

# The Effect of Communication Costs in Solid-State Quantum Computing Architectures

Dean Copsey<sup>‡</sup>, Mark Oskin<sup>†</sup>, Tzvetan Metodiev<sup>‡</sup>,  
Frederic T. Chong<sup>‡</sup>, Isaac Chuang<sup>◊</sup>, and John Kubiatowicz<sup>◊</sup>

<sup>‡</sup> University of California at Davis, <sup>†</sup> University of Washington,  
<sup>◊</sup> Massachusetts Institute of Technology, <sup>◊</sup> University of California, Berkeley

## ABSTRACT

Quantum computation has become an intriguing technology with which to attack difficult problems and to enhance system security. Quantum algorithms, however, have been analyzed under idealized assumptions without important physical constraints in mind. In this paper, we analyze two key constraints: the short spatial distance of quantum interactions and the short temporal life of quantum data.

In particular, quantum computations must make use of extremely robust error correction techniques to extend the life of quantum data. We present optimized spatial layouts of quantum error correction circuits for quantum bits embedded in silicon. We analyze the complexity of error correction under the constraint that interaction between these bits is near neighbor and data must be propagated via swap operations from one part of the circuit to another.

We discover two interesting results from our quantum layouts. First, the recursive nature of quantum error correction circuits requires a additional communication technique more powerful than near-neighbor swaps – too much error accumulates if we attempt to swap over long distances. We show that quantum teleportation can be used to implement recursive structures. We also show that the reliability of the quantum swap operation is the limiting factor in solid-state quantum computation.

## Categories and Subject Descriptors

C.1 [Processor Architectures]: Miscellaneous; C.4 [Performance of Systems]: Fault Tolerance; C.5 [Computer System Implementation]: Miscellaneous; E.4 [Coding and Information Theory]: Error control codes

## General Terms

Performance, Design, Algorithms, Reliability

## Keywords

quantum computing, quantum architecture, silicon-based quantum computing

## 1. INTRODUCTION

Physical systems that behave quantum-mechanically have dynamics which can be exploited to speed up certain computational tasks. This is the essential thought behind the field of quantum computation and quantum information. A significant challenge arises in implementing quantum computation, however, because quantum systems are unstable: their quantum state is easily altered by omnipresent extraneous noise. This problem of *decoherence* was once thought to be a fundamental problem for quantum information processing [8], but the discovery of fault-tolerant constructions [1, 23, 27, 10] changed this; it is now known that an arbitrarily reliable quantum computer can be constructed from unreliable quantum wires and gates, as long as certain conditions are met. These constructions are made possible by recursive application of quantum error correction, generalizing the classical version of von Neumann's early constructions for reliable automata [37, 39].

The conditions for fault-tolerant quantum computation are as follows: First, the probability of failure of each elementary component must be less than some threshold value  $p_{th}$ , currently estimated to be around  $10^{-4}$ . Second, current fault models assume that errors are independent and uniformly distributed (although other error models can also be dealt with by changing the scheme appropriately). Third, and most interesting, a variety of assumptions are made about both the quantum circuit and the necessary classical controller. In particular, it is essential that the quantum circuit employ maximum parallelism – executing as many quantum gates simultaneously as possible – and that the classical circuitry controlling the quantum operations run at a much higher clock speed than the quantum circuitry. Without these properties,  $p_{th}$  decreases significantly [1, 10].

Here, we take this study one step further, and consider the impact of *physical layout* on the requirements for fault-tolerant quantum computation. Do realistic physical implementations of these machines allow achievable fault-tolerance thresholds? In particular, what constraints must be satisfied in the architectural design of a quantum computer in order to allow a reliable machine to be realized?

Such questions can now be seriously considered in light of recent progress in the physical implementation of quantum computers, with a wide variety of systems ranging from spins in molecules [9] and single photons [18], to spins in semiconductors [16], trapped ions [19, ?], and superconducting systems [36], among others. These systems have led to successful demonstrations of a wide variety of quantum information processing tasks, including quantum teleportation [4], creation of multiple quantum-bit entangled states [24], fast quantum search [7, 14], and recently, Shor's fast quantum factoring algorithm [35], in factoring the number fifteen, using a seven quantum bit (qubit) machine.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SPAA '03, June 7–9, 2003, San Diego, California, USA.  
Copyright 2003 ACM 1-58113-661-7/03/0006 ...\$5.00.

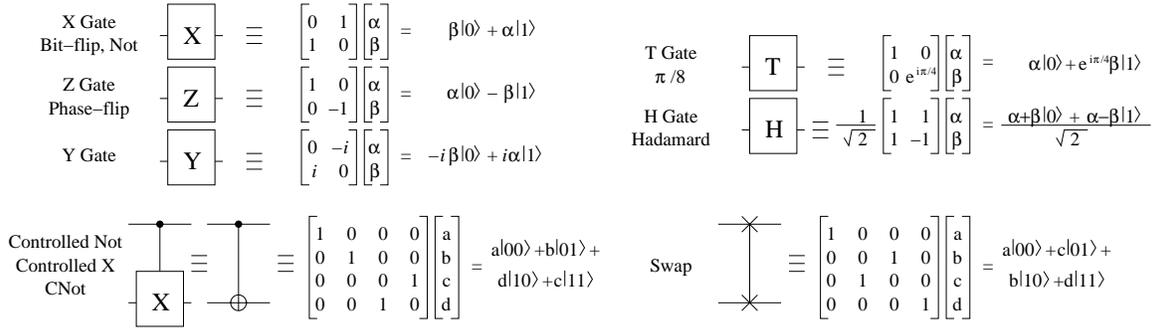


Figure 1: Basic quantum gates and their matrix representations

Among these implementations, the solid state systems are perhaps the most intriguing, because of the extensive investment that has been made in semiconductor technology for conventional classical computing, and the potential for scaling to large numbers of qubits. One such scheme, proposed by Kane, is particularly well suited for architectural study; it captures common elements from the whole range of implementations, using the nuclear spins of dopant atoms in silicon as qubits, classically controlled metal electrodes for control of quantum gates, and near neighbor, planar spin-spin interactions for multi-qubit gates. This scheme is also suitable for VLSI style CAD layout and modeling, and reveals an interesting constraint arising from pitch-matching large classical wires to small qubits, which forces computation units to be distributed in clusters rather than a single sea-of-qubits structure [22].

Our study of the architectural constraints on fault-tolerant quantum computation builds on the scenario posed by the Kane solid-state implementation proposal, and within this framework we obtain several interesting results. We first present complete layouts of qubits and gate sequences required to implement a concatenated seven-qubit Steane code for recursive quantum error correction. These layouts give us analytic expressions for the circuit's space and time resource requirements as a function of desired system reliability. We also consider the impact of planar near neighbor interactions on  $p_{th}$  and find that a huge limiting role will be played by a single gate, the SWAP gate, in determining achievable reliabilities.

We begin our study in the next two sections with a brief overview of quantum computation and error correction in quantum systems. In Section 4, we discuss the model we will be using for the rest of the paper, and the limitations it and similar models impose. Section 5 discusses implementations for error correction codes, while section 6 discusses the impact of communication on error correction algorithms. Finally, Section 7 discusses future work, while Section 8 concludes.

## 2. QUANTUM COMPUTATION

We begin with a brief overview of the basic terminology and constructs of quantum computation. Our purpose is to introduce the language necessary for subsequent sections; in-depth treatments of these subjects are available in the literature [21].

