

Quantum Information Processing through Liquid NMR

Sean McKenna, John Klingner, Suzannah Wood

December 21, 2011

1 Introduction

Decades ago, the transistor was invented. With many transistors together in certain configurations, chips and integrated circuits were born. The transistor eventually found its way into a device commonly known as a computer. Ever since, more and more devices have had these semiconductors in them, which switch on or off to store information. Nowadays, a standard CPU or brain of a computer has nearly two billion of these transistors⁶ packed into a small circuit board.

These little devices are capable of storing 0's and 1's. All information processed and stored on computers can be broken down into a binary representation. This simple unit, called a bit, is related to a transistor.

Now, a new revolution is approaching: quantum computing.¹¹ With a new unit for computing, the qubit, a quantum system can store information as a superposition of states, as both the 1 and the 0. While the information must be output as just a single state, several types of problems in computing may be faster to solve using this qubit,⁵ since quantum states exist in the complex number system.

Conducting nuclear magnetic resonance (NMR) experiments on liquid species can model qubit states.^{3-5,7,9} While these experiments are relatively quick to show on an NMR device, there is a limit to the amount of qubits that can be simulated in this manner.² This and several other limitations² reduce the likelihood of this method of quantum computing becoming commercialized. However, the principles behind liquid NMR spectra illustrating quantum states as qubits is a novel application of quantum mechanics in NMR spectroscopy and information science.

2 Background

2.1 Information Science

Information science examines the storage and use of information. Currently, computing uses standard bits that limit the ability for parallelism. David Deutsch identified this key notion that current central processing units (CPUs, or the computer's brain) expect one output to be calculated before the next output can be generated.⁵ While multiple cores have become popular, any parallelism must be created in the software; the current hardware cannot support parallelism without more than one processing core. Quantum computing allows all possible outputs to be calculated simultaneously by the quantum CPU.⁵

As explained previously, the transistor is the building block for computers. With these semiconductors, devices known as logic gates can be created, which handle input and output of bits differently. Some example gates are the NOT, AND, OR, XOR, and NAND gates. A logic gate can be as simple as transmitting information or changing it depending on other information. More gates can be built by chaining other logic gates together. This is how modern computers are constructed.

In quantum computing, a useful logic gate that is commonly used is the controlled-NOT or C-NOT gate.^{4,5,9} The first qubit is a control bit that affects the second qubit's output. When the first qubit is zero, the second qubit is not changed. Otherwise when the first qubit is one, then the second qubit is swapped. This corresponds to a pass-through of the first qubit and the second qubit output corresponds to a classical XOR gate of the two. This is illustrated in Table 1. With a C-NOT gate and adding series of rotations or 90° pulses, one can construct all components of a quantum computer.¹

As a result of this new method for computing, it is possible to run several types of algorithms on a quantum computer that would run significantly faster than on a regular computer. An algorithm is simply an instruction booklet for how to solve a specific problem, like following a recipe. This quantum parallelism allows for certain near-impossible tasks to become solvable.

Input		Output	
A	B	A	B
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Table 1: Truth table for C-NOT gate

The very first quantum algorithm, created by Deutsch and Jozsa, can determine whether a function is balanced or constant.^{5,9} This artificial problem was designed to use the features of quantum mechanics to be able to witness a measurable speed-up in the solution to the problem; a quantum computer would only need to measure the function once instead of two or more times needed by a classical computer.

The most common quantum algorithm is Shor's algorithm. With the ability to find the period of a function at exponential increases, this algorithm has the potential to crack very large semiprimes into the two primes multiplied to create that semiprime.⁵ This is a method of security used for public key encryption, so one concern is that a quantum computer could more easily crack a security encryption standard that is currently used.

Lastly, another set of algorithms discovered by Grover allows for certain searches to sped up quadratically.^{5,9} For example, instead of forcing a search across all L items in a list, it could just be searched with a maximum of \sqrt{L} items using the quantum algorithm. The differences in speed are less dramatic, but for extremely large, unorganized databases or lists it would prove to be much faster and more efficient.

2.2 Quantum Mechanics

The basics of quantum mechanics lie in the duality of quantons, a generic term for what was

known as a particle. This is because quantons exhibit properties of both a particle (like a billiard ball) and a wave (like a water wave), depending on the circumstances of observation. This quirky explanation leads into the mathematics of quantum mechanics, where the symbol Ψ denotes the entire state of a quanton.

