



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

SEMESTER THESIS

Design and Development of a High Performance Signal Processing Platform for Qubit Readout

Laboratory for Solid State Physics
ETH Zürich, Switzerland

Author:

Deniz Bozyigit

Carried out at:

Quantum Device Lab

Professor: Prof. Dr. Andreas Wallraff

Zürich, December 2008

Abstract

As shown in many measurements and publications in the Quantum Device Lab at ETH Zurich and elsewhere, devices based on superconducting Josephson Junctions are excellent candidates for qubits in a quantum information processing (QIP) architecture. To perform actual QIP conventional measurement methods based on averaging over thousands of experiment repetitions need to be replaced by a so-called *single shot* readout. This type of readout determines the qubit state for one single realization of the experiment.

In this semester thesis we developed a FPGA-based (*Field Programmable Gate Array*) platform to perform these kind of measurements. The design of the platform was driven by two main concerns. Firstly, short coherence times ($\sim 1\mu s$) require a minimal measurement and decision time ($\ll 1\mu s$). Secondly, the high noise levels have to be accounted for by optimal signal processing of the measurement data. As a demonstration of the functionality we present the implementation of an averaging measurement, which can easily be extended to more complex algorithms.

Furthermore we discuss the theoretical aspects of optimal readout schemes, based on the technical capabilities of the new measurement platform. Two schemes are proposed and analyzed in their capability to extract maximum information from the qubit.

Contents

Abstract	i
1 Introduction and History	1
2 Physics Primer	3
2.1 Quantum description of the circuit	3
2.2 Dispersive Limit	4
3 Goal	6
3.1 Restrictions	7
4 Theoretical Considerations	8
4.1 System Model	8
4.2 Qubit Lifetime	9
5 Estimators	10
5.1 Bayesian Estimator	10
5.2 Digital Modulation Theory	11
5.2.1 One Tone Measurement Scheme	12
5.2.2 Two Tone Measurement Scheme	14
6 Results	16
6.1 Simulation	16
6.2 One Tone Detection Algorithm	17
6.3 Two Tone Scheme Simulations	19
6.3.1 Test Measurements on generating a two tone signal	20
7 Hardware Implementation	22
7.1 Platform	22
7.1.1 Specifications	22
7.1.2 Hardware Design Flow - Design Time	24
7.1.3 System Architecture - Run Time	24
7.2 Practical Issues	25
7.2.1 RAM Access	25
7.2.2 Boot Loader	26

Contents

- 7.3 Example - The Averager 27
 - 7.3.1 Simulation 28
 - 7.3.2 Experiment 28
- 8 Acknowledgments** **31**
- Bibliography** **I**

1 Introduction and History

The development in the understanding of quantum mechanics together with new possibilities in material control, cooling, optics, and high performance computing opened up a new research area, called quantum information processing (QIP).

The idea behind QIP can be stated shortly, like Richard Feynman did for the probably first time. Trying to simulate the behavior of quantum systems one can observe, that this problem becomes exponential hard when taking larger systems into consideration. Thus it cannot be treated efficiently on classical computers. Thinking the other way around one could suppose now, that quantum systems must have inherently huge computing capabilities. Exactly this was confirmed by Shor in 1995 [1] finding an algorithm for prime factorization which is fundamentally different and faster than any classical algorithm.

Motivated by the promise of unlimited computing power the research community started the investigation for the principal elements of quantum computation. In the center of the efforts they put the so called qubit - an abstract model of a quantum system in analogy to the classical bit. It can be seen as the simplest kind of "memory" in a quantum computer and helps to formulate consistently arbitrary computations.

The aspiration today lies in the implementation of a device which behaves like a qubit¹. The different present technologies led to a variety of approaches, like ion traps, optical or superconducting devices.

Especially the latter are regarded to be suitable to make quantum computing reality. Their advantage is good scalability - easy and cheap production of many thousand units. This is one of the primary issues for any applicable implementation, since complexity of real world problems and the need for internal error correction might boost the required number qubits to the order of millions.

Today there exist several basic implementations of superconducting qubits which yielded promising results. All of the required criteria have been met in some way albeit not in the same implementations. Improving performance and pushing restrictions are the main issues in the experimental part of research and present an enormous technological challenge.

One of the necessary criteria for QIP devices is a viable single shot readout. This is a measurement method which allows to determine the state of the qubit in a single realization of the experiment. This is in contrary to methods which rely on extensive averaging over repetitions of the experiment. In the following I present as the result of my semester thesis our approach to perform single shot readout on the superconducting qubits in the Quantum Device Lab at ETH

¹This is stated more specifically in the set of the 5+2 DiVincenzo criteria.

1 Introduction and History

Zurich.

2 Physics Primer

For the investigation of the interaction of light and matter one often considers an atom coupled to an optical cavity, which is described theoretically by the theory of cavity quantum electrodynamics. The electrical circuits which are considered in this thesis behaves in very similar ways although they have little in common in terms of their physical representation. Using electrical circuit elements such as capacitors, inductors and Josephson Junctions made from superconducting materials one can create artificial atoms and cavities and study their interaction, which is known as cavity QED. This becomes possible at very low temperatures, where superconductors exhibit coherent electronic states on mesoscopic length scales.

The system under consideration consists of two main parts. In the center a piece of a coplanar waveguide made from niobium forms a high quality resonator (Fig. (2.1)). This resonator is coupled capacitively to an input and an output line to control and measure the system. Further, a cooper pair box is coupled to this resonator and introduces a non-linear part to the system. This setup is somewhat equivalent to the systems used in cavity quantum electrodynamics, where one conventionally considers a cavity (here resonator) whose photons interact with an atom (here cooper pair box).

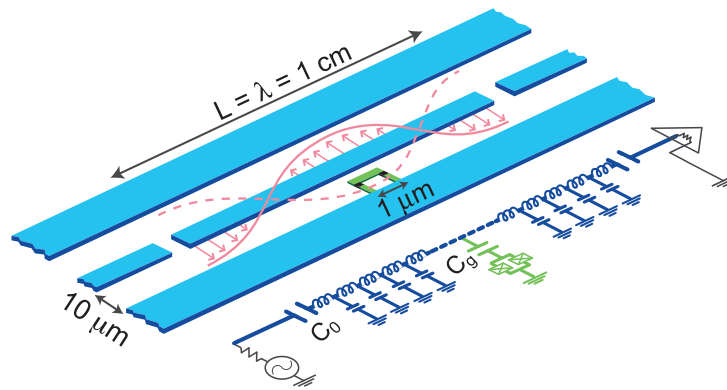


Figure 2.1: Schematic of the circuit quantum electrodynamics setup. A cooper pair box (green) is embedded into a superconducting coplanar waveguide resonator (blue).

2.1 Quantum description of the circuit

To be able to study the quantum mechanical behavior of the circuit QED system one has minimize the loss and the thermal excitations in the system. Therefore the circuits are cooled down to

20mK in a dilution refrigerator. At these temperatures the niobium becomes superconducting reducing the loss in the system and thermal excitations become so weak, that the electrical resonator will be found in its quantum mechanical ground state.

In this regime the resonator and the cooper pair box are then correctly described by discrete energy spectra. The resonator energy is given by the Hamiltonian

$$\mathbb{H}_{res} = \hbar\omega_r(a^\dagger a + \frac{1}{2}) \quad (2.1)$$

where a is the annihilation operator for photons in the resonator, ω_r the resonator frequency. The cooper pair box can be approximated as a two level system with a ground state and an excited state which has $\hbar\omega_a$ more energy. The Hamiltonian of the cooper pair box is then

$$\mathbb{H}_{atom} = \frac{\hbar\omega_a}{2}\sigma_z \quad (2.2)$$

σ_z being the Pauli z-matrix. Finally the coupling between the two systems is given by the term

$$\mathbb{H}_{int} = \hbar g(a^\dagger \sigma^- + \sigma^+ a) \quad (2.3)$$

which describes how an excitation in the qubit can be transformed into a photon and vice versa with a rate given by the coupling constant g .

Putting these three terms together one ends up with the so-called Jaynes-Cummings Hamiltonian which is more thoroughly discussed in [2].

$$\mathbb{H} = \hbar\omega_r(a^\dagger a + \frac{1}{2}) + \frac{\hbar\omega_a}{2}\sigma_z + \hbar g(a^\dagger \sigma^- + \sigma^+ a) \quad (2.4)$$

2.2 Dispersive Limit

This description can be simplified in the case of the *dispersive limit* where the two characteristic frequencies are very different, that is $\Delta = \omega_a - \omega_r > g$. One can introduce then the interaction as a second order perturbation [3] and find

$$\mathbb{H}_{eff} = \frac{\hbar}{2} \left(\omega_a + \frac{g^2}{\Delta} \right) \sigma_z + \hbar \left(\omega_r + \frac{g^2}{\Delta} \sigma_z \right) a^\dagger a. \quad (2.5)$$

From the second term we can see now in comparison with Eq. 2.1 that the system effectively behaves like a resonator, whose frequency depends on the qubit state measured by σ_z . If we write the state of the qubit as $q \in 0, 1$ the resonance frequency is

$$\omega'_r = \omega_r(2q - 1)g^2/\Delta. \quad (2.6)$$

This qubit dependent frequency shift will provide the basis for the readout schemes we will consider.

A complete summary of the physical theory and the experimental setup used in the quantum device lab at ETH Zurich can be found in [3].

3 Goal

To perform quantum information algorithms in these kind of systems one has to be able to prepare, manipulate and read out states of one or more qubits. In the current experiments this is done by measuring the shift in the resonance frequency as shown in the last section. Since the detected signal is small and significant noise is present in the measurement system one has to repeat the experiment and average the outcome, so that one finds the expectation value of the qubit state.

For the implementation of an algorithm this is not sufficient, where it is necessary to determine the qubit state for a single realization of an experiment. This type of *single shot read out* is not implemented in the current setup and therefore the goal of this semester thesis was the specification and possible implementation of a signal processing platform to be used for this type of readout.

