Molecular Scale Heat Engines and Scalable Quantum Computation

Leonard J. Schulman* Umesh V. Vazirani†

Abstract

We describe a quantum mechanical heat engine. Like its classical counterpart introduced by Carnot, this engine carries out a reversible process in which an input of energy to the system results in a separation of cold and hot regions. The method begins with a reinterpretation in thermodynamic terms of a simple step introduced by von Neumann to extract fair coin flips from sequences of biased coin flips.

Some of the experimental set-ups proposed for implementation of quantum computers, begin with the quantum bits of the computer initially in a mixed state. Each qubit is ϵ polarized — in the state $|0\rangle$ with probability $\frac{1+\epsilon}{2}$, and in the state $|1\rangle$ with probability $\frac{1-\epsilon}{2}$, independently (or nearly so) of all other bits. The heat engine may be used to transform this initial collection of n qubits into a state in which a near-optimal $m = n[\frac{1+\epsilon}{2}\lg(1+\epsilon) + \frac{1-\epsilon}{2}\lg(1-\epsilon) - o(1)]$ qubits are in the joint state $|0^m\rangle$. These qubits can then be used as the registers for a quantum computation.

The heat engine is described at the level of an algorithm implementable in any quantum system capable of massive coherent states. A particular implementation is also described for a system of nuclear spins arranged in a chain. The temperature the cold qubits reach is inverse polynomial in n.

Introduction

A sequence of results over the last decade [11, 1, 24, 23] has provided the first credible challenge to the Extended (complexity theoretic) Church-Turing thesis. At issue is the ability of computers based on quantum physics to perform certain computations (such as factorization [23]) exponentially faster than classical computers. However, realizing quantum computation in the laboratory has proved to be a formidable challenge since it requires the isolation of the computer from the effects of environmentally induced decoherence, while being able to operate upon its state to perform elementary operations. In this paper, we concentrate on one aspect of this problem — namely the initialization of the state of the quantum computer. The task here is that in some experimental set-ups, the quantum bits in the computer are initially in a mixed state (where each qubit is ϵ polarized — in the state $|0\rangle$ with probability $\frac{1+\epsilon}{2}$, and $|1\rangle$ with probability $\frac{1-\epsilon}{2}$), independently of all other bits[1], whereas we would like the bits to be in the joint state $|0^n\rangle$. Since quantum mechanics is time reversible, any initialization algorithm is necessarily constrained to extract at most $(1 - H(\epsilon))n$ purified bits, where

*College of Computing, Georgia Inst. Technology, Atlanta GA 30332-0280. Email: schulman@cc.gatech.edu

†Computer Science Division, U. C. Berkeley, Berkeley CA 94720. Supported by JSEP grant FDF 49620-97-1-0220-03-98 and NSF grant CCR 9800024. Email: vazirani@cs.berkeley.edu.

[1] correlations among bits do not present a significant hinderance. See section titled 'Initial distributions with local correlations'.

$H(x) = \frac{1+x}{2} \lg \frac{2}{1+x} + \frac{1-x}{2} \lg \frac{2}{1-x}$ — i.e. the optimal performance is extraction of the state $\left|0^{(1-H(\epsilon))n}\right\rangle$. The challenge is to extract as close as possible to this number of purified bits, without access to any scratch bits, as efficiently as possible, and using only very simple elementary device operations.

In this paper, we give a procedure that extracts the asymptotically optimal fraction of purified bits. We give a quasi-linear time implementation of our procedure in a model motivated by NMR quantum computing. Recently, small scale quantum computation (on two to three qubits) (see e.g. [4, 9, 5]) has been successfully demonstrated in the laboratory, using conventional NMR systems. The qubits in question are the nuclear spins associated with the atoms of a molecule. A major difficulty is that conventional NMR systems operate upon a macroscopic sample, rather than a single molecule. At room temperature, and in thermal equilibrium, such a sample must be regarded as constituting a statistical mixture of pure states (rather than the state $|0^n\rangle$ in which we would like to start the computation). A major breakthrough in the use of NMR techniques in quantum computation came about in [15, 8], where a scheme was given for embedding a small dimensional 'virtual' pure state within the density matrix describing the bulk sample, by exploiting the structure present in thermal equilibrium. (Some of the experimental results cited are based on this scheme.) Although this approach provides a very important proof of concept demonstration for quantum computation, it does not scale — the strength of signal output by the NMR quantum computer degrades exponentially in the number of quantum bits n in the system. Thus the exponential speedup promised by quantum computation is offset by an exponential increase in the effort required to detect the output signal. The initialization procedure given in this paper yields $(1 - H(\epsilon) - o(1))n$ purified bits when it is applied to n bits of polarization ϵ (the output signal strength is now independent of n), and thus scales optimally as a function of n and ϵ. Furthermore, in the NMR model, our initialization algorithm takes quasi-linear time. For this we have to demonstrate how to implement a certain kind of cellular automaton, which also includes the capabilities of an oblivious 2-tape Turing Machine. The molecules needed for our NMR implementation are simply linear polymers.

We should point out that for these numbers to be practical, there will have to be a breakthrough in the achievable initial polarization in liquid NMR. This is not inconceivable, given advances in polarizations achieved in Zenon using optical pumping techniques. In any case the asymptotic optimality of the bit extraction rate of our algorithm, and the near-optimality of its runtime, imply that the only substantial remaining challenge is physical.

