A PORTABLE LOW-COST NMR SPECTROMETER

by

DAVID JOSEPH ARIANDO

Submitted in partial fulfillment of the requirements

For the degree of Master of Science

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

January, 2018

A Portable Low-Cost NMR Spectrometer

Case Western Reserve University
Case School of Graduate Studies

We hereby approve the thesis[1] of

DAVID JOSEPH ARIANDO

for the degree of

Master of Science

Dr. Soumyajit Mandal

Committee Chair, Adviser November 29th, 2017
Department of Electrical Engineering and Computer Science

Prof. Francis Merat

Committee Member November 29th, 2017
Department of Electrical Engineering and Computer Science

Dr. Ming-Chun Huang

Committee Member November 29th, 2017
Department of Electrical Engineering and Computer Science

---

[1] We certify that written approval has been obtained for any proprietary material contained therein.

I dedicate this thesis to my parents,
for their faith in me.

Table of Contents

List of Tables

List of Figures

Acknowledgements

0.1 Acknowledgements

I would especially like to thank my research advisor, Dr. Soumyajit Mandal, for his tremendous support for the research in my M.S. study. I'm always amazed by how much knowledge he has, and how much I learned from him.

Thanks to my parents, my brothers, and my only sister, for being patient with me through hardships. We went through everything together and I'm glad that you always support me for everything that I do.

I would like to thank my friends in ICSP Lab: Jifu, Cheng, Xinyao, Saif, Mason, Ali, Jarred, Alex, Parisa, Yingying, Steve, Xi, and all others not mentioned here. I cannot make it without your help. You guys are the best! Also our famous lab soccer intermural team which name cannot be mentioned here, and our famous lab basketball team which is almost disbanded due to time issues. Those games were so much fun.

I would like to thank my friends in International Student Fellowship: Thomas, Megan, Bill, Joan, Richard, Anushka, Nigs, Jeff, Bok, Earnest, Kai, Xiangyi, Rosie, Roseanne, and many others, for helping my transition when I came here by being such great friends.

I would like to thank my Indonesian friends in Cleveland: Melivia, Hez, Vincent, Aman, Chapin, Ko Lion, Ci Juliani, Ko Kelvin, Ci Dewi (both), Bang Lapiga, and many others. I enjoy the fellowship with you guys! Also Indonesian friends in Columbus which names will actually fill this whole page if mentioned one by one.

The last and my greatest gratitude is for the Alpha and Omega, my savior Jesus Christ. For His unfailing love and all the hope He gives me.

Abstract

A Portable Low-Cost NMR Spectrometer

Abstract

by

DAVID JOSEPH ARIANDO

0.2 Abstract

NMR (Nuclear Magnetic Resonance) spectrometer is a very important probing technique nowadays, with applications including drugs diagnostics, studies of molecular structure in chemistry, logging in petroleum industry, Magnetic Resonance Imaging (MRI), etc. However, it is usually bulky and expensive while also being hard to understand. Therefore, there is an urgent need to reduce the cost of an NMR spectrometer. In this thesis, two portable low-cost NMR spectrometer designs are developed and tested, utilizing custom analog front end (AFE) boards, an SoC FPGA board, and MATLAB as the main signal processing tool. The main motivation for the development of our portable low-cost NMR spectrometer is to make NMR study more accessible for educational purposes. The first design is a proof of concept working at a fixed frequency, designed for a particular sample. The second design is an upgraded version in hardware and software, with the ability to work in frequency range of 3.5 MHz to 4.7 MHz, therefore ready to be used for different kinds of NMR experiments. Its main feature is semi-automated operating frequency tuning. NMR signal coming from doped water has been obtained with the spectrometer. Its performance is compared to a commercial spectrometer, and shows a comparable result.

1 Introduction

Nuclear magnetic resonance (NMR) is a probing technique based on the interaction of magnetic moments of nuclei of various atoms within external magnetic fields. It senses the energy absorbed by changes in the nuclear spin state. Therefore, its important feature is that it provides material behavior and information at atomic level. NMR has a wide range of applications, and still continues to grow. Some of its applications are drugs diagnostics, compound structure study in chemistry, data logging in the petroleum industry, and Magnetic Resonance Imaging (MRI). The latter may be considered its most prominent application, used for medical diagnosis purposes. NMR analyzes samples non-destructively. This gives a huge advantage over many other techniques. One can use same sample multiple times to get good result. In case of low signal-to-noise (SNR) ratio, one can redo the experiment multiple times, effectively improving SNR the over measurements.

NMR and its highly useful features come with its high complexity knowledge behind it, i.e. physics concepts consisting of mechanical, electricity, and magnetism theories. While also being bulky, current research-grade NMR spectrometers are usually very expensive, making their accessibility low for different areas, particularly for educational purpose. For example, a bench top NMR analyzer MQC+ from Oxford Instruments costs around \$57,500, or an Kea NMR spectrometer from Magritek costs around \$20,000. For those reasons, NMR spectrometers are not very easy to access nor fully understand. Its cost and complexity makes it unsuitable for trial-error laboratory experiments. Also it is often inaccessible for beginners as the machine accessibility is considered luxury and usually limited to certain highly-skilled people. This leads to a great need to make NMR spectrometers cheaper and easier to learn, which is why more and more people are trying to make homemade NMR spectrometers.

The main motivation for the development of our portable low-cost NMR spectrometer is to make NMR study more accessible for educational purpose, by lowering the cost of hardware and simplifying its complex hardware with high-level language, while still maintaining some flexibility for any interested users. An additional goal is to make the spectrometer system autonomous and robust considering that it will be used under different environment conditions and samples, comparable to what a commercial NMR spectrometer does. In term of user interface, a high flexibility operating system is chosen to open up many possible applications if one is interested to improve it even more in the future.

1.1 The Principal Operation of Nuclear Magnetic Resonance

1.1.1 Microscopic Magnetization

A charge in motion generates a magnetic field. If a particular charge moves in a loop, it creates a magnetic moment. This phenomenon is similar to magnetic field induced in a coil, where current flows in a loop, effectively creating magnetic field in one direction. Atomic nucleus, called nuclei, is a small and dense region consisting of proton and neutron at the center of atom. All nuclei are electrically charged due to the presence of proton. Many nuclei are known to spin, effectively creating magnetic moment μ that is tightly related to its spin angular momentum. A spinning nucleus can be seen as a microscopic magnet. The magnetic moment of the nuclei is defined as

$$\mu = \gamma\, I \tag{1.1}$$

where γ is defined as the gyromagnetic ratio, and I is angular momentum quantum number often called the nuclear spin. Here we can see magnetic moment μ is proportional to the nuclear spin. Gyromagnetic ratio itself is a constant value related to the atom properties.

NMR experiment is made possible based on the interaction between nuclei spin and external magnetic field, B_0. Spinning nuclei, like any other spinning material, tends to keep its spin angular momentum. Based on this fact, with B_0 applied at spinning nuclei, another motion happens to the nuclei, shown in Figure 1.1. From the figure we can see that in addition to spin motion, the top of the atom moves in a circular manner along vertical axis. This motion is called precessional motion. The magnetic moment direction follows the top atom direction in instance, but in average this magnetic moment is aligned with B_0 field, shown with red dotted arrow.

The nuclei precession frequency is called the Larmor frequency (ω_0). The Larmor frequency is defined as

$$\omega_0 = \gamma\, \mathrm{B}_0 \tag{1.2}$$

where γ is the the same gyromagnetic ratio shown in Equation 1.1. From both equations, γ gives the relationship between its intrinsic spin properties, i.e. nuclear spin I and magnetic moment μ, and its other properties related to NMR which we are interested

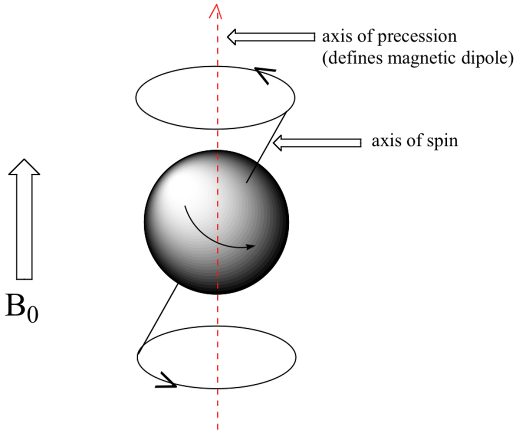

Figure 1.1. The nuclei precessional motion.
courtesy of: https://chem.libretexts.org
link: https://chem.libretexts.org/@api/deki/files/5119/image004.png?revision=1

in, i.e. the Larmor frequency ω_0, for a given permanent magnetic field B_0. In this case, Larmor frequency increases proportionally with B_0, which is also shown in Figure 1.2.

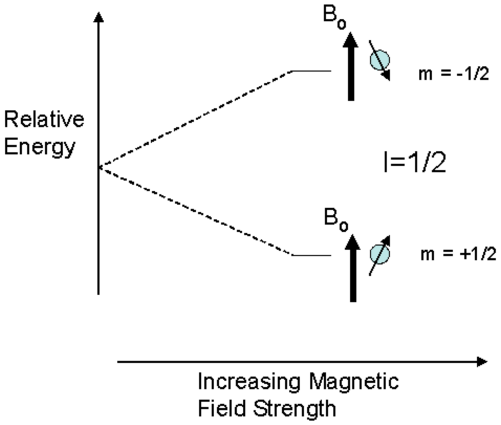

Figure 1.2. Spin states of NMR nuclei with I = 1/2 as a function of B_0.
courtesy of: https://en.wikipedia.org
link: https://en.wikipedia.org/wikiNuclear_magnetic_resonance

Without any external magnetic field applied, the magnetic moments of the nuclei are randomly oriented. The resultant of all magnetic moments is close to zero in random direction. Introducing an external magnetic B_0 aligns these magnetic moments, either aligned with the direction of B_0, often called +1/2 spin state, or opposed to the direction of B_0, often called -1/2 spin state. Figure 1.1 is an example of +1/2 spin state. The +1/2 spin state is lower in energy, causing a slightly more than half population of the molecules occupy this state. The rest of population occupy the -1/2 spin state, which is higher in energy. Difference in energy between the two spin states increases with increasing strength of B_0, shown in Figure 1.2. It is important to note that NMR technology depends heavily on population difference between the two spin states, where its total energy difference defines how much signal we can obtain from the sample. The energy difference between two spin states is defined as

$$E = \hbar\,\omega_0 = \hbar\,\gamma\,B_0 \tag{1.3}$$

where E is defined as transition energy between two spin states, $\hbar$ is defined as the reduced Planck's constant $(h/2\pi)$, where h is the Planck's constant, and ω_0 is defined as Larmor frequency. One can increase transitional energy from two spin states by increasing the Larmor frequency, which can be done by increasing B_0 field, effectively increasing total signal coming from sample. For this reason, people nowadays are trying to use as high B_0 field as possible. In the other hand, Boltzmann statistics defines the ratio of the two spin states with equation below

$$\frac{N^-}{N^+} = e^{\frac{-E}{kT}} \tag{1.4}$$

where E is the energy difference between two spin states, k is Boltzmann's constant, and T is the temperature in Kelvin. The ratio of N^- and N^+ is the ratio of spin state at lower energy level to higher energy level, respectively. One can see from the equation that the ratio goes up as temperature decreases, and approaches one as temperature increases. It interestingly predicts that NMR measurement technique is better at lower temperature.

If an RF signal perturbs the sample at a frequency whose energy matches the transition energy between two spin states, transitions will be induced in both directions, either downward direction (higher to lower spin state energy) or upward direction (lower to higher spin state energy). The upward transition absorbs energy and the downward transition releases energy. Total transitions in both direction is given by a probability multiplied by the population of molecules. Its probability is identical for both transitions, which also predicts that the net absorption or release of energy depends solely on the population difference between two spin states.

1.1.2 Bulk Magnetization

Up to this point, only microscopic magnetization has been discussed. A collection of microscopic magnets gives a macroscopic magnetization, or often called bulk magnetization. The bulk magnetization is a vector of all spin moments, created by population difference of the two spin states. In Figure 1.3, the bulk magnetization is defined as M_0. M_0 lies along the z axis, due to the fact that individual nuclear moment has equal probability of being in any direction of the x y plane. To observe NMR signal, an external

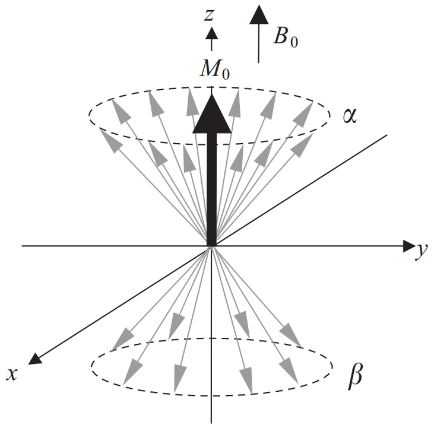

Figure 1.3. The bulk magnetization[1].

magnetic field, B_1, is introduced in a direction perpendicular to B_0. Its strength is usually really small compared to B_0. Similar to the concept of B_0 shown in Figure 1.1, where precessional motion is introduced at the direction of B_0 which is z axis in this case, if B_1 field is introduced along y axis, another motion is introduced at the direction of B_1. This new motion is called nutation. It's important to notice that nutation is created by B_1 when B_0 is also applied. In the other words, nutation of the nuclei happens on top of its precessional motion. If B_1 field is a permanent field, its effect along one circular precessional motion is effectively zero. To make B_1 field effective on top of precessing nuclei, its frequency has to be the same as the precessing motion frequency of the nuclei. This concept is mentioned by Teng in his book Structural Biology: Practical NMR Applications[1] with introduction of laboratory frame and rotating frame, which is really useful to understand this phenomenon. The laboratory frame is a steady xy plane where the nuclei is precessing around z axis. Net amplitude of its magnetic moment is then finite and rotates at xy plane in its Larmor frequency. The rotating frame is a rotating x

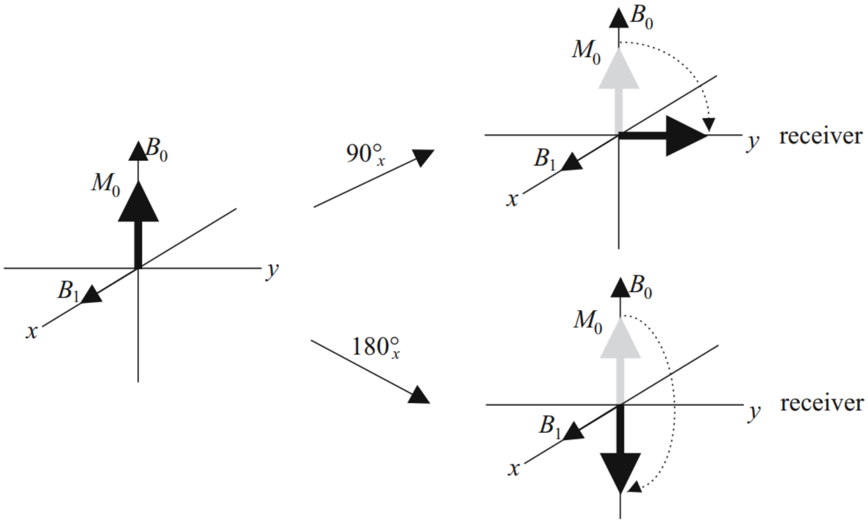

Figure 1.4. Nuclei nutation[1].

y plane at Larmor frequency. Now, if one observes the same nuclei magnetic moment at the rotating frame, its net amplitude at xy plane is still the same as before but is not rotating around the rotating frame, but instead steady. With B_1 field introduced at the same Larmor frequency at the rotating frame, it seems as it is also steady relative to the nuclei magnetic moment. Therefore, one can see the effect of B_1 field to the nuclei on the rotating frame, exactly the same as the effect of B_0 field to the nuclei on the laboratory frame. In the B_0 case, nuclei precesses around z axis, while in B_1 case, nuclei nutates around y axis of the rotating frame. Nutation angular frequency ω_1 is defined as

$$\omega_1 = \gamma \, \mathrm{B}_1 \tag{1.5}$$

where γ is the same gyromagnetic ratio of the nuclei we have been discussed before. One can see that the effect of B_1 field is analogous to what we have already seen with the effect of B_0 field in Equation 1.2. However, its nutation angular frequency is really small due to small B_1 field created by a coil, as later will be discussed. The nutation angle is defined as

$$\theta = \omega_1 t_p \tag{1.6}$$

where t_p is time length which the B_1 field is applied to the sample. In many typical cases of NMR, one might be interested to know how long it takes to nutate the nuclei by $\pi/2$ and $\pi/2$. These lengths are defined as

$$t_{\pi/2} = \frac{\pi}{2\omega_1}, t_\pi = \frac{\pi}{\omega_1} \tag{1.7}$$

From now on, the pulse which length is causing $\pi/2$ and π nutation is called $\pi/2$-pulse and π-pulse, respectively.

1.2 Basics of NMR Experiment

1.2.1 Continuous-Wave NMR

The so called first NMR experiment was developed simultaneously but independently by Edward Purcell[7] and Felix Bloch[8] in 1946. The experiment they developed is now called continuous-wave NMR. Figure 1.5 shows one example of a continuous-wave NMR. In the figure, B_0 field is generated by a strength-controllable magnet, B_1 field is generated by an RF coil placed perpendicular to the magnet, and another coil is used as a signal detector. To obtain an NMR spectrum, both continuous B_0 field and continuous B_1 field are applied to a sample. The NMR spectrum is analyzed by sweeping either B_0 field strength or B_1 frequency to obtain an NMR spectrum. Continuous B_1 field generates a low bandwidth signal in frequency domain. Thus, with one particular B_1 field, only nuclei with Larmor frequency within its bandwidth are excited. By sweeping B_1 frequency, NMR signal over frequency can be obtained. On the other hand, sweeping B_0 field results in different Larmor frequency for nuclei being observed. The effect is almost equivalent with sweeping B_1 field.

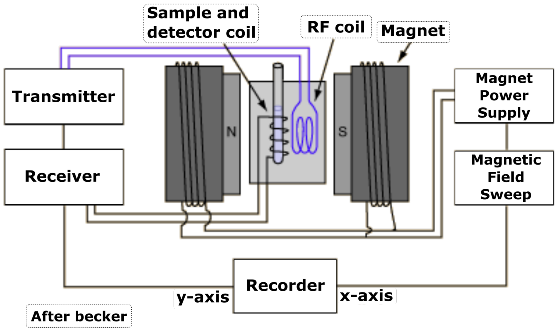

Figure 1.5. A continuous-wave NMR experiment.
courtesy of: hyperphysics.phy-astr.gsu.edu
link: http://hyperphysics.phy-astr.gsu.edu/hbase/Nuclear/nmrvar.html

1.2.2 Pulsed NMR

Modern NMR spectrometers use pulsed NMR method, which was introduced in 1950 by Erwin Hahn[9]. Instead of continuous B_1 field, high-power short RF pulses are used. In this case, the entire spectrum is excited at once, which results in an increased SNR per unit time. The output spectrum can be obtained using Fourier transformation. To do this experiment, with a given B_0 field, if $\pi/2$-pulse is applied to the sample, the nuclear magnetization M_0 is nutated to the xy plane, where it precesses at the Larmor frequency. The resulting signal is observable using Faraday induction if one uses a coil to capture its time-varying magnetic flux, shown in the following equation

$$v = n\frac{\mathrm{d}\phi}{\mathrm{d}t} = n\frac{\mathrm{d}BS\cos(\omega t)}{\mathrm{d}x} = -nBS\sin(\omega t) \tag{1.8}$$

with B is defined as the magnetic field, S defined as the area of the coil. Here we can see that the voltage generated at the coil is proportional to frequency ω, which suggests that NMR experiment sensitivity is better at higher frequency.

The nuclei itself relaxes back to thermal equilibrium via two types of processes: T_1 relaxation along the static magnetic field direction and T_2 relaxation in the transverse plane perpendicular to the field direction[10]. T_1 relaxation is the result of nuclear coupling to the surroundings, which is characterized by spin-lattice or longitudinal relaxation time, and T_2 relaxation is the result of coupling between nuclei, which is characterized by spin-spin or transverse relaxation time. T_1 relaxation is usually slower than T_2, but sometimes they can be equal. Free Induction Decay (FID) is the result of the two processes. One FID example is shown in Figure 1.6. From the figure, the red line is bulk magnetization vector after $\pi/2$-pulse has been applied. It starts at y axis far from the origin and ends at the origin. Its transverse plane is perpendicular to the B_0 field

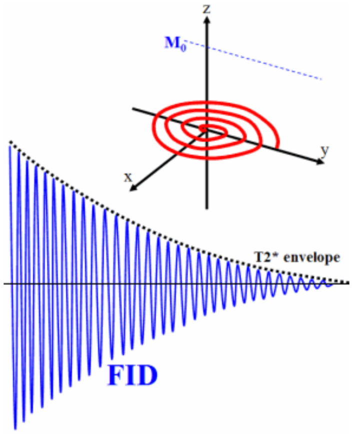

Figure 1.6. NMR Free Induction Decay (FID).
courtesy of: http://www.revisemri.com
link: http://www.revisemri.com/images/fid.gif

direction. Its magnetization is captured by a coil placed at x y plane, and the output voltage is shown as FID. We can see from the figure, the magnetization decays to zero in an exponential manner. Time in which the signal decays is called T_2 relaxation time. In this figure, the T_1 relaxation is not shown or considered to be far longer than the T_2 relaxation, therefore the magnetization hasn't relaxed back to z axis. Figure 1.7 shows a complete process with T_1 and T_2 relaxation taken into account. Here we can see the bulk magnetization relaxes back to z axis after sometime, which corresponds to its T_1 relaxation factor.

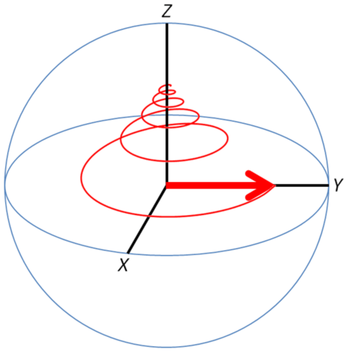

Figure 1.7. A complete NMR Free Induction Decay.
courtesy of: http://chem.ch.huji.ac.il
link: http://chem.ch.huji.ac.il/nmr/techniques/1d/pulseq.htm

1.2.3 Carr-Purcell-Meiboom-Gill (CPMG) Sequence NMR

Carr-Purcell-Meiboom-Gill (CPMG) sequence[2,11] is a well known technique for NMR measurement. It is based on E.L. Hahn's spin-echoes technique[12,13], where he found important to consider the diffusion of the molecules through an inhomogeneous external field. Inhomogeneous B_0 field induces different Larmor precession frequency in the sample. Therefore if one applies $\pi/2$-pulse and observe the sample at different space, they precess with different frequency and start to de-phase over time, causing the bulk magnetization amplitude of the whole sample to decline faster than its true T_2 relaxation time. CPMG is an important NMR technique which allows the true T_2 relaxation time to be measured even with an inhomogeneous B_0 field. The basic idea of CPMG sequence can be shown in Figure 1.8.

This method is superior over previous method developed by Hahn, which requires the sample to return to its equilibrium condition each time an additional echo is to be observed. In CPMG, the sample only starts from its equilibrium once, shown in Figure 1.9. First, $\pi/2$-pulse is applied. We can see a dephasing signal just after the $\pi/2$-pulse. After $t_e/2$ time has passed, π-pulse is applied. A rephasing signal can be seen to happen $t_e/2$ time after π-pulse, before it starts to dephase, therefore forms a peak magnitude t_e time after $\pi/2$-pulse. The phase shift between $\pi/2$-pulse and π-pulse is 90 degrees with respect to a stable continuously-running oscillator, as suggested by Meiboom and Gill. It is important in CPMG sequence, as it removes the effect of error adds-up, due to imperfect π flipping angle. $\pi/2$-pulse with its phase is usually called x-pulse and π-pulse with its 90 phase with respect to $\pi/2$-pulse is usually called y-pulse. Another π-pulse is applied to induce another signal rephasing. We can see another rephasing with less peak magnitude is shown after the second π-pulse exactly after another $t_e/2$

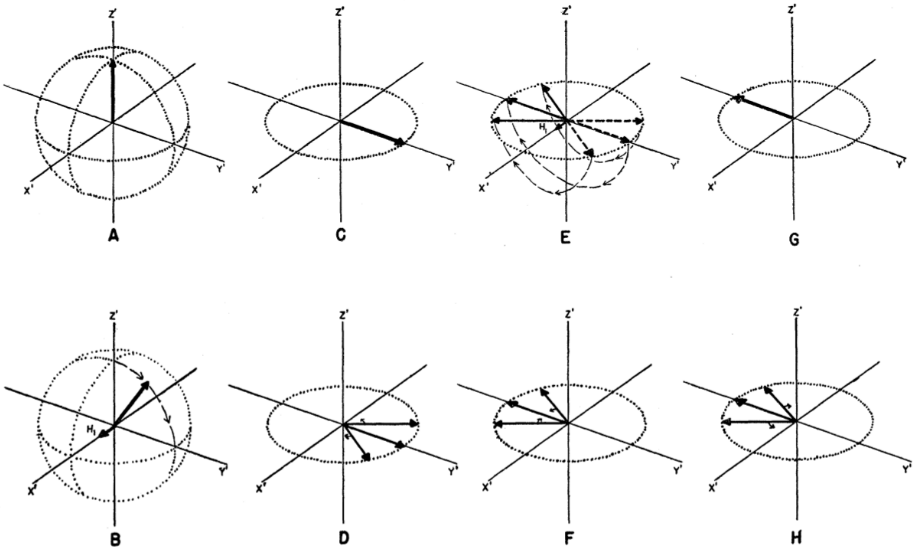

Figure 1.8. The formation of an echo[2]. Bulk magnetization is in its equilibrium position in a rotating frame viewpoint (a). With $\pi/2$-pulse applied, the bulk magnetization nutates to y axis (b) and reaches $\pi/2$ nutation at y axis (c). Microscopic magnetization starts to dephase due to inhomogeneous B_0 field (d). After some time, π-pulse is applied, microscopic magnetization nutates by π (e). Microscopic magnetization rephases (f). Microscopic magnetization completes its rephasing and the bulk magnetization is maximum in one direction (g). Microscopic starts to dephase again (h).

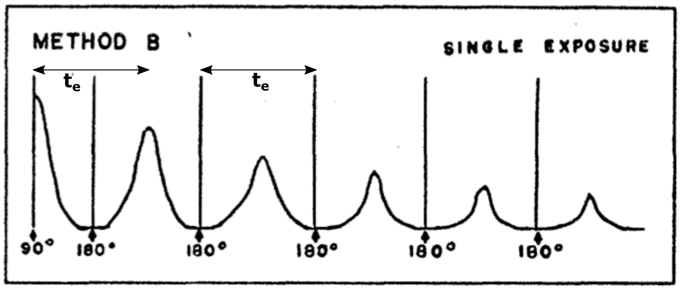

Figure 1.9. CPMG sequence[2].

time. More π-pulse is kept being applied with t_e time spacing, which is now called the echo spacing, as the spacing between two adjacent echoes is equal to spacing between

π-pulse. The process continues until signal decays to zero. From the figure we can see the true T_2 time, by observing peak amplitude decay over the whole measurement time.

1.3 Literature Review

Given some theories of NMR above, there are a lot of approaches and implementations in building an NMR spectrometer. Here we are interested in reviewing some NMR spectrometers that are considerably small, cheap, and portable.

