

## Chapter 7

### Quantum Logic with Trapped Ions

Quantum computation, is a relatively new and thus far mostly theoretical field. It is the result of the marriage of computation theory with physical theory. Many of the pioneers of classical information and computation theory (such as Shannon, Turing, Church, and Gödel) considered information in a purely abstract manner. This led to many insights into the nature of computational complexity [152, 153]: for example, the role of the binary *bit* as a fundamental carrier of classical information, the fact that any algorithm can be implemented in terms of fundamental two-bit logic gates, the existence of complexity classes for algorithms, and the existence and significance of universal computing machines (for example, the Turing machine).

However, gradually people began to realize that information is always encoded in *physical* systems [37]and, therefore, that physics had an essential role in the theory of computation. As an example, the consideration of minimum energy dissipation for computational elements [154] led to considerations of “reversible logic gates,” and the discovery that they were computationally equivalent to the usual, irreversible logic gates (e.g. AND, NOR, etc.). On the other hand, it also became apparent that information theory was a useful point of view for considering physical law. Consider, for example, the relationship between Shannon information and the physical entropy. As another example, the realization that it costs no information to store information, but only to

erase it — an idea which arose in the aforementioned study of reversible logic — offered a resolution of the paradox of “Maxwell’s Demon” [155].

The idea that “information is *physical* [37],” along with the realization that computational elements were growing physically smaller, led Deutsch [156] to consider information elements that were inherently quantum. At the same time the realization that classical computers could not efficiently simulate quantum systems (because of the growth of the dimensionality of Hilbert space with system complexity) led Feynman [157, 158] to propose that “quantum computers” might be able to simulate such systems more efficiently.

The question of efficiency (from the point of view of computer scientists) is one of how the resources required to implement some algorithm grow with the size of the input to the algorithm. Note that, from a computational complexity point of view, “resource” refers both to the number of time-steps required to implement the algorithm and the amount of physical resources (e.g. logic gates) required to implement it. Although there are different complexity classes [152], we can obtain some idea of the divisions by noting that, for example, some algorithms scale as some polynomial of the number  $N$  of bits in the input whereas others scale exponentially with this number. Algorithms in the latter class are called “computationally hard.” For example, calculating the product of two numbers scales as a polynomial in the size of the numbers, whereas it is believed that factoring a given number into its prime factors is a “hard” problem.

This problem of factorization actually lies at the heart of much of the recent excitement about quantum computation. The computational difficulty of factoring large numbers lies at the heart of popular data-encryption schemes (such as the RSA [159] protocol). So, Shor’s [39] extension of Deutsch’s discovery [38] that quantum computers could efficiently compute algorithms that were “hard” on classical computers caused quite a stir — Shor discovered a quantum algorithm that could efficiently factorize large numbers!

In general, then, quantum computation and quantum information theory deal with information in a physical and explicitly quantum context. Although the field is new, it has caused quite a stir in several different contexts. Of course, it is still an open question as to whether a quantum computer could actually be built (for reasons I will discuss below). However, quantum computation is significant for several reasons: in general, I would make the following points.

- (1) Classical information theory showed that the difficulty of algorithms was independent of the particular logic used to implement them. So, for example, changing the hardware used to compute an algorithm, or using three-state logic instead of binary logic does not move a problem from being “hard” : (super-polynomial or exponential in the input size) to “easy” (polynomial in the input size). However, there are problems which are “hard” on *any* classical computer which are “easy” on a quantum computer. This is a revolution in information theory, and the impact of the *concept* of quantum computation on information-theoreticians may be independent of the technical issue of whether we can actually build a quantum computer in practice.
- (2) It appears that much of the “heart” of quantum mechanics (for example, the collapse or apparent collapse of the wave function) deals with the transfer of information from subsystem. The new language offered by quantum information theory may well offer new insights into these issues [42, 43]. Also, a general quantification of entanglement in many-particle quantum systems is lacking at present. The language of quantum computation may shed light on the subject. Again, the significance of this language may well be independent of the question of whether a “quantum computer code-breaker” can ever be built.
- (3) As originally pointed out by Feynman [158], the exponential growth of the Hilbert space dimensionality in quantum systems makes simulating such sys-

tems difficult. Quantum computers, if they can be built, could efficiently simulate other quantum systems. That this may be of interest is indicated by the recent Nobel prize awarded for advances in the (classical-computer) molecular dynamics approach to simulating problems in quantum chemistry. Of course, the question is often asked “why not just measure the actual system, rather than simulate it on a quantum computer?” The answer, of course, is that sometimes (e.g. biotechnology) it would be desirable to understand a complex quantum system before investing in the infrastructure necessary to realize it. A general-purpose quantum computer could enable this. In other situations (e.g. QCD), it is not possible experimentally to investigate all regions of a theory’s parameter space or to change, for example, coupling strengths or interaction Hamiltonians. In such situations, quantum computers could make significant contributions. This, of course, would depend on it being possible to construct a quantum computer!

- (4) Finally, quantum computers may enable us to implement important algorithms which would be infeasible or impossible on any classical computer. For example, since the best possible (known) classical factoring algorithm scales exponentially with the size of the number to be factored, factoring a 400-digit number would require a computer with more atoms than in the universe! On a quantum computer, it would be possible to factor a number which one could not, even in principle, factor on a classical computer. Of course, this, too, depends on it being possible to actually build a quantum computer!

In general, then, I would argue that the field of quantum information theory is here to stay, in one form or another. However, since I am an experimental physicist, I shall focus, in this Chapter, on implementing a “trapped-ion quantum computer” along

the lines proposed by Cirac and Zoller [40]. However, I shall first briefly describe the basics of quantum computation.

