Quantum computing with realistically noisy devices

E. Knill

Mathematical and Computational Sciences Division, National Institute of Standards and Technology, Boulder, Colorado 80305, USA

In theory, quantum computers offer a means of solving problems that would be intractable on conventional computers. Assuming that a quantum computer could be constructed, it would in practice be required to function with noisy devices called 'gates'. These gates cause decoherence of the fragile quantum states that are central to the computer's operation. The goal of so-called 'fault-tolerant quantum computing' is therefore to compute accurately even when the error probability per gate (EPG) is high. Here we report a simple architecture for fault-tolerant quantum computing, providing evidence that accurate quantum computing is possible for EPGs as high as three per cent. Such EPGs have been experimentally demonstrated, but to avoid excessive resource overheads required by the necessary architecture, lower EPGs are needed. Assuming the availability of quantum resources comparable to the digital resources available in today's computers, we show that non-trivial quantum computations at EPGs of as high as one per cent could be implemented.

Research in quantum computing is motivated by the great increase in computational power offered by quantum computers[1–3]. There is a large and still-growing number of experimental efforts whose ultimate goal is to demonstrate scalable quantum computing. Scalable quantum computing requires that arbitrarily large computations can be implemented efficiently with little error in the output. Criteria that need to be satisfied by devices used for scalable quantum computing have been specified[4]. An important one is that the level of noise affecting the physical gates and memory is sufficiently low. The type of noise affecting a given implementation is the 'error model'. A scheme for scalable quantum computing in the presence of noise is a 'fault-tolerant architecture'. In view of the low-noise criterion, studies of scalable quantum computing involve constructing fault-tolerant architectures and providing answers to questions such as the following: is scalable quantum computing possible for error model $\mathcal{E}$? Can fault-tolerant architecture $\mathcal{A}$ be used for scalable quantum computing with error model $\mathcal{E}$? What resources are required to implement quantum computation $\mathcal{C}$ using fault-tolerant architecture $\mathcal{A}$ with error model $\mathcal{E}$?

To obtain broadly applicable results, fault-tolerant architectures are constructed for generic error models. Here, the error model is parametrized by an error probability per gate (or simply error per gate, EPG), where the errors are unbiased and independent. The fundamental theorem of scalable quantum computing is the threshold theorem and answers the first question as follows: if the EPG is smaller than a threshold, then scalable quantum computing is possible[5–8]. Thresholds depend on additional assumptions for the error model and device capabilities. Estimated thresholds vary from below 10^{-6} (refs 5–8) to 3×10^{-3} (ref. 9), with 10^{-4} (ref. 10) often quoted as the EPG to be achieved in experimental quantum computing.

In the few cases where experiments with two quantum bits (qubits) have been performed, the EPGs currently achieved are much higher, 3×10^{-2} or more in ion traps[11,12] and liquid-state nuclear magnetic resonance[13,14]. For quantum computing to become practical, it is essential to reduce the large gap between the experimentally achieved EPGs and those required by theory. The first goal of our work is to provide evidence that scalable quantum computing is possible at EPGs above 3×10^{-2}. This is encouraging, but the fault-tolerant architecture that achieves this is impractical because of its large resource requirements. To reduce the resource requirements, lower EPGs are required. The second goal of our work is to give a simple fault-tolerant architecture (called the 'C_4/C_6 architecture') well-suited to efficient computing with EPGs between 10^{-4} and 10^{-2}. The third goal is to provide a means of estimating its resource requirements depending on computation size and EPG.

Fault-tolerant architectures realize low-error qubits and gates by encoding them with error-correcting codes. A standard technique for reducing errors is concatenation. Suppose we have a scheme that, starting with qubits and gates at one EPG, produces encoded qubits and gates that have a lower EPG. If the error model for encoded gates is sufficiently well-behaved, we can apply the same scheme to the encoded qubits and gates to obtain a next level of encoded qubits and gates with much lower EPGs. This process yields a hierarchy of repeatedly encoded qubits and gates, where the physical qubits and gates are at level 0. The top level is used for quantum computing. Its qubits, gates, EPGs and so on are 'logical'.

The C_4/C_6 architecture differs from previous ones by combining a number of independently useful techniques. First, we use the simplest error-detecting codes, thus avoiding the complexity of even the smallest error-correcting codes. Error correction is added naturally by concatenation. Second, error correction is performed in one step and combined with logical gates by means of error-correcting teleportation. This minimizes the number of gates contributing to errors before they are corrected. Third, the architecture bootstraps key gates by state preparation and purification, thus enabling us to define it using a minimal and incomplete set of operations with only one unitary gate. Fourth, verification of the needed ancillary states (logical Bell pairs) largely avoids the traditional syndrome-based schemes. Instead, we use hierarchical teleportations and partial decoding. Finally, the highest thresholds are obtained by introducing the model of postselected computing with its own thresholds, which may be higher than those for standard quantum computing. Our fault-tolerant implementation of postselected computing has the property that it can be used to prepare states that are sufficient for standard scalable quantum computing.

