (Figure 3.2), we can clearly see a relatively constant offset of $\approx 1V$. In Figure 3.2 a polynomial fit of 2nd degree² is done and its coefficients extracted (Figure 4.2.1, line 9). These coefficients not only show an offset, but also some deviation in the incline and curve from the default values.

²A Fit of second degree will be used in the complete calibration process

9.6V Output

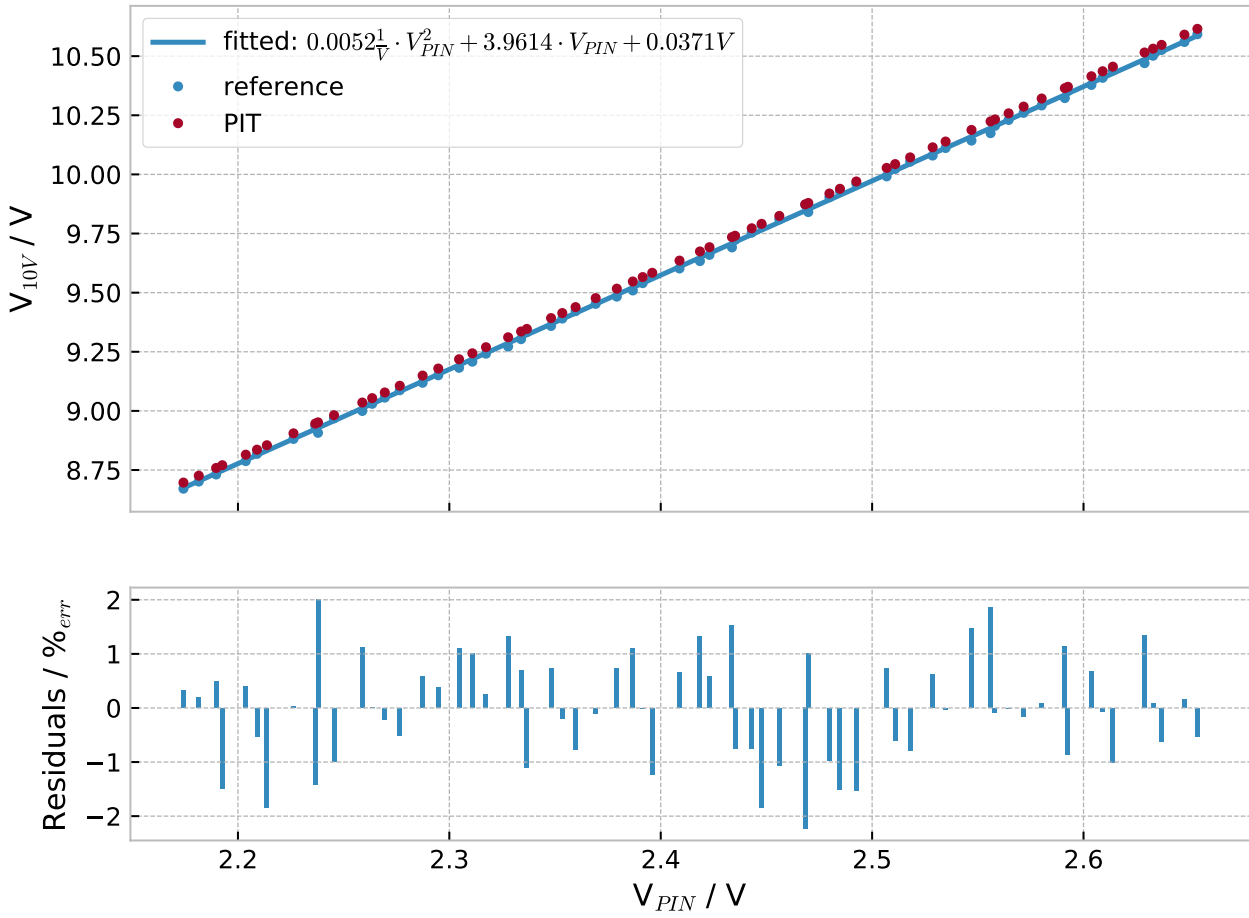


Figure 3.3: Calibration of 9.6V output Voltage, plotted are an external measurement and internal values vs the recalculated pin voltage based on the default coefficients (Figure 4.2)

The 9.6V Calibration, in contrast, shows only a slight deviation of the internal values and the reference measurement, which results in a list of coefficients (Figure 3.5, line 7), very similar to those set in the theoretical defaults.

This small difference is explained by the simple voltage division used as our circuitry, and no amplification, as for the input voltage circuit.

1.8V Output

The last Voltage to calibrate is divided into two domains, one for supplying the analog circuitry inside the wafer system, and one for the digital side. Each deliver between 1.5 and 2.022V and each is settable by its own circuit (both as in Figure 2.3).

