

# Quantum Information Processing

Evan Berkowitz

evan\_b@mit.edu

Junior, MIT Department of Physics

March 9, 2007

We present an implementation of the quantum logic gate CNOT via pulsed NMR rotations. After confirming its truth table, the Deutsch-Josza algorithm is implemented and tested with positive results. In order to determine the reliability of the quantum technique, we show that the timescale on which the computation is done is much shorter than the NMR properties  $T_1$  and  $T_2$ .

## 1. HISTORY

In the 1965 Gordon E. Moore, a co-founder of Intel, noted that chip complexity was doubling every 24 months or so. The cost of producing chips with a higher transistor density was dropping as the transistors shrank. However, traditional transistors rely on doping, a material property, to function. How small could they make transistors? Obviously transistor density could not double forever. What was the limit?

Since doping is a property of a large ensemble of molecules instead of a molecular property, it was obvious that traditional transistor technology could not be shrunk to the atomic scale. Currently, the latest traditional transistors to be incorporated into consumer products are built at a 45nm process, which is approximately 400 silicon atoms wide. Moreover, as the scale shrinks the classical electrical laws are overcome by quantum effects, and these effects would harm the operation of the transistor.

Is it possible to do computation on the atomic scale? It seems as though quantum mechanics destroys the hope of familiar computation. However, it is possible to do computation with quantum systems, taking advantage of their unusual properties. Briefly, a quantum computer can operate on a superposition of inputs, whereas a classical computer can only operate on one input at a time.

## 2. CLASSICAL AND QUANTUM COMPUTATION

Classical computers store information as bits, where each bit is the answer to a yes or no question. One bit can hold one decision, with 1 being “true” and 0 “false”. In computers, computation is built out of logic gates such as *NOT*, a unary operation which flips the value of the bit; *AND*, a binary operation which returns 1 if and only if both inputs are 1; *OR*, a binary operation which returns 1 if either or both inputs are 1; and other operations. Logic gates are built out of clever combinations of transistors. Building complex strings of these gates allows any computation. Some operations are universal, meaning that using only that operation alone one can

construct all other operations. *NAND*, the gate which gives 1 if *AND* gives 0, and 0 if *AND* gives 1 (named because it is *NOT* applied to *AND*), and *NOR* (*NOT* applied to *OR*) are both universal.

Quantum and classical computers can solve exactly the same class of problems. Any problem a quantum computer can solve is also solvable classically. The only advantage quantum computers have over their classical counterparts is the speed at which they perform their computations. Some problems which may take years to solve classically can be solved quickly on a quantum computer.[1] Problems for which we have quantum algorithms which are faster than their classical counterparts include database searching, factoring, and determining whether a given function is “fair”. The final problem is solved by the Deutsch-Josza Algorithm, and will be explored further.

There are many analogies to be drawn between quantum and classical computation. In a quantum computer data is stored in qubits, where each qubit is an arbitrary linear superposition of a 1 and a 0, taking advantage of the superposition principle of quantum mechanics. Quantum computers have gates, but instead of operating electrically, they manipulate the quantum-mechanical state of the system in some way.

Quantum computation also has some gates which are purely quantum mechanical. For example, assuming the system is made of  $k$  qubits, the Hadamard gate (denoted  $H^{\otimes}$ ) takes the zero state and rotates it into a superposition of all  $2^k$  possible states. Additionally, a gate which flips a 0 into an equal superposition of 0 and 1 can be called  $\sqrt{NOT}$ , because applying it twice will take the state from 0 to 1 and vice versa. These gates rely on superposition, and thus are purely quantum-computational. The essence of the advantage of quantum computation may be seen in  $H^{\otimes}$ , as it allows a computer to operate on all the possible inputs at once instead of looping over all inputs one at a time as a classical computer might.

If we add a third operation, denoted *CNOT*, we have a universal set of operations, meaning that our computer can perform any computation. In some sense, *CNOT* is the quantum-mechanical *XOR*, as shown in Table I. However, *XOR* is lossy, meaning that given its output,

one cannot deduce its input, as *XOR* maps two bits into one. *CNOT*, on the other hand, is a unitary operator, and therefore can be undone via its conjugate. *CNOT* on 2 bits operates in the following manner: if the first label on  $|\Psi\rangle$  is 0, do not change the second label, and if the first label is 1, flip the second label. Thus, the first bit controls whether the second bit has *NOT* applied, giving this operation the name *Controlled NOT*.