Our initialization scheme may be thought of as carrying out an approximate form of Schumacher compression [22], a fundamental primitive in quantum information theory. Indeed, our initialization scheme can be substituted in place of Schumacher's scheme in many applications. The advantage of our scheme is that it requires no scratch bits, and is considerably more efficient than the most efficient known implementation of Schumacher compression [7]. It would be fruitful to explore other applications of our scheme to quantum information theory — such as the purification of EPR pairs.

It is particularly informative to view our initialization scheme from a thermodynamic viewpoint. Indeed, the initialization scheme may be thought of as converting n warm bits into $(1 - H(\epsilon))n$ cold bits and $H(\epsilon)n$ hot bits. It is an efficient implementation of a discrete analog of the Carnot cycle — linear time for a positive change across the entire device, $O(n \log n)$ time to accomplish the full cooling/heating required for initialization of a quantum computer. Indeed, thermodynamic considerations play an important role in our analysis of the performance of our scheme. Although the algorithm consists of $O(\log n)$ iterations of a very simple transformation, the analysis showing that it achieves the desired purification is rather more involved. The analysis proceeds in two stages. The first is based on the total energy of the system (the canonical potential function!), and shows that at the end of $O(\log n)$ iterations of the transformation, the cold bits are logarithmically cold. For further iterations the energy analysis is no longer useful. Instead, in the second stage, keeping track of the inverse temperature allows us to show that in $O(\log n)$ further iterations of the transformation, the cold bits are polynomially cold. The role that thermodynamics plays in our analysis is particularly interesting, since the basic computational problem that we are solving does not necessarily call for obtaining a state describable in thermal terms (for example, one may as well ask for a preponderance of "excited" as of "ground" states).

Our work is related to a sequence of results on extracting fair coin flips, given a biased coin. This problem was first considered by von Neumann [20], who suggested taking a pair of biased coin flips, (a, b), and using the value of b *conditional* on $a \neq b$. We use the same basic step, which we have called the boosting step; our purpose, though, is opposite to that of von Neumann. We are interested, not in the fair coins, but in the highly determined ones. In a continuation of von Neumann's work, techniques for extracting asymptotically optimal quantities of fair coins were provided by Elias [13] and Peres [21]; the latter incorporated von Neumann's step into a recursive method, which forms the inspiration for our procedure, even though we have the opposite goal. The most important difference (beyond the obvious need in the present context for a simple physical implementation) is in the analysis, in which we demonstrate bounds on the quantity and temperature of the cold (highly determined) bits extracted.

In fact our thermodynamic viewpoint gives an interesting physical perspective on these previous investigations, whose goal was the extraction of fair, or in our terminology, infinite-temperature bits — given an initial assemblage of biased bits, which we have elected to interpret (and in the physical scenario, have every reason to interpret) as finite-temperature bits.

A Discrete Analog of the Carnot Cycle

Consider an n qubit system, where $|0\rangle$ and $|1\rangle$ are the energy eigenstates of each qubit, with corresponding energy levels E_0 and E_1 respectively. We assume that $E_0 < E_1$ and write $E_\Delta = (E_1 - E_0)/2$ and $E_\mu = (E_1 + E_0)/2$. In thermal equilibrium at temperature T, the system is in a Gibbs distribution. For the purposes of the procedure we describe, it is only necessary to specify the probability distribution resulting from measuring the system in the computational basis. (I.e. the basis that is the tensor product of the "spin up, spin down" pairs for each of the nuclear spins.) The result of this measurement is a Bernoulli distribution on n bits: the probability that a particular bit is $|0\rangle$ is $\frac{\exp(E_\Delta/kT)}{\exp(E_\Delta/kT)+\exp(-E_\Delta/kT)}$, and the probability it is $|1\rangle$ is $\frac{\exp(-E_\Delta/kT)}{\exp(E_\Delta/kT)+\exp(-E_\Delta/kT)}$. The bias or polarization of the distribution, i.e. the difference between these probabilities, is $\tanh(E_\Delta/kT)$, thus 0 at infinite temperature, and 1 at zero temperature. Thus if we start with bits of initial bias ϵ_0, then $x_0 = \frac{k}{E_\Delta}\tanh^{-1}\epsilon_0$ is the initial inverse temperature.

Fundamental to our method is the following "boosting" step:

Boosting: Given two bits (a, b), (use a controlled-not to) replace them by the pair $(a', b') = (a \oplus b, b)$.

Observe that a' records whether $a = b$. Suppose both a and b were in a Gibbs distribution with polarization ϵ. Then a' has polarization ϵ^2; we will refer to this as a "warm" bit. Conditional on $a' = 0$, b' has polarization $\frac{2\epsilon}{1+\epsilon^2}$; we will refer to this as a "cold" bit. Conditional on $a' = 1$, b' has polarization 0; we will refer to this as a "hot" bit.

Result

To describe our purification procedure, we begin with an "ideal" version. The analysis of this version, as well as the specification of the details of the implementation, are complex. If and when "medium-scale" (meaning, say, $10-50$ qubit) thermal quantum computers come to be built, every extracted "cold" bit will be crucial; in this case it may be worthwhile to carry out the computations required to analyze and optimize this ideal version of our method.

We go on, however, to present a second version of our method. This version can be implemented in time $O(n \log n)$ on a quantum computer with n qubits; how this can be done is described in the Architecture and Implementation section. This is close to optimal, since linear time is required simply to bring the extracted bits to one place on the molecule, ready for a subsequent computation. Furthermore, this version is amenable to analysis; and we will show that the number of cold bits it extracts is almost the maximum that is possible given the initial thermal polarization of the system. (The fraction of bits that could be but are not extracted, tends to 0 as a function of the size of the computer). In particular, this version is only slightly inferior to the ideal version of the method in terms of the number of bits extracted.