### 2.1 Quantum States: Qubits

The state of a classical digital system  $X$  can be specified by a binary string  $\mathbf{x}$  composed of a number of bits  $x_i$ , each of which uniquely characterizes one elementary piece of the system. For  $n$  bits, there are  $2^n$  possible states. The state of an analogous quantum system  $\psi$  is described by a complex-valued vector  $|\psi\rangle = \sum_x c_x |\mathbf{x}\rangle$ , a weighted combination (a "superposition") of the basis vectors  $|\mathbf{x}\rangle$ ,

where the *probability amplitudes*  $c_x$  are complex numbers whose modulus squared sums to one,  $\sum_x |c_x|^2 = 1$ .

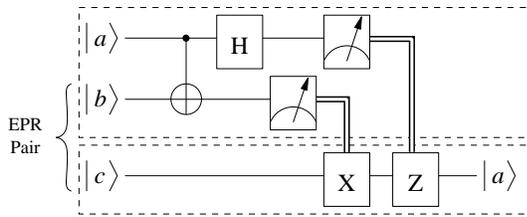
A single quantum bit is commonly referred to as a *qubit* and is described by the equation  $|\psi\rangle = c_0|0\rangle + c_1|1\rangle$ , where the  $c_i$  are complex valued. Legal qubit states include pure states, such as  $|0\rangle$  and  $|1\rangle$ , and states in superposition, such as  $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ , or  $\frac{1}{2}|0\rangle - i\frac{\sqrt{3}}{2}|1\rangle$ . Larger quantum systems can be composed from multiple qubits, for example,  $|00\rangle$ , or  $\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle - \frac{1}{\sqrt{2}}|11\rangle$ . An  $n$ -qubit state is described by  $2^n$  basis vectors, each with its own complex probability amplitude, so an  $n$ -qubit system can exist in an arbitrary superposition of the possible  $2^n$  classical states of the system.

Unlike the classical case, however, where the total can be completely characterized by its parts, the state of larger quantum systems cannot always be described as the product of its parts. This property, known as *entanglement*, is best illustrated with an example: there exist no single qubit states  $|\psi_A\rangle$  and  $|\psi_B\rangle$  such that the two-qubit state  $|\Psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$  can be expressed as the composite state<sup>1</sup>  $|\psi_A\rangle \otimes |\psi_B\rangle$ . Entanglement and superposition have no classical analogues: they give quantum computers their computational powers.

Although a quantum system may exist in a superposition of states, only one of those states can be observed, or measured. After measurement, the system is no longer in superposition: the quantum state collapses into the one state measured, and probability amplitude of all other states goes to 0. For example, when the state  $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$  is measured, the result is either 00 or 11, with equal probability; the outcomes  $|01\rangle$  or  $|10\rangle$  never occur. Furthermore, if a subset of the qubits in a system is measured, the remaining qubits are left in a state consistent with the measurement.

Since measurement of a quantum system only produces a single result, quantum algorithms must maximize the probability that the result measured is the result desired. This may be accomplished by iteratively amplifying the desired result, as in Grover's fast database search,  $O(\sqrt{n})$  for a dataset of size  $n$  [11]. Another option is to arrange the computation such that it does not matter which of many random results is measured from a qubit vector. This method is used in Shor's algorithm for factoring the product of two large primes [26], which is built upon modular exponentiation and a quantum Fourier transform. For the interested reader, quantum algorithms for a variety of problems other than search and factoring have been developed: adiabatic solution of optimization problems (the quantum analogue of simulated annealing; complex-

<sup>1</sup>The composition operator for quantum systems is the tensor product,  $\otimes: |\mathbf{x}\rangle \otimes |\mathbf{y}\rangle = \sum_x c_x |x\rangle \otimes \sum_y c_y |y\rangle = \sum_{x,y} c_x c_y |x \otimes y\rangle$ , where  $x \otimes y$  is simply the string formed by concatenating  $x$  and  $y$ .



**Figure 2: Quantum Teleportation: Quantum Teleportation of state  $|a\rangle$ . First, entangled qubits  $|b\rangle$  and  $|c\rangle$  are exchanged. Then,  $|a\rangle$  is combined with  $|b\rangle$  after which two classical bits of information (double lines) are produced via measurement (“meter” boxes). After transport, these bits are used to manipulate  $|c\rangle$  to regenerate state  $|a\rangle$  at destination.**

ity unknown) [5], precise clock synchronization (using EPR pairs to synchronize GPS satellites) [15, 6], quantum key distribution (provably secure distribution of classical cryptographic keys) [3], and very recently, Gauss sums [33], testing of matrix multiplication (in  $O(n^{1.75})$  steps versus the  $O(n^2)$  required classically) [13], and Pell’s equation [12].

## 2.2 Quantum Gates and Circuits

Just as classical bits are manipulated using gates such as NOT, AND, and XOR, qubits are manipulated with quantum gates such as those shown in Figure 1. A quantum gate is described by a unitary operator  $U$ . The output state vector is the operator applied to the input vector; that is,  $|\psi_{\text{out}}\rangle = U|\psi_{\text{in}}\rangle$ . The classical NOT has the quantum analogue  $X$  which inverts the probabilities of measuring 0 and 1. The quantum analogue of XOR is the two-qubit CNOT gate: the *target* qubit is inverted for those states where the *source* qubit is 1. Most quantum gates, however, have no classical analogue. The  $Z$  gate flips the relative phase of the  $|1\rangle$  state, thus exchanging  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  and  $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ . The Hadamard gate  $H$  turns  $|0\rangle$  into  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$  and  $|1\rangle$  into  $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ ; it can be thought of as performing a radix-2 Fourier transform. Another important single-qubit gate,  $T$ , leaves  $|0\rangle$  unchanged but multiplies  $|1\rangle$  by  $\sqrt{i}$ . Single qubit gates are characterized by a rotation around an axis:  $X$  rotates the qubit by  $\pi$  around the  $\hat{x}$ -axis;  $Z$  rotates by  $\pi$  around the  $\hat{z}$ -axis; and  $T$  rotates by  $\pi/4$  around the  $\hat{z}$  axis. By composing the  $T$  and  $H$  gates, any single-qubit gate can be approximated to arbitrary precision. The combination of  $T$ ,  $H$ , and CNOT provide a *universal set*: just as any Boolean circuit can be composed from AND, OR, and NOT gates, any polynomially describable multi-qubit quantum transform  $U$  can be efficiently approximated by composing just these three quantum gates into a circuit.

One additional important operator is the SWAP gate. Just as two classical values can be swapped using three XOR’s, a quantum SWAP can be implemented as three CNOTs. However, SWAP is often available natively for a given technology, which is valuable, given its importance to quantum communication.

Figure 2 shows a *quantum circuit* for teleportation (described in the next section). In quantum circuits, time goes from left to right, where single lines represent qubits, and double lines represent classical bits. A meter represents measurement. By convention, black dots represent control terminals for quantum-controlled gates. The symbol  $\oplus$  is shorthand for the target qubit of the CNOT gate.

## 2.3 Quantum Teleportation

Quantum teleportation is the re-creation of a quantum state at a distance, using only classical communication. It accomplishes this

feat by using a pair of entangled qubits,  $|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ , called an EPR pair<sup>2</sup>.

