Comparison of ancilla preparation and measurement procedures for the Steane [[7,1,3]] code on a model ion trap quantum computer

Yu Tomita, Mauricio Gutiérrez, Chingiz Kabytayev, and Kenneth R. Brown*
Schools of Chemistry and Biochemistry; Computational Science and Engineering; and Physics, Georgia Institute of Technology, Atlanta, Georgia 30332, USA

M. R. Hutsel, A. P. Morris, Kelly E. Stevens, and G. Mohler
Georgia Tech Research Institute, Atlanta, Georgia 30332, USA

(Dated: May 3, 2013)

We schedule the Steane [[7,1,3]] error correction on a model ion trap architecture with ballistic transport. We compare the level one error rates for syndrome extraction using the Shor method of ancilla prepared in verified cat states to the DiVincenzo-Aliferis method without verification. The study examines how the quantum error correction circuit latency and error vary with the number of available ancilla and the choice of protocol for ancilla preparation and measurement. We find that with few exceptions the DiVincenzo-Aliferis method without cat state verification outperforms the standard Shor method. We also find that additional ancilla always reduces the latency but does not significantly change the error due to the high memory fidelity.

* Author to whom correspondence should be addressed. Electronic mail:kenbrown@gatech.edu

I. INTRODUCTION

The reliability of a fault-tolerant quantum computation depends on not only the choice of error correction code but also the methods used for syndrome extraction, state preparation, and error decoding. These choices can be compared at an abstract level of quantum circuits and depolarizing channels, but realistic quantum information devices will have error rates that depend on circuit elements as well as limited connectivity for applying two-qubit gates [1]. Topological codes have an advantage in that they are naturally suited to nearest-neighbor architectures [2, 3]. Concatenated code error correction procedures require additional resources to map these circuits onto local architectures which leads to a reduced error threshold relative to the abstract model [1, 4]. Still, these codes offer potential benefits over topological codes for systems with low-error rates and fast communication between distant qubits by ballistic transport or interaction with flying qubits.

The extraction of syndromes requires the preparation and measurement of fresh ancilla states. This process is what allows us to remove the entropy from the quantum system [5]. One question that arises is how many extra qubits should one dedicate for ancilla. Consider the Steane [[7,1,3]] code [6] using the Shor method for syndrome extraction [7] based on verified cat states. Each cat state contains four qubits; six syndrome measurements are required, suggesting that between 4 and 24 ancilla qubits could be used. The proper balance of ancilla resources depends on the device details and the error rates of the physical operations. For most quantum information devices, measurement is the slowest operation. It has been shown that in a nonequiprobable error environment where Z type error is dominant, the fidelity of the Shor state may decreas with verification [8, 9]. To avoid bottlenecks due to the measurements used to verify cat states, DiVincenzo and Aliferis [10] proposed a method that does not require verification of ancilla states. Here we compare these methods on a model ion trap quantum computer.

The ion trap architecture is a promising basis for quantum computation and have already demonstrated long coherence times and high fidelity operations. A scalable architecture has been proposed based on shuttling ions between traps [11] and work is ongoing to implement this architecture experimentally [12–20]. This framework has been the basis for a number of studies on the resource requirements for implementing large quantum algorithms [21–23] and has also been considered as the elementary logical unit of hybrid schemes using photonic interconnects [24].

While an arbitrarily well-connected ion trap layout can be envisioned, such that there is little fear of collision or backlogs, this is not realistic given current technology. The ion trap layout, for example, is a grid of narrow paths where no ion may pass by another. Performing multiple two-qubit gates efficiently becomes problematic. There will be a limited number of interaction zones, and the paths to reach them will be obstructed by other qubits which adds non-trivial transport time in addition to the time required to execute gates.

This introduces the issue of latency which is defined here as the total amount of time experienced by qubits after physical state preparation. Latency includes qubit transport times, gate times, and idle times due to traffic in the layout. When mapping a quantum circuit to a series of device operations for a layout with limited connectivity, resources dedicated to the transport of qubit information quickly come to dominate the cost of algorithm execution [21]. The goal then becomes to find a schedule of qubit operations that reduces latency as much as possible, both to make operation times feasible for large algorithms and to reduce memory errors due to ever-present environmental noise. Parallelization of operations is one of the most direct ways to reduce latency and is the focus of this work.