Purification Procedure A (ideal)

1. Pair up the bits, and apply the boosting step to each pair.
2. Arrange the b bits so that the cold bits appear first, followed by the hot bits. (Note: this can be done reversibly. If we wished to use the information about where the boundary is, we would need to record it. However, we do not do this; instead, we will use the fact that we can compute the expected polarization of each bit in the sequence.)
3. Discard bits of very low bias.
4. Repeat on the remaining bits.

Ideally we would like to discard the hot bits and repeat the boosting step on the cold and the warm bits. What makes this a little tricky is that in our method, the sequence of elementary operations performed is required to be independent of the actual state of quantum bits being operated on (since the computation steps are imposed by external rf pulses). Therefore, since the number of cold and hot bits resulting from a boosting step is a random variable, we must try to achieve the same effect obliviously. One way to do this is to sort the bits by their expected polarization (the expected polarization is easily precomputed by a classical Monte Carlo simulation; thus the actual quantum computation is oblivious and moreover the number of steps of quantum computation remains unchanged), and then continue with the boosting process. This would push the cold bits toward one end, and hot bits toward the other. Then after a suitable number of iterations (namely $O(\log n)$), we should end up with the right number of cold bits at one end.

For the purpose of analyzing the yield of the procedure, we have to avoid the correlations that result from the separation process (since that involves a separation of the b bits conditional on the value of the a' bits). Set $\beta = (1-\epsilon^2)/2$. The key observation is that even though the number of cold and hot bits resulting from a boosting step is a random variable, it is very sharply concentrated. Fixing any $1/2 < \gamma < 1$, the Chernoff bound guarantees that, in a block of s bits beginning with bias ϵ, with probability at least $1 - \exp(-\Omega(s^{2\gamma-1}))$ the number of cold bits is more than $\frac{1+\epsilon^2}{4}s - \beta^{1/2}s^{\gamma}$. So, if we work only with the first $\frac{1+\epsilon^2}{4}s - \beta^{1/2}s^{\gamma}$ of the cold bits, and discard the hot bits as well as the bits near the end of the cold sector, we can say that conditional on a rare event, of probability at most $\exp(-\Omega(s^{2\gamma-1}))$, not having occurred, the bits that we continue to work on are guaranteed to be independent cold bits; and we can apply a recursive analysis. Although it is possible by some sampling to extract slightly closer to the true number of cold bits, there are only low-order gains to be obtained in this way, and we wish to keep the computation oblivious and the analysis simple.

We therefore describe in detail a method which yields:

Theorem: The algorithm described below begins with the independent ϵ_0-biased distribution on n bits, with $n > \epsilon_0^{-2}$, and in $O(n \log n)$ time, with probability $1-\exp(-n^{\Theta(1)})$, extracts $n(1-H(\epsilon_0)-o(1))$ bits each of bias $1-\exp(-n^{\Theta(1)})$ (hence temperature $\frac{E_\Delta}{kn^{\Theta(1)}}$).

The information-theoretic upper bound on the number of cold bits which can be extracted from the original distribution is $n(1-H(\epsilon_0))$, (recall $H(x) = \frac{1+x}{2}\lg\frac{2}{1+x} + \frac{1-x}{2}\lg\frac{2}{1-x}$). For, the reversible and therefore entropy-preserving transformation is required to, with high probability, specify the entire original sequence of bits, using only the bits that are not extracted.

Purification Procedure B

This procedure will be executed in two phases. In each phase we will repeat the following "basic step" $O(\log n)$ times, in the manner to be described. We will make use of the constants $\gamma = 2/3$, $b = d = \frac{3}{8\log 5}$, and $c = \frac{16\log^2 4}{3}$.

Basic Step:

1. Pair up the bits, and apply the boosting step to each pair.

2. Separate the bits into warm, cold and hot. Keep the $s/2$ warm bits, and $\frac{1+\epsilon^2}{4}s - \beta^{1/2}s^\gamma$ of the remaining bits, and discard the rest.

3. Discard the warm bits if their bias is too small (they are too warm). Output the cold bits if their bias is sufficiently large (they are sufficiently cold). The specifics of when to discard or output these bits will depend on whether this round of the purification procedure is being executed in phase I or phase II of the algorithm.

4. Repeat the Basic Step on each block of bits that was not discarded or output.

The two phases are as follows.

Phase I: Fix a target bias $\epsilon_1 = 1 - c\log\log n/\log n$. Then, for $b\log_5 n$ rounds, apply the basic procedure to those blocks of bits whose bias is less than ϵ_1 and greater than $n^{\frac{\gamma-1}{2}}$ (i.e. to all blocks that are not "hot" or nearly so, but on the other hand have not reached the target bias). Bits which complete this phase with a bias greater than ϵ_1 will be referred to as "certified". All other blocks of bits are set aside after phase I.