Error model and assumptions

The unit of quantum information is the qubit, a quantum two-level system whose states are superpositions $\alpha|0\rangle + \beta|1\rangle$ (ref. 15). Qubits are acted on by the Pauli operators $X = \sigma_x$ (bit flip), $Z = \sigma_z$ (sign flip) and $Y = \sigma_y = i\sigma_x\sigma_z$. The identity operator is I. One-qubit gates include preparation of $|0\rangle$ and $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, Z-measurement (distinguishing between $|0\rangle$ and $|1\rangle$), X-measurement (distinguishing between $|+\rangle$ and $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$), and the Hadamard gate (HAD, $\alpha|0\rangle + \beta|1\rangle \mapsto \alpha|+\rangle + \beta|-\rangle$). We use one unitary two-qubit gate, the controlled-NOT (CNOT), which maps $|00\rangle \mapsto |00\rangle$, $|01\rangle \mapsto |01\rangle$, $|10\rangle \mapsto |11\rangle$ and $|11\rangle \mapsto |10\rangle$. This set of gates is a subset of the so-called Clifford gates, which are insufficient for universal quantum computing[10]. Our minimal gate set $\mathcal{G}_{\mathrm{min}}$ consists of $|0\rangle$ and $|+\rangle$ preparation, Z- and X-measurement and CNOT. Universality may be achieved with the addition of other one-qubit preparations or measurements, as explained below. The physical gates mentioned are treated as being implemented in one 'step'; the actual implementations may be more complex.

All error models can be described by inserting errors acting as quantum operations (not necessarily unitary) after gates or before measurements. We assume that a gate's error consists of random, independent applications of products of Pauli operators with probabilities determined by the gate. It is at present too difficult to obtain a threshold that does not depend on the details of the probability distributions, so we assume unbiased, 'depolarizing' errors for each gate: $|0\rangle$ ($|+\rangle$) state preparation erroneously produces $|1\rangle$ ($|-\rangle$) with probability e_{p}. A binary (such as Z or X) measurement results in the wrong outcome with probability e_{m}. CNOT is followed by one of the 15 possible non-identity Pauli products acting on its qubits with probability $e_{\mathrm{c}}/15$ each. HAD is modified by one of the Pauli operators, each with probability $e_{\mathrm{h}}/3$. We further simplify by setting $e_{\mathrm{c}} = \gamma$, $e_{\mathrm{h}} = 4\gamma/5$ and $e_{\mathrm{m}} = e_{\mathrm{p}} = 4\gamma/15$. This choice is justified as follows: $4\gamma/5$ is the one-qubit marginal probability of error for the CNOT, which we expect to be an upper bound for all one-qubit gate errors. As for preparation errors, if they are much larger than $4\gamma/15$, then it is possible to purify prepared states using a CNOT. For example, we can prepare $|0\rangle$ twice, apply a CNOT from the first to the second and measure Z of the second. We then try again if the measurement outcome indicates $|1\rangle$, and otherwise use the first state. The probability of error is given by $4\gamma/15 + O(\gamma^2)$, assuming that CNOT error is as above and measurement and preparation errors are proportional to γ. To improve Z measurement, we can introduce an ancilla in $|0\rangle$, apply a CNOT from the qubit to be measured to the ancilla, and measure both qubits. If the measurements disagree, an error is detected. If not, the conditional measurement error probability is $4\gamma/15 + O(\gamma^2)$. Detected errors are readily managed[16].

To account for 'memory' errors, we assume that gates other than measurements take the same amount of time. Thus, the error parameters represent the total error, including any delays for faster gates to equalize gate times. For the C_4/C_6 architecture, memory is required when waiting for measurement outcomes that determine whether prepared states are good, or that are needed after teleportation, particularly when implementing non-Clifford gates[17,18]. The simplest way to account for memory errors in these situations is to distribute it equally to the surrounding gates. The maximum error thus distributed is the memory error e_{B} accumulated during the time required for a Bell measurement, here consisting of a CNOT followed by X and Z measurements. No gate is both preceded and followed by memory delays, so gate errors are adjusted by at most $e_{\mathrm{B}}/2$, which we assume is already taken into account in the errors given in the previous paragraph. To ensure that e_{B} is sufficiently small requires measurements that are fast compared to memory decoherence times. Systems such as those based on ion traps can achieve this with good qubit memories[12,19].