One simple way to increase the parallelizability is to prepare additional ancillary qubits ahead of time in states needed by the computation. Just as in classical computing, this is a trade-off between memory and latency. Maximum parallelization may call for the simultaneous creation and preparation of multiple ancilla sets (low latency), but this results in "stale" ancilla that may suffer logical errors before being used (poor memory performance). Both of these factors can be calculated quantitatively. Total latency can be calculated given a layout, a schedule of gates, and a set of operation times (gate time, qubit speed, measurement time, etc.). Logical error can be characterized in terms of fidelity, or alternatively in terms of the qubit error rate. In general, it will increase with increasing latency. Using these calculations, we can study the effect of additional resources on the error rate of the overall algorithm execution.

The impact of ancilla preparation on overhead has been previously studied for both individual logical qubits [25] and large-scale quantum computation [21]. The individual logical qubit study was done for the Steane [[7,1,3]] code, wherein various ancilla preparations were analyzed. The study found that the additional overhead introduced by preparation and verification hurt the ancilla fidelity more than the ancilla encoding and verification helped. However, the study assumed an abstracted layout, and did not include movement latency or related memory errors. The large-scale study looked at ion trap layouts holding large numbers of logical qubits, and found that ancilla generation was the primary performance bottleneck. The bottleneck was removed by creating regions dedicated to ancilla preparation and recycling. Our approach flows in part from these two studies; here, multiple ancilla blocks are assigned to individual logical qubits, and two different ancilla encodings are employed. Once an ancilla block size and encoding are chosen, execution of the Steane code is simulated using a software design tool. The design tool is used to include realistic latency and scheduling bottlenecks, pointing towards the most practical ancilla encoding and block size. Our study focuses on a single round of Steane-code quantum error correction on a model ion trap architecture as a function of ancilla encoding/decoding and ancilla resources.

II. METHODS

A. Cat state syndrome extraction and the Steane [[7,1,3]] code

The Steane [[7,1,3]] code is the best-known of the Calderbank-Shor-Steane codes [26]. It encodes one logical qubit into seven physical qubits. The resulting logical states $|0\rangle$ and $|1\rangle$ have a Hamming distance of three and the code is able to detect and correct up to one physical bit-flip error and one physical phase-flip error. The Steane code has been widely studied and has been shown to have a threshold in the range between 10^{-3} and 10^{-6} [27–29] which makes it suitable for fault-tolerant quantum error correction (FTQEC).

The Steane code has six weight-four syndrome operators. Each syndrome is extracted by measuring a four-qubit cat state after interacting with the data qubits. In this study, the Steane QEC process is simulated and the latency and fidelity are calculated varying numbers of Shor ancilla sets from one to six. The two preparation/decoding cases are: (1) "on-demand" where only two sets of ancilla are prepared at any time, and (2) "one-time," where all ancilla are prepared at once before the first use. These procedures are done for both Shor and DiVincenzo-Aliferis ancilla encodings. Circuit diagrams for the two methods are shown in Figure 1.

B. Ion trap physical machine description

Previous ion trap studies in the literature have used a gate-level error model to calculate error correction properties. Here we model our ion trap using parameters and constraints derived from the Physical Machine Description (PMD) provided by the IARPA Quantum Computer Science program [30]. The ion trap PMD is a collection of linear ion trapping regions joined by cross junctions see Figure 2. It is modeled after the ion trap charge-coupled device architecture of Kielpinski, Monroe, and Wineland [11]. Each bus segment (white) section is capable of holding four ion trapping regions or "wells." Each well is capable of hosting up to five ions. Individual ion loading wells are indicated in yellow, and interaction wells capable of executing gate or measurement operations are in green. In order to undergo a two qubit gate (such as controlled-phase), the two qubits must be co-located in an interaction well. Shuttling a qubit between adjacent empty wells takes 10 μs. There is an additional time cost of 10 μs to add or remove a qubit from a well that is occupied This reflects the increased experimental complexity of joining and splitting single ions from ion chains [31, 32].

Logical errors are assumed to arise from stochastic white noise and $1/f$ noise in the control and background Hamiltonians. The result is a very asymmetric error model that better reflects the dominance of gate errors over memory errors in the actual physical system. The model does not consider the heating of the ion motion due to transport. The error of two-qubit ion gates is modeled as a stochastic noise term in the two-qubit Hamiltonian. Table I gives the latency and error rate costs of each gate type for the ion trap PMD.