1.3.1 Opencore NMR

The first one is Opencore NMR[3–5], in which the key strategy is to fully exploit a field-programmable gate-array (FPGA) as the controller unit. The block diagram is shown in Figure 1.10. Opencore NMR can have up to three radio-frequency (RF). Each transmit channel is driven with a pair of direct digital synthesizers (DDS). First DDS, later called DDS(I), is phase-tunable and frequency-fixed, while second DDS, later called DDS(II), is phase-fixed and frequency-tunable. With two DDS, the output signal can be controlled either its frequency or its phase. DDS(I) generates a signal at a fixed frequency of 20

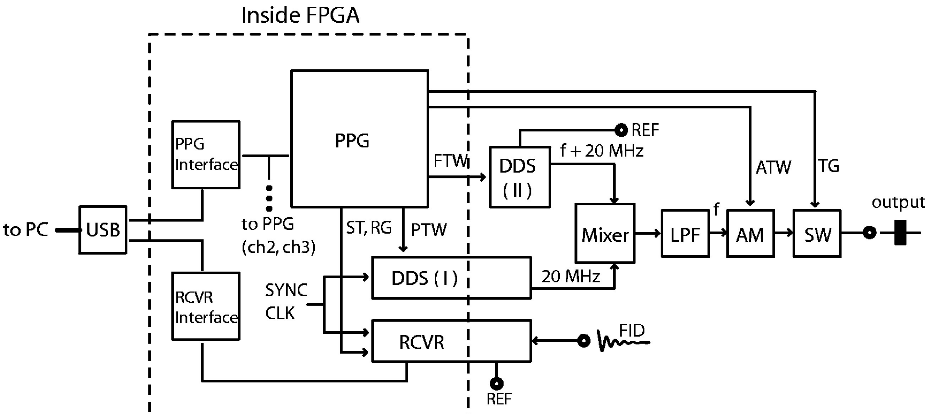

Figure 1.10. Opencore NMR block diagram[3–5].

MHz. DDS(II) is dedicated to generate various frequencies, controlled by the FPGA unit. It has two outputs, one is connected to the transmitter and the other is used for a receiver reference signal. DDS(II) generates signal at f, which equal to the intended transmit frequency + 20 MHz. Output signal from DDS(I) and DDS(II) is mixed to get f, and filtered using a low-pass filter to remove its f+40 MHz image. Its output is then sent to an amplitude modulator, which changes the amplitude level of the signal. Finally, the continuous wave output is enabled or disabled by an additional switch.

At the receiver, the NMR signal is converted from its frequency f to an intermediate frequency (IF) of 20 MHz, by mixing it with DDS(II) output frequency. Images of the signal is bandpass filtered. The signal is then amplified before it is sampled using and Analog-to-Digital Converter (ADC). Inside the FPGA, digital data at IF is digitally demodulated into in-phase (I) and quadrature (Q), and is passed to a digital low-pass filter, which are all written in VDHL. The processed data is then sent to a PC through a cable.

In the paper, an NMR experiment is demonstrated using the spectrometer, a power amplifier, 9.4T super conducting magnet, a preamplifier, and a homebuilt microcoil probe. The sample is 1μl of water at resonance frequency of 400.28 MHz. Figure 1.11 shows the water free induction decay from the experiment, represented in its real and imaginary components after digitally demodulated and low-pass filtered.

1.3.2 Open Source Relaxometry Platform

Open source relaxometry platform[6] was a thesis work of Mike Twieg at Case Western Reserve University. It is a microcontroller based portable NMR spectrometer, which is capable to operate as a standalone unit for portable or mobile applications. The block diagram is shown in Figure 1.12. It uses Atmel AT32UC3C1512 as main processor. It has

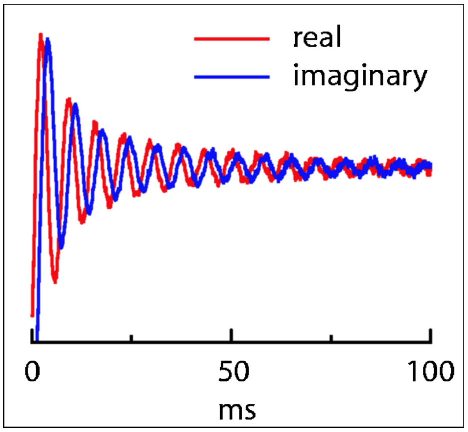

Figure 1.11. Opencore NMR, FID of water[5].

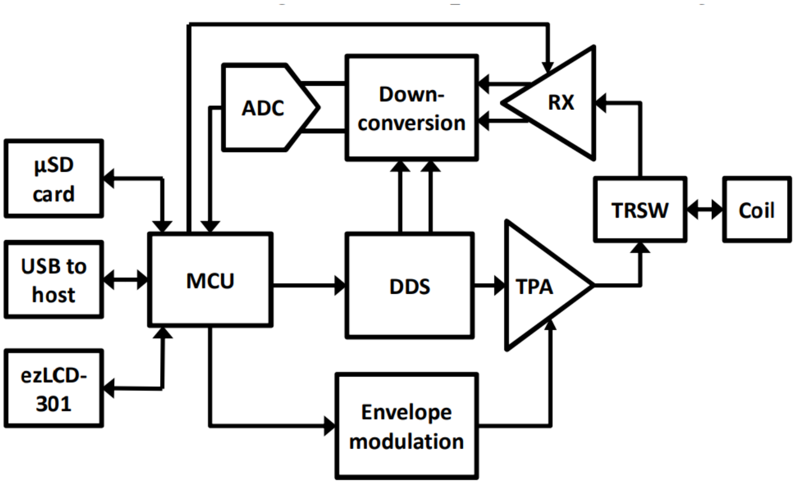

Figure 1.12. Open source relaxometry platform block diagram[6].

broadband RF chain (5-80 MHz), with the exception of transmit power amplifier (TPA). Similar to Opencore NMR, it uses DDS to generate RF signal for the TPA and receiver downconversion local oscillators. The TPA is implemented as a current-mode Class-D amplifier, which output is modulated using a digitally-controlled supply voltage. At receive side, RF signal is downconverted to baseband using DDS. The baseband output is fed to anti-aliasing filters and finally sampled by a 14-bit simultaneous sampling ADC.

Its data is stored in a micro SD memory card. Data analysis is done using MATLAB after the device is connected to a PC.

1.3.3 An ultra-low cost NMR device with arbitrary pulse programming

An ultra-low cost NMR device with arbitrary pulse programming[14] is another NMR spectrometer based on Altera Cyclone IV FPGA. A phase locked loop (PLL) is configured with predetermined parameters stored in a table within the memory, and is used as transmitter output frequency driver. Data is acquired with on-board digitizer and stored in a memory using DMA controller. Its internal ADC has sampling rate of 200 Ksps. All clocks signal are derived from same source, in which phase stability is ensured, enables signal averaging. All parameters are computed using a soft-processor implemented also in the FPGA.

An circuit board is utilized as an analog front end of the spectrometer. It has two transmitter channels and eight receiver channels. In the transmitter path, RF transmit pulse is generated directly using PLL, converted to a sine wave using an amplifier (MAX4384EUD+; Maxim Integrated, CA) and a bandpass filter (PBP-21.4+; Mini-Circuits, NY). An additional external amplifier (ZHL-3A+; Mini-Circuits) is utilized at the transmitter path. The total output power is 1W in a 50 Ω system. Its output is connected to an active transmit-receive switch (ZX80-DR230+, Mini-Circuit) before it is fed to the probe. In the receiver path, external preamplifiers (ZFL-500LN+; Mini-Circuits), filters (PBP-21.4+; Mini-Circuits), and mixer (ADE-1L+; Mini-Circuits, NY) are utilized to achieve a total gain up to 118 dB. An active low frequency band-pass filter (MAX274A; Maxim Integrated) is also used after down-mixing.

With this thesis, an FPGA-based NMR spectrometer is designed. It combines some of good properties of the 3 systems reviewed above. FPGA is chosen, instead of microcontroller used by Open Source Relaxometry Platform, as it gives more flexibility and better timing control. It is a low-field NMR in which clock frequency is low compared to Opencore NMR, therefore clock can be generated directly using FPGA, reducing external components needed. As it is portable, it needs to have an operating system to operate by itself.

1.4 Structure of the Thesis

In this project, two analog front end (AFE) boards have been designed. In Chapter 2, the idea of first AFE is simulated and presented in detail. In Chapter 3, first AFE is functionally tested and used to acquire an NMR signal coming from water. In Chapter 4, the idea of second AFE, an upgraded version of the first AFE, is simulated and presented in detail. In Chapter 5, second AFE is functionally tested and used to acquire an NMR signal coming from doped water. Chapter 5 contains conclusions and ideas for improvements of the current project.

2 First NMR Spectrometer

In this chapter, the first portable NMR spectrometer design is presented in detail. The project was started as EECS 397/600 : Instrumentation and Measurement course requirement. The initial version of spectrometer given in the course had some design problems, which were found and fixed. An updated design with all the fixes was then designed and tested in this thesis project.

2.1 Block Diagram

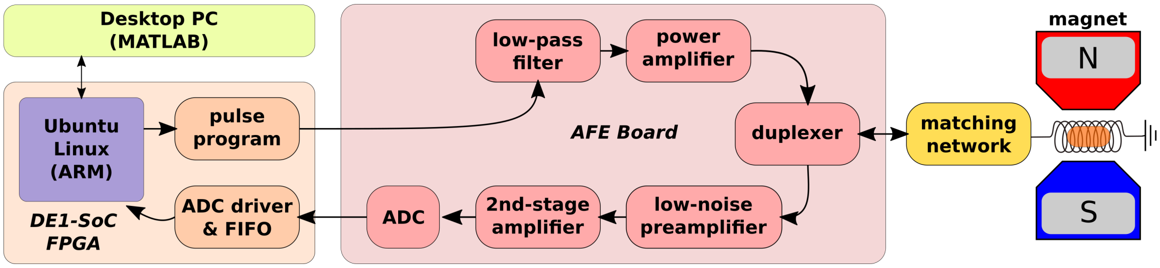

Figure 2.1. System block diagram of NMR system.

The first NMR spectrometer block diagram is shown in Figure 2.1. It consists of a coil sensor placed inside a permanent magnet, a matching network, a passive duplexer, a transmitter unit shown as power amplifier, a receiver unit shown as low-noise preamplifier and second-stage amplifier, an ADC, and an FPGA as the controller for the NMR spectrometer. The sample is placed inside the coil sensor with a probe holder which will be shown later.

Before we go to the description of the block diagram, it is useful to introduce the FPGA board we're using as the main controller. Here, we use Terasic DE1-SoC Development Kit, a System-on-Chip (SoC) which combines a dual-core Cortex-A9, later called the ARM processor or hard processor system (HPS), with Cyclone-V FPGA, later called FPGA or FPGA fabric. The communication between the ARM processor and the FPGA fabric is possible using a high-bandwidth interconnect backbone, implemented using AXI bus or Avalon-to-Memory-Map interface, configurable using Altera Quartus. Altera Quartus is the FPGA compiler, synthesizer, and programmer specifically used for Altera FPGA. The high-bandwidth interconnects backbone consists of HPS-to-FPGA bus, FPGA-to-HPS bus, and lightweight HPS-to-FPGA bus. The DE1-SoC board has 1 Gigabytes of DDR3 and therefore has the capability to run an embedded Linux with standard input/output (I/O) such as mouse, keyboard, and VGA display.

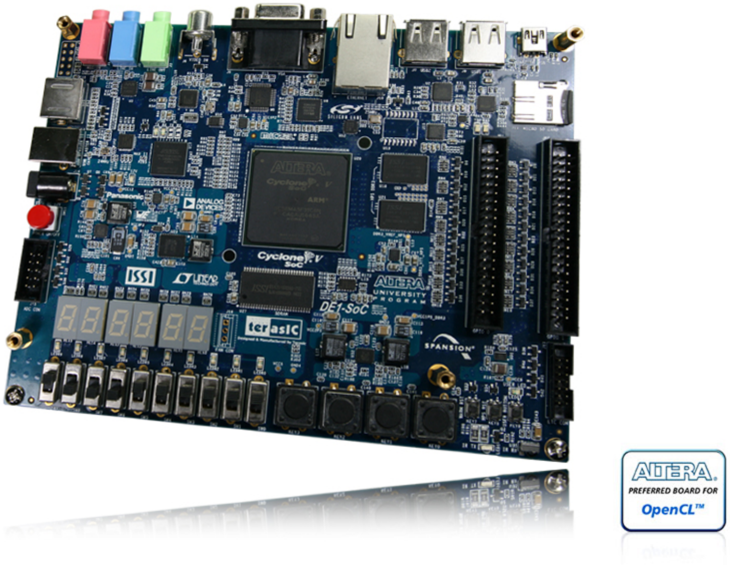

Figure 2.2. Terasic DE1-SoC Development Kit.

The principal operation of the NMR spectrometer is as follows. A *C* program is written, later called NMR *C* program, for the ARM processor under Ubuntu Linux as a master to all hardware in the FPGA fabric side, i.e. pulse program, ADC driver, and FIFO, later is called FPGA NMR controller. The NMR *C* program gives some parameters for the operation of FPGA NMR controller. Given all needed parameters, the NMR *C* program gives a start signal to the FPGA NMR controller. Given start signal, the pulse program generates an RF signal which is then filtered by a low-pass filter and amplified by power amplifier before it is sent to a duplexer. The duplexer senses a large transmit signal from the power amplifier and activates its transmit path, and also disactivates its receive path. The duplexer passes transmit signal to a matching network. The matching network purpose is to minimize signal reflection and ensure maximum signal absorption by coil. Coil converts the RF signal to a magnetic field which perturbs the bulk magnetization in sample inside the coil.

After that, the sample generates an NMR signal, which is relatively very small compared to the transmit signal. This small magnetic-field signal is converted by the same coil to voltage. The duplexer senses this small signal and activates its receive path, and also disactivates its transmit path. The duplexer passes received signal to the low-noise preamplifier, which amplifies the signal. Second-stage amplifier adds another amplification to the signal before it is digitized by an ADC. The ADC driver controls the operation of the ADC and stores the data in a large FIFO memory. The FPGA NMR controller gives a signal to the NMR *C* program when all operations have been finished. The NMR *C* program requests all the data from the FIFO buffer. Some additional signal processing is performed by the NMR *C* program. When it's done, one can fetch the data to a more powerful desktop PC to analyze the data. Here we use MATLAB to perform further analysis.

2.2 Coil Sensor Design

Given a relatively homogeneous permanent magnet, an NMR signal for a particular sample usually has a very limited bandwidth at its Larmor frequency. Before we design an NMR coil sensor, Larmor frequency of the sample needs to be computed using Equation 1.2. For the first NMR spectrometer, all circuit design parameters are not on-the-fly configurable and therefore is tuned for a specific sample. It is well known that NMR signal from water is relatively easy to be detected, as it has large amount of protons. For that reason, water is chosen as the sample and reference design parameter. For protons in water, γ is equal to 42.57 MHz/T. Now, we need to choose B_0 magnetic field source. Pasco Scientific EM-8618 variable gap magnet is used in this case, shown in Figure 2.3. We can adjust its magnetic field by adjusting the gap between its permanent magnet. Around 10 mm gap, magnetic field at the poles is 0.85T or 0.75T between poles. The figure also shows a soft magnetic pole pieces which we use to increase field homogeneity, but at the cost of lower field strength. In this case, the gap is fixed to the size of sample holder we 3D-printed, shown in Figure 2.4. Its gap is 10.5 mm. Magnetic field generated by the permanent magnet at 10.5 mm gap is 0.101 T. With B_0 of 0.101 T, Larmor frequency of water is computed to be 4.3 MHz.

Sensor coil designed in this chapter has several important characteristics, i.e. inductance, parasitic series resistance, and parasitic capacitance. Parasitic series resistance limits coil quality factor (Q). Whereas parasitic capacitance limits inductor frequency of operation, because one needs to operate it under its self resonance frequency (SRF).

Figure 2.3. Pasco Scientific EM-8618 variable gap magnet.

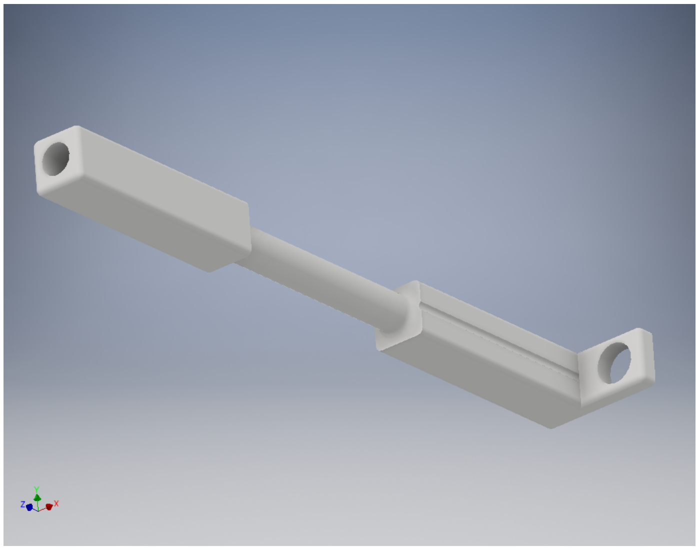

Figure 2.4. Coil and sample holder.

Over the SRF, capacitance becomes dominant and coil doesn't behave as an inductor anymore. The formula of resistance, capacitance, and SRF are not trivial to derive, and it is not discussed here. For an infinitely long solenoid, inductance is defined as

$$L = \frac{\mu_0 N^2 A}{I}, A \equiv \frac{\pi D^2}{4} \tag{2.1}$$

One approximation formula for finite length solenoids is defined as

$$L \approx \frac{\mu_0 N^2 A}{(I + 0.45D)} \tag{2.2}$$

with D defined as diameter, l defined as length, N defined as number of turns, and d as the wire diameter. Other important property of the coil for NMR sensor is how much magnetic field it generates. For an infinitely long solenoid, the magnetic field is defined as

$$B(x) = \left(\frac{\mu_0 N I}{l}\right)\hat{x} \tag{2.3}$$

However, for finite length solenoids, the field is not uniform within the coil. Along the center of the coil, its magnetic field is defined as

$$B(x) = \frac{\mu_0 N I}{4l}\left(\frac{2x + l}{\sqrt{(2x + l)^2 + D^2}} - \frac{2x - l}{\sqrt{(2x - l)^2 + D^2}}\right) \tag{2.4}$$

Using the equation above, the amount of magnetic field generated by the coil, defined as B_1, in term of length and diameter of the coil is defined as

$$B_1 = \frac{\mu_0 N I_{coil}}{2l} f\left(\frac{l}{D}\right) \tag{2.5}$$

This B_1 field is used to rotate proton from its equilibrium state, with its angular rotation frequency already defined in Equation 1.5.

A real coil is never a pure inductance, it also has parasitic resistance. Parasitic resistance introduces power loss. The amount of power dissipated by the coil due to its parasitic resistance is defined as

$$P = \frac{I^2 R}{2} \tag{2.6}$$

where R is defined as the parasitic resistance of the coil, I is the peak current flowing in the coil (assumed to be sinusoidal), and P is the power dissipated in the coil. In the other hand, if one has a source to drive a coil, due to impedance mismatch which will be discussed in the next section, there is some amount of power reflected back from the coil to the source. In this case, the power delivered to the coil is less than what is given by the source. Thus, the power generated by the transmitter, P_{tx}, is defined as

$$P_{tx} = \frac{P}{(1 - |\Gamma|^2)} \tag{2.7}$$

where Γ is the reflection coefficient factor of the transmission. The voltage generated at the coil, V, is defined as

$$V = ZI \approx \left(j\omega_{rf}L + R\right)I = \left(jQ + 1\right)RI \tag{2.8}$$

where Z is the impedance of the coil at the frequency of operation.

A real coil also has parasitic capacitance, forming its SRF. The SRF, f_{SR}, can be computed using a classic equation

$$f_{SR} = \frac{1}{2\,\pi\,\sqrt{L\,C}} \tag{2.9}$$

where L is defined as the inductance of the coil, and C is the parasitic capacitance of the coil. Any frequency above f_{SR} is practically not useful in this design.

It is also useful to discuss the bandwidth of the matching network, which is limited by its parasitic resistance. It is dependent on Q, quality factor of the coil, which is defined as

$$Q \equiv \frac{\omega_0 \left|Z'(\omega_0)\right|}{R(\omega_0)} \tag{2.10}$$

which is also,

$$Q \approx \frac{\omega_0 L(\omega_0)}{R(\omega_0)} \tag{2.11}$$

as long as the ω_0 is less than ω_{srf}, the SRF of the coil. In the matched condition, the effective quality factor of the matching network is approximately $Q_{match} = Q/2$, where Q is the quality factor of the inductor at the matching frequency. Thus, the matching bandwidth is approximately

$$\Delta\omega = \frac{\omega_0}{Q_{match}} = \frac{2\omega_0}{Q} \tag{2.12}$$

Maximum signal bandwidth is set by the coil bandwidth. If one generates B_1 pulses of length t_p, the signal bandwidth $\Delta\omega_p$ which is defined in the following equation

$$\Delta\omega_p \approx \frac{2\pi}{t_p} \tag{2.13}$$

is limited to $\Delta\omega$ of the coil to make sure all the signal within the bandwidth is absorbed by the coil. If the signal has broader bandwidth compared to the coil bandwidth, the signal is filtered out and wasted.

With equations above, a simple coil is developed, shown in Figure 2.5. An 18 turns coil was built around 5mm NMR tube. The length of the coil is 21mm, with 6mm coil diameter. The wire diameter is 0.7mm. From the calculation made in `http://hamwaves.com/antennas/inductance.html`, the expected inductance at the operating frequency is 0.43 μH, series reactance of 11.73 Ω, series resistance of 0.14 Ω, parallel parasitic capacitance of 180pF, and Q effective of 83. The self resonant frequency is about 18.1 MHz, computed from the expected inductance and the parallel parasitic capacitance.

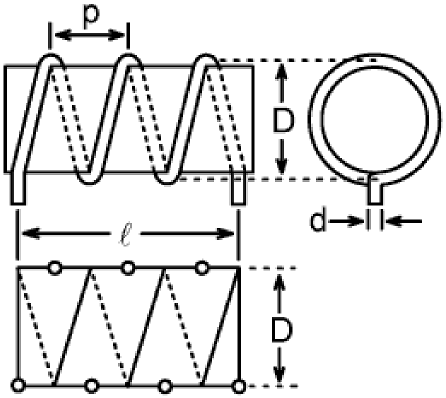

Figure 2.5. Coil sensor dimensions.
courtesy of: http://hamwaves.com
link: http://hamwaves.com/antennas/inductance.html

2.3 Matching Network Design

In general, one needs to minimize reflection by matching the source impedance to the load, which also maximizes power transfer to the load. The matching condition is defined as

$$Z_S = Z_L* \tag{2.14}$$

where Z_S is defined as the impedance of the source and Z_L is defined as the impedance of the load. The equation shows that matching condition happens when the impedance of the source is the conjugate of the impedance of the load. The amount of mismatch is quantified by the reflection coefficient, Γ or often called as S11 parameter, defined as

$$\Gamma \equiv S_{11} = \frac{Z_L - Z_0}{Z_L + Z_0} \tag{2.15}$$

It can be seen in the equation that minimum reflection can be achieved when Z_L is equal to Z_0. Z_0 is only a real number, which is typically 50 Ω for RF. In terms of power, power reflected is $|\Gamma|^2$ and therefore fraction of power transmitted is 1-$|\Gamma|^2$. From the rule of thumb, good matching is when $|\Gamma|^2$ < -10dB, or less than 10% power loss due to impedance mismatch.

Given only an inductive load, coil in this case, it's hard to achieve impedance match with the source. A matching network is used to transform the coil sensor impedance to satisfy impedance matching condition at the frequency of operation. The matching network is built using capacitors, connected series and parallel with the coil sensor, shown in Figure 2.6.

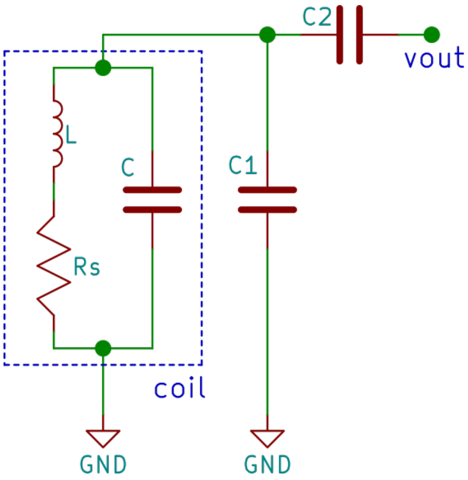

Figure 2.6. Matching network.

The input impedance, Z_{in}, looking from the source is defined as

$$Z_{in} = \frac{1}{j\omega C_2} + \frac{\left(j\omega L + R\right)\frac{1}{j\omega C_1}}{j\omega L + R + \frac{1}{j\omega C_1}} = \frac{-j}{\omega C_2} + \underbrace{\frac{j\omega L + R}{\left(1-\omega^2 LC_1\right) + j\omega RC_1}}_{Z_1} \tag{2.16}$$

Please note that the parasitic capacitance of the coil is absorbed within the total shunt capacitance C_1. Its value is actually far smaller compared to C_1 so C_1 will dominate in this case. For simplicity, C_1 described here is the total shunt capacitance, which also including C, i.e. the value we add to the actual circuit is C_1-C. For many RF circuits, the matching condition is satisfied when $\mathrm{Re}(Z_{in}(\omega_0)) = Z_0 = 50\,\Omega$ and $\mathrm{Im}(Z_{in}(\omega_0)) = 0\,\Omega$. The total input impedance can be viewed simply by plotting Z_1 and the impedance caused by C_2 independently, shown in Figure 2.7. The intersection between two lines shown by the imaginary component, shows cancellation of imaginary part of Z_1 using C_2.

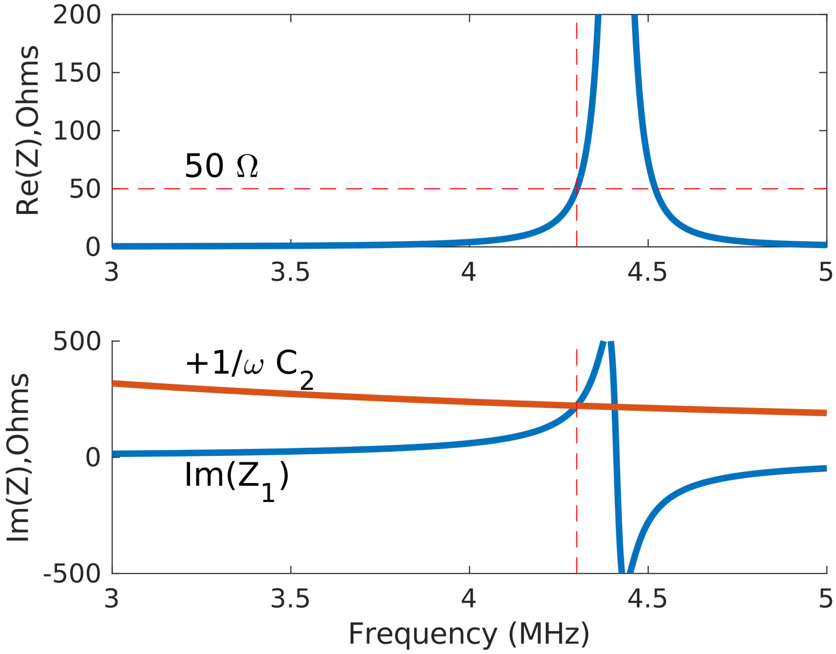

Figure 2.7. Input impedance of the matching network.