Phase II: Begin with each of the blocks of bits certified in phase I, and apply the basic procedure for $d\log_5 n$ rounds, discarding all hot (unbiased) bits as well as those which result from many applications of warm branches. To be specific: set

$$x_1 = \frac{k}{E_\Delta}\tanh^{-1}(\epsilon_1),$$

and for $i > 1$ set

$$x_i = x_{i-1}2^{(x_{i-1}/2\log 2)-1}.$$

Set $\ell_i = \frac{3E_\Delta x_i}{2k\log 2}$ and break up the $d\log_5 n$ rounds of phase II into $\log^*\log n$ subphases, consisting of rounds $\{1, ..., \ell_1\}$, $\{\ell_1+1, ..., \ell_1+\ell_2\}$, and so forth. During each subphase, discard only hot blocks. At the end of each subphase, discard every block generated by more than $2\ell_i/3$ warm branches within that subphase.

Analysis: We begin by recalling that $n > \epsilon_0^{-2}$ (and noting that indeed there is no reason to analyze this method, or use such a computing device, if n is smaller).

Phase I:
We show that at the end of phase I, the number of certified bits is at least $n(1 - H(\epsilon_0) - o(1))$. The analysis relies on a bound on the energy put into the system while carrying out the phase. Since our procedure is reversible, it preserves the entropy of the system. The initial entropy is $nH(\epsilon_0)$.

The energy of a bit of bias ϵ is $E_\mu - E_\Delta\epsilon$. Consider the change in energy that results from the application of one step of our procedure to a block of length s: each step replaces s "tepid" bits of bias ϵ, by approximately $\frac{1+\epsilon^2}{4}s$ cold bits of bias $\frac{2\epsilon}{1+\epsilon^2}$, exactly $\frac{1}{2}s$ warm bits of bias ϵ^2, and approximately $\frac{1-\epsilon^2}{4}s$ hot bits of bias 0. The energy per bit therefore increases from $E_\mu - E_\Delta\epsilon$ to $E_\mu - E_\Delta(\frac{1+\epsilon^2}{4}\frac{2\epsilon}{1+\epsilon^2} + \frac{1}{2}\epsilon^2 + \frac{1-\epsilon^2}{4}0) = E_\mu - E_\Delta(\frac{\epsilon}{2} + \frac{\epsilon^2}{2})$. The net increase in energy per bit is therefore $E_\Delta\epsilon(1-\epsilon)/2$. Our procedure may be viewed as implementing a reversible Carnot heat engine that transforms one tepid region into a cold region, a warm region, and a hot region at infinite temperature.

To continue with the analysis of Phase I, define the energy deficit of a bit of bias ϵ to be $E_\Delta\epsilon$ (this is the energy that must be added to raise the temperature of this bit to infinity). Notice that applying a round of our procedure to a block of bias ϵ reduces the energy deficit by a factor of $(1+\epsilon)/2$. Now, since we stop applying our procedure to bits whose bias exceeds ϵ_1, it follows that the energy deficit of all blocks of uncertified bits decreases by a factor of at least $(1+\epsilon_1)/2$ with each round of our procedure. Recalling that $\epsilon_1 = 1 - c\log\log n/\log n$ this implies that after the specified $b\log_5 n$ rounds, the total energy deficit of the blocks of uncertified bits is under $\delta E_\Delta n$ for $\delta = \log^{\frac{-bc\log 5}{2\log^2 4}} n$. The full collection of uncertified bits also consists of (a) those bits which were discarded at the margins of the cold blocks, and (b) blocks whose bias decreased below $n^{\frac{\gamma-1}{2}}$. The fraction of bits of type (a) discarded while working on a block of size s is $\beta^{1/2}s^{\gamma-1}$, and since there are a total of $b\log n$ rounds, the total number of such bits discarded is at most the maximum over β, s of $n\beta^{1/2}s^{\gamma-1}b\log n$. On the one hand $\beta < 1/2$, and on the other (relying on $\frac{1+\epsilon^2}{4}s - \beta^{1/2}s^\gamma \geq s/5$ which follows from $\epsilon > n^{-1/2}$), $s \geq n^{1-b}$, so at most $n^{1-(1-b)(1-\gamma)}b\log n$ type (a) bits are discarded. The total energy deficit of the type (b) bits is at most $E_\Delta n^{1+\frac{\gamma-1}{2}}$.

Therefore the total energy deficit of all the uncertified bits is under $E_\Delta n\delta'$ for $\delta' = \delta + n^{-(1-b)(1-\gamma)}b\log n + n^{\frac{\gamma-1}{2}}$.

We are now in a position to lower bound κ, the number of certified bits. Our upper bound on the energy deficit of the uncertified bits is equivalent to a statement that these bits have high average energy. Moreover because of the reversibility of our process, the total entropy of the uncertified bits is bounded above by the initial entropy of the system. In conjunction, these facts imply an upper bound on the number of uncertified bits. We have the following two inequalities for the uncertified bits.

$$\sum_{y=1}^{n-\kappa} \epsilon(y) \leq \delta' n \tag{1}$$

$$\sum_{y=1}^{n-\kappa} H(\epsilon(y)) \leq H(\epsilon_0)n \tag{2}$$

Note that H is convex and that $0 \leq \epsilon(y) \leq 1\ \forall y$. Invoking majorization [16, 19] we see that, subject to equation 1, $\sum_{y=1}^{n-\kappa} H(\epsilon(y))$ achieves its least value in case $\epsilon(y) = 1$ for $1 \leq y \leq \delta' n$ and $\epsilon(y) = 0$ for $\delta' n < y \leq n-\kappa$. In other words $\sum_{y=1}^{n-\kappa} H(\epsilon(y)) \geq n - \kappa - \delta' n$. So

$$\kappa \geq n(1 - H(\epsilon_0) - \delta')$$

$$= n(1 - H(\epsilon_0) - \log^{\frac{-bc\log 5}{2\log^2 4}} n - n^{-(1-b)(1-\gamma)}b\log n - n^{\frac{\gamma-1}{2}}).$$

Phase II:
In this phase the analysis focusses on the inverse temperature rather than on the energy. The certified bits from Phase I, which are the raw ingredient for this phase, already have a fairly high inverse temperature. This is the regime in which a temperature analysis yields good bounds. The main property of the temperature we shall use is: if we apply one round of our procedure to bits of inverse temperature x, then the cold bits that result have inverse temperature $2x$, and the warm bits that result have inverse temperature at least $x - \frac{k\log 2}{2E_\Delta}$.