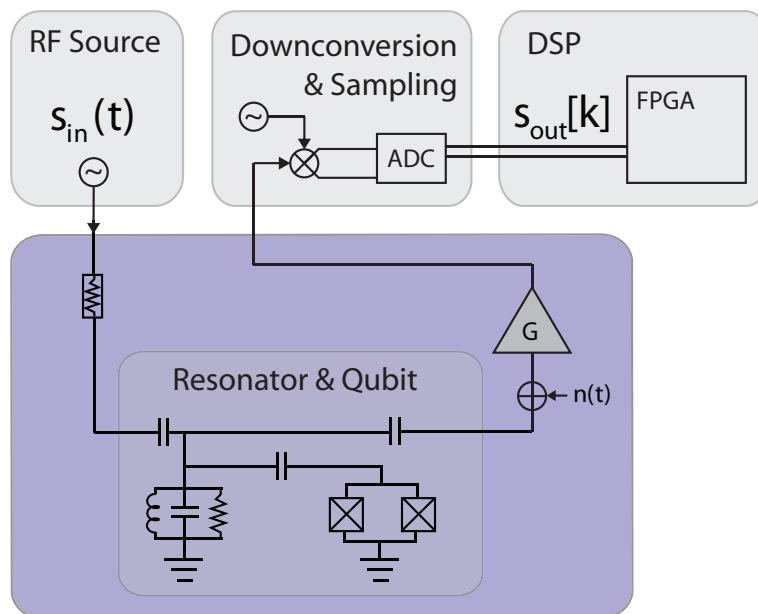


Figure 3.1: Schematics of the signal processing aspects of the qubit experiment.

For an overview of the readout process we consider Fig. (3.1) which shows a simplified schematic of the experiment setup. The goal is to read out the state of the qubit $q \in \{0, 1\}$, where $q = 0$ corresponds to a measurement where the qubit is in the ground state and $q = 1$ where it is in the excited state. For the measurement a radio frequency (RF) signal $s_{in}(t)$ is applied to the resonator/qubit system. This RF signal interacts with the qubit and the system responds with a signal which allows to identify the qubit state. After amplification, down conversion and sampling

the signal $s_{out}[n]$ is analyzed by a digital signal processing (DSP) device which estimates q as $\hat{q}(s_{out})$.

The task which has to be solved to create a working readout scheme is therefore:

- 1. Design a measurement signal $s_{in}(t)$. Find a signal form which extracts maximum information about the qubit state. Find a trade off between measurement time and power.
- 2. Design a signal processing algorithm which estimates $\hat{q}(s_{out})$. Find a design which is implementable in hardware. Determine estimator parameters which minimize error probability.

Neither of these tasks are trivial due to a set of restrictions which apply to the experiment setup. These ultimately restrict the maximal signal to noise ratio (SNR) in the analyzed signal s_{out} and as such our ability to estimate the measured qubit state q .

3.1 Restrictions

The first restriction we introduce demands that the readout should not to alter the qubit state. This is the assumption of a *quantum non-demolition measurement*. Therefore the readout signal should not introduce excitation or relaxation of the qubit which would falsify the measurement. It was found [2] that therefore the power P_{meas} of the measurement signal s_{in} is bound by a maximum power P_{max} .

$$P_{meas} < P_{max} \quad (3.1)$$

Furthermore the qubit in the excited state is subjected to spontaneous decay. This decay is characterized by a typical time T_1 after which the excited state is relaxed to the ground state. Thus the time T during which the qubit is measured can not exceed this time significantly, since in most experiments no additional information can be obtained.

$$T \lesssim T_1 \quad (3.2)$$

Finally the noise power in the measured signal is dominated by the first amplifier in the signal processing chain. In our case this is an high-mobility electron transistor (HEMT) operated at 4K ambient temperature. To this date no other setup is available to produce less intrinsic thermal noise and as such the noise power P_{noise} has to be assumed to be fixed.

$$P_{noise} = 4k_b T_{noise} \quad \text{where} \quad T_{noise} \gtrsim 4K \quad (3.3)$$

Putting these ideas together we expect the SNR to have the following dependencies and bound

$$SNR \propto \sqrt{T} \cdot P_{meas}/P_{noise} < \sqrt{T_1} \cdot P_{max}/P_{noise}. \quad (3.4)$$

4 Theoretical Considerations

In this chapter we present the details on the physics and engineering aspects of the dispersive qubit measurement protocol.

4.1 System Model

As shown in section 2 the model for the qubit resonator system is given by the James Cumming Hamiltonian

$$\mathbb{H} = \hbar\omega_r\left(a^\dagger a + \frac{1}{2}\right) + \frac{\hbar\omega_a}{2}\sigma_z + \hbar g(a^\dagger\sigma^- + \sigma^+a) \quad (4.1)$$

where ω_r is the bare resonance frequency of the resonator, ω_a the qubit transition frequency, g the coupling constant and σ the pauli-matrices.

In the dispersive limit where $\Delta = \omega_a - \omega_r > g$ this describes a resonator which has a resonance frequency shifted depending on the qubit state as in (Fig. (4.1))

$$\omega'_r = \omega_r(2q - 1)g^2/\Delta. \quad (4.2)$$

where the quantum mechanical treatment of the qubit is replaced by the outcome q of the measurement operator $q = (\sigma_z + 1)/2$.

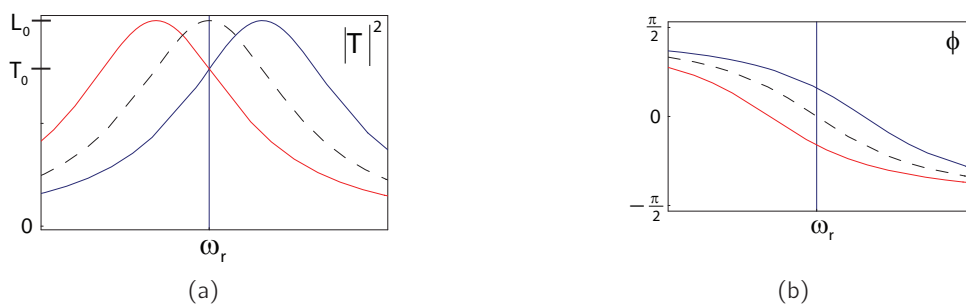


Figure 4.1: Transfer function of the resonator in the dispersive limit. a) Amplitude for bare resonator (dashed), with qubit in ground state (red) and excited state (blue). b) Phase shift with colors as in (a)

The transfer function for a linear resonator with resonance frequency ω_0 is given by

$$\mathbb{G}_{\omega_0, Q}(\omega) = \frac{i\omega}{(+\omega_0^2 - \omega^2 + \frac{\omega_0}{Q}i\omega)} \quad (4.3)$$

where Q is the quality factor of the resonator. The transfer function of the resonator qubit system can thus be written as:

$$\mathbb{F}_q(\omega) = \mathbb{G}_{\omega_r + (2q-1) \frac{g^2}{\Delta}, Q} \quad (4.4)$$

$$(4.5)$$

This transfer function will be used later in the simulations.

4.2 Qubit Lifetime

Besides the noise in the signal a second source of uncertainty which has to be considered is the energy relaxation of an excited qubit. This process is one of the fundamental limits in all existing qubit implementations and is due to coupling to the environment. In this way a qubit in the excited state loses its energy to some uncontrolled environmental degree of freedom. This effect can be included by introducing a lifetime of the qubit L as a random variable

$$L \sim \text{EXP}(T_1). \quad (4.6)$$

The qubit state for a single shot can then be written depending on the prepared qubit state q_0 :

$$q(t) = q_0 \mathbb{I}_{t < L} \quad (4.7)$$

where $\mathbb{I}_{t < L}$ is the characteristic function of the interval $[0, L]$.

For the following readout schemes it is assumed, that the measurement time T is restricted like

$$T \leq T_1 \quad \text{or} \quad T \sim T_1 \quad (4.8)$$

This is because longer measurement times bring small or no additional information on the qubit state. Also for any information processing applications requires the readout process to be faster than the typical lifetime of a qubit. The qubit decay is therefore treated as a perturbation to the ideal cases discussed below. For these cases optimal solutions are known, so we are left with close-to-optimal solutions.

5 Estimators

To design a measurement strategy for the qubit readout we have to solve two tasks. First an input signal has to be chosen, such that it extracts information about the qubit state. The output signal can be written as

$$s_{out}(t) = n(t) + \mathcal{S}_q\{s_{in}\}(t) = n(t) + s_q(t) \quad (5.1)$$

$$(5.2)$$

where $n(t)$ is a Gaussian noise process and $\mathcal{S}_q\{s_{in}\}(t)$ is the answer of the system with the qubit in state q . Secondly one has to find an estimator \hat{q} , which extracts the qubit state from the measured signal.

$$\mathbb{E}_{n(t)}(\hat{q}(s_{out})) = q \quad (5.3)$$

The difficulty in solving these two tasks lies in its interdependence. We will discuss different approaches in the following.

5.1 Bayesian Estimator

One approach is to choose a readout signal s_{in} and then find the optimal Bayesian estimator - the estimator which minimizes a-posteriori risk function - like the mean squared error - for the resulting measurement signals. We write

The optimal Bayesian estimator for a minimal square error is then written as

$$\hat{q}(s_{out}) = \mathbb{E}_{n(t)}(q | s_{out}) \quad (5.4)$$

In theory this can be solved for any given input signal.

Although an optimal estimator can be found in this way, no criteria on the quality of the input signals are given. This is for example, if the signals for the different qubit states are very similar it is difficult to distinguish them, despite the fact that this is done optimally. Therefore additional considerations have to be taken to assure that the input signal optimally extracts information from the system.

The very general approach of a bayesian estimator can also be impractical, because it might demand sophisticated numerical treatment, which is not suited and slow in digital hardware.

5.2 Digital Modulation Theory

A more practical approach is based on *digital modulation theory* - the theory of the transmission of information through noisy channels. A thorough introduction can be found in [4].