In this project, capacitors used are mica type capacitors, which was used due to its high Q properties. It is assumed that the capacitors have much higher Q than the coil. From the calculation done with MATLAB based on Equation 2.16, matching condition for the coil developed in the previous chapter happens when C_1 is equal to 3 nF, C_2 is equal to 166 pF, at the operating frequency of 4.3 MHz.

2.4 Transmitter Design

The transmitter is a class-A linear power amplifier. The purpose of the transmitter is to provide output amplitude control using duty-cycle. Its input driven by general-purpose

I/Os (GPIOs) output of the FPGA, which also provides duty-cycle control over the power amplifier. Figure 2.8 shows the schematic of transmitter unit.

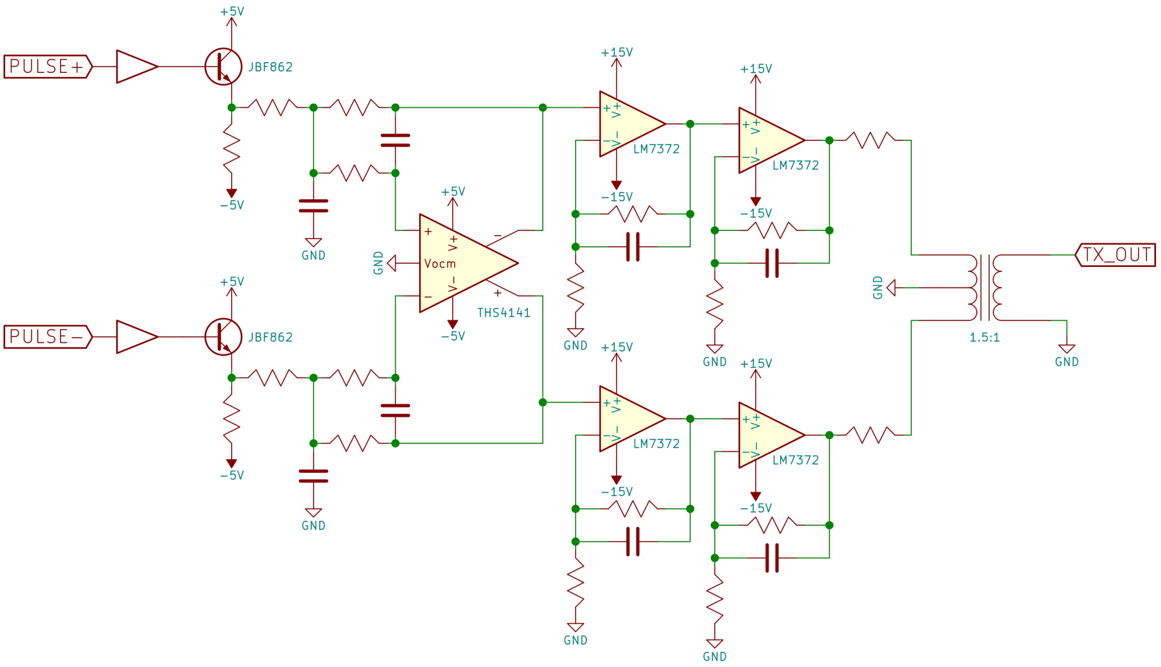

Figure 2.8. Transmitter schematic.

First, the differential signal coming from the FPGA is buffered with a line driver (74LVC2G125, Nexperia), and then further buffered with a common drain JFET (BF862, NXP). And then, the signal is filtered by a multiple-feedback active low-pass filter, using a differential Op-Amp (THS4141, Texas Instruments). After that, both differential output signals are amplified by two-stages Op-Amp (LM7372, Texas Instruments). Each stage gives signal gain of 3. By default, LM7372 output is low impedance and is not suitable for RF impedance matching directly. Therefore, to maximize power transfer, the output impedance of LM7372 needs to be transformed to 50 Ω looking from the load. For this purpose, a transformer is used to transform its output impedance. The output impedance of LM7372 can be approximated with LTSpice simulation. Figure 2.9 shows the voltage and current output of LM7372. Looking at the peak values, the resistance of the Op-Amp is approximated as

$$R_{out} \approx \frac{V_{DD} - V_p}{I_p} = \frac{15V - 11.6V}{150mA} = 22.7\Omega$$

To transform it to 50 Ω, the ideal transformer ratio is 1.5:1. Note that the current is further limited to 150mA with 50 Ω resistor at the output of the last stage LM7372, which also reduce power transfered to the transformer. This is due to current limitation of LM7372 Op-Amp. The original simulation shows peak current of 250mA without adding LM7372 output resistance. But there is a risk of destroying the Op-Amp after being used for a while, so current limiter resistor is added.

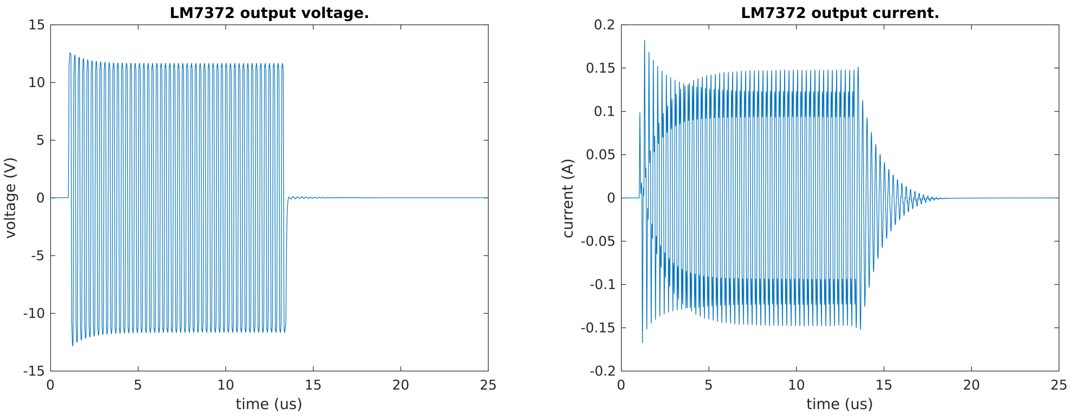

Figure 2.9. LM7372 output voltage (a) and output current (b).

The coil sensor designed at previous section is connected to TX_OUT, and its voltage and current at are simulated with LTSpice as well, shown in Figure 2.10. The coil voltage is about 60V peak-to-peak, and the coil current is about 5A peak-to-peak. It's interesting to note that harmonics shown in Figure 2.9(b) is filtered out in the coil sensor itself, giving a clean voltage and current signal shown in Figure 2.10. This is due to high-Q characteristic of the coil sensor at the operating frequency. The power dissipated on the coil is computed to be 2.47 Watts.

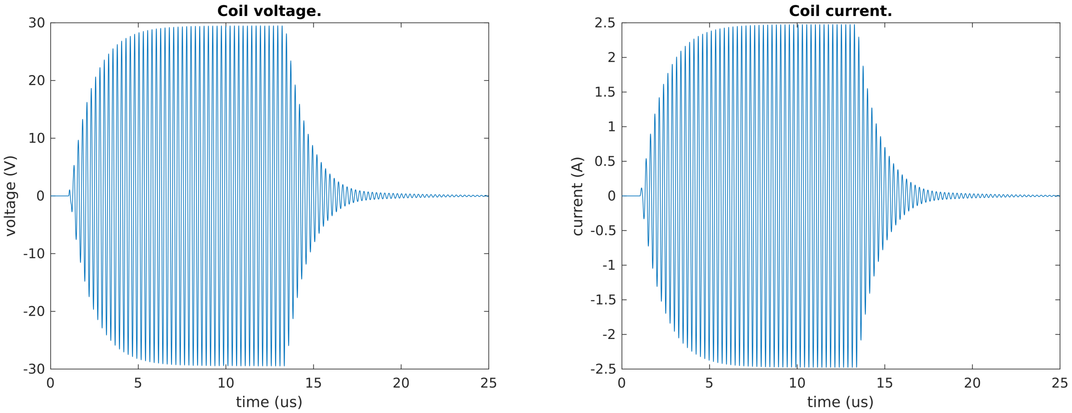

Figure 2.10. Coil transmit voltage (a) and current (b).

The transmitter is simulated in LTSpice to see its small signal gain, shown in Figure 2.11. The small-signal gain of the transmitter is about 33.3dB at 4.18 MHz. In practice, the input signal of the transmitter is already large as it's coming from the FPGA output, i.e. 3.3V peak-to-peak. As the output of the transmitter is maximum approximately at 12V, a huge gain is not needed. Therefore, the output is already saturated when the gain is only about 3.6, or around 11.2 dB. Although the transmitter is intrinsically broadband, the tuned load (i.e. the matching network) is present at the output. Therefore, the transmitter working frequency is approximately around 3.5 MHz to 4.8 MHz.

2.5 Preamplifier Design

The preamplifier is a very critical part in any instrumentation tools. This first stage of any instruments has to have low noise figure while providing a moderate amount of signal gain. To obtain a good noise figure, one needs to consider noise matching concept, which will be discussed in this section. On the other hands, to maximize the output, one needs to minimize reflection at the input, which we already discussed in the impedance

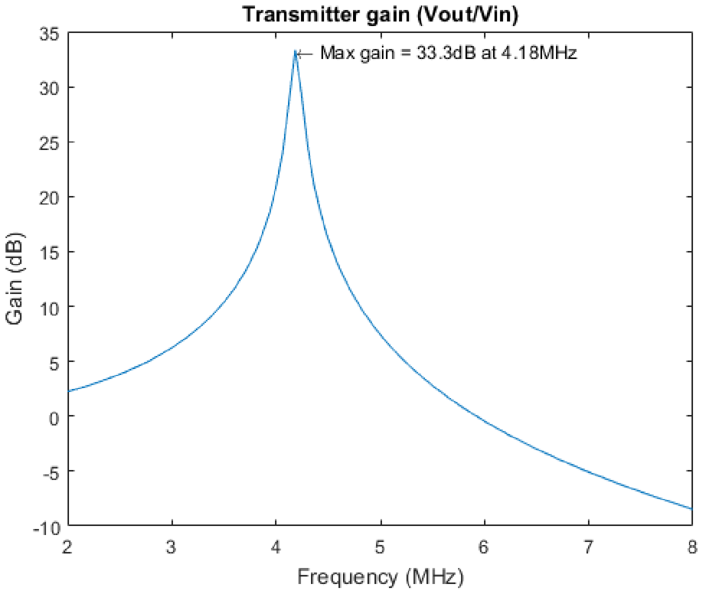

Figure 2.11. Transmitter gain.

matching concept. Those 2 concepts however, noise matching and impedance matching, is usually at different point of operation, therefore they need to be compromised between each other.

To achieve preamplifier input impedance matching to the source, there are two basic methods. The first method is to use a resistance at the input, which is intrinsically broadband but also noisy. The reason for noise is presence of resistance at the input of preamplifier, introducing a broadband thermal noise that is amplified with the same gain as the signal. This method is useful when one needs to implement a broadband preamplifier. The second method is to use reactances, i.e. capacitor/inductor networks, which is intrinsically narrowband, but is also noiseless theoretically, as ideal capacitors and inductors don't generate any noise. In practice, a small amount of parasitic resistance introduces some noise.

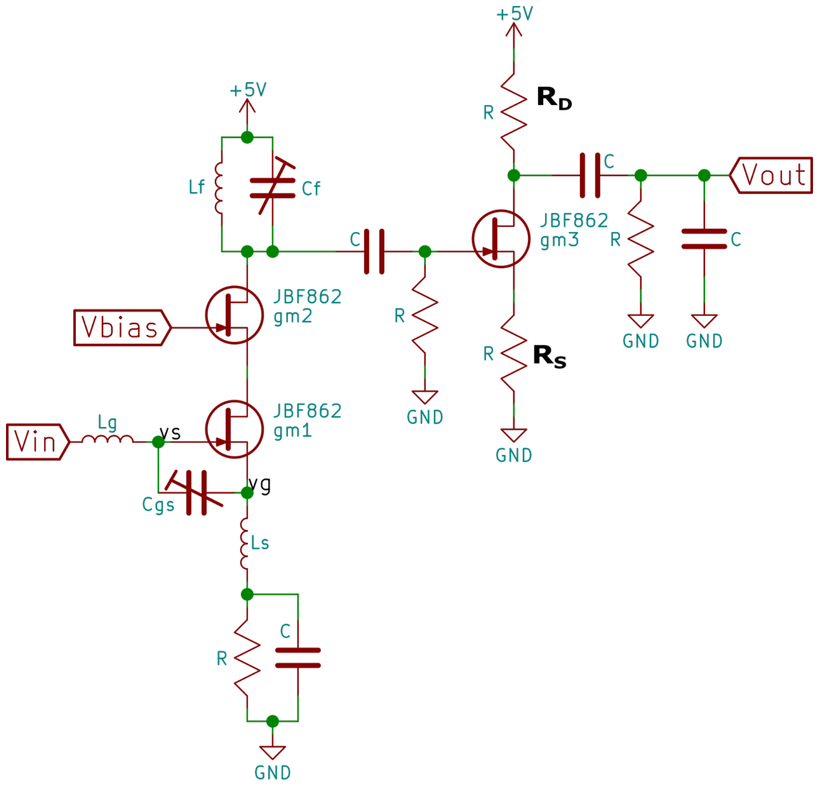

Figure 2.12. Preamplifier schematic.

In the case of NMR spectrometer, the NMR signal has a very limited bandwidth, therefore the second method is more suitable. One of the commonest reactive impedance matching method is inductive source degeneration, shown in Figure 2.12. The input impedance Z_{in} is derived in Equation 2.17 and Equation 2.18.

$$v_s = v_g\left(\frac{g_m + Y_{gs}}{g_m + Y_{gs} + Y_s}\right), i_{in} = Y_{gs}\left(v_g - v_s\right) \Rightarrow Z_{in1} \equiv \frac{v_g}{i_{in}} = Z_s + Z_{gs}(1 + g_m Z_s) \qquad (2.17)$$

$$Z_{in} = Z_g + Z_{in1} = Z_g + Z_s + \frac{(1 + g_m Z_s)}{sC_{gs}} = s\left(L_g + L_s\right) + \frac{1}{sC_{gs}} + \frac{g_m L_s}{C_{gs}} \qquad (2.18)$$

We can see the first two part of the final equation has s terms while the third one has only a real part. If we operate our circuit in a resonance frequency, i.e., when $\omega = \frac{1}{\sqrt{(L_g+L_s)C_{gs}}}$, then Z_{in} is equal only to $\frac{g_m L_s}{C_{gs}}$. For RF purpose, ideal Z_{in} is 50 Ω. This is simply done by first choosing g_m,L_s, and C_{gs} to achieve 50 Ω as real component of Z_{in}, and then choosing an L_g value so that at the operating frequency, the imaginary impedance is equal to zero.

There are three JFETs to provide amplification to the signal. The first two forms cascode amplifier with its output connected to an LC circuit. The output impedance of the first stage amplification is defined as

$$Z_{o1} = \left(g_{m2}\ r_{ds2}\ r_{ds1}\right) \parallel \left(\frac{1}{sC_f} + sL_f\right) \approx \frac{1}{sC_f} + sL_f \tag{2.19}$$

If g_{m2} is high enough, then Z_{o1} is only dominated by L_f and C_f. Bandwidth of the first stage amplification is set by the quality factor of the LC circuit. Its frequency of operation is simply $\frac{1}{2\,\pi\sqrt{L_f\ C_f}}$. Gain of the first stage amplifier is then

$$A_{v1} = g_{m1}\ Z_{o1} = g_{m1}\ \left(\frac{1}{sC_f} + sL_f\right) \tag{2.20}$$

Its output is then filtered to a first-order RC low-pass filter, before it's fed to another gain stage. The second stage has output impedance of

$$Z_{o2} = R_{ds3}||R_D \approx R_D \tag{2.21}$$

Hence, the output impedance Z_{o2} is in most cases equal to only R_D. The transconductance of second stage G_{M2} is

$$G_{M2} = \frac{g_{m3}}{1 + g_{m3}R_S} \approx 1/R_S \tag{2.22}$$

where the approximation is valid when $g_{m3}R_S \gg 1$, which is usually the case. The gain is then defined as

$$A_{v2} = \frac{v_{out2}}{v_{in2}} = G_{M2}Z_{o2} \approx R_D/R_S \tag{2.23}$$

There's another RC low-pass filter after the second-stage gain, but for simplicity is not added here. The total gain of the preamplifier is then defined as

$$A_{vtot} = A_{v1}\, A_{v2} = g_{m1} \frac{R_D}{R_S} \left(\frac{1}{sC_f} + sL_f \right) \tag{2.24}$$

As already mentioned above, maximum gain can be achieved at resonance of the LC circuit, predicted by the equation above. The preamplifier is simulated in LTSpice to see its performance characteristics. The result is shown in Figure 2.13. The maximum gain of the circuit is 59.2dB at 4.3 MHz, while the minimum noise figure is 0.04dB.

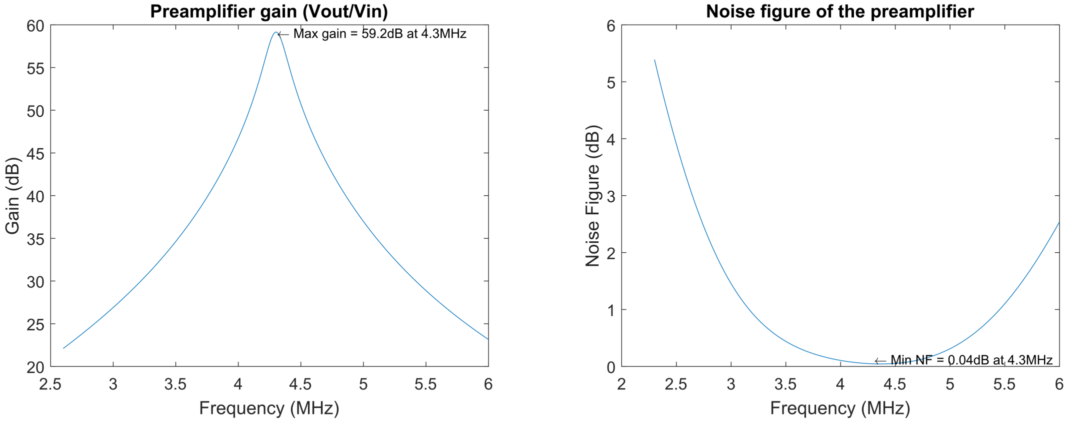

Figure 2.13. Preamplifier gain (a) and noise figure (b).

2.6 Second Gain Stage Design

The second gain stage provides additional amplification for the amplified NMR signal coming from the preamplifier. The second stage design doesn't need to have a very good noise figure, therefore can be implemented using Op-Amps. The schematic of the second-stage amplifier is shown in Figure 2.14. This stage consists of four LT6231 Op-Amps. The first Op-Amp is used as an unity-gain input buffer. The second and third Op-Amp have fixed gain of 10, approximately. The last Op-Amp has configurable gain

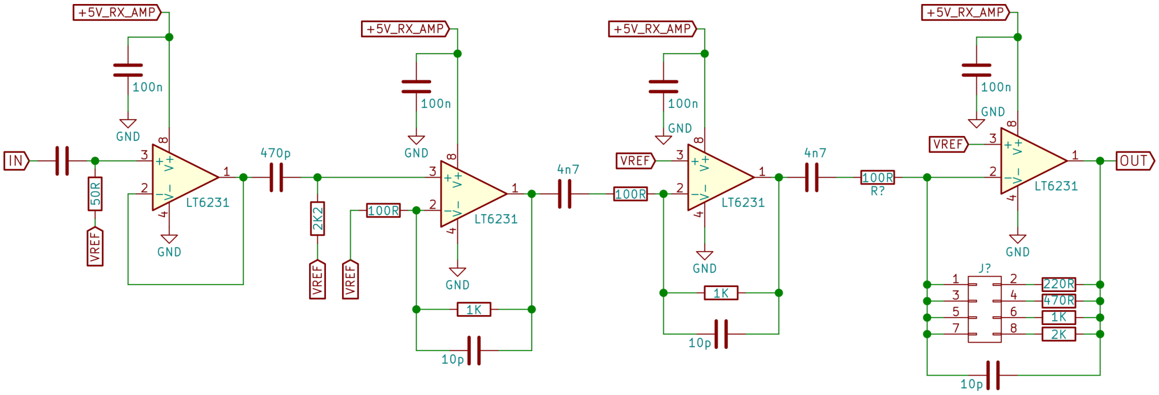

Figure 2.14. Second stage amplifier schematic.

from 1 to 20, depending on the resistance activated by the jumper. The total gain is from 40 dB to 66 dB. This configurable gain will be chosen depending on the amount of signal received by the coil sensor, in order to have as large signal as possible without saturating the voltage at the output of the last stage gain.

Some RC high-pass filters are implemented at this stage to minimize noise generated by the Op-Amps, to reduce low-frequency recovery time of saturated Op-Amps caused by transmit operation, and to reduce the low-frequency ring-down effect after transmit operation.

2.7 Duplexer Design

Duplexer purpose is to isolate the receiver part during transmit pulse, in which the voltage is usually really large. Due to the presence of $50\,\Omega$ impedance at the input and output of the duplexer, it is relatively a broadband circuit.

The duplexer design used in this project is based on passive lumped circuit duplexer of L.A. McLachlan[15]. The duplexer design is shown in Figure 2.15. When RF transmit pulse is present, D_1 and D_2 conduct current. L_1 and C_1 are designed so at the frequency of operation, i.e. the Larmor frequency of water, the impedance is maximum (parallel LC circuit), therefore acts as an isolation circuit when the signal is in transmit mode. At the same time, D_3, D_4, D_5, and D_6 also conduct current to the ground. This limits the voltage at the receiver when the transmitter is pulsing large signal, and also to minimize ring-down after transmit mode. Receive mode of the duplexer is activated when the signal is small enough, therefore D_1 to D_6 are not conducting current. In this case, the coil is also isolated from the transmitter. L_1 and C_2 are designed so that at frequency of operation, its impedance is minimum (series LC circuit), therefore acts as a low impedance path for the low level signal, i.e. the NMR signal.

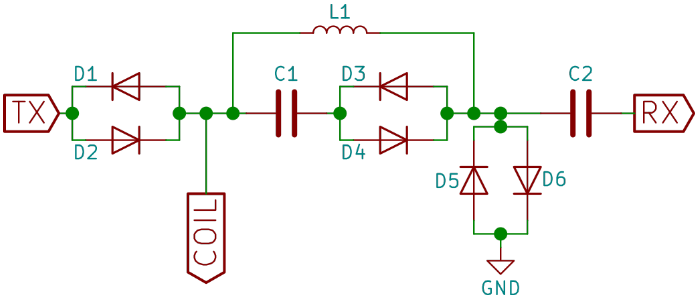

Figure 2.15. Matching network schematic.

The performance of the duplexer is shown in Figure 2.16. From the figure, it can be seen that during transmit pulse, the duplexer output voltage is clamped at 0.8V, shown in Figure 2.16 (b), even though the duplexer input voltage is close to 40V peak-to-peak range, shown in Figure 2.16 (a). The time it takes for the ring-down to get down to 5% of the original voltage is about 20μs. While during receive mode, shown in Figure 2.16 (c), the gain of the duplexer is approximately 8 dB, while the transmit signal is isolated from the receive path. We can see that the Q of duplexer is low, due to the existence of 50 Ω impedance at the input, i.e. the coil, and also 50 Ω at the output, i.e. the preamplifier, resulting in a large bandwidth also shown in Figure 2.16 (c).

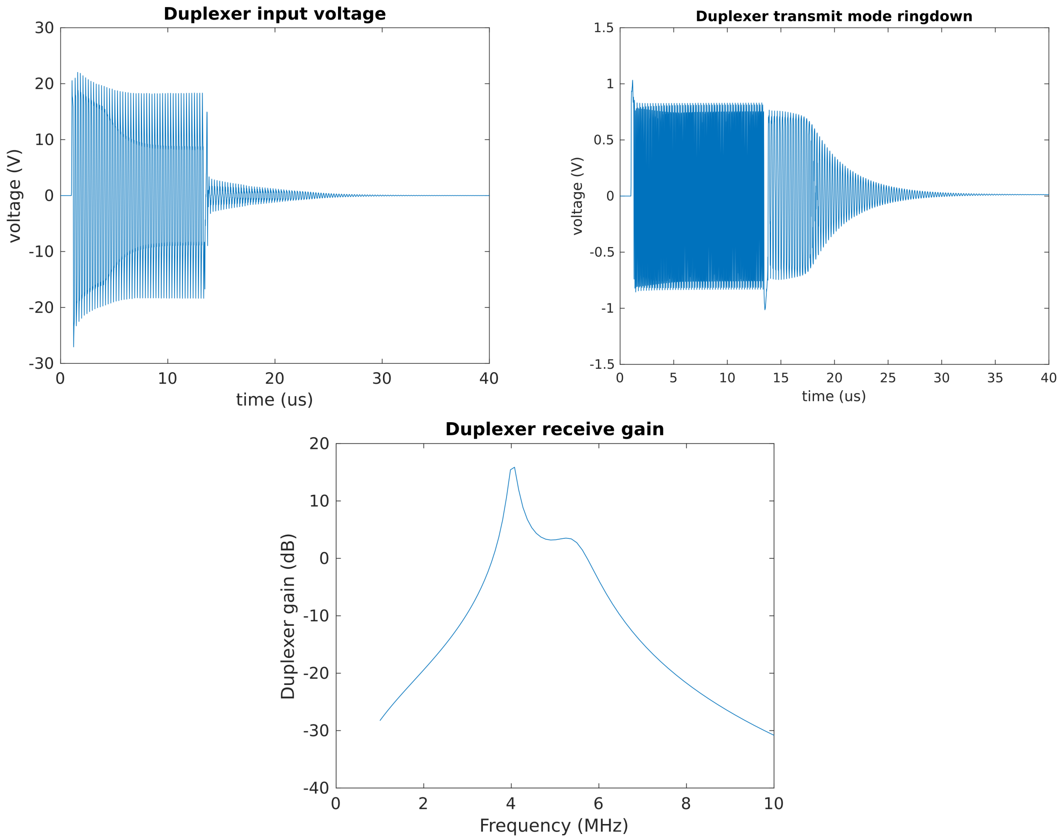

Figure 2.16. Duplexer ringdown performance. Duplexer input voltage during transmit (a). Duplexer output voltage during transmit (b). Duplexer gain during receive (c)

2.8 ADC and Bandpass Sampling

For signal digitization, LTC2314 is used. LTC2314 is a 14-bit, 4.5Msps, serial sampling Successive Approximation Register (SAR) Analog-to-Digital converter (ADC). It is the fastest serial ADC in the market by the time of the project, therefore chosen. It is driven with SPI clock frequency of 87.5 MHz to operate at its maximum sampling operation. This ADC is driven by the FPGA directly and the digitized data is stored in FIFO memory. Its block diagram is shown on Figure 2.17. Solid lines mean timing-sensitive path, while dotted lines mean timing-insensitive path. The ARM processor needs to set frequency the output frequency for custom SPI driver. In order for the SPI to operate at 87.5 MHz corresponds to 4.5 Msps operation, the PLL output frequency needs to be exactly twice of the SPI clock, i.e. 175 MHz. We can adjust its frequency for the ADC to sample slower than its maximum speed. Sample number parameter sets the number of samples taken by the ADC for every start signal given. Delay number represents the number of clock cycles the driver needs to wait after start signal is given, before it samples data. The data is sent to a FIFO memory, where later on can be fetched by the ARM processor through the AXI bus.