Consider subphase i. Any block produced by more than $2\ell_i/3$ warm branches in this subphase, is discarded. The fraction of bits discarded is equivalent to the fraction of sequences of ℓ fair coin tosses in which there are more than $2\ell/3$ "heads"; according to a Chernoff bound this is at most $\exp(-\ell D(\frac{2}{3}, \frac{1}{2}))$. (Here D is Kullback-Liebler divergence, or relative entropy [10].) In the branches which are not discarded, the bias is at least that resulting from following first $2\ell/3$ warm branches and then $\ell/3$ cold branches. If the initial inverse temperature was x then after the warm branches it is at least $x - \frac{k\ell\log 2}{3E_\Delta}$, and after the cold branches, at least $(x - \frac{k\ell\log 2}{3E_\Delta})2^{\ell/3}$. On the assumption that the inverse temperature at the beginning of the subphase satisfied $x \geq \frac{2k\ell_i\log 2}{3E_\Delta}$, it follows that the inverse temperature at the end of the subphase is at least $x2^{\ell/3-1}$, and therefore also that it is at least $\frac{2k\ell_{i+1}\log 2}{3E_\Delta}$, justifying a similar analysis in the next subphase. Certification provides the necessary guarantee for the first subphase.

The entire above process is repeated for some $\log^*\log n$ subphases until the total quota of $d\log_5 n$ steps has been exhausted. During this phase, there are two ways in which bits (besides the hot blocks) are discarded: at the margins of cold blocks, and in blocks removed by the above "$2\ell_i/3$ warm branches" rule. The number of the latter form has already been discussed; the number at the margin of cold blocks is bounded as follows. The values of γ, b, d are such that we are guaranteed that no block size decreases by a factor of more than 5 between rounds (except of course the hot blocks we are not interested in); and so that every block ever used has size at least $n^{1/4}$. Consequently the total fraction of bits discarded at the margins in any round of of phase II is at most $n^{3/4}n^{(1/4)\cdot\gamma}$, and the total number of these bits discarded in phase II is at most $n^{3/4+\gamma/4}d\log n$.

At the end of phase II, unless an error event has happened (too small a block of cold bits somewhere), which occurs with probability at most $\exp(-n^{\Omega(1)})$, the inverse temperature is at least $x_1 2^{(d\log_5 n)/3 - \log^*\log n}$ which (noting that $\frac{k}{E_\Delta}x_1 > 2^{\log^*\log n}$ for sufficiently large n), is greater than $\frac{k}{E_\Delta}n^{d/5}$; the bias is at least $1 - \exp(-2n^{d/5})$; and the number of bits output is at least (accounting for both types of loss in phase

II)

$$
\begin{aligned}
& n(1-H(\epsilon_0) - \log^{\frac{-bc\log 5}{2\log^2 4}} n - n^{-(1-b)(1-\gamma)} b \log n \\
& -n^{\frac{\gamma-1}{2}} - \exp(-\ell_1 D(\frac{2}{3},\frac{1}{2})) \log^* \log n - n^{(\gamma-1)/4} d \log n) \\
\geq\; & n(1-H(\epsilon_0) \\
& -[\log^{\frac{-bc\log 5}{2\log^2 4}} n] - [n^{-(1-b)(1-\gamma)} b \log n] - [n^{\frac{\gamma-1}{2}}] \\
& -[\exp(-D(\frac{2}{3},\frac{1}{2}) \frac{3}{4\log 2} \log \frac{2\log n}{c \log\log n}) \log^* \log n] \\
& -[n^{(\gamma-1)/4} d \log n]).
\end{aligned}
$$

Substituting the chosen constants, the number of bits output is at least

$$
\begin{aligned}
& n(1-H(\epsilon_0) \\
& -[\log^{-1} n] - [n^{-(1-\frac{3}{8\log 5})/3} \frac{3}{8\log 5} \log n] - [n^{-1/6}] \\
& -[(\log n)^{-D(\frac{2}{3},\frac{1}{2})\frac{3}{4\log 2}} (\frac{8\log^2 4}{3} \log\log n)^{D(\frac{2}{3},\frac{1}{2})\frac{3}{4\log 2}} \log^* \log n] \\
& -[n^{-1/12} \frac{3}{8\log 5} \log n]).
\end{aligned}
$$

The last five terms tend to 0 for large n, and this completes the proof of the theorem.

Architecture and Implementation

We restrict the discussion of our implementation to the context of NMR in polymers suspended in liquids. The basic ideas may carry over to other proposed implementations, e.g. that of Kane [17].

The architecture we construct will perform as a cellular automaton, with one special site which we will refer to, in the language of Turing machines (or magnetic recording media), as the "head".