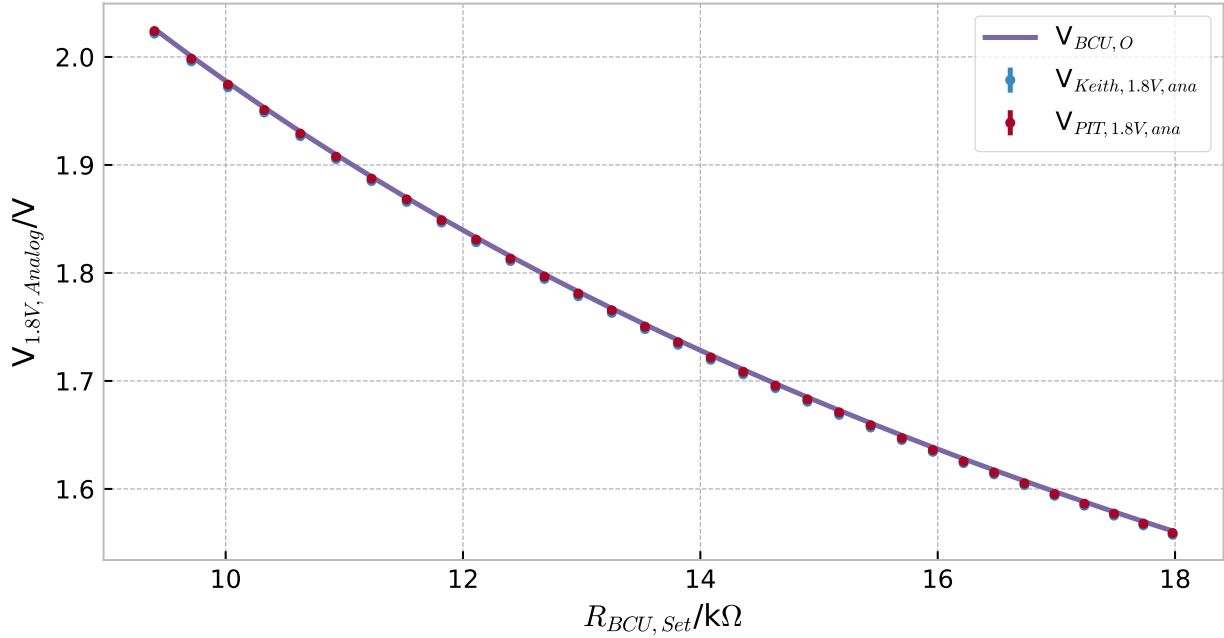


Figure 3.4: Calibration: analog 1.8V Output voltage, plotted are external measurement and internal values vs set resistance at the BCU Voltage Module.

Visualized in Figure 3.4 is the analog domains calibration, showing nearly no difference in board and reference measurements. Mostly due to direct connection between created voltage and the STM-Chips pin.

3.1.3 Currents

With now calibrated Voltages, the next step is to measure the behavior of the measuring circuits. Note that the 9.6V Output does in fact not have a include circuit for measuring its current draw, and that this number will be obtainable from all other (calibrated) measurements.