## 7.1 Quantum Computation

A quantum computer [35, 43, 160], like a classical one, may be considered to consist of a register of  $N$  information-carrying entities which, in correspondence to the classical case, are referred to as “qubits.” Each qubit is a two level system, with basis states  $|0\rangle$  and  $|1\rangle$ , representing logic “0” and logic “1.” However, since the qubits are quantum systems, they may exist not only in one or the other logic state, but also in *superpositions* of the form  $\alpha|0\rangle + \beta|1\rangle$  (with  $|\alpha|^2 + |\beta|^2 = 1$ ). The real power of quantum computation lies in this superposition property of qubits.

For, given an input register of  $N$  qubits, we may prepare the register in a superposition of *all* the  $2^n$  possible inputs. Furthermore, it is possible to produce this superposition (of an *exponential* number  $2^N$  of states) in a *linear* number of steps. For example, if our qubits are spins, with  $|\downarrow\rangle \equiv |0\rangle$  and  $|\uparrow\rangle \equiv |1\rangle$ , and the quantum register starts out in the state  $\prod_{n=0}^N |0\rangle$ , driving  $\pi/2$ -pulse on all the spins produces the state  $\prod_{n=0}^N (1/\sqrt{2})(|0\rangle + |1\rangle) \equiv \sum_{r=0}^N |r\rangle$ . In the last term, I have made the symbolic equivalence between a binary number  $r$  and the register state in which the individual qubits are in the appropriate state for the binary representation of  $r$ . Thus, the number  $r = 0010$  is represented by the four-qubit register state  $|0010\rangle$ . Since it is possible to efficiently initialize the quantum register in a superposition of all its possible inputs then, provided that our computation is made up of unitary operations, it is possible to process *in parallel and at once* all the possible outputs of the computation.

Of course, any attempt to read out the state of the quantum computer (or perform any type of non-unitary time evolution) will collapse it into one and only one of the

possible output states.<sup>1</sup> So, as it stands, although a quantum computer may be massively parallel, we have no way to access and utilize this parallelism. However, we may use another fundamental quantum property to overcome this obstacle: interference. Interference between different “computational paths” can enable us to distill information about global properties of computed functions (such as periodicity) [39, 161] while still preserving superposition until the very end. And such algorithms may be exponentially faster to execute than any known classical algorithm for solving the same problem.

To reiterate, although the exact mechanism by which a quantum computer may be faster than a classical computer is still not fully understood [162], it appears that this mechanism involves the interplay between “superposition” (in an exponentially large Hilbert space) and “interference” (which maps joint, or entangled, properties onto local ones). To obtain some idea of how this works in practice, consider one particular algorithm which allows a quantum computer to efficiently solve a classically hard problem: “Simon’s problem” [163]. Although this is somewhat of a “toy” problem, it is an example of the exponential speedup possible with quantum computers, and one which illustrates the ideas of quantum computation clearly.<sup>2</sup>

In Simon’s problem, we are given an unknown function  $f$  of  $N$  bits,  $f : \{0, 1\}^N \rightarrow \{0, 1\}^N$  (i.e.  $f$  maps  $N$ -bit numbers to  $N$ -bit numbers). We are told that  $f$  is 2-to-1. We are also told that  $f$  has the property that  $f(x) = f(y)$  iff  $y = x \oplus a$ , where  $\oplus$  is the bitwise Exclusive-OR operation (addition modulo 2). We are then asked to determine  $a$ . Classically, the best that we can do is to evaluate  $f$  for various  $x, y, \dots$  until we find  $x, y$  such that  $x \oplus y = a$ . If, for example, we evaluate  $f$   $2^{N/4}$  times, then the probability that  $x \oplus y = a$  is  $2^{-N}$  for any given  $x, y$ , and the number of pairs of function evaluations

---

<sup>1</sup> Here, the phrase “read out the state of the quantum computer” means an attempt to determine the computer’s logical state: that is, it refers to a qubit-by-qubit measurement in the “logical basis,”  $\{|0\rangle, |1\rangle\}$

<sup>2</sup> I shall follow the treatment given by Preskill [43].

is  $(2^{N/4})^2$ , so that the probability of success (i.e. of finding  $a$ ) satisfies:

$$P_{suc} < 2^{-N} (2^{N/4})^2 = 2^{-N/2}. \quad (7.1)$$

Thus, the problem is classically “hard,” as the probability of finding  $a$  is exponentially small as a function of  $N$ .

However, there exists an efficient quantum algorithm for finding  $a$ . Suppose that we have two,  $N$ -qubit quantum registers. We assume that the two registers start out in the state  $|0\dots 0\rangle|0\dots 0\rangle$  (i.e. all the qubits are in the state  $|0\rangle$ ). We then apply a unitary transformation  $\hat{\mathcal{H}}$  called the Hadamard transform to the first  $N$ -qubit register. I will discuss this unitary transformation shortly and, in particular, whether we can realize it in an efficient and scalable manner. However, for now, it suffices to assume that we *can* do so, and to specify it’s matrix representation with respect to the standard, “computation” basis,  $\{|0\rangle, |1\rangle\}$ :

$$\hat{\mathcal{H}} \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (7.2)$$

Let  $\hat{\mathcal{H}}^{(N)}$  represent the effect of  $N$  such Hadamard transformations acting in parallel on the  $N$  qubits. By applying Eq. (7.2) repeatedly, we can work out that

$$\hat{\mathcal{H}}^{(N)}|x\rangle = \sum_{y=0}^{2^N-1} (-1)^{x \cdot y} |y\rangle \quad (7.3)$$

where  $x \cdot y = x_0y_0 + x_1y_1 + \dots + x_Ny_N$  (i.e. if we treat the binary numbers in the binary expansion of  $x$  and  $y$  as components of a vector, then  $\cdot$  is the “dot product” of  $x$  and  $y$ ). Applying  $\hat{\mathcal{H}}^{(N)}$  to the  $N$ -qubit register leaves the quantum computer in the state:

$$|\mathcal{R}\rangle_0 = \frac{1}{2^{N/2}} \left( \sum_{x=0}^{2^N-1} |x\rangle \right) |0\dots 0\rangle, \quad (7.4)$$

Here,  $|x\rangle$  represents the state in which each qubit is in the state ( $|0\rangle$  or  $|1\rangle$ ) corresponding to the appropriate digit in the binary representation of the number  $x$ . For example, if  $x = 5 = (101)_2$ , then a register of three qubits representing this number would be in the state  $|1\rangle|0\rangle|1\rangle$ .

