Characterisation of Transmon qubit chips

by

L. M. Janssen

A thesis presented for the degree of
Bachelor of applied and industrial mathematics and bachelor of applied physics

TNW/EEMCS
TU Delft
The Netherlands

Contents

Introduction

Quantum computing may be the key to solving many of the problems our modern world faces today. Just as regular computers boosted our computing power to scales never imagined before, so too could quantum computers allow us to solve problems that we just do not yet have the computational power for. However, a properly functioning quantum computer that outclasses current classical computers is not yet a reality. Many different platforms for quantum computing are being researched. Experiments are being done with diamond vacancies (see, for example, [1]), trapped ions [2], nuclear spins in NMR [3] and silicon quantum dots [4], to name a few. One of the most promising types of qubits is based on superconducting circuits [5]. The most widely studied and most promising superconducting qubit is the Transmon qubit [6]. One of the advantages of this type of qubit is that it can be fabricated using currently available lithographic patterning equipment. However, when fabricating Transmon chips, there is a problem of non-uniformity: when using the exact same method to make two chips, they can be vastly different. As such, every chip needs to be characterised after fabrication. Characterising a chip means finding specific important parameters for the qubits, such as their frequency and coherence times. For use in a quantum computer, qubit frequencies will have te be targeted with more precision than is currently possible [7] [8]. As such, a technique that allows change a qubit's frequency after its fabrication is very useful. In this thesis, such a technique is presented. The characterisation process of a Transmon chip is then explained. The data of the three chips that were subjected to the aforementioned technique is also presented.

Chapter 1

Theory

1.1 Qubit basics

In essence, any two-level quantum system can be used as a qubit. It is the basic unit of quantum computation, just as the regular bit is the basic unit of classical computation. Many different quantum systems can be used to make qubits. One example is the spin of an electron, which can be up, down or a superposition of the two states. That is the essential difference between a regular bit and a qubit: where a normal bit can have only two different states, a qubit the for an infinite number of possible coherent superpositions of both. This means that a qubit can effectively be in two states simultaneously. The state of one qubit is then described as a linear combination of states 0 and 1: $\alpha\left|0\right\rangle + \beta\left|1\right\rangle$, with the complex numbers α and β subject to the normalisation condition: $|\alpha|^2 + |\beta|^2 = 1$. Another important difference is the fact that the measurement of a quantum state results in the collapse of its superposition. This means that whenever you measure, you will always find either $\left|0\right\rangle$ or $\left|1\right\rangle$. There are many different quantum mechanical interpretations of this phenomenon, but the result is the same: only two measurement outcomes are possible.

A second distinguishing feature of qubits is the fact that they have finite energy relaxation and coherence times. Unlike regular bits, in a qubit there is an exponential decay from the excited state $\left|1\right\rangle$ to the ground state $\left|0\right\rangle$. This means that after initialisation, there is only a finite amount of time available to perform computations. This type of decoherence is called energy relaxation. It is quantified as T_1, the decay constant of the energy relaxation. It happens because there are several paths through which energy can get out of the qubits. For example, there are dielectric losses, quasiparticle tunnelling and radiation all contributing to the loss of energy. In qubits, phase decoherence can also be observed. After initialising a qubit to a given state with a certain phase several times, over time this phase will not develop the same way for every instance. This phase decoherence yields another coherence time, T_2. This coherence time is finite mainly because of flux noise from the environment. Increasing these coherence times is an important and necessary step towards making a quantum computer [9].

Another important advantage of qubits its the possibility of entanglement. An example of a (two-qubit) entangled state is the Bell state

$$\left|\psi\right\rangle = \frac{\left|00\right\rangle + \left|11\right\rangle}{\sqrt{2}}. \tag{1.1}$$

When in this state, measuring one of the qubits also yields the state of the second qubit, because they are always the same (and with equal probability $\left|0\right\rangle$ and $\left|1\right\rangle$). This entanglement can be established in Transmon qubits [10], described below.

1.2 The Transmon qubit

The Transmon qubit is a kind of qubit introduced first in 2007 [6]. In essence, it is an improved version of the Cooper-pair box with drastically lower sensitivity to charge noise and an increase in qubit-photon coupling. A Cooper-pair box is a small (mesoscopic) superconducting island [5]. It is connected to a large reservoir of electrons through a Josephson junction and a capacitance. By choosing the relevant parameters correctly, the only relevant quantity in its Hamiltonian is the number of Cooper pairs N on the island. It is described by the same Hamiltonian as the Transmon qubit, the Jaynes-Cummings Hamiltonian:

$$\hat{H} = \hbar \sum_j \omega_j \left|j\right\rangle \left\langle j\right| + \hbar\omega_r \hat{a}^\dagger \hat{a} + \left(\hbar \sum_i g_{i,i+1} \left|i\right\rangle \left\langle i+1\right| \hat{a}^\dagger + H.c.\right), \tag{1.2}$$

with coupling energies

$$\hbar g_{i,i+1} = 2\beta e V^0_{rms} \left\langle i|\hat{n}|i+1\right\rangle = \hbar g_{i+1,i}. \tag{1.3}$$

Here $\hbar$ is the reduced Planck constant, ω_j is the Transmon frequency, ω_r is the resonator frequency, $\hat{a}$ ($\hat{a}^\dagger$) annihilates (creates) one photon is the transmission line, V^0_{rms} is the root-mean-square voltage of the local oscillator given by $V^0_{rms} = \sqrt{\hbar\omega_r/2C_r}$, and β is the ratio of the gate capacitance and the total capacitance, $\beta = C_g/C_\Sigma$ [6]. A Transmon consists of two superconducting islands coupled through two Josephson junctions, which are arranged in parallel. This creates a SQUID loop, which allows for tuning of the Josephson energy. This can be done by applying a flux through the loop. For readout, the Transmon is coupled to a transmission line via a readout resonator. It is possible to have several qubits on a single transmission line, as long as the readout resonator frequencies vary, so that qubit can be distinguished from one another. The main difference between a Transmon and a charge qubit is the difference in the E_J/E_C ratio, which is about 10^2. This is what yields the decrease in charge sensitivity. The Transmon is in itself not really a qubit, since it does not have only two energy levels, like an electron spin. Instead, it is essentially a weakly anharmonic oscillator. However, because of the anharmonicity, it can be operated as a qubit. By only ever sending pulses to excite the system from its ground to first excited state, all other levels in the system become irrelevant, and it behaves like a qubit.

Chapter 2

Laser annealing

Laser annealing is a technique that can be used to change qubit frequencies after fabrication. When making Transmon qubits, it is not yet possible to accurately target qubit frequencies. Laser annealing qubits after fabrication may be a solution to this problem. At the DiCarlo lab, some chips have recently been treated with this new technique. The associated Transmon qubit chips were named Zapmons. The layout of the Zapmon is given in figure 2.1. The idea of the project was to first characterise a 10-qubit chip just as one would any other. Afterwards, the fridge in which they were measured would be warmed up, and the chip taken out. Some of the qubits on the chip would then have their Josephson junctions annealed. A laser would be pointed at one of the the junctions for a certain amount of time, and then at the other for the same duration. This would yield a significant decrease in the qubit frequency.

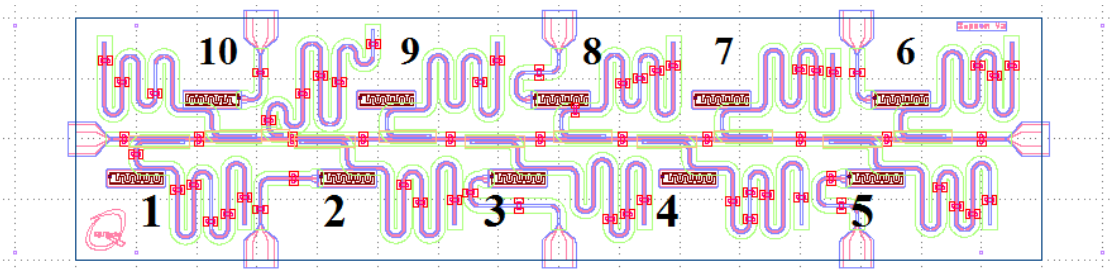


Figure 2.1: The layout of a Zapmon chip. Horizontally in the middle of the chip lays the feedline, which is capacitively coupled to 11 resonators. 10 of these are coupled to qubits, the last is a test resonator. Six of the ten qubits have dedicated flux lines. These can be used to tune the frequency of the qubit they are dedicated to. Additionally, they will generate some flux on neighbouring qubits. As such, these six flux line can be used to tune all qubits in the desired range.

After such a chip is fabricated, a resistance measurement is performed at room temperature to the parallel of Josephson junctions of each qubit. This yields a prediction of the qubit frequency. There are also predictions for the resonator frequencies, based on their length. After this is done, the chip is to be cooled down in a dilution refrigerator, and all the important parameters of interest to this research are characterised. This process is described in the next chapter. After this is done, the chip is warmed up again, and then the process of laser annealing is done. Most of the measured qubits are annealed, and a few are left as control qubits. The setup for laser annealing can be seen in figure 2.2.

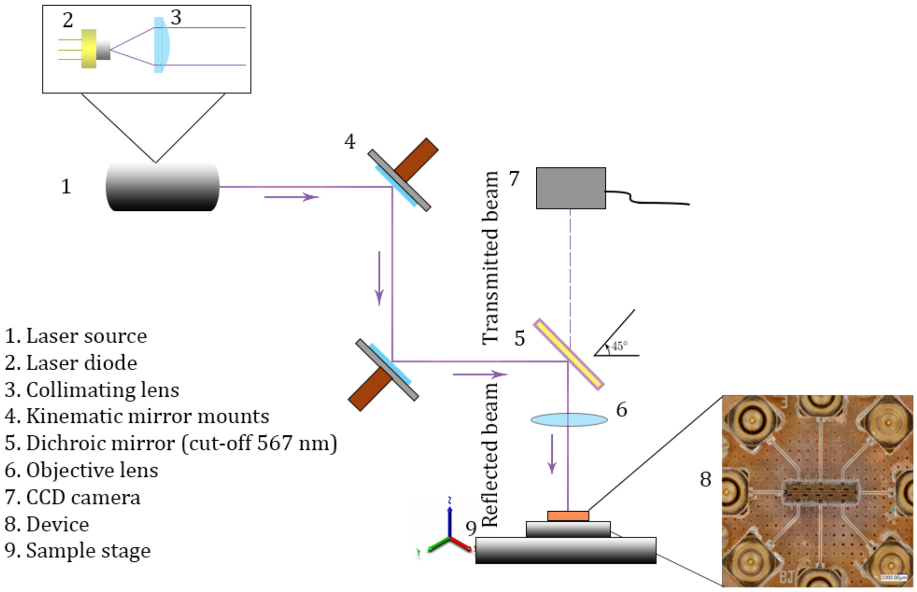


Figure 2.2: The setup used to anneal the Josephson junctions of a Zapmon device. Components one to six are used to get the laser focused and pointing at the right spot. The charge-coupled device (CCD) camera is used to monitor the local temperature on the chip.

After this process of annealing the qubits, the chip is put back into the dilution refrigerator, and characterised a second time. Since many parameters remained unchanged, the basis for the new characterisation is to reload all the previous settings from the first round of measurements. Apart from that small exception, the two characterisation processes are identical, and are detailed in the next chapter.

Chapter 3

Chip characterisation

In this chapter, the process of characterising a chip is presented step by step, with all the relevant figures. All of the results given here, with with a few specified exceptions, are for Zapmon 6.1, the best chip out of the three that have been measured thus far.

3.1 Resonator characterisation

The first step of characterising a chip is the measurement of the resonators. For all of these measurements, a Vector Network Analyser (VNA) is used. The first thing that needs to be done is to find the resonator frequencies. To do so a wide range scan is performed with the VNA, as seen in figure 3.1.