Figure 2 gives an overview of the teleportation process. We start by generating an EPR pair. We separate the pair, keeping one qubit,  $|b\rangle$ , at the source and transporting the other,  $|c\rangle$ , to the destination. When we want to send a qubit,  $|a\rangle$ , we first interact  $|a\rangle$  with  $|b\rangle$  using a CNOT gate. We then measure the phase and the amplitude of  $|a\rangle$ , send the two one-bit classical results to the destination, and use those results to re-create the correct phase and amplitude in  $|c\rangle$  such that it takes on the original state of  $|a\rangle$ . The re-creation of phase and amplitude is done with  $X$  and  $Z$  gates, whose application is contingent on the outcome of the measurements of  $|a\rangle$  and  $|b\rangle$ . Intuitively, since  $|c\rangle$  has a special relationship with  $|b\rangle$ , interacting  $|a\rangle$  with  $|b\rangle$  makes  $|c\rangle$  resemble  $|a\rangle$ , modulo a phase and/or amplitude error. The two measurements allow us to correct these errors and re-create  $|a\rangle$  at the destination. Note that the original state of  $|a\rangle$  is destroyed when we take our two measurements<sup>3</sup>.

Why bother with teleportation when we end up transporting  $|c\rangle$  anyway? Why not just transport  $|a\rangle$  directly? First, we can pre-communicate EPR pairs with extensive pipelining without stalling computations. Second, it is easier to transport EPR pairs than real data. Since  $|b\rangle$  and  $|c\rangle$  have known properties, we can employ a specialized procedure known as *purification* to turn a collection of pairs partially damaged from transport into a smaller collection of asymptotically perfect pairs. Third, transmitting the two classical bits resulting from the measurements is more reliable than transmitting quantum data.

## 3. FAULT-TOLERANT COMPUTATION

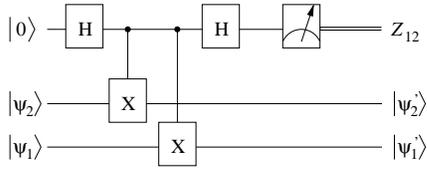
We turn now to an outline of the basic constructions of fault-tolerant quantum computation. This is a rather involved subject (for which the reader is referred to the literature [21, 10]), but three essential ideas are covered here. The main result we build upon is the following: *A quantum circuit containing  $N$  error-free gates can be simulated with a probability of failure of at most  $\epsilon$  using  $O(\text{poly}(\log(N/\epsilon))N)$  imperfect gates which fail with probability  $p$  as long as  $p < p_{th}$ , where  $p_{th}$  is a constant threshold that is independent of  $N$ .* This remarkable result, the *Threshold Theorem* [1], is achieved by three steps: (1) using quantum error-correction codes (Section 3.1), (2) performing all computations on encoded data, using *fault tolerant procedures* (Section 3.2), and (3) recursively encoding until the desired reliability is obtained (Section 3.3). All of these results are from prior literature [1, 28, 31, 21, 10], but we describe them here to make our contributions clearer in future sections.

### 3.1 Quantum Error Correction

The only error which can occur to a classical bit is a bit-flip, which can be modeled as a random NOT gate. Quantum bits suffer more kinds of error, because of the greater degree of freedom in their state representation; surprisingly, however, there are general strategies for reducing the universe of possible quantum errors to only two kinds: bit-flips (random  $X$  gates), and phase-flips (random  $Z$  gates). Classical error correction codes only take into account bit flip errors, and thus are insufficient for correcting quantum data; furthermore, quantum states collapse upon measurement, so strategies must be employed for determining errors without actually measuring encoded data.

<sup>2</sup>An EPR or Einstein-Podolsky-Rosen pair is a special instance of entanglement noted in the Einstein-Podolsky-Rosen paradox [2].

<sup>3</sup>This is consistent with the *no-cloning* theorem, which states that a quantum state cannot be copied.



**Figure 3: Quantum circuit for measuring  $Z_{12}$ , the phase difference between  $\psi_2$  and  $\psi_1$ . The meter box indicates measurement and double lines indicate classical information.**

$Z_{12}$	$Z_{23}$	Error Type	Action
0	0	no error	no action
0	1	qubit 3 flipped	flip qubit 3
1	0	qubit 1 flipped	flip qubit 1
1	1	qubit 2 flipped	flip qubit 2

**Table 1: Phase correction for a 3-qubit code**

Classical error correction relies upon distributing  $k$  bits of information across  $n$  bits,  $n > k$ , and ensuring enough redundancy to recreate the original information. Because of the no-cloning theorem, quantum information cannot be simply duplicated. Instead, redundancy is achieved through entangled states with known properties. For example, a single logical qubit,  $c_0|0_L\rangle + c_1|1_L\rangle$  can be represented using three physical qubits, as the state  $c_0|000\rangle + c_1|111\rangle$ . A bit flip error on the first (left-most) qubit would turn this into  $c_0|100\rangle + c_1|011\rangle$ ; this error can be detected by computing the *parity* of each pair of qubits, and leaving the result in an extra qubit called an *ancilla*. The three parities give the *error syndrome*, uniquely locating any single bit-flip error. Crucially, this strategy reveals nothing about the coefficients  $c_0$  and  $c_1$ , since the parities cannot distinguish between  $|000\rangle$  and  $|111\rangle$  or any single bit-flip version of the two three-qubit strings. By measuring parities, errors can be detected without collapsing encoded data.

Correcting phase flips is achieved by measuring differences in phase, using a circuit like the one in Figure 3. This works by using a Hadamard gate to transform phase flips into bit flips; parities are then measured as before, the results stored in ancilla qubits, and then the qubits are transformed back into their original basis. Figure 4 shows how a phase error syndrome can be computed and a corresponding correction procedure applied to correct the error, following the specification of Table 1.

A quantum code which encodes one qubit and allows any single bit-flip or phase-flip error to be corrected uses the encoding  $c_0|0_L\rangle + c_1|1_L\rangle$ , where the logical zero and one qubits are

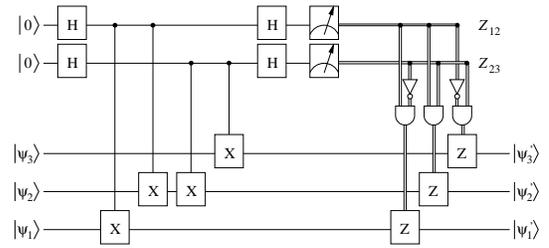
$$|0_L\rangle = \frac{(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1_L\rangle = \frac{(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle)}{2\sqrt{2}}$$

This nine qubit code, discovered by Peter Shor [28], is also known as the  $[[9, 1, 3]]$  code, in the notation  $[[n, k, d]]$ , where  $n$  is the number of physical qubits,  $k$  is the number of logical qubits encoded, and  $d$  is the quantum Hamming distance of the code. A code with distance  $d$  is able to correct  $(d - 1)/2$  errors.

## 3.2 Computing on Encoded Data

The nine qubit code has a remarkable property that illustrates a key requirement for fault tolerance: applying a  $Z$  gate to each of



**Figure 4: Syndrome Measurement for a 3-qubit Code. The classical results of measurement (double lines) control application of the  $Z$  operator.**

the nine qubits takes  $|0_L\rangle$  to  $|1_L\rangle$  and vice versa. It is the same as applying a logical  $\bar{X}$  operator<sup>4</sup> to the encoded qubit! Similarly,  $\bar{Z}$  can be performed by applying an  $X$  operator to each qubit, and  $\bar{H}$  by applying an  $H$  operator to each qubit.