Starting in the state  $|\mathcal{R}\rangle_0$ , we apply the function  $f$  (which is given to us) to the second  $N$ -qubit register. Due to the superposition property of quantum mechanics, if we apply  $f$  in a unitary manner, the computer is left in the state:

$$|\mathcal{R}'\rangle = \sum_{x=0}^{2^N-1} |x\rangle |f(x)\rangle. \quad (7.5)$$

Here, we see that the superposition property allows us to compute the function  $f$  in a highly parallel manner: we compute all of its possible values in *one* computation.

Now, consider what happens if we measure the state of the second register. We collapse the state of this register into one of the  $2^{N-1}$  possible values of  $f(x)$ , say  $f(x_0)$ . But, since the states of the two registers are entangled, this also collapses the state of the first register into a superposition of the two possible states corresponding to  $f(x_0)$ : namely  $x_0$  and  $x_0 \oplus a$ . That is, the registers are left in the state

$$|\mathcal{R}''\rangle = \frac{1}{\sqrt{2}} (|x_0\rangle + |x_0 \oplus a\rangle) |f(x_0)\rangle. \quad (7.6)$$

Here, the collapse of the wave function in the second register has, through the entanglement of the two registers, produced a state in the first register which expresses information on the “periodicity” of the function  $f$ . At this point, since the second register factors out of the problem, we may ignore it and concentrate only on the first register.<sup>3</sup>

The problem now is to extract the information on  $a$  from the first register. Of course, we can't just measure the state of the register: this would give either  $x_0$  or  $x_0 \oplus a$ , but absolutely no information about  $a$ ! Instead, we may apply  $\hat{\mathcal{H}}^{(N)}$  to the register again. This produces the state:

$$\begin{aligned} \hat{\mathcal{H}}^{(N)} \left[ \frac{1}{\sqrt{2}} (|x_0\rangle + |x_0 \oplus a\rangle) \right] &= \frac{1}{2^{(N-1)/2}} \sum_{y=0}^{2^N-1} \left[ (-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus a) \cdot y} \right] |y\rangle \\ &= \frac{1}{2^{(N-1)/2}} \sum_{a \cdot y=0} (-1)^{x_0 \cdot y} |y\rangle, \end{aligned} \quad (7.7)$$

---

<sup>3</sup> In fact, it is possible to execute the algorithm without ever measuring the state of the second register! However, the ideas behind the algorithm are somewhat more clear if this step is included

In the last line of Eq. (7.7), I have collapsed the sum according to the following observation: if  $a \cdot y = 1$ , then the two terms in the coefficient of  $|y\rangle$  interfere destructively, so that only terms for which  $a \cdot y = 0$  remain. Thus, we have used destructive interference to extract information about  $a$  from the register, since when we measure the state of the register, we obtain a  $y$  for which  $a \cdot y = 0$ .

In order to determine  $a$ , we run the algorithm  $O(N)$  times, until we have found  $N$ , linearly independent values  $y_i$  for which  $a \cdot y_i = 0$ . We then solve the resulting set of linear algebraic equations to obtain  $a$ . This repetition of the algorithm  $O(N)$  times only adds linearly to the complexity of the solution. Thus, we may solve for  $a$  efficiently on our quantum computer, whereas it is not possible to do so on any classical computer.

To “flesh out” this example a bit, consider the somewhat trivial case in which  $N = 2$ . Suppose that the function  $f$  is defined by:

$$\begin{aligned} f(00) &= 00 \\ f(01) &= 01 \\ f(10) &= 00 \\ f(11) &= 01 \end{aligned} \tag{7.8}$$

(so that  $a$ , which we are to determine, is equal to 10). In order to determine  $a$ , we start out with a quantum computer consisting of two, two-qubit registers in the state

$$|\mathcal{R}\rangle_0 = |00\rangle_1 |00\rangle_2 . \tag{7.9}$$

The subscripts label registers 1 and 2. We initialize the computation by applying the Hadamard transformation to register 1, to put it in a superposition of all its possible values:

$$|\mathcal{R}\rangle_0 \longrightarrow \frac{1}{2} \left( |00\rangle + |01\rangle + |10\rangle + |11\rangle \right)_1 |00\rangle_2 . \tag{7.10}$$

Next, we apply  $f$  to the second register, taking the first register's values as  $f$ 's input.

This leaves the registers in the state

$$|\mathcal{R}'\rangle = \frac{1}{2}(|00\rangle_1|00\rangle_2 + |01\rangle_1|01\rangle_2 + |10\rangle_1|00\rangle_2 + |11\rangle_1|01\rangle_2), \quad (7.11)$$

so that now the states of the two registers are entangled.

A measurement of register 2 collapses the state into one of the possible measurement outcomes — for example, into the state:

$$|\mathcal{R}''\rangle = \frac{1}{\sqrt{2}}(|01\rangle_1 + |11\rangle_1)|01\rangle_2. \quad (7.12)$$

Now, we discard the second register, and apply the Hadamard transform to register 1, which results in

$$\begin{aligned} \hat{\mathcal{H}}^{(2)} \frac{1}{\sqrt{2}}(|01\rangle_1 + |11\rangle_1) &= \frac{1}{2\sqrt{2}}(|00\rangle_1 - |01\rangle_1 + |10\rangle_1 - |11\rangle_1 \\ &\quad + |00\rangle_1 - |01\rangle_1 - |10\rangle_1 + |11\rangle_1) \\ &= \frac{1}{\sqrt{2}}(|00\rangle_1 - |01\rangle_1). \end{aligned} \quad (7.13)$$

Note that the two values,  $x_0 = 00$  and  $x_0 = 01$ , both satisfy  $x_0 \cdot 10 = 00$ . Finally, we measure the state of register 1, obtaining *either* the result  $x_0 = 00$  or  $x_0 = 01$ .