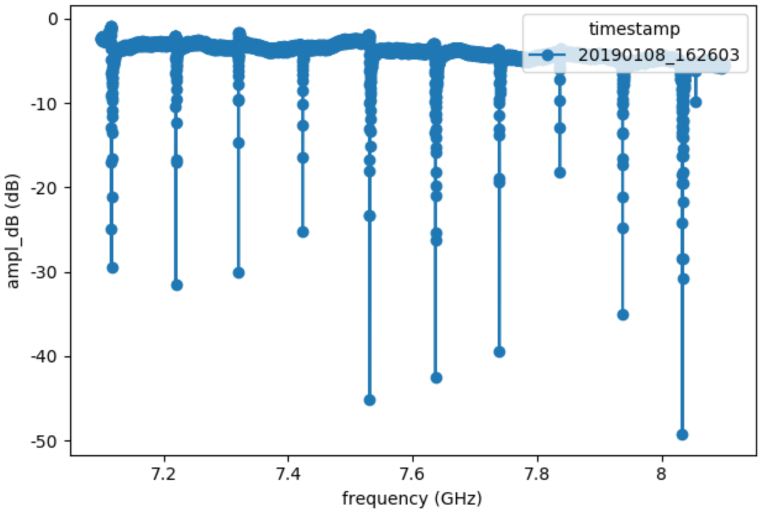


Figure 3.1: An overview scan of all the resonators. All ten qubit resonators are evenly spaced and clearly visible. The small peak on the right is the test resonator, which has a far weaker coupling to the feed line, resulting in a more shallow dip.

Now comes the time to zoom in on each resonator, and do some complex analysis. This will yield not only the frequency of each resonator, but also its quality factor. An example of such a measurement can be seen in figure 3.2. The data obtained in this measurement is fitting using the formula

$$S_{21} = A(1-\alpha\frac{\omega-\omega_r}{\omega_r})(1-\frac{\frac{Q_l}{|Q_e|}e^{i\theta}}{1+2iQ_l\frac{\omega-\omega_r}{\omega_r}})e^{i(\phi_v f+\phi_0)}, \tag{3.1}$$

where ω_r is the resonator frequency, ω is the measuring frequency, Q_e is the extrinsic quality factor, α accounts for any background slope, and ϕ_v and ϕ_0 account for the propagation delay to and from the sample [11].

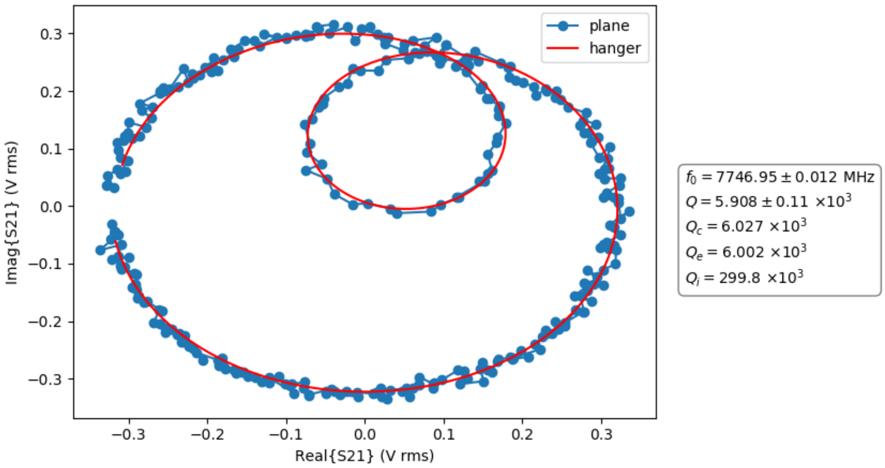


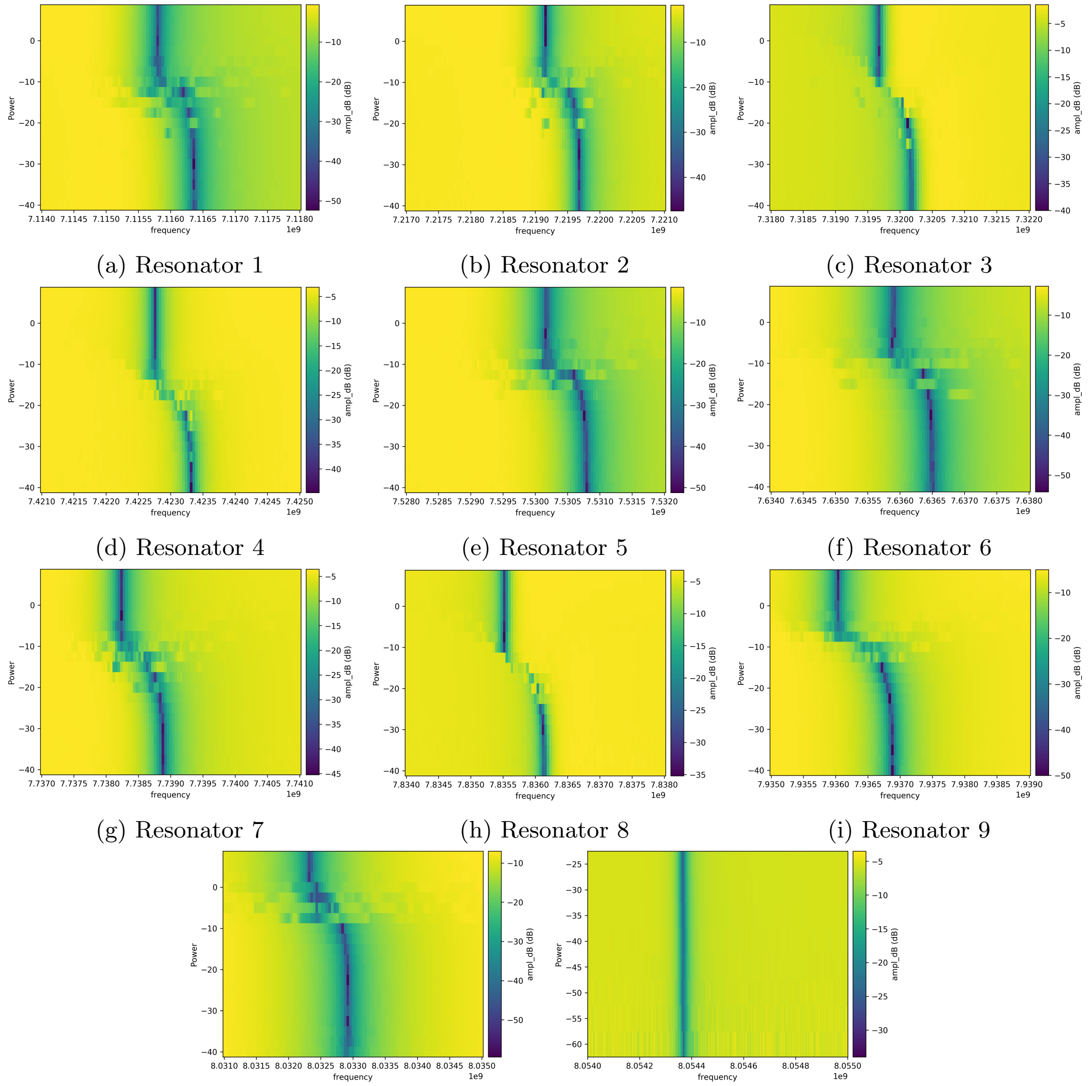
Figure 3.2: A plane view of the complex signal transmitted around a resonator's frequency. Each point represents the transmitted voltage at different frequencies. The smaller loop indicates the resonator. The different quality factors are specified in the legend. Q is the total quality factor, Q_C is the coupling quality factor, Q_e and Q_i are the external and internal qualitfy factors. In general, the higher the quality factors, the better. These values are typical for a Transmon chip.

From these scans, the resonator frequencies can be extracted, as specified in table 3.1

Resonator nr.	Frequency (GHz)
1	7.116
2	7.219
3	7.320
4	7.423
5	7.530
6	7.636
7	7.739
8	7.836
9	7.937
10	8.033
Test	8.0544

Table 3.1: Table with resonator frequencies.

The next step is to establish which resonators have working qubits coupled to them. To do this, a phenomenon called the power shift of a resonator is measured. When a measurement of a resonator is performed at low power, its resonance frequency is shifted by the nearby qubit. At higher powers, the multitude of photons in the resonator causes it to be resonant at its inherent frequency. If the same resonator frequency is observed at all powers, this indicates the absence of a qubit, as seen for the test resonator.



(a) Resonator 1

(b) Resonator 2

(c) Resonator 3

(d) Resonator 4

(e) Resonator 5

(f) Resonator 6

(g) Resonator 7

(h) Resonator 8

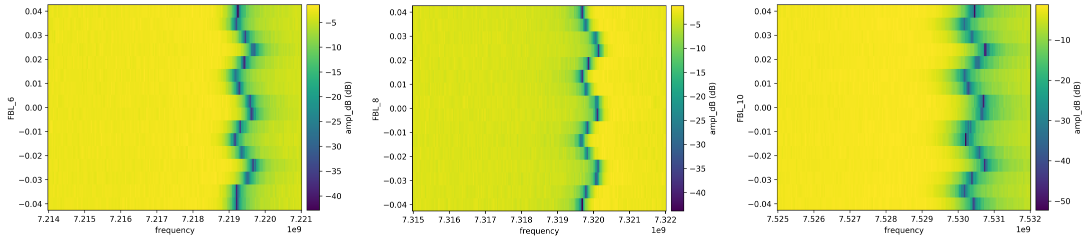
(i) Resonator 9

(j) Resonator 10

(k) Test resonator

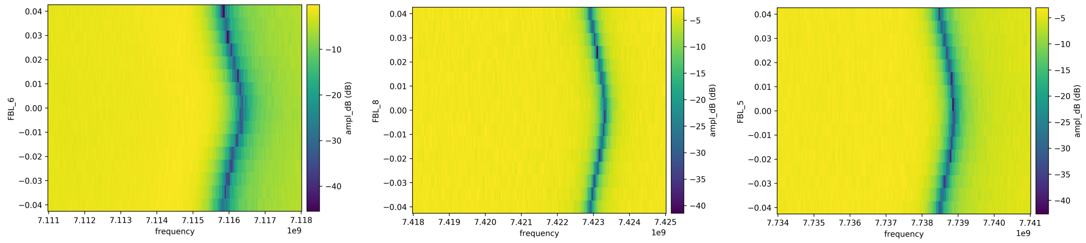
Figure 3.3: Power scans of all resonators. The frequency is varied horizontally, and the power is varied vertically. For the qubit resonators, the power shift is clearly visible. A straight line as seen for the test resonator indicates that there is no qubit present.

After the presence of qubits had been established for all resonators, it is time to do a flux-sweep of all resonator-flux line pairs. This gives us three important pieces of information: it shows which of the flux lines are working properly by showing whether there is an effect when a current is applied, which flux line has the strongest coupling to each qubit (this has the shortest period or the largest shift at maximum current), and gives an estimate of the sweet spot currents for each qubit, because the frequency of the resonator is at a maximum when the frequency of the corresponding qubit is at a maximum. By reading out which current closest to zero yields such a maximum, a good estimate for the sweet spot current is obtained. Note that for this device, a total of 60 measurements of this were performed: each resonator-flux line pair is investigated. Instead of showing all of these, some typical measurements are given in figure 3.4 and 3.5.



(a) Resonator and flux line 2 (b) Resonator and flux line 3 (c) Resonator and flux line 5

Figure 3.4: Resonator flux response to their dedicated flux bias lines. The amount of flux per unit current is large enough to generate several periods for each resonator, meaning the maximum applied flux is several times $\Phi_0 = \frac{h}{2e}$. For measurement purposes, the sweet spot closest to zero current is used, as the lowest current results in the least heating of the chip.



(a) Resonator 1, flux line 2 (b) Resonator 4, flux line 3 (c) Resonator 7, flux line 5

Figure 3.5: Resonator flux response for some resonators without dedicated flux lines. A sweet spot is clearly available for these resonator-flux line combinations. For these resonators, the strongest response is shown. By using these flux lines, the qubits without a dedicated line can still be detuned by several times 100 MHz.

3.2 Finding and preparing qubits

Having done the full resonator characterisation, it is now time to change the setup. The VNA is disconnected and instead the main setup is connected, consisting of a central controller (CC-light) to control the other instruments, a UHFQC to do the readout, some radio frequency sources to act as local oscillators and spectroscopy sources, and an HDAWG to drive our qubits with. With this setup, the search for qubits using spectroscopy can begin. First, the readout frequency for each qubit needs to be specified to be the resonant frequency of the corresponding resonator. To this end a resonator scan is first performed, which simply sends a signal within a frequency range and shows the transmitted power.