In this paper, we employ Steane’s  $[[7, 1, 3]]$  code [30], which also allows simple computation on encoded data, but requires two fewer physical qubits. In addition, a CNOT gate on two encoded qubits can be accomplished using seven CNOT gates, between each pair of corresponding physical qubits. The last remaining gate necessary to achieve the universal set from Section 2.2, the  $T$  gate, can also be performed, albeit with some extra effort [21]. Thus, universal computation is possible without requiring that the data be decoded.

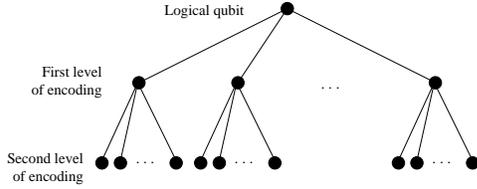
Merely computing on encoded data is not sufficient, however; one additional step is required, which is frequent, periodic error correction. Because all gates used in this task are assumed to be subject to failure, this must be done in a careful manner, such that no single gate failure can lead to more than one error in each encoded qubit block. Such constructions are known as *fault tolerant procedures*, and the impact of this requirement on our study is twofold: (1) no single operation may cause multiple failures, and (2) measurement errors must not be allowed to propagate excessively. To achieve (1), no two encoding qubits are allowed to both interact directly with a third qubit. Instead, the “third” qubit is replaced with a *cat state* (a generalization of an EPR pair),  $\frac{1}{\sqrt{2}}(|00\dots 0\rangle + |11\dots 1\rangle)$ , that has itself been verified. Cat states are used because they do not transmit errors through CNOT gates. To achieve (2), measurements are performed in a multiple fashion. While it is not possible to copy a value before measuring, it is possible to form a three-qubit state, similar to the three-qubit bit-flip encoding (Section 3.1), where all of the qubits should measure to the same value; if one of the measurements differs, it is assumed to be in error. These impacts are explained in detail in later examples.

Any logical operator may be applied as a fault tolerant procedure, as long as the probability,  $p$ , of an error for a physical operator is below a certain threshold,  $1/c$ , where  $c$  is determined by the implementation of the error correction code. For the Steane  $[[7, 1, 3]]$  code,  $c$  is about  $10^4$ . The overall probability of error for the logical operator is  $cp^2$ . That is, at some step in the application of the operator, and subsequent error correction, two errors would have to occur in order for the logical operator to fail.

## 3.3 Recursive Error Correction

A very simple construction allows us to tolerate additional errors. If a logical qubit is encoded in a block of  $n$  qubits, it is possible to encode each of those  $n$  qubits with an  $m$ -qubit code to produce an  $mn$  encoding. Such recursion, or *concatenation*, of codes can re-

<sup>4</sup>The overscore denotes an operator on a logical qubit: a logical operator.



**Figure 5: Tree structure of concatenated codes**

duce the overall probability of error even further. For example, concatenating the  $[[7, 1, 3]]$  with itself gives a  $[[49, 1, 7]]$  code with an overall probability of error of  $c(cp^2)^2$  (see Figure 5). Concatenating it  $k - 1$  times gives  $(cp)^{2^k}/c$ , while the size of the circuit increases by  $d^k$  and the time complexity increases by  $t^k$ , where  $d$  is the increase in circuit complexity for a single encoding, and  $t$  is the increase in operation time for a single encoding. For a circuit of size  $p(n)$ , to achieve a desired probability of success of  $1 - \epsilon$ ,  $k$  must be chosen such that [21]:

$$\frac{(cp)^{2^k}}{c} \leq \frac{\epsilon}{p(n)}$$

The number of operators required to achieve this result is

$$O(\text{poly}(\log p(n)/\epsilon)p(n)).$$

#### 4. TECHNOLOGY MODEL

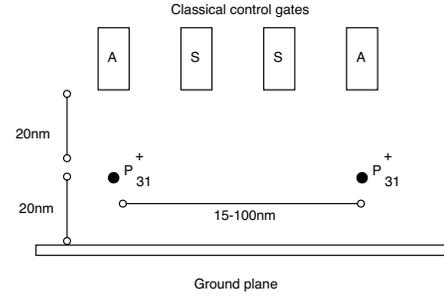
With some basics of quantum operations in mind, we turn our attention to the technologies available to implement these operations. Experimentalists have examined several technologies for quantum computation, including trapped ions [19], photons [32], bulk spin NMR [34], Josephson junctions [20, 36], electron spin resonance transistors [38], and phosphorus nuclei in silicon (the ‘‘Kane’’ model) [16] [29]. The last three of these proposals, which are built on a solid-state silicon substrate, share the following key aspects:

- 1 Qubits are laid out in silicon in a 2-D fashion, similar to traditional CMOS VLSI.
- 2 Quantum interactions are near-neighbor between qubits.
- 3 Qubits are stored at fixed locations, but quantum data may be swapped between nearest neighbors.
- 4 The control structures necessary to manipulate the bits prevent a dense 2-D grid of bits. Instead, we have linear structures of bits that can cross, but that have a minimum distance between such intersections [22]. This restriction is similar to a ‘‘design rule’’ in traditional CMOS VLSI.

These four assumptions apply to several solid-state technologies, but for concreteness, we will focus upon an updated version of Kane’s phosphorus-in-silicon nuclear-spin proposal [29]. This scheme will serve as an example for the remainder of the paper, although we will generalize our results when appropriate.

Figure 6 illustrates the Kane scheme. Quantum states are stored in relatively stable electron-donor ( $e^- - {}^{31}\text{P}^+$ ) spin pairs, where the electron ( $e$ ) and the phosphorous donor nucleus ( $n$ ) have opposite spins. The basis states,  $|0\rangle$  and  $|1\rangle$  are defined as the phase difference  $|0\rangle \equiv |\uparrow_e \downarrow_n\rangle + |\downarrow_e \uparrow_n\rangle$  and  $|1\rangle \equiv |\uparrow_e \downarrow_n\rangle - |\downarrow_e \uparrow_n\rangle$ , respectively. Twenty nanometers above the phosphorus atoms lie three classical gates, one  $A$  gate and two  $S$  gates. Precisely timed pulses on these gates provide arbitrary one- and two-qubit quantum gates.

Single qubit operators are composed of pulses on the  $A$ -gates, modulating the hyperfine interaction between the electron and nucleus to provide rotations around the  $\hat{z}$ -axis. A globally applied,



**Figure 6: The basic quantum bit technology proposed by Kane. Qubits are embodied by the nuclear spin of a phosphorus atom coupled with an electron embedded in silicon under high magnetic field at low temperature.**

static magnetic field provides rotations around the  $\hat{x}$ -axis. By changing the pulse widths, any desired rotational operator may be applied, including the identity operator<sup>5</sup>. Two-qubit interactions are mediated by  $S$ -gates, which move an electron from one nucleus to the next. Exact details of the pulses and quantum mechanics of this technique are beyond the scope of this paper and are described in [29].

The Kane proposal, like all quantum computing proposals, uses classical signals to control the timing and sequence of operations. All known quantum algorithms, including basic error correction for quantum data, require the determinism and reliability of classical control. Without efficient classical control, fundamental results demonstrating the feasibility of quantum computation do not apply (such as the Threshold Theorem used in Section 3).