To achieve this we will use a linear polymer primarily composed of eight basic species of nuclei, $A, B, C, D, E,\ F, G, H$, along with one instance of another species, I. These will be nuclei of some chemical element, whose resonant frequencies are chemically shifted by the presence of different kinds of radicals. (Alternatively, nuclei of different elements might be used.) The species will be in arranged in a periodic sequence whose fundamental unit is $ABCDEFGH$. The use of a periodic sequence was suggested already by Lloyd [18]. What is novel in our design is the implementation of several "tapes" on the single linear polymer (this is the reason we use more than the three species which suffice for a single-tape machine), and the manner in which our general purification algorithm is implemented on this multitape architecture. (It should be noted that this is a multitape cellular automaton, not multitape Turing machine; and that there can be interactions among the corresponding cells of the different tapes.) At the head location another radical is present, so that an I appears instead of an A. The entire polymer, therefore, is the sequence $(ABCDEFGH)^k IBCDEFGH(ABCDEFGH)^{n-k-1}$.

The primitive operations that we can apply to this machine, act on pairs of adjacent spins. For the device-preparation algorithm of the present paper, the only kinds of primitives required are nots, controlled-nots (CNOTs) and controlled-controlled-nots (CCNOTs) among adjacent spins. (These 3-qubit operations can be implemented by a sequence of 2-qubit operations.) For a subsequent general quantum computation, these need to be augmented also by another rotation. For a concise notation we will mix upper and lower case characters so that, for example, (aB) denotes the CNOT operation $(x,y) \to (x, x \oplus y)$ on all pairs of adjacent A and B spins, where x is the A spin and y is the B spin. Similarly $(1B)$ denotes the NOT operation applied to B spins; and (abC) denotes the CCNOT operation $(x,y,z) \to (x,y,(x \wedge y) \oplus z)$, where x is the A spin, y is the B spin and z is the C spin.

Transpositions will be used often in our algorithm, and can be implemented by a sequence of CNOTs; we will let (AB) denote the transposition of all pairs of adjacent A and B spins. Controlled transpositions can be implemented by a sequence of CCNOTs; we will let (aBC) denote the operation which exchanges B and C spins conditioned on the adjacent A spin being a $|1\rangle$.

A higher-level primitive we can construct from these operations, is to conceive of the device as a four-tape Turing Machine, with the I nucleus and the A and E nuclei forming the first tape ($A^k I A^{n-k-1} E^n$, reading the A's "to the right" along the polymer and then the E's to the left); the B and F nuclei forming the second tape ($B^n F^n$, reading the B's to the right and then the F's to the left); and similarly the C and G nuclei forming the third tape and the D and H nuclei forming the fourth tape. Tape one (the A-E tape) can be rotated by one position without affecting the other tapes, by the sequence of operations (see figure 1):

$$\begin{aligned}(\rho A) \quad = \quad & (DE)(CD)(BC)(AB)(IB)(BC)(CD)(DE) \\ & (EF)(FG)(GH)(HA)(HI)(GH)(FG)(EF)\end{aligned}$$

(We write such sequences left-to-right. In this example reversing the order merely reverses the direction of rotation of the tape.) This is essentially a 14-step procedure, plus a duplicated step with I for each step involving A.

Similarly the second (B-F) tape can be rotated by one position without affecting the other tapes, by the sequence of operations

$$\begin{aligned}(\rho B) \quad = \quad & (EF)(DE)(CD)(BC)(AB)(IB)(CD)(DE)(EF) \\ & (FG)(GH)(HA)(HI)(AB)(IB)(GH)(FG)\end{aligned}$$

and the third (C-G) and fourth (D-H) tapes can be rotated in analogous manners.

We describe the implementation of our algorithm for the case where the value of the initial polarization ϵ_0 is "moderate", not more than 0.4. Asymptotically optimal efficiency in linear time can be obtained for higher values as well, with no additional difficulty, by minor modification of our method; the periodic sequence should be such that the "tapes" on which computation is performed (as opposed to the "storage" tapes; this distinction will be clearer below) occupy a smaller fraction of the overall polymer.

We implement 10 iterations of the identical algorithm; each of these will start with $n/10$ of the original ϵ_0 bits, as its initial block of bits. From these, of course, will be formed each of the subsequent smaller blocks in the tree of the algorithm. We will execute each of the ten processes completely before beginning the next.

At any time, the bits in the computer are of several types: untouched ϵ_0-biased bits; "hot" bits from already-executed stages of the algorithm, of no remaining interest to the algorithm; and blocks of already-processed bits of various finite temperatures, which we call in-process. (Note that in-process bits include any bits which have been completely purified.)

Our implementation works by, at all times, storing all of the in-process blocks on the D-H spins, with the exception of the single block of bits that is about to undergo a purification step. Once that purification step is done, the blocks that result from it are loaded into the D-H spins, and the next block to be worked on is retrieved. Storage of all in-process bits on D-H spins is possible due to the assumption that the initial bias is at most 0.4 and the fact that we are working on units of $n/10$ bits at a time.

All that remains to be specified is how one step of the purification procedure can be applied to a block of s bits, in time proportional to s.

The block is "loaded" onto the A-E, B-F and C-G tapes, as follows. $s/2$ bits are loaded into the A sites beginning at the head (I) site and proceeding to its left (understood cyclically, so that once the left end of the device is reached, E sites are filled from left to right). Let the bias of this block be ε. Another $s(1+\varepsilon)/4$ bits of the block are loaded into the C (and if necessary G) sites to the left of the head; while the remaining $s(1-\varepsilon)/4$ bits of the block are loaded into the B (and if necessary F) sites starting immediately to the right of the head and proceeding to the left. (See figure 2.)