In order to get the error rates in Table I, we approximate the real error channel derived from the stochastic noise with the closest Pauli error channel. We denote the process matrix of the noisy gate by χ'. The process matrix of the operation corresponding to the target (error-free) unitary followed by an X gate is denoted by χ^X. The process matrices corresponding to the target unitary followed by a Y or Z gate are denoted analogously. We then calculate the error rates as the overlap between χ^i and χ': $X_{\mathrm{er}} = \frac{1}{4}\langle\chi^X, \chi'\rangle$, $Y_{\mathrm{er}} = \frac{1}{4}\langle\chi^Y, \chi'\rangle$, and $Z_{\mathrm{er}} = \frac{1}{4}\langle\chi^Z, \chi'\rangle$ where $\langle A, B\rangle = \mathrm{Tr}(A^\dagger B)$.

The measurement error is above the Steane code threshold but this is be fixed by introducing two extra qubits and CZ and Hadamard gates as shown in Figure 3. This enhancement provides us the error rate of $O(\epsilon^2)$ where ϵ is the error rate of a single measurement. The enhanced measurement operation is denoted as 'MULTIMEASURE' in Table I.

C. Quantum Machine Parameterizer

A design tool is required to model the ion trap layout and execute qubit schedules on it. We use the Quantum Machine Parameterizer (QMP) code suite developed at GTRI. QMP is used for designing architecture layouts and creating operation schedules with real locality constraints. QMP can currently be used for any hardware where the locality constraints can be mapped to a planar graph. QMP has three primary facets: quantum computer layout modeling, operation scheduling, and physical qubit state tracking.

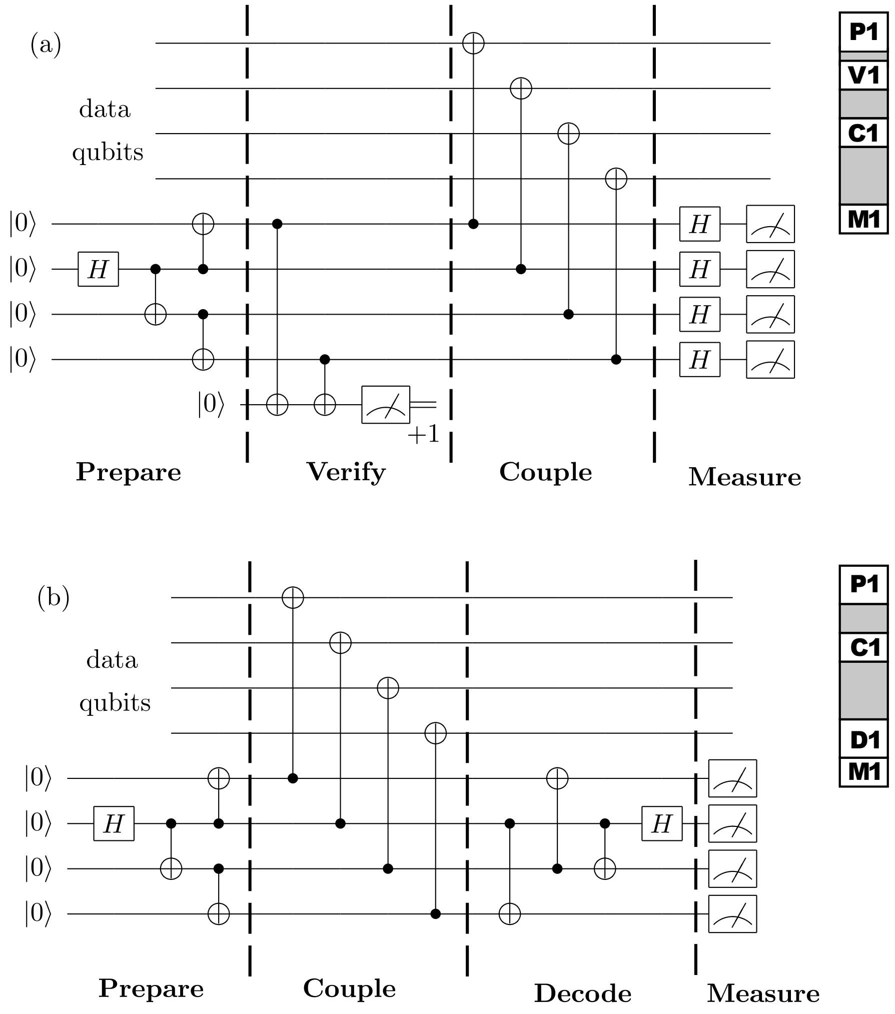

FIG. 1. Circuits for extraction of Z type syndrome measurement of the Steane code using the (a) standard Shor and (b) DiVincenzo-Aliferis method. In the DiVincenzo-Aliferis method, the cat state verification step is substituted with post-measurement decoding of the ancilla. Dashed lines demarcate different sections of the circuits. Also shown to the right are representations of the schedule of operations as a function of circuit sections. P=Prepare, V=Verify, C=Couple, D=Decode, M=Measure. Grey regions correspond to operations that are exclusively movement.