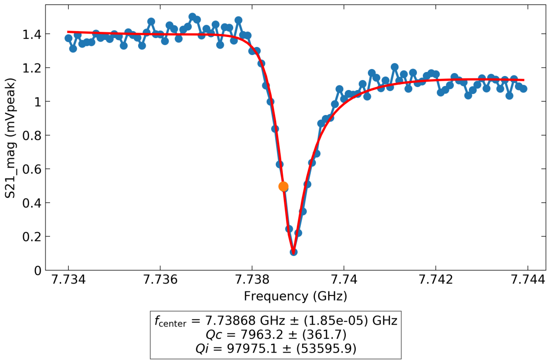


Figure 3.6: Simple resonator measurement for resonator associated with Qubit 7. The minimum is at 7.739 GHz, which corresponds to the value found by the power scan to be in the dispersive regime. As such, the readout power used for this measurement can also be used to find the qubit frequency.

With the VNA power sweeps, it can be checked whether the current power yields the frequency of the low-power regime, by seeing if the minimum of the resonator scan is equal to the value seen for lower powers. In this regime, the influence of the qubit can be seen. When the qubit transitions to one of its excited states, the resonator will shift to a slightly lower frequency. Depending on which frequency is picked for the readout, an increase or decrease in phase and amplitude can be observed when this happens. To find the frequency of the qubit, a spectroscopy tone of varying frequency is send through the feed line. When this tone is at the qubit frequency, the excitation is visible as a spike or dip in our readout, as suggested by the Hamiltonian in the theory section.

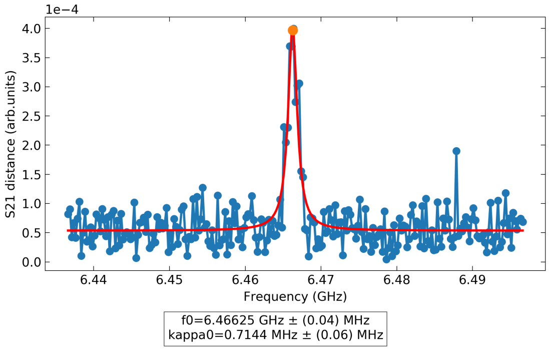


Figure 3.7: This is a spectroscopy measurement for the qubit corresponding to the resonator in figure 3.6. The peak is quite wide here: about 5 MHz. This width can be decreased by decreasing the spectroscopy power, yielding a more precise value for the qubit frequency. Another way to get a more precise value is the Ramsey method, which is explained later.

The next step is to find the qubit sweet spot. Of course, there is already an estimate for this from the resonator flux-sweeps. For every qubit, the most strongly coupled flux line is selected and the sweet spot is estimated using the resonator flux-sweep. Then a spectroscopy measurement is performed as above at current values around the predicted sweet spot. This yields a heat map as seen in figure 3.8. From this a final value for the sweet spot current and the maximum frequency of the qubit is obtained.

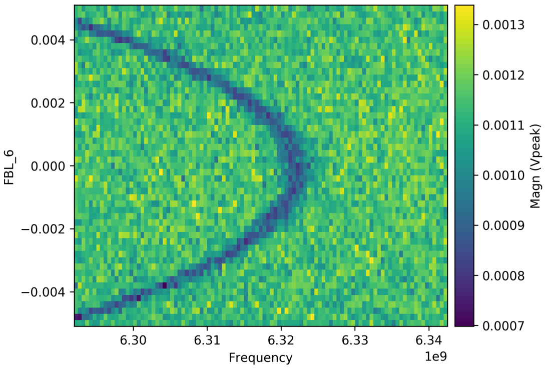


Figure 3.8: This is a Qubit DAC scan. It is a series of spectroscopy measurements in a specified range, repeated while varying the current on a flux line. This causes the qubit frequency to vary. The goal of this measurement is to extract the sweet spot, where the qubit is first order insensitive to flux noise. This is at the vertical part of the curve. The qubit frequency is expected to be a cosine function of the applied flux. Around the sweet spot, it can be approximated with a parabola.

3.3 Time domain measurements

It is now time to tune up the microwave pulses for the qubit. With the current set to a qubit's sweet spot, a Rabi oscillation can be measured. By sending a microwave pulse at the qubit frequency with different amplitudes, the qubit will be excited with different probabilities, up to 100%. Of course, sending no pulse will always result in a ground state measurement, except when a thermal excitation happens. This is very rarely the case due to the low operating temperature. As amplitude is increased, the probability of excitation will steadily rise to 100%. After this amplitude is reached, the probability will decrease again due to stimulated emission, again following a sine pattern. The resulting oscillation is known as a Rabi oscillation. It can be seen for one of the qubits in figure 3.9.

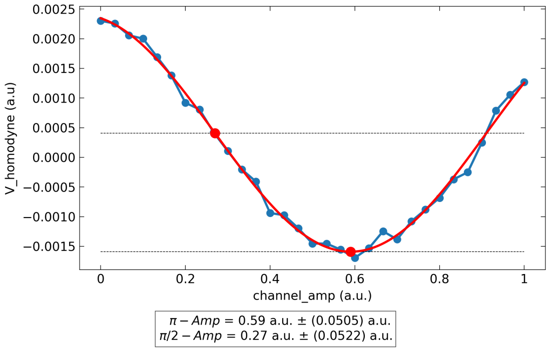


Figure 3.9: This is a Rabi calibration. A microwave pulse is send at the qubit frequency, causing it to be excited with varying probability depending on the amplitude of the pulse. This amplitude is varied along the horizontal axis. From this scan the amplitudes needed for a π-pulse (which bring the qubit into its excited state with maximum probability) and $\pi/2$-pulse (which brings the qubits into its excited states with half-maximum probability) are extracted.

Now that a Rabi pulse is set up, a more precise measurement of the frequency can be made with the Ramsey method. An artificially detuned 90 degrees X-pulse is send, then there is a pause of varying amounts of time, then the same pulse is send another time. This results in an oscillation, the period of which depends on the actual detuning from the precise qubit frequency. As such the qubit frequency can be more precisely measured using this method, with about 0.01 MHz accuracy, limited by the T_2^* and readout noise. With high artificial detuning, the method is resilient to large errors in qubit frequency, but not very precise. As such, a sequence of these measurements is normally made, with decreasing detuning and increasing measurement time, to have a constant amount of oscillations in each measurement. An example of such a sequence can be seen in figure 3.10.

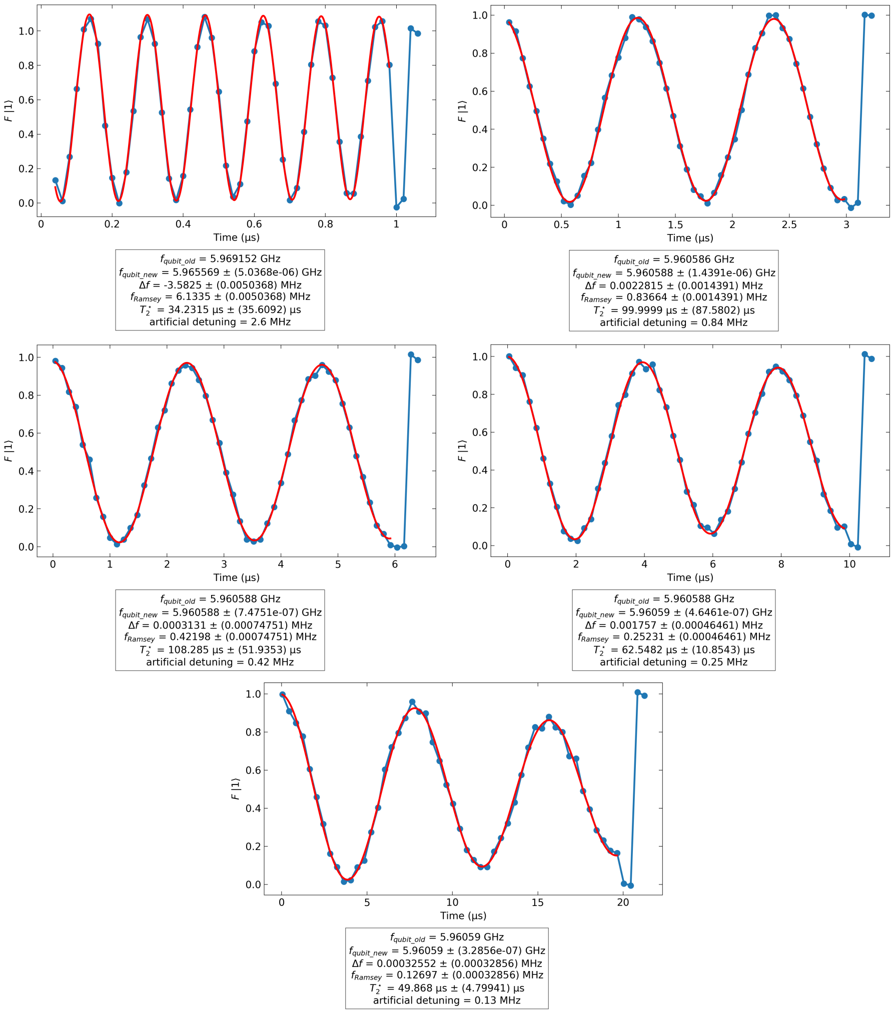


Figure 3.10: A sequence of Ramsey frequency measurements of increasing timescale, and corresponding decreasing artificial detuning. The measurement always aims to detune by an amount such that two and a half period appear in the measurement. As the qubit was just slightly tuned before this measurement, the first measurement shows many more periods. The qubit frequency is this adjusted, and for the second measurement the right amount of periods shows up. In the last of these measurements, the frequency is measured ever more precisely. In the last measurement, the Ramsey decay starts showing. As such, going to longer timescales will not yield more accurate results. The last four points, which are not included in the fit, are calibration points: the first pair is performed after a period of relaxation, the second pair after a π-pulse

Now the time domain characteristics of the qubit can be fully analysed. There are three different measurements that are done repeatedly for each qubit at sweet spot: a T_1 measurement, a Ramsey measurement and an Hahn echo measurement (later referred to simply as echo experiment), as mentioned in the theory section. Each will be explained below.

3.3.1 T_1

The first time domain measurement is a T_1 measurement. It is a measure of the energy decay rate. To measure it, a π-pulse is applied to the qubit and pauses of varying amounts of time are taken before measuring the state of the qubit. Schematically, this is given as

$$\tau X.$$

Note that the above notation should be read right to left to match the time evolution. The X indicates a π-pulse, the τ is the idle period. Repeating this measurement many times (typically around 1000 for each data point) usually yields an exponential decay. By fitting an exponential decay function to the data, the decay time can be extracted, which is T_1.