TABLE I: The truth table of *XOR* and *CNOT*. Note that the second bit of the output of *CNOT* is identical to the output of *XOR*

$A$	$B$	$A \text{ XOR } B$	$ \Psi\rangle$	$CNOT \Psi\rangle$
0	0	0	$ 00\rangle$	$ 00\rangle$
0	1	1	$ 01\rangle$	$ 01\rangle$
1	0	1	$ 10\rangle$	$ 11\rangle$
1	1	0	$ 11\rangle$	$ 10\rangle$

### 3. EXPERIMENTAL APPARATUS

Our quantum computer was, in essence, a pulsed NMR machine with a software interface. The RF circuit was in a cooled thermos in order to provide a better signal to the sample. Additionally, within the thermos there were magnetic shims, which could be adjusted if the signal indicated that the sample was experiencing inhomogeneities. The RF circuits were controlled by signal generation units and frequency control units. Observations were taken as free induction decays picked up probe circuits, passed through the signal chain, into a Linux machine running XWIN-NMR (Bruker’s control software), and then Fourier transformed to show a frequency spectrum. XWIN-NMR interfaced with Matlab running on an Athena machine via a home-written software package. Matlab scripts were written for systematic experimentation. In NMR,  $\sqrt{NOT}$  is equivalent to a  $\frac{\pi}{2}$  pulse, as a  $\pi$  pulse will completely switch the direction of spin.

The software link between XWIN-NMR and Matlab was very delicate. It requires a specific, locally-installed version of Matlab, and not the version which generally available at MIT. Using an incompatible version of Matlab reliably caused bugs and occasional kernel panics in the Athena machine.

The two qubits of our quantum computer are the hydrogen and carbon-13 nuclei in a sample of chloroform ( $\text{CHCl}_3$ ). It is important that the nuclei have an odd number of nucleons, as if it had an even number there would be an equal number of spin-up and spin-down nucleons, giving no moment. Note that each manipulated spin is actually the sum of spins of a large ensemble. In a magnetic field pointing up, at room temperature, the spins are very nearly divided evenly between spin-up

and spin-down, with slightly more up than down. Thus, operating on and observing these states will not give easily interpreted results. Fortunately, a technique known as thermal averaging uses a series of manipulations to take three different observations, the average of which will behave as though the state was pure, and not mixed. Thermal averaging capability was built into the Matlab package and was used throughout the computational procedure. Since hydrogen and carbon have very different NMR properties, they may be manipulated independently. However, the two are coupled via their chemical bond, allowing quantum mechanical computation: if they were not coupled, they would simply be very expensive substitutes for electrical storage.

One calibration required was of the overall phase of the circuit. By taking a single spectrum we could tell the overall phase and correct for it. If  $p$  is the value of a peak integral, the peak integral can be rotated to be mostly real by an angle  $\phi$ , given by  $\pi - \arctan(\frac{Im(p)}{Re(p)})$ . Averaging  $\phi$  for each peak integral of a given sample gave us the phase which we used to calibrate. After the rotation by this averaged  $\phi$  the hydrogen peak integrals were approximately 0.5% imaginary, and the carbon peaks were approximately 0.3% imaginary but much more noisy. The results of a hydrogen calibration is shown in Figure 1. Since the ordinate is measured in arbitrary units, we will only concern ourselves with its sign, and will drop the scale in following figures.

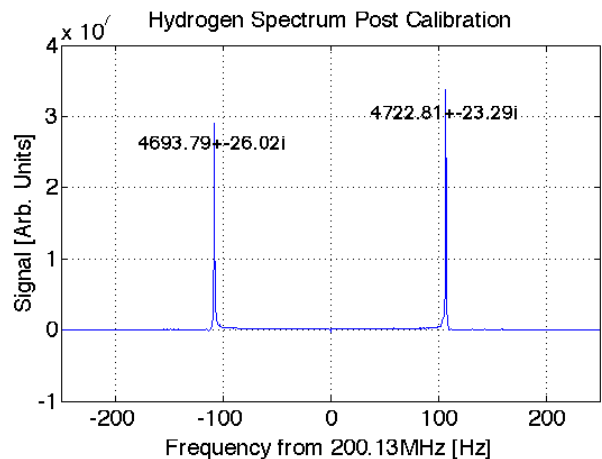


FIG. 1: The results of a typical calibration. The peak integrals may be positive or negative. Plots will always be signal against frequency. For clarity, the ordinate will not be labeled, and no scale will be given.