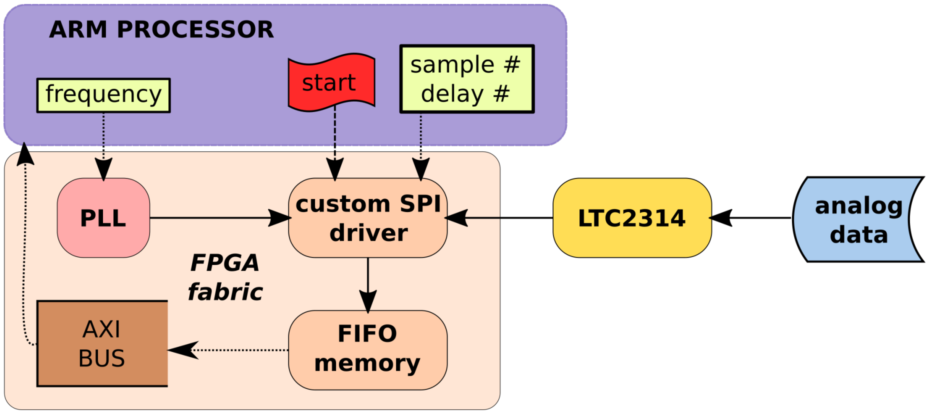

Figure 2.17. ADC driver

As 4.5 Msps ADC is used to digitize RF signal around 4.3 MHz, bandpass sampling[16] technique needs to be implemented. Bandpass sampling theory suggests that a signal can be digitized at frequency lower than its Nyquist sampling frequency, as long as the sampling rate is higher than the bandwidth of the signal at carrier frequency. The formula is shown as follows

$$\frac{2f_c - B}{m} \geq f_s \geq \frac{2f_c + B}{m+1} \tag{2.25}$$

with f_s is sampling frequency, f_c is carrier frequency, B is bandwidth of the signal at the carrier frequency, and m is an arbitrary, positive integer number ensuring that $f_s \geq 2B$.

The bandwidth itself depends on the length of transmit signal. In this experiment, transmit signal length is chosen to be 7μs for initial approximation. Length of transmit signal, however, is not fixed and highly dependent on different experiments along this project. The bandwidth is then approximated at 150kHz for the 7μs transmit signal length, with 4.3MHz RF carrier frequency f_c and 4.5 Msps sampling rate f_s. Figure 2.18 shows allowable sampling frequency to avoid aliasing. It is found that 4.5 Msps is within the allowable frequency range, therefore the ADC can be used in its maximum sampling frequency rate. s

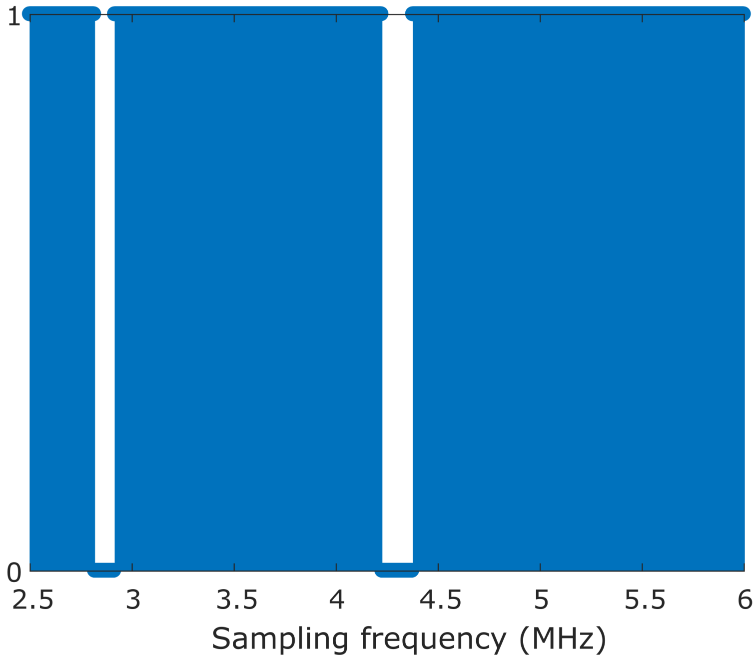

Figure 2.18. Allowable sampling frequency with bandpass sampling technique.

Additional signal post-processing needs to be done for the digitized data, to bring the signal from its corresponding frequency, i.e. $f_s - f_c$, to DC with quadrature down-conversion technique and low-pass filter. This technique is later used to quantify the T_2 time of NMR signal.

2.9 NMR Controller Design with DE1-SoC

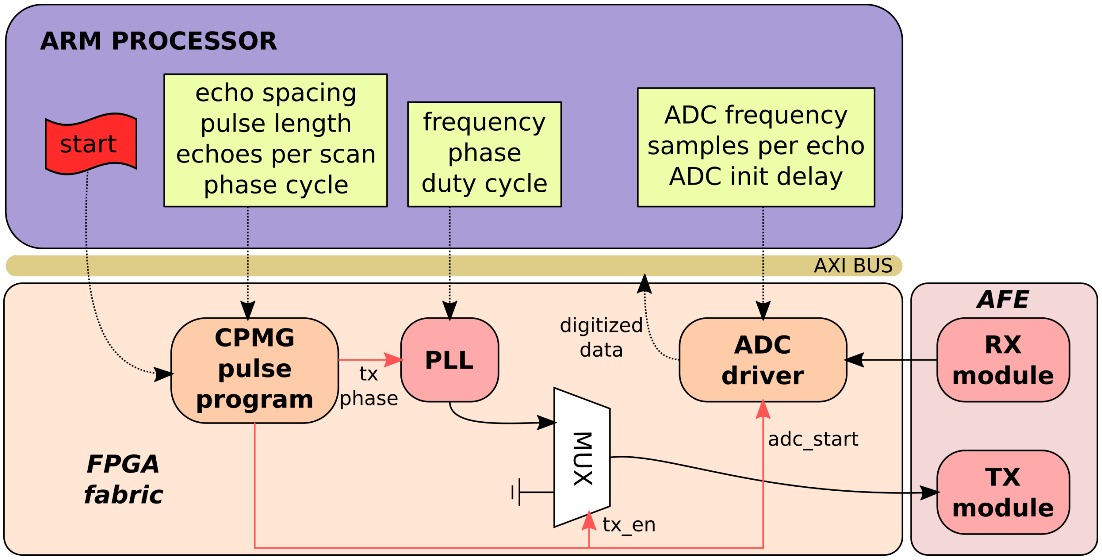

Figure 2.19. NMR FPGA controller system block diagram on DE1-SoC board.

The NMR FPGA controller system block diagram is shown in Figure 2.19, consists of CPMG pulse program, ADC driver, and some other interconnects between them. The NMR FPGA controller has to be built with some levels of configurability, for example one needs to change its operating frequency, sampling frequency, duty cycle, delay parameters, phase cycle, number of samples, etc. These parameters are sent to the NMR controller before start signal is given by the NMR *C* program, implemented inside the ARM processor in the block diagram. These parameters are captured by the NMR FPGA controller when it starts its operation, and keeps its value until the operation is done. In Figure 2.19, these parameters are shown with dotted line coming from the ARM processor to the FPGA fabric, through the AXI bus.

A CPMG pulse program is hard-coded into the the FPGA fabric. It needs several configurable parameters to run, including pulse length for π-pulse and $\pi/2$-pulse, echo spacing, echoes per scan, and phase cycle. There are some other parameters given to the

CPMG pulse program, but they are not shown here as they can be derived by all these parameters, using the NMR *C* program. The timing for *tx_phase*, *tx_en*, and *adc_start* are all derived by CPMG pulse program. While all parameters are not timing sensitive, shown with dotted lines, the *tx_phase*, *tx_en*, and *adc_start* are timing sensitive, therefore they are implemented in the FPGA fabric.

The PLL needs frequency, phase, and duty cycle as parameters before it runs. The PLL is configured with Altera PLL Reconfig module given by the QSYS, which is configurable through its registers which can be accessed with AXI bus. There are two PLLs in the block diagram, The PLL on transmit (TX) side is used with the transmitter unit, and PLL on receive (RX) side, implemented inside ADC driver, is used to generate clock used by the SPI driver to drive the ADC.

The FPGA transmit driver only consists of a PLL block, several multiplexers abstracted with one multiplexer block, and some registers to hold its parameters. The PLL block generates four phases from a single frequency, with $\pi/2$ phase difference between each signal. To avoid confusion with nutation flip angle, i.e. $\pi/2$-pulse and π-pulse, from now on these 4 different pulse phases will be called {x, y, -x, -y} to represent the common phase phase values, i.e. {0, $\pi/2$, π, $3\pi/2$} respectively. These four phases are reduced to two differential transmit signal, which is a combination of {x,-x} or {y,-y}, chosen in real time by CPMG pulse program using *tx_phase* control signal. The output of the PLL is then gated with multiplexer, also controlled in real time by the CPMG pulse program, using *tx_en* control signal. *Tx_en* signal is also synchronized internally within the multiplexer with a register, to ensure that the output phase is always the same between different measurement.

The ADC driver block needs some parameters, including samples per echo, ADC init delay length, and ADC frequency. Its functionalities and purposes has already been discussed in the previous section. It can capture series of ADC data, while adding some delay at the beginning after start signal is given by the CPMG FSM block. This feature is added and very important to ensure that spin echoes generated by the CPMG sequence are captured at the proper times. All the data captured by the SPI block is then passed to the FIFO buffer, implemented with Avalon-ST Single Clock FIFO block inside QSYS. Avalon-ST Single Clock FIFO uses on-chip memory (M10K block) of the FPGA. It is the fastest memory block provided by the DE1-SoC. The maximum memory it can provide is 150KB size. This is the amount of data limitation to the system, as we can not rely on the speed of the AXI bus to store this timing-critical data. It is possible to increase this memory limitation by using on-board SDRAM available in DE1-SoC board and to use on-chip memory only as the input buffer for the SDRAM. However, in this project it is not done yet and the idea will be kept for future work.

All parameters are given by the NMR *C* program, which are sent to the FPGA fabric through AXI bus. At the FPGA fabric side, these parameters are implemented using Parallel I/O (PIO) module with its slave Avalon-Memory-Map interface. In the other hand, to take the data, a slave AXI bus is available in the FIFO buffer, therefore the data can be captured by the NMR *C* program through the same AXI bus, after the CPMG pulse program finishes its operation.

3 Spectrometer Performance Characteristics and Measurements

In this chapter, performance of first NMR spectrometer is quantified. It consists of characterization of the sensor, transmitter, preamplifier, second-stage amplifier, ADC, and functionality of the FPGA. At the end of this chapter, water is used as a sample to see whether the NMR spectrometer can sense the NMR signal coming from it.

3.1 Coil Sensor Performance

First, the coil sensor is connected to the matching network, and its performance is tested using an impedance analyzer. C_1 and C_2 in Figure 2.6 are implemented with several mica capacitors and a variable capacitor. Variable capacitors are placed in parallel with mica capacitors for capacitance configurability purpose. Figure 3.1 shows the reflection coefficient of the matching network. Here we can see the minimum reflection is at 4.3 MHz with value of slightly smaller than -40dB. The -10dB bandwidth is around 110kHz. Therefore, the minimum pulse length allowed is around 9.1μs.

The variable capacitors used for C_1 and C_2 have 1-20 pF variability. This is a low value given that C_1 is 3nF and C_2 is 166 pF. Practically, this low variability variable capacitors are not very useful to vary the operating frequency. It can only vary the operating

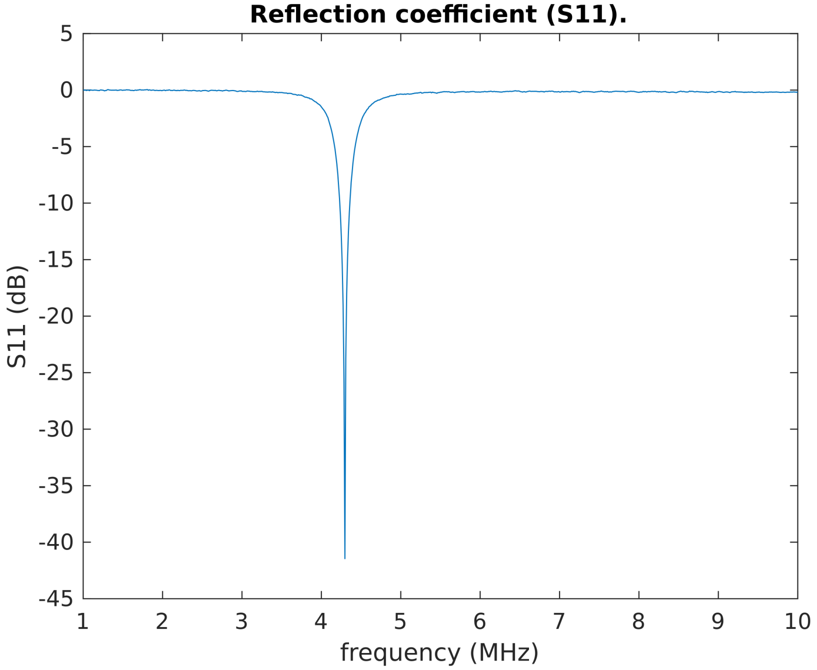

Figure 3.1. Matching network reflection coefficient.

frequency by tens of KHz, at around 4.3 MHz operational frequency. For the new design, it is helpful to use higher variability variable capacitors or to use different method.

3.2 Transmitter Performance

In this design, transmit power can be controlled with pulse duty cycle. The relationship between output voltage and the amount of duty cycle is given with this equation

$$V_{out} = \frac{4}{\pi} V_{DD}\, sin(\pi\, D) \tag{3.1}$$

with D is defined as duty cycle, and V_{out} is defined as the output voltage at the coil. This equation can be easily derived from the Fourier series expansion of a square waveform with duty cycle D. V_{out} is the fundamental component of the square voltage waveform generated by the FPGA, which is then fed into the power amplifier. Thus, the final output voltage across the coil is proportional to V_{out}. The other way to see the V_{out} is by imagining a bandpass filtered FPGA output voltage after being amplified by the transmitter gain. As the equation predicts, it still follows the characteristic of sinusoid signal from 0 to $\pi/2$. It's output voltage is linear from 0 to some angle, and then it starts to saturate.

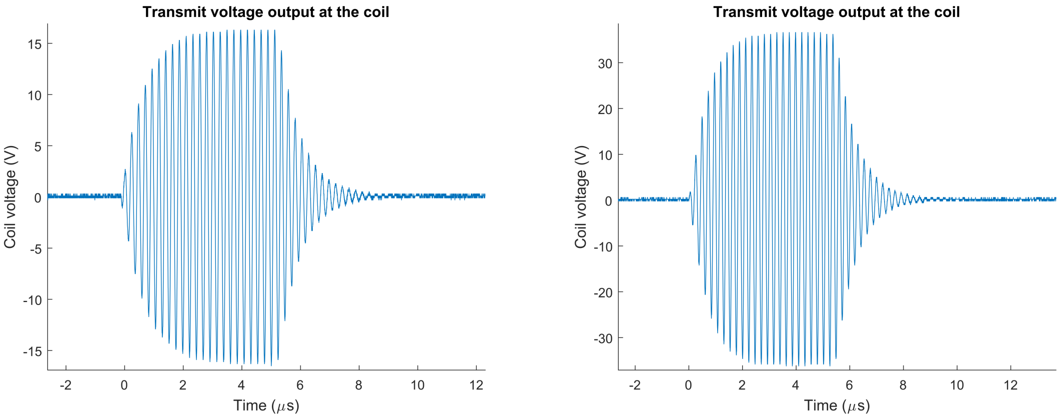

Figure 3.2. Coil output voltage during transmit mode at 5% duty cycle (a) and 50% duty cycle (b).

With this measurement, pulse duty-cycle is swept from 1% to 50% and voltage at the coil is measured using an oscilloscope. Figure 3.2 (a) and (b) shows the coil voltage with 5% duty cycle and 50% duty cycle, respectively. It can be seen that the peak voltage output at the coil with 5% duty cycle is less than it is at 50% duty cycle. A diagram showing the voltage peak-to-peak with different duty cycle parameter is shown in Figure 3.3. The voltage-peak-peak is generally increasing linearly with duty-cycle from 1% to 30%. After 30% duty cycle, the peak-to-peak voltage increases slightly with increasing duty-cycle, predicted by the equation above. The predicted and measured curve are slightly different due to the finite duty cycle step generated by the PLL in the FPGA. Therefore the output duty-cycle is not really precise.

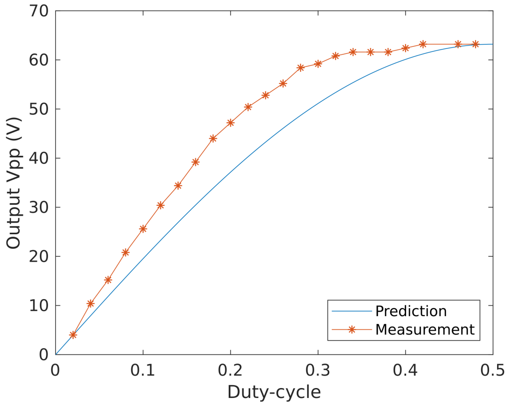

Figure 3.3. Coil voltage peak-to-peak with duty-cycle.

3.3 Preamplifier and Second-Stage Amplifier Performance

In this section, preamplifier and second-stage amplifier perfomance are measured. First, preamplifier reflection coefficient is measured using network analyzer. Figure 3.4 shows the result. The minimum reflection is at 4.184MHz. The S11 of preamplifier is less than -10dB at 4.015 MHz to 4.375 MHz. Therefore, the -10dB bandwidth of the preamplifier is 360kHz. This bandwidth is larger than the bandwidth of the coil sensor, so the coil sensor is still the limitation of the bandwidth of the system.

The receiver gain path is also measured using network analyzer. Working with network analyzer, optimum measurement can be done if input and output impedance of the circuit is 50 Ω. The input impedance of preamplifier is matched with the network analyzer. However, the output impedance of the preamplifier is not 50 Ω. Therefore, big reflection occurs if preamplifier output is connected directly to the network analyzer, where the gain is measured to be almost zero. For that reason, output of preamplifier is connected to second-stage amplifier and output of second-stage amplifier is used to measure overall gain. Third Op-Amp output, shown in Figure 2.14 is used to get the overall receiver gain, shown in Figure 3.5.

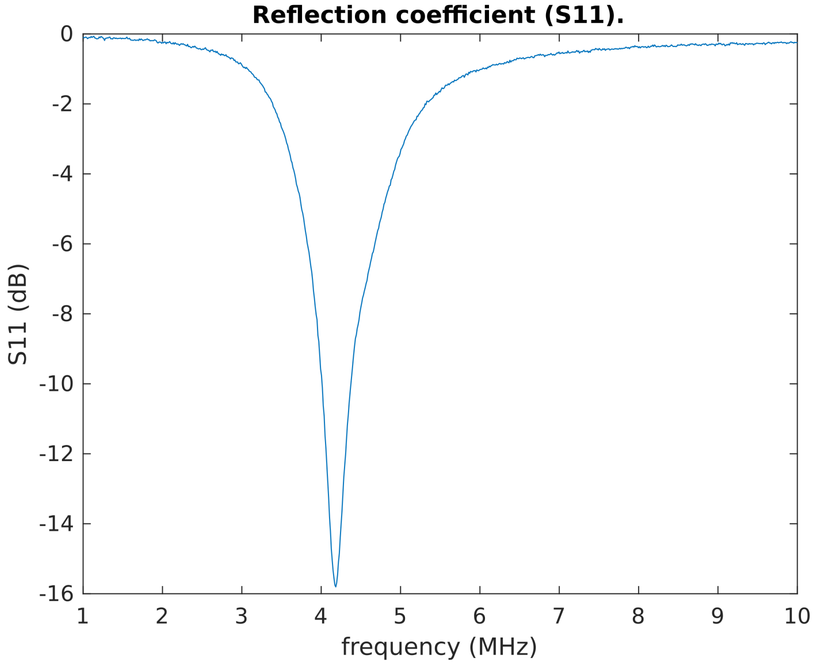

Figure 3.4. Preamplifier reflection.

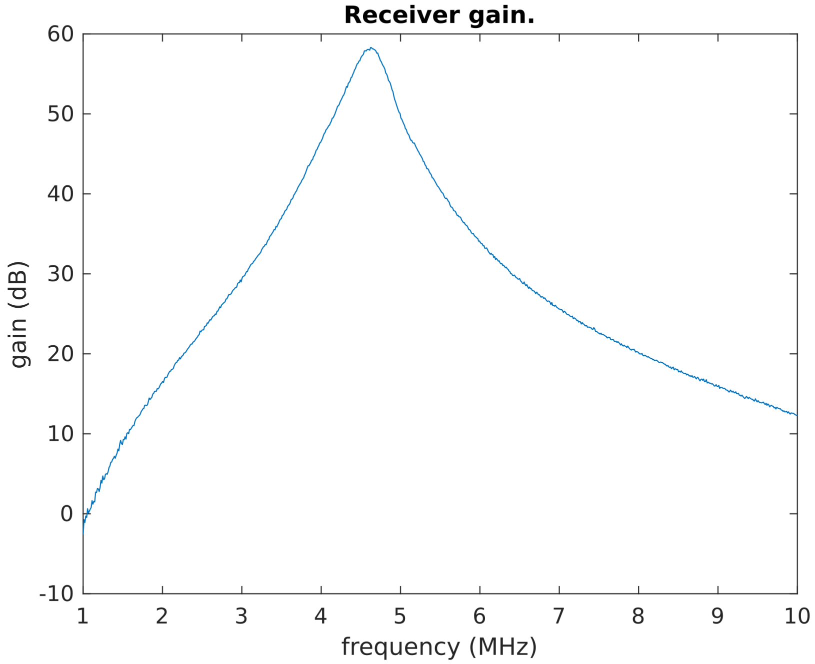

Figure 3.5. Receiver gain.

Source power of -50 dBm is generated from network analyzer. Another -20 dB attenuator is also used at the source, resulting -70 dBm source power, to avoid saturation at output of the second-stage amplifier. Even then, last stage receiver Op-Amp is already saturated and the third Op-Amp output stage is used as output port for the network analyzer. Therefore actual overall gain at the receiver is larger by 0dB to 16dB than what is shown here, depending on final gain at the fourth Op-Amp stage. The gain shown by the figure is 58.3dB at 4.62MHz. Compensating the -20dB attenuator, this means 78.3dB gain at 4.62MHz. The maximum gain after last stage Op-Amp is then 78.3dB to 84.3dB. At the desired frequency of operation, i.e. 4.3MHz, the gain is about 53.4 dB, or 73.4 dB after compensating the -20 dB attenuator. The total gain achievable including the last stage Op-Amp is between 73.4 dB to 89.4 dB. There's still a caveat in this situation, as the output impedance of third Op-Amp is not 50 Ω as well, so reflection may occur in this case. So the total gain for the actual NMR signal is relatively larger than what is shown here.

3.4 Duplexer Performance

Duplexer performance is measured by probing its output voltage when transmitter is pulsing signal. In this case, duplexer input voltage is huge and duplexer performance is quantified by measuring voltage at its output. The output of duplexer is shown in Figure 3.6 (b). It can be seen from the figure that the duplexer diodes clamp the large transmit voltage from 40V peak-to-peak around 0.8V peak-to-peak. In this case, the duplexer effectively isolates the receiver when system is in transmit mode. The ring-down of the coil sensor is also shown here from 7μs to 15μs. Ideally, we need to wait for the ring-down voltage to decrease below NMR signal voltage, around 1-2 μV, before we sample the signal.

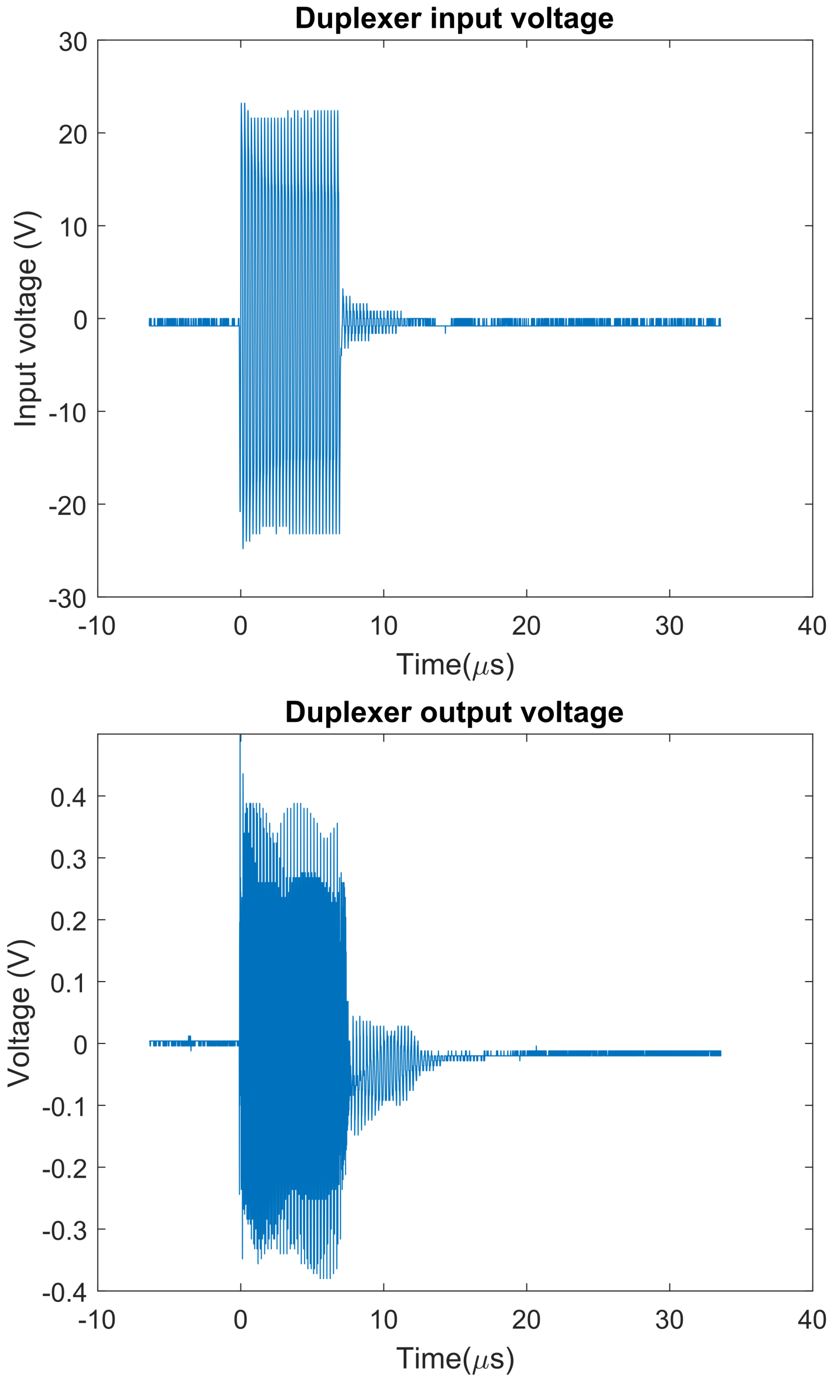

Figure 3.6. Duplexer voltage during transmit mode.

The signal shown in the Figure 3.6 (b) is amplified by the preamplifier. Overall signal at receiver output is shown in figure 3.7. It can be shown that the ring-down time is approximately 23 μs. From 0 μs to around 20 μs, various Op-Amps used in receiver are saturated and recovered simultaneously. Therefore, signal during this period does not seem to be sinusoidal. After 23 μs, only small voltage fluctuation can be seen. If there is an NMR signal, it is buried within noise in this region. This is the region we are interested to digitize and analyze further.