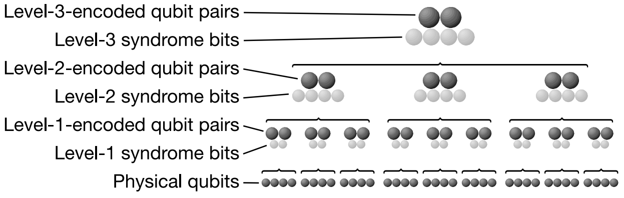
Level-3-encoded qubit pairs
Level-3 syndrome bits
Level-2-encoded qubit pairs
Level-2 syndrome bits
Level-1-encoded qubit pairs
Level-1 syndrome bits
Physical qubits

Figure 1 Block structure of C_4/C_6 concatenated codes. The bottom line shows nine blocks of four physical qubits. Each block encodes a level-1 qubit pair with C_4. The encoded qubit pairs are shown in the line above. Formally, each such pair is associated with two syndrome bits, shown below the encoded pair in a lighter shade, which are accessible by syndrome measurements or decoding for error detection and correction. The next level groups three level-1 qubit pairs into a block, encoding a level-2 qubit pair with C_6. The pair is associated with four syndrome bits. A level-2 block consists of 12 physical qubits. Three level-2 qubit pairs are used to form a level-3 qubit pair, again with C_6 and associated with four syndrome bits. The total number of physical qubits in a level-3 block is 36. In general, a level-l block has $4 \times 3^{l-1}$ physical qubits.

Two additional assumptions are used. The first is that there is no error and no speed constraint on classical computations required to interpret measurement outcomes and control future gates. The second is that two-qubit gates can be applied to any pair of qubits without delay or additional error. This assumption is unrealistic, but the effect on the threshold is due primarily to CNOTs acting within the ancillas needed for maintaining one or two blocks encoding logical qubits. To account for this, we can use higher effective EPGs or require low-error quantum communication[9,20,21].

The above assumptions are standard in analyses of fault-tolerant architectures, but idealized. They are nevertheless believed to be sufficiently realistic that results based on them are meaningful in practice[5,22–24].

The quantum codes

The C_4/C_6 architecture is based on concatenating two quantum stabilizer codes, C_4 and C_6. The codes are chosen to detect and correct errors with minimum effort. A stabilizer code is a common eigenspace of a set of commuting products of Pauli operators (the 'check operators'). Such products are denoted by strings of X, Y, Z and I. For example, XIZ is a Pauli product on three qubits with X acting on the first and Z on the last. C_4 has check operators $XXXX$ and $ZZZZ$. It encodes a 'qubit pair' whose qubits may be labelled L and S, and defined by encoded operators $X_L = XXII$, $Z_L = ZIZI$, $X_S = IXIX$ and $Z_S = IIZZ$. C_4 is an optimal qubit-based one-error-detecting code. C_6 has check operators $XIIXXX$, $XXXIIX$, $ZIIZZZ$ and $ZZZIIZ$, which act on three consecutive qubit pairs. It encodes a qubit pair defined by encoded operators $X_L = IXXIII$, $Z_L = IIZZIZ$, $X_S = XIXXII$, $Z_S = IIIZZI$. C_6 is an optimal qubit-pair-based one-error-detecting code. The C_4/C_6 architecture uses C_4 to obtain level-1-encoded qubit pairs. We build subsequent levels by using three encoded qubit pairs to form a next-level C_6-encoded qubit pair as shown in Fig. 1.

Given a joint eigenstate of the check operators, its list of eigenvalues is the 'syndrome'. The level-l encoding has check operators that can be derived from the check and encoded operators of C_4 and C_6. Ideally, the state of a level-l block has syndrome **0** (all eigenvalues are +1). In the presence of errors this is rarely the case, so the encoded qubits' state is defined only with respect to a current 'Pauli frame' and an implicit recovery scheme. The Pauli frame is defined by a Pauli product that restores the error-free state of the block to the syndrome **0** subspace. The implicit recovery scheme determines the Pauli products needed to coherently map states with other syndromes to one with the error-free syndrome. By using the Pauli frame, we can avoid explicitly applying Pauli products for error correction and teleportation compensation[9,25]. Error detection and correction are based on measurements that retroactively determine the syndrome of the state (the current syndrome has already been affected by further errors). An error is detected if the syndrome is not error-free according to the Pauli frame. In 'postselected' quantum computing, the state is then rejected and the computation restarted. In standard quantum computing, the syndrome information is used to correct errors with a Pauli frame update. To do so, we use the fact that C_4 and C_6 can detect any error of one qubit and one qubit pair, respectively. If the location of the error is known, it can be corrected. This leads to the following error detection and correction (ED/EC) procedure: first we check the level-1 C_4 syndromes of each block of four qubits. For each block where an error is detected, mark the encoded level-1 qubit pair as having an error. Proceed to level 2 and check the (encoded) C_6 syndrome for each block of three level-1 pairs. If exactly one of the level-1 pairs has an error, use the C_6 syndrome to correct it. If not, mark the encoded level-2 pair as having an error unless none of the three level-1 pairs has an error and the C_6 syndrome is error-free according to the Pauli frame. Continue in this fashion through all higher levels. For optimizing state preparation, we can replace the error-correction step by error detection at the top few levels, depending on context, as explained below.