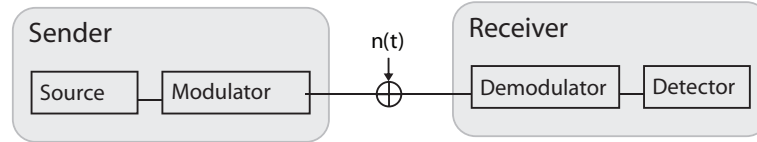


Figure 5.1: Schematics of a sender/receiver pair.

As seen in Fig. (5.1) the experiment setup can be seen as a sender/receiver pair transmitting one *bit* of information through an *additive white gaussian noise* (AWGN) channel. In this picture the sender emits a signal $s_0(t)$ or $s_1(t)$ depending on the qubit state and the input signal. The receiver then estimates from the noisy signal which of both was sent. This picture is useful, because this problem is successfully treated in digital modulation theory and is known to implement in hardware.

In the following we will use the sampled representation of the . It is common to consider sampled instead of continuous signals which shall be defined as

$$s[k] = s(kT_s) \quad (5.5)$$

where T_s is the sampling period. It is assumed throughout that the sampling rate is sufficient in all cases to accommodate for the relevant frequencies. As a mathematical tool we introduce a scalar product between such signals as

$$\langle s_0 | s_1 \rangle = \frac{1}{N} \sum_{k=0}^N s_0[k] s_1^*[k] \quad (5.6)$$

Based on this notion two sets of signals (modulations) are known [4] to allow optimal transmission of binary information through an AWGN channel.

$$\text{Antipodal signals} \quad \text{Orthogonal signals} \quad (5.7)$$

$$\langle s_0 | s_1 \rangle = -1 \quad \langle s_0 | s_1 \rangle = 0 \quad (5.8)$$

We will treat both cases at the same time now, since the strategies involved are very similar. In each case it is known [4] that there exists an optimal receiver which consists of a *demodulator* and a *detector*. The demodulator calculates a *score* Λ for antipodal (left) and orthogonal (right) signals as

$$\Lambda = \langle s_{out} | s_0 \rangle \quad \Lambda = \langle s_{out} | s_0 \rangle \left(\frac{\langle s_{out} | s_0 \rangle}{\langle s_{out} | s_1 \rangle} \right). \quad (5.9)$$

5 Estimators

The detector then makes an estimation based on the minimal distance of the score to one of two points as depicted in Fig. (5.2)ab. This is written as

$$\hat{q} = \arg \min_{q \in \{0,1\}} \|\Lambda - \lambda_q\|. \quad (5.10)$$

The points λ_q represent the score which would be obtained in a measurement in the absence of noise with the qubit in state q . Thus the estimator simply chooses the state whose score is closest to the measured score.

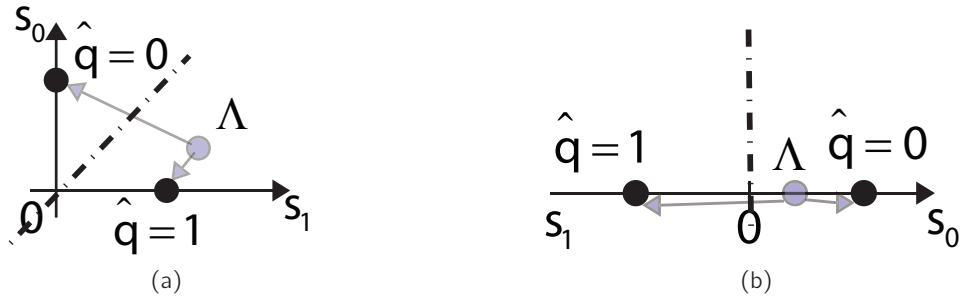


Figure 5.2: a) Complex vector space (\mathbb{C}^2) of the score Λ for orthogonal signals. The detector decides depending on the closer distance of the score to one of the reference points. This corresponds to the threshold line (dashed). b) Same for antipodal signals with one dimension (\mathbb{C}). Note that this one dimensional complex vector space corresponds to the complex plane.

The basic computational effort in this type of receiver is the calculation of the score. This is given by Eq. 5.6 and can be implemented using multiplication and accumulation operations (MAC). This special operation is available in modern DSP hardware and thus fast and cheap.

We consider now the two readout schemes, the *One Tone Scheme* which is already in use, the *Two Tone Scheme* which is newly proposed.

5.2.1 One Tone Measurement Scheme

A simple choice for the input signal is an harmonic signal with the bare resonance frequency of the resonator ω_r

$$s_{in}(t) = A_{RF} \cdot e^{i\omega_r t}. \quad (5.11)$$

From Fig. (5.3a) we see that for both qubit states the transmission T_0 through the resonator is equal and therefore contains no information on the qubit state. In contrast the phase is shifted by $\pm\Delta\phi$ depending on the qubit state. That is, the information on the qubit state is encoded purely in the phase of the measured signal. This received/measured signal is

$$s_{out}[n] = A_{RF} T_0 G \cdot e^{i(\omega_r - \omega_{LO})(nT_s) \pm i\Delta\phi} + \tilde{N} \quad (5.12)$$

where G is the gain of the amplifier and \tilde{N} its thermal noise.

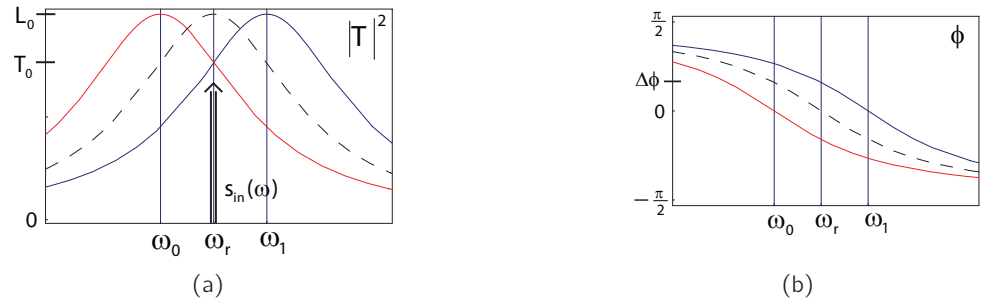


Figure 5.3: Transfer function as in Fig. (4.1) with indication of the input signal as a dirac pulse at the frequency ω_r .

This is equivalent to a binary modulation with the two signals

$$s_0[n] = e^{i(\omega_r - \omega_{LO})(nT_s) + i\Delta\phi} \quad (5.13)$$

$$s_1[n] = e^{i(\omega_r - \omega_{LO})(nT_s) - i\Delta\phi}. \quad (5.14)$$

For $\Delta\phi \sim \frac{\pi}{2}$ it is easy to check that these signals become antipodal

$$\langle s_0 | s_1 \rangle = e^{i2\Delta\phi} \sim -1. \quad (5.15)$$

For this case the optimal estimator can be implemented as shown above by calculating the score

$$\Lambda = \langle s_{out} | s_0 \rangle = \frac{1}{N} \sum_{k=0}^N s_{out}[k] s_0^*[k]. \quad (5.16)$$

Subsequently the euclidean distance to two reference points λ_0, λ_1 is calculated

$$\lambda_0 = \langle s_0 | s_0 \rangle = 1 \quad \text{and} \quad \lambda_1 = \langle s_1 | s_0 \rangle = -1 \quad (5.17)$$

which allows to estimate q

$$\hat{q}(\Lambda) = \begin{cases} 0 & |\lambda_0 - \Lambda| < |\lambda_1 - \Lambda| \\ 1 & \text{else} \end{cases}. \quad (5.18)$$

If the signals are not strictly antipodal this method is not necessarily optimal. Still for phase shifts close to $\frac{\pi}{2}$ it can be assumed to be a close-to-optimal solution. Note that in this case $\lambda_1 \neq -1$ as in Fig. (6.4).

Finally I want to point out a detail which is generally true for information encoding in the phase of a signal, which is the fact that the global phase has to be known to extract any information. In our case this does not pose any problem (in contrary to e.g. mobile communication) since in the whole measurement setup the global phase is controlled and known. As we will see this is not the case for the Two Tone Scheme, where the global phase is not necessary for the receiver.

5.2.2 Two Tone Measurement Scheme

In the second readout scheme the input signal shall consists of a superposition of two harmonic signals

$$s_{in}(t) = A_{RF}(e^{i\omega_0 t} + e^{i\omega_1 t}). \quad (5.19)$$

Fig. (5.4) shows, that depending on the qubit state one of the two components is much more attenuated than the other. We assume $L_1 \ll L_0$ so that the attenuated frequency component is small enough to be neglected. The measured signal becomes

$$s_{out}[n] = A_{RF}L_0 \cdot e^{i(\omega_q - \omega_{LO})(nT_s)} + \tilde{N}. \quad (5.20)$$

Where L_0 is the insertion loss of the resonator. Note that the phase shift is 0 in both cases and the information on the qubit state is encoded only in the frequency ω_q .

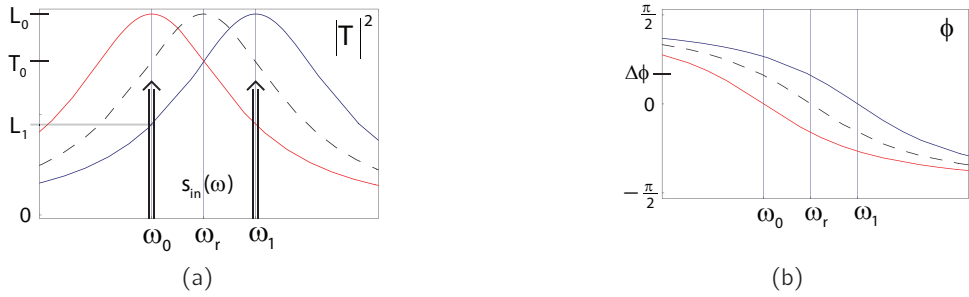


Figure 5.4: a) b)

This is again equivalent to a binary modulation with the two signals

$$s_0[n] = e^{i(\omega_0 - \omega_{LO})(nT_s)} \quad (5.21)$$

$$s_1[n] = e^{i(\omega_1 - \omega_{LO})(nT_s)}. \quad (5.22)$$

In Fig. (5.5) the value of $|\langle s_0 | s_1 \rangle|$ is plotted in dependence of the measurement time T . The signals are exactly orthogonal for

$$T = m \cdot \frac{2\pi}{\omega_1 - \omega_0} \quad m \in 1, 2, \dots \quad (5.23)$$

Also for longer measurement times the signals are approximately orthogonal. This is

$$T > 5 \cdot \frac{2\pi}{\omega_1 - \omega_0} : \quad |\langle s_0 | s_1 \rangle| \sim 0 \quad (5.24)$$

In these cases it is thus reasonable to use demodulator/detector as a good estimator.