From observations like these, the splitting between the lines,  $J$ , was determined to be  $215.18 \pm 0.12$  Hz. Over eight hydrogen observations, the peaks were always in the exact same place. This indicates that the resolution of the hydrogen measurement was finer than the bin width in Matlab. The error on  $J$  is half the width of a bin. As  $J$  is a spin coupling parameter, it is the same for both

carbon and hydrogen. The free evolution period is given by  $\frac{1}{2J}$  which evaluates to  $2.323 \pm 0.003$ ms.

#### 4. NATURAL TIMESCALES AND RELIABLE COMPUTATION

In order to do reliable computation, we must guarantee that the process takes less time than the system takes to lose coherence. By examining the NMR timescales  $T_1$  and  $T_2$  and comparing them to the  $\frac{\pi}{2}$  pulse width we can decide whether there is sufficient time to compute or not.  $T_1$  was found by searching for a minimal response to a  $\pi$ - $\tau$ - $\frac{\pi}{2}$  sequence[2]. The minimal response was found at  $13.5 \pm 0.6$  seconds. However,  $T_1$  is the time to recover by a factor of  $e$ , which is  $\ln 2$  longer than the 50% recovery time. Thus,  $T_1$  is  $19.5 \pm 0.9$  seconds. As  $T_2$  is the inverse of the line width of the peaks, is on the order of 1 second, as the peaks are approximately 2 or 3 bins wide, and 1 bin corresponds to  $\frac{1}{4}$  Hz.

For the system to remain reliable, we demand that  $T_1$  and  $T_2$  is much longer than the time it takes to perform the computation, which is on the order of the  $\frac{\pi}{2}$  pulses. Knowing the  $\frac{\pi}{2}$  pulse widths is also vital to ensure transform the states, and thereby perform the computation, properly. These pulse widths were determined by sampling between 1 and 15  $\mu s$  and fitting the response to curves. Once the curves were fit, the maximum responses for hydrogen and carbon were found at  $9.3 \pm 0.2 \mu s$  and  $8.8 \pm 0.3 \mu s$ . This data can be examined in Figure 2.

As a  $T_1$  and  $T_2$  are on the order of seconds, and the  $\frac{\pi}{2}$  pulses are on the order of ten microseconds, it is fair to say that the computation will complete before the system loses coherence.

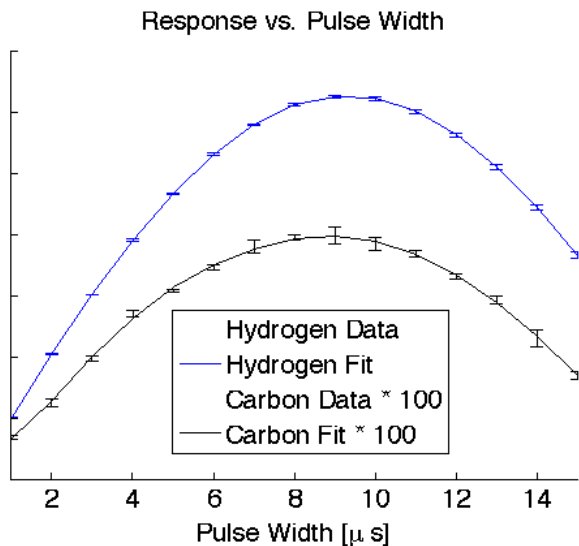


FIG. 2: Sample response as a function of pulse width. The carbon data is shown on a scale 100 times smaller than the hydrogen data.

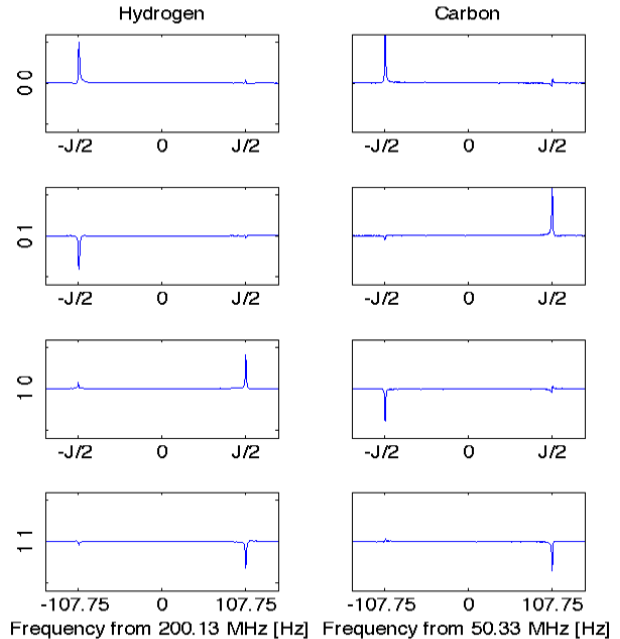


FIG. 3: The four basis states shown as both hydrogen and carbon spectra.