In order to determine  $a$ , we repeat the above procedure until we obtain the *other* value<sup>4</sup> of  $x_0$ . Then, we solve the system of equations

$$\begin{aligned} 00 \cdot a &= 00 \\ 01 \cdot a &= 00 \end{aligned} \quad (7.14)$$

to obtain  $a = 10$ .

In the above example, we see some of the fundamental properties of quantum computers: the parallel computation of functions allowed through the superposition

---

<sup>4</sup>  $x_0$  is either equal to 00 or to 01 in this example, as we may check by running through the example again, this time assuming that the measurement of register 2 produces the outcome  $|00\rangle_2$  instead of the state  $|01\rangle_2$  assumed above

principle, the entanglement of two quantum registers which, through the collapse of the wave function, selects out certain properties of the function, and the use of destructive interference to “distill” this information into a form which can be effectively read out. In essence, the challenge in finding quantum algorithms is to achieve this last point; that is, to overcome the impediment of wave function collapse by using quantum interference in a clever way.

As yet, there are few concrete examples of such algorithms. Two of the most significant are those due to Shor [39] and to Grover [164] (for a pedagogical discussion of these algorithms, see Ref. [43]). Shor’s algorithm uses quantum interference to find the period of the function  $a^b(\text{mod}M)$  (where  $M$  is given and  $a$  is a randomly-chosen number co-prime with  $M$ ) and uses this information to determine the factors of  $M$ . The resources required scale as a polynomial of the size of  $M$ , whereas all known classical algorithms scale exponentially<sup>5</sup> with  $M$ . As mentioned above, factoring large numbers is of great interest to the data-encryption community.

Grover’s algorithm searches a database of qubits for a particular, marked entry. It is faster than any classical algorithm. However, the classical algorithms themselves scale as polynomials of the database size, and so Grover’s algorithm does not change the “complexity class” of the problem.

I have not yet discussed whether it is possible to implement the Hadamard transformation (or the function  $f$ ) efficiently on a given quantum computer. However, it may be shown that any unitary transformation on a set of qubits may be modelled to any desired accuracy by a fundamental set of two-qubit and one-qubit gates [165, 166]. Furthermore, the number of such gates required scales as a polynomial in the desired accuracy. This means that one may use any given quantum computer (with any given set of basic gates) to simulate any other quantum computer without changing the efficiency of the algorithms implemented on the two computers (i.e. from “easy” to “hard”).

---

<sup>5</sup> Note, however, that there is as yet no proof that an efficient classical algorithm does not exist.

Thus, one can construct a “universal quantum computer” in the sense of Turing [152]. Thus, given *some* set of universal quantum logic gates, we can efficiently implement both  $f$  and the Hadamard transformation. So, as the above example (Simon’s problem) shows, quantum computers are inherently more powerful than classical ones.

One family of universal quantum logic gates consists of one-qubit rotations and a two-qubit logic gate: the Controlled-NOT.<sup>6</sup> The effects of the Controlled-NOT quantum logic gate are prescribed by the effects of the gate on the qubit basis states; the effects of the gate in cases in which the qubits are in superposition states follows by linearity. Thus, the Controlled-NOT gate is realized by the transformation

$$|\epsilon_1\rangle|\epsilon_2\rangle \rightarrow |\epsilon_1\rangle|\epsilon_1 \oplus \epsilon_2\rangle, \quad (7.15)$$

where  $\epsilon_1, \epsilon_2$  are arbitrary elements of the two-dimensional Hilbert space and  $\oplus$  represents addition modulo 2 (or, if you prefer, the classical Exclusive-OR Boolean logic operation). Qubit 1 is referred to as the “control qubit” and qubit 2 is referred to as the “target qubit.” With the Controlled-NOT gate and single-qubit operations, we may implement any unitary transformation to the required degree of accuracy and thus perform universal quantum logic.

## 7.2 Errors and Error Correction

Since quantum computation relies on superposition states, a quantum computer is extremely sensitive to decoherence, which may be viewed as a measurement performed by uncontrolled and inaccessible degrees of freedom of the environment [15, 167]. Thus, decoherence collapses the superposition of qubits and so destroys the massive parallelism which is at the heart of quantum computation’s power. If we are to build a quantum computer, then, we must find a system where the detrimental effects of decoherence are limited.

---

<sup>6</sup> It is interesting to note that, whereas *reversible* classical logic requires at minimum a three-bit gate along with single bit operations to form a complete logic family, quantum logic requires only one- and two-qubit gates

When the construction of quantum computers was first suggested, it was argued that *no* system could be so well-isolated from the environment so as to perform the number of computational steps required for “useful” quantum computations, without suffering almost complete loss of coherence [167, 168, 169]. In particular, it was noted that [169], in atomic systems (such as trapped ions), fundamental physics would prevent performing coherent gates with an imprecision of less than  $10^{-5}$  (in fact, these arguments neglected the possibility of Raman transition, which change the limiting imprecision by two orders of magnitude — however, a fundamental limit to the precise manipulation remains, albeit however small). Aside from questions as to the length of “useful” computations, these arguments did not account for the discovery that it is possible to *correct* errors in quantum computations.

The discovery of error correction [170, 171] was a great surprise, and one of the most significant physics insights to arise from quantum information theory thus far. It is surprising because, in order to correct errors, we have to diagnose them, and this requires performing a measurement upon the quantum computer. But the effects of a measurement are to collapse quantum superpositions — which is exactly what we were trying to avoid in the first place! Furthermore, the “quantum no-cloning theorem” [43, 172, 173] precludes us from copying the state of the quantum computer onto another set of qubits, which could be measured with impunity. Nonetheless, quantum error correction *is* possible. I will briefly describe the ideas behind it here, but the reader is referred to the excellent recent article in *Physics Today* by John Preskill [174] for a more comprehensive treatment.