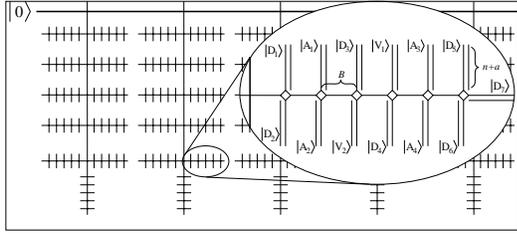
The scale required by the Kane model, on the other hand, is at odds with efficient classical control. In order to provide the fine-grained control necessary, the control lines need to operate in a classical manner. That is, there need to be enough quantum states in the control lines so that electron movement is bulk, not ballistic, and voltage transitions are smooth rather than stair-stepped. Because of this, the control lines need to be physically much larger than the qubits they are controlling [22]. Conceptually, the control lines need to be of *classical* size and pitch, and packed closely to control quantum bits placed on a *quantum* scale. This imposes a constraint that qubits be laid out in straight lines, with a certain minimum number of qubits between junctions.

Given the constraint of linearity with infrequent junctions, there are several ways to lay out physical and logical qubits. Optimally, qubits should be arranged to minimize communication overhead.

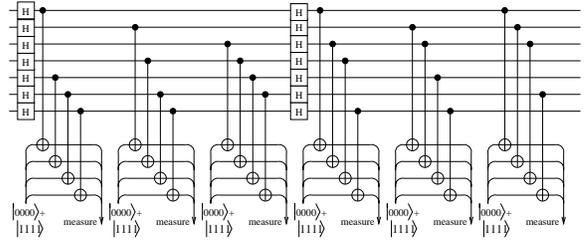
In a fault tolerant design, the main activity of a quantum computer is error correction. To minimize communication costs, qubits in an encoding block should be in close proximity. Assuming that the distance between junctions is greater than the number of qubits in an encoding, the closest the qubits can be is in a straight line. But in order to avoid interacting two qubits in an encoding with a third, a two-rail approach is used— one rail for data qubits, and one for communication.

A concatenated code requires a slightly different layout (see Figure 7). Error correction is still the important operation, but the logical qubits at all but the bottom level of the code are more complicated. For the second level, the qubits are themselves simple encodings, laid out using the two-rail construction. However, to minimize communication costs, we want these logical qubits in as close proximity to each other as possible, just like the bottom level.

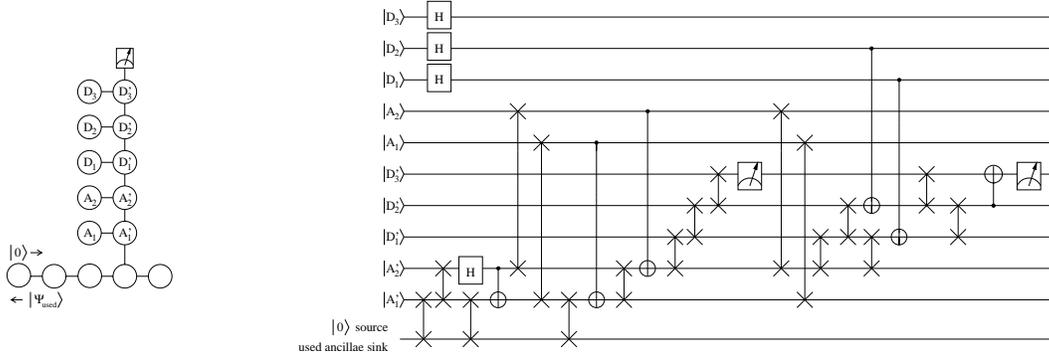
<sup>5</sup>One impact of the external magnetic field is the state of the qubit is in constant flux. The identity operator must be applied on every ‘‘cycle’’ in order to keep the current state.



**Figure 7: Schematic layout of the H-tree structure of a concatenated code. The branches in the inset represent the singly-encoded qubits. The  $|D_i\rangle$  are data qubits, and the  $|A_i\rangle$  are ancillae. The  $|V_i\rangle$  are for verification.**



**Figure 8: Measuring the error syndrome for the  $[[7, 1, 3]]$  error-correction code.**



**Figure 9: "Two-rail" layout for the three-qubit phase-correction code. The schematic on the left shows qubit placement and communication, where  $D$ 's indicate data qubits,  $A$ 's are ancillae, and  $D'$ 's and  $A'$ 's are for communication. The open qubits at the bottom swap in fresh ancillae, and remove used ancillae. The same layout is shown as a quantum circuit on the right, with the operations required to create and verify an ancillary cat state, and to measure the parity of a pair of data qubits.**

Hence, we need to arrange the bottom level as branches coming off of a main bus. Similarly, the third level would have second-level branches coming off of a main trunk, and so on for higher levels, forming an  $H$ -tree.

## 5. ERROR CORRECTION ALGORITHMS

We've discussed error correction in a general sense, and how the need for recursive error correction influences the architectural design. In addition, we have introduced several error-correction codes, such as Shor's 3-qubit phase-flip code, Shor's 9-qubit code, and Steane's 7-qubit code. The constructions in Figures 4 and 9 deal with the simplest of these codes, the 3-qubit code, which only corrects phase flips. In order to correct both bit and phase flips, a more complicated code is needed. For the remainder of this paper, we will focus on the 7-qubit code,  $[[7, 1, 3]]$ , which corrects up to a single error, and recursive codes based on  $[[7, 1, 3]]$  which can correct many errors. We choose  $[[7, 1, 3]]$  because of the ease with which logical operators may be applied. In particular, remember that the logical operators  $\bar{X}$ ,  $\bar{Z}$ ,  $\bar{H}$ , and  $\overline{\text{CNOT}}$  are applied by applying the simple operator to each qubit in the encoding block.

### 5.1 The $[[7, 1, 3]]$ Code

Error correcting using the  $[[7, 1, 3]]$  code consists of measuring the parity of the encoding qubits in various bases. As shown in Figure 8, the qubits are rotated to the measurement basis with Hadamard gates. Parity is then measured in much the same way it is on a classical code, using two-qubit CNOT operators acting as XOR's. Conceptually, the parity can be measured in the same way as the three-qubit code in Section 3.1, gathering the parity on ancilla  $|0\rangle$ 's. To perform a fault tolerant measurement, however, a cat state is

used in place of a  $|0\rangle$ . Figure 8 shows a schematic for measuring the  $[[7, 1, 3]]$  code. Not shown are cat-state creation and cat-state verification. In addition, each parity measurement must be performed twice to reduce the probability of an error from  $O(p)$  to  $O(p^2)$ ; if the measurements disagree, the parity must be measured a third time!

A parity measurement consists of the following:

- 1 Prepare a cat state from four ancillae, using a Hadamard gate and three CNOT gates.
  - 2 Verify the cat state by taking the parity of each pair of qubits. If any pair has odd parity, return to step 1. (Note that this requires six additional ancillae, one for each pair.)
  - 3 Use the four-ancillae cat state as the CNOT target of the data qubits whose parity is to be measured.
  - 4 Deconstruct the cat state by selecting one of the ancillae,  $|A_0\rangle$ , and using it as the CNOT target of the remaining three ancillae.  $|A_0\rangle$  now has the overall parity of the cat state.
  - 5 Measure this  $|A_0\rangle$ :
    - A With  $|A_0\rangle = \alpha|0\rangle + \beta|1\rangle$ , create the three-qubit state,  $\alpha|000\rangle + \beta|111\rangle$  by using  $|A_0\rangle$  as the control for two CNOT gates, and two fresh  $|0\rangle$  ancillae as the targets.
    - B Measure each of the three qubits.
  - 6 Use the majority measured value as the parity of the cat state.
- The resulting syndrome determines which, if any, qubit has an error, and which  $X$ ,  $Z$ , or  $Y$  operator will correct the error.