48V Input

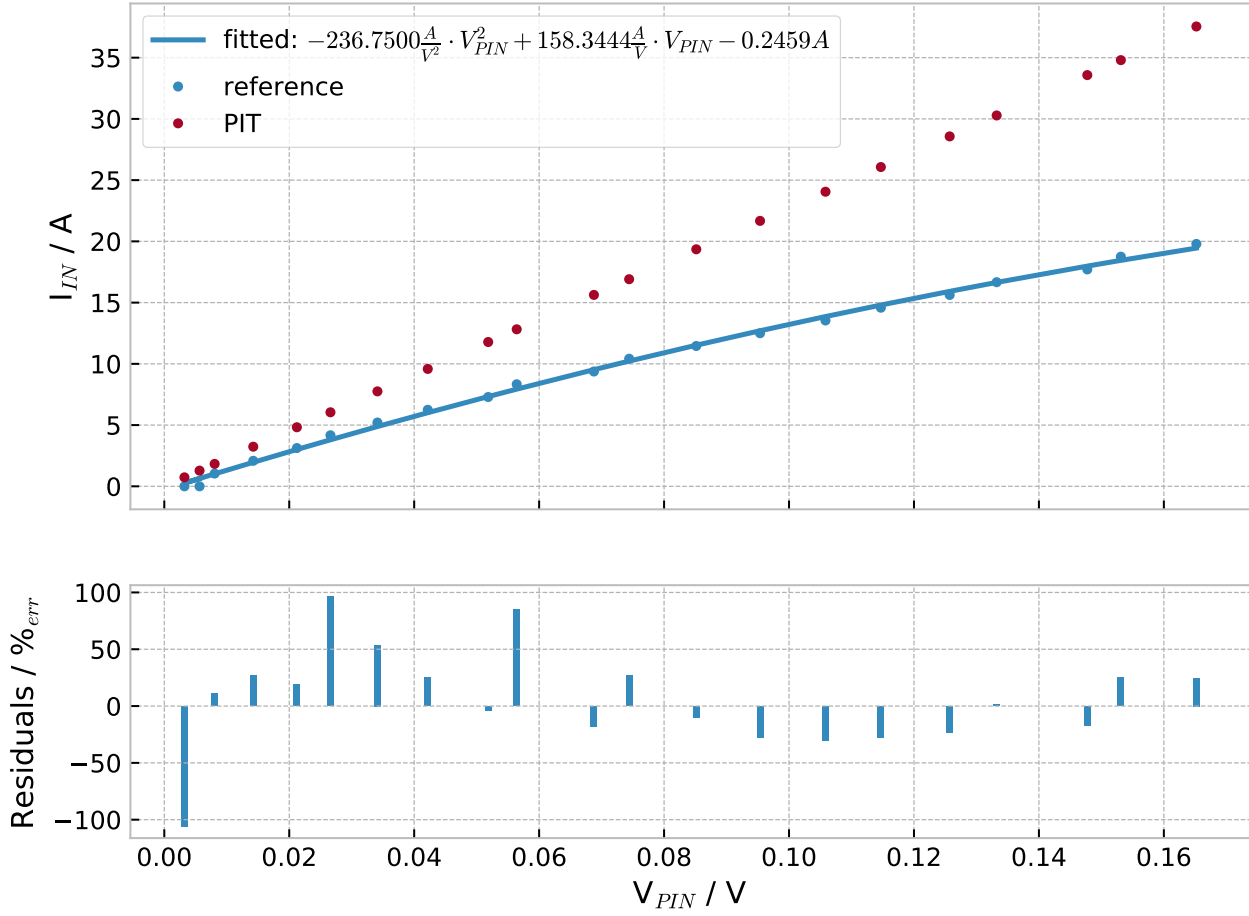


Figure 3.5: Calibration of input current ADCs 21.06.2018

1.8V Output

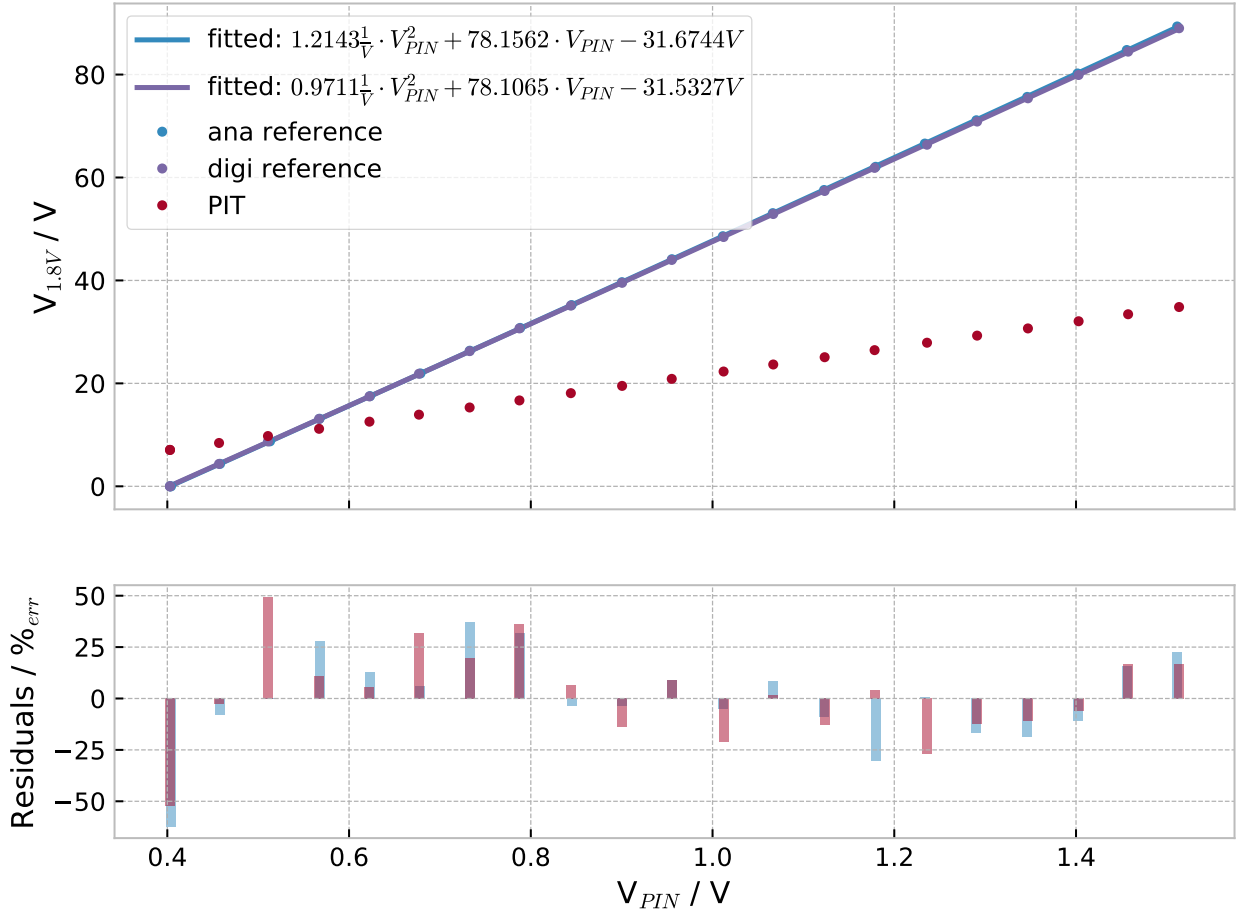


Figure 3.6: Pre Calibration Measurement of Output Current at the 1.8V Analog and Digital Terminal (2.7.2018)

3.2 1.8V Regulation

As Described beforehand the Output Voltages for both analog and digital can be adjusted to some degree and therefore we can compensate for the dropoff occurring between Powerlt Output Terminals and Reticles.

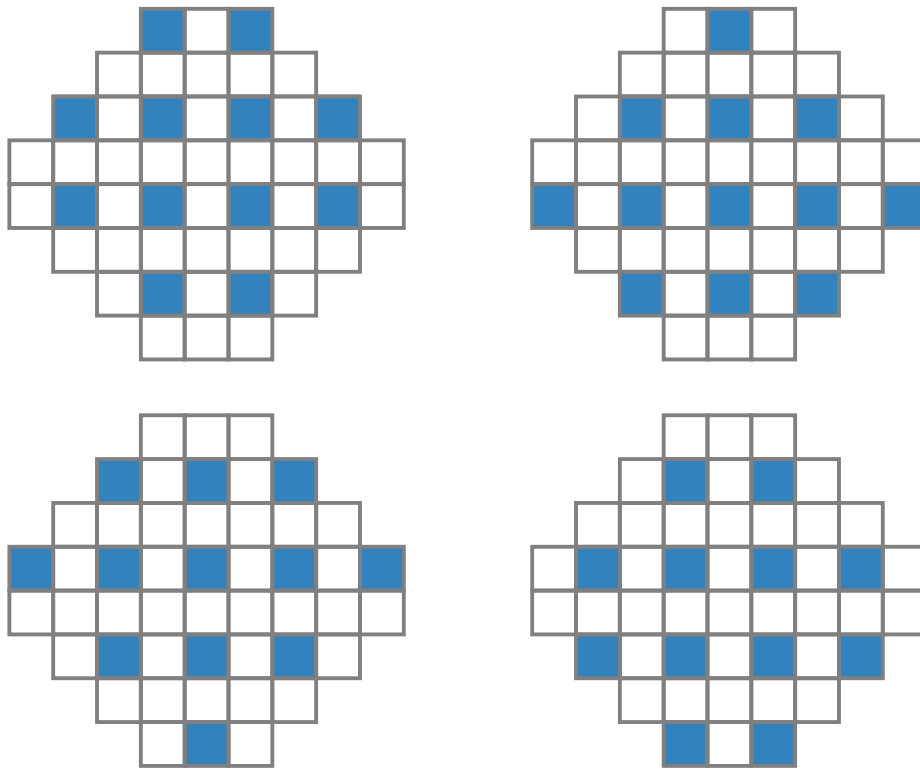


Figure 3.7: Used regular patterns for current tests on PowerWafer

3.2.1 Characterization of Dropoff

Wanting to observe and characterize the voltage drop, first the connections between PowerIt and Reticles can be measured with the in Figure 2.8 described connections, which in actuality are the PowerIT Terminal and corresponding analog readout pin on a Analog readout board.

To use the PowerWafer for testing one of the patterns in Figure 3.7 will be used, each pattern has a approximate current draw of 120A and will distribute heat and draw per terminal evenly.

In Figure 3.8 a single reticles (40) Voltage Dip for different Current Draws is visualized. A relatively linear trend and residuals of a trigonometric behavior can be observed (most likely the result of the inaccurately measurable current draw, which here is done inside the PowerIt).