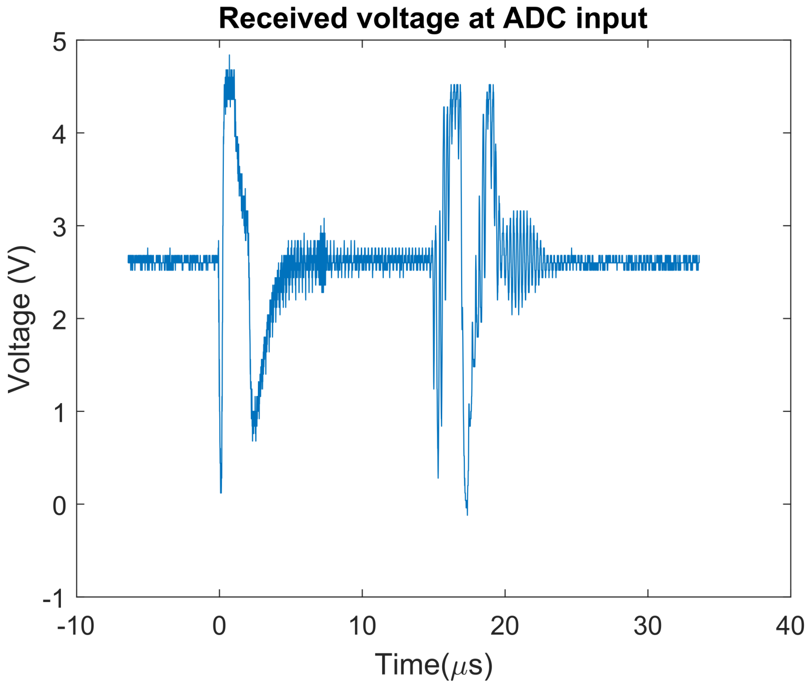

Figure 3.7. Receiver output voltage during transmit mode.

3.5 The NMR C Program

At this point, the FPGA NMR controller design has some input parameters, i.e. $\pi/2$-pulse length, π-pulse length, ADC delay length, echoes per scan, samples per echoes, etc.

These parameters give some freedom for implementing basic functions of an NMR spectrometer. However, it is hard for human to understand as all parameters are currently in integer values, counted with clock cycles by a finite state machine (FSM). Therefore, a high level language is used to incorporate a high level input for the system. Here, an ARM processor under Ubuntu Linux system is used and the program is written in C. The main function of the NMR *C* program is to interpret human understandable parameters to their particular representations used in the FPGA NMR program. The main function is shown below

```
CPMG_Sequence (cpmg_freq, pulse1_us, pulse2_us, pulse1_dtcl, pulse2_dtcl,
echo_spacing, echoes_per_scan, samples_per_echo, file_name, data_sum_name,
enable_message)
```

The function parameters are derived from CPMG sequence timing, which is shown in Figure 3.8. The first one is *cpmg_freq*, defined as transmit pulse frequency generated

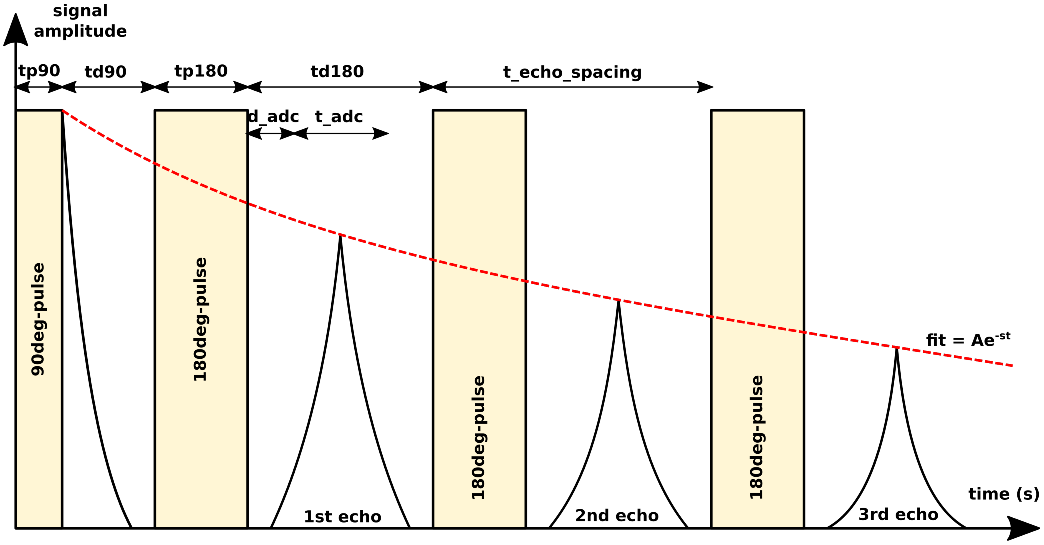

Figure 3.8. CPMG sequence timing.

by the PLL. This also corresponds to NMR data frequency sensed with coil, and therefore is reported as a parameter for MATLAB signal processing later on. This input frequency, written in a floating representation to ease the user, is translated by NMR *C* program to some integer parameters needed by the PLL module in the FPGA. *Pulse1_us* and *pulse2_us* is the length of $\pi/2$-pulse and π-pulse, shown in the figure as $tp90$ and $tp180$, respectively. One may realize that delay between pulses, i.e. $td90$ and $td180$ can be computed using $\pi/2$-pulse length, π-pulse length, and the echo spacing. It is also done by the NMR *C* program. Duty cycle parameters, i.e. *pulse1_dtcl and pulse2_dtcl*, also written in a floating representation of 0 to 0.5, are computed and derived to another integer parameters used by the PLL module. *Echo_spacing* parameter is time-spacing between echoes, shown as *t_echo_spacing* in the figure, and written in microseconds unit. It is translated by NMR *C* program to an integer number used by the clock counter in the FPGA NMR program. *Echoes_per_scan* is the number of echoes taken in a CPMG sequence, already written in integer. *Echoes_per_scan* also corresponds to number of π-pulse signal repetition within one scan. In the figure, the number of echoes is 3. *Samples_per_echo*, defined as number of samples taken in one echo, is given to ADC driver to capture a series of samples per echo data if a start signal is given by the controller, as already discussed in previous chapter. The NMR *C* program also computes the delay of ADC, shown as d_adc in the figure, by putting the ADC acquisition window, shown as t_adc, exactly in the middle of $td180$. *File_name* is the name of raw data file to be saved to the disk. This NMR *C* program also does basic signal processing the raw data to see whether an NMR signal is present, by summing echoes in one sequence and save it under *data_sum_name*. This is important in development phase as the signal can be easily seen in its echo sum form. Lastly, *enable_message* gives status of the program, computed parameters, and debugging informations for the user to see.

There are some limitations derived with these parameters in mind. First, product of *echoes_per_scan* and *samples_per_echo* is limited by the length of FIFO buffer. Maximum length of FIFO buffer is limited by M10K memory blocks in the FPGA. In the case of Terasic DE1-SoC board, available M10K blocks provide maximum data of 128K*16bits, or 128K samples of 16-bit data. Therefore, the product of *echoes_per_scan* and *samples_per_echo* is limited to only 128K. Interestingly, the Magritek Kea2, which is a well-known commercial desktop spectrometer used in our lab, also has same buffer limitation. Second limitation is derived the ADC acquisition window, where t_adc can not be larger than $td180$. This also means product of *samples_per_echo* and ADC sampling time (1/4.5MHz = 222ns for LTC2314) has to be shorter than $td180$ window. During initial development, constraints based on these limitations were not implemented and resulted in system crash. For this reason, these limitations are used as constraints in the NMR C program, which shows an error if one or more constraints are violated. It is also possible to implement these constraints in FPGA hardware, as it is safer in a way that the FPGA NMR controller won't do its operation unless all parameters are within constraints. However, it turns out to be harder to manage due to long compile time during development phase and idea was not implemented but is kept for future work.

3.6 The NMR Experiment

Before we go to the NMR experiment, it is useful to introduce the phase-cycling concept. Phase-cycling is a method to minimize the effect of transmit signal ringdown in the measurement. For normal NMR measurement without phase-cycling, {x,-x} pulse is used for $\pi/2$-pulse, and {y,-y} pulse is used for π-pulse. In this case, echo from different measurements has the same phase. By changing the phase of $\pi/2$-pulse, we can also change the phase of the echo, shown in Figure 3.9. The ringdown is shown by declining line after every $\pi/2$-pulse and π-pulse. Figure at the top uses {x,-x} for $\pi/2$-pulse, and figure at the middle uses {-x,x} for $\pi/2$-pulse. We can see the echo from these two measurements has 180 degrees of phase difference, but the ringdown for the π-pulse always has the same phase. We can then subtract the data between two measurements to remove the effect of ringdown.

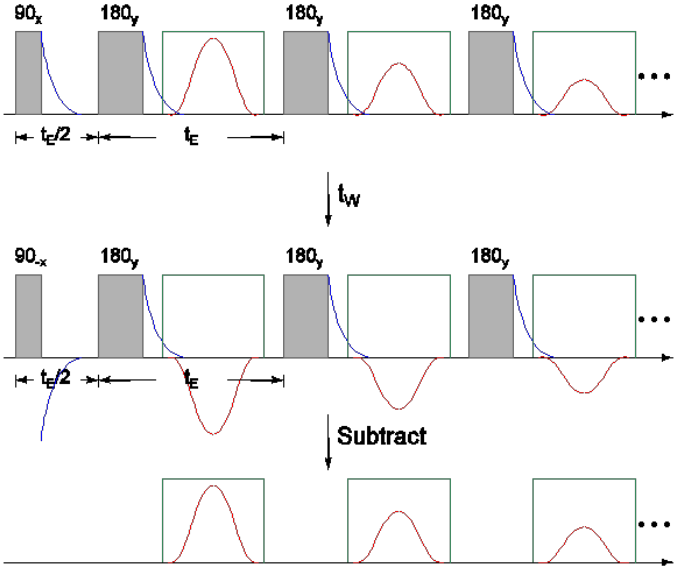

Figure 3.9. Phase cycling.

At this point, the NMR spectrometer is ready to measure an NMR signal. Sample used is water contained in a 5 mm NMR tube. Initially, it is very hard to get any data for different reasons. The FPGA NMR controller itself has a lot of parameters to be tuned and all of them needs to be configured properly to capture an NMR signal. Two important aspects are finding an optimum duty-cycle and transmit pulse length. In order to get a maximum NMR signal, one needs to rotate the proton by $\pi/2$ angle, according to equation 1.6. Initially, all parameters can only be predicted by equations but is not always exact due to external factors. Sum echo direct plot in MATLAB was the first attempt to see the data. The sum echo is computed by NMR *C* program in is stored in a file. Sum echo, in which the NMR signal is present, is found to be easy enough to be seen with bare eye, compared to the raw data. After several trial, first NMR result is found and shown in Figure 3.10. The NMR data was taken with 35% *pulse_1* duty cycle, 50% *pulse_2* duty cycle, 7μs pulse length for both *pulse_1* and *pulse_2*, 250 μs echo-spacing, 256 samples-per-echo, 512 echoes-per-scan, and 4.5 Msps sampling rate.

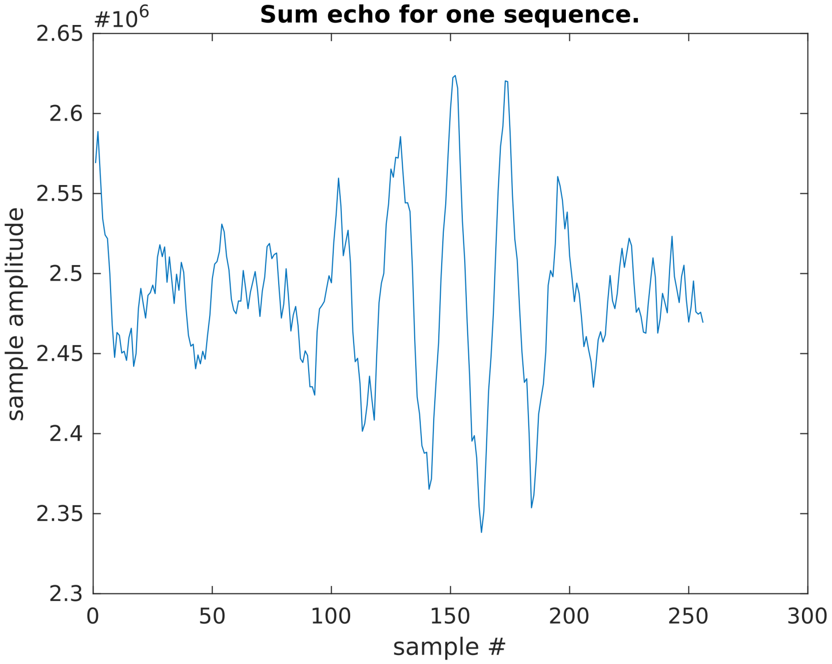

Figure 3.10. NMR echo sum example.

Here we can see an echo sum signal, which has peak around sample number 160. If the sample is removed, no signal is captured, which is one of proofs that signal is coming from the sample. The distance between the peak and the adjacent peak is about 21 to 24 samples, corresponds to the frequency of 214 KHz to 187.5 KHz, respectively. This is roughly frequency difference between data frequency and sampling frequency, which is 4.5 MHz - 4.3 MHz = 200 KHz. Therefore it can be concluded that the signal itself is an NMR signal coming from the sample.

To quantify an NMR signal, some signal processing technique is used and implemented in MATLAB, shown in Figure 3.11. First, the data is summed with phase-cycling and its in-phase and quadrature (IQ) form of data is extracted from it. This is done by down-converting every echo from its original frequency, which is 200 KHz, to baseband. Every echo is then filtered with Butterworth filter to remove its high frequency component. Then, its value is averaged for all the echoes. After that, matched filtering technique is used to find its echo amplitude. This is done by multiplying every single echo by the conjugates of the average value computed before, effectively weighting every single echo with its average value. Its average signal in this case, can be seen as signal with very high SNR, and therefore can be used as weighting window for every echo to reduce its noise. The amplitude of every single echo is then computed from the result.

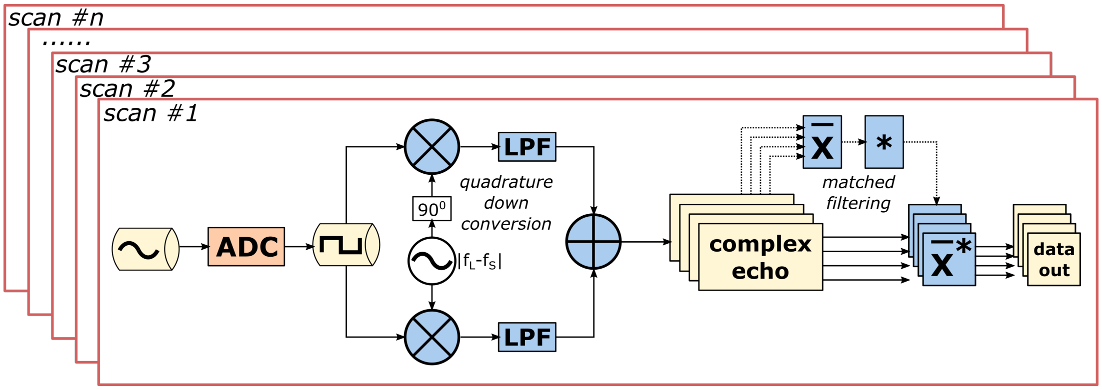

Figure 3.11. MATLAB NMR signal processing.

Figure 3.12 (c) is the average echo shape, which is also used to match filter all the echoes. Figure 3.12 (a) shows raw data after matched filtering, showing an exponential decay. The amplitude of the raw data is then computed and shown in Figure 3.12 (b),

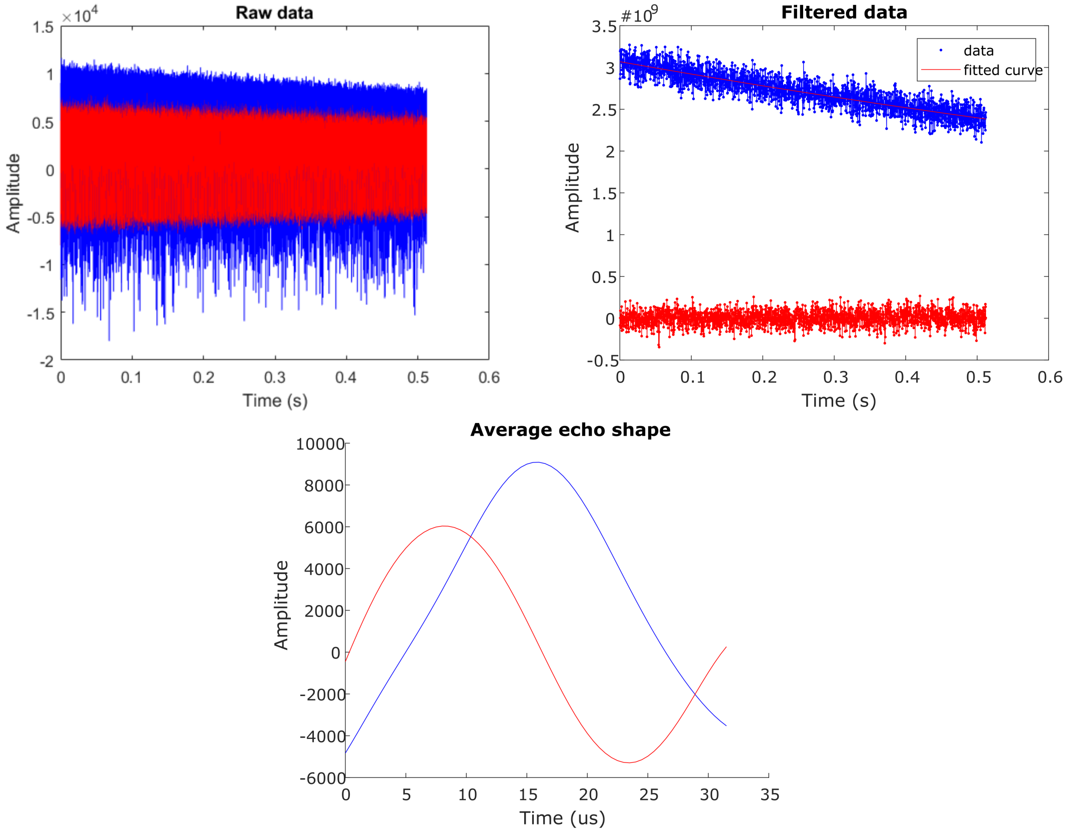

Figure 3.12. NMR raw data (a), NMR filtered data (b), and average echo shape (c).

where the amplitude is fitted within exponential fit. The exponential fit shows initial amplitude of $3.07 * 10^9$ and 2.026 s T_2 time. The SNR per echo per scan is defined as

$$SNR/echo/scan = \frac{A_{init}}{std(noise)\sqrt{total\_scan}} \tag{3.2}$$

where A_{init} is the initial NMR signal amplitude, $std(noise)$ is the standard deviation of imaginary part of the down-converted signal, and $total\_scans$ is the total number of scans. The SNR per echo per scan in this experiment is approximately 5.04.

4 Tunable NMR Spectrometer

This NMR spectrometer design is an upgraded version from the first design. The main improvements in terms of hardware are tunable matching network, a broadband input preamplifier, a tunable preamplifier output bandpass frequency, S11 reflection measurement path, higher speed ADC, configurable receiver gain, and higher power transmitter path. In this section, improvements over the first design are presented in detail.

4.1 Tunable Matching Network

The tunable matching network is developed due to the fact that the bandwidth of coil sensor is very limited, but it needs to cover a range of frequency of operation for different measurements. Tuning and measuring S11 parameter are found to be really tedious, as one needs to solder and de-solder surface mount capacitors before plugging the matching network to measure its S11 parameter. Although it can be solved using a large mechanical variable capacitor, to fully automate the system it is only possible by implementing an electronically controlled tunable matching network. In that case, we choose to use electronic switches as it can be controlled digitally.

The tunable matching network is based on the work of Cheng[17]. It is controlled using reed switches, which controls the connections of several capacitors in parallel, for either C_1 or C_2. With this approach, capacitance value of C_1 and C_2 can be controlled digitally. Figure 4.1 shows the schematic of tunable matching network design. Number of parallel capacitances are chosen to be 8, for both C_1 and C_2. Eight is chosen because it is a typical bus number for digital circuit control. C_1 capacitance options are chosen to be binary value with LSB of 18 pF. C_2 capacitance options are chosen to be binary value with LSB of 2 pF F. Ideally, C_1 is configurable with step capacitance of 18 pF and maximum capacitance of around 4.7 nF, while C_2 is configurable with step capacitance of 2 pF and maximum capacitance of around 630 pF. According to Equation 2.16, the matching network can be tuned approximately from 3.4 MHz to 17 MHz by varying C_1. C_2 value is a bit complicated to be approximated here, as for every frequency it needs to cancel all the imaginary part introduced by Z_1. But one can always use a more precise C_2 value using small capacitance step.

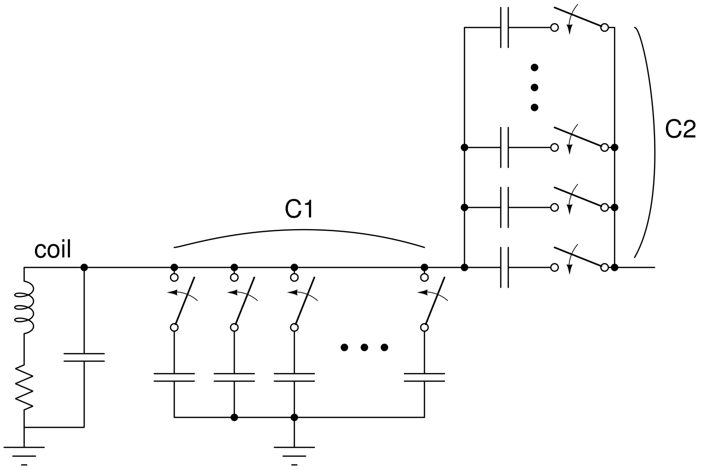

Figure 4.1. Tunable matching network.

Switches are implemented using COTO 9290-05-00, 500mA reed relay switch. To minimize number of GPIOs used to implement control signal for all switches, an I2C 16-bit I/O expander chip called TCA9555PWR is used. TCA9555PWR has 2 ports per chip, with 8 I/Os per port. So the tunable matching network control signal can be implemented with a single chip of TCA9555PWR. In this case, FPGA GPIOs occupied by for tunable matching network control signal is 2, compare to 16 without I/O expander. However, additional hardware in the FPGA is needed to control the I/O expander IC, which adds some more complexities to the FPGA design.

4.2 Reflection Power Measurement Path

Now that our matching network has some configurable properties, it is necessary to measure its performance and automate the process of finding its optimum performance at a given condition. In the case of our matching network, we are interested in finding the lowest reflection coefficient (S11 parameter) at a given frequency. In the first spectrometer design, this is done manually by repetitively tuning capacitance value and measuring its S11 parameter using network analyzer. But we need to find a way to do the measurement directly using the NMR controller. A directional coupler is used to add this particular feature. Directional coupler couples signal that flows in one direction, and not the other. It is being used here to measure reflected power from the matching network. Figure 4.2 shows how it's implemented.

The figure shows a coupling arrow between RFIN and RFCOUP. Coupling factor is one of important characteristic of a directional coupler. In this case, we use MiniCircuits SYDC-20-61VHP+ with -20 dB coupling factor. In general, if there is a signal going from RFIN to RFOUT, -20 dB power is coupled to RFCOUP. In the other hand, if there is a

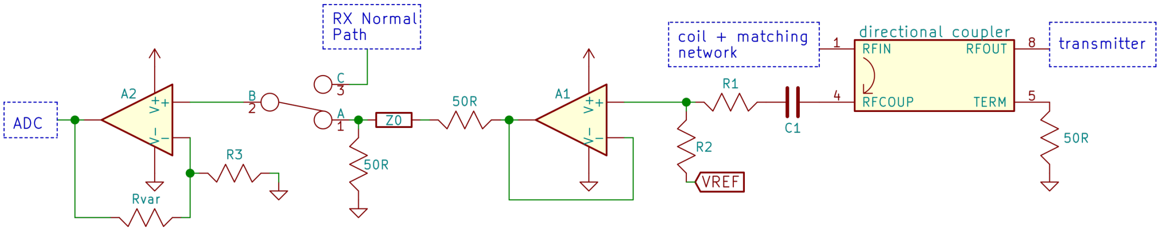

Figure 4.2. Reflection measurement.

signal going from RFOUT to RFIN, no power is coupled to RFCOUP. This concept is very useful to find reflection coefficient of coil sensor.

In our case, transmitter generates pulse signal and the signal propagates from RFOUT to RFIN. None of the power is coupled in this direction. However, in some cases where impedance of the load (coil and the matching network) is not matched to transmitter, 50 Ω in this case, fraction of power of the transmit signal will be reflected back from RFIN to RFOUT. This reflected signal is coupled by directional coupler by -20 dB and is shown in RFCOUP port. R_1-R_2 voltage divider / attenuator is used to decrease coupled signal. Its total resistance should be 50 Ω to ensure an optimum operating condition for the directional coupler. Attenuator is needed because the transmit signal is usually large and therefore needs to be attenuated to avoid saturation of the Op-Amp. The signal is then buffered and transformed into 50 Ω before it's sent to 50 Ω transmission line Z_0. At the other side, the signal is terminated with another 50 Ω resistance and amplified with a variable gain Op-Amp before fed to the ADC to be digitized and analyzed. The same ADC is used for either this or a normal NMR signal measurement. Therefore a switch is implemented to select between normal NMR signal measurement path and reflection measurement path.

4.3 Tunable Preamplifier

Preamplifier in first spectrometer design has limited configurability. Similar to the case of matching network, the preamplifier works at a particular frequency with very limited bandwidth. If the frequency of operation needs to be changed, e.g. in the case of sample change or temperature change, the only mechanism for the preamplifier to adapt is to manually tune variable capacitors, C_{gs} and C_f, shown in Figure 2.12. These variable capacitors only vary between 1pF to 20pF on top of big mica capacitors in parallel with them, therefore their impact to change operating frequency is very small.

C_{gs} and C_f are independent of each other, which is a good thing, but also means we need to configure them one by one. Output gain of the preamplifier depends on the output impedance, set by C_f. Input reflection of the preamplifier depends on input impedance, set by C_{gs}. Therefore these 2 things can be optimum at two different frequencies. If they are not aligned at the designated operating frequency, gain doesn't have a maximum at one frequency, but instead introduces 2 bumps of gain at different frequencies with lower maximum gain, which is not intended.