#### 5. COMPUTATIONAL BASIS STATES

The computational basis states are thermally-averaged states, constructed out of the equilibrium state. Since we have two spins on which to operate, we can make  $2^2$  orthonormal states. Let 0 represent “up,” (parallel to the magnetic field) and 1 represent “down” (antiparallel to the field). Also, when we write a state, we identify the left label as the state of the carbon and the right label as the state of the hydrogen. Thus, we can generate  $|01\rangle$  by applying two  $\frac{\pi}{2}$  pulses to the hydrogen of  $|00\rangle$ ,  $|10\rangle$  by applying two  $\frac{\pi}{2}$  pulses to the carbon of  $|00\rangle$ , and  $|11\rangle$  by applying those pulses to both.

The computational basis states are shown in Figure 3. It is important to note that the observation of either carbon or hydrogen will suffice to decide the state of the system. The hydrogen peak can be up-left, down-left, up-right, and down-right, in the four respective states  $|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$ , and  $|11\rangle$ . Thus, the measurement of hydrogen alone is enough to tell us the state of the whole system. Indeed, once the system is observed the correspondence between the state of the hydrogen and the state of the carbon is destroyed, so it is important to be able to discern all the information available on one of the spins. Similarly, carbon can be in the four different positions, making either nucleus readable. This means that the truth table of *CNOT* would be convincing if only one of the nuclei’s spectrum was shown. In contrast, the Deutsch-Josza experiment uses one bit as an

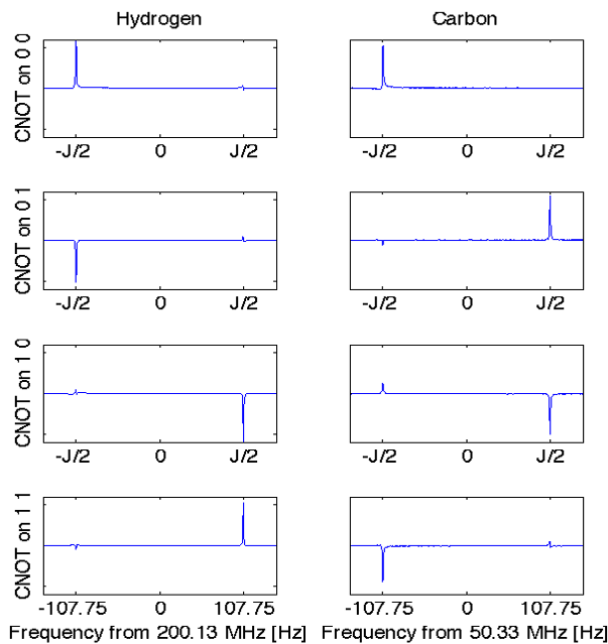


FIG. 4: The results of  $CNOT$  applied to the four basis states shown as both hydrogen and carbon spectra. It is clear from comparison with the basis states that  $CNOT$  has exchanged  $|10\rangle$  and  $|11\rangle$ .

input/output bit, and the other bit as working space.

## 6. $CNOT$

The  $CNOT$  pulse sequence used was given by

$$CNOT = R_{\bar{y}_2} R_{x_2} R_{y_2} R_{y_1} \tau R_{x_1} R_{\bar{y}_1} R_{\bar{x}_1} R_{y_1}$$

where  $R_{x_1}$  is a  $\frac{\pi}{2}$  pulse applied to the first qubit about the  $x$ -axis and  $R_{\bar{y}_2}$  is a  $\frac{\pi}{2}$  pulse applied to the second qubit about the  $-y$ -axis, etc. The matrix representations of these operators can be written and examined closely, and are available as an appendix in [3]. This pulse sequence attains the exchange of states 3 and 4, and adds an arbitrary overall phase, which is unobservable. The results of applying this pulse sequence to the four basis states are shown in Figure 4. By comparing the two figures, it is clear that  $CNOT$  functions properly, in that it swapped  $|10\rangle$  and  $|11\rangle$ , in accordance with Table I.