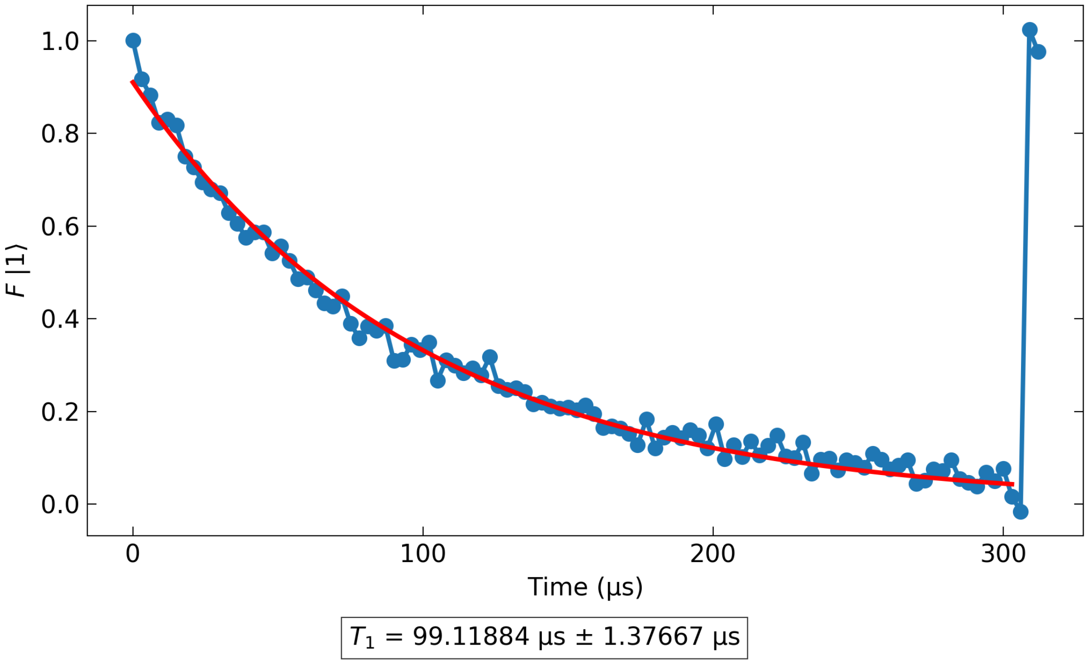


Figure 3.11: A T_1 measurement, showing the exponential decay from the excited state to the ground state. The last 4 points are calibration points. The first two are measured after an idle period to get the ground state, the last two are measured directly after a π-pulse

3.3.2 Ramsey

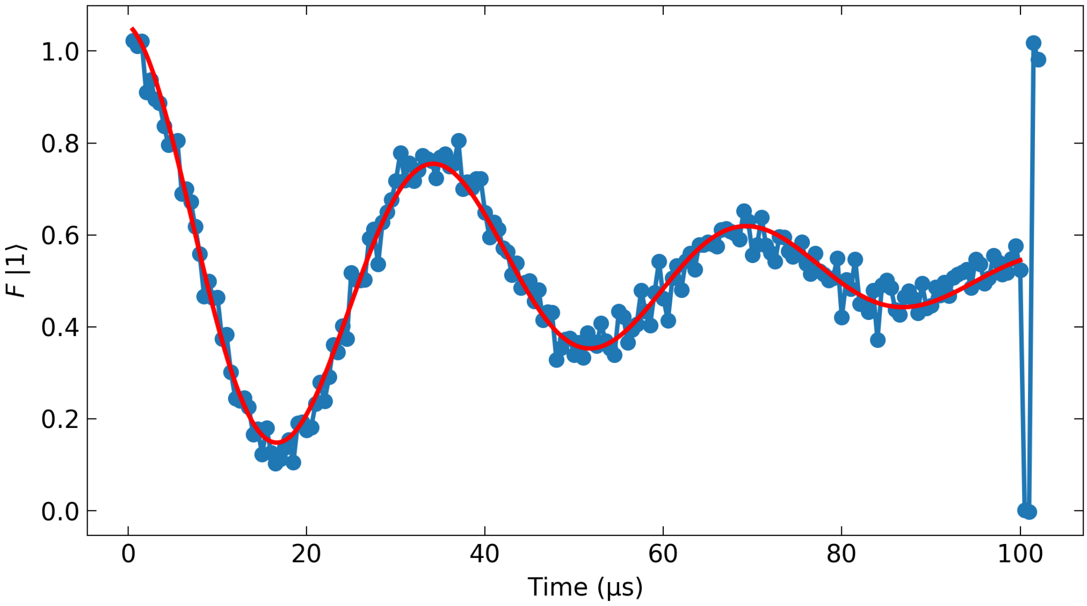
The second coherence measurement is a Ramsey experiment, which is very similar to the Ramsey frequency measurement mentioned earlier. A slightly detuned $\pi/2$-pulse is send, a varying amount of time is waited, and then the same pulse is send another time. This is schematically given by

$$X/2\tau X/2.$$

After the first $\pi/2$-pulse, the qubit state will be approximately $-\hat{y}$ on the Bloch sphere. Then it precesses around $\hat{z}$ for the varying amount of time τ. The second $X/2$ will rotate the state back onto the $+\hat{z}$ or $-\hat{z}$ state. If the artificial detuning were zero, only a decay due to decoherence of the qubit would be visible. However, as the pulse is slightly off-resonant, an oscillating pattern will be observed. From the frequency of this oscillation, the actual detuning can be determined. It is given as Δf in figure 3.12. A cosine multiplied by an exponential decay is fitted, with the following formula:

$$A * e^{(-t/\tau)^n} * (\cos(2\pi f t + \phi) + B) + C. \tag{3.2}$$

The decay time τ of the exponential decay is T_2^*.



f_{qubit_old} = 6.278059 GHz
f_{qubit_new} = 6.27806 ± (1.148e-07) GHz
Δf = 0.001507 ± (0.0001148) MHz
f_{Ramsey} = 0.028493 ± (0.0001148) MHz
$T_2^\star$ = 42.5963 μs ± (1.2592) μs
artificial detuning = 0.03 MHz

Figure 3.12: A Ramsey measurement, fitted with the product of a cosine and an exponential decay function. The oscillations are a result of the detuning. The last 4 points are calibration points. The first two are measured after an idle period to get the ground state, the last two are measured directly after a π-pulse.

3.3.3 Hahn Echo

The final coherence experiment is called the Hahn echo experiment, referred to simply as the echo experiment. It is very similar to the Ramsey measurement, with some key exceptions. The idea of the measurement is to swap around the qubit around the X-axis. This makes the precession flip, which causes decoherence picked up in the first half to partly cancel out in the second half of the sequence, because the precession on the Bloch sphere is reversed and approximately has the same speed. This is because the noise is partly low-frequency, and so is essentially constant for the duration of the sequence. In the same notation as before, it is written as

$$\tau/2X\tau/2X/2.$$

However, instead of the $X/2$ recovery pulse as seen in the Ramsey experiment, recovery pulses are now $\pi/2$-radian pulses around an axis varying in the xy-plane. This axis is rotated by by 20 degrees around the z-axis between every step in the measurement. When it is lined up with where the qubit state is expected to be at the time the pulse is send, an $X/2$-pulse is send instead. This results in a beating pattern very similar to what is observed in the Ramsey experiment. The same fitting method as in the Ramsey experiment is used, resulting once again in a decay constant, which is now T_2^{echo}. Due to the echo part of the sequence, which cancels some of the noise, this is typically longer than T_2^*.

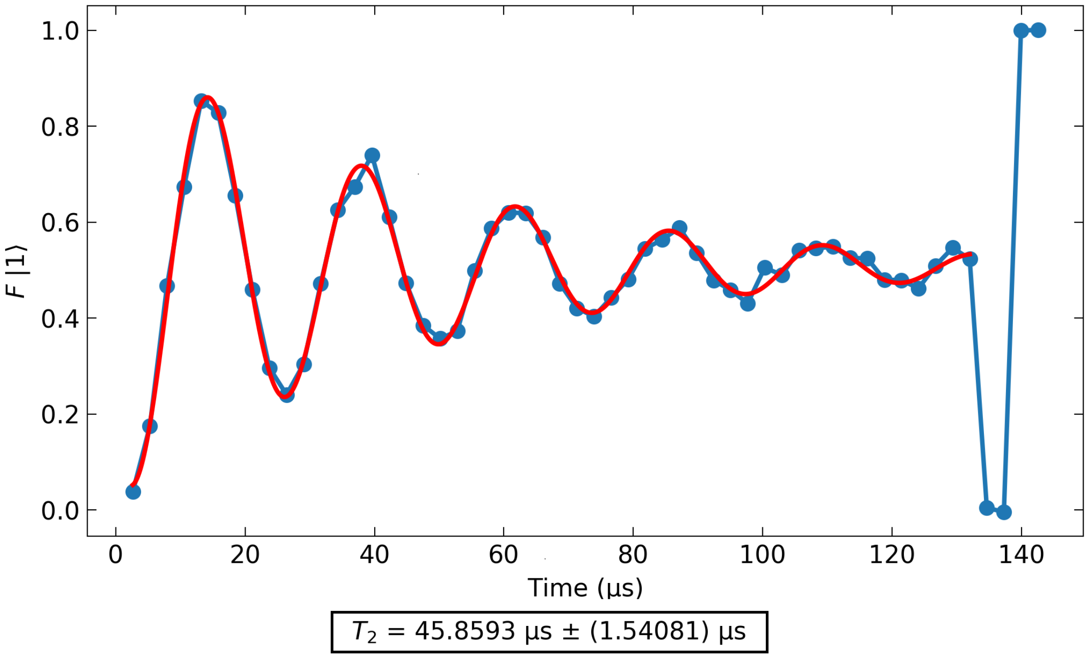


Figure 3.13: An Echo measurement, fitted with the product of a cosine and an exponential decay function. The oscillations are a result of the varying recovery pulse used. The last 4 points are calibration points. The first two are measured after an idle period to get the ground state, the last two are measured directly after a π-pulse. This specific measurement is from a different device. For the Zapmon 6.1, the varying recovery pulses were not yet used. The data shown here is from Zapmon 4.2.

3.3.4 Time variance

Each of these three measurements yields quite different results over time. Especially for higher coherence times, they can vary by up to about 50 % of their average value. In addition to this high random fluctuation, sometimes a strong decrease in all coherence times can be seen for a prolonged period of time, only to disappear later. This is attributed to two-level systems showing up with a frequency close to the resonance frequency of the qubit. This yields a channel through which energy can quickly leak out, resulting in much lower coherence times. This is why it is very important to repeat these measurements many times. It can also be a good idea to alternate these measurements between qubits, so that the time between identical measurements is increased, which makes the measurements less correlated. The mean and standard deviation is calculated for each of these measurement sequences, yielding a complete dataset of the coherence times of the chip.

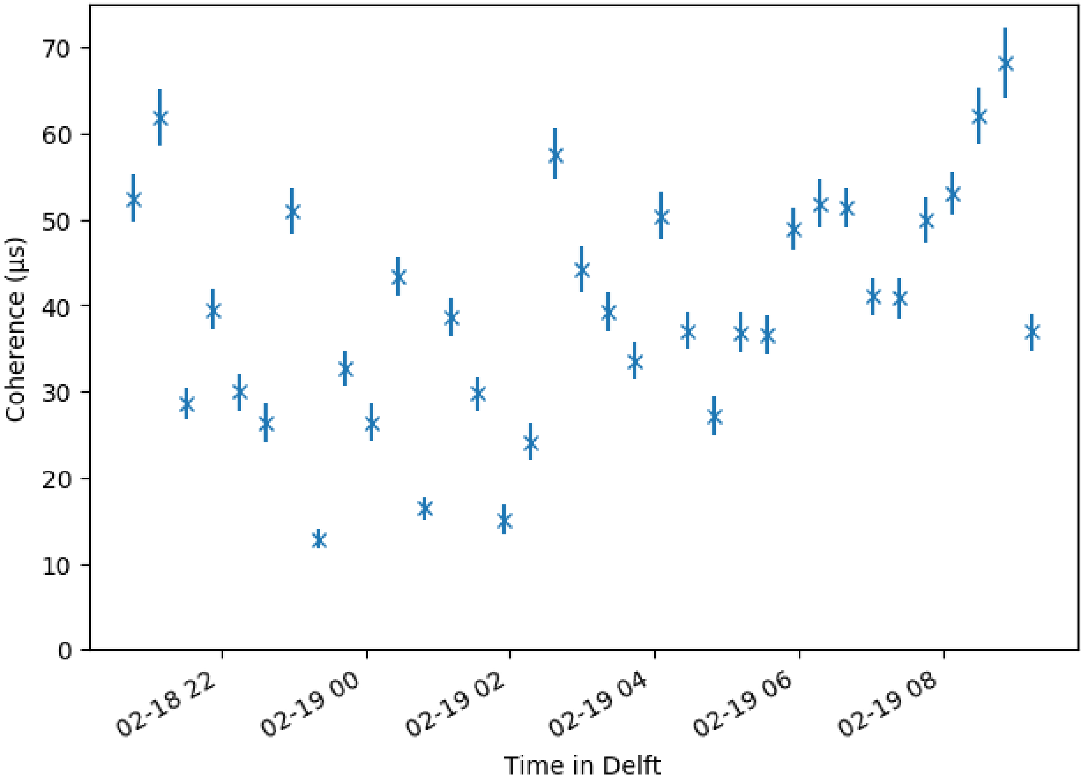


Figure 3.14: This is a strongly fluctuating time trace of a Ramsey coherence measurement, taken on another device. It shows the need to repeat coherence measurement over a long time period. Typical traces are not as noisy as this, as reflected in the standard deviations of the values in table 3.2.

