

Development of a cryogenic platform with fast-turnaround in-situ characterization of semiconductor circuits for qubit applications

Zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

von der KIT-Fakultät für Physik des
Karlsruher Instituts für Technologie (KIT)

angenommene
Dissertation

von
M.Sc. Julian Ferrero

aus Karlsruhe (Baden)

Tag der mündlichen Prüfung: 03. Mai 2024

1. Referent: Prof. Dr. Wolfgang Wernsdorfer

2. Referent: Prof. Dr. Hendrik Bluhm

Contents

List of Figures	iii
List of Tables	ix
1. Introduction	1
2. SiGe heterostructures	5
2.1. Bandstructure of crystalline Si	5
2.2. Bandstructure of strained SiGe quantum wells	6
2.3. Tunneling into the cap	12
2.3.1. The Tsu-Esaki model	12
2.3.2. Fowler-Nordheim-Tunneling	15
2.4. Gate defined Quantum Dots	18
3. Hardware	23
3.1. DC-Wiring	23
3.2. Filtering	26
3.3. Radio-frequency suited wiring	29
3.4. Magnetic Field coils	33
3.5. The $^3\text{He}/^4\text{He}$ dilution fridge	36
3.6. Turbocycle heating system	38
3.7. Experiment wiring	40
4. Measurement software	45
4.1. The concept of measurement	45
4.2. Common software components	45
4.3. Non-realtime manipulation	47
4.4. spin_tune	48
4.4.1. Usage	48
4.4.2. Use cases	49

4.5.	spin_watch	51
4.5.1.	Usage	52
4.5.2.	Use cases	53
4.6.	Realtime manipulation	55
4.7.	spin_excite	56
4.7.1.	Usage	57
4.7.2.	Use cases	65
5.	Bias cooling of SiGe heterostructures	73
5.1.	Introduction	73
5.2.	Device layout	73
5.3.	Turn-on voltage mapping	75
5.4.	Single electron transistor (SET) formation	78
5.5.	Noise measurements	79
5.6.	Schrödinger-Poisson Simulations	87
5.7.	Summary	94
6.	Conclusion and Outlook	97
	Bibliography	103
A.	Appendix	117
A.1.	Documentation	117

List of Figures

2.1.	Conventional unit cell of the silicon lattice [26].	5
2.2.	Brillouin-zone of the FCC-lattice, adapted from [103].	6
2.3.	Bandstructure of silicon (a). Iso-energy surfaces of charge carriers in the silicon conduction-band minima (b). Adapted from [37].	7
2.4.	Layer stack of the SiGe heterostructure used to confine electrons in the z-direction. The n++ regions are heavily doped with phosphor, and the silicon substrate is grown in (100) direction.	8
2.5.	Partial lifting of the degeneracy of the Δ_6 conduction band minimum by induced strain (a). Resulting band alignment in a SiGe heterostructure (b). With n being the electron density inside the channel, Δ_2 and Δ_4 are the edges of the two lowest lying conduction bands, $E_{0,channel}$ and $E_{0,cap}$ are the lowest subbands in channel or cap respectively and η is the electrochemical potential (Fermi-energy) of the system.	9
2.6.	Band-edge alignment of the heterostructure in the zero-bias case (a) and in the accumulated case (b).	11
2.7.	Relevant energy levels and distribution functions for electron tunneling in between two degenerately doped n-type semiconductors. With η_1 and η_2 being the electrochemical potentials of the semiconductors, E_{C1} and E_{C2} the corresponding conduction band edges, f_1 and f_2 the Fermi distribution functions, and Φ the height of the potential Barrier.	13
2.8.	Band alignment at the onset of significant tunneling currents into the cap, with J being the total tunneling current density.	15
2.9.	Schematic depiction of a general quantum dot device (a). The white-blue regions exemplify the tunnel-coupling in between source and drain. Equivalent circuit model of the system sketched above (b).	19
2.10.	Schematic drawing of a quantum dot with and the corresponding energy levels.	20

2.11. Schematic depiction of Coulomb-diamonds and the corresponding energy-level alignments on the quantum dot.	21
3.1. The anatomy of a wirebundle.	24
3.2. Transmission function of two DC-lines in series. The color gradient indicates the temperature of the wire, red being a mixing chamber temperature of 300 K and blue being a mixing chamber temperature of 28 mK.	25
3.3. An RC-filter circuit which includes the parasitic inductance of the traces connecting the individual components.	26
3.4. Transmission through a real RC-filter.	27
3.5. Circuit diagram of the filtering applied to each input line.	27
3.6. Layer stack a) front b) and thermalization layer c) of the filter PCB. In a) only the copper layers (shiny metallic and black) are counted.	29
3.7. The filter PCB. It is partially assembled and inserted into its housing.	29
3.8. Pulse shape used in e.g. free induction decay experiments.	30
3.9. Supposed filter function of the fridge cabling.	31
3.10. Pulse-shape reaching the sample.	31
3.11. Wiring diagram of the input RF-lines	34
3.12. Spatial distribution of the magnetic field of the coils in x and y direction. The plot shows the color-coded absolute value of the magnetic field.	36
3.13. Magnetic field distribution of the z-coil and visualization of the quench estimation.	37
3.14. a) 1, the dilution fridge. 2, 4 K-shield. 3, mK-setup. A detailed view is found in 3.15. b) The sample holder unit. 4, cold finger. 5, the actual sample. 6, split-PCI-connection for up to 24 DC-lines. 7, SMP-connectors for up to 3 RF-contacts. 8, radiation shield hosting the second filtering stage.	38
3.15. The mK setup. 9, the sample holding assembly inserted upside-down into the mK-setup. 10, elevated mK-plate 11, coaxial cabling for RF-signals. 12, superconducting 3D-vector magnet. 13, thermal anchor for the high current lines supplying the magnet. 14, DC-filters. 15, the elevated 4 K-plate. 16, the actual mK-plate.	39
3.16. System level wiring diagram of the fridge.	40

3.17.	Turbocycle heating system. 17, sample holding assembly wrapped with resistive heating wire, encased in silver-epoxy. 18, phosphor-bronze heating lines, delivering the heating current from the fridge input terminal to the mixing chamber stage. 19, thermal anchoring of the sample holding assembly to the millikelvin stage.	41
3.18.	A thermal cycle from millikelvin to room-temperature and back. The red lines highlight the beginning and the end of the thermal cycle, and are drawn at a temperature of 200 mK each. The blue curve T_{MXC} shows the temperature of the mixing chamber thermometer, which is screwed directly onto the mixing chamber. The orange curve T_{PCB} shows the temperature measured by a thermometer which is placed directly at the position of a future sample in sample holding assembly.	42
3.19.	Configuration of the fridge and outside instruments used in the experiments presented in 5. The colorcoding of the wired connections is consistent with Fig. 3.16. 250 Hz and 10 kHz denote the filtering used in the corresponding lines.	43
4.1.	Conceptual components of a measurement from the software viewpoint.	46
4.2.	Sequence diagram of a non-real-time measurement.	47
4.3.	Definition of an experiment using <code>spin_tune</code> .	49
4.4.	Gate layout of the structure employed in the following use cases.	50
4.5.	Conductivity of the quantum dot as a function of both barrier voltages.	51
4.6.	Accumulation of the device towards the conductivity G_{Target} .	52
4.7.	UML sequence diagram showcasing the measurement flow in <code>spin_watch</code> .	53
4.8.	Use case code of <code>spin_watch</code>	54
4.9.	Position in gatespace where the conductivity trace was taken.	54
4.10.	Conductivity versus time of the SET.	55
4.11.	Sequence diagram of a real-time measurement.	56
4.12.	Conceptual diagram of a realtime measurement. a) Pulse and measurement sequence to extract the excited state population after t_{wait} . b) Pulse and measurement sequence to extract the excited state population after t_{wait1} , t_{wait2} and t_{wait3} .	58

4.13. UML use case diagram of spin_excite. The left hand side shows directly which interactions the software offers to the user (active actor). On the right hand side the interactions in between the software and the measurement hardware (passive actor) can be found.	59
4.14. Definition of a realtime experiment using qpulse.	60
4.15. Upload of a defined realtime measurement to the hardware using spin_excite.	60
4.16. Device layout used to perform the demonstration experiment. The-to-be loaded dot is located underneath PR. The reservoir under RTA serves as a source of electrons and is tunnel coupled to the dot. The SET is located underneath ST.	64
4.17. Device layout used to perform the demonstration experiment. The-to-be loaded dot is located underneath PR. The reservoir under RTA serves as a source of electrons and is tunnel coupled to the dot. The SET is located underneath ST.	66
4.18. Constituent pulse of the dot loading experiment. V_{load} is applied to the plunger electrode of the dot, and the measurement is performed while the pulse amplitude is high.	66
4.19. Pulsetrain used for the dot loading. The pulsetrain is built from the constituent pulses shown in Fig. 4.18.	67
4.20. Dot population versus pulse amplitude.	67
4.21. Elzerman-readout-scheme of the spin state of an electron in a quantum dot. Adapted from [17].	69
4.22. Constituent pulse for Elzerman-readout.	69
4.23. Signature of an electron in the excited state (blip).	70
4.24. Pulse train used to find the correct readout level.	70
4.25. Result of the calibration measurement to find the readout level.	71
4.26. Result of the Elzerman- T_1 measurement, for a specific waiting time t_{wait} . Each iteration consists out of a trace similar to 4.23.	72
5.1. False colored scanning electron micrograph of the Si/SiGe device. The three gate layers are represented by different colors. The bottom layer is indicated in blue, whereas the topmost layer is indicated in green. The heterostack itself is shown in Fig. 2.4, and the purpose of each layer is discussed in sec. 2.2. The red shaded areas show the presence of an accumulated electron channel, the crossed squares denote ohmic contacts, and the red dots show possible positions of quantum dots.	74

5.2.	Typical accumulation curve. Marked in red is the conductance of $0.5 \mu\text{S}$ which is used to define the turn-on voltage.	75
5.3.	Turn-on voltage of the device over applied bias voltage during cooldown for four different devices with nominal equal layer stack and gate design.	76
5.4.	Simulation of the accumulation voltage as a function of the interface charge density Q_f	77
5.5.	Visualization of the first two steps of the tuning workflow. Artistic rendering of the electron distribution in the top half of the pictures and conductance versus gate voltage in the bottom part of the pictures.	79
5.6.	Visualization of the the third and the forth step of the tuning workflow. Steps three and four are repeated iteratively until a quantum dot is formed in between the barrier gates.	80
5.7.	Final steps of the tuning workflow. The dot position is highlighted by the green oval in between the barrier gates. After the detection of a quantum dot, a plunger sweep using the central electrode is performed to verify its existence.	81
5.8.	Single plunger trace of a formed SET, together with a secans fit and the extracted peak position.	81
5.9.	a) Peaktracking measurement, performed on sample A with a V_{BC} of 0 V . The colorplot consists of Coulomb peak traces recorded back to back over the timescale of five hours. The red line indicates the position of the peak maxima along the V_{SET} -axis. b) Noise power spectral density calculated with the welch method, fitted with a β/f^α power spectrum. To calculate the total noise power the spectrum is integrated over the highlighted frequency range.	82
5.10.	Peaktracking measurement on Sample B for a bias cooling voltage of 0 V	83
5.11.	Peaktracking measurement on Sample B for a bias cooling voltage of 0.625 V	83
5.12.	Peaktracking measurement on Sample B for a bias cooling voltage of 0.825 V	84
5.13.	Noise power spectral densities calculated with the Welch method, fitted with a β/f^α power spectrum. The resulting exponent is denoted in the legend.	85

5.14. Integrated noise power versus bias cooling voltage. While the noise increases towards negative biases for most samples, a local minimum around the value of 0.7 V can be found in each trace. The dashed line acts as a guide for the eye and shows the noise level of a sample bias cooled with zero volts. 86

5.15. Boxplot of the integrated noise power of 22 peak trackings, 11 performed each for the bias cooling voltages 0 V and 0.7 V. The orange solid line displays the median value, the green dashed line the mean value and the box includes 50 % of all data points centered around the median, called the interquartile range (IQR). The whiskers represent scores outside the middle 50 % and are capped at $1.5 \cdot \text{IQR}$. While the mean is dominated by the outliers and even falls outside the box of the 0.7 V dataset, it can be seen that the median value of 0.7 V is almost an order of magnitude lower than for 0 V. 87

5.16. (a) Simulated electric field inside the spacer layer for 0 V bias cooling (b) tunneling current from the silicon channel into the silicon cap for 0 V bias cooling. 90

5.17. (a) Simulated electric field inside the spacer layer for 0.7 V bias cooling (b) tunneling current from the silicon channel into the silicon cap for 0.7 V bias cooling 91

5.18. Measured integrated noise density plotted against simulated peak tunneling current. Each point represents a bias cooldown performed on sample A. 92

5.19. Simulated conduction band edge for a working point in which the device was cooled with a bias of 0 V. 1D slices of the conduction band edge shown in later figures are taken along the red line. . . 93

5.20. Rendering of the conduction band edge, interface charge and gate layout for a 0 V cooldown. η is the electrochemical potential of the conducting channel. 94

5.21. Rendering of the conduction band edge, interface charge and gate layout for a 0.7 V cooldown. 95

List of Tables

3.1. Summary of the coil parameters.	35
4.1. Basic modes in <code>spin_excite</code> , contained in the base version. . . .	64

List of Acronyms

CMOS	Complementary metal oxide semiconductor
HEMT	High electron mobility transistor
MOS	Metal oxide semiconductor
PCB	Printed circuit board
MIT	Metal-to-insulator transition
RT	Room temperature
SET	Single electron transistor
VLSI	Very-large-scale integrated circuit
IGFET	Insulated gate field effect transistor
FR4	Glass-reinforced epoxy laminate material
NISQ	Noisy intermediate scale quantum computing
QPU	Quantum processor unit
2DEG	Two-dimensional electron gas
MOSFET	Metal-oxide-semiconductor field-effect transistor
RF	Radio frequency
DAC	Digital to analog converter
FWHM	Full width half maximum

1. Introduction

There exists an interesting anecdote about quantum computing: It was the late nineties, and a group of physicists is trying to convince the the U.S. department of defense to fund their research. They pitch Shor's algorithm to the by now surely baffled audience: "With its ability for prime factorization we can crack all modern cryptography methods, including the encryption used by the international banking system." The board is impressed, but unsure: "Why should we fund research on a device that endangers most of today's critical data infrastructure?" Disappointed, the physicists leave the room and have an emergency discussion. Suddenly, one of them rushes back in and poses one simple question to the board: "What if the other side has it first?". The direct answer of the committee is lost to time, but now, more then twenty years later, the realization of a quantum computer has evolved into one of the largest technical efforts of mankind to date.

A very interesting aspect of quantum computing is the sheer novelty in the way it approaches computation. Even with a fully functional quantum computer, we would not be able to harness its full potential right away, because its possibilities are still vastly unexplored. While first claims to usefulness were made in the context of cryptography, this was soon extended to chemistry [78], materials- and life sciences [5, 82], and the development is far from over.

Currently, we live in the NISQ-era (Noisy Intermediate Scale Quantum computing)[90]. IBM claims to cross the utility threshold with its Quantum System two [53], consisting out of a tri-core-QPU (Quantum Processor Unit) with each core running on 133 qubits. Although computers based on superconductors are currently leading, there is a new emergent platform, promising to tackle the shortcomings of superconducting systems: Devices based on the spin states of one to many crystal-electrons in semiconductors. These platforms are fully Complementary metal oxide semiconductor (CMOS) compatible [74] (since they are CMOS), can operate at 1.5 K [81], have coherence

times of up to 28 ms [110], and can be coupled using microwave photons [95] and electron shuttles [27].

In the semiconductor spin-qubit community, silicon has arisen as the material of choice for most modern designs. The main reasons for this are the intrinsic low percentage of isotopes which contain nuclear spin, as well as the perfect compatibility with industry standard Very-large-scale integrated circuit (VLSI) techniques [89]. Currently, the two most prominent platforms in silicon quantum computing are Metal oxide semiconductor (MOS) and Si/SiGe structures, which in turn come with individual boons and weaknesses. MOS samples due to their simpler stack, are easier to fabricate. Furthermore the Two-dimensional electron gas (2DEG) is strongly confined at the oxide semiconductor interface, which leads to a high valley splitting ranging in the order of 600 μeV [29]. But with this direct proximity to the oxide interface, the electron gas becomes susceptible to impurities at the heterojunction. Typical mobilities in MOS devices range somewhere in between 5000 $\text{cm V}^{-1} \text{s}^{-1}$ [18] and 15 000 $\text{cm V}^{-1} \text{s}^{-1}$ [19], in rare cases up to 40 000 $\text{cm V}^{-1} \text{s}^{-1}$ [64], mostly being limited by the defects created from the lattice mismatch in between Si and the gate oxide SiO_2 . Complementary to MOS, Si/SiGe heterostructures offer mobilities above 800 000 $\text{cm V}^{-1} \text{s}^{-1}$ [102] for doped and beyond $1.6 \times 10^6 \text{ cm V}^{-1} \text{s}^{-1}$ [70] for undoped structures with global accumulation gates. This is due to the different confinement region for the electron gas. It is separated by a spacer layer from the dielectric interface, whose thickness usually ranges within the 30 nm region for most designs [55, 68, 15, 75]. While reducing the influence of oxide defects, Si/SiGe comes with its own set of challenges and problems. One being the low valley splitting of 60 μeV [99, 7, 52, 113] in comparison to MOS, and the other being the complexity of the heterostructure. Due to the complicated layer-stackup, devices based on SiGe show charging effects on the timescale of hours to days, making the long-term stability of the device working point a critical issue [63, 101, 11].

Helping to solve the latter problem shall be my concern from now on. Being the first student on the topic of semiconductor spin qubits in Karlsruhe, it was my responsibility to build a sustainable lab infrastructure that enables research on semiconductor quantum circuits. This thesis is structured accordingly. In chapt. 2 I will give an introduction to the theory of SiGe-Heterostructures, emphasizing the mechanisms that lead to charge redistributions.

Spin-qubits have high requirements regarding the capabilities of the experimental setup. First and foremost being that the platform itself has to be

cryogenic. Furthermore they require the full toolkit of the experimentalist, since they need to be provided with heavily filtered static voltages, complex gate waveforms with transients in the ns regime, and magnetic fields of a few Tesla. Hence, I will present the engineering that went into the construction of the semiconductor setup in chapt. 3.

In the following chapt. 4 I will present the measurement software I developed. I will go into the detail of the usage and maintenance of the three developed software packages `spin_tune` (synchronous data acquisition), `spin_watch` (asynchronous data acquisition), `spin_excite` (asynchronous data acquisition with realtime manipulation). Each of them is written to be hardware independent, easy to use and easy to adapt to the measurement setup at hand.

In the last chapt. 5 I will present how the individual gears from chapt. 3 and chapt. 4 mesh to investigate the charging processes involved in SiGe-heterostructures. I will show that a simple and readily available method, bias cooling, can reduce the charge noise up to two orders of magnitude, and present a theory regarding the trigger of these charging processes.

Let's jump right in.

2. SiGe heterostructures

2.1. Bandstructure of crystalline Si

Silicon is a group-IV element crystallizing, similar to carbon, in the diamond lattice, which is a variant of the FCC-Bravais lattice with two atoms sitting on each site. A representation of the conventional unit cell of the diamond lattice is drawn in Fig. 2.1. Which are the basis vectors of a BCC-lattice.

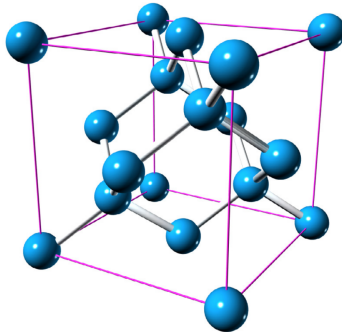


Figure 2.1.: Conventional unit cell of the silicon lattice [26].

Constructing the primitive unit cell in reciprocal space, we end up with the Brillouin-zone of silicon which is depicted in Fig. 2.2, with highlighted high-symmetry points.

Silicon is an indirect semiconductor (valence maximum and conduction minimum are not at the same point in momentum space), with a band gap of 1.17 eV [35] at \approx mK temperatures. The conduction band minimum lies on the Δ -axis close to the X point, and is six fold degenerate. The bandstructure is depicted in Fig. 2.3 (a). Fig. 2.3 (b) shows the iso-energy surfaces for charge carriers in the Δ_6 conduction band minima. The shape is ellipsoidal due to

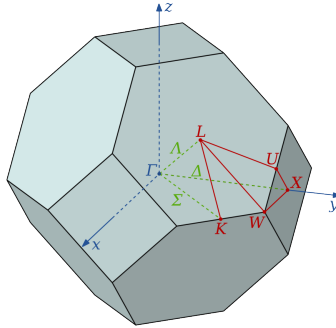


Figure 2.2.: Brillouin-zone of the FCC-lattice, adapted from [103].

the dependence on direction of the effective mass. In silicon, we differentiate in between the longitudinal effective mass $m_l^* = 0.916m_0$ and the transversal effective mass $m_t^* = 0.19m_0$ [96], with m_0 being the mass of a free electron. Longitudinal refers to a wave-vector parallel to the Δ -axis, and transversal refers to a wave-vector perpendicular to the Δ -axis.

2.2. Bandstructure of strained SiGe quantum wells

While in most electronic devices the six-fold degeneracy of silicon's conduction band does not play a major role maybe aside from increasing the charge carrier density, devices whose working principles rely on Pauli's exclusion principle are greatly affected. Pauli spin-blockade for example, a mechanism used in a plethora of quantum-dot devices [94, 80, 47, 49, 57] is not possible within the presence of valley splitting. Electrons that differ in their valley quantum number, are allowed to be in the same spin state and therefore seem unaffected by the exclusion principle. Lifting the valley degeneracy is therefore of utmost importance if one wishes to realize a silicon quantum device based on the exclusion principle. One way to eliminate the valley degeneracy and provide a high mobility 2DEG is a SiGe heterostructure, whose band structure will be discussed in the upcoming section.

First, let us define what a heterostructure is. In general it is comprised of at least two different semiconductor materials. These might differ in band

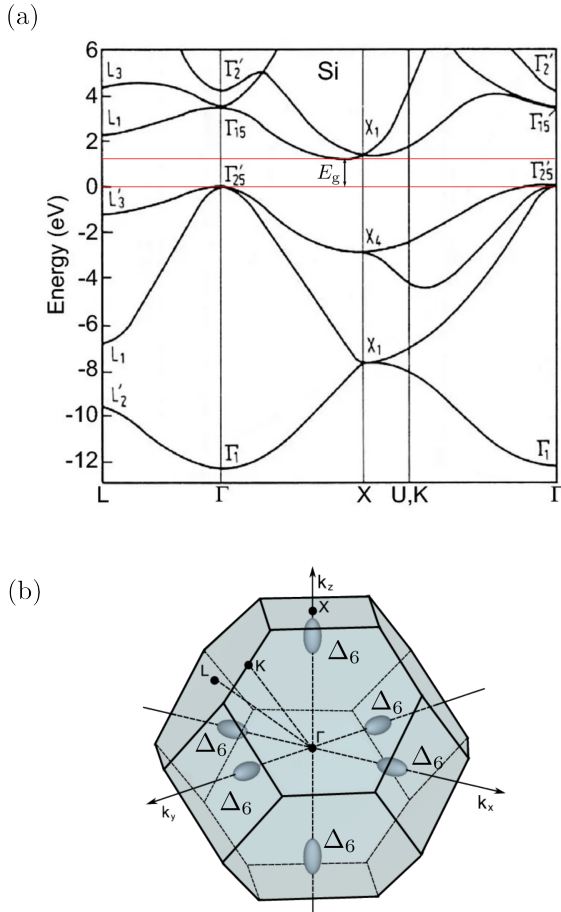


Figure 2.3.: Bandstructure of silicon (a). Iso-energy surfaces of charge carriers in the silicon conduction-band minima (b). Adapted from [37].

gap therefore electrical, optical, mechanical properties or none of the aforementioned. As long as the chemical composition changes along a specified interface and both materials are semiconductors, we have ourselves a heterostructure. Which materials are alloyed depends greatly on the design purpose. In the case of SiGe, this is the confinement of charge carriers in the growth direction, which we shall call the z -direction from now on. Fur-

thermore it is required that the valley degeneracy in between the different conduction band minima is completely lifted. A material-stack mostly fulfilling the mentioned requirements is the $\text{Si}_{0.7}\text{Ge}_{0.3}/\text{Si}/\text{Si}_{0.7}\text{Ge}_{0.3}$ stack shown in Fig. 2.4. As is visible the structure consists of a silicon (100) substrate, on

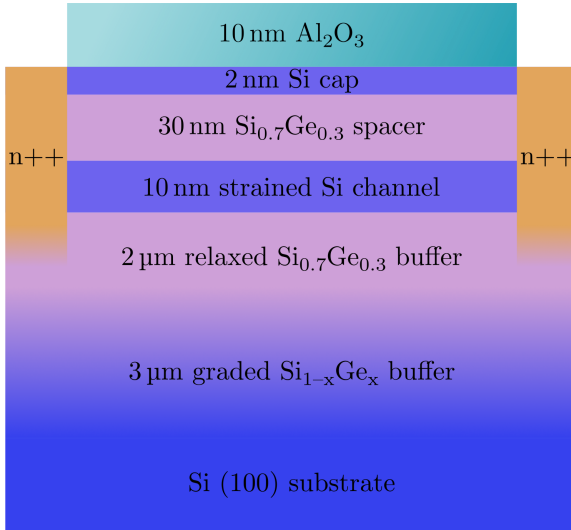


Figure 2.4.: Layer stack of the SiGe heterostructure used to confine electrons in the z-direction. The n++ regions are heavily doped with phosphor, and the silicon substrate is grown in (100) direction.

which a $3\ \mu\text{m}$ linearly graded $\text{Si}_{1-x}\text{Ge}_x$ buffer is grown, followed by a $2\ \mu\text{m}$ $\text{Si}_{0.7}\text{Ge}_{0.3}$ buffer, a $10\ \text{nm}$ strained silicon channel, a $30\ \text{nm}$ $\text{Si}_{0.7}\text{Ge}_{0.3}$ spacer layer, a $2\ \text{nm}$ silicon cap layer and a $10\ \text{nm}$ Al_2O_3 layer separating the semiconductor from the first metallic gates. We shall now clarify the the reason of existence for each of the different layers. Since to this day no large-scale SiGe wafers are commercially feasible, the bare substrate is silicon. Germanium's lattice constant is $4.2\ \%$ larger then the one of silicon [103], meaning that $\text{Si}_{0.7}\text{Ge}_{0.3}$ will also have a larger lattice constant then silicon. If it was grown directly onto silicon, the lattice mismatch would result in compressive strain in the SiGe. For the first $100\ \text{nm}$ the alloy would adapt to the lattice constant of the substrate, but after reaching this critical thickness SiGe would start to relax by forming large numbers of defects in the form of threading dislocations. To combat this problem, the lattice constant is increased linearly

by constantly increasing the germanium content in the graded buffer layer from 0 to 0.3. While this prevents most of the dislocations from forming, some will still be present, therefore a strain-relaxed $2\ \mu\text{m}$ $\text{Si}_{0.7}\text{Ge}_{0.3}$ constant-composition buffer is grown on top of the graded one to ensure minimum dislocation density at the interface to the silicon channel [87]. The silicon channel itself is grown on the $\text{Si}_{0.7}\text{Ge}_{0.3}$ alloy which has a different lattice constant and is therefore tensile strained. This strain breaks the inversion symmetry of the crystal, lifting partially the valley degeneracy in between the conduction band valleys. The four in-plane Δ_4 valleys (lying in $\pm k_x$ and $\pm k_y$ direction) are split from the lower lying out-of-plane valleys (lying in $\pm k_z$ direction) Δ_2 -valleys by 200 meV, see Fig. 2.5 (a). This results in a type-2 band alignment in between the Si channel and the SiGe buffer layer, with the Δ_2 conduction band minimum of the silicon lying approximately 180 meV below the one of the SiGe alloy. The silicon valence band maximum on the other hand is 170 meV below the SiGe valence minimum, therefore effectively creating a well for electrons and barrier for holes. [119, 96]. The resulting band structure is depicted in 2.5 (b). The thickness of the quantum well

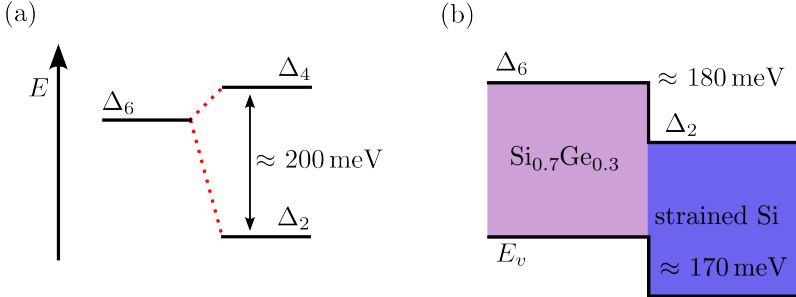


Figure 2.5.: Partial lifting of the degeneracy of the Δ_6 conduction band minimum by induced strain (a). Resulting band alignment in a SiGe heterostructure (b). With n being the electron density inside the channel, Δ_2 and Δ_4 are the edges of the two lowest lying conduction bands, $E_{0,channel}$ and $E_{0,cap}$ are the lowest subbands in channel or cap respectively and η is the electrochemical potential (Fermi-energy) of the system.

layer is constrained by the so called Matthews-Blakeslee-limit [73]. A silicon channel with a thickness surpassing this limit will start to relax the atomistic strain by forming threading dislocations, therefore lowering the valley splitting and the electron mobility due to defect scattering. For the discussed heterostructure this limit has been calculated to be 10 nm [84]. Next in the stack is the SiGe spacer layer. The purpose of the spacer layer is twofold: First,

it confines the 2DEG in the upper z-direction, second, it protects the 2DEG from the defects at the oxide interface. While this separation indisputably creates low critical electron densities [44] and the mobility of the 2DEG is limited by remote impurity scattering [67, 43], it is not possible to make this layer infinitely thick. Otherwise patterning the electron gas with gate structures on top of the stack will require increasingly high voltages due to the spatial separation. On top of the spacer resides the silicon cap, which is a Si layer with a thickness of about 2 nm [42, 98]. The reason of existence of this layer is not purely to annoy experimentalists [119], but also to protect the lower lying layers containing germanium from oxygen diffusion. Since the cap is much thinner than the channel, the first sub-band is approximately twenty-five times higher in energy than the first subband of the silicon channel. Therefore, even if placed further away from the interface, electrons will accumulate in the channel at lower voltages than in the cap if placed in an electric field. To top it off, Al_2O_3 is used as gate oxide. While the purpose of the oxide layer is immediately clear, namely to insulate the metallic gate layers from the semiconductor, the choice of material is not. Thermal SiO_2 is known to produce the highest quality oxide interface with silicon, but the problem lies in the growth method. If exposed to high processing temperatures, the germanium contained within the structure will start to diffuse and contaminate the silicon channel [87], reducing the quality of the 2DEG. Using Al_2O_3 as gate oxide represents a compromise in between interface and 2DEG quality, as it can be deposited at temperatures below 300 °C and maintain low defect densities as well as strong insulating behavior [41, 40, 69].

Since the heterostructure is undoped, no conduction electron density will be present in the Si channel at low temperatures. The conduction band edge and the electrochemical potential of the heterostructure in thermal equilibrium is shown in Fig. 2.6 (a). To accumulate electrons inside the channel a global voltage is applied to the structure, for example through a global accumulation gate. If the voltage is high enough, the first sub-band of the quantum well will be pulled below the Fermi-level, resulting in a non-zero electron density within the channel, as in Fig 2.6 (b). Since the first sub-band of the silicon cap is higher in energy due to the stronger confinement, the cap is still insulating, without any additional charge carriers. If the accumulation voltage is increased further, the first subband of the cap is also pulled below the Fermi-level, and electrons should populate the cap. This second accumulation results in a saturation of the charge density within

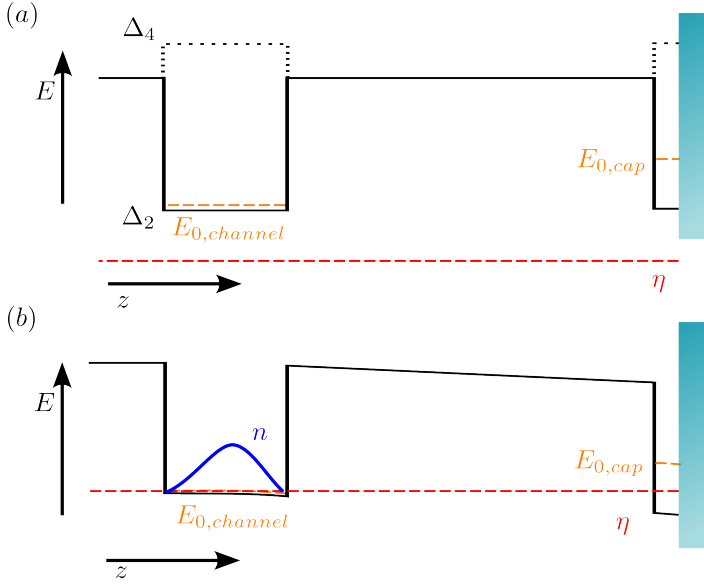


Figure 2.6.: Band-edge alignment of the heterostructure in the zero-bias case (a) and in the accumulated case (b).

the quantum well, since the additional 2DEG layer will screen the electric field at the position of the well. The accumulation voltage of the second channel can be estimated by for example, self-consistent schrödinger poisson calculations [71, 45]. Surprisingly, the cap is not populated by electrons at the calculated voltage. The carrier density in the quantum well keeps increasing far above the calculated limits, only breaking down at voltages way above the estimations. Since the cap is very thin and in direct proximity to the defect-ridden oxide interface, the disorder potential can be assumed to be quite strong, leading to a high critical density of the metal-to-insulator-transition of the silicon cap. Therefore the ohmic contacts, which also contact the cap, cannot supply electrons to the layer, and it remains insulating. Yet, the critical density is reached at a sufficiently high voltage. Therefore a source of electrons populating the cap has to be present. Fowler-Nordheim tunneling from the silicon channel into the cap represents such a source [71]. Furthermore a partially charged silicon cap is correlated to large charge redistributions affecting the low-frequency charge noise of quantum dot devices [23]. The

exact charge dynamics in the partially charged state remain elusive and are theorized to either [116] be dominated by Anderson localization [1], or to form separated charge puddles of mobile carriers, depending on the lengthscales of the potential disorder and the wave-function localization in the system. Insulated gate field effect transistor (IGFET) devices in similar heterostructures exhibit an Anderson-type metal-to-insulator transition [34, 33] in the quantum well layer. The cap on the other hand is located directly at the semiconductor-oxide interface, and therefore similar to metal-oxide-semiconductor field-effect transistor MOSFET structures, where the metal-to-insulator transition MIT is known to be percolation induced [106].

2.3. Tunneling into the cap

2.3.1. The Tsu-Esaki model

The general case of two semiconductors separated by an arbitrarily shaped tunneling barrier is considered. Presented on the left hand side of Fig. 2.7 is a semiconductor with its electrochemical potential being at η_1 , lying above the potential of the second semiconductor on the right hand side, which is at η_2 . The corresponding Fermi-distribution functions are drawn in blue. Sandwiched in between the two semiconductors is a tunneling barrier, which, for example, could be realized by an oxide.