Error-correcting teleportation

To obtain syndrome information for a block B containing an encoded qubit pair we use error-correcting teleportation. We first prepare two blocks B_1 and B_2, each encoding a logical qubit pair so that the first pair is maximally entangled with the second, in the logical state $(|0000\rangle + |0101\rangle + |1010\rangle + |1111\rangle)/2$. B_1 and B_2 form an 'encoded (or logical) Bell pair'. The encoded Bell pair is prepared 'fault-tolerantly', so that each block's errors are essentially as though the physical qubits were subject to independent errors of order γ. The next step is to apply Bell measurements (the first step of conventional quantum teleportation[26]) to corresponding physical qubits in B and B_1. This results in the transfer of B's encoded state to B_2, up to a known change in the Pauli frame. It can be shown that the Bell measurement outcomes reveal the eigenvalues of the products of corresponding check operators on B and B_1, which is sufficient for inferring the needed syndrome for error detection and correction. Error detection or correction is successful if the combined errors from B and B_1 are within the capabilities of the codes. See ref. 16 for further details.

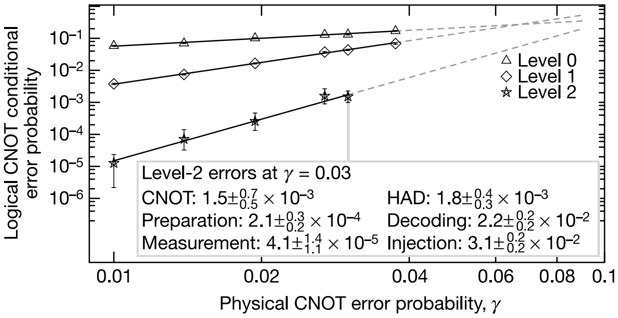

Figure 2 Conditional logical errors with postselection. The plot shows logical CNOT errors conditional on not detecting errors as a function of EPG parameter γ at levels 0, 1 and 2. The logical CNOT is implemented with transversal physical CNOTs and two error-detecting teleportations, where the output state is accepted only if no errors are detected in the teleportations. The data show the incremental error attributable to the logical CNOT in the context of a longer computation (Supplementary Information B). The error bars are 68% confidence intervals. The solid lines are obtained by gradient-descent likelihood maximization. Extrapolations are shown with dashed lines and suggest that logical EPG improvements with increasing levels are possible above $\gamma = 6\%$. Other operations' errors for $\gamma = 3\%$ and level 2 are shown in the inset table. The decoding error is the incremental error introduced by decoding a block into two physical qubits. The injection error is the error in a logical state that we prepare by decoding one block of a logical Bell pair and measuring the decoded qubits. Decoding and injection errors were found to decrease from level 1 (decoding error $4.4\pm^{0.4}_{0.4}\times10^{-2}$, injection error $5.5\pm^{0.5}_{0.4}\times10^{-2}$) to level 2.

Encoded state preparation

Fault-tolerant architectures depend on having a plentiful supply of verified, fault-tolerantly prepared encoded states. In the C_4/C_6 architecture, encoded Bell pairs are fundamental to preparing all other such states. An encoded Bell pair on blocks B_1 and B_2 is prepared at level $l+1$ from level-l-encoded Bell pairs in three steps. The first step is to prepare level-$l+1$-encoded $|00\rangle$ in block B_1 and $|{+}{+}\rangle$ in block B_2. C_4 and C_6 have the property that these states are local variants of level-l 'cat' states (states such as $(|0\ldots0\rangle + |1\ldots1\rangle)/\sqrt{2}$), which can be obtained and verified by linking level-l Bell pairs. The second step is a 'transversal' CNOT consisting of physical CNOTs applied from qubits of B_1 to corresponding qubits of B_2. The third step involves error-correcting teleportations of level-l sub-blocks of B_1 and B_2, which is required to manage errors introduced in the first two steps and limit correlations between B_1 and B_2. The state preparation networks are shown in Supplementary Information A.

Logical Clifford gates for C_4 and C_6

For simplicity, we treat the qubits in a logical qubit pair identically and ignore one of them for the purpose of computation. Preparation of logical $|00\rangle$ and $|{+}{+}\rangle$ is accomplished by using the