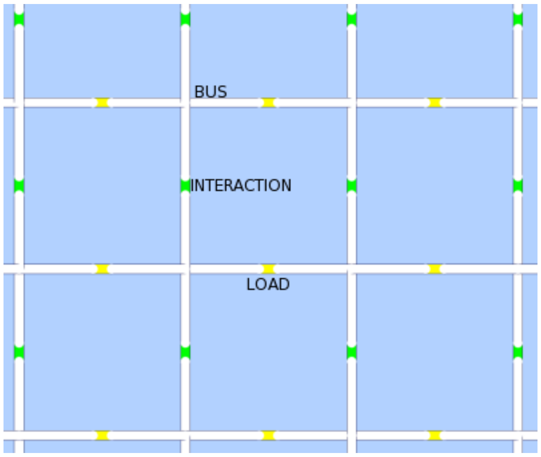

FIG. 2. Layout for QCS Ion Trap PMD

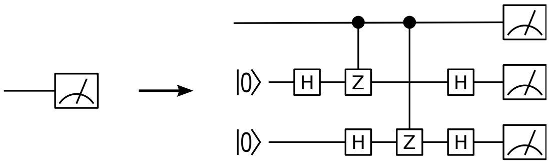

FIG. 3. Improvement of measurement gates by adding two ancilla. This reduces the failure rate when the ancilla preparation and the controlled gates are relatively reliable compare to the measurement gates. The final measurement value is determined by the majority vote of the three measurements.

TABLE I. Execution time and error rates of physical operations. MULTIMEASURE gates are the enhanced measurement gate described in Figure 3.

Gate	Latency (in μs)	Error Rate X	Error Rate Y	Error Rate Z
X	3	1.6E-8	8.0E-10	1.0E-9
Y	3	8.0E-10	1.6E-8	1.0E-9
Z	3	0.0	0.0	1.8E-8
S	2	0.0	0.0	5.5E-9
T	1	0.0	0.0	1.7E-9
HADAMARD	6	1.6E-8	4.0E-9	1.9E-9
CZ	105.5	0.0	0.0	IZ: 6.7E-8 ZI: 6.7E-8 ZZ: 2.5E-5
PREPARE Z	10	0.0	0.0	0.0
MEASURE Z	100	0.0	0.0	1.0E-4
MULTIMEASURE Z	355	0.0	0.0	3.1E-6
WAIT/MOVE	t	0.0	0.0	$t\cdot$ 5.5E-10
JOIN/SPLIT	10	0.0	0.0	5.5E-9

The layout modeling module allows the user to describe a quantum computing system, specifically the physical connectivity of allowed qubit paths, and the location of addressable zones. Also, this module accepts device-dependent parameters such as gate times and movement cost times to customize the behavior of a physical machine. This includes such device-specific operations as JOIN and SPLIT operations, required for two-qubit interactions.

The operation scheduling module allows the user to write a schedule of operations that can be performed in a circuit-model-based quantum computer (gate operations, qubit preparation, etc.). This schedule is written at a "high-level" which we define as a list of gate operations and move requests that only specify qubit and destination address. The scheduling module then calculates qubit paths using a specialized A* path-finding algorithm [33, 34], parallelizes the schedule where possible, removes possible collision events, and produces a series of qubit movement operations. Movement parallelization is performed with highest priority given in terms of move request order in a parallel block in the schedule. The first qubit is moved with no impediment, provided that a path exists. The second qubit's movement must defer to the first qubit, and WAIT commands are issued as needed to the second qubit to prevent collisions. This continues until the end of a parallel block is reached. In order to optimize this approach to parallelized movement, QMP analyzes move calls, qubit start positions, and destinations, and then re-orders the move calls as necessary. This module uses the device-dependent timing parameters such as gate times to complete the latency calculation of the operations schedule. This approach means that QMP will automatically create different (ideally optimal) qubit transport schedules and overall latencies for different choices in initial qubit arrangement, ancilla population, etc.