For the Steane  $[[7, 1, 3]]$  code, each parity measurement requires twelve ancillae—four for the cat state to capture the parity, six to verify the cat state, and two additional qubits to measure the cat state. The six parity measurements are each performed at least twice, for a minimum of 144 ancillae to measure the error syndrome! A less complex example is shown in Figure 9.

Each of the twelve parity measurements require:

- One Hadamard and three CNOT's to create the cat state;
- Twelve CNOT's to verify the cat state;
- Four CNOT's, which can be applied in parallel, to collect the parity of the data qubits;
- Three CNOT's and a Hadamard to uncreate the cat state;
- Two CNOT's to create the three-qubit state for measurement; and
- Three qubit measurements, which may be performed in parallel with the next parity measurement.

If the time required to apply a single-qubit operator is  $S$ , a CNOT is  $C$ , and a measurement is  $M$ , then the minimum time required to measure the error syndrome is  $2S + 12(2S + 24C)$ .

## 5.2 Concatenated Codes

The  $[[7, 1, 3]] \times [[7, 1, 3]]$  two-level concatenated code is measured in the same way as the  $[[7, 1, 3]]$  code, except the qubits and ancillae are encoded. For example, each logical ancilla must be prepared in the following manner<sup>6</sup>

- 1 Begin with seven ancillae.
- 2 Measure the error syndrome, and correct, as in Section 5.1. At this point, the seven qubits constitute a valid code word.
- 3 Measure the value of the logical ancilla:
  - A Create a cat state with another seven ancillae to collect the parity of the seven qubits in the logical ancilla.
  - B Verify the cat state.
  - C Use the cat-state qubits as the CNOT target of the qubits encoding the logical ancilla.
  - D Uncreate the cat-state, collecting the parity into a single qubit.
  - E With two fresh ancillae, create  $\alpha|000\rangle + \beta|111\rangle$
  - F Measure each of these three qubits.
- 4 Use the majority measured value as the value of the logical ancilla.
- 5 If the measurement is  $|1_L\rangle$ , apply  $\bar{X}$ .

The error syndrome measurement is analogous to the singly-encoded  $[[7, 1, 3]]$  case, except that the lower-level encodings must be error corrected between operations:

- 1 Prepare four logical ancillae in a cat state.
- 2 Error correct the four ancilla.
- 3 Verify the cat state.
- 4 Use the ancillae as the  $\overline{\text{CNOT}}$  target of the qubits whose parity is to be measured.
- 5 Error correct the four qubits in the cat state and the logical data qubits.
- 6 Measure each of the four logical cat-state qubits. The parity of these measurements is the parity of the four encoding qubits. This step is equivalent to the cat-state deconstruction step for the singly-encoded case.

As in the singly-encoded case, each parity measurement must be performed at least twice. The resulting syndrome determines which, if any, logical qubit has an error. The appropriate  $\bar{X}$ ,  $\bar{Z}$ , or  $\bar{Y}$  operator can be applied to correct the error. Of course, after the operator is applied to a logical qubit, that qubit must be error-corrected.

Higher levels are error-corrected analogously.

## 6. COMMUNICATION COSTS

In this section, we derive the primary results of this paper. First, we model the communication costs of our error correction algorithms under the near neighbor constraint. We show that there are

<sup>6</sup>Fault-tolerant algorithms that avoid the overhead of encoded ancilla are a topic of future research.

too many SWAP operations between upper levels of our tree structures and that too much error accumulates to be corrected. Second, we analyze quantum teleportation as an alternative to SWAP operations for long-distance communication. Finally, we show that teleportation is necessary both in terms of distance and in terms of the accumulating probability of correlated errors between redundant qubits in our code words.

### 6.1 Error Correction Costs

The error correction algorithms in the previous section are in an ideal situation, where any qubit can interact with any other qubit. Usually, qubits can only interact with their near neighbors, so before applying a two-qubit operator, one of the operand qubits must be moved adjacent to the other.

One of the easiest ways to move quantum data is to use the SWAP operator. By applying SWAP's between alternating pairs of qubits, the values of alternating qubits are propagated in one direction, while the remaining qubit values are propagated in the reverse direction. This can be used to supply  $|0\rangle$  ancillae for the purpose of error correction. As a side benefit, this also removes "used" ancillae. Figure 9 illustrates this method for the three-qubit example, using two rows of qubits, one for the encoding data qubits and one for the ancillae.

The same method can be applied to the  $[[7, 1, 3]]$  code. The actual communication costs depend on the physical implementation used. The time required for an error correction parity check is

$$t_{ecc} = 12(t_{cc} + t_{cv} + t_p + t_{cd} + t_m) \quad (1)$$

where

$t_{cc}$  is the time for cat state creation;

$t_{cv}$  is the time for cat state verification;

$t_p$  is the time to entangle the cat state with the parity qubits;

$t_{cd}$  is the time to uncreate the cat state; and

$t_m$  is the time to perform a fault-tolerant measurement.

For  $[[7, 1, 3]]$  in the ideal, sea-of-qubits model,  $t_{cc}$  is  $t_{single} + 3t_{cnot}$ ,  $t_{cv}$  is  $6(2t_{cnot} + t_{meas})$ ,  $t_p$  is  $t_{cnot}$ —four CNOT's performed in parallel,  $t_{cd}$  is  $3t_{cnot} + t_{single}$  and  $t_{overlap}$  is  $t_{decat} + t_{meas}$ , where

$t_{single}$  is the time required for a single-qubit operator;

$t_{cnot}$  is the time required for a CNOT operator;

$t_{swap}$  is the time required for a SWAP operator; and

$t_{meas}$  is the time required for the measurement operator.

If communication by swapping is used,

$$t_{cc} = \max(t_{single}, t_{swap}) + 4t_{swap} + 3 \max(t_{cnot}, t_{swap}), \quad (2)$$

$$t_{cv} = \max(t_{single}, t_{swap}) + 18t_{swap} + 12 \max(t_{cnot}, t_{swap}), \quad (3)$$

$$t_p \leq 4 \max(t_{cnot}, t_{swap}), \quad \text{and} \quad (4)$$

$$t_{cd} \leq 3t_{swap} + 2 \max(t_{cnot}, t_{swap}) + t_{single}. \quad (5)$$

In the Kane model,  $t_{single} < t_{swap} < t_{cnot}$ , so the overall cost is

$$t_{ecc} \leq 336t_{swap} + 168t_{cnot} + t_{meas}.$$

Since measurement is fully parallelizable, these times assume that there are enough measurement functional units to perform measurement in parallel with the other operations in the error-correction cycle.

### 6.2 Multilevel Error Correction

For the concatenated code, the data movement in the upper levels is more complicated. Although Eq. 1 still holds, each parity

$k$	SWAP	CNOT	1-Qubit	Measurement
1	1620	288	12	108
2	690,000	120,000	4800	43,000
3	$2.7 \times 10^8$	$4.7 \times 10^7$	$1.9 \times 10^6$	$1.7 \times 10^7$
4	$1.1 \times 10^{11}$	$1.8 \times 10^{10}$	$7.5 \times 10^8$	$6.7 \times 10^9$
5	$4.3 \times 10^{13}$	$7.3 \times 10^{12}$	$3.0 \times 10^{11}$	$2.7 \times 10^{12}$

**Table 2: Operations required for an error-correction cycle at level  $k$ .**

measurement requires the following. Since ancillae are themselves encoded, they each require their own branch, and the first step is to create the encoded ancillae, by error correcting, measuring, and if necessary, inverting. The second step is to create the four-qubit cat state from the logical ancillae, by applying  $\bar{H}$  to one of the ancilla, moving a second ancillae through the main branch to the second rail of the first ancilla, applying  $\bar{CNOT}$ , moving the second ancilla back, and error-correcting both ancillae. This is repeated for the second and third ancillae, and the third and fourth ancillae. Since the ancillae are error corrected along the way, the cat state need not be verified.