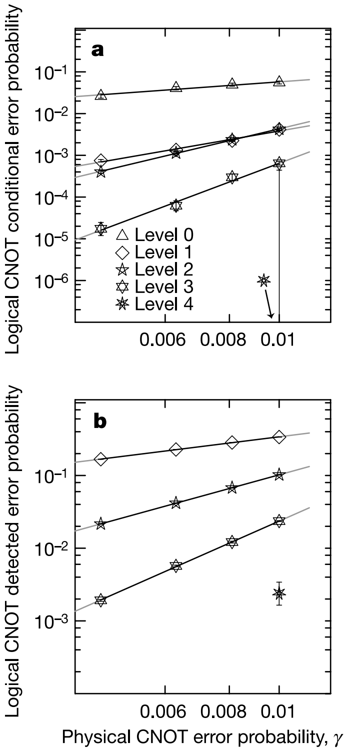

Figure 3 Conditional and detected logical errors with error correction. The plot shows incremental detected and conditional logical errors for a logical CNOT as a function of EPG parameter γ up to level 4. Error bars and lines are as described in the caption of Fig. 2. The combination of error correction and detection is as required for the error-correcting C_4/C_6 architecture. **a**, The logical CNOT's error conditional on not detecting an uncorrectable error. **b**, The probability of detecting an uncorrectable error. At $\gamma = 1\%$, the detected errors are $2.4\pm^{0.0}_{0.0}\times10^{-2}$ (level 3) and $2.4\pm^{1.0}_{0.7}\times10^{-3}$ (level 4). The conditional errors are $6.4\pm^{0.6}_{0.6}\times10^{-4}$ (level 3) and $0.0\pm^{4.4}_{0.0}\times10^{-4}$ (level 4). For comparison, the preparation errors at levels 3 and 4, respectively were found to be $2.1\pm^{0.3}_{0.3}\times10^{-4}$ and $0.0\pm^{1.0}_{0.0}\times10^{-4}$ (detected errors) and $3.3\pm^{7.5}_{2.7}\times10^{-6}$ and $0.0\pm^{1.0}_{0.0}\times10^{-4}$ (conditional errors). The measurement errors are $4.7\pm^{0.4}_{0.4}\times10^{-4}$ and $5.6\pm^{12.8}_{4.6}\times10^{-5}$ (detected errors) and $3.3\pm^{7.4}_{2.7}\times10^{-6}$ and $0.0\pm^{1.0}_{0.0}\times10^{-4}$ (conditional errors). Finally, the HAD errors at level 3 are $1.3\pm^{0.0}_{0.0}\times10^{-2}$ (detected error) and $3.5\pm^{0.6}_{0.5}\times10^{-4}$ (conditional error).

first step of the logical Bell pair preparation procedure followed immediately by error-correcting teleportations of the sub-blocks. The codes C_4, C_6 and their concatenations have the property that logical CNOTs and measurements can be implemented transversally[22]. This ensures fault tolerance. The HAD gate can be implemented transversally with a permutation of the physical qubits in a block. Permutations can be implemented by relabelling without physical manipulations and are also fault-tolerant. To control error propagation, we include with each logical gate error-correcting teleportations of its blocks.

$\mathcal{G}_{\min}$ thresholds

For the purpose of establishing high thresholds, we first consider postselected $\mathcal{G}_{\min}$ computing. Postselected computing is a model of computing that abstracts and generalizes the key non-deterministic aspects of techniques such as purification[27] and verified state preparation[28]. Here we use it to prepare states needed for scalable quantum computing without having to specify the state-preparation networks. Postselected computing is like standard quantum computing except that when a gate is applied, the gate may fail. If it fails, this is known. The probability of success must be non-zero. There may be gate errors conditional on success, but fault-tolerant postselected computing requires that such errors are small. We implement fault-tolerant postselected computing with the C_4/C_6 architecture by aborting the computation whenever an error is detected. Error-correcting teleportation is replaced by error-detecting teleportation, which uses the syndrome information only for error detection. In ref. 23 we used a computer-assisted heuristic analysis to obtain a threshold value of 3%, below which fault-tolerant postselected $\mathcal{G}_{\min}$ computing is possible. Here we use direct simulation of the error behaviour of postselected encoded CNOTs with error-detecting teleportation at up to two levels of encoding and physical EPGs of $1\% \leq \gamma \leq 3.75\%$. The simulation method is explained in Supplementary Information B. The simulated conditional logical errors are shown in Fig. 2 and suggest a threshold of above 6% by extrapolation.

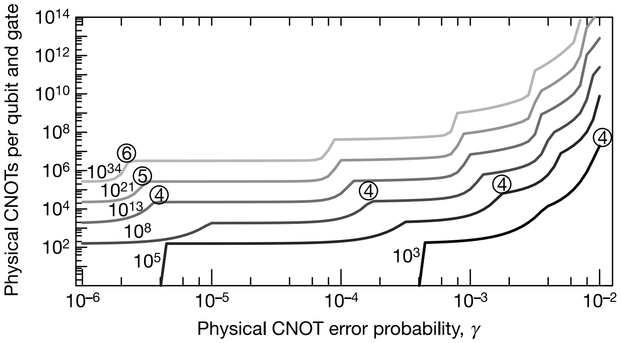

Figure 4 Estimating C_4/C_6 resource requirements. The figure shows the number N_{CNOT} of physical CNOTs required per qubit and gate to implement computations of sizes $G = 10^3, 10^5, \ldots, 10^{34}$ (curves with G indicated). Other resources are dominated by N_{CNOT}. An order-of-magnitude estimate of the total number of physical CNOTs required by a computation can be made as follows: we determine the number G of gates required, including 'memory' gates. Using the corresponding curve in the figure, we find N_{CNOT} at the physical EPG. We then multiply N_{CNOT} by G and the average number of additional logical CNOTs per gate required for fault-tolerantly preparing and purifying states such as logical $|\pi/8\rangle$. A conservative estimate for the latter number is 300. With maximum parallelism, the 'scale-up' (number of physical qubits per logical qubit) is of the same order as N_{CNOT}. If the memory error is not too large, this can be reduced to about $2(1 + 2(l - 1))3^{l-1}$ with moderate parallelism. The circled numbers are at the points on each curve above which the indicated level of concatenation must be used. Levels increment at each step-like feature of the curves.