To compute the current flowing from one semiconductor electrode to another, we start by considering the equation expressing the current density j in the Boltzmann regime for a free electron gas [46]:

$$j = \frac{q}{4\pi^3} \int_{-\infty}^{\infty} \mathbf{v}(\mathbf{k}) f(\mathbf{k}) d^3k \quad (2.1)$$

Where q is the charge of the carrier that forms the current, \mathbf{k} the wave-vector of the charge carriers, \mathbf{v} the group velocity and $f(\mathbf{k})$ the non-equilibrium distribution function. In the next step, we modify the equation at hand to compute the tunneling current in between two semiconductors separated by a barrier. The current from electrode one to electrode two can be expressed as [14, 30]:

$$j_{1 \rightarrow 2} = \frac{q}{4\pi^3} \int_{-\infty}^{\infty} TC(k_z) v_z(\mathbf{k}) f_1(\mathbf{k}) (1 - f_2(\mathbf{k})) d^3k \quad (2.2)$$

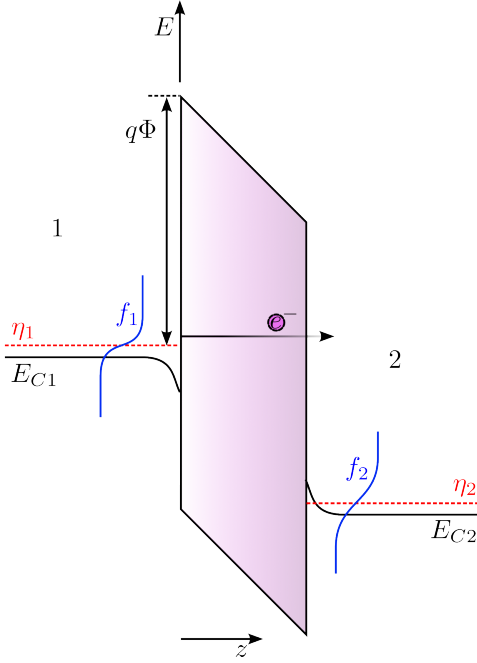


Figure 2.7.: Relevant energy levels and distribution functions for electron tunneling in between two degenerately doped n-type semiconductors. With η_1 and η_2 being the electrochemical potentials of the semiconductors, E_{C1} and E_{C2} the corresponding conduction band edges, f_1 and f_2 the Fermi distribution functions, and Φ the height of the potential Barrier.

The first main modification made in comparison to 2.1 is that the finite transmission rate through the barrier is taken into account by the tunneling coefficient TC . Since the tunneling contact lies within the x-y-plane only electrons with a wave vector component in z-direction will attempt to cross the barrier. Therefore the tunneling coefficient is only dependent on the wave-vector component orthogonal to the tunneling contact. Furthermore the non-equilibrium-distribution function can be expressed as $f_1(\mathbf{k})(1 - f_2(\mathbf{k}))$. With f_1 and f_2 being the probability distribution functions for the occupancy of states with the wavevector \mathbf{k} . The tunneling process involves only a small part of the conducting electrons, and therefore both electrodes can be described as being in equilibrium. The probability of transfer scales with the

probability of the initial state being occupied ($f_1(\mathbf{k})$) and the final state being unoccupied ($1 - f_2(\mathbf{k})$).

The total current flowing through the tunneling junction is expressed as the difference in between charge carriers flowing from electrode one to electrode two and two to one.

$$j = j_{1 \rightarrow 2} - j_{2 \rightarrow 1} \quad (2.3)$$

$$j = \frac{q}{4\pi^3} \int_{-\infty}^{\infty} TC(k_z) v_z(\mathbf{k}) (f_1(\mathbf{k}) - f_2(\mathbf{k})) d^3k \quad (2.4)$$

Finally, we perform a change in variables from wave-vector \mathbf{k} to the more intuitive energy E . For a free electron gas we can separate the energy terms arising from the motion towards (or away) from the tunneling barrier E_z and the motion in the plane of the tunneling barrier E_p . The condition $E = E_z + E_p$ is always fulfilled. For the energy arising from the motion in x-direction the following considerations can be made:

$$E_z = \frac{\hbar^2 k_z^2}{2m_t^*}, \quad v_z = \frac{1}{\hbar} \frac{\partial E}{\partial k_z}, \quad v_z dk_z = \frac{1}{\hbar} dE_z \quad (2.5)$$

The in-plane components (parallel to the tunneling barrier) are expressed in polar coordinates :

$$E_p = \frac{\hbar^2 (k_x^2 + k_y^2)}{2m_t^*} = \frac{\hbar^2 (k_\rho^2)}{2m_t^*}, \quad \frac{\partial E_p}{\partial k_\rho} = \frac{\hbar^2 k_\rho}{m^*}, \quad k_\rho dk_\rho = \frac{m^*}{\hbar^2} E_p \quad (2.6)$$

Implementing the coordinate transformation showcased in 2.5 and 2.6 in equation 2.4 yields the following expression for the tunneling current density, also known as the Tsu-Esaki-model [20, 108]:

$$j = \int_0^{\infty} TC(E_z) N(E_x) dE_z \quad (2.7)$$

with the supply function $N(E_z)$ being defined as:

$$N(E_z) = \frac{qm^*}{2\pi^2 \hbar^3} \int_0^{\infty} [f_1(E) - f_2(E)] dE_p \quad (2.8)$$

2.3.2. Fowler-Nordheim-Tunneling

After deriving the tunneling current formula for arbitrary barriers and and supply functions, it is necessary to substantiate the formalism on the problem at hand, namely the SiGe heterostructure described in 2.2. The goal of the following section is to calculate the tunneling coefficient TC . A possible tunneling path for electrons to take is from the strained silicon channel through the spacer layer into the silicon cap. At the onset of significant tunneling currents through this channel, the conduction band edge takes approximately the shape of a triangular barrier [71, 45, 43] as is depicted in fig. 2.8, with Φ denoting the effective barrier height. Therefore the resulting tunneling currents are to be computed within the regime of Fowler-Nordheim tunneling [24]. To derive the Fowler-Nordheim tunneling coefficient we will employ

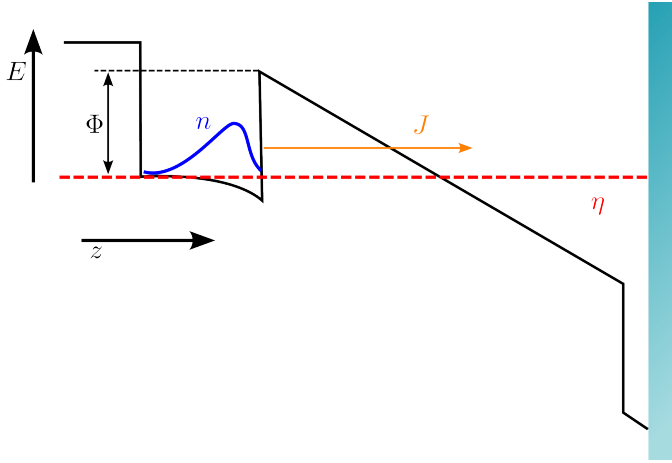


Figure 2.8.: Band alignment at the onset of significant tunneling currents into the cap, with J being the total tunneling current density.

the WKB-approximation [115, 62, 9] for the transmission coefficient:

$$TC \approx \exp \left(-2 \int_{z_1}^{z_2} |k(z)| dz \right) \quad (2.9)$$

Where $k(z)$ denotes the wave-vector inside the barrier, z_1 and z_2 the classical turning points. The shape of the barrier is included by formulating an expression for k :

$$k(z) = \sqrt{\frac{2m^*}{\hbar^2} (\Phi + \eta_1 + E_z - V(z))} \quad (2.10)$$

Where m^* is the effective mass inside the barrier, Φ is the effective Barrier height, η_1 the electrochemical potential of the source semiconductor, $V(z)$ the barrier function and z the current position inside the tunneling barrier. To compute $V(z)$ two considerations have to be taken into account. First, in the region in between both electrodes, the electron basically behaves as in a parallel plate capacitor. Its potential depends linearly on the position in between both electrodes. Second, since the source of the tunneling electrons is a two dimensional electron gas, which behaves like a metal for all intends and purposes of the following arguments, the image-force of the extruding electron has to be included as well [97]. Thus, the potential barrier function $V(z)$ includes two terms:

$$V(x) = \frac{q^2}{16\pi\epsilon_r\epsilon_0(z - z_1)} - q\mathcal{E}(z - z_1) \quad (2.11)$$

Where ϵ_r is the permittivity of the material inside the barrier, ϵ_0 the vacuum permittivity, \mathcal{E} the electric field in which the electron is moving and z its current position inside the tunneling barrier. The first term describes two attracting point charges, therefore includes the image force and the second describing the motion of an electron in a homogeneous electric field, as is present in between the silicon channel and the cap layer.

Effectively, the image-force reduces the work function by:

$$\Delta\Phi = \sqrt{\frac{q^3\mathcal{E}}{4\pi\epsilon_r\epsilon_0}} \quad (2.12)$$

The reduction is defined as the maximum difference in potential caused by the mirrored charge [103]. Substituting equation 2.11 into 2.9 and carrying out the integration yields [117]:

$$TC \approx \exp \left[\frac{-4\sqrt{2m^*\Phi^3}}{3\hbar q\mathcal{E}} v(\mathcal{T}) + \frac{2\sqrt{2m^*\Phi}}{\hbar q\mathcal{E}} t(\mathcal{T})(E_z + \Phi) \right] \quad (2.13)$$

Where $v(\mathcal{T})$ and $t(\mathcal{T})$ are slowly varying functions of the normalized image-force barrier lowering $\mathcal{T} = \Delta\Phi/\Phi$. While there is no analytical expression for v and t , tabulated values are calculated in e.g. [10].

$$TC(E_z) = \exp \left[\frac{4\sqrt{2m^*}}{3\hbar q\mathcal{E}} (\Phi - E_z + \eta_1)^{\frac{3}{2}} \right] \quad (2.14)$$

In order to finalize the derivation of the field-emission or Fowler-Nordheim tunneling current, we now focus on the supply function $N(E)$. The device is operated in the mK range, therefore it is safe to assume $T \approx 0$ mK. For the zero temperature case the fermi-dirac-distribution function can be approximated as a the Heaviside-step function:

$$f(E) = \frac{1}{1 + \exp\left(\frac{E-\eta}{k_B T}\right)} \approx \Theta(E - \eta) \quad (2.15)$$

Performing the integral in equation 2.8 yields the supply function for a fermionic system at zero temperature:

$$N(E_z) = \frac{qm^*}{2\pi^2\hbar^3} (\eta_2 - \eta_1) \quad (2.16)$$

By inserting the tunneling coefficient 2.13 and supply function 2.16 into the net tunneling current 2.7 we compute the final result for the electron flow in the Fowler-Nordheim regime [66]:

$$J = \frac{q^3 \mathcal{E}^2}{8\pi\hbar\Phi t^2(\mathcal{T})} \exp \left(-\frac{4\sqrt{2m^*}\Phi^3 v(\mathcal{T})}{3\hbar q\mathcal{E}} \right) \quad (2.17)$$

An interesting caveat hereby is that m^* does not have to correspond to the effective mass of the lowest lying conduction band of the dielectric material, since a tunneling electron does not occupy any of the bands the dielectric provides. To compute tunneling currents, an effective mass has to be used nevertheless. It has been shown that the energy dispersion of the electron inside the forbidden gap follows the predictions of the two-band model [86] and intricate calculations [2] exist to estimate the effective mass inside the barrier. Experimentally the mass within the barrier seems to deviate only slightly from the effective mass of the lowest lying conduction band of the tunneling barrier [72, 114, 79, 8].

2.4. Gate defined Quantum Dots

Generally, a quantum dot consists of a small conductive island and tunneling contacts to source (S) and drain (D) electrodes as is depicted schematically in Fig. 2.9 (a). Additionally a gate (G) electrode may be added to tune the energy levels of the system. The constraints on the size of the island are set by the fact that the capacitance has to be sufficiently small that the discrete nature of the electron charge becomes visible. While many factors play a role in the visibility of single-electron effects, the key enabling condition is that the charging energy is large in comparison to the thermal energy $\frac{e^2}{2C_{\text{island}}} \gg k_B T$. For typical low-temperature experiments performed in the ≈ 30 mK range, the capacitances involved should be smaller than 1 fF. While all quantum dots have in common that single charge effects have to be visible, their direct physical implementation vastly depends on the system they are realized in. While the first quantum dot devices were fabricated using small normal metal islands [28, 112], the quantum dot device family was soon extended by superconducting devices [93], molecules inside break-junctions [85, 32], vertical semiconductor quantum dots [59] and, most important in the context of this thesis, lateral semiconductor quantum dots [76, 60]. To understand the charge dynamics in these nanoscopic structures, we examine the equivalent circuit model in Fig. 2.9 (b). First let us only consider the dot itself, without applied voltages to any of the electrode contacts. The pure electrostatic energy arising from N electrons located on the capacitance of the metal island is described by the first term in the Hamiltonian 2.18. Additionally if we now allow the source potential V_s , drain potential V_d , and plunger gate potential V_g to be at a nonzero voltage, their influence on the energy level of the electrons on the island electrode is considered by the second term in 2.18. The last term includes effects arising from the electron being a nonclassical particle obeying the Pauli exclusion principle, meaning it takes into account the orbital energies $\epsilon_{i,\sigma}$ which are added on top of the purely electrostatic terms.

$$H_{QD} = \frac{e^2}{2C_{\Sigma}} N(N-1) - \frac{Ne}{C_{\Sigma}} (C_g V_g + C_s V_s + C_d V_d) + \sum_{i,\sigma} N_{i,\sigma} \epsilon_{i,\sigma} \quad (2.18)$$

Where e denotes the electron charge, C_{Σ} the total capacitance of all gates and leads, C_g , C_s , C_d the capacitance of the gate, source and drain electrodes, $N_{i,\sigma}$ the number of electrons in the i -th quantum state with spin σ and $\epsilon_{i,\sigma}$ the energy corresponding to said state. Furthermore, we assumed that the

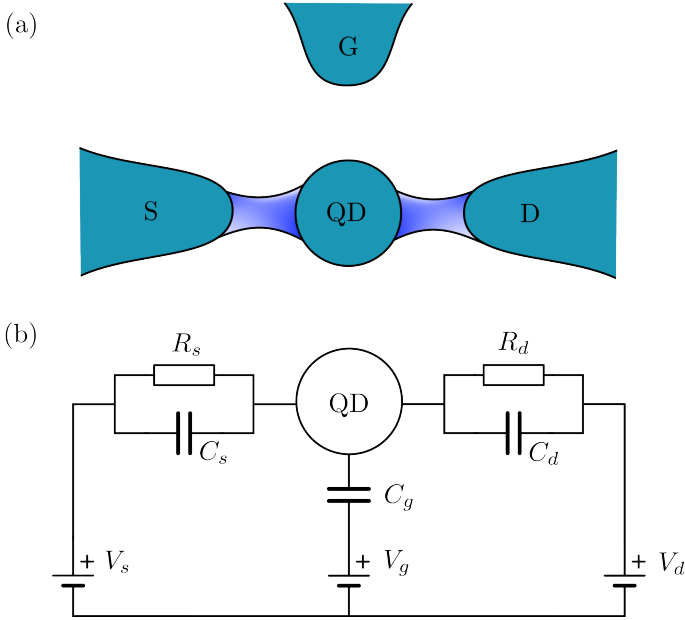


Figure 2.9.: Schematic depiction of a general quantum dot device (a). The white-blue regions exemplify the tunnel-coupling in between source and drain. Equivalent circuit model of the system sketched above (b).

capacitances are independent on the number of electrons inside the dot, and that each electron feels the exact same Coulomb repulsion to every other electron [50]. This mathematical treatment of the quantum dot system is known as the constant interaction approximation [109, 61, 58]. After familiarizing ourselves with the energies involved in the steady state of a dot system, we now turn our attention towards the current carrying capabilities of a quantum dot. As a first step we investigate the total energy U arising from N electrons being on the quantum dot, which can easily be deduced from the hamiltonian. The electrochemical potential is the energy level which will be occupied if another particle, in this case an electron, is added to the system. We can therefore write:

$$\eta(N) = U(N) - U(N - 1) \quad (2.19)$$

$$\eta(N) = \frac{e^2}{2C_\Sigma} \left(N - \frac{1}{2} \right) - \frac{e}{2C_\Sigma} (C_s V_s + C_d V_d + C_g V_g) + E_N \quad (2.20)$$

Where E_N denotes the difference in between the purely orbital energies of the N -th and $N - 1$ -th electron. The values of the electrochemical potential are quantized in the electron number, and a schematic depiction of a quantum dot with corresponding energy levels is drawn in Fig. 2.10. Electrons can only pass through the dot, if a dot-level is aligned with the source and drain potential in the zero bias case. Meaning the gate electrode has to bias the dot so that η is resonant with source and drain. In the classical Coulomb-diamond depiction this case is reflected by the points in which the diamonds cross the V_g axis, because those are the only possible configurations that allow conductance at zero bias. If we know apply a voltage in between source and drain, the afore pointlike conduction criterion becomes one dimensional (a line :), since the dotlevel does not have to be aligned with source and drain, but in a range in between source and drain, giving rise to the edges of the Coulomb diamonds. The width of a Coulomb diamond (on the x-axis) is

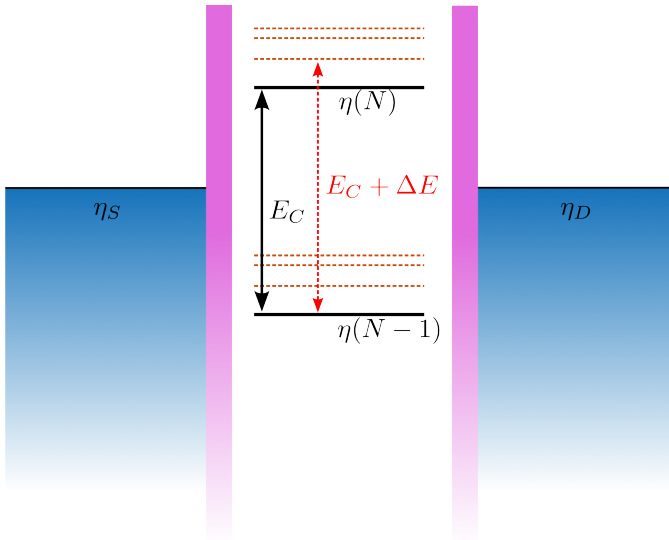


Figure 2.10.: Schematic drawing of a quantum dot with and the corresponding energy levels.

determined by the adding Energy E_{add} , which describes the electrochemical potential difference in between an N and $N - 1$ electron charging state. As a general definition for the adding energy we write:

$$E_{add} = \eta(N) - \eta(N - 1) = E_C + \Delta E \quad (2.21)$$

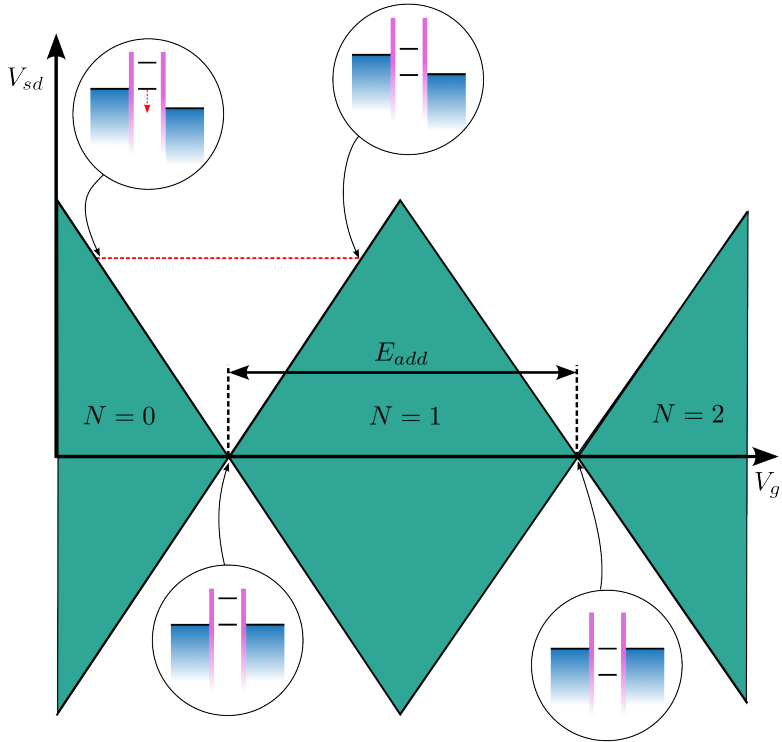


Figure 2.11.: Schematic depiction of Coulomb-diamonds and the corresponding energy-level alignments on the quantum dot.

Where E_C is the single-electron charging energy $e^2/2C_\Sigma$ and ΔE the energy difference in between two orbital states. ΔE can also take the value of zero, in the case of two electrons with opposite spin only charging the dot but being in the same orbital state. The area inside a Coulomb diamond (the area closest to the x-axis, in between two conductance-onset lines) depicts combinations in the parameterspace (V_{SD} and V_g) in which the dot is non-conductive, The bias window does not contain any energy levels, no tunneling paths are available, conduction through the dot is blocked. Since the main contributor to the level spacing is the electrostatic Coulomb repulsion in between the electrons charging the island electrode, this regime is called Coulomb blockade. Further parameters, besides a matching set of gate voltages, crucial to develop a

Coulomb blockade are the Temperature of the electron system and the orbital energy spacing ΔE . We formulate the requirements for a Coulomb blockade with the following inequalities:

$$\Delta E < k_B T < E_C \quad (2.22)$$

$$\Gamma^{-1} > \frac{\hbar}{E_C} \quad (2.23)$$

$$e|V_{DS}| < 2E_C \quad (2.24)$$

With Γ being the tunnel coupling. If the thermal energy would, at any point, be higher then the charging energy, thermally excited electrons would be able to surpass the Coulomb blockade, leading to a constant background current, and in the case of $k_B T \gg E_C$, the number of electrons on the dot is not a good quantum number anymore, and any features of the Coulomb blockade disappear completely. The second criterion for Coulomb blockade to occur is that the orbital energies ΔE are much smaller then the charging energies. A dot in which this condition is not fulfilled will also block the flow of current and, depending on the gate voltage, let it pass, but the levels will not be nearly equidistant and the blockade will be caused by orbital energy and not Coulomb repulsion, therefore not leading to the regime of Coulomb blockade. Furthermore the tunnel coupling Γ has to be low enough, that the energy of the electron on the island is well resolved and its uncertainty smaller then the charging energy. The last condition for Coulomb blockade to occur is formulated in eq. 2.24. In the blockade regime the electrons tunnel one-by-one, therefore the source-drain bias V_{SD} must be smaller than twice the charging energy. If the bias exceeds this limit, multi-electron tunneling becomes possible.

3. Hardware

3.1. DC-Wiring

A working chip in the cryostat is, in the most exciting of scenarios, a homogeneous 2DEG. The accumulation and depletion of specific spacial regions of the chip is done in-situ, by applying slowly changing voltages to the different gates hopefully present on the sample. Therefore, the quality of the semiconductor qubit is not only defined by the material of the chip, but also by the noise level of the voltages used to define the potential landscape. Since thermal noise and as a direct consequence high electronic temperature of the chip, will impact the performance only negatively, the wiring delivering the static control voltages, should mitigate heat transfer from room- and higher temperature stages in electronic form via heat conduction, and radiation traveling down the input lines.

The first of the two concerns can be eliminated by material choice. Since heat conductivity is directly linked to electrical conductivity by the Wiedemann-Franz law [25], using wires made out of a high resistivity material will protect the sample from heat traveling down from the input ports on the fridge side. A further constraint placed on the material is that its conductivity should be temperature independent, since it is going to bridge the full temperature range room temperature near to absolute zero. The conductivity of materials that fulfill this requirement is usually defect-dominated (since the amount of defects will not change greatly in the temperature range which is of interest to us). An example of such a material is constantan, and was therefore used for the wiring. A side effect of the high resistivity is that passing larger currents (mA, in this case) will ironically cause heating. To mitigate that, we used wires that embed superconducting (below 16 K) NbTi filaments inside a constantan matrix. The superconductor will not increase the thermal load on the sample, since all superconductors are poor thermal conductors in comparison to low-resistance normal metals.

With the choice of materials out of the way, it is time to turn our attention to the construction of the wirebundles delivering the control signals. A cable is seldom alone, especially in setups that run semiconductor nanostructures. Dictated by the pinout of the Jäger-connector, we chose to run the wires in bundles of twelve. From this point on, we shall discuss the anatomy of such a bundle. A schematic depiction can be found in Fig. 3.1. Explaining the schematic from the inside out, besides the twelve live wires, also eight grounded wires are present. All of these are constantan-superconductor wires with an outer diameter of $100\ \mu\text{m}$ bare diameter, and $110\ \mu\text{m}$ including the insulating lacquer. While the purpose of the live wires should be clear by now, the grounded ones are there to increase the capacitance of the wires toward the outer shield. This is done to increase the filtering effect of the cabling itself. The wire can be seen as a distributed RC-filter (with R being formed by the resistive wire and C by the capacitance to ground). Additional wires inside the shield will decrease the distance in between live and grounded components, therefore increasing the capacitance. This increased capacitance now provides a short for high-frequency signals. The wirebundle itself is encased in Apiezon-grease, which is used to increase the thermal conductivity to the outer shield as well as serving as a lubricant during the construction of the wiring. Finally, the whole bundle is encased in a grounded capillary made out of CuNi with an outer diameter of 1.7 mm, providing electromagnetic shielding as well as mechanical stability. The bundles themselves are 110 cm

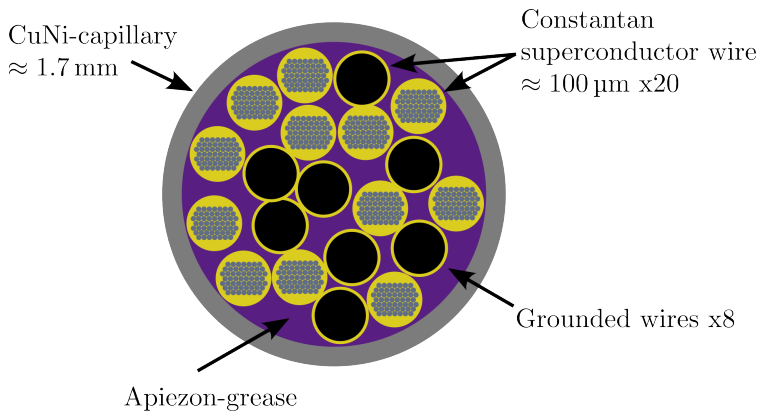


Figure 3.1.: The anatomy of a wirebundle.

long, and four of them are installed in the cryostat, summing up to a total of 48 lines to deliver static voltages to the sample. Furthermore the cabling is thermalized at every temperature stage it crosses. A copper wire is soldered to the outer shield connecting it to the corresponding temperature stage if above 4 K. If below 4 K, it is advised to glue these thermalizations with a conductive glue, like epoxy (UHU Endfest 300, in this special case) infused with silver powder, since tin becomes superconducting at 3.7 K, and therefore loses most of its heat conducting capabilities.

The transmission function of two DC lines in series (in and out) was measured up to 500 MHz during a cryostat cooldown. The results are shown in Fig 3.2. While the exact temperature of the cable is unknown (since a strong temperature gradient is present), it is nevertheless instructive to observe the behavior from room temperature (red) to the mixing chamber reaching 28 mK (blue). In the normal state, the wiring has a signal rejection of more than -30 dB from 50 MHz on. After the superconducting transition, the rejection reduces to -20 dB.

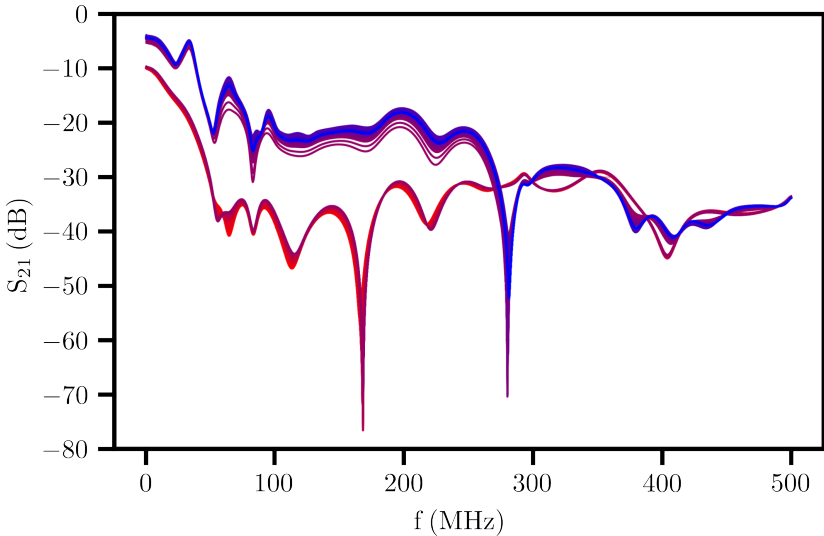


Figure 3.2.: Transmission function of two DC-lines in series. The color gradient indicates the temperature of the wire, red being a mixing chamber temperature of 300 K and blue being a mixing chamber temperature of 28 mK.

3.2. Filtering

When designing the DC input wiring, we did not make full use of the slowly varying (in time) nature of the signals which are to be sent through these connections. Since it is our goal to also minimize the heating of the sample by radiation traveling in the wires, we include low pass filters in the DC lines to block higher frequencies. When choosing the cutoff frequency the following arguments have to be weighed against each other: The lower the cutoff frequency, the lower the amount of noise and radiation which is allowed through the wire. However, this comes at the cost of measurement speed, since the cutoff frequency also constricts the speed at which the experimentalist can apply signals to the sample. In this specific example, a cutoff frequency of 250 Hz has been chosen as a reasonable compromise in between allowing higher frequency noise and minimum $\frac{dV}{dt}$ for applied control signals. Another set of filters is in place with a cutoff frequency of 10 kHz for the intended use with ohmic contacts. This higher frequency was chosen to mitigate $1/f$ -noise while measuring with, for example, lockin-amplifiers.

One of the simplest realizations of a low-pass filter is a RC-filter. It offers two main benefits: It is easy to build and the constituents are all purely passive, which minimizes the noise produced by the filter itself as well as removes the need for additional wiring. To provide a steeper frequency response once below the cutoff, a two-stage RC-filter design has been chosen.

In the case of perfect RC-filters the filtering circuit would be complete at this point. But as soon as small inductances (like the inductance of the trace connecting the filter components) are introduced to the circuit (Fig. 3.3), the filtering function of an RC-lowpass declines with higher frequencies, as is shown in Fig. 3.4. To combat this issue, an integrated intermediate-

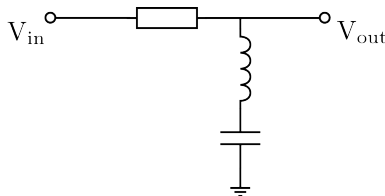


Figure 3.3.: An RC-filter circuit which includes the parasitic inductance of the traces connecting the individual components.

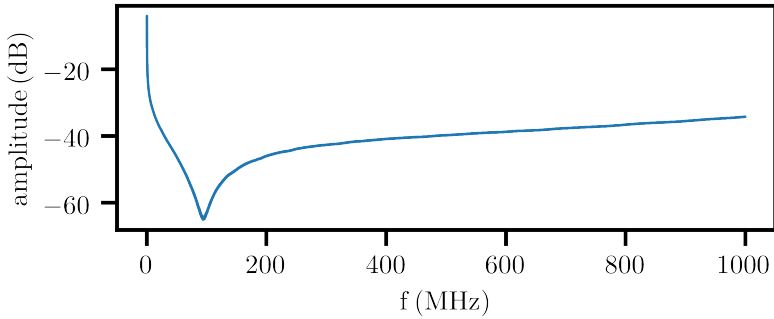
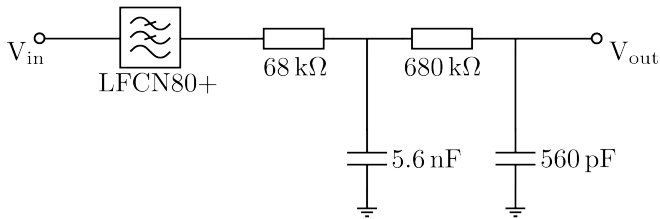


Figure 3.4.: Transmission through a real RC-filter.

range filter has been added to the circuit. The LFCN-80+ manufactured by Mini-Circuits provides, a signal rejection of -40 dB in between 225 MHz and 1550 MHz, and -20 dB up to 4500 MHz. The full circuit diagram of the filter, is depicted in Fig. 3.5.

250 Hz cutoff:



10 kHz cutoff:

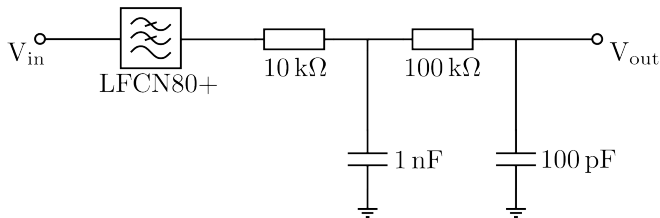


Figure 3.5.: Circuit diagram of the filtering applied to each input line.

The next step which is needed to be taken is the design of the Printed circuit board (PCB) hosting the individual components. The design goals of the PCB were the following: The additional pickup of radiation in the PCB is to be minimized. The traces carrying the incoming signal are to be thermalized. The whole design should be as small as possible. The assembly should filter a whole DC-wirebundle, therefore being able to handle 12 input lines. To tackle the first design goal, all traces and connections in between different components are as short as possible, and hidden inside the layer stack. On the front and back of the PCB, only footprints of the devices and vias are present (see Fig. 3.6 b)). Besides the footprints, the front and back are covered by grounded copper pours. Furthermore, the edges of the PCBs are metallized, to form a Faraday cage for the contained traces. To reduce coupling to the radiation environment as well as provide a better thermal contact, a grounded layer has been added in between every two planes carrying traces. The full layer stackup is shown in 3.6 a). To maximize the thermal contact to the ground planes, the dielectric separating layers 1 to 5, starting counting from the top of the layer stack, and 6 to 10 are of a thickness of only 100 μm . The central dielectric separating layers 5 and 6, has a thickness of 1 mm to provide mechanical stability to the circuit board. All dielectric layer are made of Glass-reinforced epoxy laminate material (FR4) and the copper layer have a thickness of 35 μm . It is important to mention that, especially for cryogenic applications, the layer-stackup has to be symmetrical around the central layer. If this is not the case, the circuit board will bend once the temperature changes significantly, since each side will expand (or contract) at a different rate. To thermalize the incoming signal, the individual input lines are broken out to planes to increase the surface area in layers four and seven (see Fig. 3.6 c)). The increased surface area decreases the thermal resistance in between the incoming signal lines and the rest of the PCB. Furthermore the PCB filtering 12 input lines has a footprint of only 26 mm x 40 mm. Therefore the third and fourth design goals are also reached. A picture of the final developed PCB is shown in Fig. 3.7. The effectivity of the filter was assessed using a RuO thermometer. The filter was securely thermalized to the mixing chamber plate, and the thermometer was attached to its output. While the mK-plate was at a temperature of 28 mK, the thermometer connected to the filter was at 24.4 mK. This shows that the heat input from the outside world is greatly reduced. The reference thermometer was bolted directly to the mixing chamber, yet the heat traveling down the thermometry cabling managed to raise the temperature of the thermometer. The filtered thermometer did not "see" this heat input and remained cooler.