This function Ψ is called the wavefunction in quantum mechanics. Along with Born's law, this wavefunction can illustrate the probability of a quanton exhibiting a specific state or range of states, correlated with the absolute square of the wavefunction, then integrated over the range of states. This probabilistic nature is another critical component to quantum mechanics; certainty of some measurements can affect the uncertainty in others.

All quantities of the quanton can be measured from this function: position, momentum, energy, etc. These measurable values are known as observables. To find these values, Hermitian operators must operate on the wavefunction to produce the initial wavefunction and some constant. This constant is the observed value. A common observable is energy and its operator is the Hamiltonian of the system.

The mathematics behind quantum mechanics use matrices to describe quantum states, known as density matrices. Spanning across eigenspaces, the observables and operators are connected through an eigenvalue equation, the wavefunction being the eigenfunction for the matrix.

Since matrices are being used, the values in the matrix (or the state of the system) can be affected by quantum logic gates. This means that a unitary transformation operation, like a rotation, on a matrix correlates to some quantum logic gate. This understanding is critical to see how the signal output relates to the simulation of a quantum computer.

One quantum mechanical property is known as spin. Spin is a type of angular momentum exhibited by quantons. For fermions, like electrons, protons, and neutrons, the spin is associated with the value $\frac{1}{2}$, which means there are two possible, integer-spaced values, $\pm\frac{1}{2}$. Because there are just two possible values, this is a very

simple quantum system that is often referred to as spin up (0 or \uparrow) and spin down (1 or \downarrow). A quanton’s spin is a qubit.

2.3 NMR Spectroscopy

NMR is a phenomenon where nuclei react to energy as radiation, all while aligned within a strong magnetic field. The strong magnetic field is generated by a superconductor that must remain cooled, often through liquid helium and nitrogen. The strong magnetic field aligns the z-component of the nuclei’s spin along the z-axis. One type of NMR is pulsed NMR, which uses pulses to perturb the spin from its naturally aligned state.¹⁰

With the spins aligned, external radio-frequency (RF) pulses are applied to the species along the x-y plane, which cause the spins to be knocked down from their alignment with the magnetic field. The disturbed spin causes magnetization which can be measured. The energy kicked off by the nuclei as they get perturbed can be tied to a frequency by measuring the free induction decay (FID) and conducting a Fourier transform (FT) of this data over time.¹⁰

Pulses cause the spins and thus the qubits to change in each experiment, so the pulses are the equivalents for logic gates in liquid NMR.

Finding the pulse sequence for a quantum logic gate can be quite difficult. Several papers detailed that the pulses for a near C-NOT gate are just an INEPT (insensitive nuclei enhanced by polarization transfer) pulse.^{1,5} However, it is important to note that this is not a full C-NOT gate. With a working C-NOT quantum gate, only 90°_x and 90°_y pulses are needed before one can construct any other type of logic gate for a quantum computer¹. This would allow the quantum computer to simulate any basic quantum algorithm as well.

2.4 Coupled, Two-Spin States

Often times, spins in a nucleus will cancel out for all its composing quantons. When a chemical species contains more than one nuclei with a net spin, then spin coupling occurs. This creates more possible states. For a two-spin system, there are four possible states, just as there are four possible states for a 2-bit or 2-qubit system,

reinforcing the idea that the spins are the qubits. Table 2 illustrates the possible states.

Qubits	Spin
00	$\uparrow\uparrow$
01	$\uparrow\downarrow$
10	$\downarrow\uparrow$
11	$\downarrow\downarrow$

Table 2: Possible states in a two-qubit system

With multiple spins, a single quanton’s spin energy state is split into hyperfine states.¹⁰ With a two-spin system, each spin is split into two additional hyperfine states. This can be seen in an NMR spectrum by witnessing a single peak in a single-spin system and two peaks in a two-spin system. The splitting can be measured, since the interacting spins will have a gyromagnetic ratio varying by each system. This ratio depends on the frequency at which they occur and the magnetic field and Planck’s constant, or $\gamma = \frac{2\pi f}{\mathbf{B}}$.

The goal of creating quantum logic gates is to find the correct pulses that alter and exchange the four peaks in the spectra in different combinations. The four peaks come from two peaks per quanton, each quanton corresponding to a single qubit.¹

3 Experiment

3.1 Sample Preparation

To simulate a two-spin system, a sample of ^{13}C -chloroform was used, structure in Figure 1. The isotope of carbon and proton both have a net spin of $\frac{1}{2}$, making this a simple, ideal heteronuclear two-qubit system.^{5,9} When representing the different states, the proton’s spin state is represented first, followed by the carbon’s state, as in Table 2.