Scalable $\mathcal{G}_{\min}$ computing with the C_4/C_6 architecture requires lower EPGs and the use of error correction to increase the probability of success to near 1. To optimize the resource requirements needed to achieve a given logical EPG, the last level at which error correction is used in the ED/EC procedure is d_l levels below the relevant top level, where d_l depends on context and γ. At higher levels, errors are only detected. For simplicity and to enable extrapolation by modelling, we examined a fixed strategy with $d_l = 1$ in all state-preparation contexts and $d_l = 0$ (maximum error correction) in the context of logical computation. The relevant top level in a state preparation context is the level of a block measurement or error-correcting teleportation of a sub-block, not the logical level of the state that is eventually prepared. Each logical gate now has a probability of detected but uncorrectable error, and a probability of logical error conditional on not having detected an error. Figure 3 shows both error probabilities up to level 4 for a logical CNOT with error-correcting teleportation and EPGs of $\gamma \leq 1\%$. The data indicate that the $\mathcal{G}_{\min}$ threshold for this architecture is above 1%.

Universal computation

To complete the $1\mathcal{G}_{\min}$ gate set so that we can implement arbitrary quantum computations, it suffices to add HAD and preparation of the state $|\pi/8\rangle = \cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle$[29,30] in both qubits of a logical pair. The logical errors of HAD are less than those of the logical CNOT. To prepare logical $|\pi/8\rangle|\pi/8\rangle$ in a qubit pair, we obtain a logical Bell pair, decode its first block into two physical qubits and make measurements to project the physical qubits' states onto $|\pi/8\rangle$ or the orthogonal state. If an orthogonal state is obtained, we adjust the Pauli frame by Y operators accordingly. Because of the entanglement between the physical qubits and the logical ones, this prepares the desired logical state, albeit with error. This procedure is 'state injection'. To decode the first block of the Bell pair, we first decode the C_4 sub-blocks and continue by decoding six-qubit sub-blocks of C_6. Syndrome information is obtained in each step and can be used for error detection or correction. The error in decoding is expected to be dominated by the last decoding steps. Consequently, the error in the injected state should be bounded as the number of levels increase, which we verified by simulation. To remove errors from the injected states, logical purification can be used[30,31] and is effective if the error of the injected state is less than 0.141 (ref. 31). The purification method can be implemented fault-tolerantly to ensure that the purified logical $|\pi/8\rangle$ states have errors similar to those of logical CNOTs (Supplementary Information C).

Consider the threshold for fault-tolerant postselected universal quantum computing. The logical HAD and injection errors at $\gamma = 3\%$ and level 2 are shown in Fig. 2. The injection error is well below the maximum allowed and is not expected to increase substantially for higher levels. The injection error should scale approximately linearly with EPG, so the extrapolated threshold above 6% may apply.

The injection and purification method for preparing states needed to complete the gate set works with the error-correcting C_4/C_6 architecture. Consider state injection at $\gamma = 1\%$. The context for injection is state preparation, which determines the combination of error correction and detection as discussed above. The conditional logical error after state injection was determined to be $8.6\pm^{0.6}_{0.5}\times 10^{-3}$ at level 3 and $1.1\pm^{0.1}_{0.1}\times 10^{-2}$ at level 4, comparable to γ and sufficiently low for $|\pi/8\rangle$ purification. As a result, the C_4/C_6 architecture enables scalable quantum computing at EPGs above 1%.

To obtain higher thresholds, we use fault-tolerant postselected computing to prepare states in a code that can handle higher EPGs than C_4/C_6 concatenated codes can. The states are chosen so that we can implement a universal set of gates by error-correcting teleportation. Suppose that arbitrarily low logical EPGs are achievable with the C_4/C_6 architecture for postselected computing. To compute scalably, we choose a sufficiently high level l for the C_4/C_6 architecture and a very good error-correcting quantum code C_e. The first step is to prepare the desired C_e-encoded states using level-l-encoded qubits, in essence concatenating C_e onto level l of the C_4/C_6 architecture. Suppose that the conditional error in the logical state prepared can be made arbitrarily small. The second step is to decode each block of the C_4/C_6 architecture to physical qubits to obtain unconcatenated C_e-logical states (partial decoding). Once these states are successfully prepared, we use them to implement each logical gate by error-correcting teleportation. Simulations show that the postselected decoding introduces an error $\lesssim \gamma$ for each decoded qubit (Fig. 2). There is no postselection in error-correcting teleportation with C_e, and it is sensitive to decoding error in two blocks ($\sim 2\gamma$) as well as the error of the CNOT ($\sim\gamma$) and the two physical measurements ($\sim 8\gamma/15$) required for the Bell measurement. Hence, the effective error per qubit that needs to be corrected is $\sim 3.53\gamma$. The maximum error probability per qubit correctable by known codes C_e is ~ 0.19 (ref. 32). Thus, if $\gamma \lesssim 5\%$ (conservatively below 0.19/3.53) the C_e architecture can have small logical errors, say below 10^{-3}. Scalable quantum computing is then possible by using C_e-encoded qubits as the founding qubits for the error-correcting C_4/C_6 architecture (for example). Because 5% is below the extrapolated threshold for the postselected C_4/C_6 architecture, scalable quantum computing may be possible with our architecture at EPGs as high as 5% (or at least 3% without extrapolation). Although the postselection overheads are extreme, the above architecture is theoretically efficient: the asymptotic overheads for implementing a quantum computation are polynomial in terms of the computation's size.