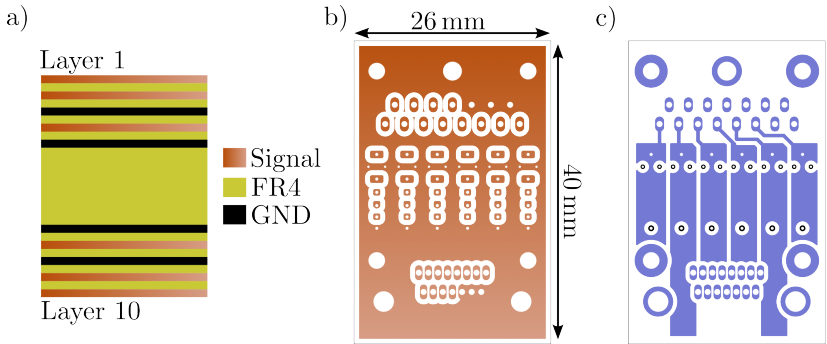


Figure 3.6.: Layer stack a) front b) and thermalization layer c) of the filter PCB. In a) only the copper layers (shiny metallic and black) are counted.

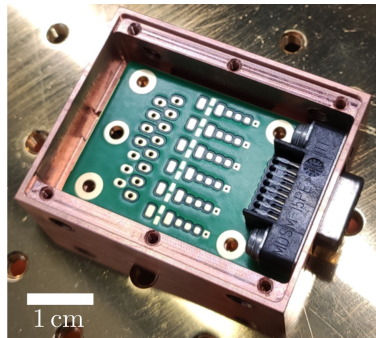


Figure 3.7.: The filter PCB. It is partially assembled and inserted into its housing.

3.3. Radio-frequency suited wiring

Besides providing the static voltages needed to tune the working point of the sample, the modern semiconductor qubit setup has to be able to deliver fast Radio frequency (RF) signals. Depending on the qubit type at hand, these can range from pulsed sine waves in the case of qubits which are driven by electron (dipole) spin resonance [57, 52], to pulses with complex shapes in the case of qubit schemes based on exchange interaction [88]. In other words, the desired system can reliably transmit frequencies of up to 18 GHz and has a flat frequency response from DC to 18 GHz.

The uniform frequency response is especially important for manipulation schemes based on non-sinusoidal pulses. Every complex pulse shape can be described by a weighted sum of its frequency components. If the system has a different transmission coefficient for the constituent frequency f_1 in comparison to the constituent frequency f_2 , then the total pulse shape will be distorted, since the sample will see another frequency superposition than intended by the experimentalist. As an example, if the waveform shown in Fig. 3.8 would pass through a transmission system with an RC-like filter function with a cutoff frequency of 10 MHz, as is depicted in Fig. 3.9, the resulting waveform (Fig. 3.10) would just barely resemble the input.

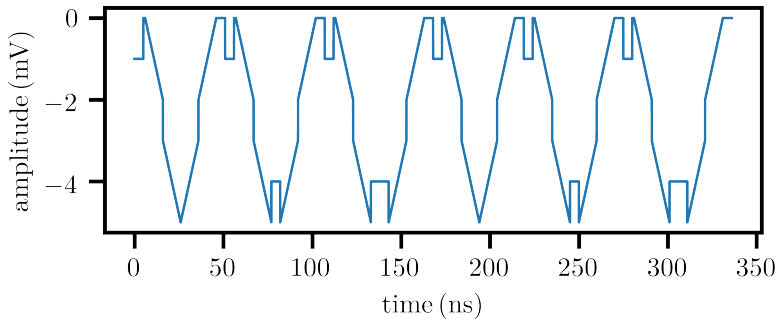


Figure 3.8.: Pulse shape used in e.g. free induction decay experiments.

As a direct consequence of this, the physical components in the drive lines have to be restricted to nominally frequency independent parts. Further requirements placed upon the drive lines are that heat radiating from room temperature and higher temperature stages inside the cryostat does not significantly raise the temperature of the mixing-chamber plate and sample.

To reduce the heat conduction by the electron system of the cable a material with a relatively high resistivity is employed. Furthermore the resistivity of the cable material should be approximately temperature independent, the material should be bendable by hand and ideally solderable without the need for specialized fluxes. CuNi fulfills all of these requirements, therefore 1.12 mm CuNi coaxial cables with a teflon dielectric were used in the construction of the fridge. While the intrinsic resistivity of the cables insures that heat conduction is limited in the case of an undisturbed system, the electromagnetic

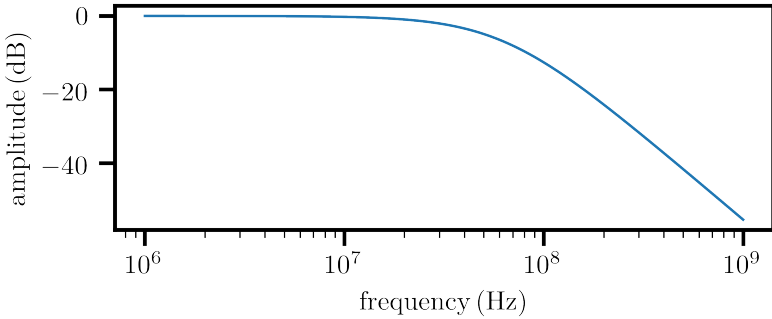


Figure 3.9.: Supposed filter function of the fridge cabling.

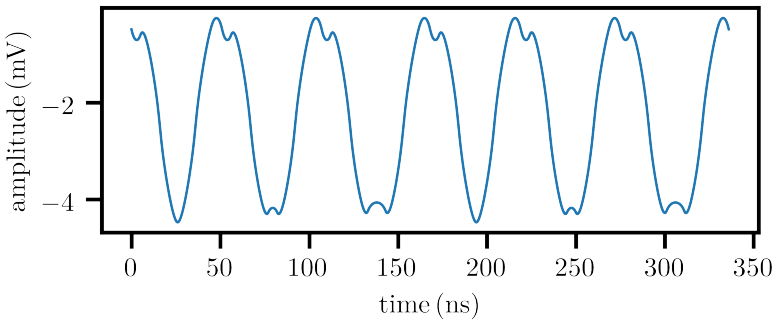


Figure 3.10.: Pulse-shape reaching the sample.

transmission of black-body radiation through the coaxial wires introduces the need for further components. While not being noticed as direct heating on the system level, the heat transported by radiation is directly transmitted to the most sensible parts of the sample and even minute amounts of thermal energy can cause excessive heating of the observed (usually quite tiny) regions. Attenuators provide protection from heating caused by black body radiation traveling down the coaxial lines. Choosing the correct amount of attenuation is a careful balance in between noise reduction (maximum attenuation) and signal integrity (minimum attenuation).

To get a grasp on the noise level that the sample needs to be protected against, we follow van Dijk et al. [13]. To achieve 99.9% exchange-gate fidelity, a 10 bit voltage-resolution is required in between $\frac{E_c}{e}$ and $-\frac{E_c}{e}$, where E_c are typical charging energies involved in the quantum dot system under test. We define the upper bound for the acceptable noise level as one order of magnitude below the minimum voltage increments dictated by the 10 bit resolution criterion. With a conservative estimate of charging energies being in the 10 meV [77] range, we can put an upper bound on the noise amplitude $A_{Nmax} \approx 2 \mu\text{V}$. Since we know the cabling to have an impedance of 50Ω , we can directly translate the noise amplitude to a maximum noise power of $P_{Nmax} \approx 80 \text{ nW}$.

Before calculating the needed attenuation there is one important detail that has to be factored into the attenuation calculation. Every attenuator does not only short part of the incoming radiation to ground (taking it out of the coaxial wire, essentially), but also emits radiation itself due to its finite temperature. The noise power attenuated and emitted by an attenuator at a certain temperature stage can be calculated using the following equation [65]:

$$P_i = \frac{P_{i-1}}{A_i} + \frac{A_i - 1}{A_i} P_{iBE} \quad (3.1)$$

P_i is the total (non thermal, due to the influence of higher temperature stages) output power of the i -th stage. A_i is the attenuation installed on the i -th stage, and P_{iBE} the total noise power of the emitted black-body radiation of the i -th stage. Usually, these noise powers are not known directly, but rather the temperatures of the corresponding stages. Thankfully, it is easy to calculate the total power emitted by a one-dimensional black body at temperature T . We assume a one-dimensional black body, since only modes in direction of the cable can travel down the coaxial line:

$$P_{iBE} = \frac{(\pi k_B T)^2}{6h} \quad (3.2)$$

Where k_B is the Boltzmann constant, and h is Planck's constant. Performing the calculation using the formulas above, we conclude that an attenuation of 20 dB at the 4 K stage, 10 dB at the 1 K stage, and 3 dB at the mixing chamber plate result in noise level of $P_N = 21.7 \text{ pW}$. This is three orders of magnitude below our defined acceptable maximum noise power.

With the choice of cable material and line circuit elements out of the way, the wiring can be installed in the dilution refrigerator. While constructing

cryogenic cabling there are a few things to keep in mind: When cooled down the cabling and the fridge will shrink, at different rates. Therefore the cabling should be slightly longer than required. Furthermore the cabling should be thermalized at every temperature stage it crosses. If it is not fed through via the usage of feedthroughs, then soldering a copper wire to the outer shield and connecting it to the temperature stage will help thermalize the outer shield of the cable if above 4 K. If below 4 K, it is advised to glue these thermalizations with a conductive glue, like epoxy (UHU Endfest 300, in this special case) infused with silver powder, since tin becomes superconducting at 3.7 K, and therefore loses most of its heat conducting capabilities. A circuit diagram of the wiring is found in the top box of Fig. 3.11.

So far we only discussed signals entering the fridge, but we are not only determined to change the qubit state, but we wish to read it out as well. Therefore another type of line, a dedicated readout line is included into the setup. While many readout schemes are present, RF-reflectometry [91, 4, 12] is becoming a standard method for the RF-readout of semiconductor quantum circuits. To perform reflectometry the circuit shown on the bottom half of Fig. 3.11 is employed. A carrier wave signal is sent down into the fridge via an input line (the details of which we already discussed), until it reaches the millikelvin plate. Here, it is fed into another RF-line via the use of a directional coupler. The forward coupled port is terminated with $50\ \Omega$ therefore any wave component which is forward coupled is dissipated. The coupled part now travels to the sample, is (partially) reflected and directed back through the coupler into a SiGe-based High electron mobility transistor (HEMT) amplifier circuit. To minimize the losses of the signal containing information about the sample state, the cable connecting the input port of the coupler and the input port the HEMT is a superconducting NbTi cable. Furthermore it provides thermal insulation in between the 4 K stage and the directional coupler mounted at the mixing-chamber plate. After being amplified by the HEMT, the signal is guided out of the fridge using a non-attenuated CuNi cable.

3.4. Magnetic Field coils

A further crucial ingredient to the spin-qubit setup, are the magnetic field coils. They are used to define the qubit levels in the simpler cases, and in more

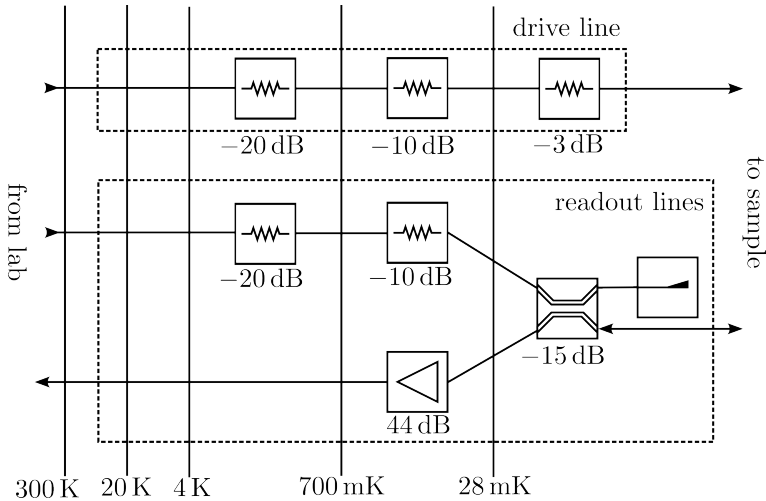


Figure 3.11.: Wiring diagram of the input RF-lines

involved designs, to isolate the qubit levels. The incorporation of magnetic field coils into a cryogenic setup can be divided into two equally important parts. The first part being the wiring that delivers the high currents from a room-temperature source to the 4K-stage, where the coils are located. The second part being the coils themselves. All wires used to connect the coils should have a low to non-existing electrical DC-resistance, to deliver high currents to the coils without introducing an additional heatload into the fridge. The choice of materials and cable composition is dictated by the temperature range the cable in question is crossing. For the connection in between room temperature and 70 K, a thick copper wirebundle is used. Bridging the gap in between 70 K and 4 K is done by a superconducting Yttrium-Barium-Copper-Oxide flat-band cable. Finally, the current is transported from the 4K stage to the coils using a NbTi wire ($T_c = 10$ K), whose mechanical stability and thermalization is provided by an adjacent stiff copper wire.

The design goals of the actual magnet are the following: The design should be able to generate a freely-rotatable magnetic field, meaning being a three dimensional vector magnet. Furthermore the system should provide an in-plane field (from the sample's perspective) of 1.5 T, and up to 500 mT in the out-of-plane and second in-plane axis. In addition, the system should be as

small as possible. This has many benefits, especially for cryogenic operation. First and foremost, a smaller system can be directly translated to a lower thermal mass. Therefore cooldown times are greatly reduced. The smaller the system, the more concentrated the magnetic field. The less volume is engulfed by the magnetic field, the less energy is stored in it. In the case of a quench (a rapid, abnormal transition from the superconducting to the normal conducting state), the current through the coil changes abruptly, and the energy stored in the magnetic field discharges. In case of a large coil system and therefore large magnetic field this can lead to catastrophic failures including destruction of the whole fridge. A small system however fills only a small amount of space with its field (while maintaining the same amplitude in the filled area) and therefore the amount of energy released in a quench is drastically reduced, eliminating the need for additional shunting circuitry like expensive and bulky Zener-Diode clamps.

The coil system shown in Fig. 3.15 presents a real-world compromise of the aforementioned design goals. The system has a footprint of only 38 mm x 38 mm, and a height of 34 mm. The magnetic properties were simulated by a finite-element method using Comsol multiphysics, and the results are shown in Figs. 3.12 and 3.13. To speed up the simulation, each coil was simulated individually, and the rotational as well axial symmetry were used to effectively reduce the three dimensional simulation to a two dimensional problem. The calculated parameters are presented in Table 3.1. The maximum current is

direction	central field [mT A ⁻¹]	resistance [Ω]	max. current [A]
x	60	970	8.03
y	56	690	8.17
z	180	407	8.61

Table 3.1.: Summary of the coil parameters.

reached, once a small section of superconductor in the coil reaches its critical field for that current. The dependence of the critical field as a function of the current is usually supplied by the manufacturer. An example of a quench estimation is shown in Fig. 3.13.

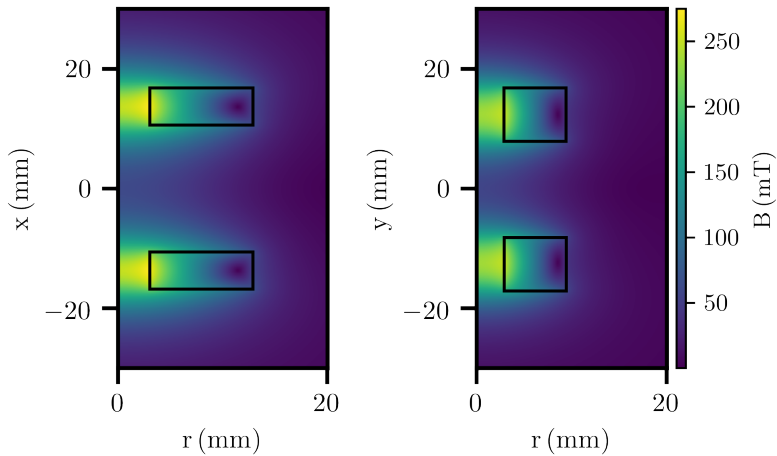


Figure 3.12.: Spatial distribution of the magnetic field of the coils in x and y direction. The plot shows the color-coded absolute value of the magnetic field.

3.5. The $^3\text{He}/^4\text{He}$ dilution fridge

The fridge is used to reach mK-temperatures. Fig. 3.14 a) shows the assembled fridge and Fig. 3.14 b) shows the sample holding assembly. The basis of this setup is a Sionludi-XL fridge. Unlike most conventional cryostats, the Sionludi features an upside-down design, meaning the coldest part of the cryostat, the mK-plate, is at the top of the assembly, and not at the bottom like in concurrent products as in other commercially available systems. In the present case, the mK-plate being on top still holds true, but the exact assembly is more intricate. The magnetic field coils have to be thermalized to the 4 K-stage (due to the heat input of the high current lines and the possibility of a quench), while still enveloping the complete sample, which is thermalized to mK. This in turn means that an additional 4 K plate, physically located above the mK-plate is needed. This plate is elevated by and thermalized to the original 4 K-plate by a stable shield made out of copper. This shield is featured in Fig. 3.14 a). To be able to exchange samples quickly and easily, the sample holder unit shown in Fig. 3.14 was used. Since this assembly was developed for top-loading, and potentially exchanging samples while keeping the cryostat cold a second

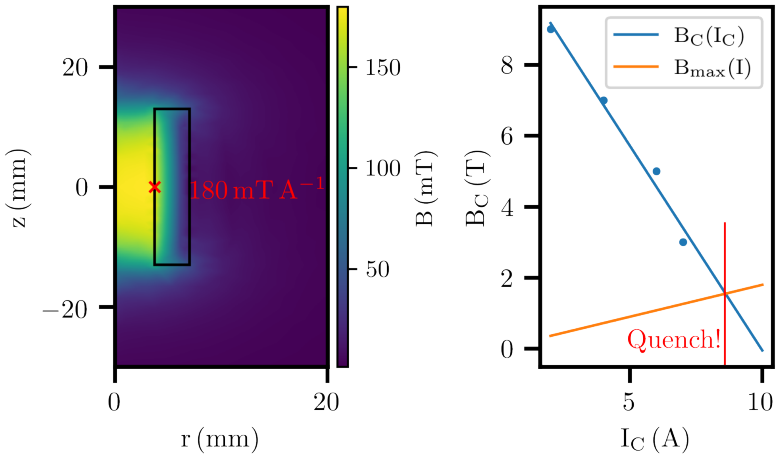


Figure 3.13.: Magnetic field distribution of the z-coil and visualization of the quench estimation.

mK-plate, physically located above the original as well as the elevated 4 K plate had to be added resulting in the assembly seen in 3.14 a) 3, for which a detailed view is shown in Fig. 3.15. The sample holder 3.14 b) is built to be plug-and-play. It is inserted into the mK-setup sample first. The cold finger is encased in an additional mK-shield, which is located inside and concentric to the magnetic field coils. All connectors on the sample holding assembly are secured by pressure only, there is no need for tightening of RF-connectors or nuts and bolts. The DC (up to 50 V, and few mA.) is provided via the split-PCI connection, and RF via the SMP-connectors. The three 4 mm banana-plugs provide mechanical stability after plugging the sample, as well as securely grounding the shield. In Fig. 3.16 a system level view of the fridge setup is presented. The different colors indicate different types of connections. The thermal anchoring of the magnetic field coils is symbolized by the black cross connecting it to the 4 K stage. All connections are thermalized at each temperature level they cross. While most wiring types present in Fig. 3.16 have already been discussed in detail in 3, there is one exception, the input lines carrying the current to the heating unit. These lines consist out of two twisted pairs of phosphor-bronze wires.

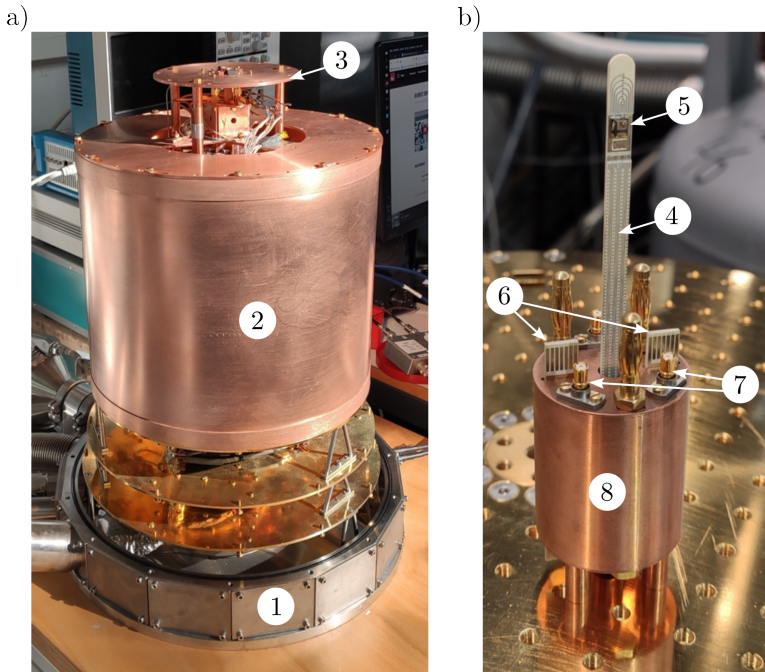


Figure 3.14.: a) 1, the dilution fridge. 2, 4K-shield. 3, mK-setup. A detailed view is found in 3.15. b) The sample holder unit. 4, cold finger. 5, the actual sample. 6, split-PCI-connection for up to 24 DC-lines. 7, SMP-connectors for up to 3 RF-contacts. 8, radiation shield hosting the second filtering stage.

3.6. Turbocycle heating system

The cooldown-to-cooldown variation of the stability of SiGe devices can vary significantly (see Fig. 5.15). Therefore it is beneficial to include a heating unit in the fridge which allows for a fast thermal cycle of the sample from mK to room-temperature and back to mK again. The heating system which was used in the scope of this thesis is depicted in Fig. 3.17. To perform a thermal cycle, the normal mode operation of the fridge is stopped, and all the $^3\text{He}/^4\text{He}$ -mixture is collected back to the tank. This serves two purposes. The obvious one being that, when heated, the mixture will dramatically expand which may cause mixture loss to detonation, depending on the existence of safety valves. The other purpose is to thermally decouple the mixing chamber stage

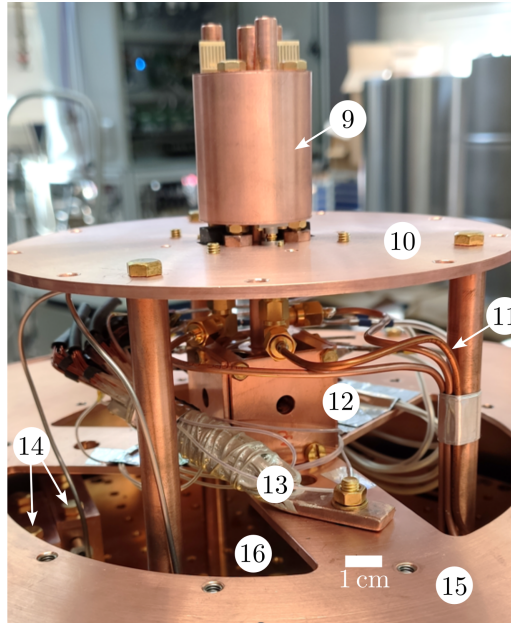


Figure 3.15.: The mK setup. 9, the sample holding assembly inserted upside-down into the mK-setup. 10, elevated mK-plate 11, coaxial cabling for RF-signals. 12, superconducting 3D-vector magnet. 13, thermal anchor for the high current lines supplying the magnet. 14, DC-filters. 15, the elevated 4 K-plate. 16, the actual mK-plate.

from the fridge, since it is thermalized to the other temperature stages by the gas flow through the mixture circuit. After the gas mixture is collected, the 4 K stage is flooded with a high flow of helium (10 L min^{-1}), to keep the fridge as cool as possible during the heating process. Next, the heating current is switched on. This is done in a two-step process. First the sample assembly is warmed with a total heating current of 1.3 A, resulting in a heat input of 39 W. After room-temperature has been reached, the current is reduced to 1.18 A, to keep the the sample assembly at room-temperature for ≈ 10 min. In the next stage, the current is switched off, and the mixture circulation is turned back on, cooling the heated stages back down again. This type of automated thermal cycle can be performed in less then two hours. A temperature versus time curve is shown in Fig 3.18

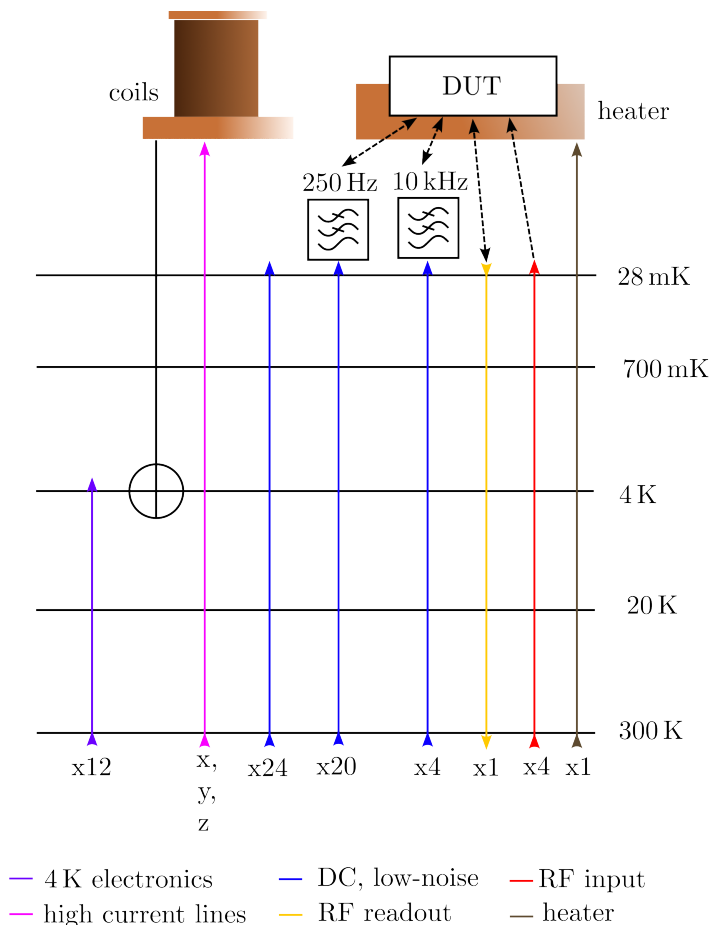


Figure 3.16.: System level wiring diagram of the fridge.

3.7. Experiment wiring

While the fridge and software system are very modular and configurable in many ways, let's constrict ourselves to a practical example in this section and investigate the system-level setup used in the bias cooling-experiments (chapt. 5). The measurements are performed in a synchronous way, using `spin_tune`,

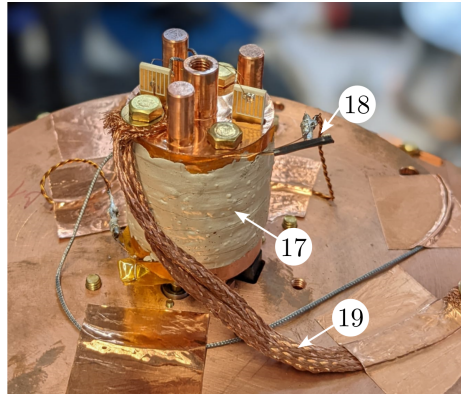


Figure 3.17.: Turbocycle heating system. 17, sample holding assembly wrapped with resistive heating wire, encased in silver-epoxy. 18, phosphor-bronze heating lines, delivering the heating current from the fridge input terminal to the mixing chamber stage. 19, thermal anchoring of the sample holding assembly to the millikelvin stage.

which is discussed in more detail in chapt. 4. The main takeaway from that information for the wiring diagram is that the computer is controlling the measurement flow. The peripherals are all connected using a standard ethernet protocol. Besides the computer and the fridge, there are four main devices present. The ADwin-Pro II by Jäger Messtechnik, the Keithley 2230-30-1 power supply, the UHF Lock-in Amplifier by Zürich Instruments, and current to voltage (IV) converter by Basel Precision Instruments (Baspi). In this configuration, the ADwin is used as a voltage source delivering static voltages to the sample, with practically no current flowing. The UHLFI is used as a lock-in amplifier. Its outputs are connected to the sample, and the returning current is converted to a voltage by the IV-converter and read via the signal inputs of the lock-in. The Keithley power supply is connected directly to the heater, and delivers the heating current to the heater described in section 3.6.

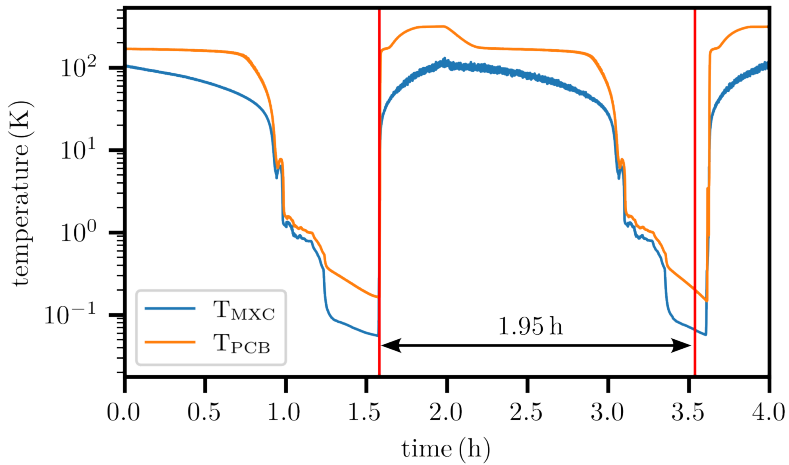


Figure 3.18.: A thermal cycle from millikelvin to room-temperature and back. The red lines highlight the beginning and the end of the thermal cycle, and are drawn at a temperature of 200 mK each. The blue curve T_{MXC} shows the temperature of the mixing chamber thermometer, which is screwed directly onto the mixing chamber. The orange curve T_{PCB} shows the temperature measured by a thermometer which is placed directly at the position of a future sample in sample holding assembly.

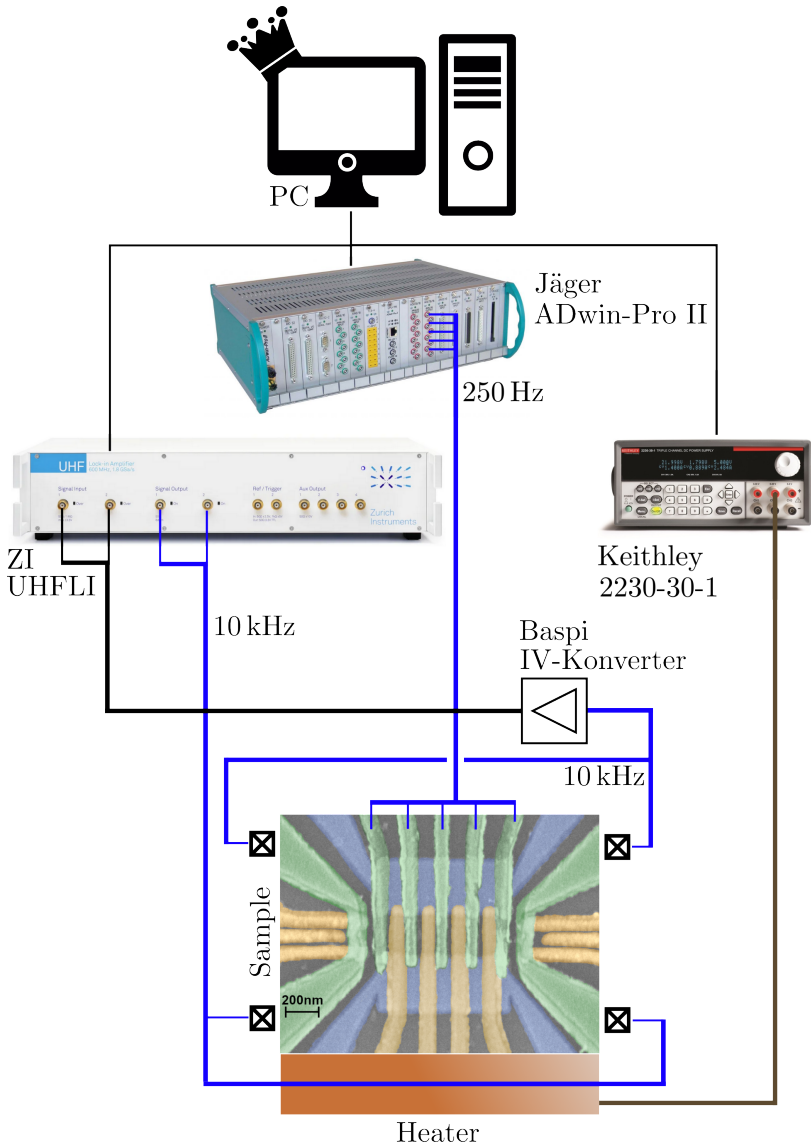


Figure 3.19.: Configuration of the fridge and outside instruments used in the experiments presented in 5. The colorcoding of the wired connections is consistent with Fig. 3.16. 250 Hz and 10 kHz denote the filtering used in the corresponding lines.

4. Measurement software

4.1. The concept of measurement

To the experimentalist a measurement device is a box with some (mostly BNC) connectors to interface it with the desired experiment. For a measurement control software, the measurement device is a defined set of interactions, among them a method to call to receive a value measured on a specified channel. A visualization of this concept is depicted in Fig. 4.1. The returned value can be of complex nature, such as the output for a lockin-amplifier. It demodulates the incoming signal and returns the in phase and quadrature components. Upon calling the measurement method, the control software will receive two values. These values are from now on classified as "data nodes", with a data node being a one of a set of values which is returned by the measurement device upon a singular call of the measure command.

4.2. Common software components

The measurement control software presented in the following sections shares common software components. These are briefly presented and discussed within this section, as they are stand-alone modules and can be easily re-used for other projects. One of these stand-alone-components is the `multiplexer`. It acts as a repository for measurement methods, with a measurement method being a function or a class method obeying the general definition in 4.1. If the measurement method of the multiplexer is invoked, all registered and active measurement functions are called sequentially, and their combined output is returned. Another shared component is the `Watchdog`. Within the `Watchdog` object, node bounds for different data nodes can be registered. Once the bounds for a specific node are known by the watchdog, the method `limit_check` can be invoked upon a specific return from a measurement. If

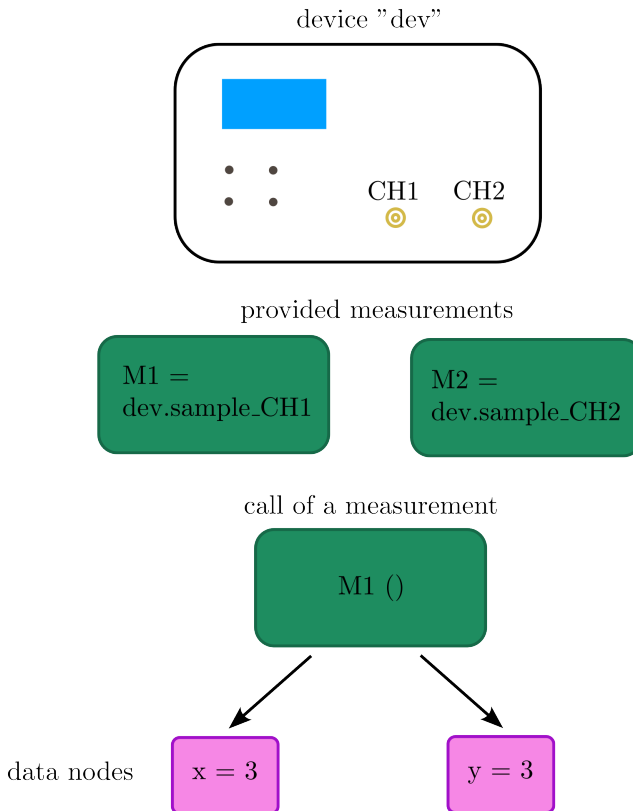


Figure 4.1.: Conceptual components of a measurement from the software viewpoint.

all registered data nodes lie within the specified limits, no further actions are taken. If a node value is detected which lies outside the specified range, the `Watchdog.flag` attribute is set to `True`, and the `Watchdog.message` attribute will display the details about which node passed the limits and whether the upper or lower bound was violated.