Finally, the qubit state tracking module allows the user to visualize the positions of the physical qubits within the layout as a function of time, and produce as output the total latency of the operations schedule. The module also produces an "error schedule" file which reduces the detailed physical machine schedule to a sequence of error-relevant events that the Quantum Circuit Fault Tracer (Section II D) uses to calculate failure rate.

D. Quantum Circuit Fault Tracer

The Quantum Circuit Fault Tracer (QCFT) is a tool created to efficiently compute logical failure rates of concatenated FTQEC codes. This tool is based on the "fault paths" concept, introduced by Aliferis, Gottesman, and Preskill to calculate thresholds of distance-3 concatenated FTQEC codes [35]. It takes as input a quantum circuit and failure rates of physical gates including WAIT and MOVE. Starting from the output state qubits, the circuit is traced backwards, marking possible fault points. Fault points are circuit locations where an error can propagate into a fatal error for the FTQEC circuit. The QCFT then combines the fault points and calculates the overall failure rate of the given circuit. To improve the accuracy of the output logical failure rate, we separate errors into X-type and Z-type, and propagate them on the circuit independently. Each Y-type error rate in Table I is added into both X-type and Z-type rates. When the input circuit is a distance-3 QECC circuit, the output is the probability of having errors propagated into two or more output data qubits.

III. PROCEDURE

Executing an algorithm on the ion trap layout requires a set of starting positions for all of the qubits present in the computation. Each starting position corresponds to an interaction well. That is, each qubit has its own home interaction well which it starts in at the beginning of the QEC round; see Figure 4. The qubits are ordered into rows by function. The data qubits sit in the top row, and never move from their initial positions. Each additional row of qubits is a self-contained ancilla set, which is prepared according the Steane-Shor or Steane-DiVincenzo-Aliferis circuit, coupled with the data, returned to the ancilla area (although not necessarily to the set's original position) and measured to obtain the error syndrome. Each qubit returns to its home well after being called away for a sequence of two-qubit gates. By this convention, the control bit travels and the target bit stays home. These choices remove some optimization capability, but allow us to test and compare different QEC circuits from the same initial conditions and scheduling assumptions.

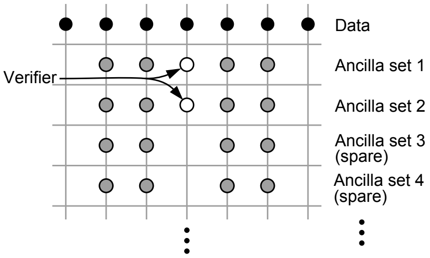

FIG. 4. Sample layout for the Steane algorithm with four sets of Shor ancilla and two preparation rows. The data qubits never move from their positions in the top row. The preparations rows are indicated by the presence of static verifier qubits.

Once the initial state and scheduling assumptions are established, QMP is used to calculate the time required to perform level one error correction assuming different operation times and ancilla management strategies. This includes varying the method of ancilla preparation and measurement, the number of ancilla, the parallelization of ancilla manipulation, and the time of gates and measurements. For each set of conditions, QMP uses the A* algorithm to optimize the latency from an initial hand-crafted schedule. An error schedule is then produced containing all gate and latency information. Finally, QFCT reads the error schedule and determines the logical error rate for a set of these conditions.

The Steane code is theoretically improved in terms of latency and error rate by creating multiple ancilla sets in parallel. To compare the efficacy of "on-demand" ancilla with "one-time" ancilla, we look at two different parallelizations for both Shor and DiVincenzo-Aliferis ancilla. At this point, we use the following notation to represent choices of ancilla management: yPxR where y is the number of ancilla sets that can be prepared and measured simultaneously (y=All means complete parallelization over the set) and x are the number of ancilla sets, one set per row in the layout.

The on-demand approach is 2PxR where ancilla preparation is only allowed in the two rows immediately below the data row. In this arrangement, ancilla qubits are moved up into one of the preparation rows, prepared, coupled with the data, and then moved to the bottom of the ancilla "stack" for measurement, which makes room for the next ancilla set. For the Shor case, verifier qubits are only kept in the two preparation rows. Six sets of ancilla are prepared in total in order to perform the three bit-stabilizer and three phase-stabilizer measurements.

The one-time approach is AllPxR wherein all ancilla are prepared at once and coupled with the data as soon as possible. For fewer than six ancilla sets, the ancilla sets are prepared at once, coupled with the data, measured, and then prepared again as soon as possible, repeated until all stabilizer measurements are performed. For the Shor case, every row has a verification qubit.