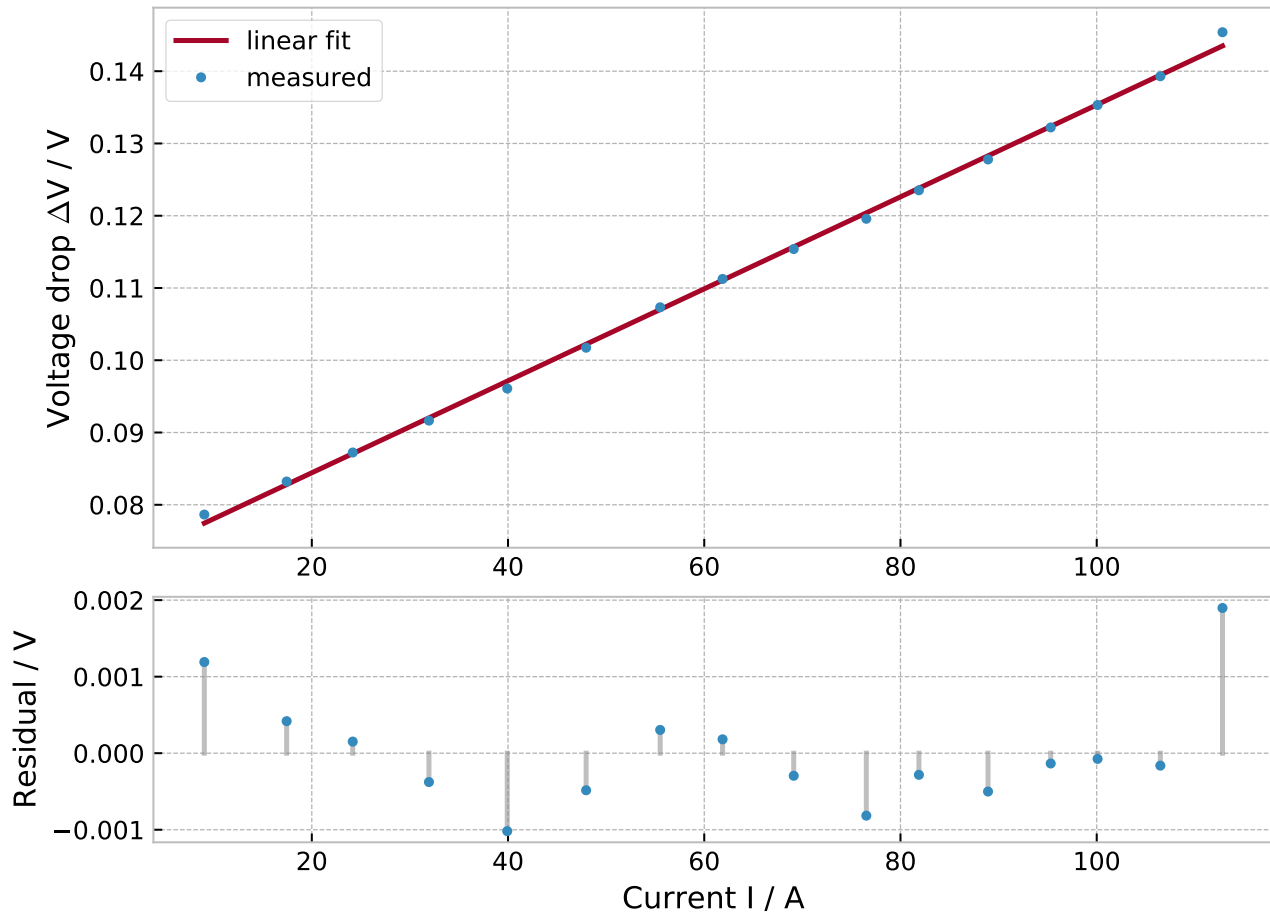


Figure 3.8: Voltage dip observed between PowerIt and HICANN, each point represents a state after enabling additional Reticles on the PowerWafer (right upper wafer in Figure 3.7)

Here a Voltage Drop vs. Current draw of the wafer shows a linear behavior and therefore can be regulated on basis of the current measurement done by on board Measurement circuit.

3.2.2 Numerical-Correction (Regulation)

The initial idea, to approach the correction of this dropoff is a Numerical: the SWRM (subsection 2.3.2) and its corresponding Equations can be applied here. Equation 2.15, which maps the measured output current to a corresponding potentiometer setting, requires the

Dropoff to be linear, which was observed.

To apply this approach, two assumptions need to be made:

- all reticles have the same current draw (already not accurate, see Figure 3.8)
- all reticles experience the same Voltage Dip (as observed for reticle 40)

and the following four values are required, before a regulation can be attempted:

- I_{ret} , the current draw of a single reticle,
- R_0 , the resistance between PowerIt and FET,
- R_1 , the resistance of a single Reticle
- V_{off} , the wanted Voltage at a Reticle

To get a representative value of I_{ret} for use in the SWRM, the mean of all reticles current draw was taken (Figure 3.9):

$$I_{ret,mean,corr} = (9.9089 \pm 0.6049) \text{ A} \quad (3.1)$$

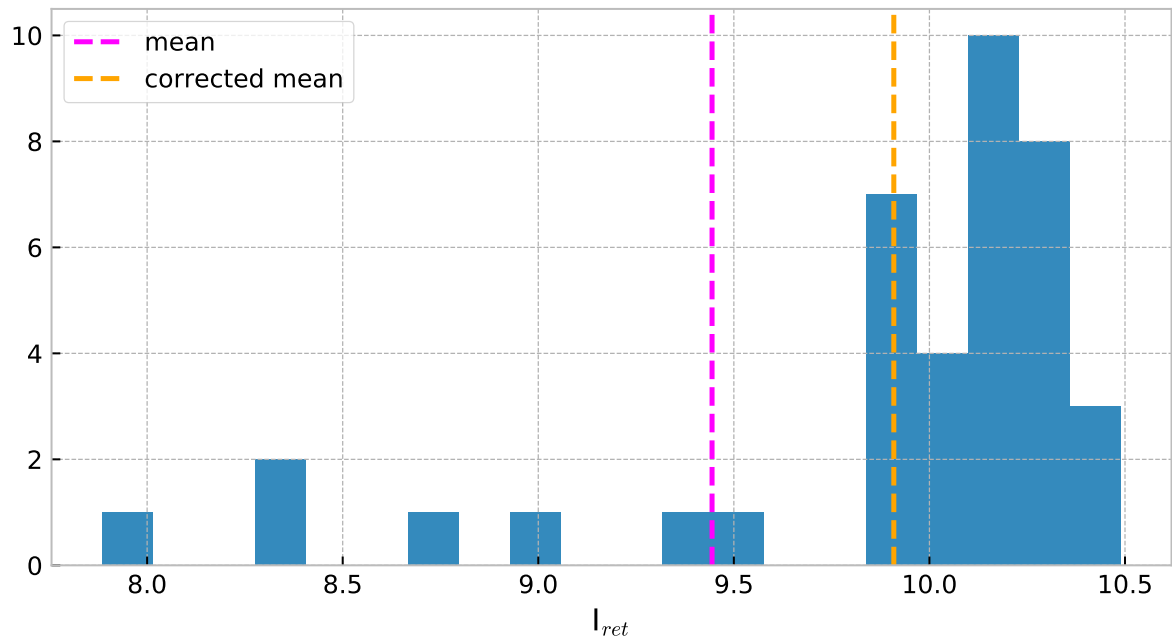


Figure 3.9: Distribution of analog current draw for all reticles on the PowerWafer (which were possible to measure → section 3.3)

The Figure 3.9 was obtained by measuring the increase in current draw for each reticle, for each of the 4 patterns (Figure 3.7).