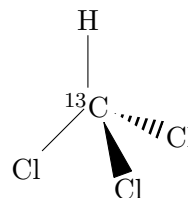


Figure 1: ^{13}C -chloroform molecular structure

The ^{13}C -chloroform was purchased from ICON Isotopes, #IC3186. Two samples were prepped utilizing standard NMR preparation by Suzannah Wood. She made each sample in a

0.5mL NMR tube and 7% by weight of chloroform and the solvent of Acetone-D6. The solvent provides an NMR locking signal that does not interfere with pulses and measurements.⁹

In testing each sample, no significant differences were found between the two tubes, but no experimental results compared the potency or strength of the sample in each tube either.

3.2 NMR Spectrometer

All data was collected using a JEOL Eclipse 270 MHz spectrometer at room temperature. The spectrometer spins the sample while conducting pulsed NMR spectroscopy. After collection, the Delta v.4.3 software was used to process the free induction decay (FID) with the standard proton and carbon processing lists in order to produce a spectra by Fast Fourier transform (FFT).

Since the carbon channel had a lower signal-to-noise ratio, these results had more noise and were more difficult to ascertain a good baseline for integrating the peaks. As such, the experiments conducted focused on the proton spectra, which still illustrates the four possible states of the two-qubit system.

3.3 Experiment Programming

Learning how to program the experiments using the Delta software was the most time-consuming portion of the research. Each experiment file is associated with one pulse sequence. The complications came in both transferring the diagrams in other research papers to this language and the limitations of the pulse programming language.

For example, the manual never stated that IF statements could not encapsulate a COLLECT statement. This meant that every experiment can only collect either proton or carbon spectra, and the only way to change the output was to manually create a separate experiment file for this purpose. The only way to find this out was to write the experiment file and see if it opened or not in the Delta software.

Collection occurred automatically on the channel that was selected. Using a COLLECT statement, either the OBS or IRR channel could be used to collect carbon and proton spectra, respectively.

3.4 Programming Pulses

To continue with the research, basic pulses had to be programmed at the end of the experiment file. The software allows control of the frequency, pulse width (timing), and phase of the pulses.

The frequency of the pulses would be set to either the proton or carbon channel. Each nucleus was slightly different, so its energy differences would resonate at a different frequency. Due to the hardware limitations, only the OBS channel could send pulses for the carbon and only the IRR channel could send pulses for the proton. Setting the frequency was done automatically when defining the domain of the channel. Once the channel's domain is set, then the remaining pulse parameters can be defined.

The pulse width is directly related to the timing. The pulse width was determined by default values in the program. A 90° pulse would knock the nuclei spin from the z-axis into the x-y plane to be measured. For ¹³C, this value was 8.8μs and the 90° pulse for proton was 9.8μs, provided by default through the Delta software. A 180° pulse is simply two 90° pulses in a row. Pulses could also be simultaneous with different alignment possibilities (leading, centered, or trailing).

Occasionally, a pulse sequence would use a wait time, which is just defined as a set length of time where no pulses would be sent. These wait times are key to receiving the correct output states. Very often, these wait times are determined by a fraction of the coupling constant for the system, J. The coupling constant for the ¹³C-chloroform system is approximately 215 Hz.

The phase corresponds to the axis of the RF pulse. The most common phases are x, y, -x, and -y. In the program, this corresponds to values of 0, 90, 180, and 270, respectively.

Lastly, the method of phase cycling was utilized for the experimental results. Phase cycling incorporates many repeated experiments while adjusting the axis of measurements. The relative axes (angle between) remain consistent, but the measurements average out inconsistencies in the NMR tube spinning at 15 Hz.

3.5 Temporal Averaging

Since all data was taken at room temperature, the spectra would be in a combination of all four possible states, a thermal state. As such, this means there is not any singular input state, like the ground state, to start with for any quantum experiment. To simulate a ground state, the method of temporal averaging was performed.