IV. RESULTS AND DISCUSSION

A. Scheduled error correction: Two-row preparation

The total latencies for the Steane-Shor and Steane-DiVincenzo-Aliferis algorithms are shown in Figure 5. The latency for one ancilla set represents the standard one-set strategy for the algorithms. For the case of a single ancilla row, the latencies for Steane-Shor and Steane-DiVincenzo-Aliferis are almost identical, with a slight advantage to Steane-Shor. In terms of latency, the verification step in Steane-Shor is roughly equivalent to the decoding step in Steane-DiVincenzo-Aliferis. However, for Steane-Shor, ancilla qubits can be moved to the data qubits while the verifier qubit is being measured. By contrast, the ancilla qubits are tied up during decoding, and no further parallelization is possible. Thus, assuming the verifier show successful ancilla creation, the Shor encoding is slightly more efficient. As will be shown, increasing the gate time, increasing the measurement time, or adding additional ancilla rows (increasing parallelizability) will separate Steane-Shor and Steane-DiVincenzo-Aliferis performance.

The latencies decrease greatly for both encodings with the addition of a second preparation row (2PxR) and, in general, continue to decrease with the addition of spare ancilla sets. The Steane-Shor latency reaches a minimum with four ancilla sets (two spare sets). This occurs because the preparation step in the Steane-Shor algorithm dominates the time required to perform a single bit or phase stabilizer measurement. Since in this scheme, only two sets can be prepared at a time, four total ancilla sets is sufficient to continuously utilize the preparation rows.

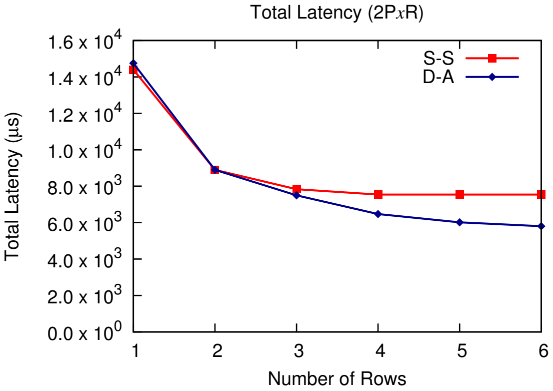

FIG. 5. Total latency for the Steane-Shor(S-S) and Steane-DiVincenzo-Aliferis(D-A) algorithms as a function of number of ancilla sets for the case of two preparation rows (2PxR).

The latency for the Steane-DiVincenzo-Aliferis algorithm reaches a minimum with a full six sets of ancilla. This occurs because, unlike preparation, decoding can occur at any interaction well. Additional ancilla rows allow for greater parallelized preparation and decoding and consequently less down-time between data-coupling steps. The elimination of the limited verifier measurement in mid-circuit clears out a critical bottleneck in the QEC execution.

More specifically, the efficiency of the algorithm execution depends on the degree of overlap between separate operations. The individual stabilizer measurements must be performed sequentially when using the one-set strategy. With two rows of ancilla, two stabilizer measurements can be performed in parallel. This is achieved by overlapping partially the preparation of the second set with the first. The preparation of the second set is delayed so that moving it to the data takes place when the first set is moved back to the top row for measurement. Following coupling with the data, the second set is moved to its original row via the unoccupied outer columns of the layout for measurement. Once a set of ancilla is measured, it is re-prepared for its next stabilizer measurement. This process, shown in Figure 6(a), is repeated two more times to perform the remaining four stabilizer measurements.

Adding a third or fourth set of ancilla provides additional spare sets that can move up and begin preparation as soon as the second set moves to the data. Furthermore, the measurements of the first and second sets complete before it is necessary to reuse them to perform the final two stabilizer measurements. Therefore, unnecessary delays are removed by using more sets. This process is shown in Figure 6(b). For Steane-Shor, adding fifth and sixth sets removes the need to reuse ancilla but does not provide any further decrease in the latency, due to the verification bottleneck.