4.3. Non-realtime manipulation

To the user the division in between realtime and non-realtime manipulation may seem arbitrary, but from the software development point-of-view the two modes of operation differ greatly. When using non-realtime manipulation, the experimentalist's personal computer can act as control center for the whole experimental procedure. The general measurement flow is shown in Fig 4.2. In the first step the computer sends out instructions to various manipulation

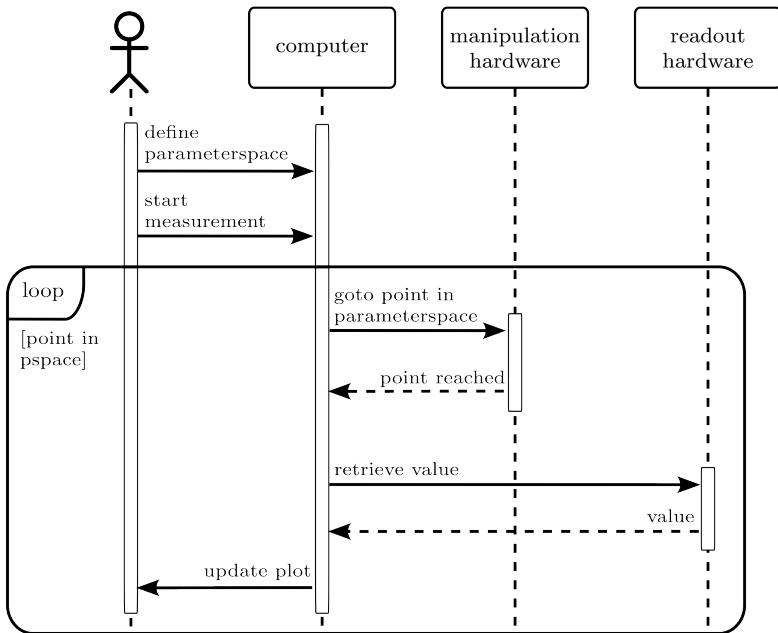


Figure 4.2.: Sequence diagram of a non-real-time measurement.

hardware to approach a specific point in the sample's parameter space, and waits until it is reached. Once at the working point, the computer requests a measurement, which is performed by the measurement hardware. The returned value is then stored to the hard-drive, and the procedure start anew. With this approach, the only commands which are needed are "go to value x" and "retrieve measurement value". The cost of this simplicity is, that the

timing in between the different "goto" and "retrieve" commands is determined by the computer, which limits the timing precision to ≈ 100 ms. Sadly, this lies far above the coherence times of most of today's qubits. Therefore, to measure a T_2 , a different approach is needed.

4.4. spin_tune

The main design goal of `spin_tune` was to develop a simple, yet flexible measurement control software for non-time-critical experiments. Measurement equipment is represented by functions which allow the transfer of measured values to the controlling computer (see 4.1), and the to-be scanned parameter-space of the sample is represented by coordinates. A coordinate contains the values at which the sample is to be probed and the functions to set the manipulation hardware to the desired points in parameter space. Upon invoking a one up to three dimensional measurement (the dimensionality is limited by `qkit`'s file system) the software follows the control-flow of a measurement with non-real-time manipulation, as is depicted in Fig. 4.2. Furthermore it is important to mention that `spin_tune` supports sequential multiplexing.

4.4.1. Usage

The usage of `spin_tune` is visualized in Fig. 4.3. In line 1 the measurement control object `Tuning` is created. The next step is to register the measurements which are to be performed. To register a measurement, the `register_measurement` function is called in lines 4 to 5. The arguments which need to be provided are the name of the measurements, in this case `lockin_channel1`, the dictionary containing the data nodes provided by the measurement and their according units, the measurement function itself, and additional arguments which are to be passed to the measurement function upon invocation. After defining how to record the sample state, the next step is to define how to change the sample state, defining the coordinates along which to measure. The definition of coordinates is accomplished via the usage of the `set_x_parameters`, `set_y_parameters` and `set_z_parameters` methods. When calling one of the above mentioned functions, the values which are to be set, the name of the axis, the function which actually interacts with the manipulation device and the coordinate of the unit have to be passed.

Optionally, the user can define bounds for specific data nodes. In case the recorded measurement value is outside the specified range, the measurement is stopped, and the last point in parameter space held until further commands are provided. Finally, the measurement is called in line 29. The measurement file is built sequentially, and the process is visualized live, by the usage of `qviewkit`.

```

1 spin_tune = Tuning()
2
3 #register lockin_channel1 as a measurement
4 spin_tune.register_measurement(name = "lockin_channel1", nodes = {"x": "V", "y": "V"},
5                               get_tracedata_func = lockin.sample_ch1)
6 #set node bounds for lockin_channel1
7 spin_tune.set_node_bounds(measurement = "lockin_channel1", node = "x", bound_lower = -3, bound_upper = 3)
8 spin_tune.set_node_bounds(measurement = "lockin_channel1", node = "y", bound_lower = -2, bound_upper = 8)
9
10 #register a second measurement
11 spin_tune.register_measurement(name = "lockin_channel2", nodes = {"x": "V", "y": "V"},
12                               get_tracedata_func = lockin.sample_ch2)
13
14 #define the parameterspace which is to be scanned
15 v_xrange = np.linspace(500e-3, 900e-3, 100)
16 v_yrange = np.linspace(300e-3, 700e-3, 100)
17
18 spin_tune.set_x_parameters(vec = v_xrange, coordname = "vbarrier1",
19                           set_obj = voltage_source.set_output1, unit = "V", dt = 0.1)
20 spin_tune.set_y_parameters(vec = v_yrange, coordname = "vbarrier2",
21                           set_obj = voltage_source.set_output2, unit = "V", dt = 0.1)
22
23 #Tweak the behavior while measuring
24 spin_tune.meander_sweep = False
25 spin_tune.report_static Voltages = True
26
27 #Run the measurement
28 spin_tune.activate_measurement("lockin_channel1")
29 spin_tune.measure2D()
30 spin_tune.deactivate_measurement("lockin_channel1")

```

Figure 4.3.: Definition of an experiment using `spin_tune`.

4.4.2. Use cases

4.4.2.1. 2D-Pinchoff of a quantum dot

The physical system used in this example is a gated SiGe heterostructure. A simplified diagram of the sample is found in Fig. 4.4. The goal of the following experiment is to pinch off the one-dimensional electron channel located underneath the top-gate RTG using the barrier gates RB1 and RB2. The anticipated dot region is marked in red. When the accumulation voltage applied to the top gate and the barrier voltages are within a particular

range, a quantum dot emerges in between the barrier gates. To conduct this experiment, the voltages applied to each barrier gate are to be swept, and the conductivity of the accumulated channel is to be measured. Therefore, two coordinates representing each Barrier gate voltage are entered, and a lock-in-measurement is registered. Upon starting the measurement a result

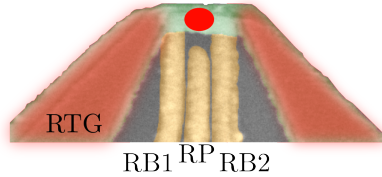


Figure 4.4.: Gate layout of the structure employed in the following use cases.

similar to Fig. 4.5 will be found. In the top right corner, both barriers are at a high positive voltage and there is a region of high conductivity. Starting in this region and following either the x- or y-direction a range is entered in which there is a sharp decrease in current. Here, either RB1 or RB2 block the current completely. A pinch-off which is caused by only one barrier can be determined by its direction in gatespace. If it is parallel to the axis, only one barrier is involved. If both barriers are involved, the pinch-off is tilted in respect to the axis, as is seen in the bottom left corner of the conducting area of the 2D pinchoff. In this specific case, the conductivity re-emerges at more negative gate voltages, which is a clear signature of the Coulomb oscillation of a quantum dot. The oscillation is present, because the barriers do not only change the tunnel coupling of the dot to the leads, but also have a cross capacitance on the dot itself, acting basically also as a plunger gate, shifting the dot level through the bias window. The oscillations are fading out since the barrier gates gradually close the conducting channel.

4.4.2.2. Measurement of the accumulation voltage

The measurement in this example is performed on the structure shown in 4.4. The goal of this experiment is to ramp all visible gates towards positive voltages, until a specified conductivity is reached, and record the voltage at which this is the case. Since all gates are ramped in parallel this is a

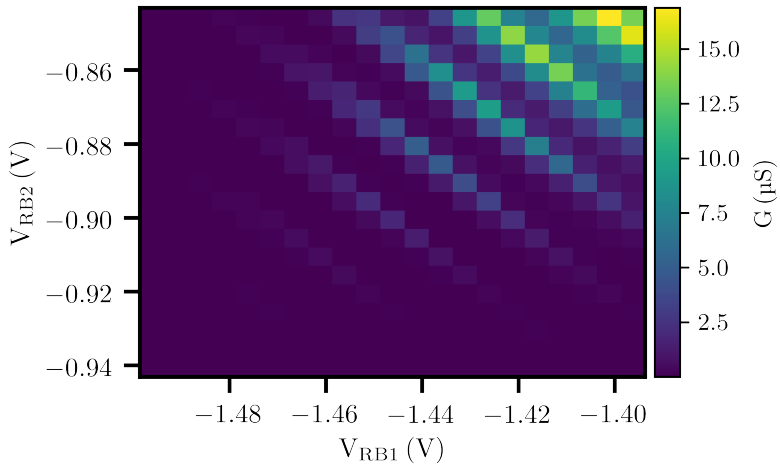


Figure 4.5.: Conductivity of the quantum dot as a function of both barrier voltages.

one-dimensional measurement. To conduct the experiment, a coordinate representing the global applied voltage is specified. The range of this coordinate extends beyond the expected onset of the desired conductivity. As is done in example 4.4.2.1, a lock-in measurement is registered. Furthermore, the node bounds of the conductivity node are set to the desired value, and the measurement is run. The resulting measurement is found in Fig. 4.6. As is evident, the conductivity increases with higher gate voltages. This is due to the Fermi-level being pulled inside the conduction band. It is important to mention, that shown accumulation curve is not the result of the measurement. The result of the measurement is the voltage at which the software was stopped by the watchdog.

4.5. spin_watch

spin_watch was developed to record the time dynamics of a sample in a steady state, while providing maximum flexibility for the user. As is the case for spin_tune, measurement equipment is represented by functions, and the to-be-scanned parameter space by coordinates. The main difference

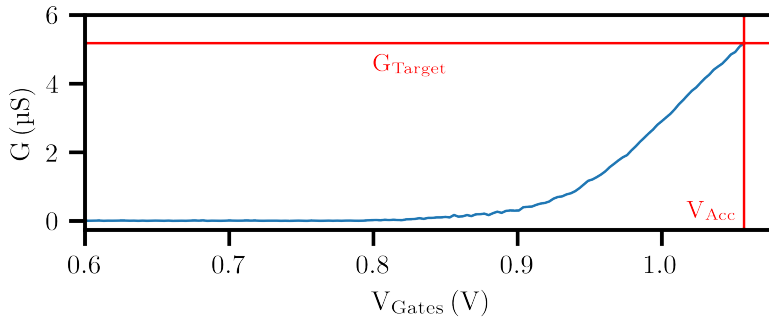


Figure 4.6.: Accumulation of the device towards the conductivity G_{Target} .

in between the two software modules is the way the measurement axis is handled. While in `spin_tune`, each value in the sample's parameter space is approached and measured sequentially, `spin_watch` has a different approach. As the point-of-interest in parameter-space is reached, the measurement hardware is started, and samples continuously. The computer repeatedly streams data from the measurement hardware, and maps it on the x-axis. The general measurement flow in `spin_watch` is visualized in Fig. 4.7.

4.5.1. Usage

The usage of `spin_watch` is analogous to the usage of `spin_tune` and follows the same paradigms, as is evident from figure 4.8. While most of the code is identical for both programs, there is a crucial difference in the function passed when registering the measurement. While in `spin_tune` the measurement function is expected to return the last measured value, in `spin_watch`, the function is expected to return all measured values since its last call.

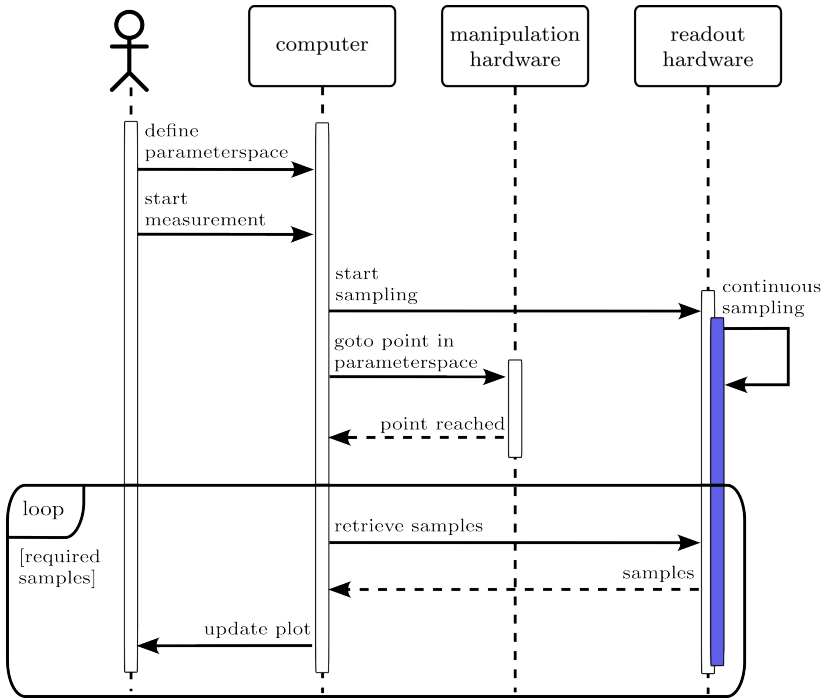


Figure 4.7.: UML sequence diagram showcasing the measurement flow in `spin_watch`.

4.5.2. Use cases

4.5.2.1. Measurement of the current noise on a single electron transistor

In this example the conductivity noise of a SET is measured at a specific pre-defined working point, sitting on the edge of a Coulomb oscillation, as is shown in Fig. 4.9. The goal of the experiment is to follow the jitter of the conductivity in real-time. While the exact timing of the onset of the measurement is not known, the time delay in between incoming samples is known, and determined by the measurement hardware. Therefore the time coordinate has to be defined based on the sampling rate of the measurement device. To communicate the data from the measurement device to the computer,

4. Measurement software

```
1 spin_watch = Watching()
2
3 #register lockin_channel1 as a measurement
4 spin_watch.register_measurement(name = "lockin_channel1", nodes = {"x" : "V", "y" : "V"},
5                               get_tracedata_func = lockin.stream_ch1)
6 #set node bounds for lockin_channel1
7 spin_watch.set_node_bounds(measurement = "lockin_channel1", node = "x", bound_lower = -3, bound_upper = 3)
8 spin_watch.set_node_bounds(measurement = "lockin_channel1", node = "y", bound_lower = -2, bound_upper = 8)
9
10 #define the time coordinate based on the measurement device's sampling rate
11 watch_time = 5
12 samples_to_watch = watch_time * lockin.get_ch1_sampling_rate()
13 t_range = np.linspace(0, watch_time, samples_to_watch, endpoint = False)
14
15 spin_watch.set_x_parameters(vec = t_range, coordname = "watching_time",
16                             set_obj = lambda val: True, unit = "s")
17
18 #Tweak the behavior while measuring
19 spin_watch.report_static_voltages = True
20
21 #Run the measurement
22 spin_watch.activate_measurement("lockin_channel1")
23 spin_watch.measureID()
24 spin_watch.deactivate_measurement("lockin_channel1")
```

Figure 4.8.: Use case code of spin_watch

we register the continuous streaming function of the lock-in-amplifier. The resulting trace is found in 4.10.

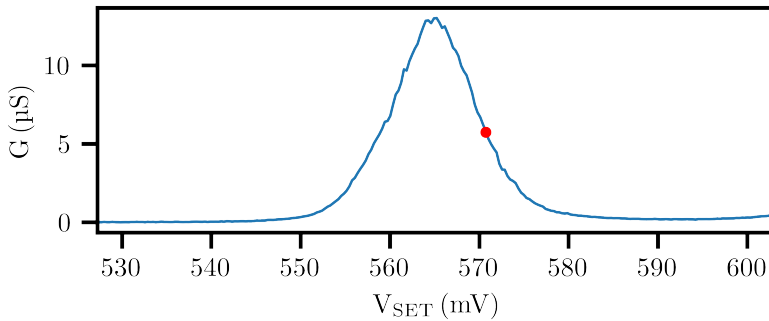


Figure 4.9.: Position in gatespace where the conductivity trace was taken.

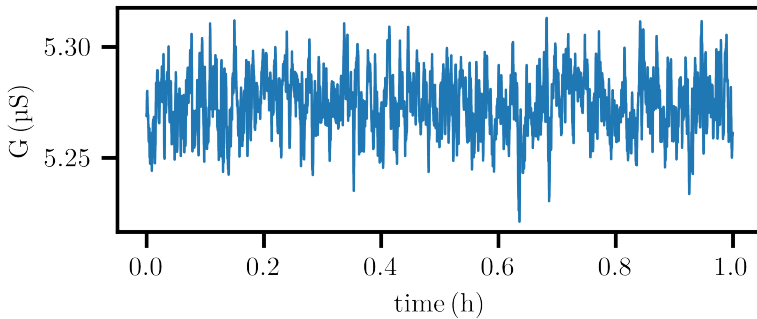


Figure 4.10.: Conductivity versus time of the SET.

4.6. Realtime manipulation

In contrast to non-realtime manipulation (see section 4.3), in realtime-manipulation the computer is not able to call manipulation or measurement start commands during experiment runtime. This in turn means that the whole measurement instructions have to be compiled and sent to the hardware before starting the experiment. A general sequence diagram of a realtime measurement is shown in Fig 4.11. During compilation, the manipulation hardware is provided with the full time course of each controlled parameter, as well as the triggering instructions for the readout electronics. The readout electronics in turn are provided with instructions concerning the recording time upon receiving a trigger, and how many records are to be collected until the measurement is regarded as completed. Starting the experiment, the readout hardware is armed and the manipulation hardware started. Upon receiving the first triggers, data will be accumulated, and written to internal buffers. The only task of the computer during the run time of the experiment is to stream data from the hardware, and show a live plot of the measurement progress. Since the readout is buffered, the non-real-time data transfer to the computer does not affect the experiment's time accuracy. The completion of the experiment is communicated by the readout hardware. Consequently, the measurement file on the computer is safely closed, and the hardware is reset to expect the beginning of a new measurement.

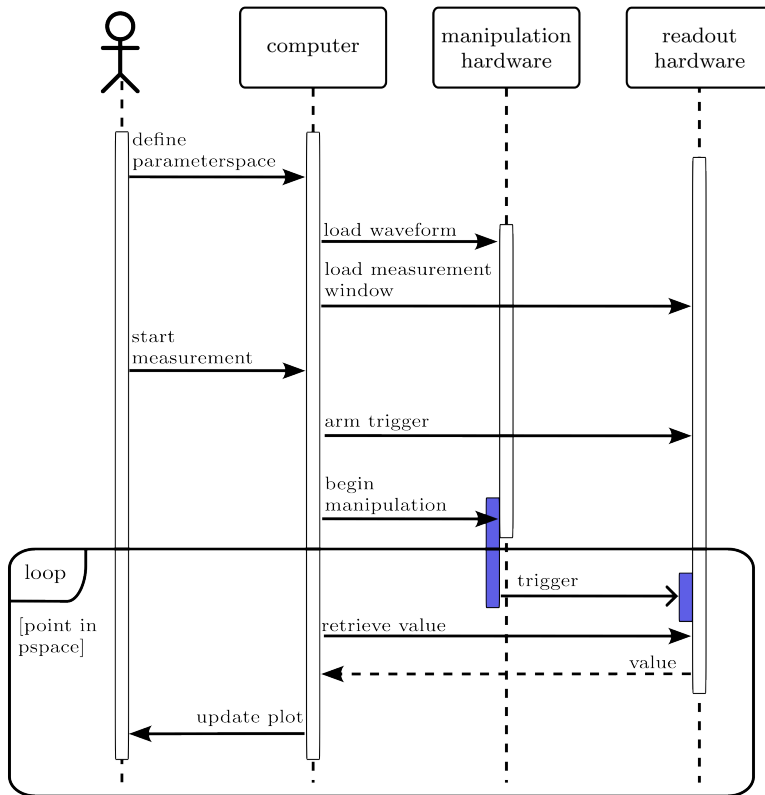


Figure 4.11.: Sequence diagram of a real-time measurement.

4.7. spin_excite

spin_excite was developed to be a compiler of virtual real-time experiments to existing hardware. Its design goals are hardware independence, ease of use and speed. Internally, the hardware is treated as an abstract set of instructions which are only translated to the machine specific commands in the very last possible moment. The inclusion of new hardware is made easy, since the only thing which needs to be provided is a translation in between the abstract

command and the machine specific code. The design of the experiment itself is done in with a virtual arbitrary waveform generator. The user defines arbitrarily shaped waveforms and measurement windows, to give a full set of instructions for the experiment. Furthermore `spin_excite` provides full parallel multiplexing. The software is able to handle multiple real-time experiments at the same time. After the experiments are compiled to hardware they are executed in parallel, data is streamed from each measurement device and saved to the local hard drive.

4.7.1. Usage

The general use case for `spin_excite` consists of experiments that measure the non-steady state of a system, like a pump-probe experiment. One of the easiest examples of this is the classic T_1 measurement. The typical experimental procedure in such an experiment is denoted in Fig. 4.12. The qubit is prepared in the ground state and excited by a π -Pulse. After a waiting time t_{wait} the state of the system is measured by the readout. Each readout will yield a binary answer, meaning the qubit will be found remaining in the excited or relaxed to the ground state. Since the experiment is repeated multiple times, the average of all binary answers will yield the probability of the system having relaxed to the ground state after the waiting time t_{wait} . To map out the full decay probability versus time it is necessary to not only measure after a specific waiting time, but to vary t_{wait} itself, meaning that the T_1 experiment consists out of a pulse train of excite-wait-measure sequences, with the waiting time being changed in each iteration. The whole process is repeated until all the desired averages are acquired.

In the following paragraphs, a way of implementing the T_1 experiment in `spin_excite` will be discussed.

4.7.1.1. Experiment design

As denoted in the UML use case diagram 4.13 the first interaction the user can take with the software is the compilation of the experiment, the translation in between a human-readable experiment format and a machine-readable format. An example for a human readable experiment-design package is `qupulse` [92], which is written in native python and actively developed by the RWTH-Aachen. In order to design a T_1 experiment with `qupulse`, four

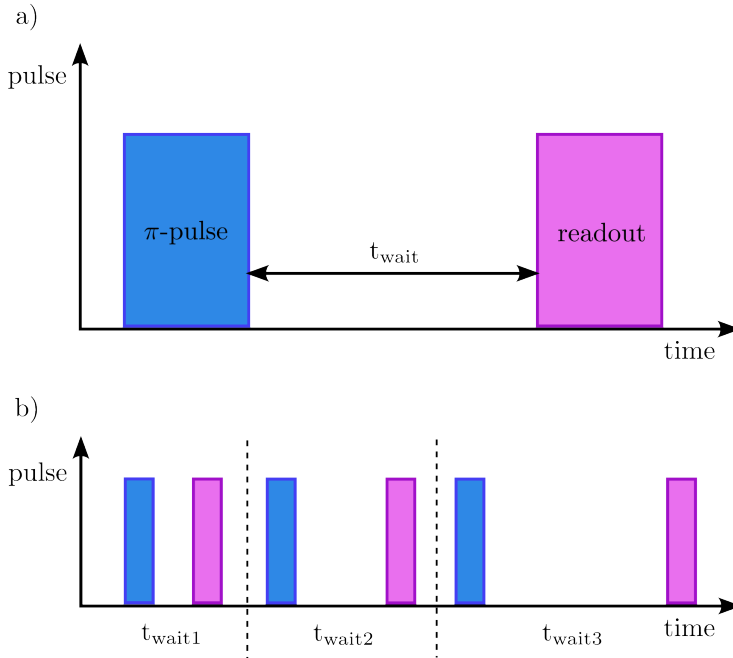


Figure 4.12.: Conceptual diagram of a realtime measurement. a) Pulse and measurement sequence to extract the excited state population after t_{wait} . b) Pulse and measurement sequence to extract the excited state population after t_{wait1} , t_{wait2} and t_{wait3} .

main steps are to be undertaken and showcased in Fig. 4.14. First, the user needs to define an excitation pulse (lines 2-5), then a waiting "pulse"(lines 8-11), define a measurement window and the trigger waveform for the measurement device (lines 9-11), and lastly combine all four components (line 14). This is a single iteration of the T_1 experiment. To map out the whole decay, the pulse is iterated within a for-loop (line 17), which extends the single pulse instance to a pulse train, in which the for-loop-parameter, t_{wait} is stepped. After creating the virtual experiment, it is passed to `spin_excite`. The creation of a realtime-measurement control object, called `Exciting`, requires the user to pass defined experiments as well as additional parameters, like the number of averages which are to be recorded for each measurement. This

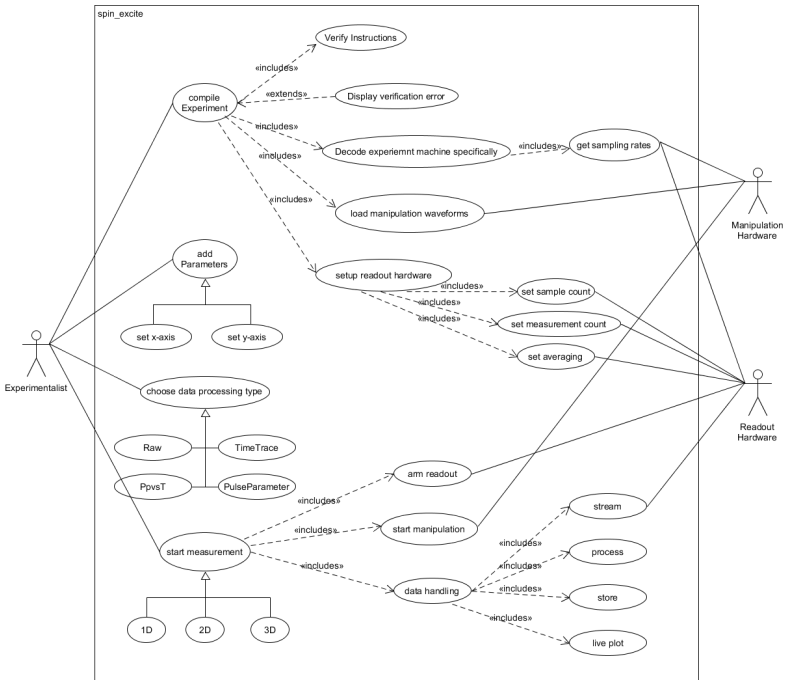


Figure 4.13.: UML use case diagram of spin_excite. The left hand side shows directly which interactions the software offers to the user (active actor). On the right hand side the interactions in between the software and the measurement hardware (passive actor) can be found.

is done in Fig. 4.15 in lines 1-4. If the user is interested in performing a T_1 measurement in dependence of an external parameter, for example a global magnetic field, additional parameter can be added (lines 7-8). To run the T_1 experiment versus magnetic field, the `measure2D` method is invoked (line 11). It is considered good practice to only invoke one Exciting Instance, and change the experiment by the usage of the `compile` function. To understand the machinations of the compilation process, the text will follow through the software components in order of their invocation during runtime.

4. Measurement software

```
1 #define excitation waveform on channel A, with trigger on channel TrigA
2 exc = FunctionPT("5 * sin(2 * pi * f * t)", "pulse_length",
3               channel = "A")
4 trigger = TablePT({"TrigA" : [(0, 0), ("pulse_length", 0)]})
5 excitation = AtomicMultiChannelPT(trigger, exc)
6
7 #define waiting "waveform" with consecutive trigger and measurement M1
8 waiting_time = TablePT({"A" : [(0, 0),
9                               "TrigA" : [(0, 0), ("t_wait", 3.3, "hold"),
10                                             ("t_wait + trig_duration", 0), ("t_wait + t_meas + post_measurement_wait", 0)],
11                               measurements = [("sensor_dot", "t_wait", "t_meas")])}
12
13 #combine excitation and waiting waveform to a T1 pulse
14 t1_pulse = SequencePT(excitation, waiting_time)
15
16 #create a pulse train to probe a range of waiting time
17 t1_pulse_train = ForLoopPT(t1_pulse, "t_wait", (0, 80, 5))
```

Figure 4.14.: Definition of a realtime experiment using qupulse.

```
1 spin_excite = Exciting(ro_backend, mac_backend, (t1_pulse_train, parameters),
2                   averages = {"M1" : 100}, active_modes = {"PulseParameter"},
3                   channel_mapping = {"A" : "Ch1", "TrigA" : "Ch2"},
4                   measurement_mapping = {"sensor_dot" : "M1"})
5
6 #define an additional non-realtime parameter
7 b_range = np.linspace(1, 1.5, 10)
8 spin_excite.set_x_parameters(b_range, "Bfield", current_source.set_output1, "T")
9
10 #run the measurement
11 spin_excite.measure2D()
```

Figure 4.15.: Upload of a defined realtime measurement to the hardware using spin_excite.

4.7.1.2. Virtual arbitrary waveform generator decoder

When calling the `compile` method, the virtual experiment is first passed to the virtual arbitrary waveform decoder. First, simple checkups are performed. Since all compiled experiments are run at the same time, no two experiments are allowed to share manipulation channels or readout measurements, because that would make their state ambiguous. Therefore `ValueErrors` are raised in case overlapping channels or measurements. Once the integrity of the instructions is insured, the actual decoding process begins.

First the measurement windows are decoded. The user defines a time during which the measurement has to record data. This is translated to the amount of samples which needs to be recorded. The sampling rate per requested measurement is enquired from the hardware, and then the number of samples is calculated. In case this results in a floating point number, the amount of samples is rounded down to the next integer, in order to prevent the measurement from becoming too long and possibly causing an unwanted overlap

with manipulation pulses which can distort the recorded data. Furthermore metadata about the measurements is collected. In `spin_excite` an experiment is completed once all measurements have fully acquired their samples per iteration and have acquired all iterations. Therefore the occurrence of each measurement (a single iteration) during the experiment is counted, and the total number of iteration per measurement is saved. In the next step, the requested manipulation waveforms are decoded. Since the virtual experiment contains instructions for the manipulation channels versus time, they have to be sampled. For this the sampling rates of the requested channels are inquired, and the waveforms are rendered based on the sampling rates of the manipulation hardware. The result of this are one dimensional arrays where each entry represents the voltage which is to be reached during the next DAC clock cycle. In the base configuration, a whole pulse train is rendered into a single one dimensional array. In case the `deep_render` keyword argument is used and set to `True`, each pulse of the pulse train will be rendered individually, and all pulses together are stored in a two dimensional `list`, where the first dimension is the iteration of the pulse in the pulse train, and the second dimension contains the voltage-per-clock-cycle array. In the current version of `spin_excite`, the decoder understands the `qupulse` language, other experimental design languages are not guaranteed to work. Nevertheless, if another design language should provide noteworthy benefits over `qupulse`, the only part which needs to be adapted is the Virtual arbitrary waveform generator decoder. The others do not need to be changed.

4.7.1.3. Generalized hardware interface

In the next step, the decoded information is passed to the generalized hardware interface. First, a simple checkup whether each requested channel and measurement is provided by the currently attached hardware is performed. If it fails, an `AttributeError` is raised, stating the missing channels or measurements. After the verification of the instructions coming from the virtual arbitrary waveform generator, they are sent to the measurement hardware. This is done in a two-step process. The generalized hardware interface calls the standardized commands for each measurement and channel, which in turn are translated into machine specific commands in the hardware adapters.

4.7.1.4. Hardware adapters

The only part that is hard-coded directly onto specific measurement devices are the hardware adapters. In essence, they provide the main code with information about each measurement and channel provided by the hardware and implement the device dependent form of the standardized commands. A hardware adapter is recognized by `spin_excite` by being a child class of either the `RO_backend_base` for readout devices or the `MA_backend_base` for manipulation devices, respectively. To create a hardware adapter an object must be generated which inherits one of the abstract backend base classes. To communicate to the rest of the Software that a specific channel or measurement is provided by a specific hardware configuration, the functions `register_channel` or `register_measurement` are used. It is recommended to invoke them in the constructor of the hardware adapter object. Once they are called, the user created adapter is scanned whether all standardized commands for a specific measurement or channel are implemented. Specifically, it checks whether classmethods with the the measurement or channel name as prefix and the standardized method name as suffix are defined. If the check is passed, a private variable containing the name of the measurement, the provided datanodes and the unit of the measurement are created. In case of a channel the variable contains the name and the unit, only. These variables are later on read by the generalized hardware interface and communicate to the system which channels and measurements are provided. Furthermore the abstract global methods contained within the backend bases have to be implemented, these include the loading of a waveform onto a manipulation device or the arming of the trigger for a readout device. It is important to note, that a hardware backend does not need to be limited to one device only. If the user wishes to interact with devices simultaneously both can be added into one hardware adapter. Measurements or channels provided by device one will contain those device's specific commands, whereas measurements and channel provided by device two will, in turn, contain its specific commands.

4.7.1.5. Datastream handling

After the successful compilation of the measurement instructions, the hardware is set up perform the requested experiment. To run the experiment three main commands are available to the user as seen in Fig. 4.13. `measure1D`, to start a measurement without, `measure2D` to start a measurement with one and

measure3D to start a measurement with two additional extrinsic coordinates. These coordinates do not contain a parameter which is to be iterated upon in real time, which rather would be included in the design of the experiment. However they can contain any parameter which is to be swept during the measurement which is not time critical, for example a global magnetic field, or the temperature of a cryostat. Upon calling a measurement function, a `h5` is prepared, containing the empty data structures of the experiment, which are to be filled on runtime. After the successful creation of the container for the incoming data, the readout electronics are armed, and the manipulation electronics started. During the main measurement loop, the measurement PC is continuously reading the buffer of the readout electronics, while querying whether the measurement has been completed. A measurement is recognized as completed once all samples for each iteration for each average are collected. This is done via the implementation of the `finished` function in the readout adapter. It returns `False`, while the aforementioned condition is not met and `True` after it is met. The data structure returned by the readout backend must obey the following conventions: It must be of a `dict` type. The first layer of the dictionary is keyed with the names of the measurements. The second layer is keyed with the name of the data nodes of said measurement. The values contained are `lists` of full received averages since the last time the buffer was read. In case no averages were completed, the `list` must be empty. An average itself is represented by a $m \times n$ matrix, where m is the number of parameter iterations (in the example of a T_1 -measurement, all measured waiting times t_{wait}) required for the completion of an average, and n the number of samples contained within each iteration (the response of the system, sampled in time). Basically, the raw measurement datastream consists out of vectors of matrices, where the last index denotes the number of the average, as is depicted in Fig. 4.16. In case one is interested in a quantity derived from the raw data stream, it is possible to treat the data live during the measurement. This is especially helpful since visual feedback upon the state of the measurement during runtime can help to identify faulty measurements before their completion, thus saving time. Live data processing is done using so called modes, which perform a mathematical operation on the raw data, for example averaging along the averages and samples axis. After all averages are collected, the `finished` function of the backend returns `True`, and the streaming loop is ended. In the last step, all hardware is stopped and reset to expect the next experiment. Additionally, all gathered data is plotted and saved into the folder containing the measurement.