To obtain R_0 , the pattern in Figure 3.10 was used to take measurements for both the Neighborhood as well as the Farthest Reticles.

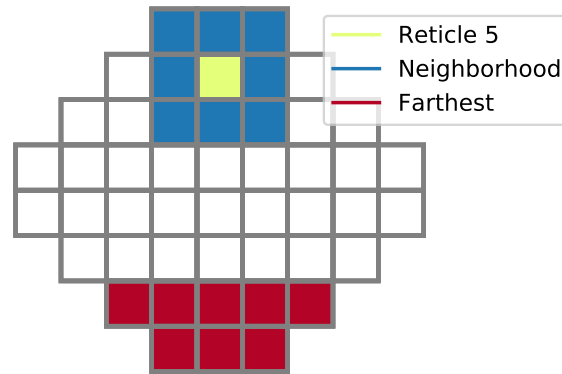


Figure 3.10: Reticles used to determine correlation between distance and Voltage Drop

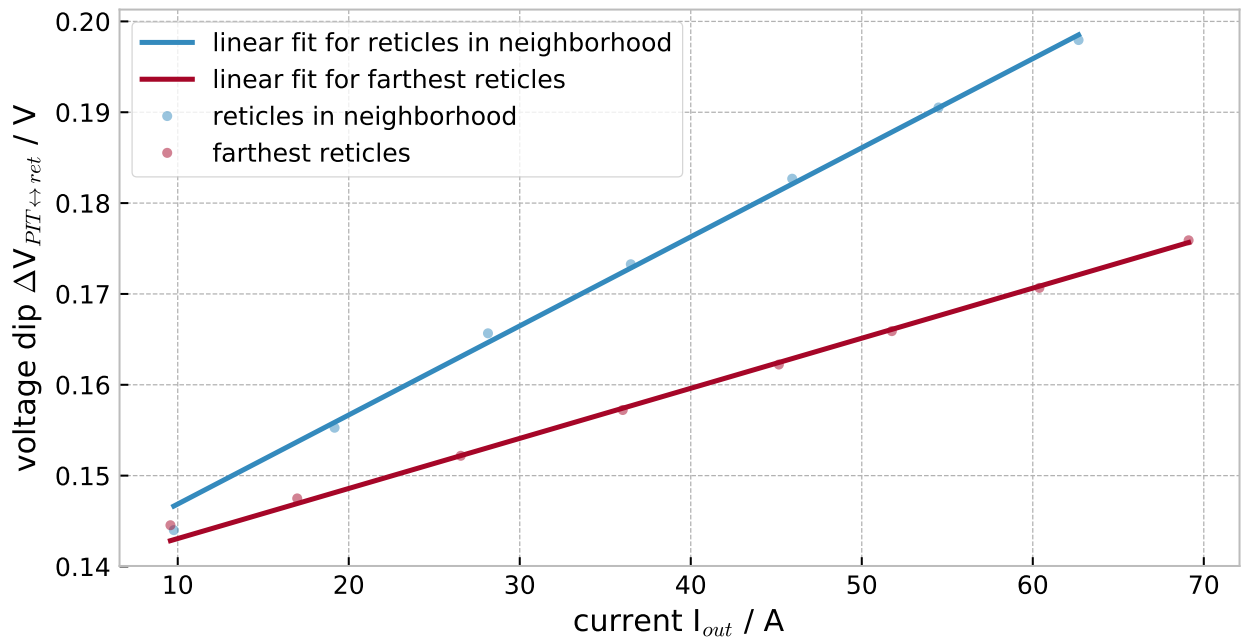


Figure 3.11: Voltage Dip vs current for both Reticles in direct neighborhood and farthest possible Reticles

From Figure 3.11 it is possible to see that the distance between reticles that are used gives different behavior of the Voltage Dip. Both Inclines happen to be the extreme cases, while either being completely uncorrelated, the case for farthest Reticles, or being directly correlated by their distance, here observable for the neighboring Reticles.

Therefore we obtain two values for R_0 :

$$R_{0,neighbor} = (7.1278 \pm 0.1567) \text{ m}\Omega \quad (3.2)$$

$$R_{0,farthest} = (4.0079 \pm 0.0537) \text{ m}\Omega \quad (3.3)$$

from the same measurement it is also possible to extract R_1 by extrapolating to 0, which results in:

$$R_{1,neighbor} = (14.1708 \pm 0.1779) \text{ m}\Omega \quad (3.4)$$

$$R_{1,farthest} = (14.2218 \pm 0.1503) \text{ m}\Omega \quad (3.5)$$

here the values obtained are within error margin of each other.