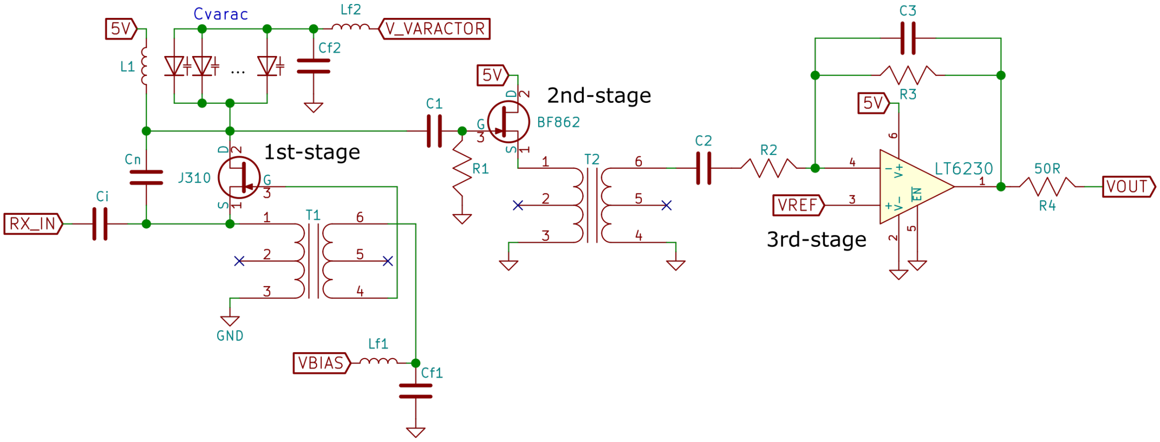

Figure 4.3. Preamplifier schematic.

A new preamplifier design is built to overcome all issues above. In addition, it still needs to have low noise figure and gives a moderate gain level. The schematic is shown in Figure 4.3. At the input side, a common-gate amplifier topology is used to provide a matched input impedance with the source. Its input impedance is simply $\frac{1}{g_m}$, neglecting C_i and C_n at the input. The biggest advantage of using common-gate topology compared to previous one is its broadband input impedance. Once its input impedance is configured to 50 Ω, which is done by changing its bias voltage, it is expected to retain its impedance over a broad frequency. However, to achieve 50 Ω input impedance, we need to bias the transistor and get high g_m. With high g_m, the noise figure will suffer. To get low noise figure, a transformer is used at the input side. The noise figure is then defined as

$$NF = 1 + \frac{g_m\,\gamma}{(A+1)\;R_s\;g_m^2} = 1 + \frac{\gamma}{A+1} \tag{4.1}$$

at the impedance matching condition, i.e., when $g_m R_s = 1$. A is defined as the gain or turn ratio of the transformer. We can see from the equation, the γ is now divided by $(A+1)$ factor, which effectively reduces the noise figure.

At the output side, varactors are used as small signal variable capacitors to tune preamplifier output frequency. Varactor's capacitance depends on the voltage given across it. For this design, 20 varactors are used in parallel to provide enough capacitance corresponds to inductance L_1 being used in parallel with the varactors. To provide bias voltages for varactors and transistor gate, two independent DACs are used. These voltages are individually filtered with first order low-pass filter, to remove external pick-up noise and digital noise coming from DAC. The filter cutoff frequency is designed to be very low, and is implemented using large value of inductor and capacitor available in small form factor. The DAC used for this preamplifier is AD5722R chip, a 12-bit DAC that operates with bipolar supplies. Its output can go from -5V to +5V, which is needed for biasing the transistor gate and also for biasing varactors. It has an SPI interface which can be clocked to 30MHz. However, speed is not a concern here as its only purpose is for providing bias voltage.

With this preamplifier design, we effectively remove the need to configure preamplifier input impedance as it is intrinsically broadband. The varactor bias voltage is the only configurable parameters to change frequency of operation of the preamplifier. It is therefore an easier task for the NMR controller, compared to previous design, to find preamplifier's optimum operating condition at a given frequency.

Two transformers are used in the circuit to give additional signal gain. Transformer is known to work very well to amplify an AC signal, although it is not effective for DC signal. As an NMR signal is an AC signal, transformer is a perfect solution. Here, transformer T_1 is used to give a signal gain at input side. The second transformer gives additional gain after signal is buffered with a common drain / voltage follower transistor, i.e. BF862.

Some additional filters are used in the design. C_1 and R_1 form a high-pass filter to minimize noise bandwidth of the preamplifier. C_3 and R_3 form a low-pass filter also to minimize noise bandwidth of the preamplifier. Output of second transformer is connected to an Op-Amp with a gain of (R_3/R_2), to give an additional gain if needed. A resistor is added at the output of the Op-Amp to set its output impedance to 50 Ω to minimize reflection when it is connected to a transmission line.

The preamplifier gain can be mathematically computed using this equation

$$v_{gs1} = v_{in}(1-(-n_1)) = v_{in}(1+n_1) \tag{4.2}$$

where n is defined as the first transformer turn ratio. It can be seen that the transformer gives a free gain to v_{in} by n+1 factor. The gain from J310 is the similar as already derived in Equation 2.20. Transistor BF862 has a unity gain, as it is used only as a buffer. But at the output, a transfomer of ratio n_2 gives additional gain of n_2 factor. Therefore the total preamplifier gain is defined as

$$A_{vtot} = \frac{1}{2}\, g_{m1}\, (1+n_1)(n_2) \left(\frac{1}{sC_{varac}} + sL_f\right) \tag{4.3}$$

where the 1/2 factor arises from the impedance matching.

The preamplifier design is simulated with LTSpice. Varactors bias voltage is swept from 0V to 4V with 0.5V voltage spacing. Resulting preamplifier gain is shown in Figure 4.4 (a). Note that the gain is probed at the output of second transformer, therefore gain provided by the Op-Amp is not quantified in the figure. The preamplifier maximum gain is almost flat around 51 dB at frequency range of 3.2 MHz to 5 MHz. Reflection coefficient parameter of the preamplifier is less than -14 dB over same frequency range, shown in Figure 4.4(b). The noise figure of the preamplifier is shown in Figure 4.5. The minimum is at 0.7dB at the operating frequency.

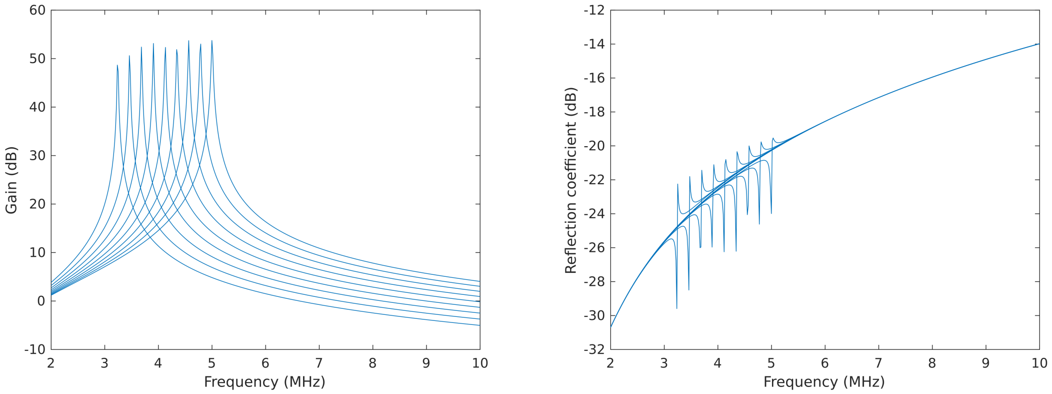

Figure 4.4. Preamplifier gain (a) and S11 parameter (b).

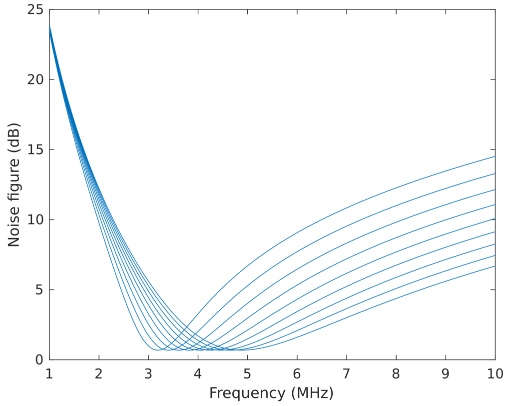

Figure 4.5. Preamplifier noise figure.

4.4 Analog-to-Digital Converters (ADC)

With first spectrometer, data is taken and processed using bandpass sampling technique. After that, data is down-converted to baseband and low-pass filtered to see its NMR features. Frequency difference between sampling frequency (f_s) and NMR larmor frequency (f_l) needs to be carefully computed. For some data, it's hard to find an exact frequency difference between f_s and f_l. For this reason, it's sometimes hard to see NMR signal echo sum or T_2 decay signal due to this problem in down-conversion. To further simplify NMR signal processing, it is useful to digitize signal data beyond its Nyquist frequency. For water as a sample, minimum sampling frequency is 8.6 Msps. So an ADC with sampling frequency larger than 8.6 Msps needs to be used here. It turns out that a serial-based ADC cannot handle such speed, therefore a parallel-bus ADC is used.

For this second spectrometer design, Linear Technology LTC1746 ADC is used. LTC1746 is a 14-bit parallel high-speed ADC with 25 Msps sampling rate. It has some advantages over LTC2314 used in the first design, i.e. it can sample in higher frequency than LTC2314, it can be driven with lower clock frequency due to its parallel topology, and it

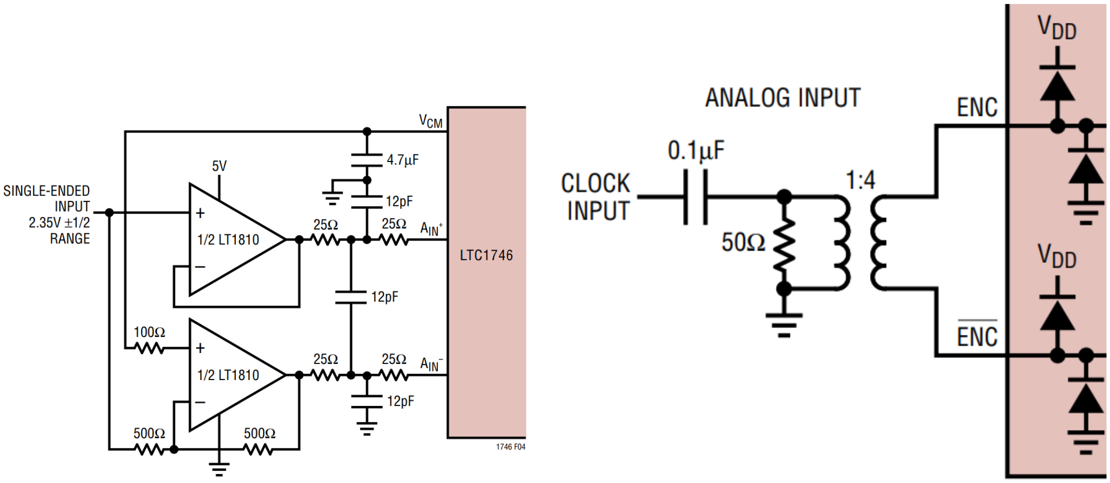

Figure 4.6. LTC1746 differential input driver (a) and differential clock driver (b).

is easier to drive with FPGA because signal acquisition is straightforward with no serial protocol. However, there are some disadvantages as well, i.e. it needs a lot of passive components to support its operation. Differential signal is needed by the ADC to drive its analog input and clock input. It needs LT1810 Op-Amp to convert a single-ended input signal to a differential signal to drive the ADC input, shown in Figure 4.6 (a). Input clock of this ADC is also differential. Therefore, if we utilize a single-ended clock to drive the ADC, a transformer needs to be used to make a differential clock signal, shown in Figure 4.6 (b).

4.5 High Power Transmitter Path

In addition to a same low-power transmitter path implemented using LM7372, in this new spectrometer design a high power transmitter path is implemented using LT1210 Op-Amp instead of LM7372 Op-Amp as the last stage transmitter amplifier. It can source up to 1.1A current to the load. A custom transformer is needed to be implemented as well to support this large current. It also has a shutdown pin, so the Op-Amp can be powered off whenever it's not being used.

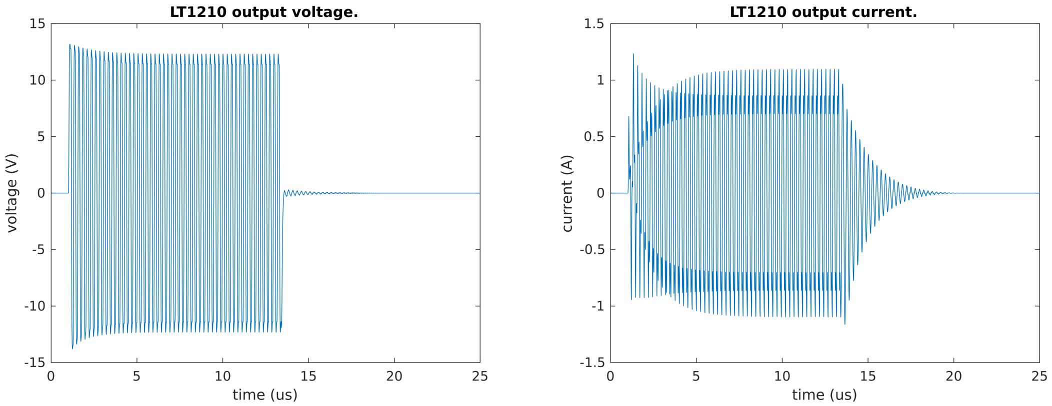

Figure 4.7. LT1210 Op-Amp output voltage (a) and output current (b).

LTSpice is used to simulate the high power transmitter path designed. The result is shown in Figure 4.7. It can be seen at steady-state, voltage output is about 12.2V peak-to-peak at current output 1.1A peak-to-peak. The output impedance of the Op-Amp is therefore 2.54 Ω. To transform its output impedance output from 2.54 Ω to 50 Ω, approximately, a transformer with N ratio of 1:4.4 is used, or with inductance ratio of 1:19.7, to maximize its power transfer. An output resistance of 7.3 Ω is also added to limit its current output to 1.1A, the maximum output current of LT1210, in order to prolong the life of the Op-Amp during operation.

Another LTSpice simulation is done to see the output voltage and current at the coil itself. Figure 4.8 shows that the coil voltage is approximately 164V peak-to-peak at 13.8A peak-to-peak coil current. This is resulting a huge power handling advantage over the first design in which the coil voltage is about 60V peak-to-peak at 5A peak-to-peak coil

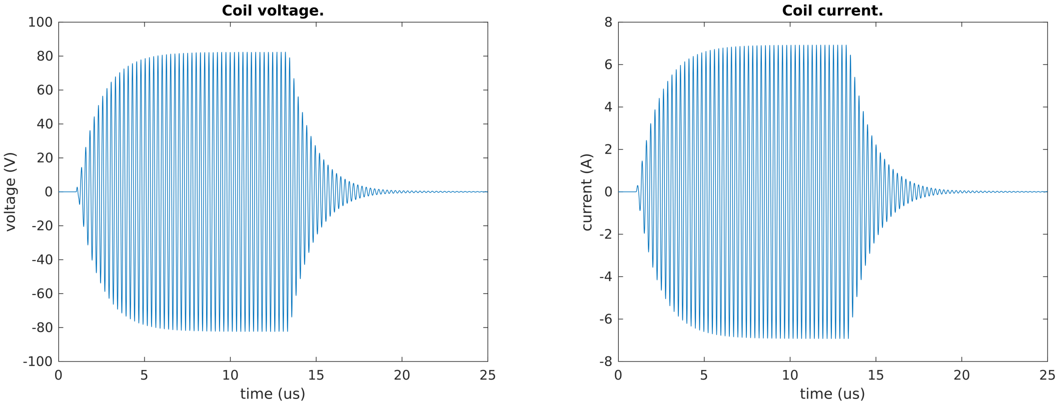

Figure 4.8. Coil voltage (a) and current (b) during transmit.

current. The increase in available power is defined with this equation

$$\frac{P_2}{P_1} = \frac{I_2^2\,R}{I_1^2\,R} = \frac{I_2^2}{I_1^2} = \frac{13.8^2}{5^2} \approx 7.62 \tag{4.4}$$

where P_2 and P_1 are the power available for the first and second power amplifiers respectively, I_1 and I_2 are the peak-to-peak current for the first and second power amplifiers respectively, and R is the load, which is the same for both power amplifiers. The increase in available power is approximately 7.62 in this case.

Until the writing of this thesis paper, the high power transmitter path hasn't been used to get an NMR signal. The PCB board has two transformer spaces, which can be used once this path is needed in the future. First one is for Coilcraft transformer that is already used in the previous spectrometer design, and second one is for a custom transformer that needs to be manually wired.

4.6 Other Minor Changes

With this design, a slight change is done for the receiver second stage amplifier. Four jumpers at the last stage of Op-Amp that select the gain of the Op-Amp, shown in Figure 2.14, are now implemented using digital switches. This switch can be controlled using the NMR controller. Total gain of the amplifier is same as before, i.e. 40dB to 66dB. An algorithm needs to be written for the controller to sense whether the last stage Op-Amp is saturated, and to decrease the gain so saturation does not occur. Switches are implemented with DG612A chip, a quad electronic SPST switches. TCA9555PWR, an I2C I/O expander that has been discussed before, is used to control state of the switches.

High voltage power supplies, i.e. +15V and -15V power supplies, shutdown function is implemented and can be controlled by NMR controller. This is necessary to reduce power usage when system is not active. Analog ground and digital ground are now isolated via ferrite bead, to minimize digital noise feed-through to analog signal. Besides LTC1746 ADC, LTC2314 ADC is also there as a backup if LTC1746 can't work properly, or as an additional ADC if another path of digitization is needed.

4.7 FPGA Design Changes

Some FPGA design improvements are made for this new spectrometer. The previous design is made with low configurability aspects in mind. The goal of first spectrometer is to capture an NMR signal from fixed sample, i.e. water. In that case, target Larmor frequency is around 4.3 MHz, and doesn't change a lot over measurements, given that external factors aren't changed, e.g. room temperature, magnet geometry, etc. Even if there are external factors, usually Larmor frequency change doesn't go over the bandwidth of its initial condition. Given a fixed Larmor frequency, NMR controller is designed and implemented using Larmor frequency as its frequency reference. All timing parameters, e.g. the 90 degrees pulse and the 180 degrees pulse, are also referenced to the Larmor frequency. Clock gate for the transmitter is also registered using the same frequency. This is advantageous in term of post-ADC signal processing, as phase of the signal is always same for every echo in any scan. Echo-sum can be computed directly from raw data, without down-conversion or scan rotation operation first. In this case, seeing whether an NMR signal presents or not is a relatively trivial case. One can plot its the NMR echo-sum and tell whether an NMR signal is present or not. One example is shown in Figure 3.10.

However, for the new spectrometer design, previous approach is not particularly a good idea. A spectrometer needs to adapt itself to the change of sample, i.e. changing Larmor frequency for different samples. Changing the FPGA frequency of operation is not hard, as we can use PLL to generate any frequency. But if we use the same design as previous, when the NMR controller changes its input clock, e.g. due to change of sample, there's a risk of setup and hold violations inside the FPGA as all modules inside the FPGA use the same input clock for their operation. This is due to the fact that hardware in an FPGA itself is usually compiled and routed with fixed frequency as a constraint. Therefore, data integrity problem will arise if input clock of the FPGA is changed without recompilation. To avoid this issue, it is usually a good practice to make transmit signal independent from the NMR controller operational clock. In this case, the NMR controller clock is chosen to be 50 MHz. Therefore, all NMR controller parameters, such as length of 90 degrees pulse and 180 degrees pulse, ADC delay length, echo spacing length, etc, are dependent only on the NMR controller clock. Now that system clock is separated from transmitter clock, we can simply change its transmit pulse frequency and at the NMR controller side, nothing should be changed. It also simplifies all parameters computations with the NMR C program under the ARM processor, as now the system clock frequency is always fixed.

On the other hand, independence of controller clock from transmit clock introduces a new problem inside the NMR system. It gives different starting phase for every echo in one scan, and also different starting phase for every scan in different measurement. For first case, if transmit pulse that is long enough, different starting phase in every echo in a scan doesn't introduce any issues. Please note it does not remain true for short transmit pulse, where only few number of pulses generated as transmit signal. The effect of this problem is not in the scope of this thesis and is not discussed here. Interested reader should read a work by Mandal[18] to learn it. For the second case, however, echoes can not be summed directly from the raw data without initial processing. Phase-cycle ring-down can not be canceled directly by subtracting data like it was done in previous spectrometer design. These problems will be discussed in the next section.

4.8 Changes to NMR Signal Processing in MATLAB

With the new hardware design, some additions need to be added to the previous signal processing. The complete signal processing chain is shown in Figure 4.9. For this design, as the ADC operates over the Nyquist sampling rate of the signal, we can directly use the Larmor frequency to downconvert the digitized data, therefore simplifies the signal processing. To fix the problem arises due to different start phase of the transmit signal, scan rotation is added for every scan. The compute the rotation angle, echoes in each scan is summed first. Then, the angle of the sum is computed and used as the rotation angle of the scan. After that, the same signal processing is performed to the signal.

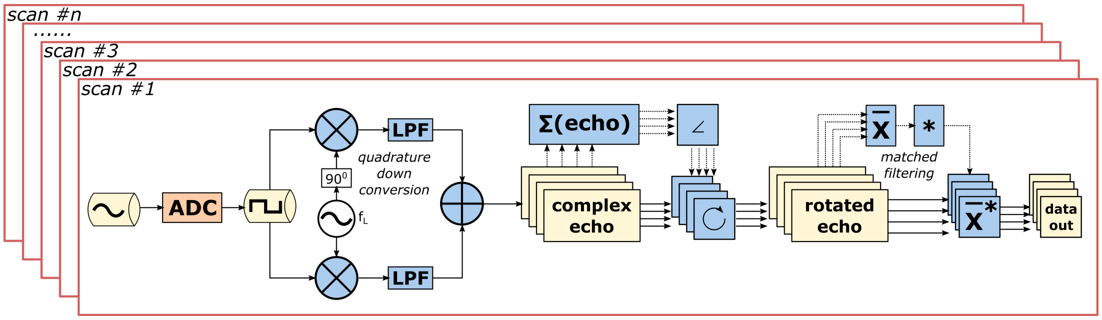

Figure 4.9. MATLAB NMR signal processing.

5 Spectrometer Performance Characteristics and Measurements

In this chapter, characteristics and performance of the second spectrometer are discussed. The discussion includes the tunable matching network performance and its reflection measurement, preamplifier performance, supporting functions in NMR *C* program, NMR related functions in NMR *C* program, frequency sweep and amplitude sweep functions, and comparison with commercial spectrometer.

5.1 Tunable Matching Network Performance and Its Reflection Measurement

An NMR design often needs capability to change samples, permanent magnet, environment condition that usually change the Larmor frequency used. It was not possible with the first matching network design. The main advantage of this matching network design is the ability to change its capacitances electronically, such as C_1 and C_2, shown in Figure 4.1, to ensure maximum power transfer to the coil over a range of frequencies. The main goal of this matching network is to have a minimum of -10dB reflection coefficient for a broad range of frequencies. Matching network S11 parameter is measured using a network analyzer. The first approach used to find an optimum capacitances for a given frequency is to sweep all C_1 and C_2 values, measure its S11 parameter, and store the data. This is found to be inefficient in terms of time, as for 8 switches selections for both C_1 and C_2, total combination is 256*256 = 65,536. One measurement with network analyzer takes a minimum of 4 seconds, therefore total time needed is around 72 hours. Second, there are 2 operating points for an LC circuit where the impedance is matched to 50 Ω, i.e. before resonance frequency and after resonance frequency. We are not interested in the impedance matching after resonance, as capacitance of the coil sensor is more dominant than its inductance at the point. Using this random search approach, we may find a good impedance match, but the data doesn't tell if its matching condition happens after or before resonance.

A more intelligent approach is to use predicted C_1 and C_2 values for initial S11 measurement, which uses Equation 2.16. Then C_1 and C_2 value is increased and decreased one by one until an optimum S11 is found. With this approach, minimum S11 parameter with frequency is measured and is shown in Figure 5.1. From the figure, it can be seen that a minimum of -10 dB S11 parameter can be achieved from 2.5 MHz to 4.8 MHz. Minimum S11 parameter for every frequency is different due to finite step of C_1 and C_2 capacitances, which either introduces a slight offset from 50 Ω resistance or introduces imperfect reactance cancellation at the desired frequency of operation, already predicted by Equation 2.16. This effect is worse at certain frequencies, shown with a gap around 4.7 MHz to 5 MHz, where reflection is more than -10 dB. This problem can be solved by adding smaller capacitance step, either by changing whole capacitance to lower values or by adding more switches for smaller capacitance. It effectively makes tunable matching network useful for a broader range of frequencies. However, it is not needed in this project as the preamplifier itself has narrower range frequency of operation.

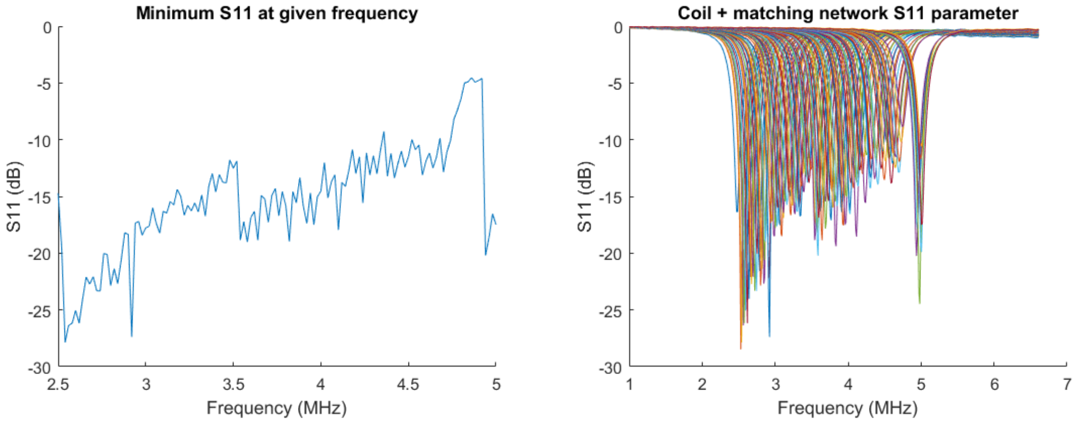

Figure 5.1. Tunable matching network S11 parameter.

For every frequency tested, C_1 and C_2 values correspond to minimum reflection are stored in a table. This table is used initially to select an optimum C_1 and C_2 for a chosen frequency of operation. Although this approach is good to compute an accurate S11 measurement during development phase due to the use of a network analyzer, it is also not really portable. In effort to make this spectrometer portable, a directional coupler is implemented. As already discussed, directional coupler is used to couple reflected signal from matching network during transmit phase, as shown in Figure 4.2. The output of the reflection path is a digitized data that is -20 dB of reflected signal from coil sensor. Figure 5.2 shows one example of measured reflection from the coil. The y-axis is digitized voltage amplitude in decibel format, directly plotted from the ADC output, and are not normalized. It can be seen from the figure, at the frequency of interest, the minimum reflection is about -20 dB compare to the maximum amplitude. While this value is not calibrated with a more sophisticated instrument, it gives an idea of frequency at which the minimum reflection happens.

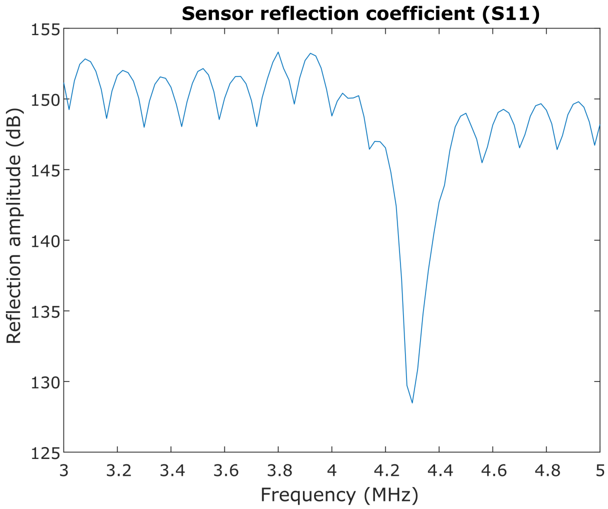

Figure 5.2. Sensor S11 parameter using coupler.