3.4 Anharmonicity

The last important measurement to be performed is an anharmonicity measurement. In a Transmon qubit, there are more than two energy levels. However, due to the non-linearity of Josephson junctions, levels are not evenly spaced. The anharmonicity of a Transmon qubit is the difference between the transition between the ground state and the first excited state, and the transition between the first and second excited states. Two different ways were used to measurement. For the chip that has been discussed so far, a simple spectroscopy measurement was done as before, only with the power increased by about 30 dB. This results in the qubit often being excited to the second excited state by two photons being absorbed simultaneously. The frequency of these two photons will show up in the spectroscopy. The 02-transition is being driven here, the frequency of which is $f_{01} + f_{12}$. The peak will thus appear at $(f_{01} + f_{12})/2$ The anharmonicity of the qubit is equal to the difference between the 01- and 12-transitions, which can be calculated quickly by doubling the difference between the qubit frequency and the 02-peak that shows up in the spectroscopy measurement.

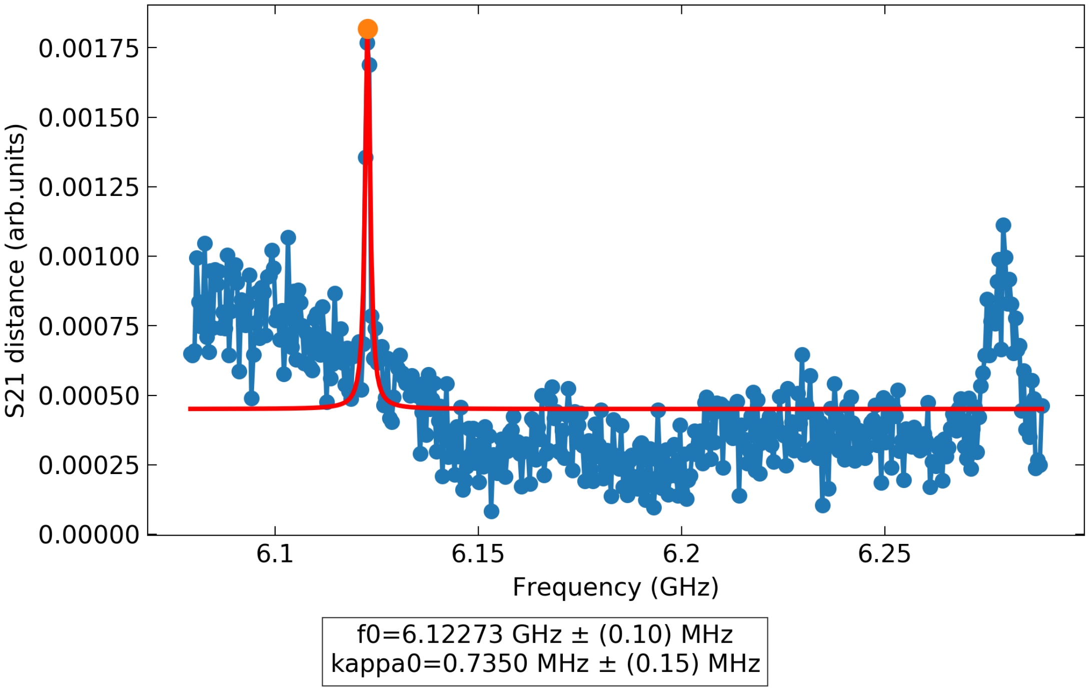


Figure 3.15: A high-power spectroscopy scan where both the 01 and 02 transitions are visible. From the width of the 01 transition it is clear that the power is a lot higher than in a regular spectroscopy measurement. The 02 peak is still quite narrow, and so the frequency found by the peak finder can be used to quite accurately predict the anharmonicity of the qubit.

There is another, more accurate way to measure the anharmonicity, which was used for the chips measured later. A second spectroscopy source is added to the measurement setup. Now two different frequencies can be send simultaneously. One tone is send in a range around the qubit frequency. This is the 01 tone. The second tone is send in the same range around the expected frequency of the 12 transition from the previous measurement. Whenever the first tone is at the qubit frequency, a change in magnitude, frequency or both can be seen, just as in a normal spectroscopy. When the two tones add up to the frequency of the 02 transition, a dip or peak shows up as well. When plotting this in a 2D plane, as in figure 3.16, a vertical line can be seen (the 01 transition), as well as a diagonal where the sum of the two tones is constant. The value of the second tone at the intersection of the two lines is the frequency of the 12 transition. From these values the anharmonicity can be extracted.

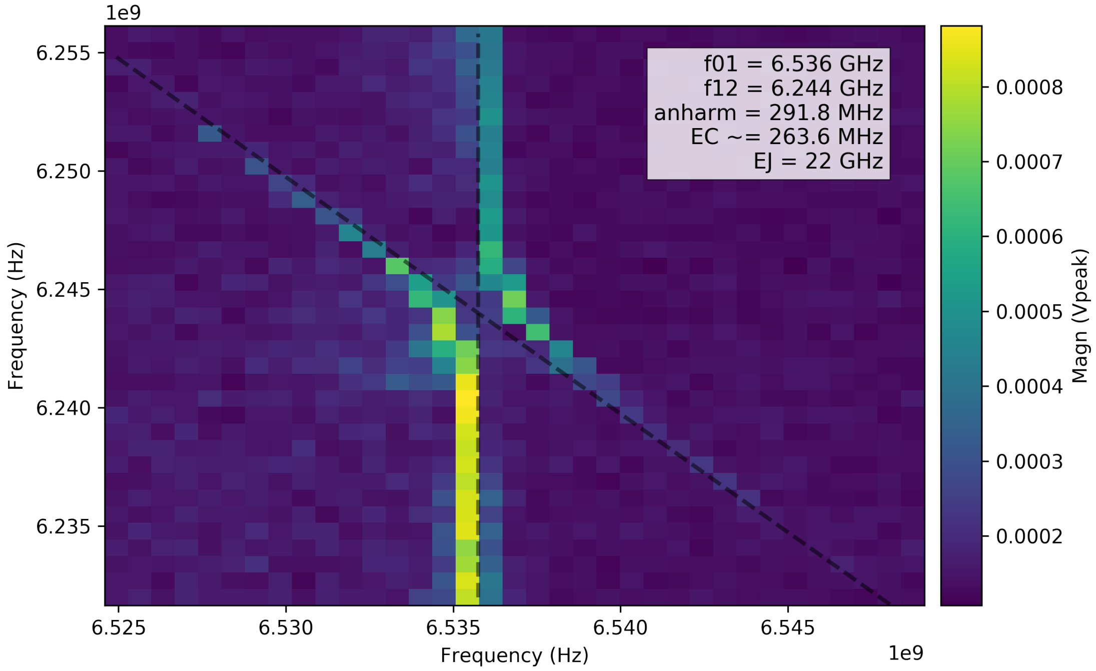


Figure 3.16: A two-tone spectroscopy, showing the 01-transition as a vertical line, and the 02-transition at the constant-sum diagonal of the two tones. The 01-transition occurs simply when the first spectroscopy tone is at the qubit frequency. The 02-transition happens when the two frequencies add up to the 02-transition. At the intersection of the two lines, the first source's frequency corresponds to the 01-transition, and the sum corresponds to the 02-transition, meaning that the second tone's frequency is the frequency of the 12-transition. Substracting the 12 from the 01 yields the anharmonicity, which can be used to calculate E_C and E_J.

From these anharmonicity measurements, two important additional figures can be extracted: E_J and E_C. These can be calculated by optimising the values found for f_{01} and f_{12} from the following Hamiltonian [6]:

$$\hat{H} = 4E_C(\hat{n} - n_g)^2 - E_J \cos\hat{\phi}. \tag{3.3}$$

This formula ignores the coupling to the transmission line via the resonator.

3.5 Time domain measurements, varying frequency

For some of the qubits on some of the chips, the time domain was measured not only at the sweet spot, but also at frequencies up to about a gigahertz below. Because a Transmon qubit is a lot more sensitive to flux noise when it is not at its sweet spot, the T_2 and T_2^* coherence times decrease very rapidly as the frequency is lowered. T_1 does not suffer from this effect. In fact, it tends to increase slightly. The frequency is varied, as before, by applying a current to the most strongly coupled flux line. One method to perform the full measurement is by simply looking for the new qubit frequency every time the current is changed, and then immediately performing the set of coherence measurements. Another option is to choose a set of currents and find all the corresponding frequencies before moving on to measuring coherence times. This is especially useful when doing multiple rounds of measurements, as it is not necessary to measure the frequency every time.

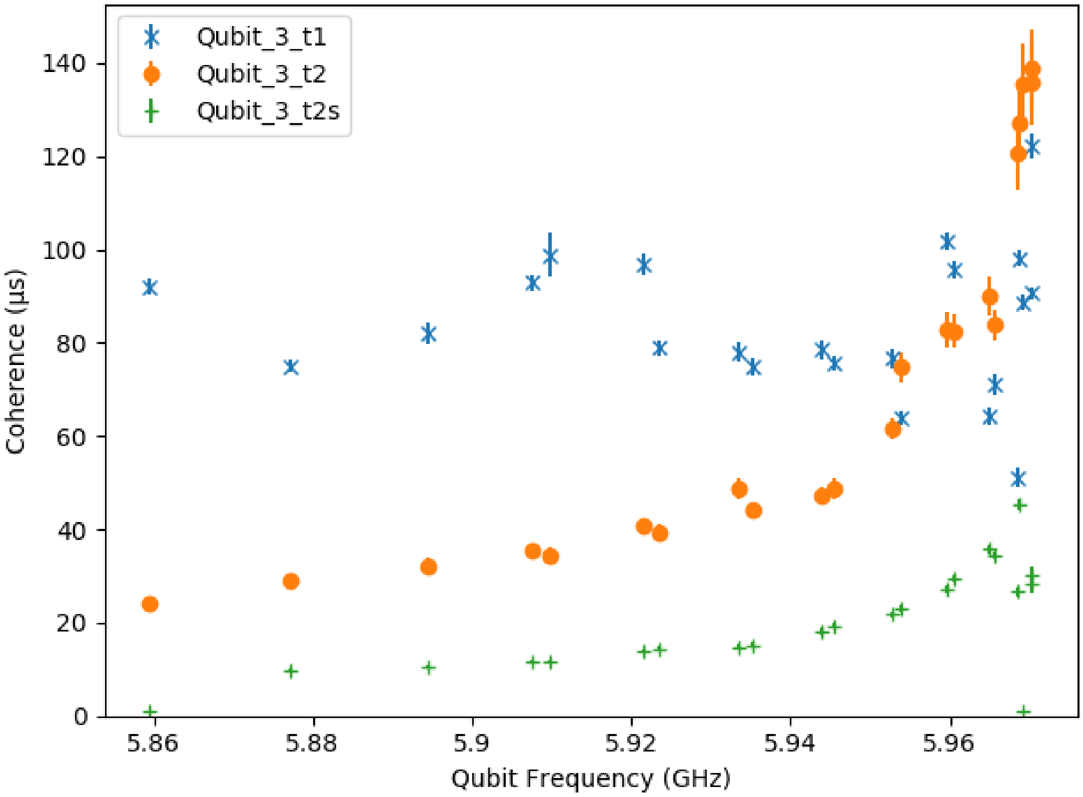


Figure 3.17: This is an example of a set of coherence measurements with varying frequencies on qubit 3. Every point represents a single measurement. This already clearly shows the quick reduction in T_2 when moving away from the sweet spot. T_1 does not vary more than it would in a time trace where the frequency is not varied.