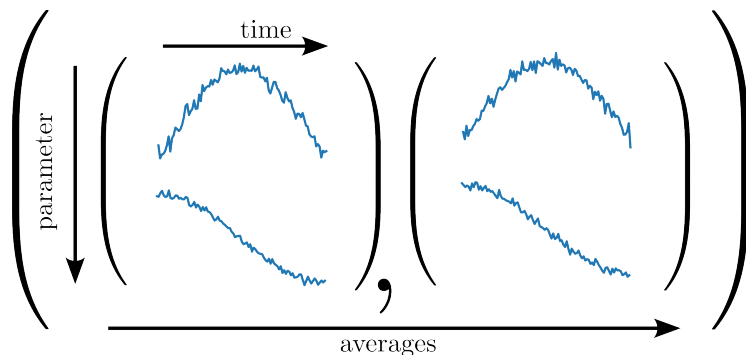


Figure 4.16.: Device layout used to perform the demonstration experiment. The-to-be loaded dot is located underneath PR. The reservoir under RTA serves as a source of electrons and is tunnel coupled to the dot. The SET is located underneath ST.

4.7.1.6. Modes

Currently, four data processing modes are included within the main code of `spin_excite`. Their function are summarized in table 4.7.1.6. Furthermore it is important to note that multiple modes can be used in at the same time. To activate multiple data processing modes in `spin_excite`, a tuple of strings containing the names of the modes which are desired is passed to the `active_modes` attribute. Due to limitations of the data storage handling in `qkit`, the maximum dimensionality of a stored array is three. Therefore modes that reduce the raw data to a dimensionality can be used with two additional coordinates, whereas modes that reduce the dimensionality to two can be used with one additional coordinate. In case the modes provided out

mode name	averaging axis	output dimensionality
PulseParameter	time, averages	1
TimeTrace	parameter, averages	1
PpvsT	averages	2
NoAvg	-	2

Table 4.1.: Basic modes in `spin_excite`, contained in the base version.

of-the-box do not fit the need of the planned experiment, new modes can

be added by the user relatively simply. On startup, modes are loaded from external python files into the main code. If the user wishes to add a new mode, a .py file containing a child class of the `ModeBase` (similar to readout and manipulation adapters) has to be added in the `/modes` directory relative to the `spin_excite.py` file. When creating a new mode object, the user has to provide implementations for the abstract methods defined in the `ModeBase`, and make sure that the inherited `create_tag` function is invoked upon startup. Further technical details are given in the documentation.

4.7.2. Use cases

In this next section, experiments performed on real-life devices are presented, to showcase the functionality of `spin_excite`.

4.7.2.1. Pulsing of a quantum dot above the $0 \rightarrow 1$ charge transition

The physical system used in this example is a gated SiGe heterostructure, similar to the one presented in 2.4. The focus is laid upon a single dot located underneath PR, tunnel coupled to a reservoir, whose charge is sensed by a proximal SET charge sensor located underneath ST, see Fig. 4.17. The goal of this demonstration experiment is to apply a pulse to the dot's plunger gate ST, and readout whether an additional electron has been loaded during the pulse. In the first step, a single iteration of the experiment is constructed, namely a square voltage pulse, with a readout window shadowing the pulse. V_{load} is the voltage additionally applied to the plunger gate, and I_{SET} the current through the SET. As the goal is to measure the response of the system to a pulse with increasing amplitude, the template is looped, and the pulse train for the experiment formed 4.18. As described in section 4.7.1.2, the pulse as well as readout windows are sampled, and sent to the corresponding hardware. `PulseParameter` is chosen as data processing mode. This choice has been made because the quantity of interest is the loading probability of the electron in dependence of the applied pulse height. The y-axis is defined by the readout of our system, while the x-axis is defined by the pulse parameter, the loading amplitude. Furthermore the output of the experiment is expected to be static during the time of recording (the electron is either loaded or not), and we are expecting similar results in each average, only affected by noise. Starting the experiment yields the result shown in Fig. 4.20. Coming from

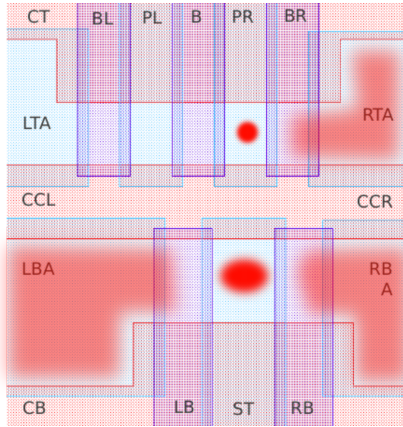


Figure 4.17.: Device layout used to perform the demonstration experiment. The-to-be loaded dot is located underneath PR. The reservoir under RTA serves as a source of electrons and is tunnel coupled to the dot. The SET is located underneath ST.

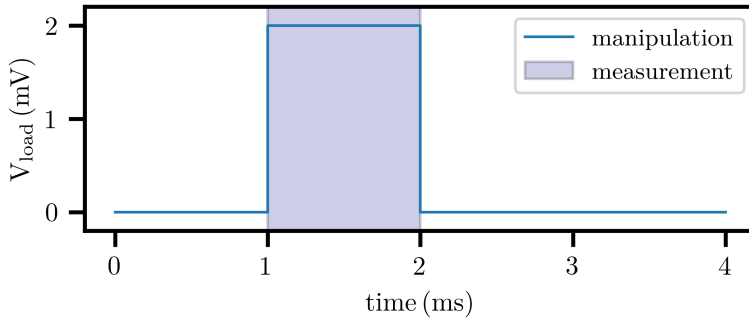


Figure 4.18.: Constituent pulse of the dot loading experiment. V_{load} is applied to the plunger electrode of the dot, and the measurement is performed while the pulse amplitude is high.

low pulse heights, the SET response stays constant until 12 mV, even though the pulse height is increased with each iteration. This leads to the conclusion that the cross capacitance in between the plunger of the observed dot and the SET is negligible in this voltage range. In the range of 12 mV to 20 mV, a washed out decrease in the SET current is observed. In this region, the pulse amplitude is high enough to allow an electron to tunnel into the dot, but

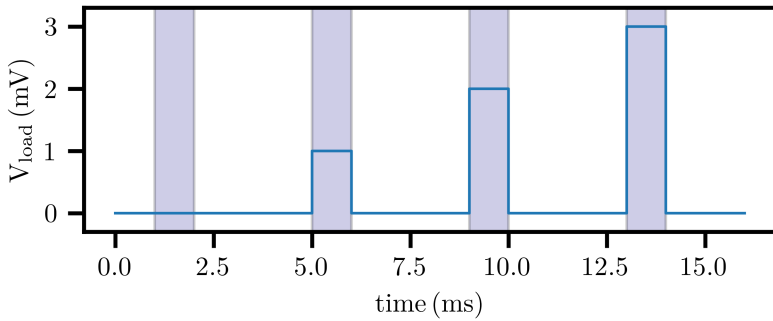


Figure 4.19.: Pulsetrain used for the dot loading. The pulsetrain is built from the constituent pulses shown in Fig. 4.18.

the energy difference to the reservoir is too low to make the electron tunnel with an approximate probability of one. In the remaining measured voltage range the electron is always loaded, and the SET current response lowered, accordingly.

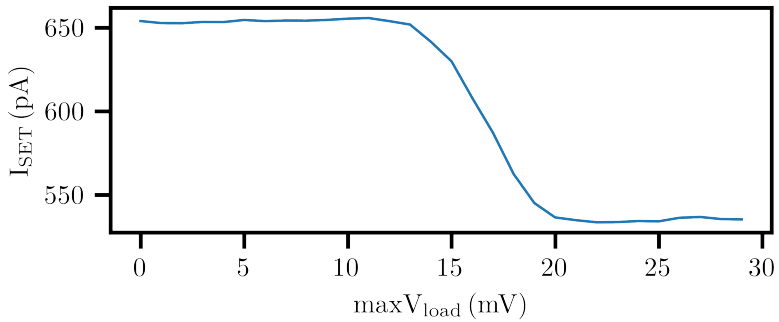


Figure 4.20.: Dot population versus pulse amplitude.

4.7.2.2. Determination of the Elzerman readout level

In the next example, the readout level V_{RO} for an Elzerman- T_1 [17] measurement is determined experimentally for the same device used in 4.7.2.1. The measurement scheme consists out of a three-step process, displayed in Fig. 4.21. First, both energy levels for spin up and spin down are pulsed below the fermilevel E_F , and kept there for a varying time t_{wait} . This is done with a positive voltage pulse of high amplitude, and duration t_{wait} . The pulse is shown in Fig. 4.22. In step 2., the excited energy level is pulsed above the reservoir, meaning that the pulse amplitude is reduced. This state is held for a fixed readout time t_{RO} , which is chosen according to the tunnel coupling. Possible excited electron will tunnel out of the dot in the vast majority of cases and will be replaced by electrons from the reservoir of opposite spin. These tunnel into the dot level below the Fermi level of the reservoir. For this step it is crucial to find the correct alignment in between the reservoir and the dot levels. The measurement window is chosen so that it covers the whole t_{RO} , as well as the end of the first step and the beginning of the last step. This is done to ensure that the whole readout window lies within the measurement window. In third step the electrochemical potential of the dot is increased (the pulse amplitude lowered further) such that both energy levels lie above fermilevel, the dot is emptied, and the system reset. The data expected in this experiment consists out of current traces of the SET, where sometimes an electron signature, a so called blip, is found. This signature presents itself as square shaped spike in the recorded tunneling current, as can be seen in Fig. 4.23. The blip is caused by the excited electron leaving the dot (current going up), and an electron from the reservoir tunneling into the dot (current going back down). As the blip is short in comparison to the measurement window, integrating the samples along the time dimension will not yield a clear signature whether an electron has tunneled or not. A further complication is that the blip is expected at a quasi random point in the readout window. Meaning, taking the average will smear out the response in time. Nevertheless, our goal is to find the correct voltage level for readout phase, and a high readout fidelity is not needed for this calibration experiment. To find the correct level, we employ the pulsetrain depicted in Fig. 4.24, where the readout voltage level is varied. To identify whether possible excited electrons leave the dot, and ground state electrons stay in the dot, we need to graph the response of our SET versus the measurement time, while varying the readout level. Since we are not interested in a high readout fidelity but

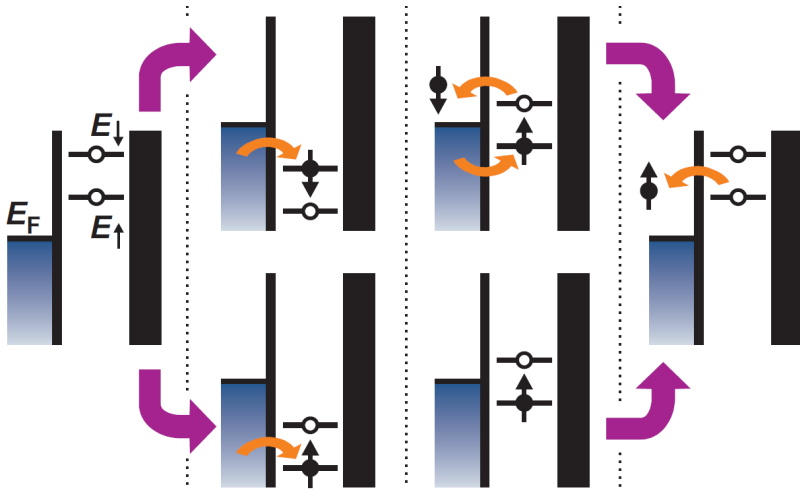


Figure 4.21.: Elzerman-readout-scheme of the spin state of an electron in a quantum dot. Adapted from [17].

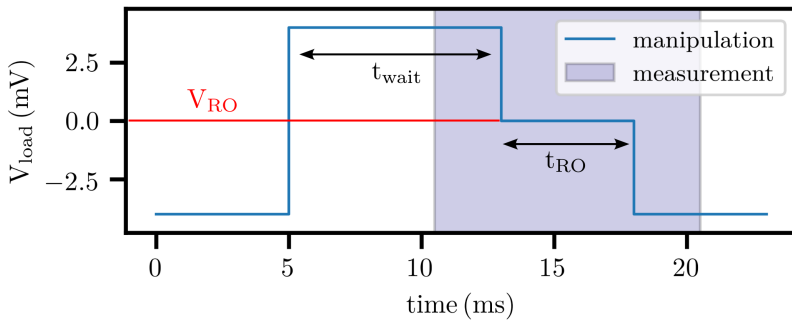


Figure 4.22.: Constituent pulse for Elzerman-readout.

rather in a quick calibration, an averaging of multiple blip events is acceptable and we shall choose the PpvsT mode. For pulse amplitudes which are too low (both energy levels are above the Fermi-level) the sensor response will remain mostly constant during the measurement, showing the signature of no electron in the dot. In our specific case, the signature of an empty dot is a

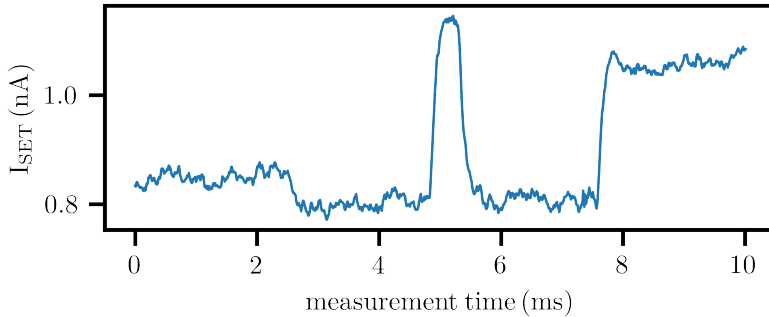


Figure 4.23.: Signature of an electron in the excited state (blip).

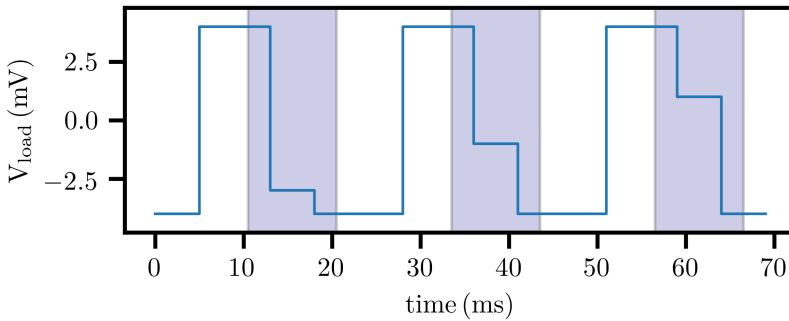


Figure 4.24.: Pulse train used to find the correct readout level.

high current through the sensor. For pulse amplitudes which are too high (both energy levels are below the Fermi-level) the sensor response will also be constant in time, showing the signature of an electron inside the dot, which in our specific case is a low current through the sensor. If the energy levels are aligned correctly, meaning excited above and ground state below the Fermi level, the sensor response will vary in time. The sensor will briefly show the signature of an empty dot (heightened current) as the excited electron left, followed by the signature of an electron on the dot (lowered current), as the dot gets occupied by an electron with the opposite spin state. This signature

is usually referred to as spin-tail, and is visible in Fig. 4.25 in between 0 mV to 15 mV. Therefore the correct readout level also lies within this range.

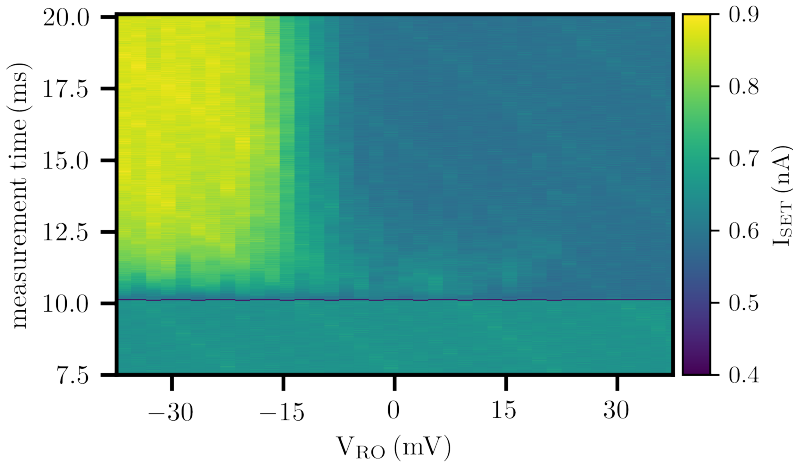


Figure 4.25.: Result of the calibration measurement to find the readout level.

4.7.2.3. Elzerman T_1 measurement

In this example, the spin-relaxation time T_1 of an electron in a quantum dot is determined. The experimental protocol is the same as in section 4.7.2.2, only this time not readout level, but the waiting time t_{wait} is varied.

As in the example above the data produced by this experiment consists out of electron blips, 4.23. As the blip is short in comparison to the measurement window, integrating the samples along the time dimension will not yield a clear signature whether an electron has tunneled or not. A further complication is that the blip is expected at a quasi random point in the readout window. Meaning, taking the average will just smear out the response in time. Since we are interested in a high readout fidelity a mode without averaging, the NoAvg mode, has to be chosen for data processing.

To conduct this experiment, a single iteration of the pulse 4.22 is loaded (with a specific holding time t_{wait}), and the experiment is repeated average times. For a single waiting time, this yields the data shown in Figure 4.26. The NoAvg mode simply appends all recorded timetraces into one matrix. The y-axis is defined by the measurement time, and the x-axis consists of the number of the recorded trace. To sample another waiting time, the experiment is loaded again with a different t_{wait} . This process is repeated until all desired waiting times are measured.

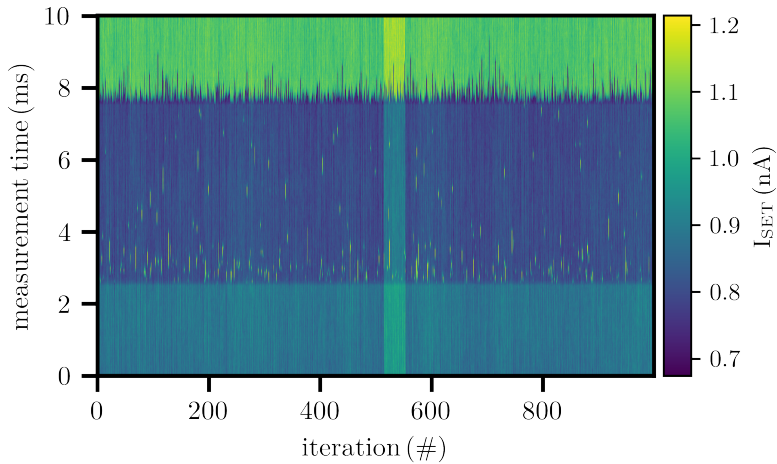


Figure 4.26.: Result of the Elzerman- T_1 measurement, for a specific waiting time t_{wait} . Each iteration consists out of a trace similar to 4.23.

5. Bias cooling of SiGe heterostructures

5.1. Introduction

In this chapter the results of a systematic bias-cooling study are presented. Bias-cooling refers to the method of cooling down a device with applied gate voltages. In this study, the same voltage is applied to all gates which are not ohmic contacts. SiGe devices suffer from slow charging effects and drift, which we quantify measuring the total noise power in the frequency band in between 200 μHz and 10 mHz. We track the noise power as a function of V_{BC} . V_{BC} is defined as the voltage which is applied to all gates during cooldown, starting from room temperature, until reaching a base temperature of 30 mK. We will show that bias cooling can drastically reduce the emitted noise power, and consequently investigate it in high resolution for two devices. Two further devices are measured with a lower resolution in the voltage range in between -1 V and 1 V . The turn-on voltage of the conductive channel in dependence of V_{BC} is measured for four devices in between -3 V and 2 V . More than 80 cooldowns were performed, on a total of four different samples.

5.2. Device layout

The devices used in this study are gated quantum dot Si/SiGe devices. The gate layout is depicted in the false-colored scanning-electron-micrograph image 5.1 and is nominally equal to the device from Ref. [98] and was kindly supplied to us by the Group of Prof. Dr. Hendrik Bluhm from the RWTH Aachen. Blue indicates the bottom most and green the top most gate layer. The gate stack is an overlay of three metal layers consisting of 15 nm, 22 nm and 29 nm Pt, on top of a 5 nm Ti adhesion layer. The metal layers are electrically isolated from

the substrate and from each other by 10 nm atomic layer deposited Al_2O_3 . The gatestack is fabricated upon a commercially available 8 inch virtual SiGe substrate, which consists of a $3\ \mu\text{m}$ $\text{Si}_{1-x}\text{Ge}_x$ graded buffer layer on a Si substrate, a strain relaxed $2\ \mu\text{m}$ $\text{Si}_{0.7}\text{Ge}_{0.3}$ buffer, a 10 nm strained Si layer, followed by a 30 nm $\text{Si}_{0.7}\text{Ge}_{0.3}$ spacer layer, and capped by 2 nm Si. The stack itself is depicted in 2.4, and the purpose of each layer is discussed in section 2.2. Ohmic contacts to the quantum well layer are realized by implantation of phosphorus ions and activated by rapid thermal processing at $700\ ^\circ\text{C}$ for 15 s. The device layout consists out of a central region harboring up to four

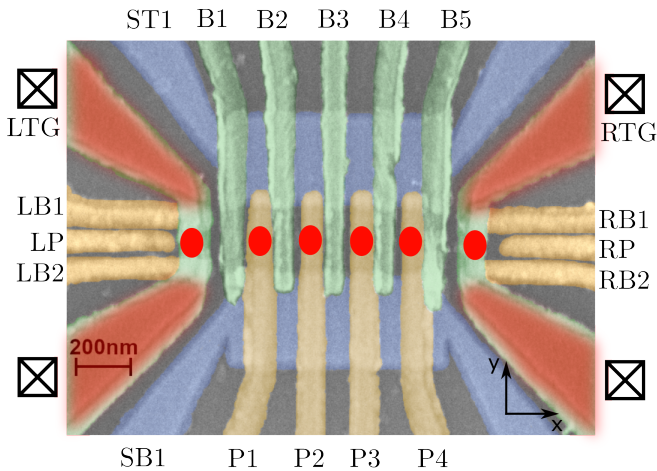


Figure 5.1.: False colored scanning electron micrograph of the Si/SiGe device. The three gate layers are represented by different colors. The bottom layer is indicated in blue, whereas the topmost layer is indicated in green. The heterostack itself is shown in Fig. 2.4, and the purpose of each layer is discussed in sec. 2.2. The red shaded areas show the presence of an accumulated electron channel, the crossed squares denote ohmic contacts, and the red dots show possible positions of quantum dots.

individual quantum dots, formed by the barrier gates B1 to B5, the plunger gates P1 to P4 and the two confinement gates ST1 and SB1. The dot region is enclosed to the left and right by two SETs. The SETs are formed by the accumulation gates LTG and RTG, and confined by the barriers LB1, LB2 and RB1, RB2 respectively. LP and RP serve as plunger gates for each of the sensing dots. Furthermore the conducting channel formed by LTG and RTG is in contact with the implant region, and therefore effectively creates the ohmic contacts for the device. The contacts are drawn as checked boxes in

Fig. 5.1. The dot region is accumulated by pulling the conduction band below the electrochemical potential by the plunger gates P1 to P4. To separate the dots from each other in x direction, the conduction band can be adequately modulated by the barriers B1 to B5, and to confine the dot structure in the y -direction, the two confinement gates are used.

5.3. Turn-on voltage mapping

After cooling the sample down from room temperature to 30 mK with the applied voltage V_{BC} , the turn on voltage is measured. The procedure is as follows: The bias cooling voltage is turned off, and all gates of the sample lie at zero volts. Since the sample operates in enhancement mode, at this point, no conductive channel is present. To ensure that the current channel is not affected by singular defects in the heterostructure, we ramp all gates in parallel to the same voltage V_{Gates} until we reach a conductance that corresponds $0.5 \mu S$. The voltage at which this threshold conductance is reached, we define as the turn-on voltage. A typical accumulation curve is shown in Fig. 5.2, and the turn-on voltage is referenced by the red line. This measurement procedure

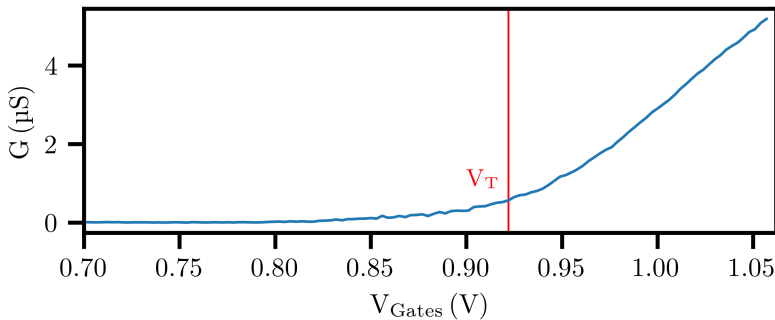


Figure 5.2.: Typical accumulation curve. Marked in red is the conductance of $0.5 \mu S$ which is used to define the turn-on voltage.

is repeated for each V_{BC} , with the use of the turbocycle heating system, which is described in section 3.6. The resulting turn-on voltage as a function of V_{BC} is shown in Fig. 5.3 for the four measured samples A , B , C and D . These

samples are all nominally equal, and were fabricated on the same wafer in the same batch. The first apparent effect of bias cooling is that the accumulation voltage of the devices shifts nearly linearly with the bias cooling voltage. This is due to charge carriers getting trapped in between the quantum well and the metal gates. The AlO_x -Si interface is known to host an interstitial SiO_x layer, which forms as a byproduct of the ALD process [40, 31, 39]. SiO_2 as well as SiO_x can incorporate aluminum atoms originating from the oxide, where it replaces the silicon and effectively forms an acceptor state inside the SiO_x [38]. The acceptor state is located 0.5 eV below the silicon valence band edge [56]. Spatially, the acceptors are located in a roughly 1.5 nm thick layer of silicon oxide, which forms directly at the Si- Al_2O_3 interface, so they are separated by a tunneling barrier from the silicon bulk [40]. The trap density is in the $1 \times 10^{11} \text{ cm}^{-2}$ range for as-deposited samples, and can be increased by more than an order of magnitude by annealing the sample [31]. Another source of charge traps is presented by dangling bonds at the silicon surface, which has the a defect density of $1 \times 10^{11} \text{ eV}^{-1} \text{ cm}^{-2}$ [40], nevertheless they should be passivated by the presence of the Al-acceptor states. Thus, we suspect that the shift in accumulation voltage stems from a change in population of the Al defect states. To verify our theory, we perform one-dimensional Schrödinger-

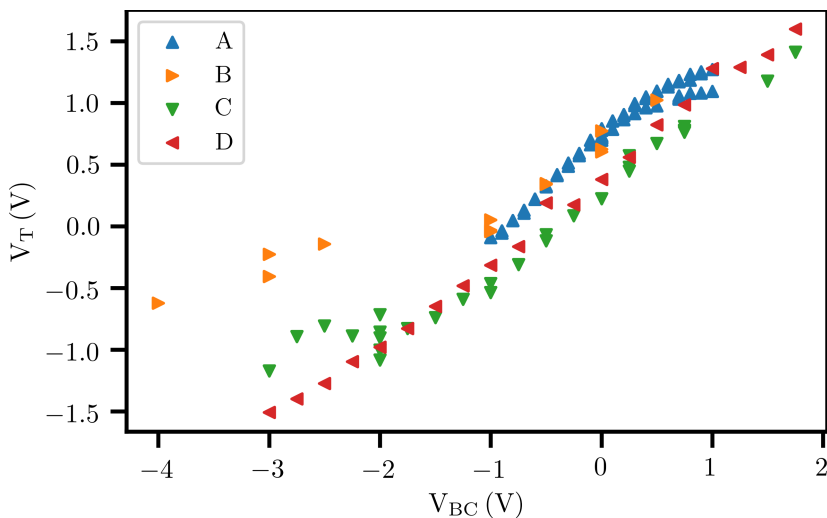


Figure 5.3.: Turn-on voltage of the device over applied bias voltage during cooldown for four different devices with nominal equal layer stack and gate design.

poisson simulations, in which we incorporate interface charges and monitor the accumulation onset voltage. For the simulation, we assume the layer stack at the interface to be the following, in order of the growth direction: 2 nm Si-cap, 1.2 nm SiO₂ and Al₂O₃ with thicknesses $t_{\text{Al}_2\text{O}_3}$ according to the gate layer of interest. To model the charged acceptors at the SiO_x-Al₂O₃ interface we homogeneously distribute charges in the first nm of Al₂O₃. The result of the simulation, complete with all three nominal gate layer thicknesses of the used device is presented in Fig. 5.4.

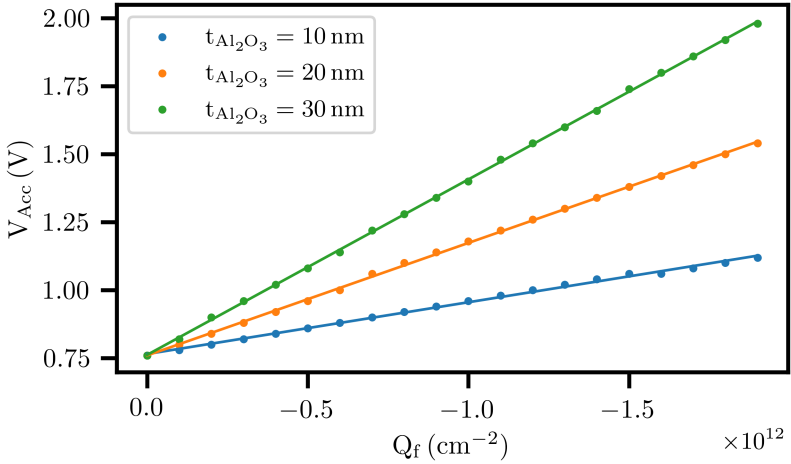


Figure 5.4.: Simulation of the accumulation voltage as a function of the interface charge density Q_f .

The interface charge density shifts the accumulation voltage linearly and therefore reproduces the behavior of the experiment. By comparing the measured shift in accumulation voltage to the simulated shift and assuming all gates to be located in the central gate layer with an oxide thickness $t_{\text{Al}_2\text{O}_3}$ of 20 nm we can extract the interface charge density that is induced with each volt of bias cooling, which is $\frac{Q_f}{V_{\text{BC}}} = -1.741 \times 10^{11} \text{ V}^{-1} \text{ cm}^{-2}$ and lies in good agreement with the literature values for the interface charge density. The real value might slightly deviate, since in the accumulation all gate layers with different oxide thicknesses take part in inducing the electron gas. Further it is interesting to note that the slope in between the accumulation onset

voltage and the bias cooling voltage is not unity. This could be due to either geometric effects or the finite density of trap states in the interface layer.

5.4. SET formation

The quantitative investigation of bias cooling is challenging, since the experimentalist is faced with a large parameterspace ranging from room temperature down to mK, the exact cooldown procedure and tuning of the quantum dot device itself. Therefore it is essential to follow a strict routine to exclude the influence of hidden parameters. For this work, the procedure can be described as follows: First the sample is cooled down from room temperature down to 30 mK. The cooldowns have been performed using the turbocycle heating system, and therefore follow very similar temperature profiles 3.18. Next, the two-dimensional electron gas is accumulated using all gates, and the gate voltage at which the 2DEG reaches a third of the saturation conductance of the sample (G_{sat}) is recorded and defined as the new reference voltage V_{ref} , see figure 5.5, panel one. With all gates at V_{ref} the device is in a state where a two-dimensional electron gas (2DEG) is accumulated in the silicon quantum well below every metal gate. In step two we lower the voltages of all gates except the topgates and barriers of the SETs by 500 mV to deplete the sample and confine the 2DEG only below the top gates. This value has proven itself to reproducibly deplete the electron gas. The goal of the following steps is to form a quantum dot in between the barrier gates LB1 and LB2. The great experimental challenge hereby is to tune the SETs to comparable working points, although the sample provides a different electrostatic environment during each cooldown. It has been shown that the noise seen by SET devices is dependent on their working point [16]. In step three (see Fig. 5.6, the top-gate voltage is increased until a third of the saturation conductance is reached in the measured channel. In step four the 'cutoff' voltage of the barrier gates is determined by lowering the voltages of the barrier gates to the point where the conducting channels cut off and the measured current is below the noise level. Slightly above this cutoff voltage, we sweep both barrier gates against each other, performing a $100 \times 100 \text{ mV}^2$ sweep towards lesser voltages. For each potential configuration in gatespace we record the current through the channel to see whether Coulomb oscillations of a quantum dot are present for this voltage configuration. If not we now lower the barrier gates by another 100 mV and re-accumulate the conducting channel with the top gate again up

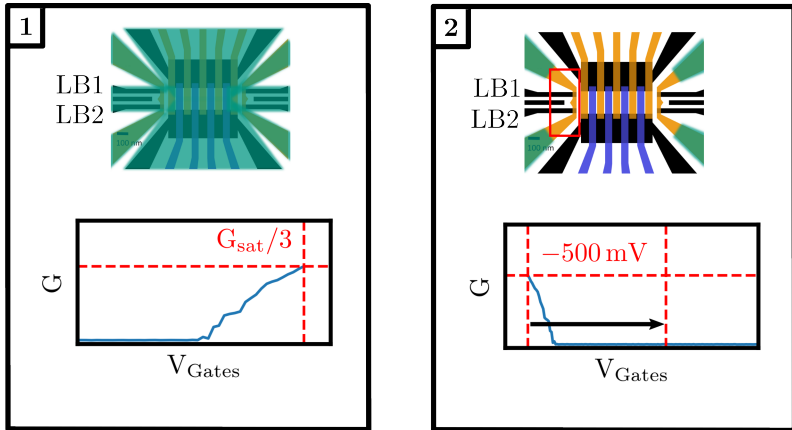


Figure 5.5.: Visualization of the first two steps of the tuning workflow. Artistic rendering of the electron distribution in the top half of the pictures and conductance versus gate voltage in the bottom part of the pictures.

to a conductance of $G_{sat}/3$, which was chosen as the reference conductance of the channel. Steps three and four are repeated iteratively until the 2D barrier sweep exhibits Coulomb oscillations typical of a SET, see step five, Fig. 5.7. Once the oscillation is found the barrier gates are tuned to the first Coulomb peak (step six), and a plunger trace is recorded to verify the success of the tuning routine.