Resources

The resource requirements for the error-correcting C_4/C_6 architecture can be mapped out as a function of γ for different sizes of computations. We do not have analytical expressions for the resources for logical Bell pair preparation or for the logical errors as a function of γ and, with our current capabilities, we are not able to determine them in enough detail by simulation. We therefore use naive models to approximate the expressions needed. The number of physical CNOTs used in a logical Bell-pair preparation is modelled by functions of the form $C/(1-\gamma)^k$, which would be correct on average if the state-preparation network had C gates of which k failed independently with probability γ, and the network was repeatedly applied until none of the k gates failed. C and k depend on the level of concatenation. The logical error probabilities are modelled at level $l \geq 1$ by $p_d(l) = d(l)\gamma^{f(l+1)}$ (detected error) and $p_c(l) = c(l)\gamma^{f(l+2)}$ (conditional logical error), where $f(0) = 0$, $f(1) = 1, f(l+1) = f(l) + f(l-1)$ is the Fibonacci sequence. These expressions are asymptotically correct as $\gamma \rightarrow 0$. We verified that they model the desired values well and determined the constants at low levels by simulation and at high levels by extrapolation (Supplementary Information C).

Figure 4 shows the resource requirements as a function of computation size. Following the instructions in the caption, we obtain the following order-of-magnitude estimates: at an EPG of 1%, a computation with 10^3 or 10^5 gates and (say) 100 or more qubits requires 6×10^{12} or 2×10^{17} physical CNOTs, respectively. A more precise calculation shows that 1.2×10^{14} physical CNOTs are required for 1,000 logical $|\pi/8\rangle$ preparations (Supplementary Information C). Given current capabilities, the outputs of these computations are not predictable with classical algorithms. The quantum resource requirements are large and at present difficult to realize. However, comparable complexity is achieved in today's classical computers: central processing units have 10^8 or more transistors operating at rates of 10^9 steps per second[33], making available up to 10^{17} bit operations per second.

The resource requirements decrease rapidly with lower EPGs. At EPGs well below 10^{-3}, an architecture based on unconcatenated block codes such as that of Steane[9] is expected to be more efficient. Indeed, at an EPG of 10^{-4}, such architectures use one to two orders of magnitude fewer resources. The C_4/C_6 architecture still has the advantage of simplicity, and of yielding more reliable answers conditional on having no detected errors.

Discussion

An important use of studies of fault-tolerant architectures is to provide guidelines for EPGs that should be achieved to meet the low-error criterion for scalability. Such guidelines depend on the details of the relevant error models and constraints on two-qubit gates. Nevertheless, the value of $\gamma = 10^{-4}$ has often been cited as the EPG to be achieved. With architectures such as that of Steane[9,34] and the one introduced here, resource requirements at $\gamma = 10^{-3}$ are now comparable to what they were for $\gamma = 10^{-4}$ at the time this value was starting to be cited[22].

Several open problems arise from the work presented here. Can the high thresholds evidenced by our simulations be mathematically proved? Are thresholds for postselected computing strictly higher than thresholds for scalable standard quantum computing? Recent work by Reichardt[34] shows that Steane's architecture can be made more efficient by the judicious use of error detection, improving Steane's threshold estimates to around 10^{-2}. How do the available fault-tolerant architectures compare for EPGs between 10^{-3} and 10^{-2}? It would be helpful to improve significantly the resource requirements of fault-tolerant architectures, particularly at high EPGs. □

Received 12 November 2004; accepted 4 January 2005; doi:10.1038/nature03350.

1. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26,** 1484–1509 (1997).
2. Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21,** 467–488 (1982).
3. Abrams, D. S. & Williams, C. P. Fast quantum algorithms for numerical integrals and stochastic processes. Preprint at ⟨http://arXiv.org/quant-ph/9908083⟩ (1999).
4. DiVincenzo, D. The physical implementation of quantum computation. *Fort. Phys.* **48,** 771–783 (2000).
5. Preskill, J. Reliable quantum computers. *Proc. R. Soc. Lond. A* **454,** 385–410 (1998).
6. Kitaev, A. Y. Quantum computations: Algorithms and error correction. *Russ. Math. Surv.* **52,** 1191–1249 (1997).
7. Aharonov, D. & Ben-Or, M. Fault-tolerant quantum computation with constant error. Preprint at ⟨http://www.arXiv.org/quant-ph/9906129⟩ (1999).
8. Knill, E., Laflamme, R. & Zurek, W. H. Resilient quantum computation. *Science* **279,** 342–345 (1998).
9. Steane, A. M. Overhead and noise threshold of fault-tolerant quantum error correction. *Phys. Rev. A* **68,** 042322 (2003).
10. Gottesman, D. *Stabilizer Codes and Quantum Error Correction.* PhD thesis, California Institute of Technology, Pasadena (1997).
11. Leibfried, D. *et al.* Experimental demonstration of a robust, high-fidelity geometric two ion-qubit phase gate. *Nature* **422,** 412–415 (2003).
12. Roos, C. F. *et al.* Bell states of atoms with ultralong lifetimes and their tomographic state analysis. *Phys. Rev. Lett.* 220402 (2004).
13. Knill, E., Laflamme, R., Martinez, R. & Tseng, C.-H. An algorithmic benchmark for quantum information processing. *Nature* **404,** 368–370 (2000).
14. Childs, A. M., Chuang, I. L. & Leung, D. W. Realization of quantum process tomography in NMR. *Phys. Rev. A* **64,** 012314 (2001).
15. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge Univ. Press, Cambridge, UK, 2001).
16. Knill, E. Scalable quantum computation in the presence of large detected-error rates. Preprint at ⟨http://www.arXiv.org/quant-ph/0312190⟩ (2003).
17. Gottesman, D. & Chuang, I. L. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature* **402,** 390–393 (1999).
18. Zhou, X., Leung, D. W. & Chuang, I. L. Methodology for quantum logic gate construction. *Phys. Rev. A* **62,** 052316 (2000).
19. Bollinger, J. J., Heinzen, D. J., Itano, W. M., Gilbert, S. L. & Wineland, D. J. A 303 MHz frequency standard based on trapped Be$^+$ ions. *IEEE Trans. Instrum. Meas.* **40,** 126–128 (1991).
20. Steane, A. M. Quantum computer architecture for fast entropy extraction. *Quant. Inf. Comput.* **4,** 297–306 (2002).
21. Svore, K. M., Terhal, B. M. & DiVincenzo, D. P. Local fault-tolerant quantum computation. Preprint at ⟨http://www.arXiv.org/quant-ph/0410047⟩ (2004).
22. Steane, A. Space, time, parallelism and noise requirements for reliable quantum computing. *Fort. Phys.* **46,** 443–457 (1998).
23. Knill, E. Fault-tolerant postselected quantum computation: Threshold analysis. Preprint at ⟨http://www.arXiv.org/quant-ph/0404104⟩ (2004).
24. Knill, E. Quantum computing with very noisy devices. Preprint at ⟨http://www.arXiv.org/quant-ph/0410199⟩ (2004).

25. Raussendorf, R. & Briegel, H. J. A one-way quantum computer. *Phys. Rev. Lett.* **86,** 5188–5191 (2001).
26. Bennett, C. H. *et al.* Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels. *Phys. Rev. Lett.* **70,** 1895–1899 (1993).
27. Bennett, C. H. *et al.* Purification of noisy entanglement and faithful teleportation via noisy channels. *Phys. Rev. Lett.* **76,** 722–725 (1996).
28. Shor, P. W. In *Proc. 37th Symp. Foundations of Computer Science (FOCS)* 56–65 (IEEE Press, Los Alamitos, California, 1996).
29. Knill, E., Laflamme, R. & Zurek, W. Resilient quantum computation: Error models and thresholds. *Proc. R. Soc. Lond. A* **454,** 365–384 (1998).
30. Knil, E. Fault-tolerant postselected quantum computation: Schemes. Preprint at ⟨http://www.arXiv.org/quant-ph/0402171⟩ (2004).
31. Bravyi, S. & Kitaev, A. Universal quantum computation based on a magic states distillation. Preprint at ⟨http://www.arXiv.org/quant-ph/0403025⟩ (2004).
32. DiVincenzo, D. P., Shor, P. W. & Smolin, J. A. Quantum-channel capacity of very noisy channels. *Phys. Rev. A* **57,** 830–839 (1998).
33. Intel Cooperation. Microprocessor quick reference guide. ⟨http://www.intel.com/pressroom/kits/quickreffam.htm⟩ (2004).
34. Reichardt, B. W. Improved ancilla preparation scheme increases fault tolerant threshold. Preprint at ⟨http://www.arXiv.org/quant-ph/0406025⟩ (2004).

Supplementary Information accompanies the paper on **www.nature.com/nature**.

Acknowledgements This work is a contribution of NIST, an agency of the US government, and is not subject to US copyright. Partial support from the DARPA QuIST programme is acknowledged.

Competing interests statement The author declares that he has no competing financial interests.

Correspondence and requests for materials should be addressed to E.K. (knill@boulder.nist.gov).