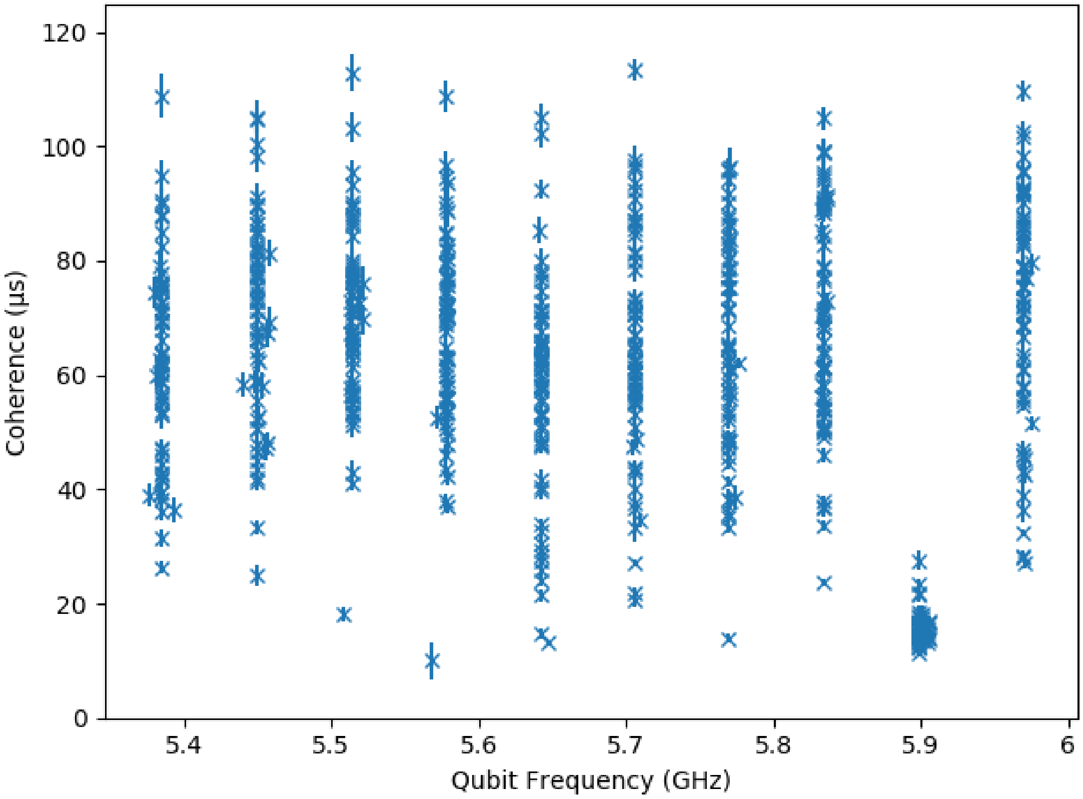


Figure 3.18: Repeated measurements of T_1 for different frequencies, again for qubit 3. The large spread of coherence times for repeated measurements is again evident. Note also that for the frequency of 5.9 GHz, the measured T_1 value is consistently a lot lower, presumably due to an energy leakage source, such as a two level system at this specific frequency. For this sequence, the frequencies were first determined for the different DAC current values. Before each measurement they were quickly measured again, resulting in the slight spread

3.6 Data summary

All the data collected in the measurements explained here, with exception of the time domain vs frequency part, can conveniently be summarised into a single number per measurement type and qubit. As such, it can be neatly arranged into table 3.2. The coherence times for this chip are exceptionally high. At the end of the next chapter, the two most important measurements, qubit frequency and T_1 are visualised in figure 4.1, along with the data of the two other chips presented in the next chapter.

Qubit nr.	Resonator frequency (GHz)	Qubit frequency (GHz)	T_1 (μs)	T_2^* (μs)	T_2 (μs)	Anharmonicity
1	7.116	5.963	90 ± 19	37 ± 6	136 ± 15	325
2	7.219	5.983	72 ± 25	24 ± 5	154 ± 17	315
3	7.320	5.970	85 ± 11	25 ± 5	140 ± 15	321
4	7.423	6.140	71 ± 5	41 ± 3	90 ± 7	308
5	7.530	6.278	78 ± 5	60 ± 10	80 ± 17	314
6	7.636	6.322	70 ± 13	82 ± 10	120 ± 24	316
7	7.739	6.466	77 ± 12	30 ± 5	98 ± 12	306
8	7.836	6.460	71 ± 18	57 ± 10	75 ± 10	314
9	7.937	6.822	44 ± 10	76 ± 10	60 ± 10	294
10	8.033	6.658	70 ± 10	100 ± 10	120 ± 20	312

Table 3.2: Table containing the summary of all data for a chip. All the previously mentioned measurements are included. The values after the ±-signs are standard deviations. The data for qubit 4 is from the second characterisation, described in section 3.7.

3.7 Post-annealing characterisation

After this first characterisation, the chip is warmed up and laser annealed as described in the previous chapter. Then a second characterisation is done. This second process is identical to the first characterisation. Interestingly, it turns out that the qubit frequency is not the only parameter affected by the annealing process. For example, the mean T_1 decreased from 73 to 37, almost a 50% decrease. In one case, this figure went down all the way from 70 to 2.4 μs. Interestingly, one of the qubit's coherence times actually increased significantly, almost by a factor of two. However, the overall trend remains: laser annealing seems to decrease the overall quality of the chips. The full data can be seen in table 3.3

Qubit nr.	Resonator frequency (GHz)	Qubit frequency (GHz)	T_1 (μs)	T_2^* (μs)	T_2 (μs)	Anharmonicity
1 [Z]	7.115	5.913	57 ± 4	37 ± 4	106 ± 10	328
2 [Z]	7.219	5.900	24 ± 2	14.0 ± 0.7	43 ± 5	316
3 [Z]	7.320	5.842	10 ± 2	14 ± 1	21 ± 2	316
4 [C]	7.425	6.129	71 ± 5	41 ± 3	90 ± 7	316
5 [C]	7.530	6.261	23 ± 5	35 ± 6	45 ± 7	310
6 [C]	7.636	6.276	2.4 ± 0.3	3.4 ± 0.8	5.0 ± 0.6	316
7 [Z]	7.739	6.404	61 ± 5	30 ± 5	95 ± 13	308
8 [Z]	7.836	6.377	69 ± 15	36 ± 6	95 ± 18	316
9 [Z]	7.936	6.601	79 ± 8	120 ± 8	111 ± 21	316
10 [Z]	8.034	6.391	9.8 ± 1.0	13.7 ± 0.7	21 ± 1	308

Table 3.3: Table containing the summary of all data for a chip. All the previously mentioned measurements are included. The values after the ±-signs are standard deviations. The [C] indicates a control qubit, the [Z] indicates it was annealed ('zapped')

Chapter 4

Data summary

So far, three Zapmon devices have been characterised. The date for one of these devices has already been presented in the last chapter. The other two devices' parameters are presented here, first for Zapmon 6.2, then for Zapmon 4.2. For visualisation of the qubit frequency and T_1 data, see figure 4.1.

4.1 Zapmon 6.2

Qubit nr.	Resonator frequency (GHz)	Qubit frequency (GHz)	T_1 (μs)	T_2^* (μs)	T_2 (μs)	Anharmonicity
1	7.112	6.031	45 ± 8	35 ± 12	86 ± 24	317
2	7.210	5.907	59 ± 7	44 ± 18	94 ± 14	321
3	7.305	6.145	5.5 ± 0.7	3.2 ± 0.2	10 ± 1	311
4	7.406	6.124	3.7 ± 0.2	1.24 ± 0.04	3.6 ± 0.4	295
5	7.530	6.272	13 ± 1	22 ± 3	25 ± 5	304
6	-	-	-	-	-	-
7	7.715	6.531	6.1 ± 0.4	11.8 ± 0.7	15 ± 2	288
8	7.815	6.195	6.4 ± 0.4	8.7 ± 0.3	12 ± 1	298
9	7.920	6.825	3.6 ± 8	49 ± 12	81 ± 12	295
10	-	-	-	-	-	-

Table 4.1: Table containing the summary of all data for chip 6.2. All measurements from chapter 4 are included, except the time domain vs frequency measurements. The values after the ±-signs are standard deviations. The dashes indicate missing resonators: resonator 6 had a broken air bridge, resonator 10's frequency was in the travelling wave parametric amplifier(TWPA, used to amplify the ouptut signal) band gap.

Qubit nr.	Resonator frequency (GHz)	Qubit frequency (GHz)	T_1 (μs)	T_2^* (μs)	T_2 (μs)	Anharmonicity
1	-	-	-	-	-	-
2 [C]	7.210	5.900	41 ± 6	29 ± 12	79 ± 9	320
3 [Z]	7.305	5.872	8.1 ± 0.4	13.4 ± 0.9	15 ± 1	317
4 [Z]	7.406	5.843	1.9 ± 0.1	0.9 ± 0.1	1.6 ± 0.1	301
5	-	-	-	-	-	-
6 [Z]	7.614	6.171	10.7 ± 0.6	13 ± 2	21 ± 1	311
7 [Z]	7.715	6.286	6.5 ± 0.8	11 ± 1	14 ± 1	292
8 [C]	7.815	6.183	2.3 ± 0.2	4.1 ± 0.2	4.7 ± 0.4	298
9 [Z]	7.920	6.537	6.8 ± 0.3	6.7 ± 0.8	13 ± 1	292
10 [Z]	8.028	6.297	45 ± 10	40 ± 13	80 ± 20	309

Table 4.2: Table containing the summary of all data for chip 6.2. All measurements from chapter 4 are included, except the time domain vs frequency measurements. The values after the ±-signs are standard deviations. The dashes indicate missing qubits: these were damaged during the annealing. The air bridge of qubit 6 was repaired and the TWPA removed, so qubits 6 and 10 are included here. [C] indicates a control qubit, [Z] indicates an annealed (zapped) qubit.

4.2 Zapmon 4.2

Qubit nr.	Resonator frequency (GHz)	Qubit frequency (GHz)	T_1 (μs)	T_2^* (μs)	T_2 (μs)	Anharmonicity
1	7.123	6.223	68 ± 15	27 ± 4	90 ± 20	316.5
2	7.226	6.301	54 ± 12	39 ± 5	97 ± 17	314.2
3	7.326	6.236	49 ± 8	50 ± 10	74 ± 13	320.0
4	7.431	6.438	48 ± 12	55 ± 15	71 ± 22	305.5
5	7.538	6.684	5.0 ± 0.5	9.9 ± 0.9	10.7 ± 0.7	306.2
6	7.643	6.833	41 ± 8	35 ± 10	60 ± 9	298.2
7	7.747	6.826	11 ± 3	5 ± 1	20 ± 6	246.1
8	7.842	6.804	52 ± 6	61 ± 10	80 ± 15	307.0
9	7.943	7.080	47 ± 8	32 ± 3	21 ± 4	299.2
10	8.041	7.088	22 ± 4	22 ± 7	36 ± 10	297.0

Table 4.3: Table containing the summary of all data for chip 4.2. All measurements from chapter 4 are included, except the time domain vs frequency measurements. The values after the ±-signs are standard deviations.

Qubit nr.	Resonator frequency (GHz)	Qubit frequency (GHz)	T_1 (μs)	T_2^* (μs)	T_2 (μs)	Anharmonicity
1 [C]	7.123	6.214	50 ± 11	25 ± 5	89 ± 9	317
2 [-]	7.226	-	-	-	-	-
3 [Z]	7.326	6.007	17 ± 1	7.1 ± 0.6	10.3 ± 0.5	315
4 [Z]	7.431	5.680	62 ± 6	22 ± 5	102 ± 8	321
5 [Z]	7.538	5.789	66 ± 8	23 ± 3	114 ± 12	321
6 [-]	7.643	-	-	-	-	-
7 [Z]	7.747	6.010	49 ± 6	24 ± 2	64 ± 8	319
8 [-]	7.842	-	-	-	-	-
9 [Z]	7.943	5.948	46 ± 9	-	-	316
10 [C]	8.041	7.082	26 ± 8	34 ± 10	45 ± 10	297

Table 4.4: Table containing the summary of all data for chip 4.2. All measurements from chapter 4 are included, except the time domain vs frequency measurements. The values after the ±-signs are standard deviations. [C] indicates a control qubit, [Z] indicates an annealed (zapped) qubit.

4.3 Visualisation

Two of the most important parameters that are influenced by the zapping procedure are the qubit frequency and the T_1 coherence time. The result of the laser annealing can be visualised by displaying the value of the parameter before (after) the annealing on the horizontal (vertical) axis. These figures show that laser annealing always significantly decreases the qubit frequency, and can have varying effects on the T_1, normally decreasing it. Interestingly, the decrease in T_1 seems to be quite similar for the control (non-annealed) and zapped qubits, indicating that perhaps another step in the process is responsible for the change in coherence times.