In fact, the ideas behind error correction in quantum computers are very similar to those behind classical error correction, with some caveats to respect the laws of quantum mechanics. So first let us consider how classical error correction works. If we have some collection of classical bits (each of which can be either in the state 0 or 1), then classical errors correspond to bit flips. In order to protect against such errors,

we may redundantly encode each bit in several: for example, we may use extra bits to store the original logical bit “0” as “000.” Then, we can protect against errors in which a single physical bit flips (i.e. “000” $\rightarrow$ “010”) by looking at the three bits, performing a majority vote, and resetting the “disagreeing” bit to the value of the other two. Of course, this error correction scheme does not protect against errors in which *two* physical bits flip, but in practice these errors are much more rare (typically, if the probability of a single bit flip is  $\varepsilon$ , then the probability of two bits flipping is  $\varepsilon^2$ ).

With qubits, the situation is more complicated. First, we can’t observe the qubit without destroying the entanglement and superposition which are necessary for quantum computation’s power. Second, we can’t simply copy the state onto extra qubits, as we could with the classical bits. Third, there are more types of errors with qubits than with classical bits. For example, given a qubit in the general state  $\alpha|0\rangle + \beta|1\rangle$ , we may certainly experience “bit flip” errors  $\alpha \leftrightarrow \beta$ . But we may also experience “phase” errors such as  $\beta \rightarrow -\beta$ . And, indeed, since we are dealing with a quantum system, we may experience a continuum of either sort of error: from no error to the “full” errors listed above. However, by being clever, we can overcome all these issues.

As an example, suppose we only want to protect against amplitude errors (i.e. the continuum of errors from no error to a complete bit flip). We may circumvent the “quantum no-cloning theorem” by redundantly encoding the information stored in a qubit without actually copying it. So, for example, if we want to protect a qubit in the state  $\alpha|0\rangle + \beta|1\rangle$  from amplitude errors, we may (using a sequence of Controlled-NOT gates) produce the state  $\alpha|000\rangle + \beta|111\rangle$ , which is *not* a “clone” of the original state (*that* would be the state  $(\alpha|0\rangle + \beta|1\rangle)^3$ ). Now, suppose a single error occurs, and the encoded state becomes  $\alpha|000\rangle + \beta|111\rangle + \varepsilon(\alpha|100\rangle + \beta|011\rangle)$ . If we were to measure the complete state of the encoded qubit, then we would destroy the superposition. However, we can still extract information about the error by making a *partial* measurement of the system; that is, we may measure collective properties of the three-qubit encoded state.

So, in order to determine whether an error has occurred (and in which qubit), we may make a measurement to determine whether qubits 1 and 2, 2 and 3, and/or 3 and 1 have the same value. Note that we do *not* determine the values of any of the qubits, but only ask whether they are the same or different. Now, assume  $\varepsilon \ll 1$  (this isn't necessary, but reflects the probable situation in a practical quantum computer). Since the error occurred in the first qubit, we will always measure that qubits 2 and 3 are in the same state. With probability  $1 - \varepsilon^2$ , our joint measurement of qubits 1 and 2 will project qubit 1 back into the state where it is identical to qubit 2 (the quantum Zeno effect [59]). On the other hand, with probability  $\varepsilon^2$ , the measurement will project qubit 1 into a state where it *definitely* is in the *opposite* logical state to qubit 2. So in this case, our joint measurement has turned a very small error into a very large one (the largest possible amplitude error!). However, this is “not a bug, it’s a feature.” Since we now know that qubit 1 is opposite to qubits 2 and 3 (but we still don't know the state of any of the three qubits), we can apply a unitary transformation to flip the logical state of the first qubit, restoring the three-qubit system to the state  $\alpha|0\rangle + \beta|1\rangle$  we started out in, without ever knowing what exactly that state was.

So, the idea behind quantum error correction is to redundantly encode the logical information of a single qubit into a system of several qubits. By making measurements of collective properties of this system, we may make “enough” of a measurement to determine that an error occurred, and how to correct it. However, we may be careful not to remove so much information from the quantum system to destroy the superposition and entanglement necessary for quantum computation. One way of looking at the situation is to view the extra qubits as an “entropy dump” which holds the extra entropy put into the quantum system by the measurements made.

The above example treated only amplitude errors. However, one may also simultaneously protect against phase errors. One way to do this [170, 174] is to further encode each three-qubit block (which protects against amplitude errors) into three, three-qubit

blocks. By comparing the relative phases of  $|000\rangle$  and  $|111\rangle$  among the three “super-blocks,” we can determine whether a phase error occurred, and in which block. We then can correct the error much like before.

Thus far, I have discussed ways to correct errors in qubits caused by interactions of the qubits with the environment. However, if the logic gates themselves are imperfect, we might imagine that this would make the situation hopeless — in trying to correct the errors, we would add *more* errors. So the question arises as to whether it is possible to come up with error correction schemes which also correct for logic gate imperfections: so-called “fault-tolerant” error correction schemes. The short answer is “yes” [170, 175, 176, 177, 178]. By using nested blocks of qubits, it is possible to perform completely faithful quantum computation even in the presence of environmentally induced decoherence and gate errors, given some minimum fidelity of the gates. This “opening bid” of fidelity lies somewhere between  $10^{-4}$  and  $10^{-6}$ , depending on the particular decoherence mechanisms which are most significant. The price to be paid is an increase in the total number of qubits and gates required to perform quantum computations (but this increase in resources grows only logarithmically with the number of steps in the algorithm). There is some possibility that error correction schemes may exist which require only a fidelity of  $10^{-2}$  for fault tolerant correction [179], using “non-concatenated” error correction codes, but ways to perform quantum computation using such codes have not yet been found.