Next, an ancillae is moved through the main branch to the data branch that holds a bottom-level encoding. After applying  $\bar{CNOT}$ , the ancilla is moved back to its own branch, and both it and the logical data qubit are error-corrected. Since measuring the ancillae can be performed completely in parallel, all four ancillae are measured, and the parity of the measurements is the parity of the data qubits.

For  $[[7, 1, 3]]$  concatenated with itself  $k$  times,

$$t_{anc,k} = t_{ecc,k-1} + t_{m,k-1}; \quad (6)$$

$$t_{cc,k} = 6t_{ecc,k-1} + 6t_{b,k}; \quad (7)$$

$$t_{cv,k} = 18t_{ecc,k-1} + 28t_{b,k}; \quad (8)$$

$$t_{p,k} = 8t_{ecc,k-1} + 10t_{b,k}; \quad (9)$$

$$t_{cd,k} = 4t_{m,k-1} \quad (10)$$

$$t_{m,k} = 4^{k-1}t_{m,1}; \text{ and} \quad (11)$$

$$t_{b,k} = \begin{cases} 1, & k = 1 \\ t_{B,arch}, & k = 2 \\ (n+a)t_{b,k-2} + t_{B,arch}, & k > 2 \end{cases} \quad (12)$$

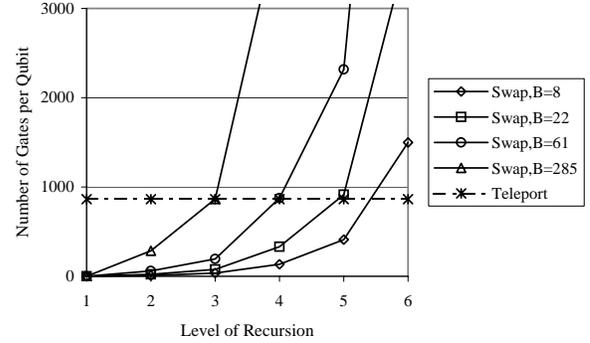
where the subscript  $k$  indicates the level of encoding,  $t_{anc,k}$  is the cost of encoding an ancilla,  $t_{b,k}$  is the branch distance between logical qubits at level  $k$ ,  $t_{m,1}$  is the time required to measure a singly-encoded qubit,  $t_{B,arch}$  is the minimum number of qubits between two branches for a given architectural model,  $n$  is the number of physical qubits in the non-concatenated code and  $a$  is the number of ancillae per parity measurement. For concatenated codes, parallel operation is determined by the ratio of ancillae delivery to ancillae consumption for a singly-encoded parity check. For  $[[7, 1, 3]]$  and a single-qubit-wide branch this ratio is around 3. Arranging the ancillae as in the inset of Figure 7 minimizes the distance that ancillae must travel.

The recurrence relation given in Eqs. 6 through 12 give an overall time to perform an error-correction cycle at a given level of recursion. A similar recurrence relation gives the total number of operations required. The number of operators required for different levels of encoding are summarized in Table 2, which shows that the SWAP operator is very important in a realistic model, compared to the sea-of-qubits model, where SWAP's are not required. In this realistic model, SWAP's account for over 80% of all operations.

$k$	Teleportation	Swapping, $t_{B,arch} = 22$	Swapping, $t_{B,arch} = 61$	Swapping, $t_{B,arch} = 285$
1	864	1	1	1
2	864	22	61	285
3	864	77	194	866
4	864	330	876	4,012
5	864	913	2,317	10,381
6	864	3,696	9,819	44,987

**Table 3: Comparison of the cost of swapping an encoded qubit to the cost of teleporting it. The “swapping” values are  $b_k$ , the distance between adjacent qubits.**

**Swapping and Teleportation**



**Figure 10: Cost of teleportation compared to swapping. The values chosen illustrate the break-even point for different levels of recursion.**

### 6.3 Teleportation

Fortunately, we can use quantum teleportation as an alternative to swapping for communication over long distances. To use teleportation for our circuit, we must evaluate the number of computation and communication operations within the teleportation circuit. By comparing this number of operations with the swapping costs from the previous section, we can decide at what level  $k$  of the tree to start using teleportation instead of swapping for communication.

### 6.4 Distance Tradeoff

By calculating the number of basic computation and communication operations necessary to use teleportation for long-distance communication, we can quantify when we should switch from swapping to teleportation in our tree structure. Figure 10 illustrates this tradeoff. We can see that for  $t_{B,arch} = 22$ , teleportation should be used when  $k \geq 5$ .

### 6.5 Avoiding Correlated Errors

An important assumption in quantum error correction is that errors in the redundant qubits of a codeword are uncorrelated. That is, we do not want one error in a codeword to make a second error more likely. To avoid such correlation, it is important to try not to interact qubits in a codeword with each other.

Unfortunately, we find that a 2D layout cannot avoid indirect interaction of qubits in a codeword. At some point, all the qubits in a codeword must be brought to the same physical location in order to calculate error syndromes. In order to do this, they must pass through the same line of physical locations. Although we can avoid swapping the codeword qubits with each other, we cannot avoid swapping them with some of the same qubits that flow in the other direction.

For concreteness, if two qubits of codeword  $d_0$  and  $d_1$  both swap with an ancilla  $a_0$  going in the opposite direction, there is some probability that  $d_0$  and  $d_1$  will become correlated with each other through the ancilla. This occurs if both SWAPs experience a partial failure. In general, if  $p$  is the probability of a failure of a SWAP gate, the probability of an error from swapping a logical qubit is

$$n^k b_k p + \binom{n^k}{2} b_k p^2 + \binom{n^k}{3} b_k p^3 + \dots,$$

where  $b_k$  is the number of qubits between branches at level  $k$ , and the higher order terms are due to correlation between the qubits. From this form, it is clear that correlated errors are dominated by uncorrelated errors, when  $n^k p \ll 1$ .

## 7. FUTURE WORK

Our results have interesting implications for the Threshold Theorem, effectively increasing the reliability requirements for quantum operators, particularly SWAP operators. In addition to the teleportation solution to long-distance communication, it may be possible to modify the straightforward recursive structure used in quantum error correction codes to include intermediate error correction steps in the middle of long chains of SWAP operators. There are, however, serious challenges of getting the ancillae to all of these intermediate points in such a layout.

At the lowest level, the largest consumer of ancillae for error correction is cat-state verification. However, at higher levels, the cat states themselves are constructed from logical ancillae, each of which must be error corrected, measured, and the whole cat state verified. This approach is a straightforward analog to the lowest level, but there may be more efficient algorithms from the standpoint of ancilla use.

The proposed teleportation solution assumes that the distribution of reliable EPR pairs is significantly easier than transporting arbitrary quantum data. EPR pairs are precommunicated in a pipelined fashion, then “purified” using an entanglement-concentrating algorithm that eliminates bad EPR pairs [25]. Quantifying the reliability and bandwidth of this mechanism is the subject of future study.