5.5. Noise measurements

Once the quantum dot is formed, we characterize its noise in the frequency band of $200\ \mu\text{Hz}$ to $10\ \text{mHz}$ using peak tracking [63]. To extract the exact peak position in each sample we fit each individual trace with a secant function and define the position of its maximum as the current peak position. An example of such a fit is shown in Fig.5.8. Even in the presence of imperfections due to non-ideal device tuning, the fitting routine pinpoints the correct peak position with utmost accuracy. Furthermore, the fitting routine allows us to extract peak positions with a finer resolution than that of our Digital to analog converter (DAC), since the peak positions do not have to lie on a DAC sample but are extrapolated based on the positions of the measured samples. For this

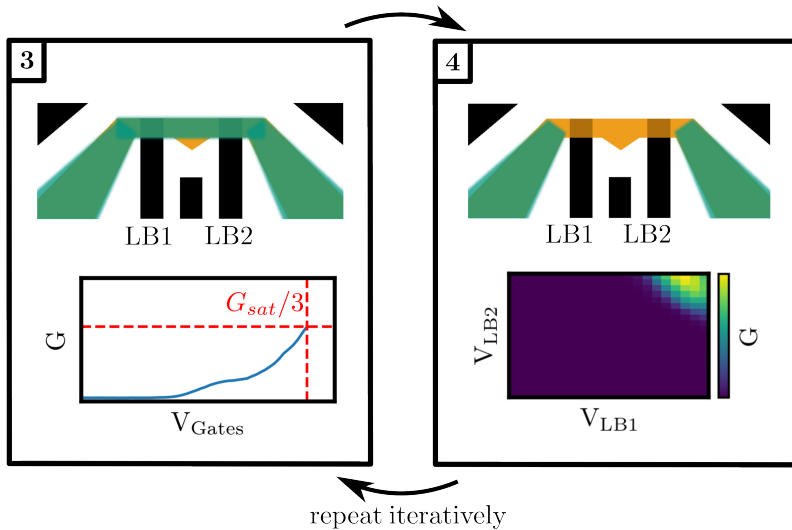


Figure 5.6.: Visualization of the the third and the fourth step of the tuning workflow. Steps three and four are repeated iteratively until a quantum dot is formed in between the barrier gates.

we continuously sweep the voltage of the plunger gate V_{SET} and record the current response of the SET device. A typical peak tracking result can be found in Fig. 5.9 (a). The shown result was recorded on sample A, after being cooled down at a bias cooling voltage of $0\text{ V }V_{BC}$. After tracking the peak, we perform a Welch-estimation of the power spectral density, using a Hann-type windowing function to exclude artificial frequencies arising from the drift in between the first and the last sample. The result can be found in Fig. 5.9 (b). The resulting spectrum follows a $1/f^\alpha$ -noise distribution, with $\alpha = 1.57$. This lies in between $\alpha \approx 1$, which suggests the presence of many two level fluctuators [83], and $\alpha \approx 2$, suggesting the presence of random walk noise [3] or an ensemble of slowly switching two level fluctuators [101]. Strong coupling to a single TLF with low switching frequency can be excluded, since the position of the Coulomb peak does not switch in between two distinct levels. To verify the noise reduction per bias-cooling each measured trace has to be reduced to one singular point in order to easily compare the performance in between cooldowns. To do this we integrate the fitted spectra in the band of $200\text{ }\mu\text{Hz}$ to 10 mHz , highlighted in purple in figure 5.9 (b). The integration is performed on the fit because of the varying duration of the peak tracking

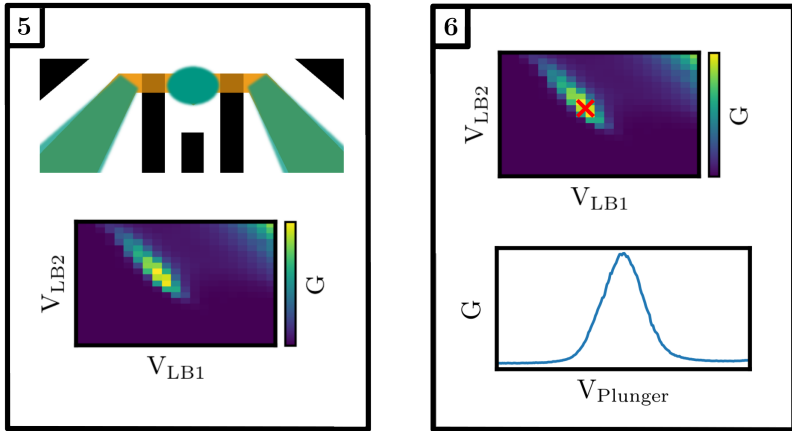


Figure 5.7.: Final steps of the tuning workflow. The dot position is highlighted by the green oval in between the barrier gates. After the detection of a quantum dot, a plunger sweep using the central electrode is performed to verify its existence.

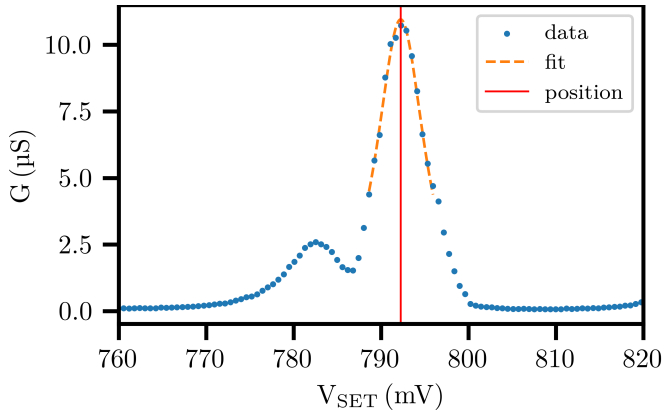


Figure 5.8.: Single plunger trace of a formed SET, together with a secans fit and the extracted peak position.

measurements. In some measurements the peak drifted out of the monitored plunger gate range after four hours. This in turn means that we would loose datapoints in the low frequency range. To not constrict the low frequency end

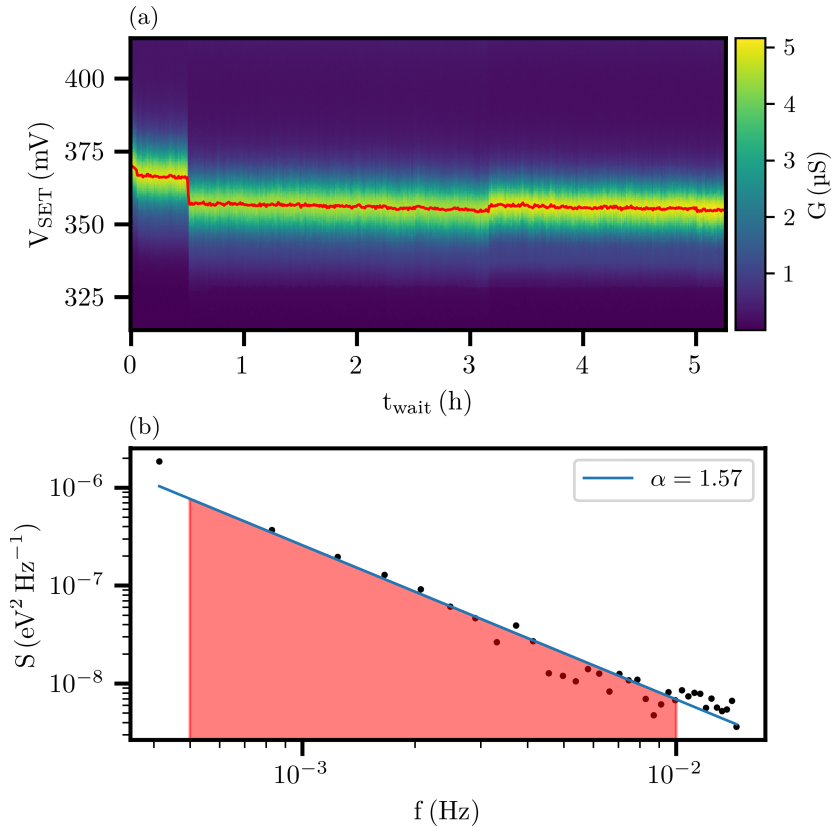


Figure 5.9.: a) Peaktracking measurement, performed on sample A with a V_{BC} of 0 V. The colorplot consists of Coulomb peak traces recorded back to back over the timescale of five hours. The red line indicates the position of the peak maxima along the V_{SET} -axis. b) Noise power spectral density calculated with the welch method, fitted with a β/f^α power spectrum. To calculate the total noise power the spectrum is integrated over the highlighted frequency range.

of the analysis by those unlucky incidences, we allow ourselves to extrapolate into the low frequency range by integrating the fit.

In the next step, the procedure is applied to all measured peaktracks. To ensure maximum comparability, we always track the first visible Coulomb peak. A selection of peak tracks measured on sample B is found in figures 5.10, 5.11, 5.12. The resulting noise spectra are shown in Fig. 5.13. Peaktracks

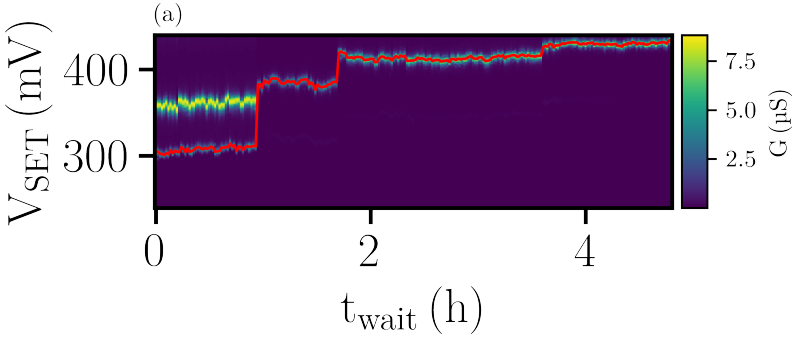


Figure 5.10.: Peaktracking measurement on Sample B for a bias cooling voltage of 0 V.

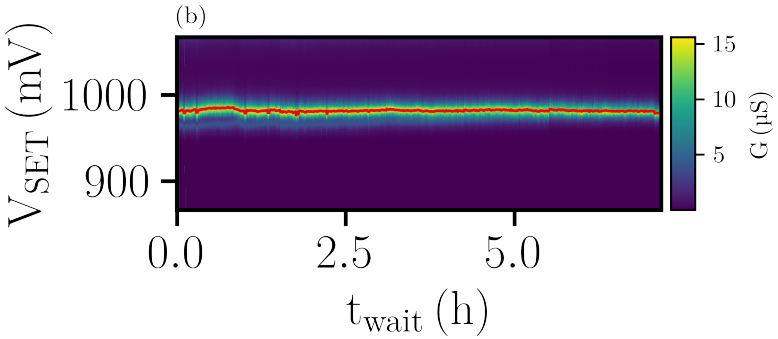


Figure 5.11.: Peaktracking measurement on Sample B for a bias cooling voltage of 0.625 V.

dominated by large jumps tend towards $\alpha \approx 2$, while peaktracks which are dominated by jitter around the original working point tend towards $\alpha \approx 1$, like 5.11. Samples dominated by jitter therefore see the background of many singular two-level fluctuators. On the other hand we have traces that are dominated by large jumps as the ones in Figs. 5.10 and 5.12. Here it is important to note that the magnitude of these jumps can be larger than a single Coulomb oscillation, as is seen in the first jump in Fig. 5.10. The SETs working point might be influenced by local charge redistributions in

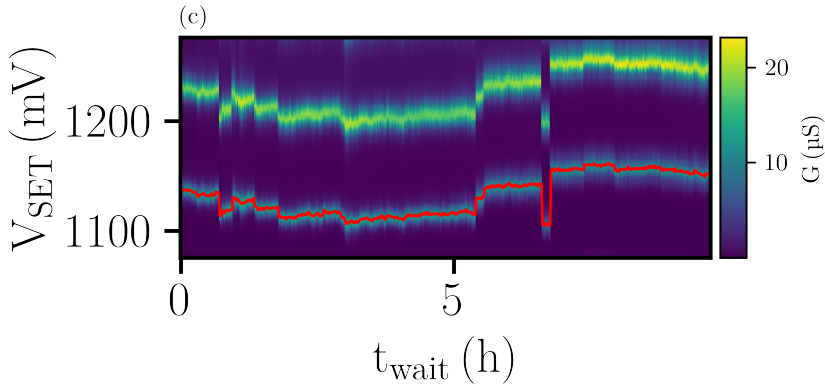


Figure 5.12.: Peaktracking measurement on Sample B for a bias cooling voltage of 0.825 V.

the silicon cap performing a random walk. This picture is feasible, since the cap-oxide interface itself is MOSFET-like. In Metal-oxide-semiconductor field-effect transistor (MOSFET) devices the lengthscales of localization length versus potential variation [116] are known to create a percolation induced metal to insulator transition [106]. To put it in more human terms, the 2DEG in the cap breaks down into charge carrier puddles consisting out of mobile carriers. If one of these puddles changes position to another minimum in the disorder potential of the cap, this might lead to a shift of the working point of the SET which is larger then one Coulomb oscillation, since the number of charge carriers taking part can be quite large.

The noise power versus bias cooling voltage can be seen in Fig. 5.14. First we convince ourselves of the feasibility of the presented noise metric. The recorded tracks in 5.10 (a) and 5.12 (c) are unstable and noise to the naked eye, and therefore should produce a high value in the noise metric. In comparison, 5.11 (b) shows a very stable working point, and should therefore produce a comparatively low value in the noise metric. Thankfully, this is exactly the behavior we extract for sample B. The arrows in 5.14 represent the position of the integrated total noise power for the discussed peak-tracking measurements (a) to (c). Out of four individually measured samples, four show a significant noise reduction at 0.7 V. It is interesting to note that the accumulation onset in the simulation, where the channel becomes significantly populated with charge carriers, occurs at ≈ 0.7 V, see Fig. 5.4. One sample even shows an improvement in the integrated noise power by two orders

of magnitude. Towards more positive voltages samples B and C show an increase in noise again, which may be related to the fact that at these higher voltages the samples get closer to the inversion point, at which the 2DEG is accumulated without the application of an accumulation voltage. Therefore the trapped charges mimic the role of a dopant. For more negative voltages the noise is increasing in the case of sample A and D and not showing a global trend.

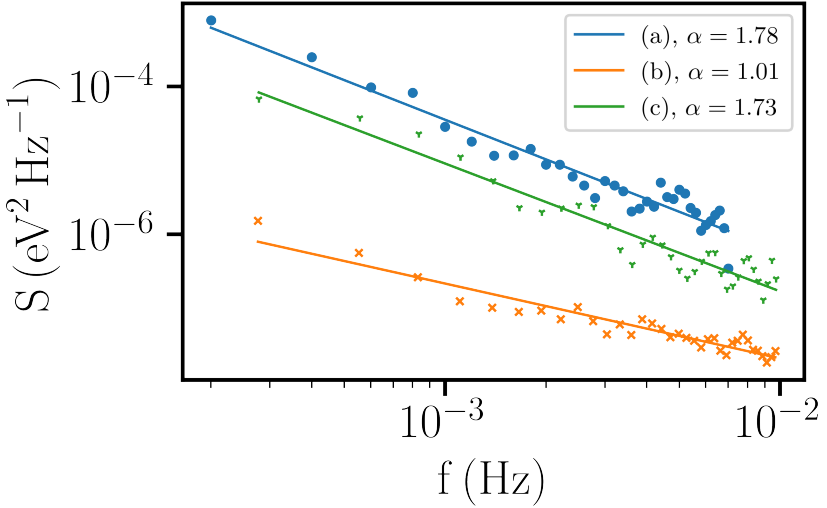


Figure 5.13.: Noise power spectral densities calculated with the Welch method, fitted with a β/f^α power spectrum. The resulting exponent is denoted in the legend.

In principle, we identified two types of noise on the SET peak position. A high frequency, low amplitude jitter and rarely occurring jumps with a large amplitude. These jumps occur on the timescale of hours and drastically affect the noise performance as can be seen in Fig. 5.14. This poses a great challenge because the individual peak-tracking measurements have a length of five hours, and may therefore not lead to statistically significant data. To dispel these doubts, we performed a measurement campaign consisting of 22 cooldowns of sample A. In total we measured eleven times the zero-bias followed by the 0.7 V bias case.

The results of the noise power integrated from 200 μHz to 10 mHz are shown in the Fig. 5.15. The orange line shows the median. The value box is defined

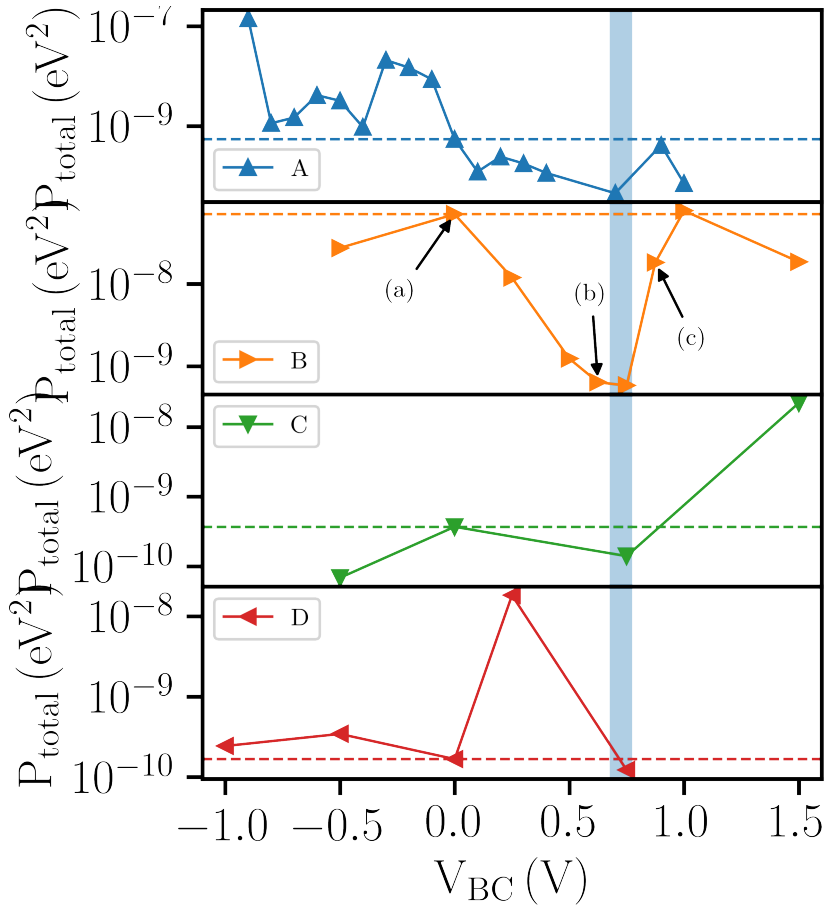


Figure 5.14.: Integrated noise power versus bias cooling voltage. While the noise increases towards negative biases for most samples, a local minimum around the value of 0.7 V can be found in each trace. The dashed line acts as a guide for the eye and shows the noise level of a sample bias cooled with zero volts.

as such, that it includes 50 % of all conducted measurements for the respective bias cooling voltage. The median value of the 0.7 V measurements is reduced by a factor of 6.16 in comparison to the median of the 0 V measurements. To determine the statistical significance of the measured results, we performed an unpaired t-test comparing the measured data sets for 0 and 0.7 V, which

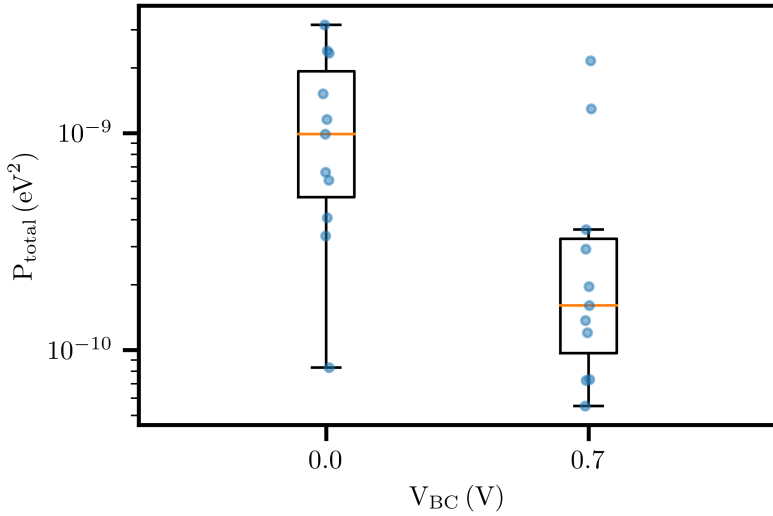


Figure 5.15.: Boxplot of the integrated noise power of 22 peak trackings, 11 performed each for the bias cooling voltages 0 V and 0.7 V. The orange solid line displays the median value, the green dashed line the mean value and the box includes 50 % of all data points centered around the median, called the interquartile range (IQR). The whiskers represent scores outside the middle 50 % and are capped at $1.5 \cdot \text{IQR}$. While the mean is dominated by the outliers and even falls outside the box of the 0.7 V dataset, it can be seen that the median value of 0.7 V is almost an order of magnitude lower than for 0 V.

results in a probability of 0.143 % that the two measured sample means follow the same normal distribution.

5.6. Schrödinger-Poisson Simulations

To verify the feasibility of our theory that the charge redistributions in the silicon cap are causing the shifts in the measured working point, we performed three-dimensional Schrödinger-Poisson simulations of the SET-area using the software package next-nano, to calculate possible tunneling currents for different bias-cooling conditions. The cap is still below its metal-to-insulator transition density in the performed measurements, which can be deduced by

the fact that an increase in accumulation gate potential still yielded a change in the potential landscape, i.e. the shaping and formation of dots was possible. If the cap was already accumulated, it would screen the potential of the gates shaping the 2DEG and no further changes would be visible. Therefore, the charges rearranging in the cap have to stem from a different location. SiGe heterostacks are known [45, 71] to induce parasitic tunneling currents from the Si channel into the cap, depending on the gate voltage configuration.

Bias cooling itself was included in the simulation by placing interface charges in the interstitial silicon oxide layer under the metal gates. The density of the interface charges under each gate-layer was calculated based on the shift of the accumulation voltage. Depending on the gate layer, a different charge density was used. The charge densities were extracted by finding the intersection in between the measured shift in accumulation voltage and the the simulated shift in accumulation voltage per interface charge 5.4. In the case of overlapping gate layers the lowest lying gate layer was used to calculate the density, since it screens the gates lying above. The working points of the 0 V and 0.7 V volt measurements performed on sample A were reproduced in simulation. The zero volt cooldowns displayed an average accumulation voltage of (0.73 ± 0.03) V, and the 0.7 V cooldowns an average accumulation voltage of (1.041 ± 0.007) V, corresponding to a shift in accumulation voltage of 0.31 V. Based on these voltages the calculated interface charge density for gate layer one (with $t_{\text{Al}_2\text{O}_3} = 10$ nm) is 1.628×10^{12} cm⁻² defects with a charge of $1 e$, for gate layer two (with $t_{\text{Al}_2\text{O}_3} = 20$ nm) is 7.49×10^{11} cm⁻² and 4.8×10^{11} cm⁻² for gate layer three (with $t_{\text{Al}_2\text{O}_3} = 30$ nm). The spread in the accumulation voltage does not vary much in between different cooldowns with same bias cooling voltage (as can be seen by the small statistical error of the mean), therefore the same interface charge densities are used for each 0.7 V cooldown. For 0 V cooldowns an interface charge density of zero is assumed.

To reflect the device's real working point, the gate-voltage configurations used in the actual measurement as well as calculated interface charge densities have been reproduced in simulation, whereby treating the spacially large regions (substrate, buffer and spacer) in Thomas-Fermi [105, 22, 104] approximation, and the smaller regions, where quantum confinement plays a major role (channel and cap) with a self-consistent Schrödinger-Poisson approach in $\mathbf{k} \cdot \mathbf{p}$ approximation [51, 107]. Electron densities inside the silicon channel exceed their expected equilibrium value [43]. The maximum density should, and in fact at a later stage also is, be limited by the accumulation of the cap

layer. Free charge carriers inside the upper cap layer screen the electric field of inducing gates in the channel and an increase in accumulation gate voltage at this stage just leads to an increase in charge density in the cap. Nevertheless, the high critical charge density of the cap for the metal-to-insulator transition prevents the formation of macroscopic mobile charge carrier paths. Only at a later stage, when enough electrons are present in the cap layer and the critical density is surpassed the expected reduction in charge carrier density in the channel takes place. Since the cap is not conductive on the macroscopic scale, charge carriers cannot stem from the ohmic contacts. Therefore the cap is out of equilibrium with the ohmic contacts. As stated above, the main suspect for the source of the carriers which bring the cap above its transition is tunneling from the channel into the cap. Before that process is completed, the cap is out of equilibrium with the contacts. To reflect this in simulation, the electrochemical potential of the cap is pinned to -0.7 eV, so that it does not accumulate. This value was chosen arbitrarily, and does not influence the simulation further. With these boundary conditions set in place, the simulation was finally carried out.

The resulting electric field, band-edge and electrochemical potential are used to calculate the Fowler-Nordheim-tunneling rates [24] from the SiGe-channel into the silicon cap using the analytical expression 2.17 [111]. A selection of simulation results are shown in Figs. 5.16 for the zero volt case and 5.17 for the 0.7 V bias-cooled case. First, let us compare the electric fields inside the spacer, $\mathcal{E}_{\text{SiGe}}$. Depicted in Figs. 5.16 (a) and 5.17 (a) is the electric field in the spacer layer 7.5 nm above the quantum well in growth direction. An obvious difference in between the two is the reduced field amplitude. In the zero volt case, the electric field maximum is above 10 MV m^{-1} , while staying below 7 MV m^{-1} in the 0.7 V case. Furthermore the electric field is more homogeneous and does not form "hot spots" in the bias cooled case. This is also reflected directly in the computed tunneling currents in Figs. 5.16 (b) and 5.17 (b). Again, the overall tunneling current is larger in the zero volt case, but this time by six orders of magnitude. Furthermore, the dot region itself acts as a source of electrons. Tunneling from the contacts is also much stronger in the case of the conventionally cooled sample. In the 0.7 V case, the tunneling is not only weaker, but also localized further away from the dot region. For this set of cooldowns the noise reduction directly correlates with reduced tunneling into the cap. In a next step the simulation was repeated for each working point depicted in Fig. 5.15, to verify an interrelation in between tunneling events and noise. In Fig. 5.18 the

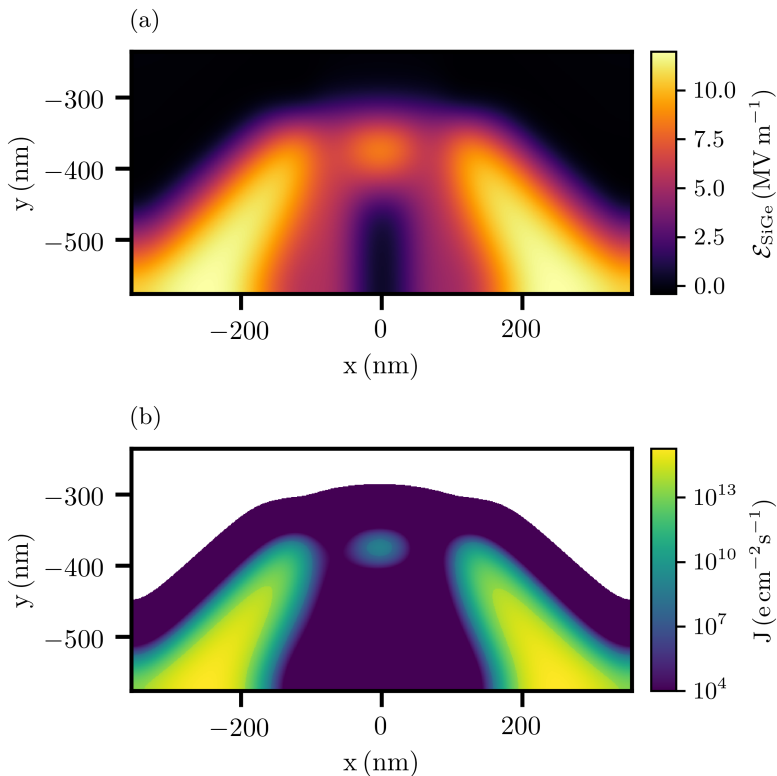


Figure 5.16.: (a) Simulated electric field inside the spacer layer for 0 V bias cooling (b) tunneling current from the silicon channel into the silicon cap for 0 V bias cooling.

calculated maximum tunneling current was graphed against the measured noise power. The 0 V cases show a bunching towards high tunneling rates and high noise, whereas the 0.7 V cases show a bunching towards low tunneling and low noise. The datapoint belonging to the 0.7 V dataset in the high-tunneling, high-noise quadrant showed an excess of telegraph noise, which was only recorded in this one specific cooldown. Nevertheless it confirms the relation in between high tunneling and high noise. Furthermore, the 0 V outlier present in the high-noise low-tunneling quadrant is revealed by simulation to be the working point of a double dot. In that case, the

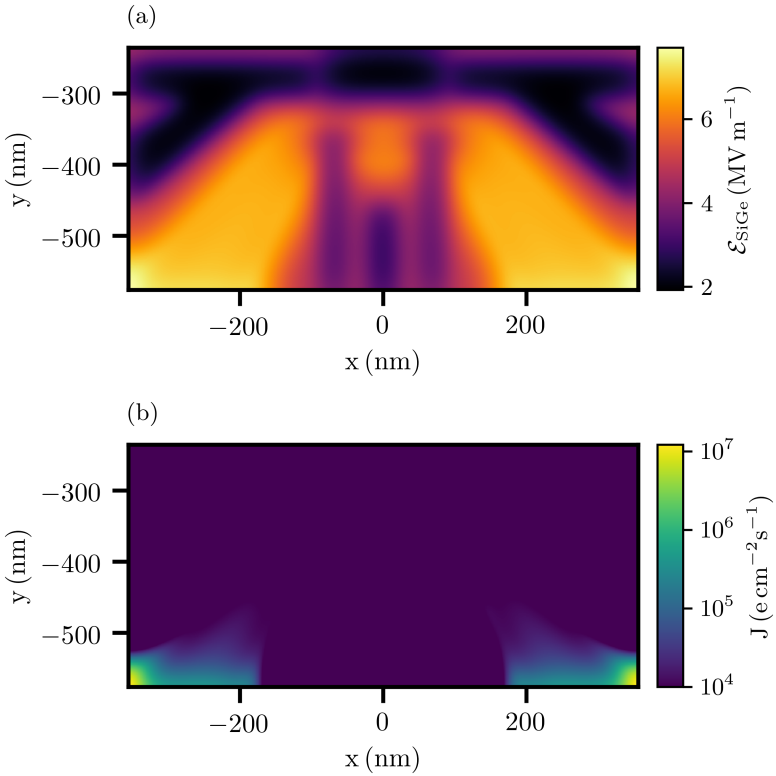


Figure 5.17.: (a) Simulated electric field inside the spacer layer for 0.7 V bias cooling (b) tunneling current from the silicon channel into the silicon cap for 0.7 V bias cooling

barriers are at a comparatively high potential and the accumulation gate is only biased with 0.99 V. Therefore forming dots underneath the individual barrier gates instead of in between them. The noise therefore most likely stems from the proximity to the reservoirs and the comparatively high tunnel coupling. Although outliers exist, our model suggests that there is a correlation in between high noise and high tunneling. To better understand the effect of bias cooling and why it reduces the tunneling current, I wish to draw the readers attention towards Figs. 5.20 and 5.21, which depict a horizontal slice taken through the dot region, along the red line shown in 5.19. In the

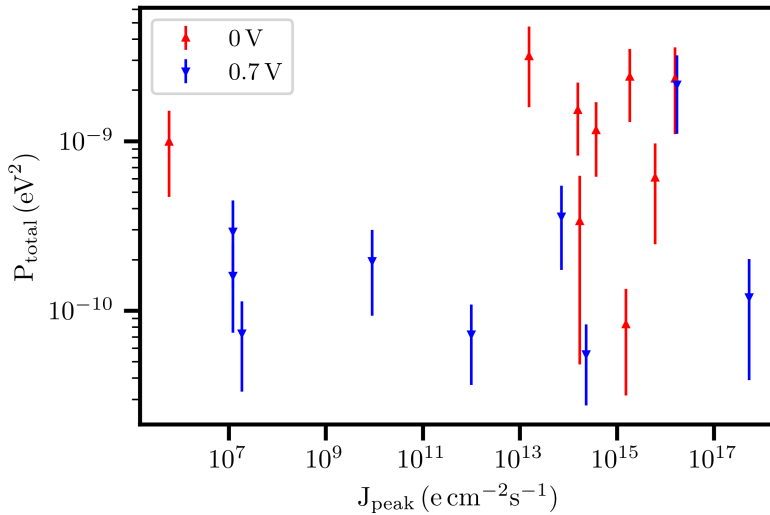


Figure 5.18.: Measured integrated noise density plotted against simulated peak tunneling current. Each point represents a bias cooldown performed on sample A.

bottommost area the conduction band edge is depicted, in the central area the assumed interface charge density, and on the top a schematic rendering of the gate oxide and the gates themselves. The conduction band edge in the dot area is fitted with the sum of two Lorentzians which are drawn separately, with their respective Full width half maximum (FWHM) indicated. The minimum in between the two Lorentzians is the quantum dot itself, and the peaks form the tunneling barriers to the reservoir. The barriers in the zero volt case have a FWHM of ≈ 127 nm, whereas in the bias cooled case the width is reduced to ≈ 97.5 nm. Furthermore the minimum of the dot is more pronounced. So, why is the FWHM reduced, even though the applied voltages are higher? The reason lies within the pointspread of the gate potential. In the zero volt case, the barriers are nominally 52 nm away from the channel, meaning that their projection in the plane of the 2DEG is quite wide. In the bias cooled case on the other hand, the effective gate potential is a sum of the interface charge and the actual gate potential. The interface charge is located 19 nm closer to the channel, therefore its projection in the 2DEG plane is finer. The interface charge is of negative nature, therefore it raises the conduction

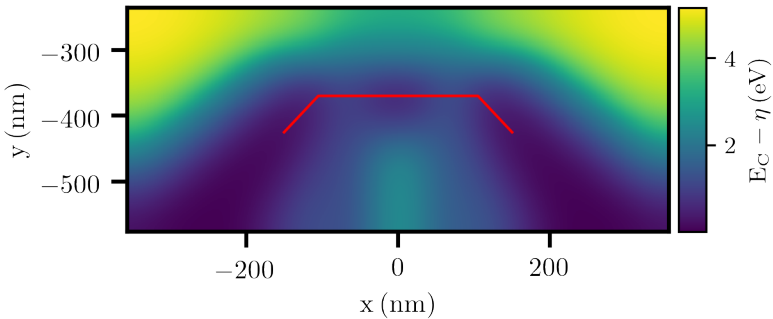


Figure 5.19.: Simulated conduction band edge for a working point in which the device was cooled with a bias of 0V. 1D slices of the conduction band edge shown in later figures are taken along the red line.

band edge, just as the Barrier would do. The combined potential of applied gate voltage and interface charge therefore results in an effectively more localized, thinner barrier with less cross-capacitance to the dot area. It is exactly because of this reason that the tunneling is reduced. In the zero volt case the cross-capacitance of the barriers to the dot area is larger, and therefore the inducing gate potential has to be higher in comparison to the barriers, because it has to effectively overpower the cross capacitance of the barriers. Higher inducing gate voltages can be translated directly into higher field in the sample and therefore higher tunneling. In the bias cooled case the cross capacitance is reduced, and therefore the inducing gate voltage can be lower in comparison to the barrier voltages. A weak link in this explanation is its reliance on the form of the interface charge density. To calculate the tunneling currents we assumed the interface charges to be only present directly under the gates, basically projecting the gates down directly at the Si/Al₂O₃ interface. This assumption is plausible, since the population of the acceptor states is done by tunneling, as well as thermal activation, both processes that decay exponentially in energy. Therefore it is reasonable to assume that charges are only accumulated directly under the gates, where the electric field is stronger, and the population decays exponentially as a function of the internal field.