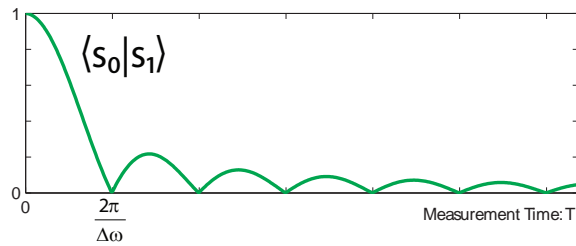


Figure 5.5: Correlation of the resulting signals in the two tone scheme. For certain measurement times and in the limit for long measurement times the signals are orthogonal

The score to be calculated is

$$\Lambda = \begin{pmatrix} \langle s_{out} | s_0 \rangle \\ \langle s_{out} | s_1 \rangle \end{pmatrix}. \quad (5.25)$$

The reference points are

$$\lambda_0 = \begin{pmatrix} \langle s_0 | s_0 \rangle \\ \langle s_0 | s_1 \rangle \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (5.26)$$

$$\lambda_1 = \begin{pmatrix} \langle s_1 | s_0 \rangle \\ \langle s_1 | s_1 \rangle \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (5.27)$$

The estimation is

$$\hat{q}(\Lambda) = \begin{cases} 0 & \|\lambda_0 - \Lambda\| < \|\lambda_1 - \Lambda\| \\ 1 & \text{else} \end{cases}. \quad (5.28)$$

Again in the case of near orthogonal signals the reference point λ_1 will differ from $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

6 Results

Preceding the hardware implementation the designated readout schemes were investigated in simulations. This helped to understand the influence of signal processing on the SNR in a quantitative manner. This includes the understanding of mixers, amplifiers and low-pass filters in the analog signal chain. Secondly the simulation should generate meaningful synthetic measurement data which could be used to test estimation algorithms.

6.1 Simulation

The simulations were done in Simulink/MATLAB (Fig. (6.1)) and focused on the signal processing aspects of the problem. Therefore the physics of the resonator qubit system was simplified. Depending on the state of the qubit the resonator was modeled with a linear filter as in Eq. 4.4. The qubit state is initialized by the experiment control to 1 or 0. Afterwards it decays to 0 after a random life time which is exponentially distributed. On a change of the qubit state the filter implementing the resonator is simply swapped. This does not take into account transient effects in the resonator due to qubit decay which might be relevant.

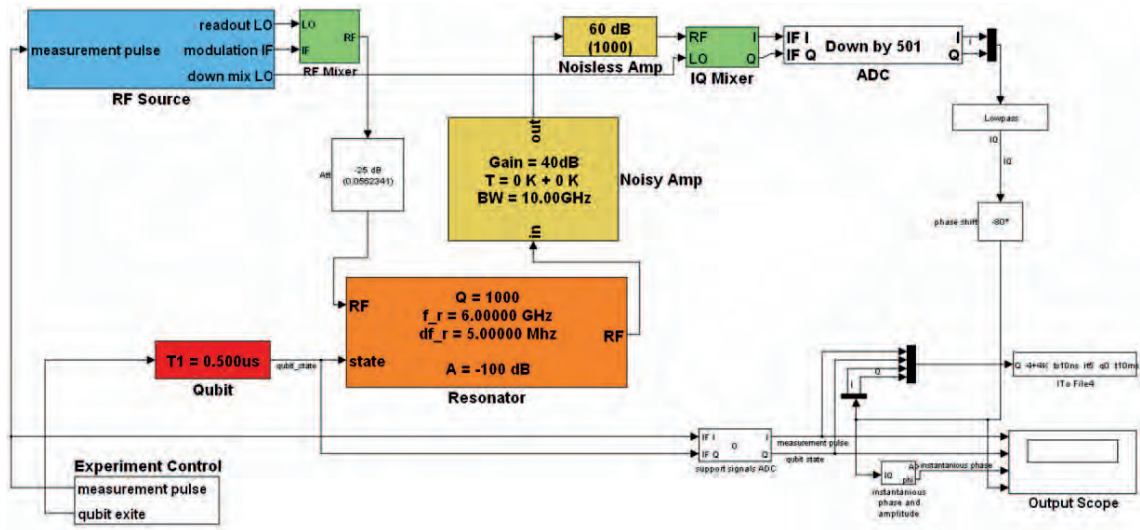


Figure 6.1: Simulation Scheme in Simulink/MATLAB.

The experiment control and the RF source were designed similar to the real experiment setup. The experiment control would repeat the same experiment continuously, that is preparing a certain qubit state, turning on the measurement signal for a timespan T .

In the RF source a *carrier frequency* (ω_{RF}) and a *modulation frequency* (ω_{mod}) could be configured to generate either one tone ($\omega_{mod} = 0\text{Hz}$) or two tone ($\omega_{mod} \neq 0\text{Hz}$) signals. In the two tone case, both frequency components are separated by $2\omega_{mod}$.

Furthermore a *local oscillator (LO)* signal is provided for down-conversion of the output signal. The resulting intermediate frequency ($IF = \omega_{RF} - \omega_{LO}$) of the output signal can be tuned, such that *homodyne* ($IF = 0\text{Hz}$) or *heterodyne* ($IF > 0\text{Hz}$) detection is possible.

The thermal noise of the HEMT amplifier was modeled by an additive source of Gaussian white noise. The noise is parameterized by its variance σ^2 which is determined by the bandwidth Δf of the signal processing chain and the equivalent noise temperature $T_{eq} = 4\text{K}$ of the amplifier.

$$\sigma^2 = 4k_b T_{eq} Z_0 \Delta f \quad (6.1)$$

where k_b is the Boltzmann constant.

Finally an analog-digital converter (ADC) is simulated by integrating over the sampling time, which was normally chosen to be $T_s = 10\text{ns} = (100\text{MHz})^{-1}$. The ADC samples both quadratures (real and imaginary part). The output signal which is used for estimation is then

$$s_{out}[k] = I[k] + iQ[k]. \quad (6.2)$$

To define the SNR the output signal can be written in terms of the system response and the amplifier noise

$$s_{out} = G \cdot \mathcal{S}_q\{s_{in}\} + \tilde{N}. \quad (6.3)$$

The theoretical SNR and the measured SNR are defined as

$$SNR = \frac{\langle G^2 \cdot \mathcal{S}_q\{s_{in}\}^2 \rangle}{\langle \tilde{N}^2 \rangle}. \quad (6.4)$$

6.2 One Tone Detection Algorithm

For the one tone readout scheme with homodyne detection the RF source is configured with the following parameters:

$$\omega_{RF} = \omega_r \quad \omega_{mod} = 0\text{Hz} \quad \omega_{LO} = \omega_{RF} = \omega_r. \quad (6.5)$$

In Fig. (6.2) a typical result of a one tone readout is shown. The qubit was excited three times (Fig. (6.2)b) and decayed during the measurement of $T = 2\mu\text{s}$ (Fig. (6.2)a). In Fig. (6.2)c both quadratures are shown without noise. This component of the output signal carries the information and is expected to be a constant, since $IF = \omega_{RF} - \omega_{LO} = 0\text{Hz}$.

The strong overshoot and the following oscillations result from the transient behavior of the resonator which is excited off resonance. These transients are emphasized through the high

quality factor of the resonator and have a similar timescale as the qubit lifetime. Equivalently one sees the decay of the resonator energy after the measurement signal was switched off. From the design of the simulation the qubit decay does not result in further transient behavior. Although this might be incorrect we neglect this problem for now.

In comparison Fig. (6.2)d shows the same signal with a typical noise level (SNR=-10dB). This figure is to illustrate that estimation has to be carefully designed.

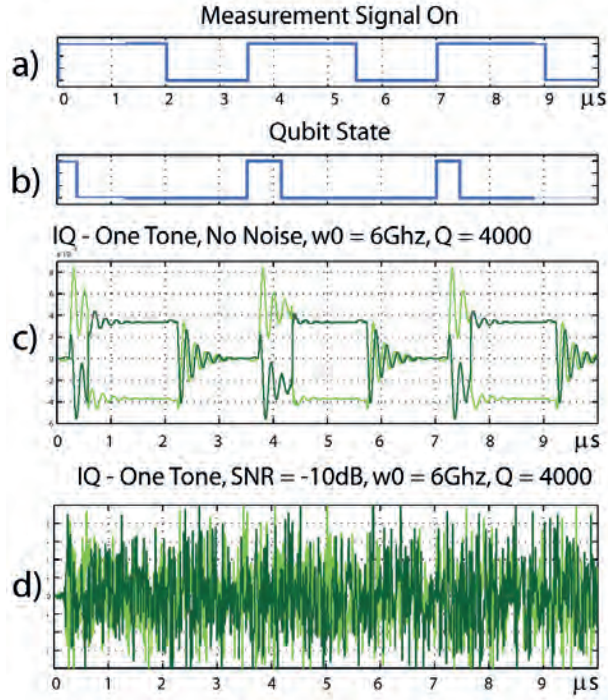


Figure 6.2: Typical simulation results for a one tone readout scheme. a) Indicates the time window during which the readout signal is applied. b) Shows the internal state of the qubit. c) Quadratures I and Q without noise after down conversion and sampling as they are available to the FPGA. d) Same as above but with a signal to noise ratio of -10dB.