This method involves taking three experiments on different states and adding the results to yield the ground state of the system. It works by exploiting the fact that NMR measures a difference of states. As such, it is possible to emulate a ground state by having the differences of states cancel out for all but the ground state. This means that a total of twelve experiments would be required to attain the four different input states.⁸

The first two experiments, labeled as A and B permutations in the data, were created by pulse sequences found to be detailed out for both the carbon and proton channels.⁸

Additionally, the third and final state used was the thermal state of the sample. This was generated using several different methods. First, a description of a customized thermal state was found in one resource which showed a very clear proton thermal state. Then, a simpler set of pulses (just a 90° pulse to each channel) was tested and sent to the sample to reveal similar results. The data presented here is the result of the simpler pulse sequence.

With the three experiments per state, the processed FFT data was exported from Delta as text into Mathematica where processing averaged the three FFT's together. This was done by taking the relative intensity at each frequency such that each single FFT was normalized from -1 to 1.⁸ Adding the three normalized experiments together produced a proper ground state.

From there, the ground $\uparrow\uparrow$ state was hit with a combination of pulses to produce the remaining three initial states. John Klingner was able to simulate some of the matrices in Mathematica and produce the pulse sequences to flip either carbon spin ($\uparrow\downarrow$) or proton spin ($\downarrow\uparrow$) into the proper states. Combining both pulse sequences produced the highest-energy $\downarrow\downarrow$ state.

3.6 C-NOT Gate

With the four initial states, a quantum logic gate could be created. As the simplest, effective gate for two qubits,^{4,5,9} implementing the C-NOT gate was the next experimental task. Two pulse sequences were tested and did not come close to producing proper C-NOT gate results.⁵ The final pulse sequence that was tested to perform the C-NOT operation on the two-qubit system was detailed in research by Gershenfeld and Chuang.⁴ This sequence was appended onto each experiment's temporal averaging preparation pulses, and it turned out to be nearly successful. Again, this resulted in twelve additional experiments, three for each initial state.

4 Analysis

4.1 Data Analysis

The first part of data analysis involved adjusting the baseline of the resulting temporal averaged spectra. This is because the spectra had resulting noise and bumps. Before analyzing the combination of states, the baseline had to be cleared up. Using the ACD Labs NMR Processor v.12 software, the spectra were cleaned up using spectrum averaging of 80 points and noise factor 75, with minor adjustments to ensure no sharp, jagged lines in the resulting, baselined spectra.

With the corrected spectra, the integrals on each side of the spectra were measured. The integration of the real-valued spectra correlates to the probability of the existence of different states. Since ideally there would only be one state resulting, the integral of a non-peak should be approximately zero, while the peaks themselves are normalized to 1. As such, an approximate error can be calculated for each spectra based on the non-zero (or less than one) integration terms. Integration values reported here were also calculated utilizing the ACD NMR software.

State	$\downarrow f$	$\uparrow f$
00	1.000	-0.259
01	-1.000	0.222
10	-0.560	1.000
11	-0.205	-1.000