Finally, this paper has focused on solid-state implementations with static qubits. There is a proposal for scalable ion-trap quantum computers, built using conventional microfabrication techniques, where the qubits are mobile [17]. How the mobility constraints of such a system compare to swapping with static qubits is a subject of future study.

## 8. CONCLUSION

Quantum computation is in its infancy, but now is the time to evaluate quantum algorithms under realistic constraints and derive the architectural mechanisms and reliability targets that are needed in order to scale quantum computers to their full potential. This paper has focused upon the spatial and temporal constraints of solid-state technologies, and has shown that the recursive construction for quantum error correction codes requires a long-distance communication technology such as quantum teleportation. We derived the tradeoff point between short- and long-distance technologies. Also, the reliability of the quantum SWAP operation used in short-distance communication is the dominant factor in system reliability. These results are a beginning. The next step is moving quantum computation from theory to practice, unlocking an unprecedented tool to attack difficult problems.

## 9. ACKNOWLEDGMENTS

Thanks to John Owens, Matt Farrens, Paul Sultana, and Mark Whitney for their helpful comments on preliminary material for this paper. This work is supported in part by the DARPA Quantum Information, Science and Technology Program, by NSF CAREER grants to Mark Oskin and Fred Chong, an NSF NER grant and a UC Davis Chancellor’s Fellowship to Fred Chong.

## 10. REFERENCES

- [1] D. Aharonov and M. Ben-Or. Fault tolerant computation with constant error. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pages 176–188, 1997.
- [2] J. S. Bell. On the Einstein-Podolsky-Rosen paradox. *Physics*, 1:195–200, 1964. Reprinted in J. S. Bell, *Speakable and Unsayable in Quantum Mechanics*, Cambridge University Press, Cambridge, 1987.
- [3] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, 1984.
- [4] D. Bouwmeester, J. W. Pan, K. Mattle, M. Eibl, H. Weinfurter, and A. Zeilinger. Experimental quantum teleportation. *Nature*, 390(6660):575–579, 1997.
- [5] A. M. Childs, E. Farhi, and J. Preskill. Robustness of adiabatic quantum computation. *Phys. Rev. A*, (65), 2002.
- [6] I. L. Chuang. Quantum algorithm for clock synchronization. *Phys. Rev. Lett.*, 85:2006, Aug 2000.
- [7] I. L. Chuang, N. Gershenfeld, and M. Kubinec. Experimental implementation of fast quantum searching. *Phys. Rev. Lett.*, 18(15):3408–3411, 1998.
- [8] I. L. Chuang, R. Laflamme, P. Shor, and W. H. Zurek. Quantum computers, factoring, and decoherence. *Science*, 270:1633, Dec 1995. arXiv e-print quant-ph/9503007.
- [9] N. Gershenfeld and I. Chuang. Quantum computing with molecules. *Scientific American*, June 1998.
- [10] D. Gottesman. Theory of fault-tolerant quantum computation. *Phys. Rev. A*, 57(1):127–137, 1998. arXiv e-print quant-ph/9702029.
- [11] L. Grover. In *Proc. 28<sup>th</sup> Annual ACM Symposium on the Theory of Computation*, pages 212–219, New York, 1996. ACM Press.
- [12] S. Hallgren. *Quantum Information Processing ’02 Workshop*, 2002.
- [13] P. Hoyer. *Banff workshop on quantum algorithms*, 2002.
- [14] J. A. Jones, M. Mosca, and R. H. Hansen. Implementation of a quantum search algorithm on a nuclear magnetic resonance quantum computer. *Nature*, 393(6683):344, 1998. arXiv e-print quant-ph/9805069.
- [15] R. Jozsa, D. Abrams, J. Dowling, and C. Williams. Quantum atomic clock synchronization based on shared prior entanglement. *Phys. Rev. Lett.*, pages 2010–2013, August 2000.
- [16] B. Kane. A silicon-based nuclear spin quantum computer. *Nature*, 393:133–137, 1998.
- [17] D. Kielpinski, C. Monroe, and D. J. Wineland. Architecture for a large-scale ion-trap quantum computer. *Nature*, 417:709–711, 2002.

- [18] S. Lloyd. Quantum-mechanical computers. *Scientific American*, 273(4):44, Oct. 1995.
- [19] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland. Demonstration of a fundamental quantum logic gate. *Phys. Rev. Lett.*, 75:4714, 1995.
- [20] Y. Nakamura, Y. A. Pashkin, and J. S. Tsai. Coherent control of macroscopic quantum states in a single-cooper-pair box. *Nature*, 398:786–788, 1999.
- [21] M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge University Press, Cambridge, England, 2000.
- [22] M. Oskin, F. Chong, I. Chuang, and J. Kubiawicz. Building quantum wires: The long and the short of it. In *Proc. International Symposium on Computer Architecture (ISCA 2003)*, New York, 2003. ACM Press.
- [23] J. Preskill. Reliable quantum computers. *Proc. R. Soc. London A*, 454(1969):385–410, 1998.
- [24] C. Sackett, D. Kielpinsky, B. King, C. Langer, V. Meyer, C. Myatt, M. Rowe, Q. Turchette, W. Itano, D. Wineland, and C. Monroe. Experimental entanglement of four particles. *Nature*, 404:256–258, 2000.
- [25] L. Schulman and U. Vazirani. Molecular scale heat engines and scalable quantum computation. In *31st STOC*, 1999.
- [26] P. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proc. 35<sup>th</sup> Annual Symposium on Foundations of Computer Science*, page 124, Los Alamitos, CA, 1994. IEEE Press.
- [27] P. Shor. Fault-tolerant quantum computation. In *37th FOCS*, 1994.
- [28] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 54:2493, 1995.
- [29] A. Skinner et al. Hydrogenic spin quantum computing in silicon: a digital approach. *quant-ph/0206159*, 2002.
- [30] A. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, 77, 1996.
- [31] A. Steane. Simple quantum error correcting codes. *Phys. Rev. Lett.*, 77:793–797, 1996.
- [32] Q. A. Turchette, C. J. Hood, W. Lange, H. Mabuchi, and H. J. Kimble. Measurement of conditional phase shifts for quantum logic. *Phys. Rev. Lett.*, 75:4710, 1995.
- [33] W. van Dam and G. Seroussi. Efficient quantum algorithms for estimating gauss sums. *quant-ph*, page 0207131, 2002.
- [34] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, R. Cleve, and I. L. Chuang. Experimental realization of order-finding with a quantum computer. *Phys. Rev. Lett.*, to appear, 2000.
- [35] L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang. Experimental realization of shor’s quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414:883, 2001.
- [36] D. Vion, A. Aassime, A. Cottet, P. Joyez, H. Pothier, C. Urbina, D. Esteve, and M. H. Devoret. Manipulating the quantum state of an electrical circuit. *Science*, 296:886, 2002.
- [37] J. von Neuman. *Automata Studies*. Princeton University Press, Princeton, NJ, 1956.
- [38] R. Vrijen, E. Yablonovitch, K. Wang, H. W. Jiang, A. Balandin, V. Roychowdhury, T. Mor, and D. DiVincenzo. Electron spin resonance transistors for quantum computing in silicon-germanium heterostructures. *arXiv e-print quant-ph/9905096*, 1999.
- [39] S. Winograd and J. D. Cowan. *Reliable Computation in the Presence of Noise*. MIT Press, Cambridge, Mass., 1963.