To test the detection algorithm described in section 5.2.1 a set of measurements was generated (Fig. (6.3)). For both qubit states 28 measurements, as well as the two noise-free reference signals s_0 and s_1 were generated. Note that the reference signals can be equally extracted by real measurement through extensive averaging of noisy data.

The score of the demodulation Λ is shown in the complex plane in Fig. (6.4). As expected for antipodal signals the scores with preparation $q=0(q=1)$ cluster around $+1(-1)$.

The theoretical noise-free score (λ_0, λ_1 big triangles) is calculated for both cases which is in good agreement with the average scores ($\langle \Lambda_0 \rangle, \langle \Lambda_1 \rangle$, black triangles). These noise-free (or averaged) scores define a threshold line (dashed lines), which is the decision rule for the detector. This is here, scores left(right) of the threshold line are estimated $\hat{q} = 0(\hat{q} = 1)$.

Without further quantitative error analysis, the simulations show that this estimation algorithm gives correct results. We know its performance becomes optimal in the case of antipodal signals.

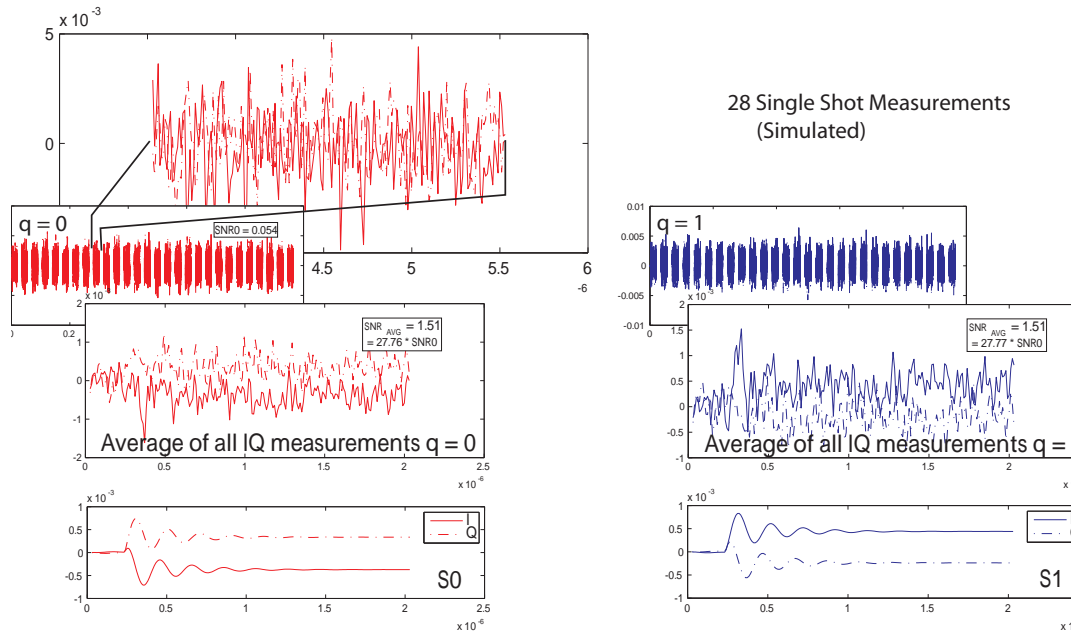


Figure 6.3: Simulation results of the resonator response for qubit in the ground state (red) and in the excited state (blue). Top: single simulated response, with $\text{SNR}=0.054$ out of 28 simulated responses. Middle: Average of 28 traces. Bottom: Ideal response without noise.

This assumption is good for long qubit lifetimes ($T_1 \gg T$) and large phase shifts ($\Delta\phi \sim \pi/2$) in the resonator.

For shorter qubit lifetimes Fig. (6.4) shows that the average score of $q=1$ measurements $\langle \Lambda_1 \rangle$ (green triangles) move towards $\langle \Lambda_0 \rangle$. This renders the intrinsic difficulty of determining the prepared qubit state, when the qubit decays very fast. In any case though it is possible to determine a good threshold line from the measurement of $\langle \Lambda_0 \rangle$, $\langle \Lambda_1 \rangle$.

6.3 Two Tone Scheme Simulations

For the two tone readout scheme the RF source was configured like

$$\omega_{RF} = \omega_r \quad \omega_{mod} = \frac{\omega_1 - \omega_0}{2} \quad \omega_{LO} = \omega_{RF} - \omega_{mod} = \omega_r - \frac{\omega_1 - \omega_0}{2} \quad (6.6)$$

For these settings one expects a signal with frequency $\omega_1 - \omega_0$ for an excited qubit and a 0Hz signal for a qubit in ground state. This is found in a typical result (Fig. (6.5)). Transient behavior of the resonator is visible as the exponential loading in the beginning of each measurement. Overshoot is not visible, because the resonator is driven at its resonance frequency.

The estimation algorithm for this type of measurement is in principle similar to the case of the one tone scheme and was not further investigated.

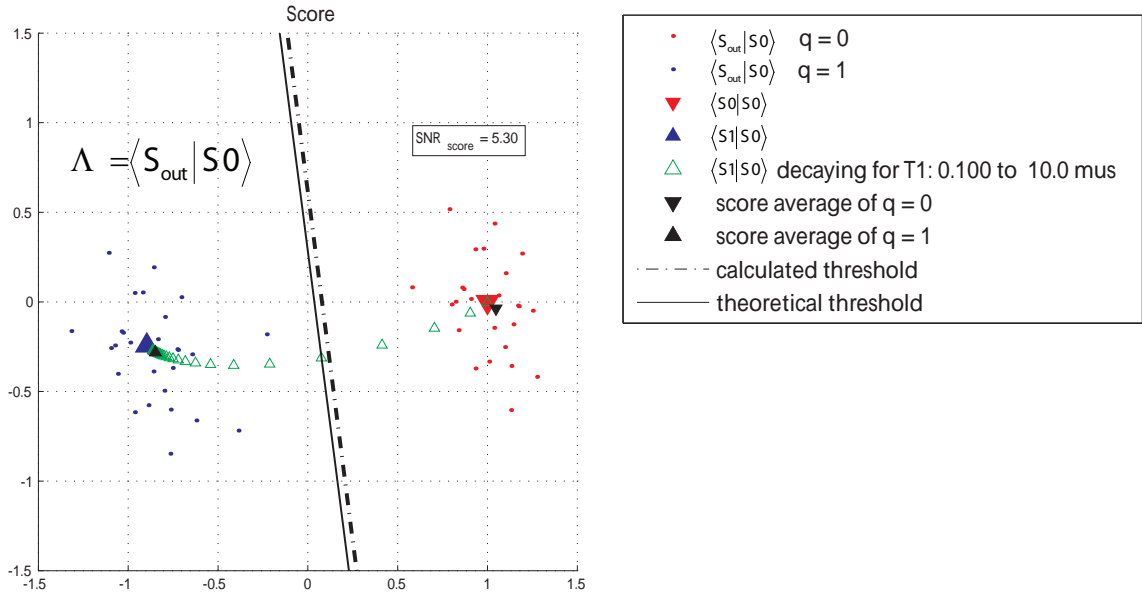


Figure 6.4: Scores calculated from the 28 simulated resonator responses in the complex plane (blue: $q=1$, red: $q=0$). Threshold lines (solid: theoretical, dashed: extracted from data) divide the plane in two decision areas (left: $\hat{q} = 1$, right: $\hat{q} = 0$).

6.3.1 Test Measurements on generating a two tone signal

To show that two tone measurements are feasible the experiment in Fig. (6.6) was conducted. First the generation of a two tone signal was demonstrated using a common RF-mixer using a RF-source with frequency ω_{RF} and a modulation source with $\Delta\omega = 1MHz$.

An exemplary output spectrum of the mixer (Fig. (6.6b)) shows clearly the two desired tones. The different side tones which are present result from impreciseness of the mixer. By careful biasing of the mixer a suppression of $> 37dB$ was achieved, which is considered sufficient for our application.

Secondly it was shown that a detuned resonator suppresses one of the tones sufficiently to make a distinction possible. A superconducting Nb resonator $\omega_r = 6.210GHz$ in liquid Helium was used. To imitate the situation of a detuned resonator one can choose:

$$\omega_{RF} = \omega_r + \Delta\omega \quad \approx \quad \text{resonator with qubit in ground state} \quad (6.7)$$

$$\omega_{RF} = \omega_r - \Delta\omega \quad \approx \quad \text{resonator with qubit in excited state} \quad (6.8)$$

The measurement results in Fig. (6.7) show that in both cases the second largest tone is $\sim 7dB$ smaller than the largest one.

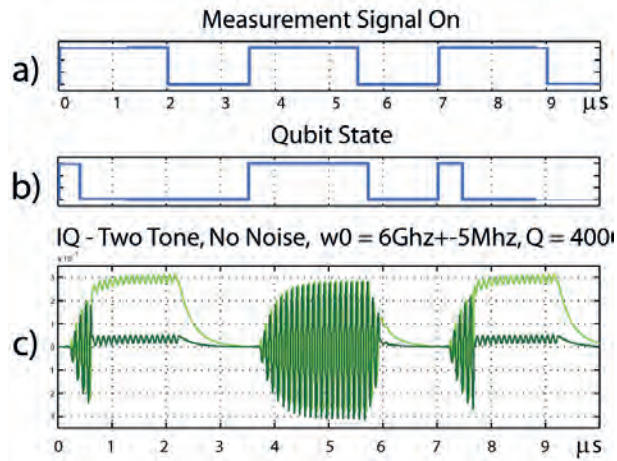


Figure 6.5: Typical simulation results for a two tone readout scheme. a) Indicates the time window during which the readout signal is applied. b) Shows the internal state of the qubit. c) Quadratures I and Q without noise after down conversion and sampling as they are available to the FPGA.