Comparing DiVincenzo-Aliferis to Shor shows that Steane-DiVincenzo-Aliferis requires less time for preparation but more time for measurement. As a consequence, the ancillae cannot be reused as quickly as they are needed for a subsequent preparation. For example, with four sets of ancillae, the first four stabilizer measurements can be performed in rapid succession. However, ancilla sets three and four are prepared quickly and move to the data before

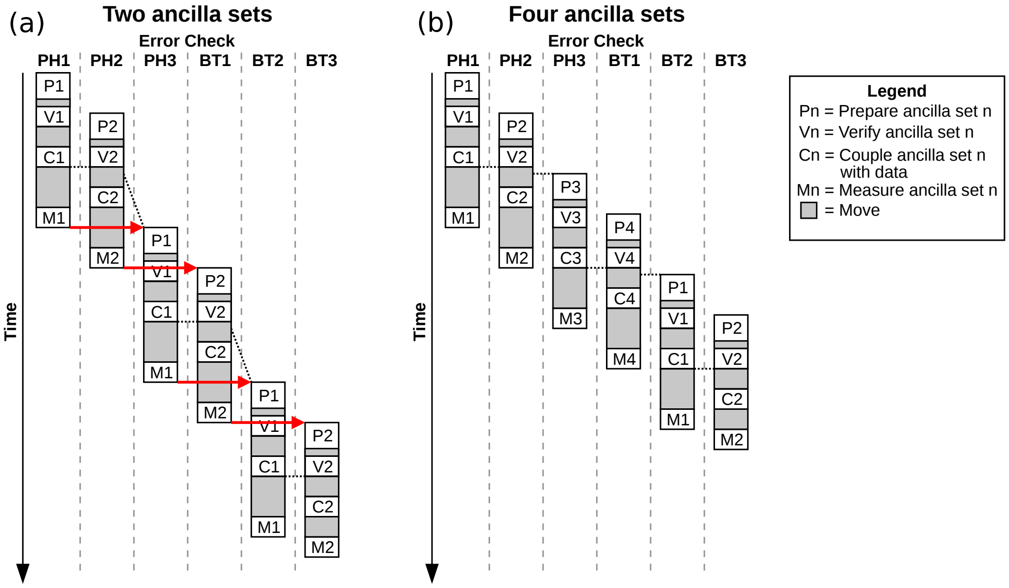

FIG. 6. Parallel strategy for implementing the Steane-Shor algorithm with (a) two sets of ancilla and two preparation rows and (b) four sets of ancilla and two preparation rows. "PHx" and "BTx" indicate bit-stabilizer or phase-stabilizer operations, respectively. Dashed lines indicate steps for separate ancilla sets that must occur in sequence. Red arrows indicate steps that must occur in sequence because an ancilla set is reused. The need to reuse the two ancilla sets, as indicated by the red arrows in (a), prevents the first set from being prepared as early as possible, as indicated by the non-horizontal dashed lines between PH2 and PH3 and BT1 and BT2. The additional ancilla sets in (b) ensure that the ancilla are used at the speed of computation, with measurement of an ancilla set occurring before the need to prepare that ancilla set.

the decoding/measuring of sets one and two are complete. Thus, the preparation of sets one and two for the last two stabilizer measurements cannot begin as early as possible. This process is shown in Figure 7. A unique set of ancilla must be available for each stabilizer measurement operation (six sets) to optimize fully the parallelization of the DiVincenzo-Aliferis algorithm.

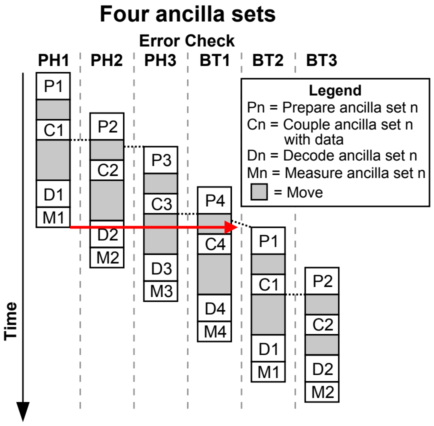
Four ancilla sets
Error Check
PH1
PH2
PH3
BT1
BT2
BT3
P1
C1
D1
M1
P2
C2
D2
M2
P3
C3
D3
M3
P4
C4
D4
M4
Time
Legend
Pn = Prepare ancilla set n
Cn = Couple ancilla set n with data
Dn = Decode ancilla set n
Mn = Measure ancilla set n
▨ = Move

FIG. 7. Parallel strategy for implementing the Steane-DiVincenzo-Aliferis algorithm with four sets of ancilla and two preparation rows. The need to reuse ancilla set one, as indicated by the red arrow, does not permit this set to be prepared for BT2 as early as possible, as indicated by the non-horizontal dashed line between BT1 and BT2.