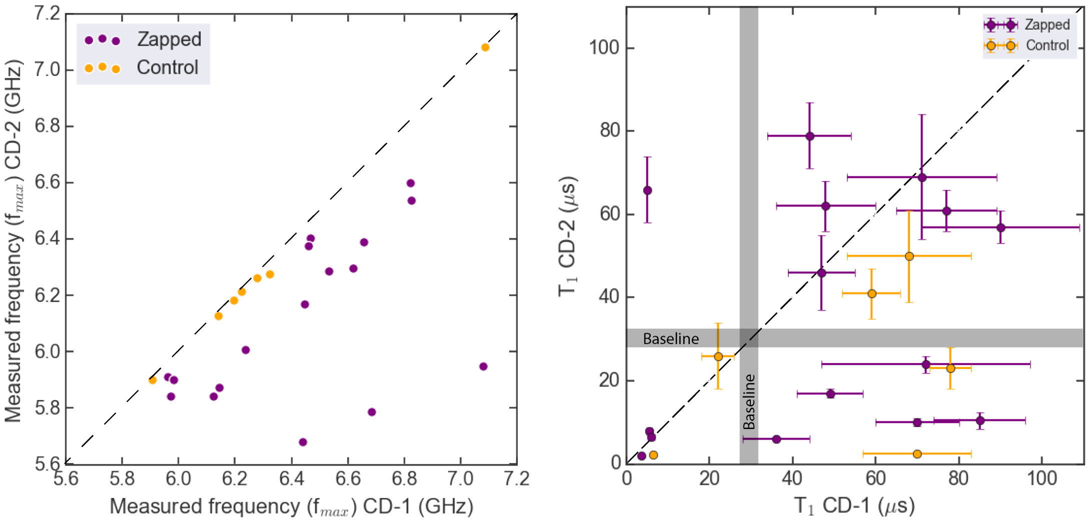


Figure 4.1: In these figures, the values of the frequency (left) and T_1 are compared for the first and second characterisation of all the chips. Qubits that have had their Junctions annealed are shown in purple, the control qubits are in orange.

Conclusion

From the data presented in this thesis, we conclude that laser annealing is a method that can be used to decrease qubit's frequencies after fabrication. However, while the frequency does consistently decrease after the laser process, the actual amount it decreases by varies a lot, even when the laser procedure is done in the exact same way. This limits the techniques applicability. Additionally, the coherence times of the annealed qubits suffer a significant decrease of up to 97% in rare cases. While instances of increased coherence have been measured, this has been quite rare. The average decrease in coherence times is about 50%. However, since the measured chips had very high coherence times during the first characterisation, they are still quite high after the laser procedure. In order to make laser annealing a truly useful technique, several criteria need to be met. Firstly, the decrease in frequency needs to be more predictable, so qubits can be annealed to a specific frequency. Secondly, the impact on other qubit parameters should be as small as possible.

One goal of this project was to decrease the time it takes to characterise a qubit chip. By using the scripts found in the appendix, this time was decreased from two weeks to one week. Further decrease of the characterisation time can be achieved by bridging the gaps between the different steps. During the characterisation of the chips described in this thesis, sweet spots had to be read out by eye, as well as resonator frequencies from the overview and the qubit susceptibility to flux. By creating routines that take these task from the user and perform them automatically, characterisation time might be reduced to just a couple of days.

Bibliography

[1] H. Bernien. *Control, measurement and entanglement of remote quantum spin registers in diamond*. PhD Dissertation, TU Delft, 2013.

[2] J. I. Cirac and P. Zoller. Quantum computations with cold trapped ions. *Phys. Rev. Lett.*, 74:4091, 1995.

[3] L. M. K. Vandersypen and I. L. Chuang. NMR techniques for quantum control and computation. *Rev. Mod. Phys.*, 76:1037–1069, Jan 2005.

[4] Guilherme Tosi, Fahd A. Mohiyaddin, Vivien Schmitt, Stefanie Tenberg, Rajib Rahman, Gerhard Klimeck, and Andrea Morello. Silicon quantum processor with robust long-distance qubit couplings. *Nat Commun*, 8:450, Sep 2017. 378[PII].

[5] Alexandre Blais, Ren-Shou Huang, Andreas Wallraff, S. M. Girvin, and R. J. Schoelkopf. Cavity quantum electrodynamics for superconducting electrical circuits: An architecture for quantum computation. *Phys. Rev. A*, 69:062320, Jun 2004.

[6] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the Cooper pair box. *Phys. Rev. A*, 76:042319, 2007.

[7] Austin G. Fowler, Ashley M. Stephens, and Peter Groszkowski. High-threshold universal quantum computation on the surface code. *Phys. Rev. A*, 80:052312, Nov 2009.

[8] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073, 1996.

[9] David P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48:771, 2000.

[10] L. DiCarlo, M. D. Reed, L. Sun, B. R. Johnson, J. M. Chow, J. M. Gambetta, L. Frunzio, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf. Preparation and measurement of three-qubit entanglement in a superconducting circuit. *Nature*, 467:574, 2010.

[11] S. Asaad. Exploring frequency re-use with transmon qubits in a cQED architecture. MEP report, TU Delft, 2015.

Acknowledgements

Throughout the work I did for this thesis, many different people of the DiCarlo lab were ready to help me when I needed them. I would like to thank each of these people, for various contributions to the establishment of this thesis. Firstly I would like to thank Filip Malinowski. During my time at the DiCarlo lab, he has been my daily supervisor. He showed me how to characterise chips, how to work with the fridges, how to debug when things were not going the way they were supposed to, and much more. Filip, I hope I was not too annoying during this project, thank you for all the time you invested in me. Apart from Filip, many others have helped me with the experiments. Leo, thank you for having me in your lab, and for applying just the right pressure for optimal results. Adriaan, who was often working near me, thank you for helping me when I struggled with the code, and for helping implement the new echo measurement sequence. Nandini, thank you for providing all the different chips that are part of this thesis, and for doing the annealing and room temperature measurements. Gerco and Slava, thank you for getting my measurement computer working, I do not know how I would have managed without you. Jacob, thank you for all the different technical support, especially with getting my measurement setup ready. Many thanks to Florian and, again, Filip, for helping write this thesis by having a critical look at my first version. Without the help of these people, my thesis would never have been what it is. Thank you all again, I hope this thesis will be of use to you in the future!

Appendix A

Code and explanation

This appendix follows the measurements as presented in chapter 3: chip characterisation. For more explanation about the goal and results of these measurements, see the main text.

A.1 Resonator characterisation

The code below is used to search for resonators. With this combination of number of points and bandwidth, the measurement takes a few minutes. Power -40 is chosen as it is almost always in the dispersive regime of the resonators. Basic 1D analysis does not extract the resonator frequencies automatically. They will have to be read by the user, from a figure like 3.1

```
# Basic resonator search

VNA.start_frequency(7.000e9)
VNA.stop_frequency(8.200e9)

VNA.power(-40)

VNA.bandwidth(200)
VNA.timeout(200)
VNA.npts(24001)

swf_fct_1D = swf.ZNB_VNA_sweep(VNA, npts=VNA.npts())
swf_fct_1D.prepare()

MC.set_sweep_function(swf_fct_1D)
MC.set_sweep_points(swf_fct_1D.sweep_points)
MC.set_detector_function(det.ZNB_VNA_detector(VNA))
MC.run("Zapmon_resonator_search")
ma2.Basic1DAnalysis()
```

Next up is the zoom-in for each resonator. This looks very similar to the previous measurement, only with reduced range. One should input the approximate resonator frequencies from the previous measurement in the third line. An example of the analysis of a measurement like this can be seen in figure 3.2

```
#Zoom in on each resonator

for freq_guess in [7106e6,7203e6,7302e6,7399e6,7500e6,7592e6,7708e6,7811e6,7913e6,8021e6]:
    t_start_meas = ma.a_tools.current_timestamp()
```

```
        VNA.start_frequency(freq_guess-20e6)
        VNA.stop_frequency(freq_guess+20e6)

        VNA.power(-40)

        VNA.bandwidth(200)
        VNA.timeout(200)
        VNA.npts(601)

        swf_fct_1D = swf.ZNB_VNA_sweep(VNA, npts=VNA.npts())
        swf_fct_1D.prepare()

        MC.set_sweep_function(swf_fct_1D)
        MC.set_sweep_points(swf_fct_1D.sweep_points)
        MC.set_detector_function(det.ZNB_VNA_detector(VNA))
        MC.run("Zapmon_resonator_characterisation_"+str(freq_guess/1e9))
        t_stop_meas = ma.a_tools.current_timestamp()
        ma2.Basic1DAnalysis()
        t_start = ma.a_tools.get_timestamps_in_range(t_start_meas, t_stop_meas, label = 'Z')[0]
        ma2.VNA_analysis(t_start = t_start, options_dict = {'fit_options':{'model': 'complex'}})
```

After this, the resonator power sweeps can be performed. These yield images as seen in figure 3.3. Since the previous measurement yields precise values for the resonators, these can be put into the list in the third line.

```
1  #Qubit resonator power sweeps
2
3  for freq_res in [7106e6,7203e6,7302e6,7399e6,7500e6,7592e6,7708e6,7811e6,7913e6,8021e6]:
4          VNA.start_frequency(freq_res-2e6)
5          VNA.stop_frequency(freq_res+2e6)
6
7          VNA.bandwidth(100)
8          VNA.timeout(600)
9          VNA.npts(201)
10
11         swf_fct_1D = swf.ZNB_VNA_sweep(VNA, npts=VNA.npts())
12         MC.set_sweep_function(swf_fct_1D)
13         MC.set_sweep_function_2D(VNA.power)
14
15         swf_fct_1D.prepare()
16         MC.set_sweep_points(swf_fct_1D.sweep_points)
17         MC.set_sweep_points_2D(np.arange(-50, 10,2.5))
18
19         MC.set_detector_function(det.ZNB_VNA_detector(VNA))
20
21         MC.run("Zapmon_power_sweep_"+str(freq_res/1e9),mode='2D')
22         ma.TwoD_Analysis()
```

For all resonators that react to power variation, a flux sweep is to be performed. Examples of results of these measurements can be found in figures 3.4 and 3.5. If, like in the aforementioned figures, only a small part of the horizontal range is actually used, the frequency range and number of steps could be reduced, decreasing the measurement time.

```
# Qubit resonators flux sweeps (all resonator-line pairs, full current range)

for freq_res in [7106e6,7203e6,7302e6,7399e6,7500e6,7592e6,7708e6,7811e6,7913e6,8021e6]:
        VNA.power(-40)
        VNA.start_frequency(freq_res-5e6)
        VNA.stop_frequency(freq_res+2e6)
        for FBL_i in ['FBL_2','FBL_3','FBL_5','FBL_6','FBL_8','FBL_10']:
                VNA.bandwidth(500)
                VNA.timeout(600)
                VNA.npts(201)

                swf_fct_1D = swf.ZNB_VNA_sweep(VNA, npts=VNA.npts())
                MC.set_sweep_function(swf_fct_1D)
                MC.set_sweep_function_2D(fluxcurrent[FBL_i])

                swf_fct_1D.prepare()
                MC.set_sweep_points(swf_fct_1D.sweep_points)
                MC.set_sweep_points_2D(np.linspace(-40e-3,40e-3,21))

                MC.set_detector_function(det.ZNB_VNA_detector(VNA))

                MC.run("Zapmon_flux_sweep_"+str(freq_res/1e9)+'_'+FBL_i,mode='2D')
                ma.TwoD_Analysis()

                set_currents_zero()
```

This concludes the resonator characterisation.