Then the following sequence of steps is repeated $s/2$ times:

$$(Gb)(\rho A)(\rho B)(\overline{a}BC)(\overline{e}FG)$$

To understand this operation, consider a bit x lying at an A site. If it is a 1, then as it travels from A site to A site, it is accompanied by the bit at the adjacent B site; this pair of bits arrives together at the head site. However, if the A bit is a 0, then it repeatedly exchanges the adjacent B and C bits and then is accompanied by that B bit for just one rotation; so, the A bit will arrive at the head accompanied by the bit it collected at the last C site before the head. Thus, the 1-bits at the A sites arrive accompanied by bits that originated at their neighboring B sites; while the A site holding the k'th 0 away from the head, arrives at the head accompanied by the bit originating at the k'th C site from the head.

Let a and b denote the pair of bits arriving at the head and at the B to its right, in some step. After passing through the head, these bits have been transformed into $a' = a \oplus b$ and $b' = b$. Therefore if the bits continue, if $a \neq b$ (and thus $a' = 1$), the b bit will be carried through the B sites; while if

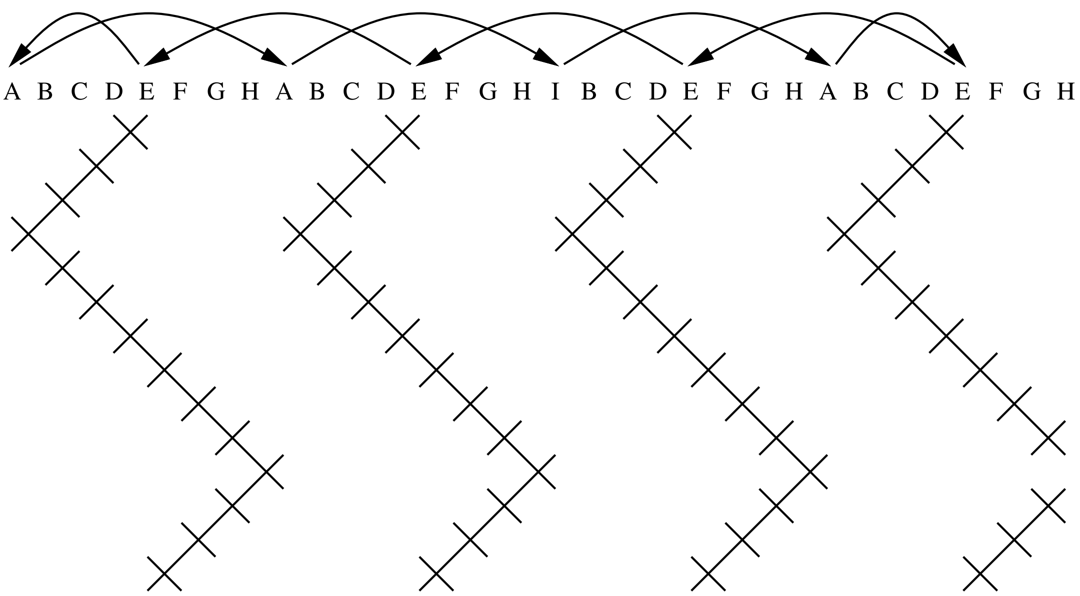

Figure 1: Transpositions for rotation of the A-E tape.

ABCDEFGHABCDEFGHABCDEFGHABCDEFGHABCDEFGHI BCDEFGHABCDEFGHABCDEFGH
* * *   * * *   *** *   ***   ***   **

Figure 2: A block of length $s = 18$ loaded for a purification step.

$a = b$, the b bit will be deposited at the first C site, and the remaining C sites will be shifted aside to make room for it.

At the end of the phase, the three resulting blocks can be collected as follows. The "warm" bits are simply at the $s/2$ A sites to the right of the head. The "cold" bits are in a contiguous block of C sites to the right of the head; a large deviation bound ensures a good estimate of the length of this block. The "hot" bits are at the B sites to the right of A sites carrying a 1, and that can be used to separate them from the other B sites which carry unrelated bits (in fact, bits not originating in this block at all, which resided at the C sites to the right of the head.)

(Note: actually the sequence of steps is repeated slightly fewer than $s/2$ times, to guarantee that with high probability no bits from outside of the block become mixed in with the block; this is accounted for in the large deviation bounds of the analysis. The remaining bits of the block, along with a few extra beyond the estimated end of the block, are collected in a "garbage" block of $O(\log n)$ bits. The bits around the estimated end of the C block, at the end of the processing phase, are also collected into the garbage block so that with high probability only truly cold bits are collected from the C sites.)

Once this purification phase has been completed, the in-process blocks that result from it are loaded into the D-H tape, which is treated like a "stack" data structure, with the "push" and "pop" operations performed adjacent to the head (I) site: hence storing or retrieving a block of s bits from the "stack", and storing it in or extracting it from the A, B, C, E, F, G sites nearest the head, can be performed in time proportional to s.

Initial Distributions with Local Correlations
The analysis has been conducted assuming complete independence in the initial thermal distribu-

tion.

It is possible that in a laboratory situation it will be found that the assumption of full independence among the initial spins of the polarized system (or more precisely, the corresponding assumption for the density matrix), is inaccurate to a degree which invalidates the analysis given. In that case a slight modification of the process is possible, which should alleviate most concerns; this is described below.