Table 3: Normalized integral values for the different initial preparation states

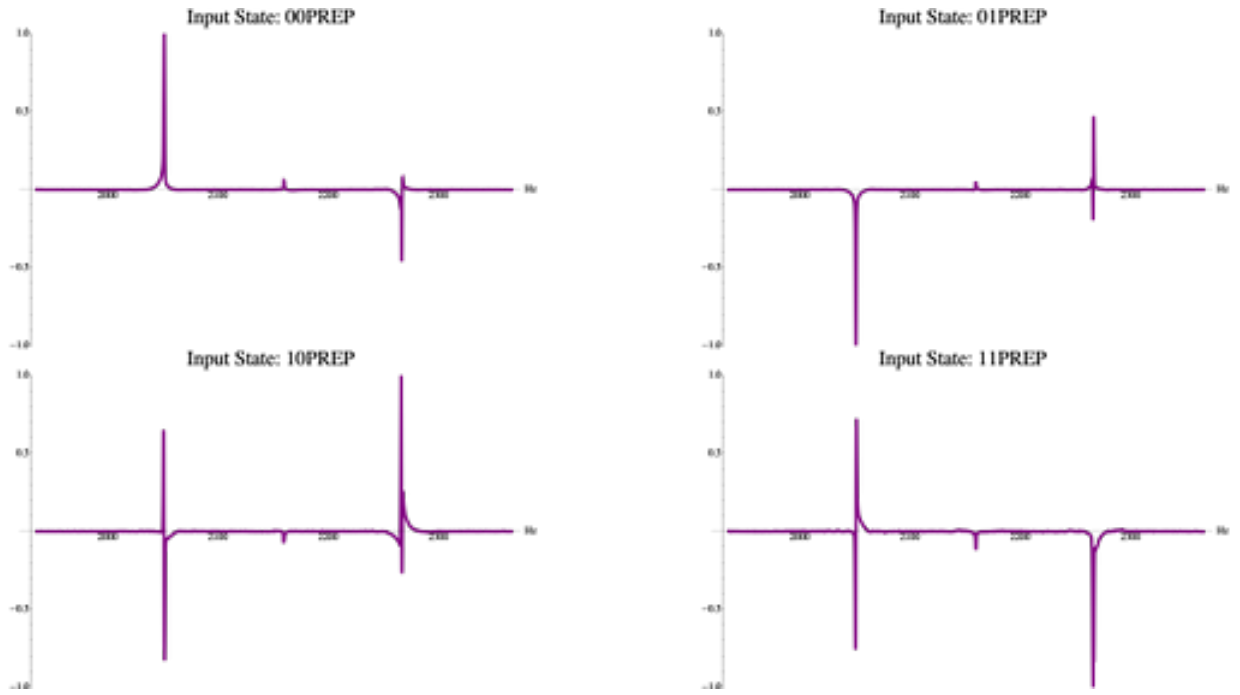


Figure 2: Normalized spectra for the four initial states, frequencies given in Hz from standard proton resonance frequency. The lower frequency peaks (left) corresponds to the proton's 0 bit, positive peaks correspond to the carbon's 0 bit.

4.2 Initial States

The initial states produced from the pulse sequences from temporal averaging correspond to the pulses seen above in Figure 2. Important features include the positive peaks for two states with carbon spin up (\uparrow or 0), and the two peaks at lower frequency (lower energy) correspond to the proton spin up (\uparrow or 0). All four states are represented with these proton spectra.

Additionally, the integral values at each peak frequency were calculated, presented in Table 3. As can be seen from the table, there is approximately 20%-25% error for most states. The only problem is the 10 or $\downarrow\uparrow$ state, since it had over 50% error. It is quite possible that the preparation of this state was not entirely complete using the custom pulse sequences. Looking at the spectra, this state is also the least clear peak of them all. Since a 11 or $\uparrow\uparrow$ state was created by using part of this same pulse sequence, it is interesting to note that this state had the least error associated with it for the initial states. It is possible that the procedure for baselining the spectra may have played a role in the larger and smaller values of error, though.

4.3 C-NOT Gate

With the initial states, a C-NOT gate was attempted. All but one state worked, the 10 or $\downarrow\uparrow$ state. The resulting truth table and integration values for the experiment is in Table 4. The spectra for each input state are shown in Figure 3, sorted in order of input state.

Input	Output	$\downarrow f$	$\uparrow f$
00	00	1.000	0.143
01	01	-1.000	0.183
10	10	0.002	1.000
11	10	0.062	1.000

Table 4: Truth table and normalized integral values for experimental C-NOT gate

Oddly enough, the integration values for the C-NOT gate were much lower than the initial states. The overly complex pulse sequence may have done something to ensure proper transfer of states more effectively. The baseline may have been stricter to the resulting FFT because of the greater noise in the C-NOT gate, too. Aside from the one erroneous state, the logic gate did what it was expected to do.

Since a C-NOT gate was not completely implemented, there must have been something