### 7.3 Ion Trap Quantum Computers

With the advent of error correction, the question becomes not “can we find a system with so little decoherence as to allow long quantum computations?” but rather “can we find a system where the decoherence is below the threshold required for fault-tolerant error correction?” However, said system must also allow for strong interactions between different qubits so that logic gates may be implemented. Given the long spin-

coherence times observed in trapped-ion systems [180] and the success which we have had in manipulating the spin and motional degrees of freedom of trapped ions (see Ch. 6), we believe that trapped ions offer some promise, at least as far as building a rudimentary quantum information processor.

In particular, we are trying to implement a simple quantum computer of several qubits, according to the scheme proposed by Cirac and Zoller [40], and examine the technological feasibility of building even larger quantum computers along the same lines. In the Cirac-Zoller scheme, the qubits used to store information consist of two electronic levels of a collection of trapped ions, one qubit per ion. Quantum information is exchanged between qubits through two vibrational levels of one mode of the ions' quantized collective motion in the ion trap. The coupling is provided by focused laser beams that can individually address each of the ions.

In a slightly modified version of their original proposal, we can imagine a string of ions trapped in a linear trap. In our case, we may imagine that the motional modes have been laser cooled to low temperature and that one (say, the stretch mode) has been laser cooled to its ground state. In order to perform a Controlled-NOT gate between ions  $j$  and  $k$ , we illuminate ion  $j$  with a laser beam (or laser beams, for stimulated Raman transitions) on the stretch red sideband. If the ion is in  $|\downarrow\rangle \equiv |0\rangle$ , then this has no effect, and the stretch mode remains in the state  $|0\rangle$ . On the other hand, if ion  $j$  is in the state  $|\uparrow\rangle \equiv |1\rangle$ , and the interaction is left on long enough to drive a  $\pi$ -pulse, then the motional mode is left in the state  $|1\rangle$ , while ion  $j$  is left in  $|\downarrow\rangle$ . This interaction has mapped the spin state of ion  $j$  onto the motional state of the entire string, which is a joint property of all the ions. In this sense, the motional mode plays the role of a “quantum data bus,” which makes the quantum information stored in one ion's spin available to all the other ions.

With the information from ion  $j$  transferred to the motional “data bus,” it remains to implement a Controlled-NOT between the data bus and ion  $k$ . In order to do this,

we may use a third atomic level of  $k$  to put a motion- dependent phase onto the spin state of ion  $k$ , as will be described below, in Sec. 7.4. By sandwiching this transition between two  $\pi/2$ -pulses on the  $|\downarrow\rangle \leftrightarrow |\uparrow\rangle$  system, we can realize the desired interaction. Finally, we can restore the spin state of the original ion  $j$  and the motional mode by applying a final  $\pi$ -pulse on the red sideband.

Of course, the question arises as to whether it will be possible to reach the required levels of accuracy for large-scale quantum computations. We have treated many of the possible barriers (laser intensity and frequency fluctuations, magnetic field fluctuations, background electric fields, motional heating, trap imperfections, etc.) in Refs. [73, 111]; see also Refs. [36, 181]. The reader is referred to these articles for such discussion. In the end, however, there seems to be no fundamental barrier to ion-trap quantum information processors — as to the technical difficulties which must be overcome, that seems to be a question best answered by experiment.

## 7.4 Quantum Logic on One Ion

We have demonstrated a two-qubit “quantum Controlled-NOT” logic gate between the spin and motional degrees of freedom of a single trapped ion [41] (this is the important step in the general quantum computation scheme described immediately above). The qubits in our realization consisted of (i) the two-dimensional subspace of the motional Hilbert space spanned by the basis states  $|n = 0\rangle$  and  $|n = 1\rangle$ , and (ii) the two-dimensional Hilbert space spanned by the basis states  $|\downarrow\rangle$  and  $|\uparrow\rangle$ . If the motional qubit is the control and the spin the target, then the effect of the Controlled-NOT is to flip the spin if and only if the motion is in the state  $|n = 1\rangle$ .

In order to realize the Controlled-NOT, we applied the following three Raman beam pulses:

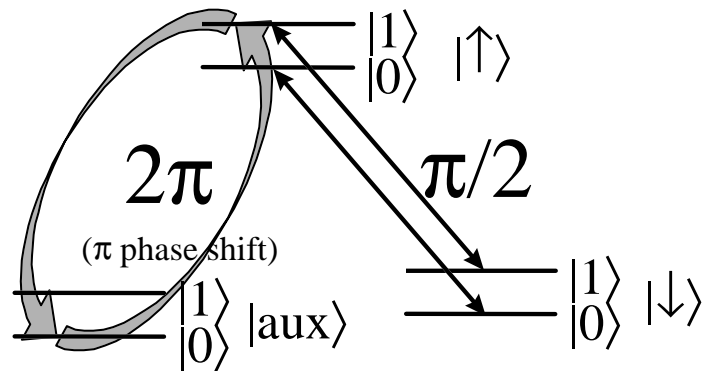
- (1) A  $\frac{\pi}{2}$  pulse on the carrier transition. By way of example,  $|\downarrow\rangle \rightarrow \frac{1}{\sqrt{2}}(|\downarrow\rangle + |\uparrow\rangle)$ .

- (2) A  $2\pi$  pulse on the first upper sideband transition between levels  $|\uparrow\rangle$  and an auxiliary level: the  $2s\ ^2S_{1/2}|F=2, m_F=0\rangle \equiv |aux\rangle$ . This transition couples  $|\uparrow, n\rangle$  and  $|aux, n-1\rangle$ . Due to the formal equivalence between two-level systems and spin-1/2 particles [2], this  $2\pi$  rotation in the two-dimensional Hilbert space spanned by  $|\uparrow, 1\rangle$  and  $|aux, 0\rangle$  changes the sign of the state vector component in  $|\uparrow, 1\rangle$ . However, since there is no lower motional state to which  $|n=0\rangle$  can couple, the pulse does not affect the component of the state vector in  $|\uparrow, n=0\rangle$ . Any component of the state vector in  $|\downarrow\rangle$  is also unaffected, due to the Zeeman shift between levels  $|aux\rangle$  and  $|\downarrow\rangle$ .
- (3) Another  $\frac{\pi}{2}$  pulse is applied on the carrier, but with a  $180^\circ$  phase shift relative to the first pulse. If there was no component of the state vector in  $|n=1\rangle$ , then this simply reverses the effects of the first  $\frac{\pi}{2}$  pulse. However, due to the minus sign introduced by Step 2, the transition started by the first pulse is *completed* by the second for any component of the state vector which was in  $|n=1\rangle$ .