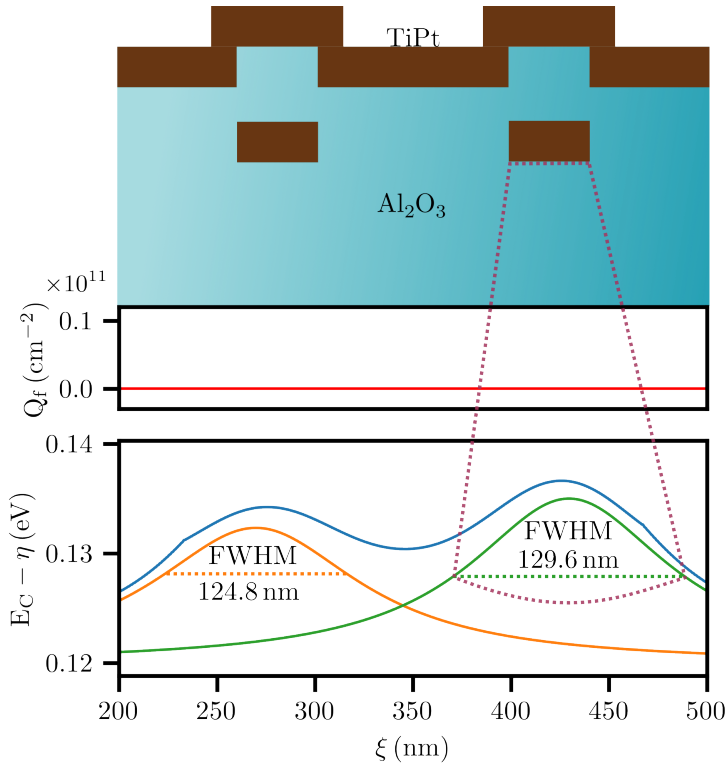


Figure 5.20.: Rendering of the conduction band edge, interface charge and gate layout for a 0V cooldown. η is the electrochemical potential of the conducting channel.

5.7. Summary

We found that bias cooling of undoped $\text{Si}/\text{Si}_{1-x}\text{Ge}_x$ heterostacks causes charges to be trapped in between the silicon channel and the metal gates, shifting the turn-on voltages linearly with the applied cooldown bias in the range of -1 V to 1 V . Consequently we investigated the low frequency charge noise of bias cooled devices. For this we used peak tracking measurements. To quantify the noise for each bias cooling voltage we computed the total noise power for every peak track, which has a minimum around 0.7 V bias cooling voltage. In samples A and B, a global minimum is visible in the

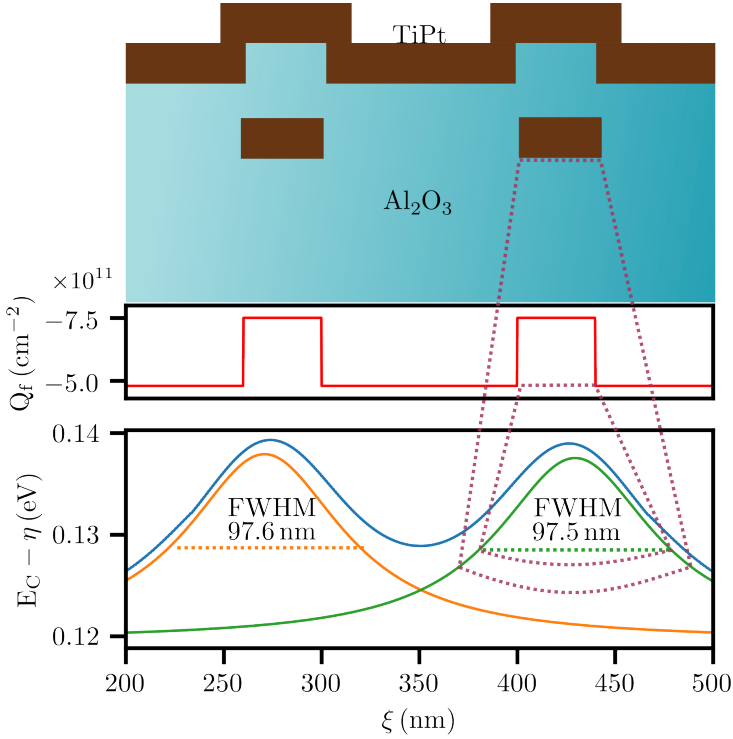


Figure 5.21.: Rendering of the conduction band edge, interface charge and gate layout for a 0.7 V cooldown.

total noise power. Samples C and D show a local minimum around 0.7 V. In one case we could reduce the total noise power by a factor of 120.73 during the 0.75 V cooldown in comparison to the 0 V cooldown, and during a 22-cooldown campaign the measured median of the total noise power was reduced by a factor of 6.16 for the 0.7 V cooldown in comparison to the 0 V value. In addition, we present the results of a three-dimensional Schrödinger-Poisson simulation, based on measured working points. Here we find that samples cooled down with a 0.7 V bias show a by seven orders of magnitude reduced tunneling current from the channel into the cap. While the direct proof remains elusive, we correlate the simulation to our noise measurement results, and find a bunching of datapoints in the high noise, high

tunneling as well as in the low noise, low tunneling quadrants. Furthermore, the simulations indicate that a homogeneous field distribution along the accumulation gate improves the device's performance, meaning that designs where the dot region is accumulated by a separate plunger are favorable. As a next step, the simulations could be rounded by simulating the population process of the acceptor states during cooldown, and therefore providing a better approximation of the true spatial distribution of the interface charges. Experimentally, bias cooling could be extended to qubit samples, investigating the effect on coherence. Furthermore, tunneling in the cap has been identified as one of the root causes of the quantum dot instabilities.

6. Conclusion and Outlook

Stabilizing the working point of SiGe devices using simple means is a tangible scientific goal which has been achieved in the course of this thesis. The road towards the presented findings was, as is often the case in science, paved with many obstacles and challenges to overcome.

The first challenge was to develop and build a cryogenic infrastructure able to support the research with semiconductor quantum circuits, while also providing a fast turnaround cooldown process. With cycle times from mK to Room temperature (RT) and back to mK in the 2 h range, I succeeded in meeting the set design goals.

Indispensable for any setup is the control software providing the handling of measurement data, device control and synchronization. Over the course of this thesis I wrote three extensive software packages which are device-agnostic, and able to run complex measurement schemes with interleaved control and measurement intervals in real-time.

Furthermore, the amount of data created by a systematic noise study as the one presented in this thesis was very large. One of our main accomplishments, which is the tracking of the integrated noise density versus the bias cooling voltage (presented in Fig. 5.14) consists out of 35 noise measurements. Each of these measurements includes a full thermal cycle from mK to RT to mK, the tuning of the working point of the device and a 5 h peak tracking. In order to obtain comparable noise results in face of a changed electrostatic environment, a SET-tuning routine which converges to the lowest possible topgate voltage was devised and implemented.

Our main findings include the identification of a minimum in integrated noise density at 0.7 V bias cooling voltage. The minimum was reproduced in four out of four samples. The noise reduction at 0.7 V is supported by a statistical analysis of 22 cooldowns in which we alternated the bias cooling voltages in between 0 V and 0.7 V, and found a noise reduction by a factor of

6.16. The probability of both point groups (0 and 0.7 V) stemming from the same normal distribution is 0.143 %, which is regarded as statistically proven. Additionally, we support our claims by simulating the tunneling currents for each of the working points measured in this 22 cooldown campaign, and correlate the reduction in noise with a reduction in parasitic electron tunneling from the Si-channel through the spacer into the Si-cap. Our findings can be directly applied to existing samples, in order to improve their stability. Furthermore they give implications for essential improvements to sample design by minimizing these parasitic currents.

Extensions to the current study would be to simulate the population process of the charge traps providing a better representation of the interface charge density when cooled down with a bias. The study could be extended to more involved designs including qubits, to measure the impact of reduced tunneling currents on coherence times and long-term sample stability.

Acknowledgements

So I guess this is it. At the moment of typing these words I still cannot fully comprehend that this long and adventurous journey filled with ups and downs is coming to an end, even though reality is trying to convince me otherwise. In the spirit of the "real treasure was the friends we made along the way" I do have to absolutely thank a few people, because they deserve it.

First, I would like to thank Wolfgang for being a down-to-earth, approachable and extremely competent supervisor. He made me feel welcome, from the very beginning. The memory of the day he convinced me to do a PhD Thesis in his group still puts a smile on my face four (oh god) years later. It was a late friday evening, and Dennis and I were finishing the measurements for my master thesis. Wolfgang approached us, a little electrical buzz-saw in hand, turned it on and pointed it in our direction. "Finish your masters, and start working for me. Or otherwise..." After finally joining his Team I was truly impressed by the amount of practical knowledge one single person can acquire, and I feel privileged that I was able to siphon some of it over the course of my PhD. From the time we spent together working on the Sionludi-XL, over all the little details that are to be taken into account once something has to go cryogenic to very general life lessons, two of which will most likely stay with me forever. The first one being: A small change in relation to a large quantity will in most cases not have a great impact, and the other being: If one wishes to be unhappy, one can always be so. His fearless "Let's have a go at it" attitude left a deep impression on me, and I hope to carry on this spirit in the world beyond the PhD. So all in one sentence: Wolfgang, I am truly thankful for giving me the opportunity to do a PhD in your group, your support and your fearless down-to-earth outlook on science and the world in general.

I wish to express my gratitude towards Hendrik Bluhm, my co-referee. His group provided us with the samples we measured, and he with the guidance we needed on our last scientific journey. Without him the Bias-Cooling project

would never have come to a satisfying conclusion. Although roughly 400 km separate Karlsruhe from Aachen, he was always available for a scientific discussion. Fondly I remember one day when I tried to schedule a zoom meeting with him, but he was not convinced one was needed, so he replied with emails. We ended up exchanging emails until 22:00 o'clock.

Also, Lars Schreiber is more than deserving of a big thank you. He guided us through the scientific process from the conceptualization phase until the final proof reading of our Paper.

And finally, I wish to thank my beloved Team Semiconductor. Thomas, Sonja, Daniel, Viktor and Noah. You guys are just plainly amazing. From the moment we started working on this new scientific subject in Karlsruhe, to the day of writing this, we have achieved so much. Starting with one setup and way too many ideas what to do with it, to four individual subjects. Ranging from materials and device physics to completely insane semiconductor-molecule hybrid devices.

Anyways, Thomas I wish to thank you for sharing an office these last years. I will try, though I am sadly pretty sure I will never find another office mate that has your specific blend of confident competence and sheer undiluted lunacy. After the countless discussions we had, I think that by now a specific part of my brain has evolved to formulate opinions and questions to throw in your general direction. Once I asked you with whom you're spending more time, with me or your girlfriend. To this day I have to grin about how long it took you to think the answer over.

Sonja, you always were the 4.18 sunshine. From your Bachelors thesis with Michi to your Masters with us, I am happy that we shared our journeys. Be it in uni, Croatia or Cyprus. Two things you impressed me with: Your no-nonsense efficient thinking and your perfect sense for situational comedy. You master the timing of saying things no one would expect from you at a moment where everyones guard is down.

Daniel and Viktor, the molecule-men. Though I never said it before, and never will again, I am impressed of your willingness to jump into the unknown, and walk the terra incognita in the way you guys do. I wish you best of luck on your endeavours. Daniel, I thank you for bringing much needed organization and routines into our Team. Viktor, I do truly appreciate your out-of-the box thinking style, and thank you for the all the mornings we spent together in the gym.

I wish to thank my friends with whom I did not work on a daily basis. Michi, you are my best buddy, thanks for always having my back and being up for random shenanigans be it on the archery range or on Hoxxes V. Richard this is something one cannot say to many people, but with you it's true: Thanks for being you, and all the times we could do a "Bauernhofurlaub" at your place. Thiemo, the quiet engineer. Don't let him near a fire. Or do and enjoy the show. Melanie, thank you for the nights in which we discussed our lives or danced away the hours in Karlsruhe's underground. Viki, thank you for being as cheerful as you are, and the great holidays we spent together at the Breeze or in Wacken. Luca, it's time for more bass. Heiko, we are very different people with a shared passion for long discussions about obscure topics. Suse, your costume parties are the bomb. Lachlan, we need another bottle of Buckfast. And to everyone I didn't mention by name: You are amazing.

Finally I wish to thank my Family, especially Larinchen, Lady and my mum. Larinchen and Lady, you are the best fluffy dog-like creatures I have encountered. And mum, if you managed to read to this point without rolling your eyes too hard, I thank you for your unending support, your willingness to put up with me, your sharp outlook and for generally putting me on this planet.

Thank you.

Bibliography

- [1] P. W. Anderson. “Absence of Diffusion in Certain Random Lattices”. In: *PR* 109.5 (Mar. 1958), pp. 1492–1505. DOI: 10.1103/PhysRev.109.1492. URL: <https://link.aps.org/doi/10.1103/PhysRev.109.1492>.
- [2] T. Ando and H. Akera. “Connection of envelope functions at semiconductor heterointerfaces. II. Mixings of Γ and X valleys in GaAs/Al_xGa_{1-x}As”. In: *Phys. Rev. B* 40 (17 Dec. 1989), pp. 11619–11633. DOI: 10.1103/PhysRevB.40.11619.
- [3] J. A. Barnes and D. W. Allan. “A statistical model of flicker noise”. In: *Proceedings of the IEEE* 54.2 (1966), pp. 176–178. ISSN: 1558-2256. DOI: 10.1109/PROC.1966.4630.
- [4] C. Barthel et al. “Fast sensing of double-dot charge arrangement and spin state with a radio-frequency sensor quantum dot”. In: *PRB* 81.16 (Apr. 2010), p. 161308. DOI: 10.1103/PhysRevB.81.161308. URL: <https://link.aps.org/doi/10.1103/PhysRevB.81.161308>.
- [5] Bela Bauer et al. “Quantum Algorithms for Quantum Chemistry and Quantum Materials Science”. In: *Chem. Rev.* 120.22 (Nov. 2020), pp. 12685–12717. ISSN: 0009-2665. DOI: 10.1021/acs.chemrev.9b00829. URL: <https://doi.org/10.1021/acs.chemrev.9b00829>.
- [6] Felix Bloch. “Über die Quantenmechanik der Elektronen in Kristallgittern”. In: *Zeitschrift für Physik* 52.7 (1929), pp. 555–600. ISSN: 0044-3328. DOI: 10.1007/BF01339455. URL: <https://doi.org/10.1007/BF01339455>.
- [7] M. G. Borselli et al. “Measurement of valley splitting in high-symmetry Si/SiGe quantum dots”. In: *Appl. Phys. Lett.* 98.12 (Mar. 2011), p. 123118. ISSN: 0003-6951. DOI: 10.1063/1.3569717. URL: <https://doi.org/10.1063/1.3569717>.

- [8] B. Brar, G. D. Wilk, and A. C. Seabaugh. “Direct extraction of the electron tunneling effective mass in ultrathin SiO₂”. In: *Appl. Phys. Lett.* 69.18 (Oct. 1996), pp. 2728–2730. ISSN: 0003-6951. DOI: 10.1063/1.117692.
- [9] Léon Brillouin. “La mécanique ondulatoire de Schrödinger; une méthode générale de résolution par approximations successives”. In: *Compt. Rend. Hebd. Seances Acad. Sci.* 183.1 (1926), pp. 24–26.
- [10] R. E. Burgess, H. Kroemer, and J. M. Houston. “Corrected Values of Fowler-Nordheim Field Emission Functions $v(y)$ and $s(y)$ ”. In: *Phys. Rev.* 90 (4 May 1953), pp. 515–515. DOI: 10.1103/PhysRev.90.515.
- [11] Elliot J. Connors et al. “Charge-noise spectroscopy of Si/SiGe quantum dots via dynamically-decoupled exchange oscillations”. In: *Nature Communications* 13.1 (2022), p. 940. ISSN: 2041-1723. DOI: 10.1038/s41467-022-28519-x.
- [12] A. Crippa et al. “Gate-reflectometry dispersive readout and coherent control of a spin qubit in silicon”. In: *Nature Communications* 10.1 (2019), p. 2776. ISSN: 2041-1723. DOI: 10.1038/s41467-019-10848-z. URL: <https://doi.org/10.1038/s41467-019-10848-z>.
- [13] J. P. G. van Dijk et al. “Impact of Classical Control Electronics on Qubit Fidelity”. In: *PRAPPLIED* 12.4 (Oct. 2019), p. 044054. DOI: 10.1103/PhysRevApplied.12.044054. URL: <https://link.aps.org/doi/10.1103/PhysRevApplied.12.044054>.
- [14] C.B. Duke. *Tunneling in Solids*. Solid state physics : advances in research and applications: Supplement. Academic Press, 1969. ISBN: 9780126077704. URL: <https://books.google.de/books?id=T1XwAAAAMAAJ>.
- [15] A. Elsayed et al. “Comprehensive 300 nm process for Silicon spin qubits with modular integration”. In: *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. 2023, pp. 1–2. DOI: 10.23919/VLSITechnologyandCir57934.2023.10185272.
- [16] Asser Elsayed et al. “Low charge noise quantum dots with industrial CMOS manufacturing”. In: (Dec. 2022). arXiv: 2212.06464 [cond-mat.mes-hall].
- [17] J. M. Elzerman et al. “Single-shot read-out of an individual electron spin in a quantum dot”. In: *Nature* 430.6998 (2004), pp. 431–435. ISSN: 1476-4687. DOI: 10.1038/nature02693. URL: <https://doi.org/10.1038/nature02693>.

- [18] K. Eng, R. N. McFarland, and B. E. Kane. “High mobility two-dimensional electron system on hydrogen-passivated silicon(111) surfaces”. In: *Applied Physics Letters* 87.5 (July 2005), p. 052106. ISSN: 0003-6951. DOI: 10.1063/1.2001734. eprint: https://pubs.aip.org/aip/apl/article-pdf/doi/10.1063/1.2001734/13138967/052106_1_online.pdf. URL: <https://doi.org/10.1063/1.2001734>.
- [19] K. Eng, R. N. McFarland, and B. E. Kane. “Integer Quantum Hall Effect on a Six-Valley Hydrogen-Passivated Silicon (111) Surface”. In: *Phys. Rev. Lett.* 99 (1 July 2007), p. 016801. DOI: 10.1103/PhysRevLett.99.016801. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.99.016801>.
- [20] L. Esaki and R. Tsu. “Superlattice and Negative Differential Conductivity in Semiconductors”. In: *IBM Journal of Research and Development* 14.1 (1970), pp. 61–65. DOI: 10.1147/rd.141.0061.
- [21] Peter Fejes. “Optimization of the Operation of the Sionludi Cryostats”. Bachelor’s Thesis. Karlsruhe Institute of Technology, 2020.
- [22] E. Fermi. “Eine statistische Methode zur Bestimmung einiger Eigenschaften des Atoms und ihre Anwendung auf die Theorie des periodischen Systems der Elemente”. In: *Zeitschrift für Physik* 48.1 (1928), pp. 73–79. ISSN: 0044-3328. DOI: 10.1007/BF01351576.
- [23] J. Ferrero et al. “Noise reduction by bias cooling in gated Si/Si_xGe_{1-x} quantum dots”. In: *in preparation* (2023).
- [24] Ralph Howard Fowler and L. Nordheim. “Electron emission in intense electric fields”. In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 119.781 (1928), pp. 173–181. DOI: 10.1098/rspa.1928.0091. eprint: <https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1928.0091>.
- [25] R. Franz and G. Wiedemann. “Ueber die Wärme-Leitungsfähigkeit der Metalle”. In: *Ann. Phys.* 165.8 (Jan. 1853), pp. 497–531. ISSN: 0003-3804. DOI: 10.1002/andp.18531650802. URL: <https://doi.org/10.1002/andp.18531650802>.
- [26] Kenichi Fujii et al. “Realization of the kilogram by the XRCD method”. In: *Metrologia* 53.5 (2016), A19. ISSN: 0026-1394. DOI: 10.1088/0026-1394/53/5/A19. URL: <https://dx.doi.org/10.1088/0026-1394/53/5/A19>.

- [27] Takafumi Fujita et al. “Coherent shuttle of electron-spin states”. In: *npj Quantum Information* 3.1 (2017), p. 22. ISSN: 2056-6387. DOI: 10.1038/s41534-017-0024-4. URL: <https://doi.org/10.1038/s41534-017-0024-4>.
- [28] T. A. Fulton and G. J. Dolan. “Observation of single-electron charging effects in small tunnel junctions”. In: *Phys. Rev. Lett.* 59 (1 July 1987), pp. 109–112. DOI: 10.1103/PhysRevLett.59.109.
- [29] John King Gamble et al. “Valley splitting of single-electron Si MOS quantum dots”. In: *Applied Physics Letters* 109.25 (Dec. 2016), p. 253101. ISSN: 0003-6951. DOI: 10.1063/1.4972514. eprint: https://pubs.aip.org/aip/apl/article-pdf/doi/10.1063/1.4972514/14490006/253101_1_online.pdf. URL: <https://doi.org/10.1063/1.4972514>.
- [30] Andreas Gehring. “Simulation of Tunneling in Semiconductor Devices”. PhD thesis. Technische Universität Wien, 2003.
- [31] J. J. H. Gielis et al. “Negative charge and charging dynamics in Al₂O₃ films on Si characterized by second-harmonic generation”. In: *Journal of Applied Physics* 104.7 (Oct. 2008), p. 073701. ISSN: 0021-8979. DOI: 10.1063/1.2985906.
- [32] C. Godfrin et al. “Operating Quantum States in Single Magnetic Molecules: Implementation of Grover’s Quantum Algorithm”. In: *Phys. Rev. Lett.* 119 (18 Nov. 2017), p. 187702. DOI: 10.1103/PhysRevLett.119.187702.
- [33] A. Gold. “Metal-insulator transition in Si/SiGe heterostructures: mobility, spin polarization and Dingle temperature”. In: *Semiconductor Science and Technology* 26.4 (2011), p. 045017. ISSN: 0268-1242. DOI: 10.1088/0268-1242/26/4/045017. URL: <https://dx.doi.org/10.1088/0268-1242/26/4/045017>.
- [34] A. Gold. “Mobility and metal-insulator transition of the two-dimensional electron gas in SiGe/Si/SiGe quantum wells”. In: *J. Appl. Phys.* 108.6 (Sept. 2010), p. 063710. ISSN: 0021-8979. DOI: 10.1063/1.3482058. URL: <https://doi.org/10.1063/1.3482058>.
- [35] Martin A. Green. “Intrinsic concentration, effective densities of states, and effective mass in silicon”. In: *J. Appl. Phys.* 67.6 (Mar. 1990), pp. 2944–2954. ISSN: 0021-8979. DOI: 10.1063/1.345414. URL: <https://doi.org/10.1063/1.345414>.

- [36] Giuseppe Grosso and Giuseppe Pastori Parravicini. *Solid State Physics*. San Diego, UNITED KINGDOM: Elsevier Science & Technology, 2013. URL: <http://ebookcentral.proquest.com/lib/karlsruhetech/detail.action?docID=1495641>.
- [37] Marius Grundmann and Marius Grundmann. “Band Structure”. In: *The Physics of Semiconductors: An Introduction Including Nanophysics and Applications*. Cham: Springer International Publishing, 2021, pp. 135–178. DOI: 10.1007/978-3-030-51569-0_6. URL: https://doi.org/10.1007/978-3-030-51569-0_6.
- [38] Daniel Hiller et al. “Structural Properties of Al-O Monolayers in SiO₂ on Silicon and the Maximization of Their Negative Fixed Charge Density”. In: *ACS Appl. Mater. Interfaces* 10.36 (Sept. 2018), pp. 30495–30505. ISSN: 1944-8244. DOI: 10.1021/acsami.8b06098. URL: <https://doi.org/10.1021/acsami.8b06098>.
- [39] Daniel Hiller et al. “The negative fixed charge of atomic layer deposited aluminium oxide—a two-dimensional SiO₂/AlO_x interface effect”. In: *Journal of Physics D: Applied Physics* 54.27 (2021), p. 275304. ISSN: 0022-3727. DOI: 10.1088/1361-6463/abf675.
- [40] B. Hoex et al. “On the c-Si surface passivation mechanism by the negative-charge-dielectric Al₂O₃”. In: *Journal of Applied Physics* 104.11 (Dec. 2008), p. 113703. ISSN: 0021-8979. DOI: 10.1063/1.3021091.
- [41] B. Hoex et al. “Ultralow surface recombination of c-Si substrates passivated by plasma-assisted atomic layer deposited Al₂O₃”. In: *Appl. Phys. Lett.* 89.4 (July 2006), p. 042112. ISSN: 0003-6951. DOI: 10.1063/1.2240736. URL: <https://doi.org/10.1063/1.2240736>.
- [42] Arne Hollmann et al. “Large, Tunable Valley Splitting and Single-Spin Relaxation Mechanisms in a Si/Si_xGe_{1-x} Quantum Dot”. In: *PRAPPLIED* 13.3 (Mar. 2020), p. 034068. DOI: 10.1103/PhysRevApplied.13.034068. URL: <https://link.aps.org/doi/10.1103/PhysRevApplied.13.034068>.
- [43] Chiao-Ti Huang. “Electrical and Material Properties of Strained Silicon/Relaxed Silicon Germanium Heterostructures for Single-Electron Quantum Dot Applications”. PhD thesis. Princeton University, 2015.

- [44] Chiao-Ti Huang, Jiun-Yun Li, and James C. Sturm. “Very Low Electron Density in Undoped Enhancement-Mode Si/SiGe Two-Dimensional Electron Gases with Thin SiGe Cap Layers”. In: *ECS Transactions* 53.3 (2013), p. 45. ISSN: 1938-5862. DOI: 10.1149/05303.0045ecst. URL: <https://dx.doi.org/10.1149/05303.0045ecst>.
- [45] Chiao-Ti Huang et al. “Screening of remote charge scattering sites from the oxide/silicon interface of strained Si two-dimensional electron gases by an intermediate tunable shielding electron layer”. In: *Appl. Phys. Lett.* 104.24 (June 2014), p. 243510. ISSN: 0003-6951. DOI: 10.1063/1.4884650.
- [46] Siegfried Hunklinger. *Festkörperphysik*. Berlin, Boston: De Gruyter Oldenbourg, 2018. ISBN: 9783110567755. DOI: 10.1515/9783110567755.
- [47] A. K. Hüttel et al. “Spin blockade in ground-state resonance of a quantum dot”. In: *Europhysics Letters* 62.5 (2003), p. 712. ISSN: 0295-5075. DOI: 10.1209/epl/i2003-00431-5. URL: <https://dx.doi.org/10.1209/epl/i2003-00431-5>.
- [48] Eric R. Jette and Frank Foote. “Precision Determination of Lattice Constants”. In: *J. Chem. Phys.* 3.10 (Nov. 1935), pp. 605–616. ISSN: 0021-9606. DOI: 10.1063/1.1749562. URL: <https://doi.org/10.1063/1.1749562>.
- [49] A. C. Johnson et al. “Triplet-singlet spin relaxation via nuclei in a double quantum dot”. In: *Nature* 435.7044 (2005), pp. 925–928. ISSN: 1476-4687. DOI: 10.1038/nature03815. URL: <https://doi.org/10.1038/nature03815>.
- [50] Alexander Comstock Johnson. “Charge Sensing and Spin Dynamics in GaAs Quantum Dots”. PhD thesis. Harvard University, 2005.
- [51] Evan O. Kane. “The semi-empirical approach to band structure”. In: *Journal of Physics and Chemistry of Solids* 8 (1959), pp. 38–44. ISSN: 0022-3697. DOI: 10.1016/0022-3697(59)90269-0. URL: <https://www.sciencedirect.com/science/article/pii/0022369759902690>.
- [52] E. Kawakami et al. “Electrical control of a long-lived spin qubit in a Si/SiGe quantum dot”. In: *Nature Nanotechnology* 9.9 (2014), pp. 666–670. ISSN: 1748-3395. DOI: 10.1038/nnano.2014.153. URL: <https://doi.org/10.1038/nnano.2014.153>.

- [53] Youngseok Kim et al. “Evidence for the utility of quantum computing before fault tolerance”. In: *Nature* 618.7965 (2023), pp. 500–505. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06096-3. URL: <https://doi.org/10.1038/s41586-023-06096-3>.
- [54] C. Kittel. *Introduction to Solid State Physics*. Wiley, 2004. URL: <https://books.google.de/books?id=kym4QgAACAAJ>.
- [55] T. J. Knapp et al. “Characterization of a gate-defined double quantum dot in a Si/SiGe nanomembrane”. In: *Nanotechnology* 27.15 (2016), p. 154002. ISSN: 0957-4484. DOI: 10.1088/0957-4484/27/15/154002. URL: <https://dx.doi.org/10.1088/0957-4484/27/15/154002>.
- [56] Dirk König et al. “Modulation Doping of Silicon using Aluminium-induced Acceptor States in Silicon Dioxide”. In: *Scientific Reports* 7.1 (2017), p. 46703. ISSN: 2045-2322. DOI: 10.1038/srep46703. URL: <https://doi.org/10.1038/srep46703>.
- [57] F. H. L. Koppens et al. “Control and Detection of Singlet-Triplet Mixing in a Random Nuclear Field”. In: *Science* 309.5739 (Aug. 2005), pp. 1346–1350. DOI: 10.1126/science.1113719. URL: <https://doi.org/10.1126/science.1113719>.
- [58] L P Kouwenhoven, D G Austing, and S Tarucha. “Few-electron quantum dots”. In: *Reports on Progress in Physics* 64.6 (June 2001), p. 701. DOI: 10.1088/0034-4885/64/6/201.
- [59] L. P. Kouwenhoven et al. “Excitation Spectra of Circular, Few-Electron Quantum Dots”. In: *Science* 278.5344 (Dec. 1997), pp. 1788–1792. DOI: 10.1126/science.278.5344.1788.
- [60] L. P. Kouwenhoven et al. “Quantized current in a quantum-dot turnstile using oscillating tunnel barriers”. In: *Phys. Rev. Lett.* 67 (12 Sept. 1991), pp. 1626–1629. DOI: 10.1103/PhysRevLett.67.1626.
- [61] Leo P. Kouwenhoven et al. “Electron transport in quantum dots”. In: *Mesoscopic electron transport* (1997), pp. 105–214.
- [62] H. A. Kramers. “Wellenmechanik und halbzahlige Quantisierung”. In: *Zeitschrift für Physik* 39.10 (1926), pp. 828–840. ISSN: 0044-3328. DOI: 10.1007/BF01451751.
- [63] Ludwik Kranz et al. “Exploiting a Single-Crystal Environment to Minimize the Charge Noise on Qubits in Silicon”. In: *Adv. Mater.* 32.40 (Oct. 2020), p. 2003361. ISSN: 0935-9648. DOI: 10.1002/adma.202003361.

- [64] S V Kravchenko and M P Sarachik. “Metal–insulator transition in two-dimensional electron systems”. In: *Reports on Progress in Physics* 67.1 (Dec. 2003), p. 1. DOI: 10 . 1088 / 0034 - 4885 / 67 / 1 / R01. URL: <https://dx.doi.org/10.1088/0034-4885/67/1/R01>.
- [65] Sebastian Krinner et al. “Engineering cryogenic setups for 100-qubit scale superconducting circuit systems”. In: *EPJ Quantum Technology* 6.1 (2019), p. 2.
- [66] M. Lenzlinger and E. H. Snow. “Fowler-Nordheim Tunneling into Thermally Grown SiO₂”. In: *Journal of Applied Physics* 40.1 (Nov. 1969), pp. 278–283. ISSN: 0021-8979. DOI: 10.1063/1.1657043. eprint: https://pubs.aip.org/aip/jap/article-pdf/40/1/278/10556240/278_1_online.pdf.
- [67] Jiun-Yun Li et al. “Extremely high electron mobility in isotopically-enriched ²⁸Si two-dimensional electron gases grown by chemical vapor deposition”. In: *Appl. Phys. Lett.* 103.16 (Oct. 2013), p. 162105. ISSN: 0003-6951. DOI: 10.1063/1.4824729. URL: <https://doi.org/10.1063/1.4824729>.
- [68] Yujia Liu et al. “Role of critical thickness in SiGe/Si/SiGe heterostructure design for qubits”. In: *J. Appl. Phys.* 132.8 (Aug. 2022), p. 085302. ISSN: 0021-8979. DOI: 10.1063/5.0101753. URL: <https://doi.org/10.1063/5.0101753>.
- [69] T. M. Lu et al. “Capacitively induced high mobility two-dimensional electron gas in undoped Si/Si_{1-x}Ge_x heterostructures with atomic-layer-deposited dielectric”. In: *Appl. Phys. Lett.* 90.18 (May 2007), p. 182114. ISSN: 0003-6951. DOI: 10 . 1063 / 1 . 2736273. URL: <https://doi.org/10.1063/1.2736273>.
- [70] T. M. Lu et al. “Observation of two-dimensional electron gas in a Si quantum well with mobility of 1.6×10^6 cm²/Vs”. In: *Applied Physics Letters* 94.18 (May 2009), p. 182102. ISSN: 0003-6951. DOI: 10.1063/1.3127516. eprint: https://pubs.aip.org/aip/apl/article-pdf/doi/10.1063/1.3127516/14412611/182102_1_online.pdf. URL: <https://doi.org/10.1063/1.3127516>.
- [71] T. M. Lu et al. “Upper limit of two-dimensional electron density in enhancement-mode Si/SiGe heterostructure field-effect transistors”. In: *Appl. Phys. Lett.* 99.15 (Oct. 2011), p. 153510. ISSN: 0003-6951. DOI: 10.1063/1.3652909.

- [72] J. Maserjian. “Tunneling in thin MOS structures”. In: *J. Vac. Sci. Technol.* 11.6 (Nov. 1974), pp. 996–1003. ISSN: 0022-5355. DOI: 10.1116/1.1318719.
- [73] J. W. Matthews and A. E. Blakeslee. “Defects in epitaxial multilayers: II. Dislocation pile-ups, threading dislocations, slip lines and cracks”. In: *Journal of Crystal Growth* 29.3 (1975), pp. 273–280. ISSN: 0022-0248. DOI: 10.1016/0022-0248(75)90171-2. URL: <https://www.sciencedirect.com/science/article/pii/0022024875901712>.
- [74] R. Maurand et al. “A CMOS silicon spin qubit”. In: *Nature Communications* 7.1 (2016), p. 13575. ISSN: 2041-1723. DOI: 10.1038/ncomms13575. URL: <https://doi.org/10.1038/ncomms13575>.
- [75] Thomas McJunkin et al. “SiGe quantum wells with oscillating Ge concentrations for quantum dot qubits”. In: *Nature Communications* 13.1 (2022), p. 7777. ISSN: 2041-1723. DOI: 10.1038/s41467-022-35510-z. URL: <https://doi.org/10.1038/s41467-022-35510-z>.
- [76] U. Meirav, M. A. Kastner, and S. J. Wind. “Single-electron charging and periodic conductance resonances in GaAs nanostructures”. In: *Phys. Rev. Lett.* 65 (6 Aug. 1990), pp. 771–774. DOI: 10.1103/PhysRevLett.65.771.
- [77] B. Meurer, D. Heitmann, and K. Ploog. “Single-electron charging of quantum-dot atoms”. In: *PRL* 68.9 (Mar. 1992), pp. 1371–1374. DOI: 10.1103/PhysRevLett.68.1371. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.68.1371>.
- [78] Mario Motta and Julia E. Rice. “Emerging quantum computing algorithms for quantum chemistry”. In: *WIREs Comput Mol Sci* 12.3 (May 2022), e1580. ISSN: 1759-0876. DOI: 10.1002/wcms.1580. URL: <https://doi.org/10.1002/wcms.1580>.
- [79] Yoshimasa Murayama and Kasai Jun-ichi. “Experimental determination of effective mass within the energy gap”. In: *Surface Science* 263.1 (1992), pp. 604–608. ISSN: 0039-6028. DOI: 10.1016/0039-6028(92)90418-6. URL: <https://www.sciencedirect.com/science/article/pii/0039602892904186>.
- [80] K. Ono et al. “Current Rectification by Pauli Exclusion in a Weakly Coupled Double Quantum Dot System”. In: *Science* 297.5585 (Aug. 2002), pp. 1313–1317. DOI: 10.1126/science.1070958. URL: <https://doi.org/10.1126/science.1070958>.