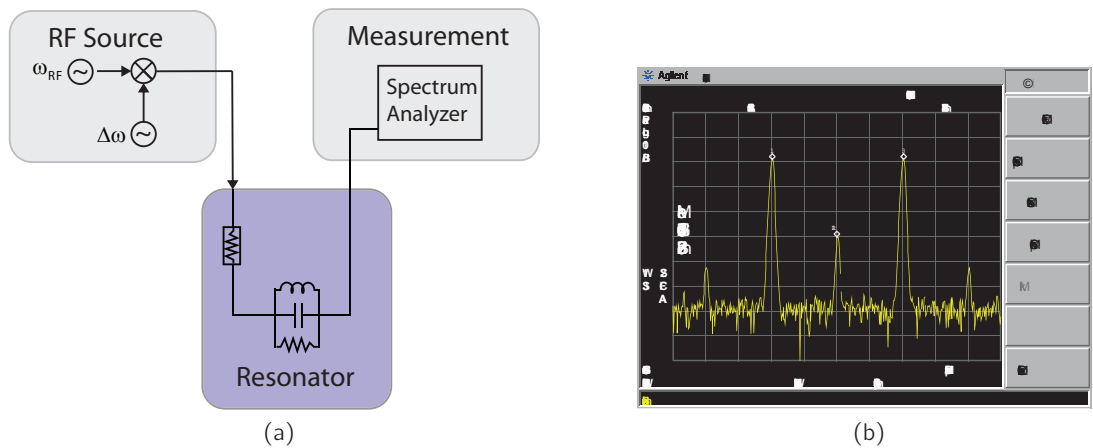


Figure 6.6: a) Experiment schematics b) Measurement at the output of the RF source, featuring the two desired tones as well as small parasitic components.

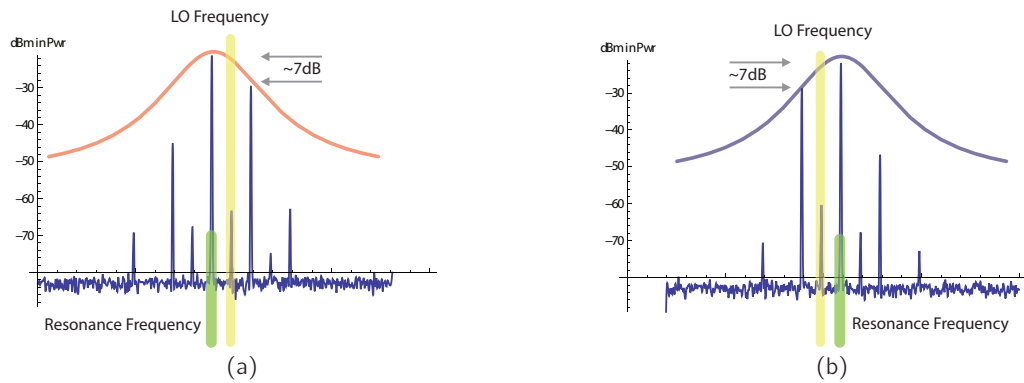


Figure 6.7: a) Experiment schematics b) Measurement at the output of the RF source, featuring the two desired tones as well as small parasitic components.

7 Hardware Implementation

7.1 Platform

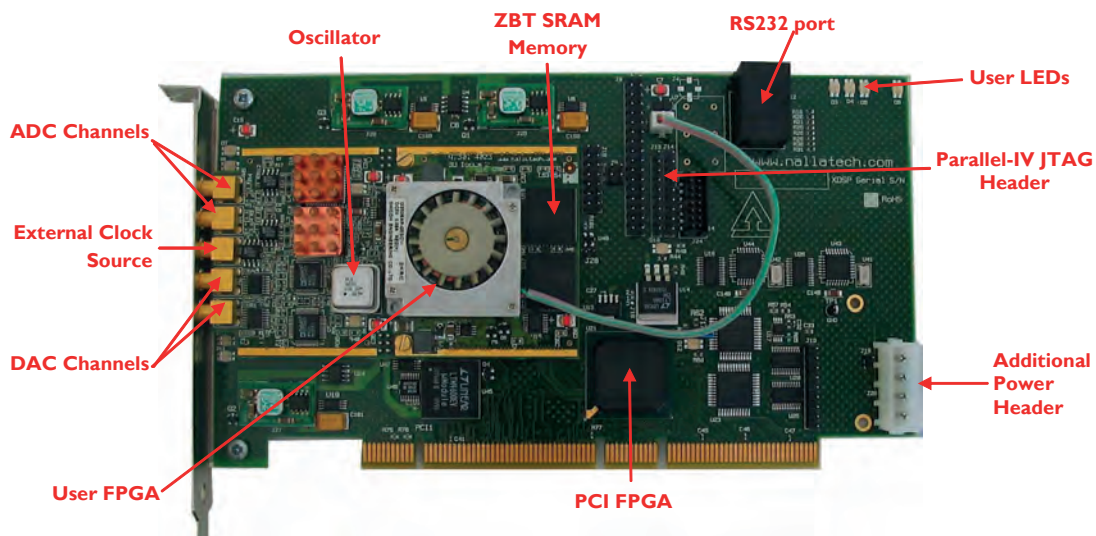


Figure 7.1: Xilinx XtremeDSP Development Kit IV.

For the implementation we chose the *Xilinx XtremeDSP Development Kit IV* as a hardware platform. The hardware (Fig. (7.1)) is based on the Virtex4 FPGA, which is a *high performance Field Programmable Gate Array*. This device can be programmed on a hardware level, which allows highly specialized and extremely fast logical operation.

7.1.1 Specifications

The FPGA is supported by a set of additional special purpose hardware as analog-digital-converters (ADC), digital-analog-converters (DAC), memory (ZBT-RAM), facilities to generate clock signals, on board LEDs and multipurpose input output connections (IOPINs):

The two ADCs (Fig. (7.2)) feature a maximum *sampling rate of 105MSPS* which allows to sample signals up to 52.5 MHz. The *voltage range of $\pm 1V$ is resolved with 14-bit*.

The two DACs (Fig. (7.3)) are not presently used but offer the possibility of generating analog signals with 160MSPS in a range of $\pm 1V$ with 14-bit resolution.

In the ZBT-RAM (Fig. (7.4)) which is organized in 2 banks with 32-bit words up to 2x8 MB can be stored. The connection to the RAM can be used at 100MHz which is to read or write one word per bank and cycle.

The 4 LEDs can be used as required e.g. to signal the state of the FPGA application or for debug purposes.

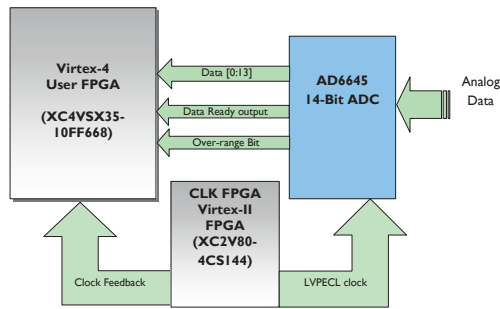


Figure 7.2: Xilinx XtremeDSP Development Kit IV.

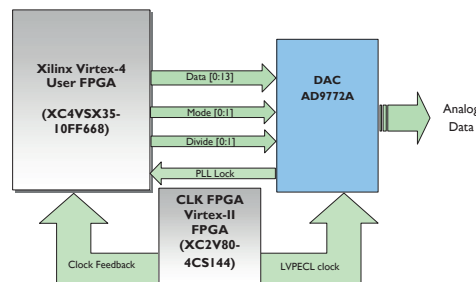


Figure 7.3: Xilinx XtremeDSP Development Kit IV.

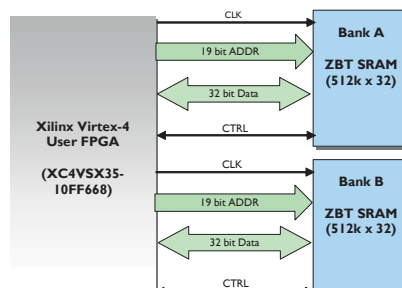


Figure 7.4: Xilinx XtremeDSP Development Kit IV.

7.1.2 Hardware Design Flow - Design Time

To program the FPGA hardware with the desired signal processing algorithms a chain of software tools is necessary, as depicted in Fig. 19. There is multiple ways to succeed in this task. The design flow which is presented here is specific to our solution. The different steps are presented from the design of the signal processing algorithm to the final startup procedure of the hardware.

To reduce the labor to change or add new algorithms the algorithmic part of the programming is separated from the technical parts. These parts including memory control or communication with the host computer are modified rarely and therefore encapsulated.

The algorithm is designed through the graphical interface *Simulink* in MATLAB. This interface provides all necessary logical blocks, high level blocks (e.g. FFT) as well as building blocks to access hardware resources (e.g. ADCs or memory). This part is compiled using *Xilinx System Generator* into a netlist (*ngc* file). This netlist describes the hardware layout (wiring) of the FPGA to achieve the desired functionality.

This netlist of the DSP algorithm is embedded into a wrapper, which provides access to the external hardware (Fig. 21). The wrapper is written in *VHDL* - a hardware description language which is industry standard. It contains a default configuration of a *memory controller*, a *clock manager* and a *boot loader*. The netlist and the wrapper can be combined into a complete hardware configuration file for the FPGA (*bit* file) with *Xilinx ISE Studio*.

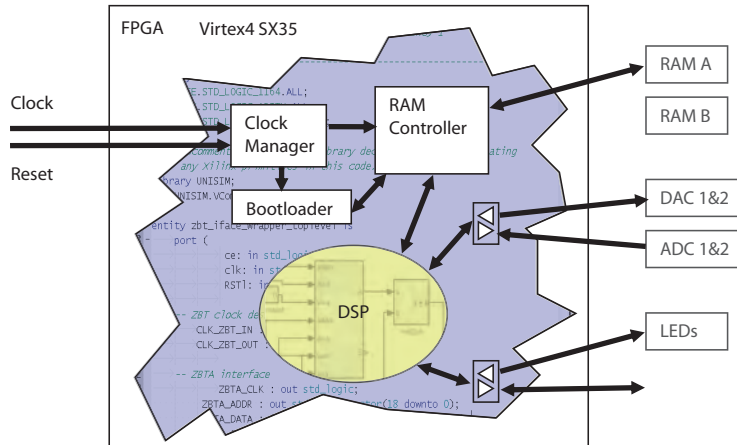


Figure 7.5: Application architecture contains a signal processing core (DSP) designed in Simulink, which is embedded in a wrapper written in VHDL to handle various hardware issues.