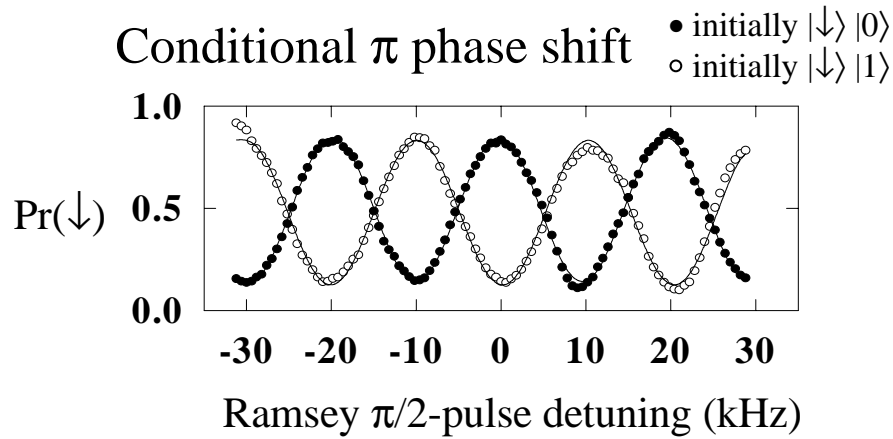
The pulse sequence is illustrated schematically in Fig. 7.1(a). The overall effect is the desired one: the spin of any component of the state vector in the motional state  $|n=1\rangle$  is flipped, while the motion is unaffected.

The effect of the Controlled-NOT gate on the computational basis states is shown in Table 7.1, which lists the probabilities for the various basis states before and after the gate. Note that a major limitation on the gate's apparent fidelity was our ability to accurately prepare the various basis states. This was primarily due to technical sources of noise (laser beam intensity fluctuations, etc.).

The conditional dynamics which are the heart of the Controlled-NOT gate are illustrated in Fig. 7.1(b). This figure shows a Ramsey spectrum in which the two  $\pi/2$ -pulses of the Controlled-NOT gate (steps 1 and 3, above) form the Ramsey zones, and the conditional dynamics occurs during what would normally be the "dark" period.



(a)



(b)

Figure 7.1: (a) Schematic representation of the Controlled-NOT gate. The crucial part of the gate is the  $2\pi$  pulse between  $|\uparrow\rangle$  and  $|\text{aux}\rangle$  on the upper motional sideband of this transition. The pulse puts a phase factor of  $-1$  in front of only the part of the motional wave function that is in  $|\uparrow, n=1\rangle$ . This produces the conditional dynamics necessary for the quantum logic gate. (b) Ramsey spectra of the Controlled-NOT (CN) gate. The detuning of the Ramsey  $\pi/2$  pulses of the gate is swept, and  $P_{\downarrow}$  is measured. The filled circles correspond to initial preparation in the  $|\downarrow, n=0\rangle$  state and the open circles to preparation in the  $|\downarrow, n=1\rangle$  state. The resulting patterns are shifted in phase by  $\pi$  radians, indicating the conditional dynamics of the gate. Similar curves result for preparation in the other two basis states.

Table 7.1: Effect of the Controlled-NOT three-pulse sequence  $\pi/2(\text{carrier}), 2\pi(\text{aux,RSb}), -\pi/2(\text{carrier})$  on the computational basis states. The initial state probabilities differ from 1 and 0 due to state preparation imperfections. The further decline in state probabilities in the “Final State” column is due to imperfections in the gate operations. Nonetheless, with high probability, the Controlled-NOT gate preserves the value of the control qubit (the motional state  $|n\rangle$ ), and flips the target qubit (the spin) if and only if  $n = 1$ .

Initial State		Final State	
P( $n = 1$ )	P( $\uparrow$ )	P( $n = 1$ )	P( $\uparrow$ )
0.02	0.01	0.09	0.16
0.03	0.99	0.04	0.88
0.92	0.05	0.77	0.88
0.94	0.98	0.88	0.19

The solid circles show the Ramsey fringes for the case when the ion started out in  $|\downarrow\rangle$  and in the motional state  $|n = 0\rangle$ : in this case, step 2 had no effect. The hollow circles show the fringes when the ion started out in the state  $|\uparrow, n = 1\rangle$ : in this case, the conditional dynamics were in effect, and caused the  $\pi$  radians phase shift with respect to the previous case. Similar fringes resulted when the ion was prepared in the other two computational basis states.

#### 7.4.1 Simplified Controlled-NOT Gate

As pointed out in Ref. [182], it is not necessary to use a third atomic level to realize a Controlled-NOT between the spin and motional degrees of freedom. Instead, we may use the nonlinear corrections to the Rabi frequencies discussed in Sec. 3.2. To do this, we set the Lamb-Dicke parameter  $\eta$  so that the  $n$ -dependent carrier frequencies (on the  $90^\circ$  carrier) result in a  $2\pi$ -pulse for one motional level when a  $\pi$ -pulse is driven on the other. For example, we have from Eq. (3.12) that

$$\Omega_{0,0} = \Omega e^{-\eta^2/2} \quad (7.16)$$

but that

$$\Omega_{1,1} = \Omega e^{-\eta^2/2} (1 - \eta^2). \quad (7.17)$$

If we choose  $\eta = 0.707$ , then  $\Omega_{0,0} = 2\Omega_{1,1}$  and so the carrier flips the ion's spin if its motion is in the state  $|1\rangle$  but only adds a trivial phase factor if the motional state is  $|0\rangle$ . This realizes an effective Controlled-NOT gate (up to phase factors) without the need for an auxiliary level. On the other hand, specific operating conditions ( $\eta$ ) are required for the trap. Other schemes have been proposed [183, 184] which do not require specific values of  $\eta$  but which do require more laser beams and/or higher-order transitions.