5.2 Preamplifier Performance

According to simulation, the broadband preamplifier shows very broad frequency range for S11 parameter, shown in Figure 4.4 (b). The simulation shows that a fix gate bias voltage can be used to achieve a broad S11 parameter that is less than -10 dB. A real measurement for the preamplifier done by sweeping transistor gate bias voltage from -5 V to 0 V, and the result is shown in Figure 5.3 (a). A line in the figure corresponds to a value of gate bias voltage. The line with minimum S11 parameter at higher frequency is for -5 V gate bias voltage and the line with minimum S11 parameter at lower frequency is for 0 V gate bias voltage. Increasing gate bias voltage further doesn't seem to push the line further to lower frequency, as the g_m of the transistor is already at its maximum. Decreasing gate bias voltage further was not tested as -5V is minimum voltage the DAC can supply. But it doesn't necessarily mean its working frequency can not be increased further. In this project, maximum frequency of 5 MHz is enough so further testing is not performed.

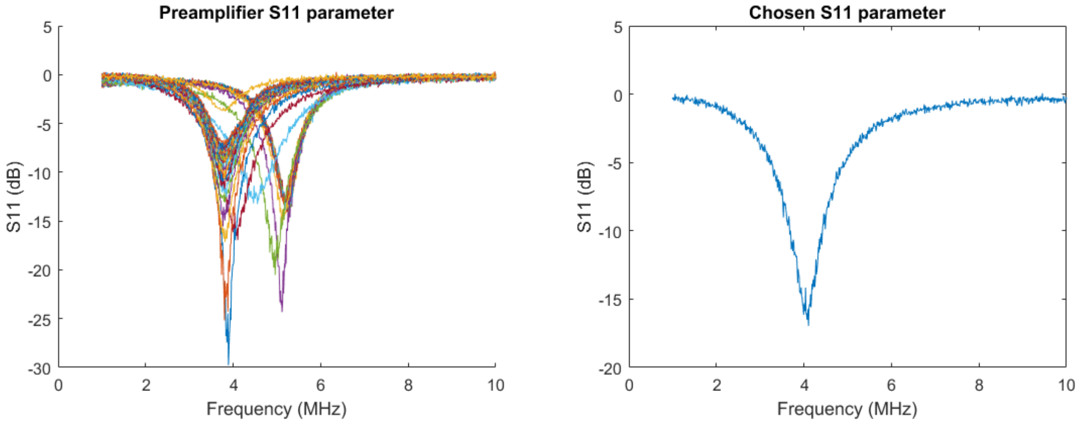

Figure 5.3. Preamplifier S11 parameter with gate voltage change.

It can be concluded from measurement shown in Figure 5.3 (a), input bandwidth of preamplifier still depends on the gate bias voltage, which is quite far from what simulation predicts. There are a couple of possibilities for mismatch between simulation and experiment:

- The simulation ignored parasitic components (e.g., winding capacitance) within the transformer.
- The JFET models used for the simulations are simple level-1 FET models, and are probably not very accurate over a wide range of V_{GS} values.

However, the -10 dB bandwidth is larger than what we have from previous preamplifier. Previous preamplifier design has -10 dB bandwidth around 360 KHz, shown in Figure 3.1. Here, with a single gate bias voltage is chosen, its S11 parameter is shown in Figure 5.3 (b). From the figure, minimum S11 parameter is seen to be around -17 dB at 4 MHz, with its -10 dB bandwidth of 740 KHz around 3.68 MHz and 4.42 MHz. From now on, this S11 parameter is used for all tests and measurements as its bandwidth is enough to cover a useful frequency range.

Automatic process to find minimum preamplifier S11 parameter by sweeping gate bias voltage is not possible with current design. To realize this function, we need to generate low voltage in a particular frequency at input port of the preamplifier, digitize the signal at the output of preamplifier, and analyze it with some signal processing techniques. As it is not possible at this point, the only automation we can implement with current hardware is to generate a look-up table storing optimum gate bias voltage versus frequency, where NMR C program can read and give a gate bias voltage corresponds to the value suggested by the table. The look-up table itself is pre-generated using MATLAB, utilizing a network analyzer in the process. This look-up table is good if environment doesn't change, but is not valid through a change in environment parameter, e.g. temperature and pressure.

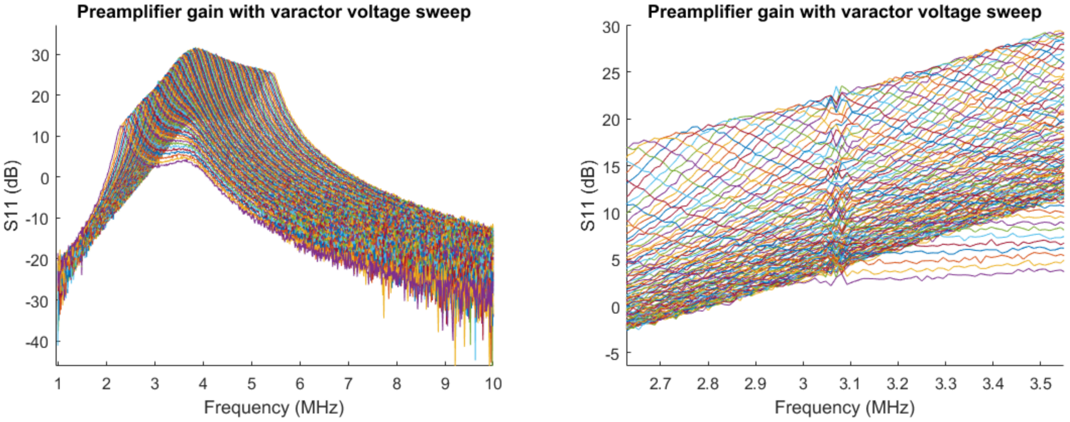

Figure 5.4. Preamplifier S21 parameter with gate voltage change (a) and its zoomed-in figure around 2.7 MHz to 3.5 MHz (b).

With one gate bias voltage chosen and shown in Figure 5.3 (b), gain of the preamplifier is plotted. This time, varactor bias voltage is swept from -5 V to 5 V, effectively changes output impedance of the transistor. Figure 5.4 (a) shows the result. In the figure, we can see max gain of 31 dB at 4 MHz. However, the maximum gain changes over frequency due to different S11 parameter at the input side with different frequency of operation. From 3.5 MHz to 5 MHz, the preamplifier gain is more than 27 dB, which is considered enough for the spectrometer purpose. Figure 5.4 (a) shows an external noise at 3.08 MHz during measurement, which disturbs the signal received by network analyzer and is shown as a fluctuation in preamplifier gain.

5.3 The NMR C Program

Here we will discuss the NMR C program developed for the new spectrometer. Figure 5.5 shows all the functions are related to each other.

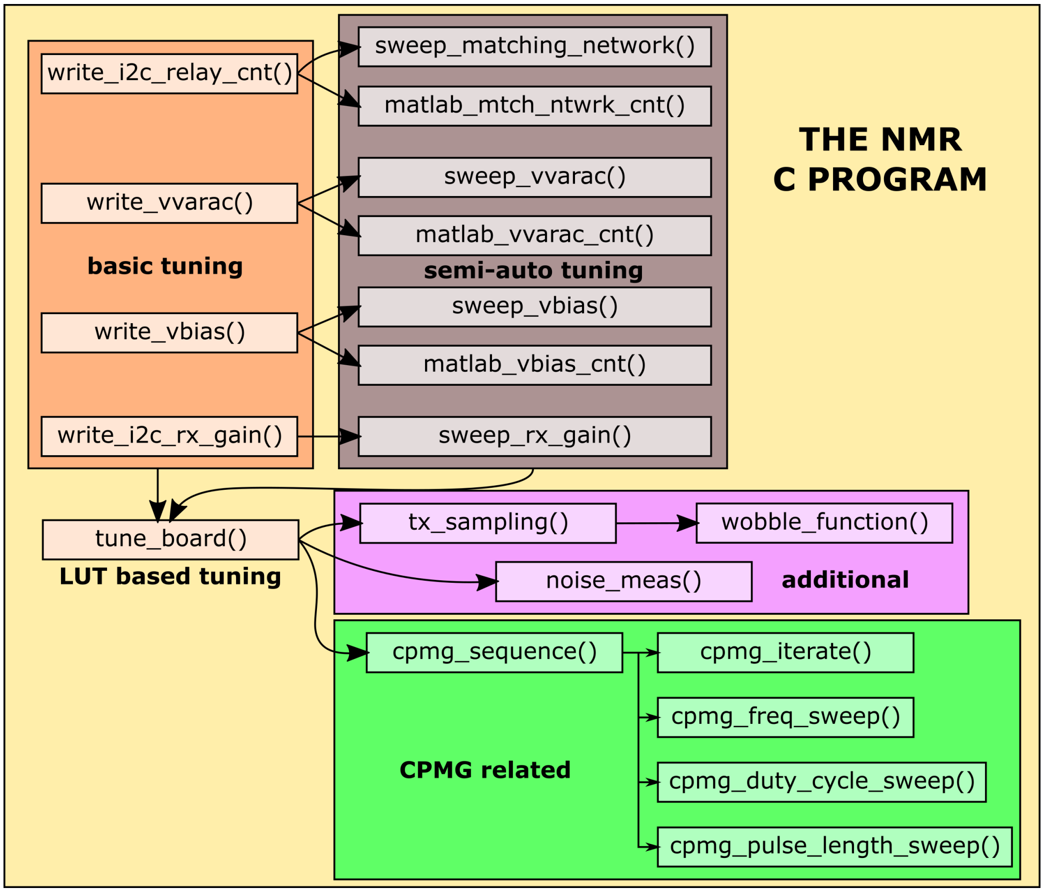

Figure 5.5. The NMR *C* program.

5.3.1 Supporting Functions

The spectrometer setup is actually more or less the same as what is already done with the previous spectrometer. Some differences need to be implemented in the NMR *C* program to configure additional parameters introduced with this more advanced spectrometer. The first software is I2C controller to configure matching network relays, called *write_i2c_relay_cnt()*, which basically just connects or disconnects C_1 and C_2 parallel capacitors using relay. It has 2 child functions, *sweep_matching_network()* and *matlab_mtch_ntwrk_cnt()*. *Sweep_matching_network()* is very useful to find a right capacitance value by sweeping the matching network C_1 and C_2 while the result, i.e. reflection parameter, can be seen with network analyzer in real time. *Matlab_matching_network_cnt()* is a function which uses serial as an input to make changes in capacitance value. In this case, it is used in conjunction with MATLAB as the serial controller. Characterization of tunable matching network performance in this chapter was done using MATLAB as the serial controller and a network analyzer.

A simple function *write_i2c_rx_gain()* is written to control the final gain stage of the receiver path, by enabling or disabling resistances at Op-Amp feedback, effectively changing its gain. This is controlled by using the same I2C controller already implemented with the matching network. A simple *sweep_rx_gain()* is also written to configure sweep the gain value, which is useful during development phase.

DAC used in this spectrometer is controlled using SPI controller in the NMR FPGA controller. The SPI controller itself is controlled by NMR *C* program. Some useful functions are also developed for the DAC. The first basic function is to give warning about the DAC problem during operation, including thermal shutdown and overcurrent alert.

The overcurrent alert shows up every start up as the maximum load the DAC can handle is 2 kΩ and 4000 pF, while the DAC load in this case is a big capacitance 100 μF and inductance of 1 mH. In this case, the current is clamped at 20 mA, and a delay needs to be used so that the voltage at the capacitance ramps up to its desired voltage. *Write_vbias()* and *write_vvarac()* functions are written to control DAC output voltage for gate bias and varactor bias, respectively. *Sweep_vbias()* and *sweep_vvarac()* functions are written to sweep the transistor gate bias voltage and the varactor bias voltage, respectively, which is advantageous to characterize the reflection coefficient and gain of the preamplifier during the development phase. *Matlab_vbias_cnt* and *matlab_vvarac_cnt* are also written to enable configuration using MATLAB, in the same manner as *matlab_mtch_ntwrk_cnt()* function. The preamplifier performance in previous sections are characterized using MATLAB as the controller for the C program on the FPGA side and an external network analyzer.

The software developed with this spectrometer is also more detailed compared to the previous one. One nice feature is to create a folder with its measurement date and time as name to automatically separate between measurements. All measurement parameters are stored in the same folder with measurement data. Therefore if in the future one needs to check what measurement was done, every detail is stored in this configuration file. The function also generates a corresponding MATLAB handler to process the data.

5.3.2 NMR Related Functions

Some NMR related functions are also developed with this spectrometer. The first one is *CPMG_sequence()*, which was also implemented for the previous spectrometer. Some changes need to be made to support its new features. As now the transmit signal is independent of the controller system clock, the computation for all CPMG parameters and timing is made according to the changes, which is computed with the C program. There's an option for the CPMG parameters to be shown during runtime, which is really useful for debugging purpose. This CPMG_sequence() is used in almost every NMR related functions below.

The second function is the *tx_sampling()*. This function activates the transmitter and samples ADC data at the same time. This is useful for two purposes. First is to simulate signal coming from the coil and to sample its data. This is handy for testing whether the chain of the receiver functions well, testing the gain of the receiver path with different frequencies (this one is not implemented yet). Second is to measure reflection coefficient from the coil using the directional coupler. This function also controls the corresponding receive path, i.e. from normal receive path or coupler reflection path, depending on the measurement being performed. One child function written in conjunction with *tx_sampling()* is the *wobble_function()*. The *wobble_function()* sweeps transmit frequency and digitizes data directly from the directional coupler with little gain. This function has some parameters needed as input, i.e. start frequency, stop frequency, spacing frequency, and the number of samples taken for every frequency. It also generates MATLAB handler to process the data. Figure 5.2 is one of the results of this function.

The third function is *CPMG_freq_sweep()*. The idea of the function is to measure and plot the NMR signal amplitude at a range of frequency. This requires the function to configure all frequency sensitive paths in the spectrometer when sweeping the frequency, including the preamplifier output varactor bias voltage, preamplifier input gate bias voltage, and the matching network. For every single frequency, one or more CPMG sequence is performed.

The last function is *CPMG_iterate()*. This function does *CPMG_sequence()* several times with or without phase cycle. Parameters needed for this function is almost identical to *CPMG_sequence()*, with some additional parameters, i.e. number of iteration and phase cycle. It also generates MATLAB handler to process the data, to compute the average echo sum, filtered T_2 decay, SNR per echo per scan, T_2 time, and the NMR signal initial amplitude.

5.3.3 Frequency Sweep and Amplitude Sweep

NMR signal is very sensitive to frequency, as it is a resonant phenomenon with very limited bandwidth. An exact transmit frequency needs to be given in order to excite an NMR signal. Therefore, NMR spectrometer needs to have a frequency sweep function in order to find operating frequency in which the sample sits on. There are several challenges to implement this frequency sweep function.

The first one is to find the optimum capacitance values for the matching network at a particular frequency. Although a directional coupler is already implemented and its functionality is already tested, the main software processor to analyze the data is MATLAB in a desktop computer. To fully automate the system, a program needs to be written for Linux inside the DE1-SoC board, either with C or with Python, which was not possible during the writing of this thesis. Therefore, a table of frequency and integer value corresponds to capacitance value for the matching network is precomputed as parameter for the frequency sweep function.

The second one is to find the optimum voltage for the transistor gate bias and varactor bias. The transistor gate has quite large bandwidth as its input impedance is 50

Ω over 600 KHz bandwidth, so it doesn't need to be configured every time frequency is changed. But the transistor gain has a very limited bandwidth, due to the high-Q characteristic of inductance and varactor capacitance at the transistor output. The varactor capacitance depends on the voltage bias applied to it. For the same reason it was discussed with the matching network, a table of frequency and varactor voltage bias is precomputed as parameter for the frequency sweep function.

Tune_board() function is written to utilize the look-up table generated above. It has a frequency as an input parameter. This function changes matching network capacitances, transistor gate bias voltage, and varactor bias voltage to ensure every path of the hardware is operating at the desired frequency. This function is then combined with the frequency sweep function to obtain data. One measurement data is shown in Figure 5.6. The sample used for the experiment is water, and its Larmor frequency is known to be around 4.3 MHz. Looking at the figure, it can be clearly seen the signal coming from the water at 4.3 MHz. Its bandwidth is really narrow. At the time of the experiment, there's a noise at 3.08 MHz, which is picked up and shown clearly at Figure 5.6(a). However, the initial signal amplitude is small compared to the water.

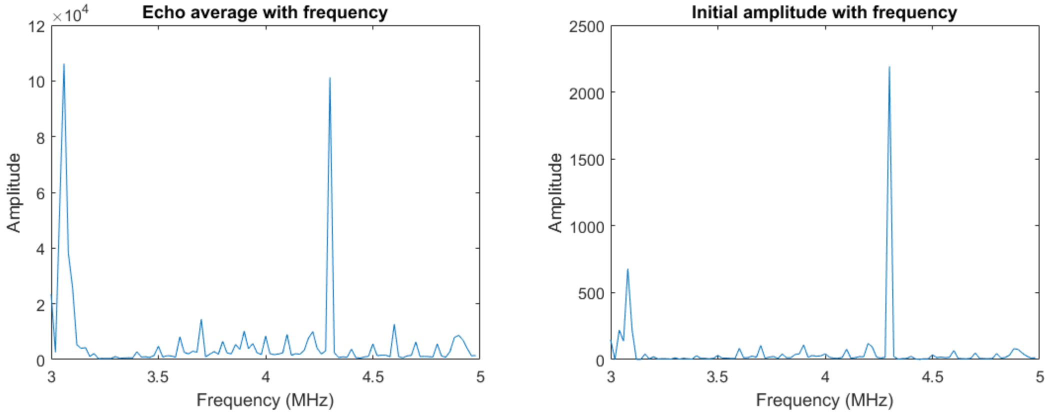

Figure 5.6. Frequency sweep echo average (a) and initial amplitude (b).

Amplitude sweep is another necessary function needs to be implemented in an NMR spectrometer. It is needed based on the fact that maximum amount of signal can be captured if the proton is nutated exactly $\pi/2$ from its initial condition, and then π for echo re-phasing (CPMG sequence). The amount of nutation given in the sample depends on the magnetic field B_1 generated by the coil. In order to do the amplitude sweep, there are two control parameters available in the system. The first one is to sweep the duty cycle of the pulse signal, which can be controlled by the PLL in the FPGA, keeping the same pulse length. The minimum duty cycle step written for this project is 1%. To achieve as low as 1% duty cycle, the FPGA PLL internal frequency needs to be at least a hundred times its output frequency. Therefore, for 4.3 MHz output signal, its internal frequency is at least 430 MHz. There is a limitation in which this internal PLL frequency can go up to, so minimum duty cycle is fixed to 1% given that the output signal is around 3 MHz to 6 MHz.

Figure 5.7 shows the relationship between duty-cycle and measured NMR signal amplitude. The measurement settings are 20 μs pulse length, 180 μs echo-spacing, 1024 samples per echo, 128 echoes per scan, averaging factor of 4 with phase cycling. The duty-cycle sweep is 2% to 50% with 1% spacing. It is shown in the figure that the signal amplitude and echo average start small close to zero duty-cycle. Signal amplitude and echo average ramp up until around 10% duty-cycle, where they are maximum, before they go down and reach almost zero at 18% duty-cycle. Sinusoid shape around this region suggests that duty-cycle in relation with transmit voltage generated at the coil is linear. We can also see from the figure that signal amplitude and average ramp up after 20% duty cycle to 50%, but not as big as before. At this point, the relationship between

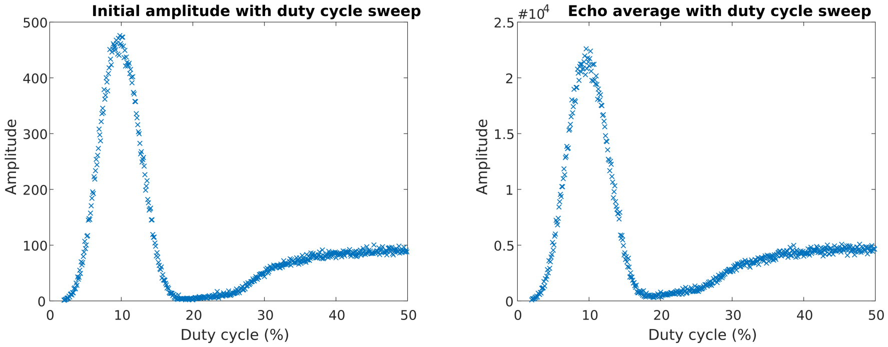

Figure 5.7. Duty cycle sweep NMR initial amplitude (a) and echo average (b).

duty-cycle and transmit voltage generated at the coil is not linear, as already shown in Figure 3.3.

The second degree of freedom is pulse length. The longer pulse length, the larger the nutation angle, as predicted by Equation 1.5. Figure 5.8 shows the relationship between pulse length and the output NMR signal amplitude. There is a peak signal amplitude around 7.5 μs transmit pulse length, shown either by initial signal amplitude or echo average. The reason behind this peak is exactly the same as what was discussed with the duty-cycle sweep above. However, here the figure shows a smooth sinusoidal peak, as the power absorbed by the coil is linear to the pulse length. Whereas at the duty-cycle sweep case, the power absorbed is not necessarily linear to the duty-cycle. There is another peak at 23 μs, which is significantly lower than the first peak. The longer pulse we generate in the transmitter, the lower the signal bandwidth. In this case, it suggests that we have inhomogeneous B_0 magnetic field that sample Larmor frequency is spread to a certain bandwidth from its 4.3 MHz center frequency. Shorter transmit signal has larger bandwidth, therefore excites more population of the sample. The first peak corresponds to $\pi/2$ - π nutation, while the second corresponds to $3\pi/2$ - 3π nutation. That is, both nutation angles are 3x larger for the second peak. However, the second peak

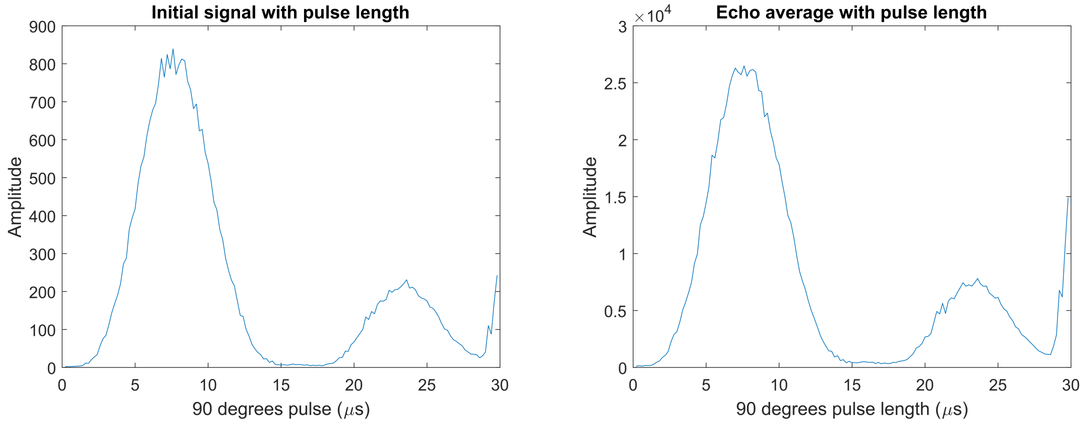

Figure 5.8. Pulse length sweep echo average (a) and initial amplitude (b).

corresponds to less population excitation due to longer pulse and less bandwidth, and therefore generates less signal. Increasing signal reception around 30 μs happens due to ring-down pickup, as the refocusing (π) pulse also gets longer and longer and eventually captured inside ADC acquisition window.

5.4 The NMR Experiment

With the FPGA NMR controller, NMR *C* program, and MATLAB post-processing chains have already been discussed and developed, we are ready to test the spectrometer with real sample. In this case, sample used is water doped with copper(II) sulfate, to decrease its T_1 time from approximately 2 s to 100 ms, therefore reducing a single experiment duration by 20 times. This is due to the strong magnetic interaction between a paramagnetic spin, i.e. the copper(II) sulfate, and nearby nucleus in the water, as suggested by Foley et.al.[19]. In this case, the T_1 of the paramagnetic ion dominates the T_1 of the water. General settings used are 4.3 MHz transmit frequency, 7.5μs $\pi/2$-pulse length, 35% duty-cycle, and 400ms scan spacing. For all figures in the NMR experiment, figure (a) shows echo average, figure (b) shows NMR signal T_2 exponential decay, figure (c) shows individual echo shape, and figure (d) shows scan phase rotation. Also for all figures in the NMR experiment, t_e is defined as the echo spacing, SpE is defined as the number of samples per echo, NoE is defined as number of echoes within one scan, and $t_{adc,init}$ is defined as the delay after π-pulse before the ADC starts sampling the signal.

First experiment is shown in Figure 5.9. For the particular measurement, T_2 relaxation time of the sample is found to be 8.27ms and its SNR per echo per scan of 7.15. In this case, the SNR increased due to shielded sample with a metal box, decreasing external pickup noise level. Due to its good SNR, we can see fairly consistent individual echo shape in Figure 5.9 (c). The scan phase rotation is not constant due to the unsynchronized transmitter output from the PLL with the system clock.

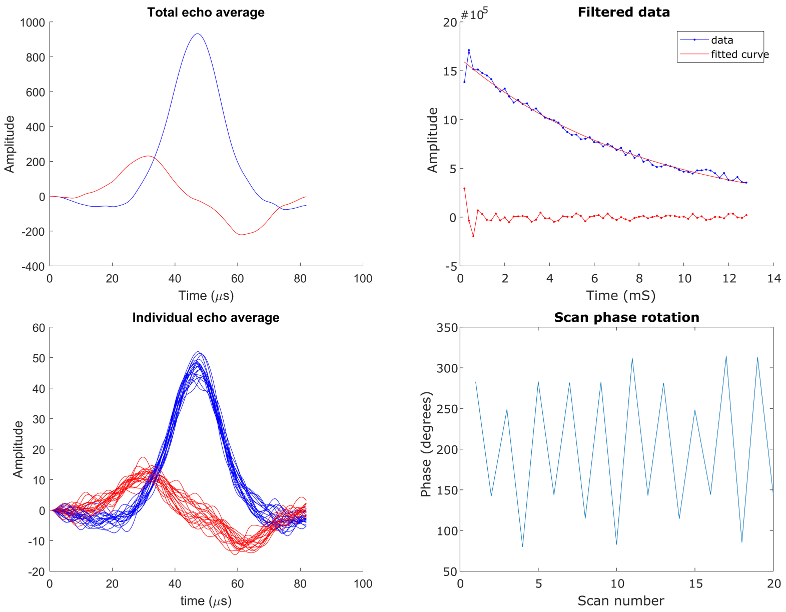

Figure 5.9. NMR data with t_e 200μs, SpE 2048 (acq. window 81.92μs), NoE 64 (experiment time 12.8ms), $t_{adc,init}$ 53.84μs.