The most likely kind of failure of the independence assumption is that there are local correlations among nearby spins on the molecule; these correlations may fall off polynomially with the distance between the spins. Moreover, we can handle even arbitrary correlations subject only to an assumption on the total expected polarization of the machine.

We begin by permuting the bits. If we can make no assumption of locality of the the dependencies among spins in the original thermal mixture, then the permutation of $\{1, ..., n\}$ is chosen at random, uniformly, by the experimenter. (And applied deterministically to the quantum computer.) If we can assume greater locality of correlations then it is enough to "shuffle" the bits in any predetermined manner that guarantees that all bits that start out close to each other (say, within distance $n^{1/3}$) end up far apart (at least distance $n^{1/3}$.) These permutations can be executed in time $O(n^2)$. The result is that the distribution on every segment of $n^{1/3}$ consecutive bits is exponentially close in variation distance to the independent distribution with the same overall polarization.

Next partition the computer into segments of length $n^{1/3}$, and carry out the entire purification process separately within each segment. Finally, gather the cold bits obtained from each of the $n^{2/3}$ segments.

The runtime has been adversely affected, but it should be noted that a runtime proportional to nr suffices to execute a permutation which ensures that every bit is at least distance r away from any other bit that began within distance r. Assuming that only local correlations dominate in practice, the actual value of r required to ensure a satisfactory computation is likely to be very small, say $O(n^\beta)$ for small β, hence the actual runtime of the algorithm will remain close to optimal even in the presence of local correlations in the initial polarized thermal mixture. Alternatively, with a small change in architecture the runtime can remain quasi-linear.

Acknowledgments

Thanks to Isaac Chuang, Peter Hoyer, Alex Pines and Richard Singerman for helpful discussions.

References

[1] Bernstein, E. and Vazirani, U., "Quantum complexity theory", *Special issue of SIAM J. Comp., October, 1997. Proc. 25'th ACM Symp. on Theory of Computing*, 1993, pp. 11 – 20.

[2] S. Carnot, *Reflections on the Motive Power of Heat (1824).* ASME, New York 1943.

[3] Chirac, J. and Zoller, P., *Phys. Rev. Lett.* 74, pp. 4091 (1995).

[4] I. Chuang, N. Gershenfeld and M. Kubinec, "Experimental Implementation of Fast Quantum Searching", *Physical Review Letters*, 80(15) 3408-3411, 13 April 1998.

[5] I.L. Chuang, L.M.K. Vandersypen, X. Zhou, D.W. Leung and S. Lloyd, "Experimental realization of a quantum algorithm", *Nature* 393, 143-146 (1998).

[6] Chuang, I., Yamamoto, Y., *Phys. Rev. Lett.* 52, pp. 3489 (1995).

[7] Cleve, R., and DiVincenzo, D., "Schumacher's quantum data compression as a quantum computation", *Phys Rev A* 54, 1636, 1996.

[8] Cory, D. G., Fahmy, A. F., and Havel, T. F. "Ensemble Quantum Computing by Nuclear Magnetic Resonance Spectroscopy", *Proc. Natl. Acad. Sci.*, 94:1634-1639, 1997.

[9] D. G. Cory, W. Mass, M. Price, E. Knill, R. Laflamme, W. H. Zurek, T. F. Havel and S. S. Somaroo, "Experimental Quantum Error Correction", 1998.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley 1991.

[11] Deutsch, D. and Jozsa, R., "Rapid solution of problems by quantum computation", *Proc. Royal Society London*, vol. A439, 1992, pp. 553 – 558.

[12] Domokos, P., Rainond, J., Brune, M., Horache, S., *Phys. Rev. Lett.* 52, pp. 3554 (1995).

[13] P. Elias, "The efficient construction of an unbiased random sequence", *Ann. Math. Statist.* 43 865-870, 1972.

[14] R. P. Feynman, R. B. Leighton and M. Sands, *The Feynman Lectures on Physics*, vol. I, Addison-Wesley 1964.

[15] Gershenfeld, N., Chuang, I., *Science*, 275, pp. 350 (1997).

[16] G. Hardy, J. E. Littlewood and G. Pólya, *Inequalities*, second edition, Cambridge University Press, 1952.

[17] B. Kane, *Nature* 393 1998, 133-137.

[18] S. Lloyd, "A Potentially Realizable Quantum Computer". *Science*, 17 September 1993 1569-1571.

[19] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and its Applications*, Academic Press, 1979.

[20] J. von Neumann, "Various Techniques used in connection with random digits", *National Bureau of Standards Applied Mathematics Series* 12 36-38 1951.

[21] Y. Peres, "Iterating von Neumann's Procedure for Extracting Random Bits", *Ann. Stat.* 20 (1): 590-597 1992.

[22] B. Schumacher, "Quantum Coding", *Phys. Rev. A.* 51, 2738 (1995).

[23] Shor, P.W., "Algorithms for quantum computation: Discrete logarithms and factoring", *Special issue of Siam J. Comp., October, 1997. Proc. 35'th Annual IEEE Symp. on Foundations of Computer Science*, 1994, pp. 124 – 134.

[24] Simon, D., "On the power of quantum computation", *Special issue of Siam J. Comp., October, 1997. Proc. 35'th Annual IEEE Symp. on Foundations of Computer Science*, 1994, pp. 116 – 123.

[25] M. Tribus, *Thermostatics and Thermodynamics*, van Nostrand 1961.

[26] Turchette, Q., *Phys. Rev. Lett.* 75, pp. 4710 (1995).