A.2 Qubit characterisation

The first step in qubit characterisation is to input the relevant data from the room temperature measurements and the resonator characterisation. All resonator frequencies are noted, as well as their most strongly coupled flux lines. The sweet spot predictions from the flux sweeps are also useful, as are the qubit frequency predictions from the room temperature measurements. The relevant information is put into the qubit objects.

```
# Implementing parameters from room temperature and VNA measurements

# Resonator frequencies (from VNA)
res_freq_list =[7106e6,7203e6,7302e6,7399e6,7500e6,7592e6,7708e6,7811e6,7913e6
    ,8021e6]

# List of all qubit objects
all_qubits = [Qubit_1,Qubit_2,Qubit_3,Qubit_4,Qubit_5,
Qubit_6,Qubit_7,Qubit_8,Qubit_9,Qubit_10]

# Dictionary of most strongly coupled flux line for each qubit. See VNA DAC sweeps
fluxline_dict = {Qubit_1:'FBL_2',
Qubit_2:'FBL_2',Qubit_3:'FBL_3',
Qubit_4:'FBL_3',Qubit_5:'FBL_5',Qubit_6:'FBL_6',
Qubit_7:'FBL_6',Qubit_8:'FBL_8',
Qubit_9:'FBL_2',Qubit_10:'FBL_10'}

# Predictions of sweetspot current on fluxline defined above (from VNA DAC sweeps)
sweetspot_predictions_dict = {Qubit_1:0,Qubit_2:0,Qubit_3:-7e-3,Qubit_4:-2e-3,
    Qubit_5:0,
Qubit_6:-2e-3,Qubit_7:2e-3,Qubit_8:0,Qubit_9:0,Qubit_10:0}

# Qubit frequency predictions (from room temperature measurements)
freq_qubit_predictions = {Qubit_1:5.852,Qubit_2:5.504,Qubit_3:5.694,Qubit_4
    :5.865,Qubit_5:5.981,
Qubit_6:6.124,Qubit_7:6.206,Qubit_8:6.274,Qubit_9:6.238,Qubit_10:6.431}

```

```
# Set relevant parameters to the qubit objects
for i in np.arange(10):
        all_qubits[i].freq_res(res_freq_list[i])
        all_qubits[i].ro_freq(res_freq_list[i])
```

Next up the resonator frequencies are again determined, to check if at the current power, the number of photons in the resonator is below the critical photon number, and thus shifted by the qubit. Since the resonator frequency has already been added to the qubit object, there is no need to specify a range to search in; the object will choose that by itself based on the given resonator frequency. An example of a measurement like below can be found in figure 3.6.

```
# Finding resonators

for Q in all_qubits:
        Q.ro_pulse_amp_CW(.3)
        Q.find_resonator_frequency()
```

The next step is a set of spectroscopy measurements to find all the qubits. This set of measurements can take anywhere between an hour and a full night, depending on how precise the frequency predictions are expected to be. The most reliable way of finding all qubits in one measurement series is to vary the spectroscopy power over quite a large range, and to take a generous estimate of the maximum frequency deviation of room temperature measurement to actual value. An extreme example is given below. This measurement took a full night to complete. For most chips, the ranges of the power and frequency can be a lot smaller, resulting in a much quicker measurement series. For the result of one of the measurements performed with the code below, see figure 3.7.

```
# Finding qubit frequencies at zero current

set_currents_zero()
for pwr in np.arange(-30,10.1,5):
        for i,Q in enumerate(all_qubits):
                Q.spec_pow(pwr)
                Q.find_frequency(freqs = np.linspace(-400e6,400e6,1601)+(1e9*
                    freq_qubit_predictions[Q]))
```

From this series of measurements, an ideal spectroscopy power can be chosen for each qubit. This is important to note down for later measurements. Thus a quick dictionary is made. As spectroscopy power increases, a qubit peak first gets higher and higher, up to a certain point, after which it will only get wider and wider. The ideal spectroscopy power yields a peak of maximum height with minimum width.

```
# Using the above measurement, set the right spectroscopy power for each Qubit

spec_pow_dict = {Qubit_1: -22,Qubit_2: -20,Qubit_3: -30,Qubit_4: -25,Qubit_5: -30,
    Qubit_6: -35,Qubit_7: -35,Qubit_8: -25,Qubit_9: -30,Qubit_10: -25}
```

Now that all qubits have been found and a good spectroscopy power is noted for each, the sweet spot can be determined. The previously defined dictionaries are used to choose the spectroscopy power, flux line and centre of the flux range.

```
# Make qubit DAC arcs based on resonator DAC arcs

for Q in all_qubits:
        Q.spec_pow(spec_pow_dict[Q])
        set_currents_zero()
        Q.measure_qubit_frequency_dac_scan(Q.freq_qubit()+np.linspace(−30e6,20e6
            ,101), sweetspot_predictions_dict[Q]+np.linspace(−5e−3,5e−3,101),
            fluxChan = fluxline_dict[Q])
        set_currents_zero()
```

The results found here are not stored in the qubit object automatically. Manually, it can be done as below.

```
# Set sweepspots in Qubit objects

Qubit_6.fl_dc_V0(−2e−4)
```

Having found the qubit sweet spot, it is now time to accurately determine the frequency. To do so, the qubit is set to its sweet spot, the microwave pulses are prepared for time domain measurements, and a Rabi calibration is performed. The frequency is then finely determined with the Ramsey method. The retrieved frequency is set as the maximum frequency of the qubit, and another Rabi calibration is performed, now at the more accurately determined frequency. If, during this step and in general, a Rabi calibration does not yield a full oscillation, increasing the width of the Gaussian envelope of the microwave pulse using Q.mw_gauss_width is normally a good solution.

```
#Set sweetspot frequency with high accuracy

for Q in all_qubits:
        set_currents_zero()
        fluxcurrent[fluxline_dict[Q]](Q.fl_dc_V0())
        Q.find_frequency()
        Q.prepare_for_timedomain()
        MW_lutman.load_waveforms_onto_AWG_lookuptable(force_load_sequencer_program
            =True)
        Q.calibrate_mw_pulse_amplitude_coarse()
        Q.find_frequency(method = 'ramsey', steps = [1,3,10,30,100])
        Q.freq_max(Q.freq_qubit())
        Q.calibrate_mw_pulse_amplitude_coarse()
set_currents_zero()
```

Next up is the preparation of the qubits for a long time domain characterisation. Since the qubit frequency is already very precisely known, all that needs to happen is a Rabi calibration, and a first measurement of all coherence times to determine the range to measure in during the repeated measurements.

```
#Preparation for full overnight coherence time measurement

for Q in all_qubits:
        Q.freq_qubit(Q.freq_max())
        set_currents_zero()
        fluxcurrent[fluxline_dict[Q]](Q.fl_dc_V0())
        Q.prepare_for_timedomain()
        MW_lutman.load_waveforms_onto_AWG_lookuptable(force_load_sequencer_program
            =True)
        Q.calibrate_mw_pulse_amplitude_coarse()
        Q.measure_T1()
        Q.measure_ramsey()
        Q.measure_echo()
```

With this preparation done, the coherence times of each qubit can be determined. This is usually done overnight, since there is no need for human interference during the measurements. Each round of the three measurements for a single qubit takes about 2 minutes and 20 seconds.

For ten qubits, this means that a single data point is acquired for each qubit and each coherence time every 23 minutes. This yields about 40 data points in an overnight measurement, a reasonable number of measurements to base a value and standard deviation on. Sometimes, a measurement does fail. When this happens, the update = False statement ensures that the measurement series simply continues. The long calculations in the specified time range ensure that the step size is valid.

```
#Full overnight time domain measurement

while True:
    for Q in all_qubits:
        set_currents_zero()
        fluxcurrent[fluxline_dict[Q]](Q.fl_dc_V0())
        Q.prepare_for_timedomain()
        MW_lutman.load_waveforms_onto_AWG_lookuptable(force_load_sequencer_program = True)
        Q.measure_T1(times = np.linspace(0,4*Q.T1(),31),update=False)
        Q.measure_ramsey(times = np.linspace(40e-9*((4/50*Q.T2_star())//40e-9),4*40e-9*(Q.T2_star()//40e-9),50),update=False)
        Q.measure_echo(times = np.linspace(40e-9*int(((4/50*Q.T2_echo())//40e-9)),40e-9*int(((4/50*Q.T2_echo())//40e-9))*50,50),update=False)
```

Time domain measurements with varying frequency can be done in two ways, as explained in the main text. Below the code is shown for a measurement where new frequencies are determined as the current on a flux line is varied.

```
Q = Qubit_1
Q.prepare_for_timedomain()
MW_lutman.load_waveforms_onto_AWG_lookuptable(force_load_sequencer_program=True)
for s in [1,-1]: # Both positive and negative currents are examined
    set_currents_zero()
    fluxcurrent[fluxline_dict[Q]](Q.fl_dc_V0())
    Q.find_frequency(freqs=np.linspace(-30e6,10e6,81)+Q.freq_max())
    Q.calibrate_mw_pulse_amplitude_coarse()
    Q.find_frequency(method = 'ramsey',steps = [1,3,6,10])
    for n,cur in enumerate(np.linspace(0,8e-3,17)*s+fluxcurrent[fluxline_dict[Q]]()): # Sweep around the sweetspot
        fluxcurrent[fluxline_dict[Q]](cur) # Set the current
        if n%5==0:
            Q.find_resonator_frequency() # The resonator can shift with current, this is used to compensate
        Q.find_frequency(freqs=np.linspace(-30e6,10e6,81)+Q.freq_qubit())
        Q.find_frequency(method = 'ramsey',steps = [1,3,6,10])
        Q.calibrate_mw_pulse_amplitude_coarse()
        Q.measure_T1(times = np.linspace(0,180e-6,31),update = False)
        if abs(cur) < 3e-3: # T2 measurements are not very interesting at high bias
            Q.measure_ramsey(times = np.linspace(36e-7,18e-5,50),update = False)
            Q.measure_echo(times = np.linspace(36e-7,18e-5,50),update = False)
```

The other method described in the main text is shown below. First, the qubit frequency is determined for a range of current values on a flux line. Then these frequencies are used to significantly speed up the process of finding the coherence times.

```
# setting the frequencies
Q = Qubit_2

```

```
bias = np.sqrt(np.linspace(0,(5e-3)**2,11))
for i,b in enumerate(bias):
        fluxcurrent[fluxline_dict[Q]](Q.fl_dc_V0()+b)
        Q.find_frequency(freqs = np.linspace(-10e6,10e6,41)+Q.freq_max()-60e6*i)

freq_Qubit_2 = [5.983e9,5.920e9,5.855e9,5.790e9,5.724e9,5.658e9,5.591e9,5.524e9
    ,5.456e9,5.387e9]

# measuring coherence times
Q.prepare_for_timedomain()
MW_lutman.load_waveforms_onto_AWG_lookuptable(force_load_sequencer_program=True)
set_currents_zero()
fluxcurrent[fluxline_dict[Q]](Q.fl_dc_V0())
Q.find_frequency(freqs = np.linspace(-1e7,1e7,41)+Q.freq_max())
for n,b in enumerate(bias):
        fluxcurrent[fluxline_dict[Q]](Q.fl_dc_V0()+b)
        Q.find_frequency(freqs = np.linspace(-1e7,1e7,41)+freq_Qubit_2[n])
        Q.calibrate_mw_pulse_amplitude_coarse()
        Q.measure_T1(times = np.linspace(0,90e-6,31,update=False)
        Q.measure_ramsey(times = np.linspace(36e-7,18e-5,50),update = False)
        Q.measure_echo(times = np.linspace(36e-7,18e-5,50),update = False)
```

Anharmonicity measurements can also be done, in principle right after finding the qubit sweet spot. Two methods can be used as described in the main text. First, there is the high power spectroscopy measurement which will show the 02 peak. An increase in power of about 30 dB should suffice to see the new peak. In Transmon qubits the peak is usually between 120 and 170 MHz below the qubit frequency. It can be quite narrow, which is why a high resolution is used. The data is put into a dictionary so it can be easily accessed when doing the next measurement.

```
# High-power spectroscopy

for Q in all_qubits:
        set_currents_zero()
        fluxcurrent[fluxline_dict[Q]](Q.fl_dc_V0())
        Q.spec_pow(min([spec_pow_dict[Q]+30,20]))
        Q.find_frequency(freqs = np.arange(-120e6,-170e6,-.125e6)+Q.freq_max(),
            update=False)
        Q.spec_pow(spec_pow_dict[Q])

# Input the frequencies obtained from the above measurement

f_02_dict = {Qubit_1:5.643e9, Qubit_2:5.277e9, Qubit_3:5.49e9, Qubit_4:5.663e9,
    Qubit_5:5.835e9, Qubit_6:5.928e9, Qubit_7:5.995e9, Qubit_8:6.103e9, Qubit_9
    :6.017e9, Qubit_10:6.208e9}
```

Next up, the two-tone method discussed in the main text can be used. By using the previously found 02-frequency, a good estimate of the 12-frequency can be made by simply doubling the difference with the 01 transition. Putting this into the measure_anharmonicity usually shows the cross seen in 3.16.

```
for Q in all_qubits:
        set_currents_zero()
        fluxcurrent[fluxline_dict[Q]](Q.fl_dc_V0())
        Q.measure_anharmonicity(f_01=Q.freq_max(),f_02=f_02_dict[Q],f_01_power=
            spec_pow_dict[Q], f_12_power=min([spec_pow_dict[Q]+30,20]),
            spec_source_2=spec_source_2, f_01_span = 24e6,f_12_span = 24e6)
```