So applying these Values, the following behavior for regulation can be visualized:

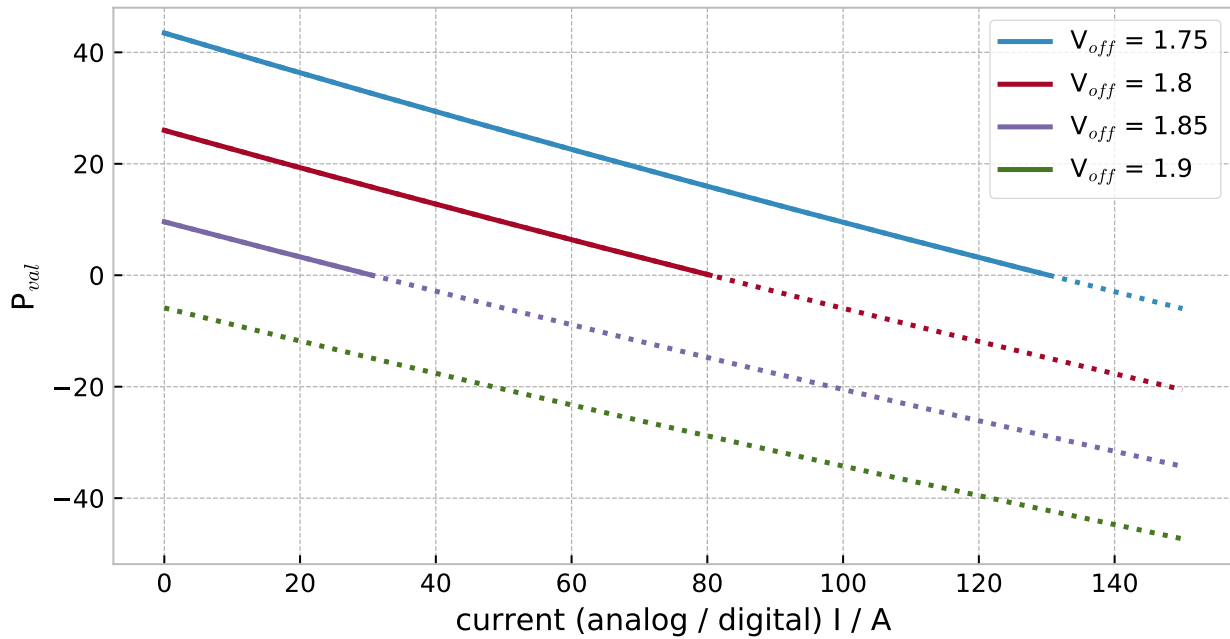


Figure 3.12: possible P_{val} curves after SWRM, dotted lines represent not achievable values

The in Figure 3.12 visualized values show the theoretical P_{val} for the corresponding Current, while all dotted parts depict the values which would be needed to achieve full correction at the Reticle level. Note that the 1.8V regulation, should fail at about 80A of current draw.

Now that the SWRM is applicable, what about the DWRM, which removes the assumption of a equal Voltage Dip per Reticle, applying an offset to the initially observed Voltage of each Reticle.

To account for that, the Voltage Dip per Reticle, in a single Reticle power state, was observed:

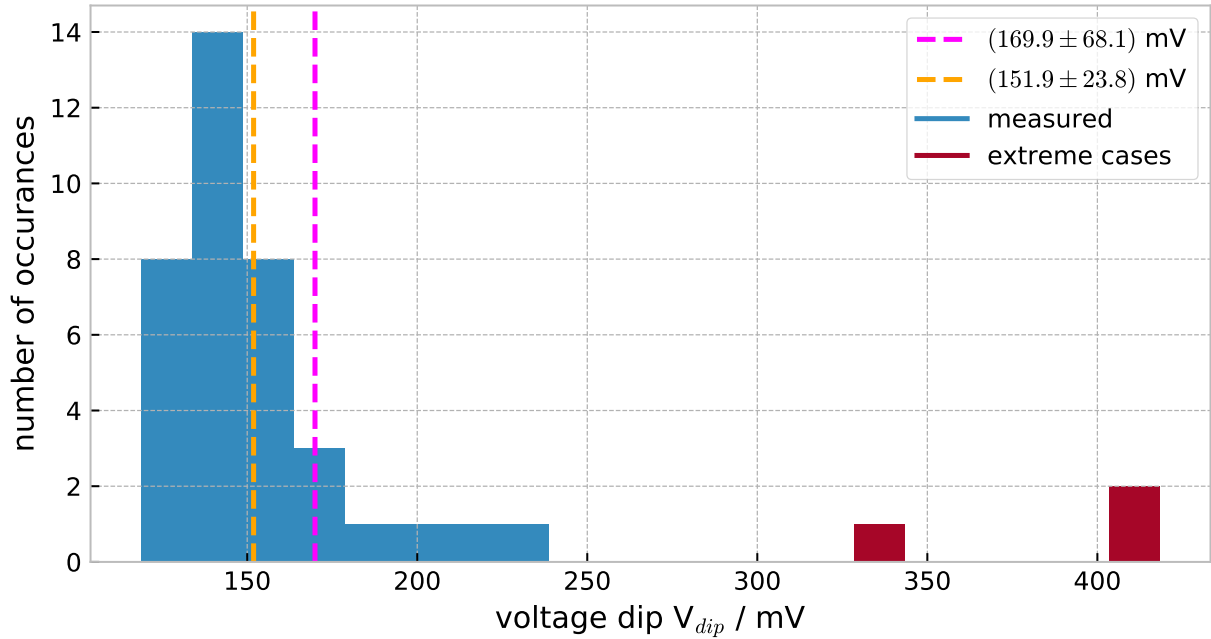


Figure 3.13: initially observed Voltage Dip, Red values are ignored for corrected mean

and a mean of:

$$V_{dip,mean,corrected} = (151.8811 \pm 23.8138) \text{ mV} \quad (3.6)$$

can be observed.

Figure 3.14 shows how those Voltages are Distributed over the complete PowerWafer. All white Reticles are not measurable, and those colored in Red and Yellow are the outliers in Figure 3.13. Note that in a deployed, working, Wafer System inside BrainScaleS the middle two Reticles (19 & 28) are not used and also give grounds to ignoring the outliers.

This results in a distribution, which when combined with the spread of R_0 from Figure 3.11, gives an approximate range for all Reticles Voltage Dip at a given Current Draw (Figure 3.15).