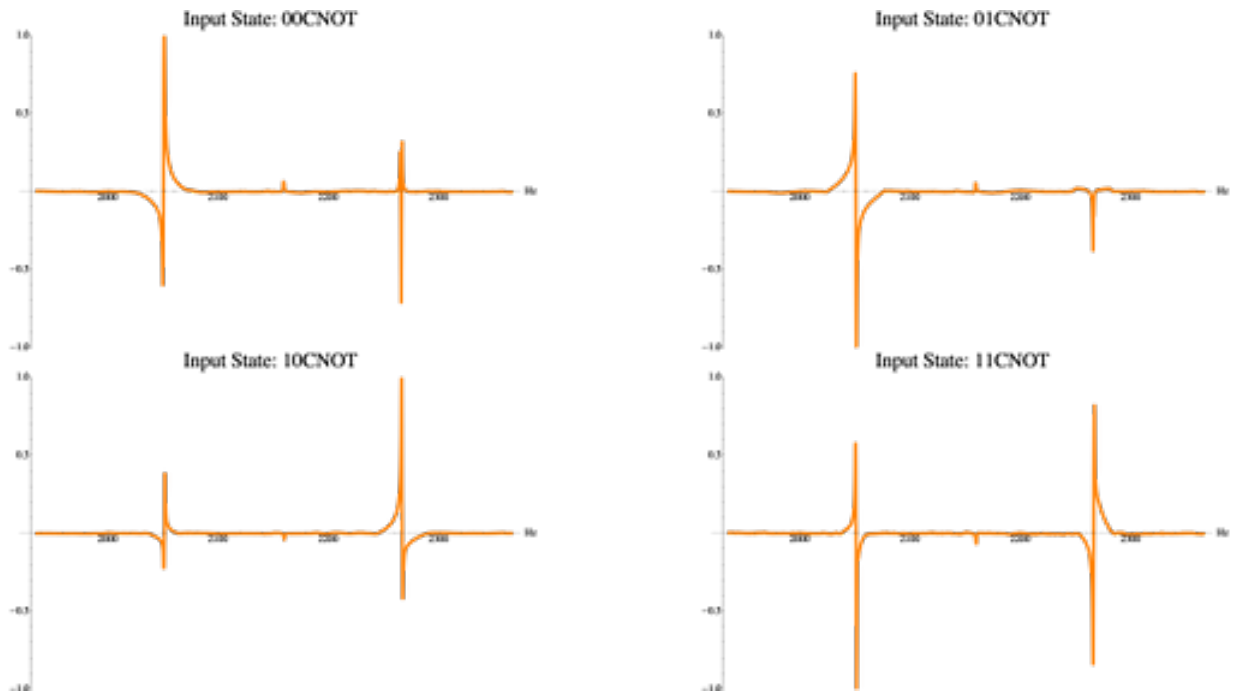


Figure 3: Normalized spectra for the C-NOT logic gate, only one state was incorrect

wrong with the pulse sequence used, how it was programmed, or incorrect input states were used. It is highly probable that the 10 or $\downarrow\uparrow$ state was not prepared properly due to its high error in the preparation state, resulting in an incorrect value for the C-NOT gate.

5 Conclusion

The very basics of quantum computing using liquid NMR was demonstrated with these experiments. Utilizing a two-qubit system, the four initial states were prepared by combining pulses on the ground state, all conducted via temporal averaging. With all four initial states, a simple quantum logic gate (C-NOT) was implemented successfully on three out of the four inputs. With further time and understanding, a complete C-NOT gate and simple pulses could be combined to demonstrate quantum algorithms, using liquid NMR.

References

[1] Chuang, I. L., Gershenfeld, N., Kubinec, M. G., and Leung, D. W. (1998). Bulk quantum computation with nuclear magnetic resonance: theory and experiment. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):447–467.

[2] Cory, D. G., Boulant, N., Boutis, G., Fortunato, E., Pravia, M., Sharf, Y., Laflamme, R., Knill, E., Martinez, R., Negrevergne, C., Zurek, W. H., Havel, T. F., Viola, L., Lloyd, S., Weinstein, Y. S., and Teklemariam, G. (2005). *NMR Based Quantum Information Processing: Achievements and Prospects*, pages 105–137. Wiley-VCH Verlag GmbH & Co. KGaA.

[3] Cory DG, Fahmy AF, H. T. (1997). Ensemble quantum computing by NMR spectroscopy. *Proc Natl Acad Sci U S A.*, 94(5):1634–1639.

[4] Gershenfeld, N. A. and Chuang, I. L. (1997). Bulk spin-resonance quantum computation. *Science*, 275(5298):350–356.

[5] Grant, D. M. (2002). *Liquid State NMR Quantum Computing*, volume 9. Wiley.

[6] Intel (2011). Microprocessor quick reference guide. www.intel.com/pressroom/kits/quickreffam.htm#Xeon. [Online; accessed 7-December-2011].

[7] Jones, A. (2010). Nuclear magnetic resonance quantum computation.

<http://nmr.physics.ox.ac.uk/pdfs/lhnmrqc.pdf>.
[Online; accessed 30-November-2011].

<http://web.mit.edu/8.13/www/49.shtml>.
[Online; accessed 28-November-2011].

[8] Knill, E., Chuang, I., and Laflamme, R. (1998). Effective pure states for bulk quantum computation. *Phys. Rev. A*, 57:3348–3363.

[9] MIT (2010). Quantum information processing with NMR.

[10] Roberts, J. D. (2000). *ABCs of FT-NMR*. University Science Books.

[11] Warren, W. S. (1997). The usefulness of NMR quantum computing. *Science*, 277(5332):1688–1690.