Second experiment is shown in Figure 5.10. It is done by changing the samples per echo to 1024. The acquisition window is now half of before, therefore we can decrease echo spacing to 120 μs, and still minimize ring-down pickup as we still have 33.4 μs delay after π-pulse before ADC starts. With less echo spacing, number of echoes can be increased to 128. The T_2 relaxation time for this experiment is 8.29ms and its SNR per

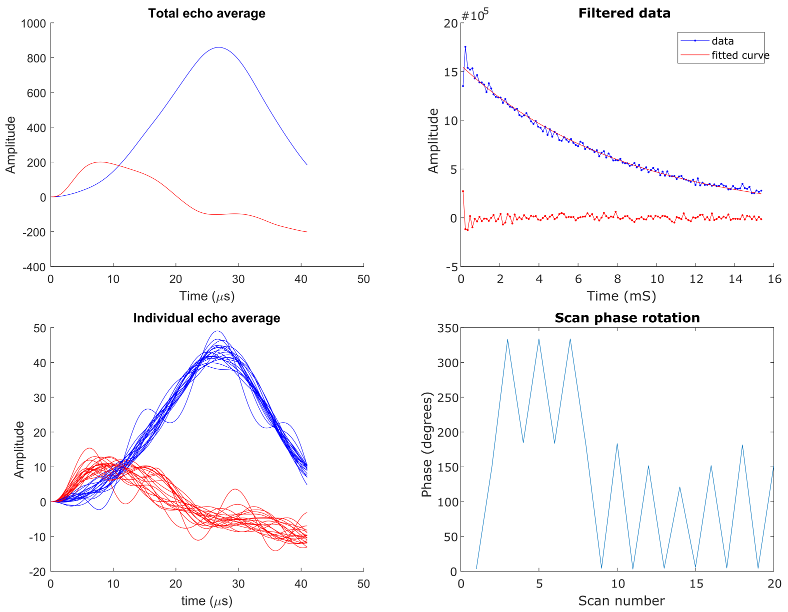

Figure 5.10. NMR data with t_e 120μs, SpE 1024 (acq. window 40.96μs), NoE 128 (experiment time 15.36ms), $t_{adc,init}$ 33.48μs.

echo per scan is increased to 9.21.

Third experiment is done by reducing echo spacing to 90 μs, shown in Figure 5.11. The T_2 relaxation time for this experiment is 8.34ms and its SNR per echo per scan is increased to 9.35. It is intuitive to obtain larger SNR as we capture data closer to its initial amplitude by decreasing echo spacing. One may realize by seeing the individual echo shape, that we pick more ring-down voltage as we don't wait long enough before sampling data. Its effect is however canceled by phase cycling.

Only by reducing the echo spacing from 90 μs to 87 μs, now we pick up more ring-down effect and the result is shown in Figure 5.12. The amplitude in Figure 5.12 (a) and (c) goes down a lot compared to previous one. It suggests that the ring-down voltage and

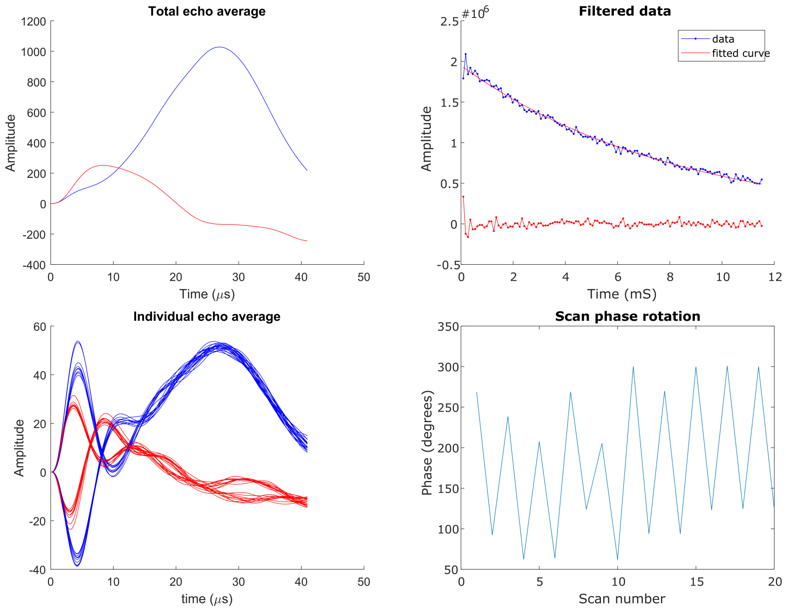

Figure 5.11. NMR data with t_e 90μs, SpE 1024 (acq. window 40.96μs), NoE 128 (experiment time 11.52ms), $t_{adc,init}$ 18.5μs.

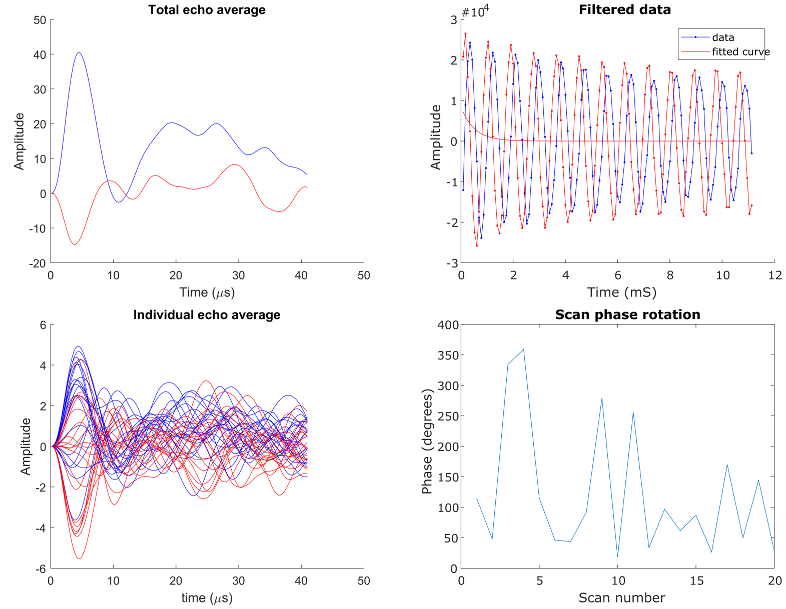

Figure 5.12. NMR data with t_e 87μs, SpE 1024 (acq. window 40.96μs), NoE 128 (experiment time 11.136ms), $t_{adc,init}$ 17μs.

the echo is almost equal in this case, therefore cancels each other. The echo is already hard to see in this case, and the T_2 and SNR reported are wrong.

If we decrease the echo spacing even further to 80 μs, ring-down become dominant, shown in Figure 5.13. Now, the ring-down effect dominates the measurement. We can see the effect of phase cycling shown in Figure 5.13 (c), where two groups of samples are shown with π rotation difference between them.

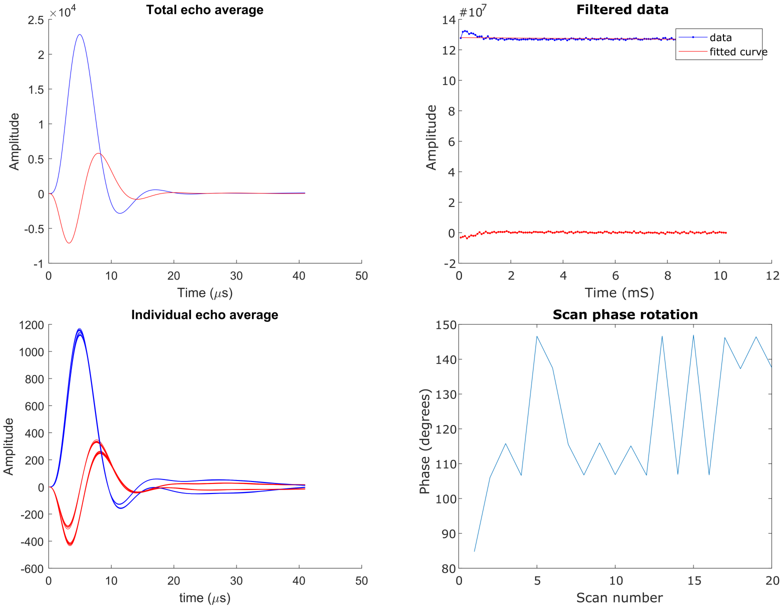

Figure 5.13. NMR data with t_e 80μs, SpE 1024 (acq. window 40.96μs), NoE 128 (experiment time 10.24ms), $t_{adc,init}$ 13.52μs.

From the experiments above, it can be concluded that the spectrometer can measure NMR spectrum when the configuration is right. Ring-down problem can be minimized with phase-cycling, but in this case, it is not applicable as the starting phase of the transmitter signal is not always the same between measurements. It is therefore really critical to fix the transmitter inconsistent starting phase design problem. We suggest to use Direct Digital Synthesizer (DDS), which are already used in many other NMR spectrometers, to achieve consistent output phase.

5.5 Comparison with Commercial Spectrometer

In this section, this project is compared to a commercial spectrometer, i.e. Magritek Kea2, which costs around \$20,000. For comparison, this project costs around \$1,500 to \$2,000. Here, both of the spectrometers are tested with similar settings, shown in Table 5.1.

Table 5.1. NMR spectrometer comparison test settings.

Parameters	this work	Magritek Kea2
points per echo	2048 at 25 Msps	128 at 100 Msps
echo window	$81.92\mu s$	$128\mu s$
echoes per scan	64	64
echo spacing	$180\mu s$	$180\mu s$
scan spacing	400ms	400ms
pulse length	6	6
pulse duty cycle	30%	N/A
pulse power	N/A	1.995W
total scan	50	50

The sample used is doped water, with T_2 time approximately 10ms. Figure 5.14 shows its echo decay. An exponential fit is used to approximate its T_2 time. The T_2 time is approximately 9.55ms with this work and 8.9ms shown with Magritek Kea2. Figure 5.15 shows the echo average shape. Signal-to-noise (SNR) ratio is 3.58 for this work and 4.87 for Magritek Kea2. For this comparison, these two plots were acquired with the magnet and probe in the open (without a metal shield), which resulted in increased noise and thus lower SNR compared to the previous section.

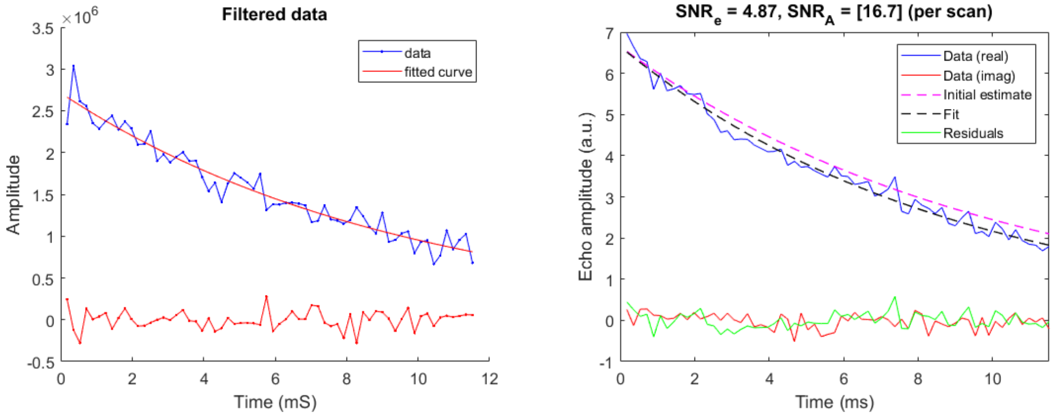

Figure 5.14. T2 echo decay for this work (a) and Magritek Kea2 (b).

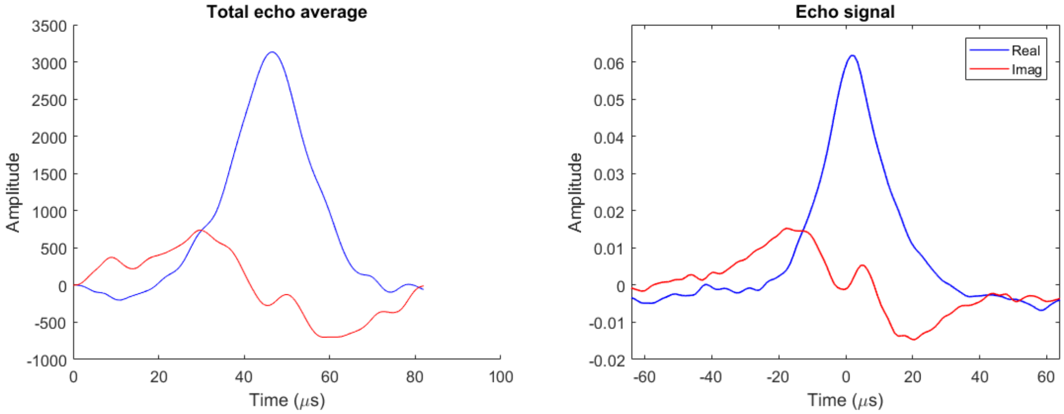

Figure 5.15. Echo average with this work (a) and Magritek Kea2 (b).

6 Conclusions

This thesis has demonstrated two working portable NMR spectrometers. Simulation, PCB design, hardware assembly, hardware characterization, controller development in Verilog, software development in C, and MATLAB post-acquisition signal processing are done to finish the project. Two NMR spectrometer designs have been simulated, tested, and used to measure real NMR signals from water. First design is a fixed frequency NMR spectrometer. It is a tuned hardware, works particularly for a given magnet geometry and temperature.. Its frequency of operation is 4.3 MHz. NMR signal from water is captured during experiment. NMR echo-sum shape and T_2 decay time of water is captured with the NMR measurement. It is also a proof of concept for a portable NMR spectrometer system using a custom board and an FPGA as a controller.

Second design is a tunable frequency NMR spectrometer. It is based on the first spectrometer design with some additional features. It works at frequency range of 3.5 MHz to 4.8 MHz, which is limited by the preamplifier at lower bound, and by the matching network at the upper bound. Some additional hardware is added to support its tunable frequency range of operation include tunable matching network using reed relays, tunable preamplifier input impedance using common-gate transistor topology, and tunable preamplifier output impedance using varactors. Some additional functions implemented are S11 reflection coefficient measurement using directional coupler, higher speed ADC to support above-Nyquist sampling-rate, and high power transmitter path. Basic NMR characterization capabilities are also implemented, including wobble function, frequency sweep, amplitude sweep using duty-cycle, amplitude sweep pulse length, noise measurement, and automatic tuning based on look-up tables. The spectrometer is compared to the Magritek Kea2, a commercial spectrometer, and achieves SNR of 3.58 comparable to SNR of 4.87. The T_2 relaxation time constant measured with both spectrometer is comparable as well, being 9.55ms and 8.9ms, respectively.

7 Suggested Future Research

Here are some suggested future improvement can be done with the spectrometer:

- Using DDS for the transmitter driver, instead of a direct PLL output, to ensure identical starting phase with different scan.
- Implementation of quadrature downconversion, low-pass filter, and data averaging inside the FPGA due to the limitation of the FIFO memory. The same method is used inside the Magritek Kea2 spectrometer. A higher speed ADC can be used as well with this implementation, without worrying about FIFO memory limitation. Another option is to use the 64 Mbytes SDRAM available in the FPGA fabric to store the data and use the FIFO only for the SDRAM input buffer.
- Implementation of a more precise C_2 with more relay stages. This is to ensure that the upper bound frequency range is not limited by the matching network, which is relatively easy to fix, but purely by the preamplifier. Once this is implemented, the operational frequency range of the system can be expanded from 3.5 MHz to 5 MHz, instead of 4.7 MHz.
- Signal processing migration from MATLAB to a more portable solution, like python, which can be implemented directly in DE1-SoC board. This will make the system truly portable. It is also useful to add screen to the system after this

point is achieved. In this thesis migration from Yocto Linux to a more established Ubuntu Linux was done to ensure compatibility and update in the future.

- T_1 measurement function. An inversion recovery pulse sequence will be written for this purpose.
- Implementation of a configurable pulse programming inside the FPGA NMR controller, similar to what the Kea spectrometer does.
- Program to do automated tuning for the preamplifier and the matching network, which is possible in hardware side. However, software has not been implemented yet.

Appendix A

PCB for testing the AFE

This appendix presents the photographs of the PCB board used in the project.

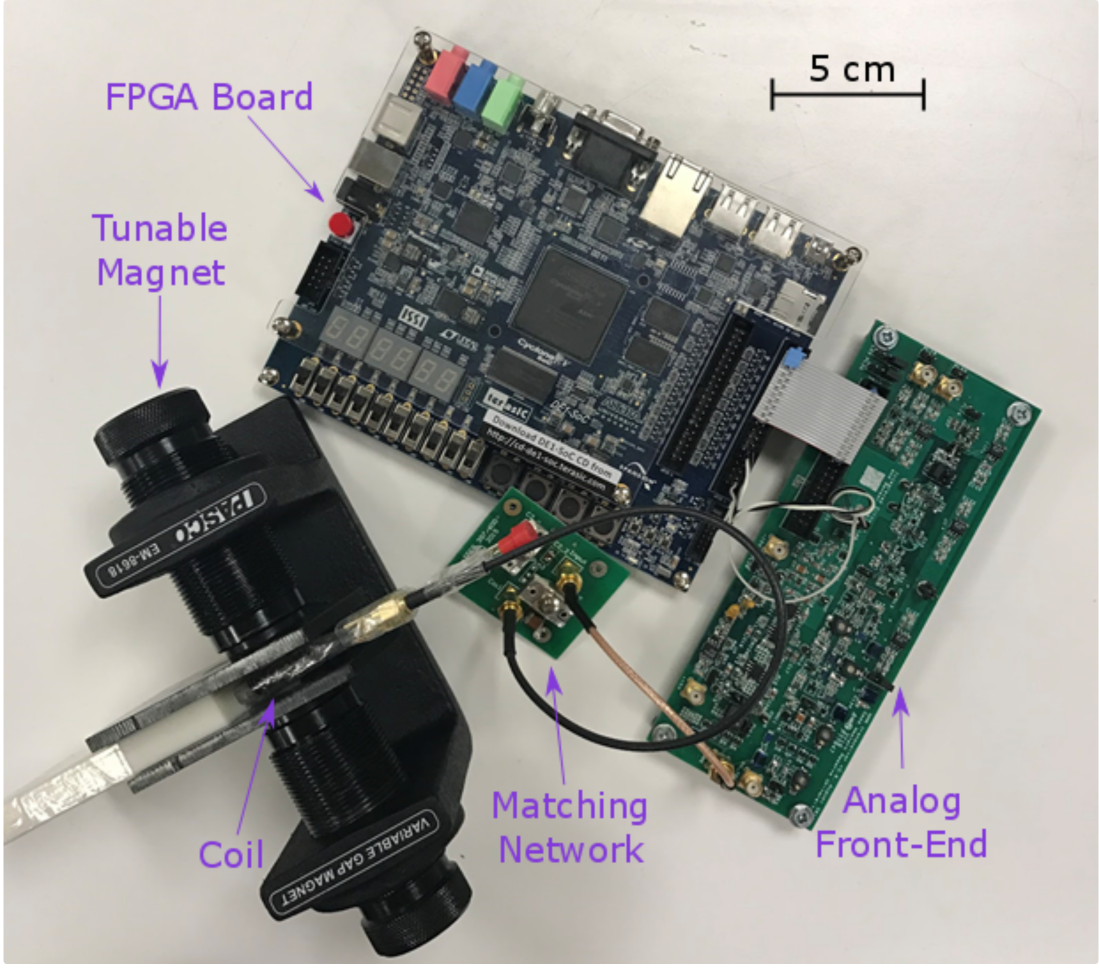

Figure A.1. Photograph of the first NMR spectrometer.

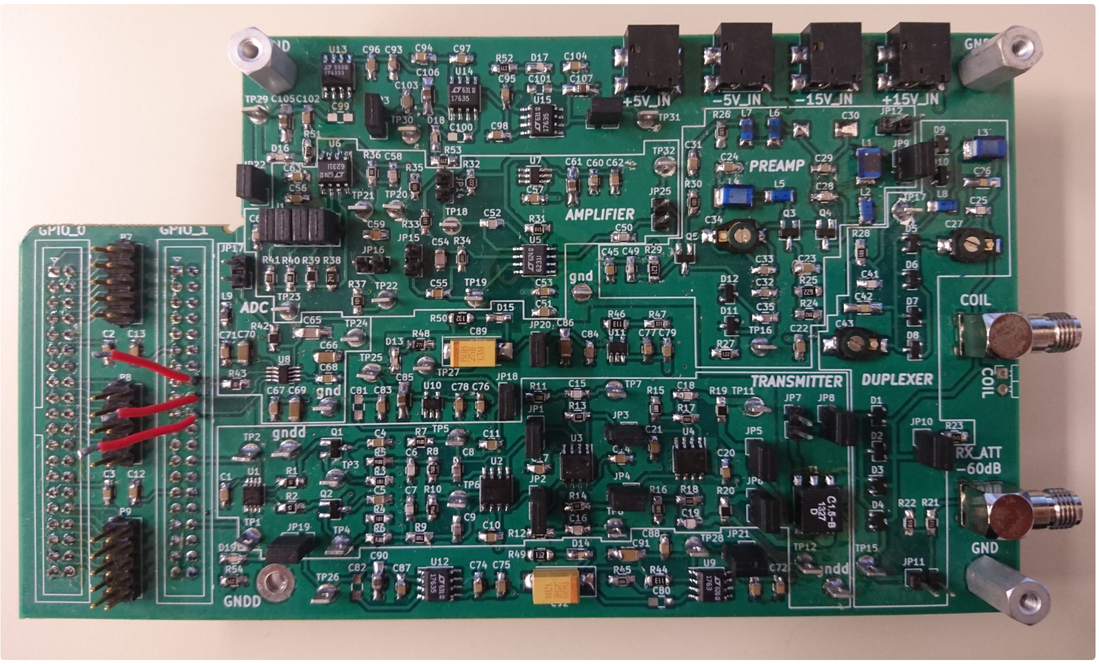

Figure A.2. Photograph of the second version of the first NMR spectrometer AFE.

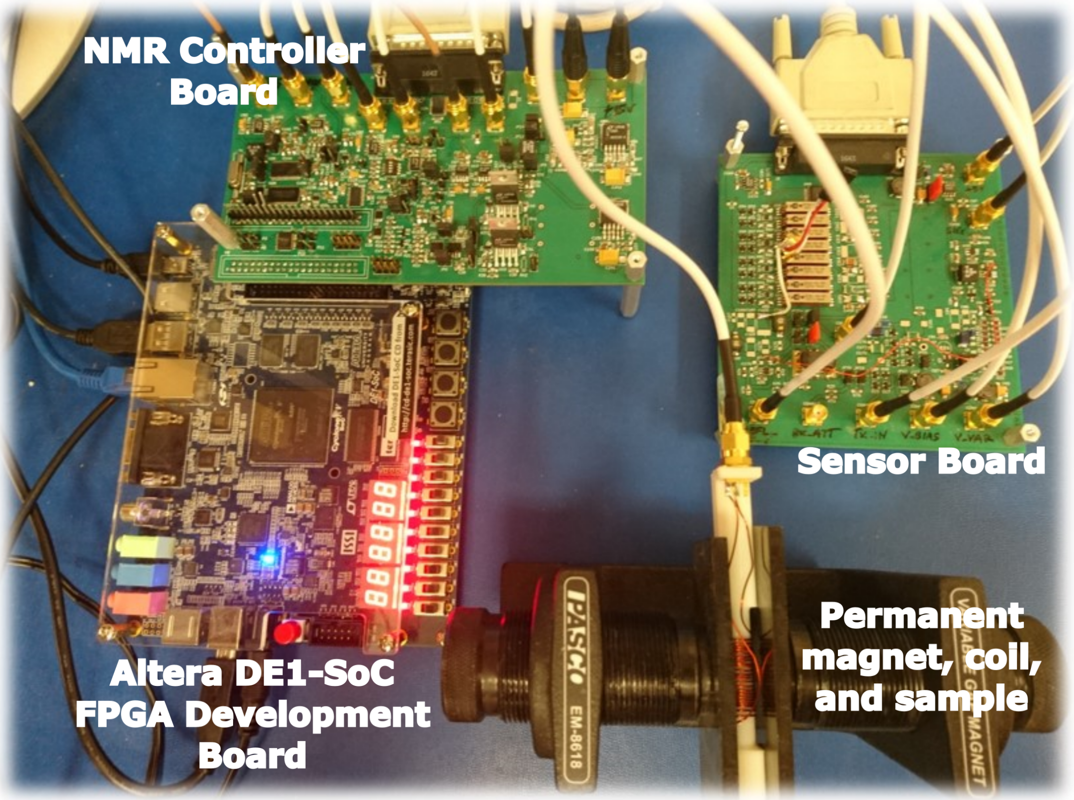

Figure A.3. Photograph of the second NMR spectrometer.

Complete References

[1] Quincy Teng. Structural Biology: Practical NMR Applications. Springer, New York, 2005.

[2] H. Y. Carr and E. M. Purcell. Effects of diffusion on free precession in nuclear magnetic resonance experiments. Phys. Rev., 94:630–638, May 1954.

[3] Kazuyuki Takeda. Chapter 7 - highly customized NMR systems using an open-resource, home-built spectrometer. volume 74 of Annual Reports on NMR Spectroscopy, pages 355 – 393. Academic Press, 2011.

[4] Kazuyuki Takeda. Opencore NMR: Open-source core modules for implementing an integrated FPGA-based NMR spectrometer. Journal of Magnetic Resonance, 192(2):218 – 229, 2008.

[5] Kazuyuki Takeda. A highly integrated FPGA-based nuclear magnetic resonance spectrometer. Review of Scientific Instruments, 78(3):033103, 2007.

[6] M.D. Twieg. Open Source NMR Relaxometry Platform. Case Western Reserve University, 2013.

[7] E. M. Purcell, H. C. Torrey, and R. V. Pound. Resonance absorption by nuclear magnetic moments in a solid. Phys. Rev., 69:37–38, Jan 1946.

[8] F. Bloch, W. W. Hansen, and M. Packard. The nuclear induction experiment. Phys. Rev., 70:474–485, Oct 1946.

[9] E. L. Hahn. Nuclear induction due to free larmor precession. Phys. Rev., 77:297–298, Jan 1950.

[10] A. Abragam. Principles of Nuclear Magnetism. Oxford [Oxfordshire], 1978.

[11] S. Meiboom and D. Gill. Modified spin - echo method for measuring nuclear relaxation times. Review of Scientific Instruments, 29(8):688–691, 1958.

[12] E. L. Hahn. Spin echoes. Phys. Rev., 80:580–594, Nov 1950.

[13] E. L. Hahn. Free nuclear induction. Physics Today, 6(11):4 – 9, 1953.

[14] Hsueh-Ying Chen, Yaewon Kim, Pulak Nath, and Christian Hilty. An ultra-low cost NMR device with arbitrary pulse programming. Journal of Magnetic Resonance, 255(Supplement C):100 – 105, 2015.

[15] L.A McLachlan. Lumped circuit duplexer for a pulsed NMR spectrometer. Journal of Magnetic Resonance (1969), 39(1):11 – 15, 1980.

[16] R. G. Vaughan, N. L. Scott, and D. R. White. The theory of bandpass sampling. IEEE Transactions on Signal Processing, 39(9):1973–1984, Sep 1991.

[17] Cheng Chen, Fengchao Zhang, Swarup Bhunia, and Soumyajit Mandal. Broadband quantitative NQR for authentication of vitamins and dietary supplements. Journal of Magnetic Resonance, 278(Supplement C):67 – 79, 2017.

[18] Soumyajit Mandal, Sangwon Oh, and Martin D. Hürlimann. Absolute phase effects on CPMG-type pulse sequences. Journal of Magnetic Resonance, 261(Supplement C):121 – 132, 2015.

[19] I. Foley, S.A. Farooqui, and R.L. Kleinberg. Effect of paramagnetic ions on NMR relaxation of fluids at solid surfaces. Journal of Magnetic Resonance, Series A, 123(1):95 – 104, 1996.