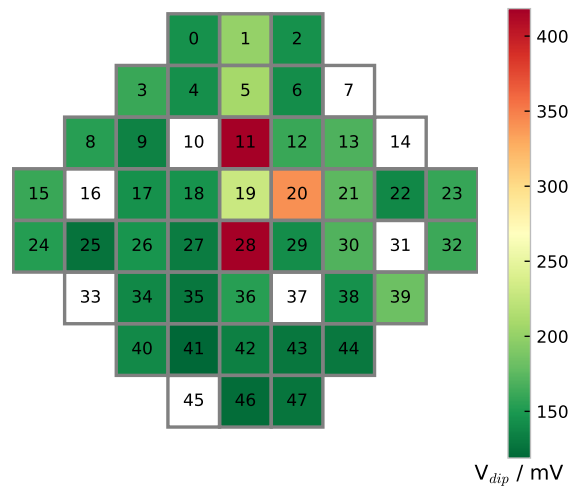


Figure 3.14: V_{dip} Distribution over full Power Wafer; White have no measurement; Red and Orange are marked red in Figure 3.13

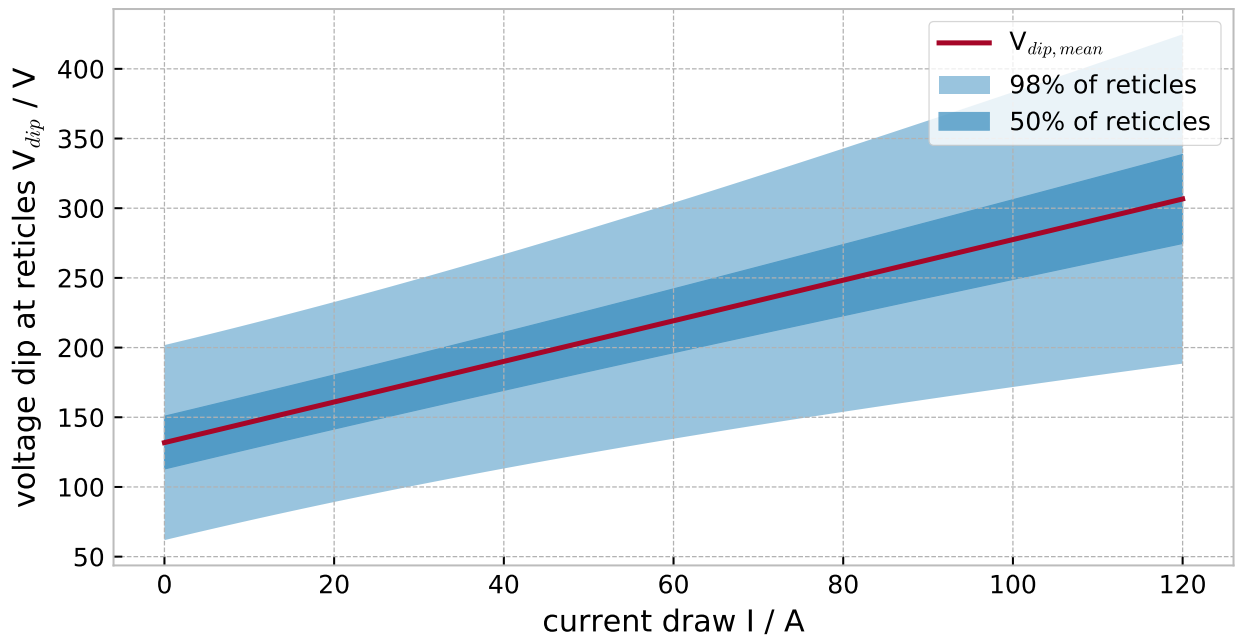


Figure 3.15: Experimentally obtained Voltage Ranges in which most Reticles Voltage Dip will lie, this does not include outliers

3.3 Pitfalls

4 Results

This Chapter summarizes all of the resulting Workflow that has been developed during this Bachelor Thesis Work. Mainly the Firmware Changes compared to the state at which it was left off after the previously taken Internship [3]

4.1 Firmware

The PowerIt Firmware was updated to allow for the Calibration Procedure (described in 4.2) and the Regulation of its 1.8V output. That resulted in a new Version of the I2C Protocol between the Monitoring System and PowerIts.

4.1.1 I2CProto v2

The new Communication protocol is based on the old one, and while the from the Host send Commands are compatible with version 1 the returned Message is not. And just like in the old Version a Master can send a message to the PowerIt with the in the `command_t` struct described structure:

```
pi2c::command_t tosend { <crc>, 2, <CMD>, <optional_data0>,  
↪ <optional_data1> };
```

Here the `<CMD>` can be `CMD_SET` for setting a value in the corresponding table (fig. 4.1), using `<optional_data0>` as address and `<optional_data1>` as value.

Reading a number of bytes from the table can be accomplished with `CMD_READ`, giving the address to start and number of bytes in that order.

While reading the complete Table is done with the `CMD_READALL` command.

The `<crc>` byte is the CRC8 value of all following bytes inside `command_t`

4.1.2 I2C mapped Register-Table

When sending or reading values to or from the PowerIt, the address required is one of the 152 bytes documented in this table.

addr	name	type	size	perm
0x00	onmask	byte	1	rw
0x01	offmask	byte	1	rw
0x02	anapot	9bit	2	rw
0x04	digipot	9bit	2	rw
0x06	polyFit.V48	float arr	12	rw
0x12	polyFit.I48	float arr	12	rw
0x1e	polyFit.V8	float arr	12	rw
0x2a	polyFit.V18	float arr	12	rw
0x36	polyFit.I18	float arr	12	rw
0x42	polyFit.T	float arr	12	rw
0x4e	sampleTicks	byte	1	rw
0x4f	V_out	float	4	rw
0x53	TEMP_SENSOR	float	4	r
0x57	EXT_AIN	float	4	r
0x5b	MONITOR_48V	float	4	r
0x5f	MONITOR_48I	float	4	r
0x63	MONITOR_8VBUS	float	4	r
0x67	MONITOR_8IBUS	float	4	r
0x6b	MONITOR_8V_0	float	4	r
0x6f	MONITOR_8V_1	float	4	r
0x73	MONITOR_8V_2	float	4	r
0x77	MONITOR_8V_3	float	4	r
0x7b	VDD_1V8_ANA	float	4	r
0x7f	VDD_1V8_IOUT_ANA	float	4	r
0x83	VDD_1V8_DIGI	float	4	r
0x87	VDD_1V8_IOUT_DIGI	float	4	r
0x8b	CommitHash	float	4	s
0x8f	CommitDirtyFlag	byte	1	s
0x90	STM32UUID	96bit	12	s

Figure 4.1: memory mapping of the packed struct moved over i2c, `addr` is the address to use, `type` is the c++ type, `size` is in bytes and `perm` denotes read-writability. writability

4.2 Calibration

One of the goals of this Bachelor Thesis is to provide anyone required to calibrate a PowerIt Board with a comprehensive guide of, and inside into this process.

```

---
id: '0x280029000F51333332343638'
name: B05
poly18iana: [-31.5155, 78.516, -0.0688]
poly18idigi: [-31.3536, 78.3701, -0.196]
poly18i: [-31.4396, 78.443, -0.1324]
poly48i: [-0.1765, 153.0021, -204.1858]
poly10v: [0.6348, 3.459, 0.1118]
poly48v: [-4.5248, 33.3195, -1.6167]
poly18v: [0.0234, 0.9728, 0.0072]

```

4.2.1 Calibration-Table

Of important note is that all calibration values of each PowerIt can be mapped via the naming and uuid scheme provided either by the corresponding stickers or the STM32-Chips internal uuid (accessible through address 0x8c, see figure 4.1). There now also exists a global calibration Database, which will be loaded by the system on startup.