## 7. THE DEUTSCH-JOSZA ALGORITHM

Suppose we had a function which took a mapped one bit into another bit. The four possible functions are given in Table II. Our goal is to determine whether a given mystery  $f$  is fair (meaning it returns 0 on half of the inputs and 1 on half) or unfair (it returns the same bit for

all inputs). By constructing pulse sequences out  $\pi$  pulses and  $CNOT$  it is possible to create operations which take the input bit  $x$  and replace it by the bit given by  $f(x)$ . These pulse sequences were sought by trial-and-error using a Matlab simulation, though some transformations are clear ( $f_2$  is simply the identity). Those sequences were surrounded by  $H^{\otimes \dagger}$  and  $H^{\otimes}$ .

Extending this problem to  $k$  inputs, a classical computer must check  $\frac{k}{2} + 1$  possible inputs before deciding the nature of  $f$  with 100% confidence. The Deutsch-Josza algorithm allows the application of  $f$  once, on a superposition of all possible inputs, distinguishing fair functions from unfair functions. If cost is associated with applications of  $f$ , the Deutsch-Josza algorithm offers a big improvement over the classical mechanism. In our instance of the problem,  $f$  is a function of one qubit to one qubit, and thus only one nucleus will contain correct distinguishing information. We arbitrarily chose to perform the pulse sequence such that the output would be on the carbon and so the hydrogen would be scratch space, and equivalent pulse sequences can easily be written such that the output is on the hydrogen. As  $f_0$  and  $f_3$  are unfair, the Deutsch-Josza algorithm should distinguish these  $f$  from  $f_1$  and  $f_2$ . The results of the pulse sequences are shown in Figure 5. Thus, we have successfully distinguished between the fair and unfair  $f$ .

TABLE II: The truth table of  $XOR$  and  $CNOT$ . Note that the second bit of the output of  $CNOT$  is identical to the output of  $XOR$

$x$	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
0	0	1	0	1
1	0	0	1	1

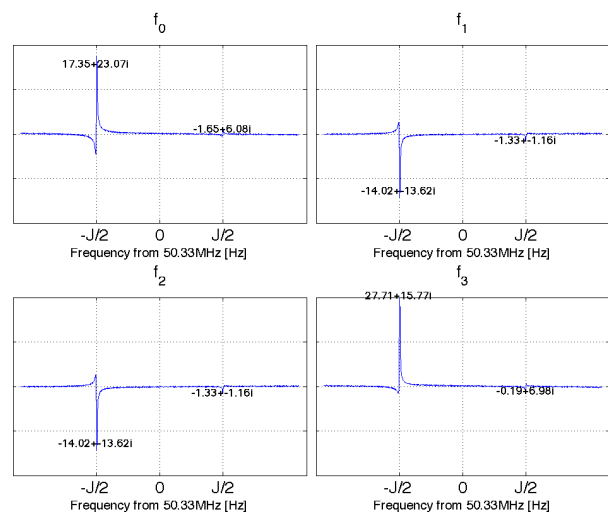


FIG. 5: The spectra of the informative bit in the Deutsch-Josza algorithm. Note the fair  $f$  had negative peak integrals while the unfair had positive.

ecute an algorithm provably faster than a classical algorithm can. These results have confirmed the feasibility of quantum computation and its advantage over classical computation. The long decoherence times makes the NMR implementation of quantum computing easy, and the technique's success is apparent.

## 8. CONCLUSIONS

We have successfully shown *CNOT*'s operative truth table and have used the Deutsch-Josza algorithm to ex-

- 
- [1] M. Sipser, *18.440 -  $A_{TM}$  is Undecidable* Massachusetts Institute of Technology (2006).
  - [2] E. Berkowitz, *Pulsed Nuclear Magnetic Resonance*, [http://web.mit.edu/~evan\\_b/Public/8.13/Papers/nmr.pdf](http://web.mit.edu/~evan_b/Public/8.13/Papers/nmr.pdf) (2006).
  - [3] S. Sewell, *Quantum Information Processing with NMR*, <http://web.mit.edu/8.13/www/JLExperiments/JLExp49.pdf> (2007).