7.1.3 System Architecture - Run Time

To run and operate the FPGA board the *DIME* framework by Nallatech can be used to access the FPGA board, *configure* (i.e. load a hardware design) the FPGA with a certain design and communicate between host computer and FPGA. During the runtime the whole system is like in

Fig. 22. It is important to note, that the host computer and the FPGA board are independent entities which run different programs. On the FPGA a hardware design is running, whereas on the host computer some software is active which can communicate with the FPGA over the PCI bus.

The DIME framework is accessible in multiple ways. The easiest one is by scripts which typically look like in Fig. 23. It is straight forward to open the board, load a hardware design and run it. For more complex interfaces, where one wants to communicate or extract data from the hardware, a C/C++ interface is available.

To avoid the development of complex protocols between host and FPGA used this C++ interface to extract the RAM on the hardware. This tool *zbt_readwrite.exe* was implemented in C++ based on an example given by Xilinx. It could be used to extract results after the hardware design finished running. This is a generic scheme and can be used for any hardware design which does not need direct interaction with the host computer.

A more sophisticated design would include direct access to the *PCI bus* on the FPGA board. Over this bus it is possible to implement communication schemes with the host computer. This was not done in this project and is left for future work.

7.2 Practical Issues

7.2.1 RAM Access

At present time the RAM can be extracted after the run of any hardware design. To read the complete content of the RAM to a file *foo.ramdata* one calls from a command line:

```
zbt_readwrite.exe -r foo.ramdata
```

or to extract only a section of the RAM it is possible to specify a starting address (here 1000), the number of words to read (here 20) and the RAM bank which is used (here A):

```
zbt_readwrite.exe -r foo.ramdata A-1000+20
```

Equally the program can be used to write a certain *bar.ramdata* to the RAM with

```
zbt_readwrite.exe -w bar.ramdata
```

where the length and position of the data is specified in the *bar.ramdata* file.

Besides the Virtex4 FPGA on which the main hardware design runs the board features a small Virtex2 FPGA which is intended to handle the synchronization of the different hardware elements on the board. This so called clock FPGA has two sources of clock signals - programmable on board clocks and an external clock. These clocks can be used as references to generate new clock signals by multiplication (faster) or division (slower).

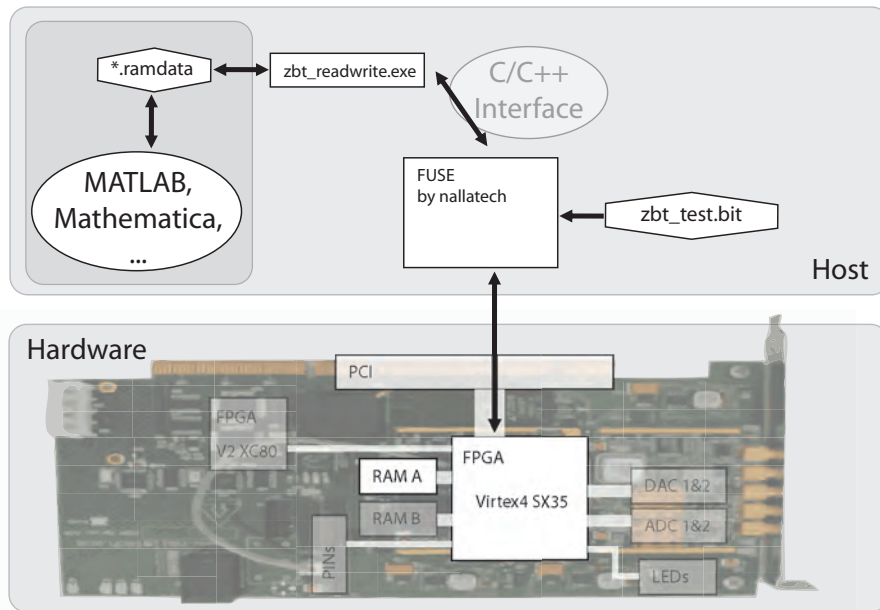


Figure 7.6: Schematics of the RAM readout process.

These clocks then have to be distributed to all the hardware elements - Virtex4 FPGA, ADCs, DACs, RAM. To assure that all elements run synchronous the physical traveling times of the clock signals have to be compensated. This is achieved through *deskewing*, where the clock phase is shifted through designated wire loops such that synchronicity is ensured.

The clock design with clock generation, deskewing and distribution has to be designed in a similar design flow as for the main FPGA. For the application in a lab environment a clock design *pl_clock.bit* was developed. This design generates a 100MHz system clock out of the 10MHz phase lock signal which is commonly used to synchronize all the electronic equipment. The system clock is deskewed and distributed to all the hardware elements on the board. In this way a synchronous design is achieved where data travels with constant speed through the signal processing pipeline.

7.2.2 Boot Loader

To start up the DSP algorithm correctly a *boot loader* part was integrated in the wrapper. When turning on the FPGA one has to wait to start the application until reset signals are turned off and clock signals are *locked* (stable). If this is not respected unpredictable results will occur.

When reset is turned off and the clock for the DSP algorithm is locked the boot loader will turn on the ram controller. In the following cycle the DSP algorithm is started. This assures that wrapper is in a defined state when the algorithm starts.


```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

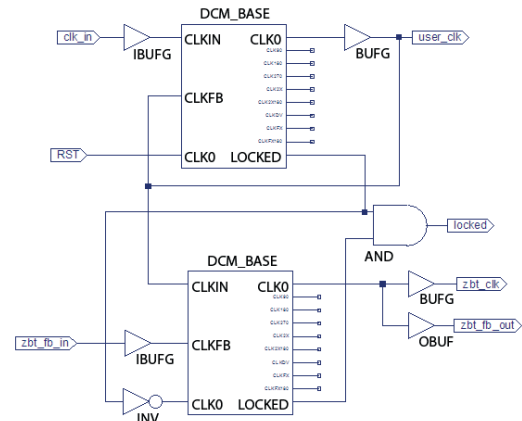
library UNISIM;
use UNISIM.VComponents.all;

entity bootloader is
port(
rstl_ext: in std_logic;
clk_lock : in std_logic;
user_clk : in std_logic;

app_stopped : in std_logic := '0';

app_ce_clr : out std_logic;
cenl : out std_logic;
gst_rstl : out std_logic
);
end bootloader;
    
```

(a) VHDL interface of the boot loader.



(b) Schematics of the clock implementation pl_clock.bit.

Figure 7.7:

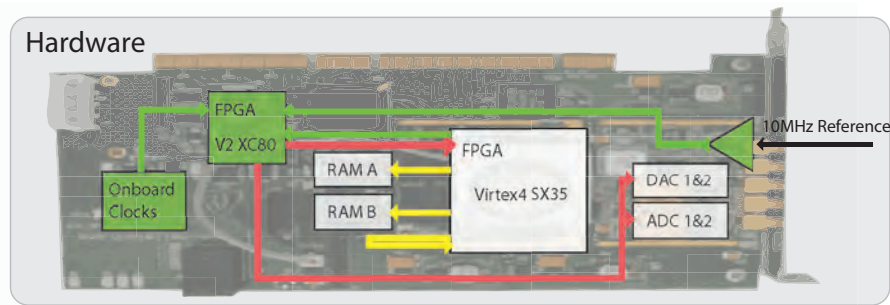


Figure 7.8: Schematic of the clocking architecture on the FPGA board. Clock inputs lines (green) are sources for the clock generation in the clock FPGA (green). Clock output lines (red) are designed such that the system clock is available in all devices without timeshift. The clock for the RAM (yellow) is deskewed by reference lines.

7.3 Example - The Averager

As an example to test the system we implemented a so called *averager*, which is already present in the current measurement setup in a different hardware module. The averager records measurements during N repetitions of an experiment and calculates the average over the repetitions. Thereby the SNR of the measurement can be improved as \sqrt{N} . This is because the noise is assumed to be Gaussian and therefore *averages* out.

The schematics of our averager can be seen in Fig. (7.9). It features two channels and was tested for up to 200000 repetitions. The inset shows how the actual averaging is implemented using a Dual Port RAM module. The Dual Port RAM allows to simultaneously read an old value of a sample and write a new value back.

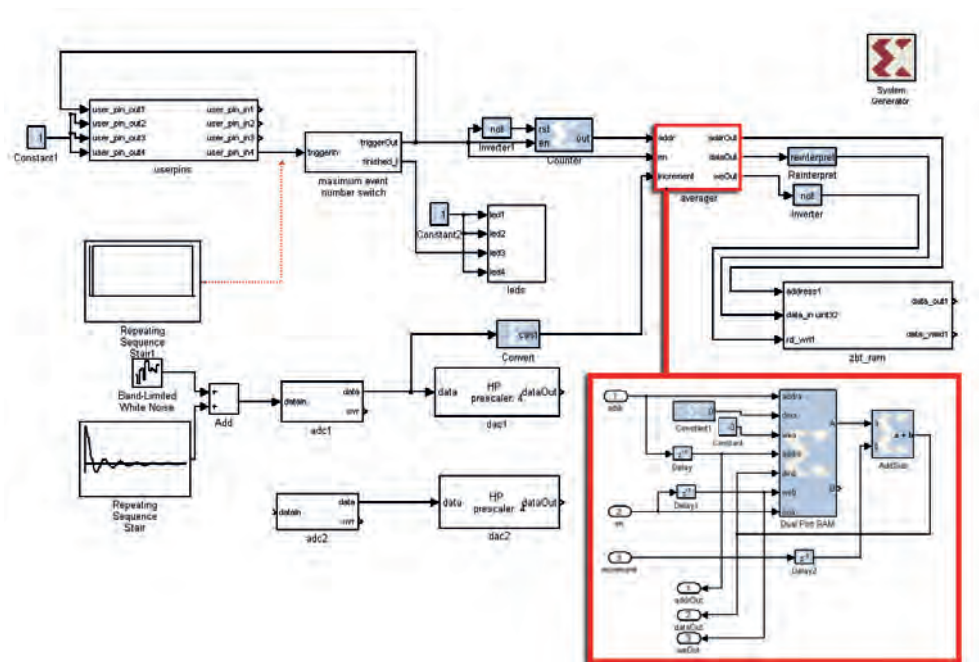


Figure 7.9: Schematics of the averager application in Simulink. Inset: The content of the averager box performing the actual averaging. Bottom left: Generation of the synthetic signals for simulation