An example entry for each PowerIt entry looks like figure 4.2

Figure 4.2: example entry of pitdb.yaml

```

---
uuid: 'default'
name: 'Bxx'
poly18i: [-3.0, 25.0, 0.0]
poly48i: [0.0, 227.27, 0.0]
poly10v: [0.0, 4.0, 0.0]
poly18v: [0.0, 1.0, 0.0]
poly48v: [0.0, 27.386, 0.0]

```

4.2.2 How to calibrate a PowerIt Board

The Calibration process is based on the PltSTOP Python scripts¹. These are split into server and aggregator. While the Server is handling the translation between raw I²C data, and the JSON formatted result, the Aggregator takes this JSON and calculates a calibration.

Using the script any one of the following Values can be tested and calibrated:

- Input Voltage (`pitstop.Aggregator.test_v_48()`)
- Input Current (`pitstop.Aggregator.test_i_48()`)

¹ PltSTOP Repo

- 9.6V Output Voltage (`pitstop.Aggregator.test_v_10()`)
- 1.8V Output Voltage (`pitstop.Aggregator.test_v_18()`)
- 1.8V Output Current (`pitstop.Aggregator.test_i_18()`)

Setting up the Test Environment

The simplest way to setup your environment consists of cloning the PitSTOP Project onto your Client:

```
> git clone https://url.to.pitstop
```

then substituting the `rsync` target:

```
all:
    rsync --progress ./*.py /remote.url/
```

, to be your server (should be a RaspberryPi connected to the PowerIt)

Running a Test

Running the test requires the following commands

Serverside:

```
> python server.py
```

Clientside:

```
> python aggregator.py
```

Now just following the instructions given, the selected test can be run:

```
Setting up calibration test for {}
Please be sure to:
- connect the {} to the RaspberryPi running server.py.
- connect the PowerIt to the RaspberryPi as described in the
  ↪ documentation
- and be sure to connect the {} to the {} Terminal.

Continue (y/N): y

What is the Name given to the connected PowerIt? [Bxx]: B05
```

4.3 Regulation

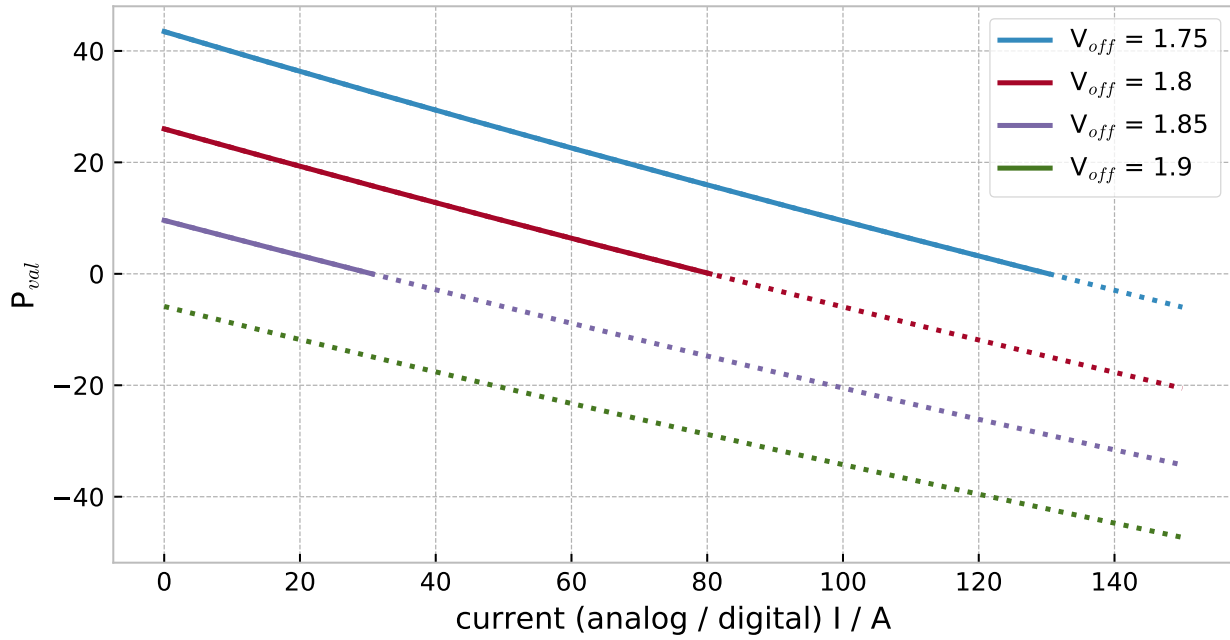


Figure 4.3: theretical potentiomete settings for meausured current

5 Outlook

Bibliography

- [1] A. P. Davison, E. Müller, *et al.*, *HBP Neuromorphic Computing Platform Guidebook*. Human Brain Project, 2015-2018. EV Guide [↗](#).
- [2] A. P. Davison, E. Müller, *et al.*, *HBP Neuromorphic Computing Platform Guidebook - BrainScaleS Hardware Configuration*. Human Brain Project, 2015-2018. https://electronicvisions.github.io/hbp-sp9-guidebook/pm/pm_hardware_configuration.html.
- [3] P. Nisble, "Firmware upgrade des powerit (de)," 2018. [↗](#).
- [4] STMicroelectronics, *STM32F405xx Datasheet (DM00037051)*. STMicroelectronics, 2016. STM32F405RG.
- [5] G. di Sirio. ChibiOS RTOS [accessed 2018/01/11].
- [6] TexasInstruments, *AMC1200/B Fully-Differential Isolation Amplifier*, 2015.
- [7] Texas Instruments, *PTH08T250W Datasheet*, 2017.
- [8] D. Husman, *Analog Pin Distributions*. electronic visions, Universtity of Heidelberg, 2016.