## 7.5 Quantum Logic on Multiple Ions

As mentioned in Sec. 6.4, the next logical step in trying to reach a quantum computer is to implement a gate between two ions. But, for the reasons discussed in that section, we have not yet done this. However, we expect that a two-ion gate will be demonstrated in the near future, most likely in a new version of the micromachined linear trap.

It is worth noting that, for two ions, the techniques used to create the Bell states (Sec. 6.4) can be used to implement a quantum logic gate [49]. To do this, we displace the two ions in an rf trap such that one ion experiences no micromotion ( $J_0(0) = 1 \neq 0$ ), whereas the other one satisfies  $\Omega J_0(|\delta k| \cdot x_{\mu 0, i}) = 0$  (see Eq. 6.34) and following discussion). In this configuration, laser-driven transitions near the carrier are only driven on the first ion. In order to allow transitions for the second ion, we may drive near the first upper or lower rf sideband. In this case, the zeroth-order Bessel functions are replaced by  $J_1(0) = 0$  and  $J_1(|\delta k| \cdot x_{\mu 0, i}) \neq 0$ , so that now the second ion interacts with the lasers while the first does not. However, it is not immediately clear how to apply a similar technique to more than two ions.

One step we have taken towards implementing multi-ion logic is demonstrating the preservation of spin coherence while moving ions back and forth in the linear trap using electric fields. Although the insensitivity of the spin degree of freedom to ion motion (in the absence of motional sideband interactions) is hardly surprising it is, nonetheless,

reassuring to observe this. Some implementations of quantum logic in ion traps [73] may require moving ions back and forth between “storage areas” and “accumulators,” in which quantum logic operations are performed between small numbers of qubits. In order to demonstrate the spin coherence, we performed a traditional Ramsey experiment (Sec. 3.4) on a single, trapped ion. Between the two Ramsey zones, we changed the voltage on the trap rods to move the ion along the  $z$ -direction some distance, and then back to its starting point. This process did not significantly degrade the contrast of the Ramsey fringes.

By shining a second, off-resonant laser beam into the trap at the position of the translated ion, we were able to map out this second laser beam’s profile, by measuring the Ramsey fringe shift due to the AC Stark effect. Fig. 7.2 shows the phase shift as a function of the displacement of the ion into the laser beam. This “laser beam” profiler indicates the approximately Gaussian beam profile at the trap. Note that the measurement occurs with negligible probability that the ion actually absorbs a photon from the second beam. The beam profile information is mapped onto the ion’s spin.

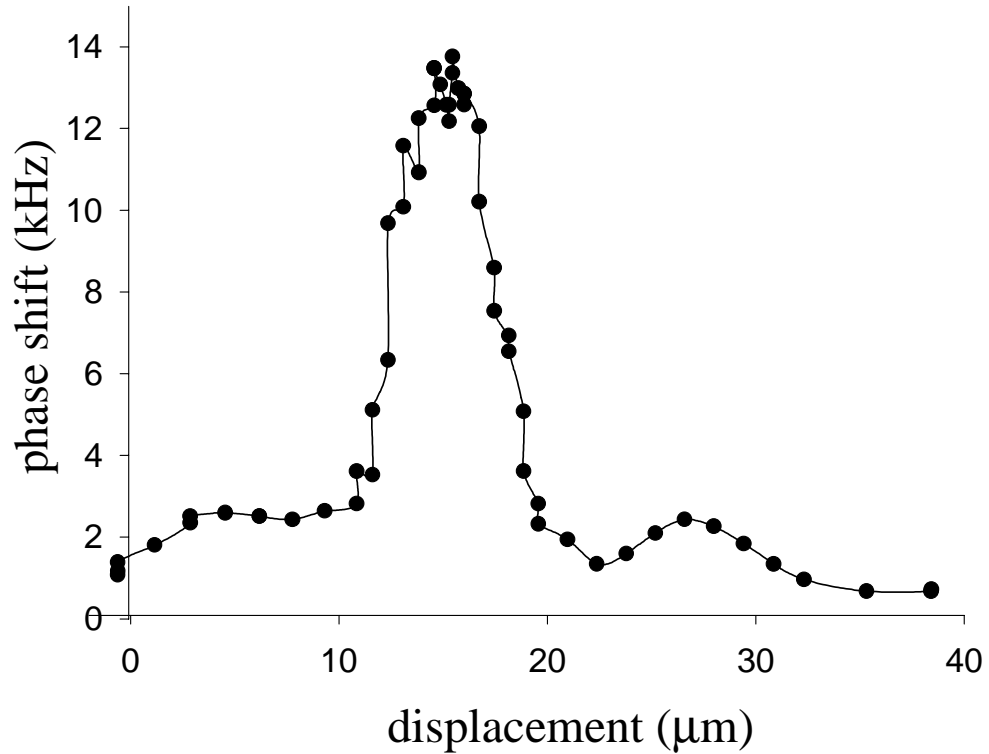


Figure 7.2: “Interaction-free” laser beam profiler. A traditional Ramsey experiment is performed on a single ion confined in the micromachined linear ion trap. In between the Ramsey zones, the trap voltages are changed so as to translate the ion into a second, nearby laser beam. By measuring the AC Stark shift phase shift of the Ramsey fringes, we can determine the second beam’s profile. Note that there is negligible probability that the ion absorbs a photon from the second beam.