- [81] Keiji Ono, Takahiro Mori, and Satoshi Moriyama. “High-temperature operation of a silicon qubit”. In: *Scientific Reports* 9.1 (2019), p. 469. ISSN: 2045-2322. DOI: 10.1038/s41598-018-36476-z. URL: <https://doi.org/10.1038/s41598-018-36476-z>.
- [82] Soumen Pal et al. “Future Potential of Quantum Computing and Simulations in Biological Science”. In: *Molecular Biotechnology* (2023). ISSN: 1559-0305. DOI: 10.1007/s12033-023-00863-3. URL: <https://doi.org/10.1007/s12033-023-00863-3>.
- [83] E. Paladino et al. “ $1/f$ noise: Implications for solid-state quantum information”. In: *RMP* 86.2 (Apr. 2014), pp. 361–418. DOI: 10.1103/RevModPhys.86.361.
- [84] Brian Paquelet Wuetz et al. “Reducing charge noise in quantum dots by using thin silicon quantum wells”. In: *Nature Communications* 14.1 (2023), p. 1385. ISSN: 2041-1723. DOI: 10.1038/s41467-023-36951-w. URL: <https://doi.org/10.1038/s41467-023-36951-w>.
- [85] Jiwoong Park et al. “Coulomb blockade and the Kondo effect in single-atom transistors”. In: *Nature* 417.6890 (2002), pp. 722–725. ISSN: 1476-4687. DOI: 10.1038/nature00791.
- [86] G. H. Parker and C. A. Mead. “Tunneling in CdTe Schottky Barriers”. In: *Phys. Rev.* 184 (3 Aug. 1969), pp. 780–787. DOI: 10.1103/PhysRev.184.780.
- [87] Douglas J. Paul. “Si/SiGe heterostructures: from material and physics to devices and circuits”. In: *Semiconductor Science and Technology* 19.10 (2004), R75. ISSN: 0268-1242. DOI: 10.1088/0268-1242/19/10/R02. URL: <https://dx.doi.org/10.1088/0268-1242/19/10/R02>.
- [88] J. R. Petta et al. “Coherent Manipulation of Coupled Electron Spins in Semiconductor Quantum Dots”. In: *Science* 309.5744 (Sept. 2005), pp. 2180–2184. DOI: 10.1126/science.1116955. URL: <https://doi.org/10.1126/science.1116955>.
- [89] R. Pillarisetty et al. “Si MOS and Si/SiGe quantum well spin qubit platforms for scalable quantum computing”. In: *2021 IEEE International Electron Devices Meeting (IEDM)*. 2021, pp. 14.1.1–14.1.4. DOI: 10.1109/IEDM19574.2021.9720567.
- [90] John Preskill. “Quantum Computing in the NISQ era and beyond”. In: *Quantum* 2 (Aug. 2018), p. 79. ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79. URL: <https://doi.org/10.22331/q-2018-08-06-79>.

- [91] Hua Qin and David A. Williams. “Radio-frequency point-contact electrometer”. In: *Appl. Phys. Lett.* 88.20 (May 2006), p. 203506. ISSN: 0003-6951. DOI: 10.1063/1.2205159. URL: <https://doi.org/10.1063/1.2205159>.
- [92] *qupulse: A Quantum compUting PULse parametrization and SEquencing framework*. URL: <https://github.com/qutech/qupulse>.
- [93] D. C. Ralph, C. T. Black, and M. Tinkham. “Spectroscopic Measurements of Discrete Electronic States in Single Metal Particles”. In: *Phys. Rev. Lett.* 74 (16 Apr. 1995), pp. 3241–3244. DOI: 10.1103/PhysRevLett.74.3241.
- [94] L. P. Rokhinson et al. “Spin transitions in a small Si quantum dot”. In: *PRB* 63.3 (Jan. 2001), p. 035321. DOI: 10.1103/PhysRevB.63.035321. URL: <https://link.aps.org/doi/10.1103/PhysRevB.63.035321>.
- [95] N. Samkharadze et al. “Strong spin-photon coupling in silicon”. In: *Science* 359.6380 (Mar. 2018), pp. 1123–1127. DOI: 10.1126/science.aar4054. URL: <https://doi.org/10.1126/science.aar4054>.
- [96] Friedrich Schäffler. “High-mobility Si and Ge structures”. In: *Semiconductor Science and Technology* 12.12 (1997), p. 1515. ISSN: 0268-1242. DOI: 10.1088/0268-1242/12/12/001. URL: <https://dx.doi.org/10.1088/0268-1242/12/12/001>.
- [97] W. Schottky. “Über den Einfluss von Strukturwirkungen, besonders der Thomsonschen Bildkraft, auf die Elektronenemission der Metalle”. In: *Physikalische Zeitschrift* 15 (1914), pp. 872–878.
- [98] Inga Seidler et al. “Conveyor-mode single-electron shuttling in Si/SiGe for a scalable quantum computing architecture”. In: *npj Quantum Information* 8.1 (2022), p. 100. ISSN: 2056-6387. DOI: 10.1038/s41534-022-00615-2.
- [99] Zhan Shi et al. “Tunable singlet-triplet splitting in a few-electron Si/SiGe quantum dot”. In: *Appl. Phys. Lett.* 99.23 (Dec. 2011), p. 233108. ISSN: 0003-6951. DOI: 10.1063/1.3666232. URL: <https://doi.org/10.1063/1.3666232>.
- [100] Elliot Snider et al. “RETRACTED ARTICLE: Room-temperature superconductivity in a carbonaceous sulfur hydride”. In: *Nature* 586.7829 (2020), pp. 373–377. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2801-z. URL: <https://doi.org/10.1038/s41586-020-2801-z>.

- [101] Tom Struck et al. “Low-frequency spin qubit energy splitting noise in highly purified 28Si/SiGe”. In: *npj Quantum Information* 6.1 (2020), p. 40. ISSN: 2056-6387. DOI: 10.1038/s41534-020-0276-2.
- [102] N Sugii et al. “High electron mobility in strained Si channel of heterostructure with abrupt interface”. In: *Semiconductor Science and Technology* 13.8A (Aug. 1998), A140. DOI: 10.1088/0268-1242/13/8A/040. URL: <https://dx.doi.org/10.1088/0268-1242/13/8A/040>.
- [103] S. M. Sze and Kwok K. Ng. “Metal-Insulator-Semiconductor Capacitors”. In: *Physics of Semiconductor Devices*. John Wiley & Sons, Ltd, 2006, pp. 197–240. ISBN: 9780470068328. DOI: 10.1002/9780470068328.ch4. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470068328.ch4>.
- [104] Ting-Wei Tang, Terrance O’Regan, and Bo Wu. “Thomas-Fermi approximation for a two-dimensional electron gas at low temperatures”. In: *J. Appl. Phys.* 95.12 (June 2004), pp. 7990–7997. ISSN: 0021-8979. DOI: 10.1063/1.1739292.
- [105] L. H. Thomas. “The calculation of atomic fields”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 23.5 (1927), pp. 542–548. DOI: 10.1017/S0305004100011683.
- [106] L. A. Tracy et al. “Observation of percolation-induced two-dimensional metal-insulator transition in a Si MOSFET”. In: *PRB* 79.23 (June 2009), p. 235307. DOI: 10.1103/PhysRevB.79.235307. URL: <https://link.aps.org/doi/10.1103/PhysRevB.79.235307>.
- [107] Alex Trellakis, Till Andlauer, and Peter Vogl. “Efficient Solution of the Schrödinger-Poisson Equations in Semiconductor Device Simulations”. In: *Large-Scale Scientific Computing*. Ed. by Ivan Lirkov, Svetozar Margenov, and Jerzy Wasniewski. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 602–609.
- [108] R. Tsu and L. Esaki. “Tunneling in a finite superlattice”. In: *Appl. Phys. Lett.* 22.11 (Oct. 1973), pp. 562–564. ISSN: 0003-6951. DOI: 10.1063/1.1654509.
- [109] H. Van Houten, C. W. J. Beenakker, and A. A. M. Staring. “Coulomb-blockade oscillations in semiconductor nanostructures”. In: *Single charge tunneling: Coulomb blockade phenomena in nanostructures* (1992), pp. 167–216.

- [110] M. Veldhorst et al. “An addressable quantum dot qubit with fault-tolerant control-fidelity”. In: *Nature Nanotechnology* 9.12 (2014), pp. 981–985. ISSN: 1748-3395. DOI: 10.1038/nnano.2014.216. URL: <https://doi.org/10.1038/nnano.2014.216>.
- [111] Sonja Vogel. “Optimization of the single-electron transistor working point in gated Si/Si_{0.7}Ge_{0.3} heterostructures using Schrödinger-Poisson simulations”. MA thesis. Karlsruhe Institute of Technology, 2023.
- [112] Jan von Delft and D.C. Ralph. “Spectroscopy of discrete energy levels in ultrasmall metallic grains”. In: *Physics Reports* 345.2 (2001), pp. 61–173. ISSN: 0370-1573. DOI: 10.1016/S0370-1573(00)00099-5. URL: <https://www.sciencedirect.com/science/article/pii/S0370157300000995>.
- [113] T. F. Watson et al. “A programmable two-qubit quantum processor in silicon”. In: *nature* 555.7698 (2018), pp. 633–637.
- [114] Z. A. Weinberg. “On tunneling in metal-oxide-silicon structures”. In: *Journal of Applied Physics* 53.7 (July 1982), pp. 5052–5056. ISSN: 0021-8979. DOI: 10.1063/1.331336. eprint: https://pubs.aip.org/aip/jap/article-pdf/53/7/5052/7978702/5052_1_online.pdf.
- [115] Gregor Wentzel. “Eine Verallgemeinerung der Quantenbedingungen für die Zwecke der Wellenmechanik”. In: *Zeitschrift für Physik* 38.6 (1926), pp. 518–529. ISSN: 0044-3328. DOI: 10.1007/BF01397171.
- [116] Z. Wilamowski et al. “Screening Breakdown on the Route toward the Metal-Insulator Transition in Modulation Doped Si/SiGe Quantum Wells”. In: *PRL* 87.2 (June 2001), p. 026401. DOI: 10.1103/PhysRevLett.87.026401. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.87.026401>.
- [117] Russell D. Young. “Theoretical Total-Energy Distribution of Field-Emitted Electrons”. In: *Phys. Rev.* 113 (1 Jan. 1959), pp. 110–114. DOI: 10.1103/PhysRev.113.110.
- [118] Peter Y. Yu et al. “Electronic Band Structures”. In: *Fundamentals of Semiconductors: Physics and Materials Properties*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 17–106. DOI: 10.1007/978-3-642-00710-1_2. URL: https://doi.org/10.1007/978-3-642-00710-1_2.

- [119] Floris A. Zwanenburg et al. “Silicon quantum electronics”. In: *RMP* 85.3 (July 2013), pp. 961–1019. DOI: 10.1103/RevModPhys.85.961. URL: <https://link.aps.org/doi/10.1103/RevModPhys.85.961>.

A. Appendix

A.1. Documentation

```
class utils.multiplexer.SequentialMultiplexer()
```

Constructor of `SequentialMultiplexer`, a measurement management class.

no_active_nodes is an integer value. It represents the total number of nodes of the currently active measurements. Read only.

```
register_measurement(name, nodes, get_tracedata_func, *args, **kwargs)
```

Registers a measurement.

name is a string, representing the name of the measurement which is to be registered.

nodes is a dictionary, keyed by strings which represent the names of the different data nodes of the measurement to be registered. The values are strings which determine the unit of the corresponding data node.

get_tracedata_func is a callable, which upon call records the latest value of the measurement it is assigned to. E.g. a function transmitting the last measured value from the acquisition electronics to the computer.

**args, **kwargs* additional arguments which are passed to the `get_tracedata_func` callable internally each time when it is invoked.

```
activate_measurement(measurement)
```

Activates a measurement. It will be called during the next iteration of `measure1D`, `measure2D` or `measure3D`.

Raises a `KeyError` if the given measurement is not registered.

measurement is a string, representing the name of the measurement which is to be activated.

`deactivate_measurement(measurement)`

Deactivates a measurement. It will not be called during the next iteration of `measure1D`, `measure2D` or `measure3D`. It is best practice to activate the needed measurements before starting a measurement, and deactivate them right after.

Raises a `KeyError` if the given measurement is not registered.

measurement is a string, representing the name of the measurement which is to be activated.

`prepare_measurement_datasets(coords)`

Creates `qkit.measure.measurement_base.Data` datasets for all active measurements and returns them in a list. Raises an `AssertionError` if no datasets could be created.

coords is a list of `qkit.measure.measurement_base.Coordinate` objects. Each of the contained objects represents one of the axis along which the measurements will be taken.

`measure()`

Calls all active measurements, and returns the recorded values in a dictionary. The dictionary is keyed by strings representing the current measurement node in the following structure:

"`measurement_name.node_name`". The values of the dictionary are filled with the corresponding measured data.

`class` `utils.multiplexer.Watchdog()`

Constructor of `Watchdog`, a measurement bounds management class.

`stop` is a boolean value. Defaults to `False`. If the value of any of the registered measurement nodes is out of bounds, `stop` will be set to `true`.

message is string. Defaults to "". If the value of any of the registered measurement nodes is out of bounds, the message will be set.

register_node(*data_node*, *bound_lower*, *bound_upper*)

Registers the measurement bounds for a data node.

Raises a `ValueError` if *bound_lower* is larger or equals to *bound_upper*.

data_node is a string, representing the name of the node whose limits are to be set.

bound_lower is a floating point number, representing the the lowest allowed value for the measurement node.

bound_upper is a floating point number, representing the the highest allowed value for the measurement node.

reset()

Resets the watchdog. The `stop` attribute will be set to `false`.

limits_check(*data_node*, *values*)

Checks whether the given *values* are within the measurement bounds for *data_node*. If not, the `stop` attribute is set, and the `message` attribute is set, informing which node is out of bounds, and whether the upper or lower bound has been exceeded.

Raises a `KeyError` if no bounds are defined for the given *data_node*.

data_nodes is a string, representing the name of the data node whose limits are to be checked.

values is a number or an iterable containing numbers which must be convertible to a `numpy.array`. It should contain the latest measured values for the given data node.

`class spin_tune.Tuning`(*exp_name* = "", *sample* = `None`)

Constructor of `Tuning`, the measurement control class for synchronous data acquisition.

exp_name is a string representing the name of the next run measurement. Is reset to "" after the measurement is completed. When creating the file containing the measurement data, the `measurement_name` will be included in the file name, besides the UUID.

sample is a `qkit.measure.samples_class.Sample` object, which can be used to store parameters of the studied sample.

meander_sweep is a boolean value which is used toggle in between normal mode parameter sweeps and meandering mode parameter sweeps. In measurements with more than one dimension, the values of the two innermost dimensions will be swept in a snake style pattern if `meander_sweep` is set to true, and in a comb style pattern if it is not.

report_static_voltages is a boolean value. If true, the `static_voltages` tab is added to the measurement file. In this tab, all parameters which contain the substring "gate", end with the substring "out", and have a value larger then 0.0004 are logged separately.

register_measurement(*name*, *nodes*, *get_tracedata_func*, **args*, ***kwargs*)

Registers a measurement.

name is a string, representing the name of the measurement which is to be registered.

nodes is a dictionary, keyed by strings which represent the names of the different data nodes of the measurement to be registered. The values are strings which determine the unit of the corresponding data node.

get_tracedata_func is a callable, which upon call records the latest value of the measurement it is assigned to. E.g. a function transmitting the last measured value from the acquisition electronics to the computer.

args*, *kwargs* additional arguments which are passed to the `get_tracedata_func` callable internally each time when it is invoked.

activate_measurement(*measurement*)

Activates a measurement. It will be called during the next iteration of `measure1D`, `measure2D` or `measure3D`.

Raises a `KeyError` if the given measurement is not registered.

measurement is a string, representing the name of the measurement which is to be activated.

deactivate_measurement(*measurement*)

Deactivates a measurement. It will not be called during the next iteration of `measure1D`, `measure2D` or `measure3D`. It is best practice to activate the needed measurements before starting a measurement, and deactivate them right after.

Raises a `KeyError` if the given measurement is not registered.

measurement is a string, representing the name of the measurement which is to be activated.

set_node_bounds(*measurement*, *node*, *bound_lower*, *bound_upper*)

Sets the upper and the lower bounds for a registered measurement node.

Raises a `KeyError` if the given measurement does not exist or the given node does not exist within the given measurement.

measurement is a string, representing the name of the measurement the measurement node belongs to.

node is a string, representing the name of the name of the node whose limits are to be set.

bound_lower is a floating point number, representing the the lowest allowed value for the measurement node.

bound_upper is a floating point number, representing the the highest allowed value for the measurement node.

set_x_parameters(*vector*, *coordname*, *set_obj*, *unit*, *dt = 0*)

Creates the measurement axis in x-direction as qkit coordinate object. A measurement will be performed for each point on the axis.

vector is an array-like where the values which are to be measured are ordered in the sequence in which they are to be measured.

coordname is the name of the coordinate as a string.

set_obj is a callable which is able to set the value of the coordinate in you physical setup. E.g. for a measurement against magnetic field, the driver function which sets the current in your source.

unit The unit of the coordinate values as a string.

dt is the waiting time which is spent in between two measurements as a float. Accurate in the seconds timescale.

set_y_parameters(*vector*, *coordname*, *set_obj*, *unit*, *dt* = 0)

Creates the measurement axis in y-direction as qkit coordinate object. A measurement will be performed for each point on the axis. Arguments have the same meaning as for the x direction, just for another coordinate.

set_z_parameters(*vector*, *coordname*, *set_obj*, *unit*, *dt* = 0)

Creates the measurement axis in z-direction as qkit coordinate object. A measurement will be performed for each point on the axis. Arguments have the same meaning as for the x and y direction, just for another coordinate.

measure1D(*data_to_show* = None)

data_to_show is an optional list or strings representing the names of the datasets which are to be shown by the viewer qviewkit upon measurement start.

Raises `AssertionError` if the x parameter has not been set.

Performs a measurement along the x axis. All active measurements stream data to `Watching`. The data is mapped onto the x axis. E.g. during a streaming event `Watching` received 10 samples for a specific measurement and maps them to the next 10 values of the x axis.

measure2D(*data_to_show* = None)

data_to_show is an optional list or strings representing the names of the datasets which are to be shown by the viewer qviewkit upon measurement start.

Raises `AssertionError` if no x or y parameter has been set.

Performs a measurement along the x and y axis. X is used as outer and y is used as inner coordinate. All active measurements stream data to `Watching`. The data is mapped onto the y axis. The x axis is now used as synchronous coordinate. A use case would be recording a current noise trace versus applied magnetic field.

measure3D(*data_to_show* = None)

data_to_show is an optional list or strings representing the names of the datasets which are to be shown by the viewer qviewkit upon measurement start.

Raises `AssertionError` if no *x*, *y* or *z* parameters have been set.

Performs a measurement along the *x*, *y* and *z* axis. *X* is used as outer, *y* as intermediate and *z* is used as inner coordinate. All active measurements stream data to `Watching`. The data is mapped onto the *z* axis. The *x* and *y* axis are now used as synchronous coordinates. A use case would be recording a current noise trace versus applied magnetic field and temperature.

```
class spin_watch.Watching(exp_name = "", sample = None)
```

Constructor of watching, the measurement control class for asynchronous data acquisition.

exp_name is a string representing the name of the next run measurement. Is reset to "" after the measurement is completed. When creating the file containing the measurement data, the `measurement_name` will be included in the file name, besides the UUID.

sample is a `qkit.measure.samples_class.Sample` object, which can be used to store parameters of the studied sample.

report_static_voltages is a boolean value. If true, the `static_voltages` tab is added to the measurement file. In this tab, all parameters which contain the substring "gate", end with the substring "out", and have a value larger then 0.0004 are logged separately.

```
register_measurement(name, nodes, get_tracedata_func, *args,
**kwargs)
```

Registers a measurement.

name is a string, representing the name of the measurement which is to be registered.

nodes is a dictionary, keyed by strings which represent the names of the different data nodes of the measurement to be registered. The values are strings which determine the unit of the corresponding data node.

get_tracedata_func is a callable, which upon call records the latest value of the measurement it is assigned to. E.g. a function transmitting the last measured value from the acquisition electronics to the computer.

**args, **kwargs* additional arguments which are passed to the *get_tracedata_func* callable internally each time when it is invoked.

activate_measurement(*measurement*)

Activates a measurement. It will be called during the next iteration of *measure1D*, *measure2D* or *measure3D*.

Raises a `KeyError` if the given measurement is not registered.

measurement is a string, representing the name of the measurement which is to be activated.

deactivate_measurement(*measurement*)

Deactivates a measurement. It will not be called during the next iteration of *measure1D*, *measure2D* or *measure3D*. It is best practice to activate the needed measurements before starting a measurement, and deactivate them right after.

Raises a `KeyError` if the given measurement is not registered.

measurement is a string, representing the name of the measurement which is to be activated.

set_node_bounds(*measurement, node, bound_lower, bound_upper*)

Sets the upper and the lower bounds for a registered measurement node.

Raises a `KeyError` if the given measurement does not exist or the given node does not exist within the given measurement.

measurement is a string, representing the name of the measurement the measurement node belongs to.

node is a string, representing the name of the name of the node whose limits are to be set.

bound_lower is a floating point number, representing the the lowest allowed value for the measurement node.

bound_upper is a floating point number, representing the the highest allowed value for the measurement node.

set_x_parameters(*vector*, *coordname*, *set_obj*, *unit*, *dt* = 0)

Creates the measurement axis in x-direction as qkit coordinate object. A measurement will be performed for each point on the axis.

vector is an array-like where the values which are to be measured are ordered in the sequence in which they are to be measured.

coordname is the name of the coordinate as a string.

set_obj is a callable which is able to set the value of the coordinate in you physical setup. E.g. for a measurement against magnetic field, the driver function which sets the current in your source.

unit The unit of the coordinate values as a string.

dt is the waiting time which is spent in between two measurements as a float. Accurate in the seconds timescale.

set_y_parameters(*vector*, *coordname*, *set_obj*, *unit*, *dt* = 0)

Creates the measurement axis in y-direction as qkit coordinate object. A measurement will be performed for each point on the axis. Arguments have the same meaning as for the x direction, just for another coordinate.

set_z_parameters(*vector*, *coordname*, *set_obj*, *unit*, *dt* = 0)

Creates the measurement axis in z-direction as qkit coordinate object. A measurement will be performed for each point on the axis. Arguments have the same meaning as for the x and y direction, just for another coordinate.

measure1D(*data_to_show* = None)

data_to_show is an optional list or strings representing the names of the datasets which are to be shown by the viewer qviewkit upon measurement start.

Raises `AssertionError` if the x parameter has not been set.

Performs a measurement along the x axis. For each point on the x axis all active registered measurements are called.

measure2D(*data_to_show* = None)

data_to_show is an optional list or strings representing the names of the datasets which are to be shown by the viewer qviewkit upon measurement start.

Raises `AssertionError` if no x or y parameter has been set.

Performs a measurement along the x and y axis. For each point all active registered measurements are called. X is used as outer and y is used as inner coordinate.

measure3D(*data_to_show* = None)

data_to_show is an optional list or strings representing the names of the datasets which are to be shown by the viewer qviewkit upon measurement start.

Raises `AssertionError` if no x,y or z parameters have been set.

Performs a measurement along the x,y and z axis. For each point all active registered measurements are called. X is used as outer, y as intermediate and z is used as inner coordinate.

```
class spin_excite.Qupulse_decoder2(*experiments,  
channel_sample_rates, measurement_sample_rates,  
deep_render = False, **kwargs)
```

Constructor of `Qupulse_decoder2`, the virtual experiment decoder class. The constructor will raise a `TypeError` if the experimental instructions are not qupulse pulse templates, and their parameters are not given in the form of dictionaries. Will raise a `ValueError` if different experiments share the same measurements or channels. Will raise a `ValueError` if channels or measurements defined in the experiments have no assigned sampling rates by *channel_sample_rates* and *measurement_sample_rates*

**experiments*, Tuples of experimental instructions and their parameters. Currently qupulse is supported as experiment design framework.

channel_sample_rates is a dictionary where each key represents a channel by its name as a string, and the value the sampling rate of said channel in samples per second as an integer.

measurement_sample_rates is a dictionary where each key represents a measurement by its name as a string, and the value the sampling rate of said measurement in samples per second as an integer.

deep_render is a Boolean value. If set to True, waveforms of different iterations are rendered separately and returned as a list where each entry represents the samples for the given iteration. If set to False, the list is returned flattened.

channel_pars is a dictionary containing the decoded channel instructions. It is a nested dictionary, with the first level containing the channels represented by their names as strings, and the second level containing the channel parameters which are keyed as follows:

"samples" Contains the the values to which to set the voltages of the manipulation channel during each DAC-clockcycle.

measurement_pars is a dictionary containing the decoded measurement instructions. It is a nested dictionary, with the first level containing the measurements represented by their names as strings, and the second level containing the measurement parameters which are keyed as follows:

"measurement_count" The number of iterations to complete until a measurement has collected all requested data.

"measurement_duration" The duration of the measurement in nanoseconds.

"sample_count" The number of samples to record upon a trigger for a single measurement iteration.

"loop_step_name_pp" The name of the pulse parameter iterated during the pulse train. In case a ForLoopPT is passed to the decoder, this corresponds to the name of the loop variable.

"loop_step_name_tt" defaults to "measurement_time".

```
class spin_excite.Settings(ro_backend, ma_backend,  
channel_params, measurement_params, averages, **kwargs)
```

Constructor of Settings, the generalized hardware interface class. The constructor will raise a `ValueError` if not all measurements defined in `measurement_params` are assigned a number of averages by `averages`.

Will raise an `AttributeError` if channels or measurements requested in `measurement_params` and `channel_params` are not provided by the manipulation and readout backends.

`ro_backend` is a subclass of the `RO_backend_base`, used as interface to the readout hardware.

`ma_backend` is a subclass of the `MA_backend_base`, used as interface to the manipulation hardware.

`channel_params` is a nested dictionary, where on the first level each key represents a channel by its name. The second level contains the parameters for each channel. An example of the required structure is given by `spin_excite.Qupulse_decoder2.channel_pars`.

`measurement_params` is a nested dictionary, where on the first level each key represents a measurement by its name. The second level contains the parameters for each measurement. An example of the required structure is given by `spin_excite.Qupulse_decoder2.measurement_pars`.

`averages` is a dictionary where each key represents a measurement by its name as a string, and each value the number of averages which are to be taken by said measurement, represented by an integer.

channel_settings is a nested dictionary with a similar structure to `Qupulse_decoder2.measurement_pars`. Hardware relevant information can be added by the `Settings` object.

measurement_settings is a nested dictionary with a similar structure to `Qupulse_decoder2.measurement_pars`. Hardware relevant information is added by the `Settings` object. The second level of the dictionary is expanded by following parameters which are keyed as:

"unit" The unit of the measurement's values.

"data_nodes" The different data nodes which the measurement hardware returns for each measurement.

load()

Applies the settings specified by `measurement_settings` and `channel_settings` to the hardware backends.

```
class spin_excite.FileHandler()
```

Constructor of `FileHandler`, the class responsible for saving the data to the hard drive.

measurement_name is a string representing the name of the next run measurement. Is reset to "" after the measurement is completed. When creating the file containing the measurement data, the `measurement_name` will be included in the file name, besides the UUID.

measurement_function_name is a string representing the name of the function used to run the measurement. Is reset to "" after the measurement is completed. The `measurement_function_name` is logged in the measurement tab of the measurement file.

additional_watch is a boolean value. If true, the "additional_watch" tab is added to the measurement file. The separately logged instrument parameters are chosen as described in `par_search_string`.

par_search_string is a string. Specify "prefix\$\$\$suffix", and all instrument parameters beginning with prefix, having an integer at the position of the placeholder "\$\$", and ending with suffix are logged separately, and can be found in the `additional_watch` tab of the measurement file.

par_search_placeholder is a string. Will be interpreted as placeholder instead of "\$\$" in `par_search_string`.

multiplexer_coordinates behaves like a nested dictionary. The first level is used to group different coordinate dictionaries according to their modes. The second level contains a dictionary where each key is the name of a measurement, and the entry is a list of the associated coordinates.

datasets is a dictionary containing the datasets handled by the `FileHandler` object. The keys hereby are strings representing the names of the datasets, and the entries are `qkit.measure.measurement_base.Data` objects.

update_coordinates(*tag*, *coord_instructions*)

tag is a string which is used to group different coordinate dictionaries, and serves as the first level key in `multiplexer_parameters`.

coord_instructions is a dictionary. It is keyed by strings which represent the names of the measurements to which the values belong. The values are lists of dictionaries, which contain the information needed to construct the measurement coordinates. The dictionaries within the list need to include the following keys:

"coordname" is a string representing the name of the coordinate.

"unit" is a string representing the unit of the coordinate's values.

"values" is an iterable which must be convertible to a `numpy.array`. Contains the values of the coordinate.

add_dset(*set_name*, *coords*, *unit*)

Adds a `qkit.measure.measurement_base.Data` to the datasets dictionary.

set_name is a string representing The name of the dataset which is to be added.

coords is a list of `qkit.measure.measurement_base.Coordinate` objects. coordinates of the dataset.

`qkit.measure.measurement_base.Data` are limited to three coordinates.

unit is a string representing the unit of the dataset.

reset()

Empties `multiplexer_coords` and datasets.

prepare_measurement(*coords*)

Creates a measurement file containing all the specified datasets and their coordinates, as well as additional coordinates, not intrinsic to the datasets. Starts `qviewkit`, the file viewer UI.

coords is a list of `qkit.measure.measurement_base.Coordinate` objects. It represents additional coordinates, not intrinsic to the datasets.

write_to_file(*set_name*, *value*, *data_location*)

Writes the given values directly to the file on the hard drive.

set_name is a string representing the name of the dataset to which the incoming is written.

value is an iterable with a dimensionality from one to three. Must be convertible to a `numpy.array`. It contains the values which are to be written.

data_location is a tuple of integers, representing the location to which *value* is to be written within the file matrix.

next_matrix(*dset_name*)

Iterates through the outermost dimension in case of a three dimensional dataset.

dset_name is a string representing the name of the dataset on which the operation is to be performed.

end_measurement()

Closes the measurement file safely, making sure no data is lost. Additionally it saves the created measurement plots as `.png` files, in a sub directory of the location of the measurement file.

```
class spin_excite.Exciting(readout_backend, manipulation_backend,  
*experiments, averages, active_modes = ("PulseParameter",),  
deep_render = False, **add_pars)
```

Constructor of `Exciting`, the experiment control class for synchronized asynchronous data acquisition. The constructor will raise a `TypeError` if the experimental instructions are not qpulse pulse templates, and their parameters are not given in the form of dictionaries. Will raise a `ValueError` if different experiments share the same measurements or channels. Will raise a `ValueError` if not all measurements defined in the experimental instructions are assigned a number of averages by averages. Will raise an `AttributeError` if channels or measurements requested in the experimental instructions are not provided by the manipulation and readout backends.

readout_backend is a subclass of the `RO_backend_base`, used as interface to the readout hardware.

manipulation_backend is a subclass of the `MA_backend_base`, used as interface to the manipulation hardware.

**experiments*, Tuples of experimental instructions and their parameters. Currently qpulse is supported as experiment design framework.

averages is a dictionary where each key represents a measurement by its name as a string, and each value the number of averages which are to be taken by said measurement, represented by an integer.

active_modes is a tuple where each entry is a string or the whole argument is a single string, representing the used data processing mode or modes.

deep_render is a Boolean value. If set to True, waveforms of different iterations are rendered separately and returned as a list where each entry represents the samples for the given iteration. If set to False, the list is returned flattened.

***add_pars* represents additional keyword arguments that can be passed to the constructor. For example, the *measurement_mapping* keyword argument can be passed, which is a dictionary whose keys represent the names of measurements in the experiment instructions as strings, and whose values represent the names of the measurements provided by the hardware as strings, to which the measurement in the experiment instruction is to be assigned. The keyword argument *channel_mapping* is similar, it assigns channels from the experimental instructions to channels provided by the hardware.

mode_path is a path-like object, e.g. a string or a `pathlib.Path`, pointing to the directory where the files containing the data processing modes are stored.

active_modes is a string or a tuple of strings, representing the names of the modes which are used to process the raw data before saving it to file.

```
compile(*experiments, averages, active_modes =  
("PulseParameter",), deep_render = False, **add_pars)
```

Compiles the experimental instructions given by **experiments* to the measurement hardware.

Upon the next call of `measure1D`, `measure2D`, `measure3D` the experiment will be executed as given by the instructions.

This method will raise a `TypeError` if the experimental instructions are not qpulse pulse templates, and their parameters are not given

in the form of dictionaries. Will raise a `ValueError` if different experiments share the same measurements or channels. Will raise a `ValueError` if not all measurements defined in the experimental instructions are assigned a number of averages by `averages`. Will raise an `AttributeError` if channels or measurements requested in the experimental instructions are not provided by the manipulation and readout backends.

The arguments are the same as in the `spin_excite`. **Exciting** constructor.

change_averages(*averages*)

Changes the amount of averages performed without recompiling the whole experiment.

averages is a dictionary where each key represents a measurement by its name as a string, and each value the number of averages which are to be taken by said measurement, represented by an integer.

set_x_parameters(*vector*, *coordname*, *set_obj*, *unit*, *dt* = 0)

Creates the measurement axis in x-direction as qkit coordinate object. A measurement will be performed for each point on the axis.

vector is an array-like where the values which are to be measured are ordered in the sequence in which they are to be measured.

coordname is the name of the coordinate as a string.

set_obj is a callable which is able to set the value of the coordinate in your physical setup. E.g. for a measurement against magnetic field, the driver function which sets the current in your source.

unit The unit of the coordinate values as a string.

dt is the waiting time which is spent in between two measurements as a float. Accurate in the seconds timescale.

set_y_parameters(*vector*, *coordname*, *set_obj*, *unit*, *dt* = 0)

Creates the measurement axis in y-direction as qkit coordinate object. A measurement will be performed for each point on the axis. Arguments have the same meaning as for the x-direction, just for another coordinate.

measure1D()

Performs a measurement with no additional coordinates. The base coordinates are determined by your active modes.

measure2D()

Performs a measurement versus the x-coordinate. The number of coordinates in your measurements will be one plus the number of base coordinates determined by your modes. Due to limitations in the file storage, the maximum number of coordinates is three.

measure3D()

Performs a measurement versus the x- and y-coordinate. The number of coordinates in your measurements will be two plus the number of base coordinates determined by your modes. Due to limitations in the file storage, the maximum number of coordinates is three.