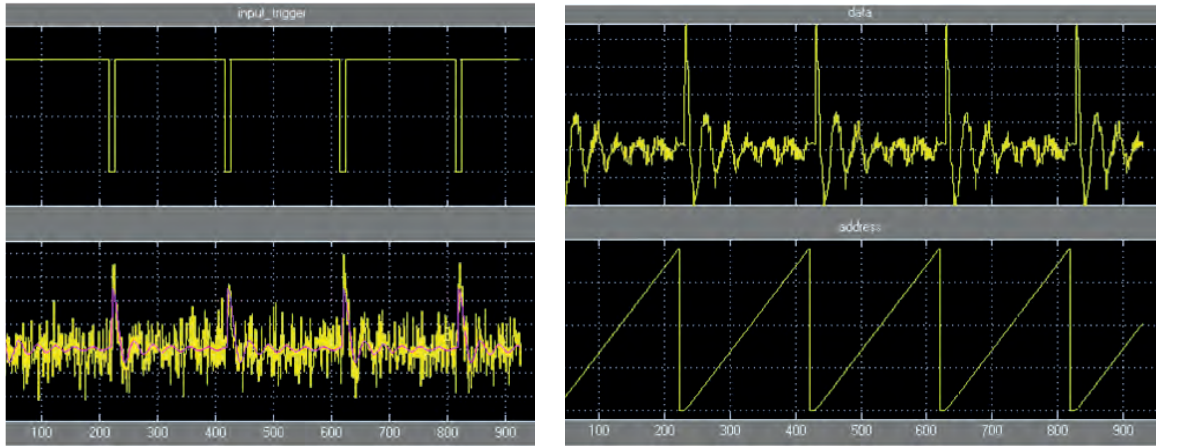
7.3.1 Simulation

Before performing real experiments the Simulink environment can be used to test the algorithm in a simulation. The signal which is synthesized for the simulation can be seen in Fig. (??) (violet). This signal is mixed with Gaussian noise (yellow) and is feed to the averager. Additionally a signal has to be provided which defines the time window for the measurement Fig. (7.10a).

The results in Fig. (7.10b) show that after 100 repetitions the averager was able to improve the SNR and recover the original signal. Below the address signal (Fig. (??)) shows how the measured signal is distributed in the Dual Port RAM module.

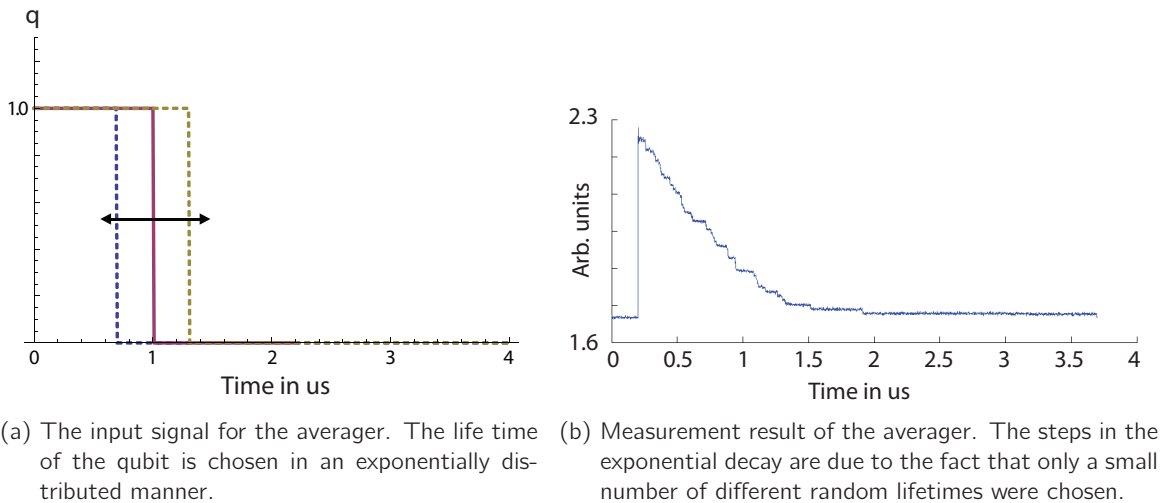
7.3.2 Experiment

To show that the averager works we performed a verification experiment. As the input to the averager we programmed an arbitrary waveform generator such, that it generates square pulses of random length with an average length of 1000ns, as shown in Fig. (7.11a). The measurement result in Fig. (7.11b) shows correctly the average of such a random signal, which is an exponential decay with a characteristic time scale of 1000ns. The irregularities in the signal are not due to noise or problems in the implementation but are due to the limited randomness (100 different lengths) of the input signal.



(a) Simulated input for the averager. Top: Defines the the begin and end of the averaging window. Bottom: Original input signal (violet) and the same signal with Gaussian noise (yellow). (b) Simulated output of the averager. Top: Signal stored in the Dual Port RAM and provided as the output of the averager. Bottom: The address in the RAM of the values shown above.

Figure 7.10: Simulation of the averager.



(a) The input signal for the averager. The life time of the qubit is chosen in an exponentially distributed manner. (b) Measurement result of the averager. The steps in the exponential decay are due to the fact that only a small number of different random lifetimes were chosen.

Figure 7.11: Input and Output of the averager test measurement.

7 Hardware Implementation

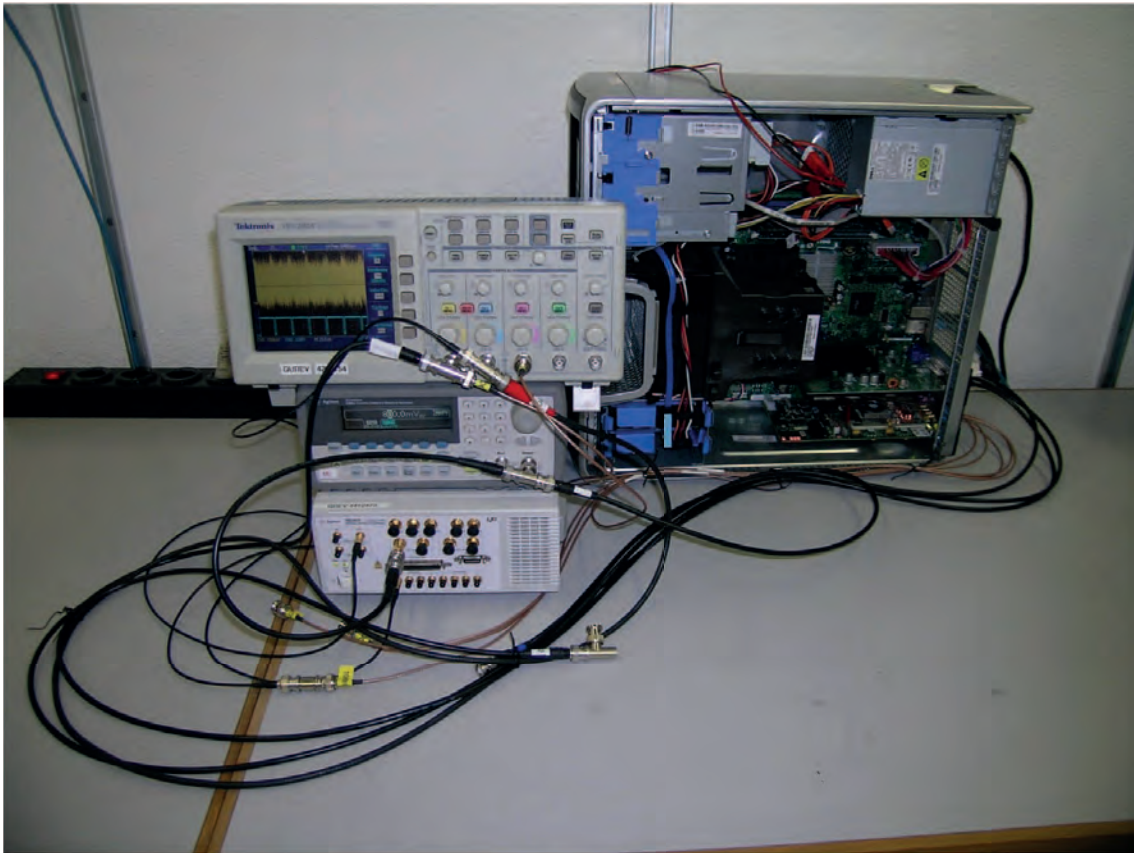


Figure 7.12: Experiment setup to test averager application on hardware with real signals.

8 Acknowledgments

I want to thank Prof. Andreas Wallraff for this valuable chance to work in his group. I enjoyed the possibility to learn about this new and interesting field of circuit QED and at the same time applying the tools I was given by the first years of studies.

In particular I want to thank Peter Leek for long discussions on the pitfalls of signal processing and helpful explanations throughout my work. Furthermore I want to thank Johannes Fink for introducing me to the measurement setup and answering never ending questions.

Finally I am thankful for the opportunity continue working on this project in a different context.

Bibliography

- [1] Shor, P. W. *SIAM Journal on Scientific and Statistical Computing* **26**, 1484 (1997).
- [2] Blais, A., Gambetta, J., Wallraff, A., Schuster, D. I., Girvin, S. M., Devoret, M. H., and Schoelkopf, R. J. *Phys. Rev. A* **75**(3), 032329 March (2007).
- [3] Fink, J. Master's thesis, Universität Wien, (2007).
- [4] Proakis, J. *Digital Communications*. McGraw-Hill Science/Engineering/Math, 4 edition, August (2000).