B. Scheduled error correction: All-row preparation

Further decreases in the latency for the Steane-Shor and Steane-DiVincenzo-Aliferis algorithms can be achieved when three or more sets of ancillae are used by preparing every set in parallel. This all-row preparation allows as many stabilizer measurements as there are ancilla sets to be performed in rapid succession. However, preparing on three or more rows introduces delays between preparing the lower sets of ancillae and coupling them with the data. Here, the performance of the all-row strategy is studied for two, three, and six sets of ancillae. Four and five sets of ancilla are not considered because they are not commensurate with the total number of stabilizer measurements that must be performed.

The total latencies for the Steane-Shor and Steane-DiVincenzo-Aliferis algorithms for all-row preparation are shown in Figure 8. The latencies again decrease consistently with the addition of extra ancilla sets, due to the ability to prepare and measure ancilla sets in parallel. The biggest change from "on-demand" ancilla is that for Shor method, all ancilla rows are allowed to have verification. This removes the bottleneck seen in the two-row case. The parallelization advantage of Steane-Shor (moving ancilla qubits to data qubits for coupling while verification measurement is occurring) then gives it a lower latency than DiVincenzo-Aliferis for any number of ancilla sets.

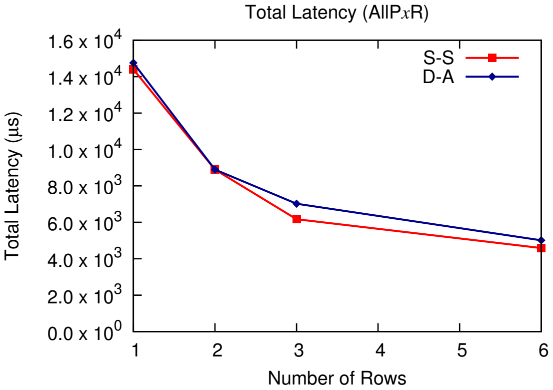

FIG. 8. Total latency for the Steane-Shor(S-S) and Steane-DiVincenzo-Aliferis(D-A) algorithms as a function of number of ancilla sets.

C. Gate time variation

The total latencies and the effectiveness of using spare ancilla sets to parallelize the algorithms vary with the gate times. For the case of two-row preparation, the latencies of the Steane-Shor and Steane-DiVincenzo-Aliferis algorithms as a function of a gate-time multiplier for various numbers of ancilla sets are shown in Figure 9(a) and (b) respectively. Latencies asymptotically approach a linear dependence on the gate time for large gate times. This is expected because the time spent on moves and measurements becomes negligible compared to the time spent on gates specifically controlled gates.

As the gate times get larger and dominate the total latency, run time improvement is dependent on the time spent on CNOT gates. The one-set strategy for the Steane-Shor algorithm requires 30 CNOT stages to be performed, where a stage is defined as one or more overlapping CNOT gates. In contrast, the two-row preparation strategy requires only 16 CNOT stages, dropping the latency almost in half. This is because more CNOT gates can be performed in overlapping pairs or triplets. The two-row preparation limit prevents any further significant latency reduction, since additional rows have to wait for a preparation row to clear out before they are prepared. In particular, having three ancilla sets reduces Steane-Shor to 14 CNOT stages, a very minor improvement over two ancilla sets, while having four to six ancilla sets offers no additional reduction beyond three sets.

For the DiVincenzo-Aliferis algorithm in the case two preparation rows, time saved with two ancilla sets follows a similar trend to that achieved with Steane-Shor algorithm. However, using more sets provides greater latency reduction since decoding is not subject to the two-row preparation limit. Similar to the Steane-Shor algorithm, the one-set strategy for the Steane-DiVincenzo-Aliferis algorithm requires 30 CNOT stages. Two rows reduces this to 16 CNOT stages, three rows to 12 stages, four rows to 11 stages, and five or six rows to 10 stages. Six rows reduces the latency a slight additional amount due to better parallel transport. The diminishing returns in adding ancilla rows for the case of long gate times can be seen in Figure 10(a).

All-row preparation results as a function of a gate-time multiplier for various numbers of ancilla sets are shown in Figure 11(a) and (b) respectively. These times again asymptotically approach a linear dependence on the gate time for large gate times, because of the dominance of gate time in the latency. In this case, Steane-Shor and Steane-DiVincenzo-Aliferis are nearly identical in latency. Without the preparation limit, both approaches have the exact

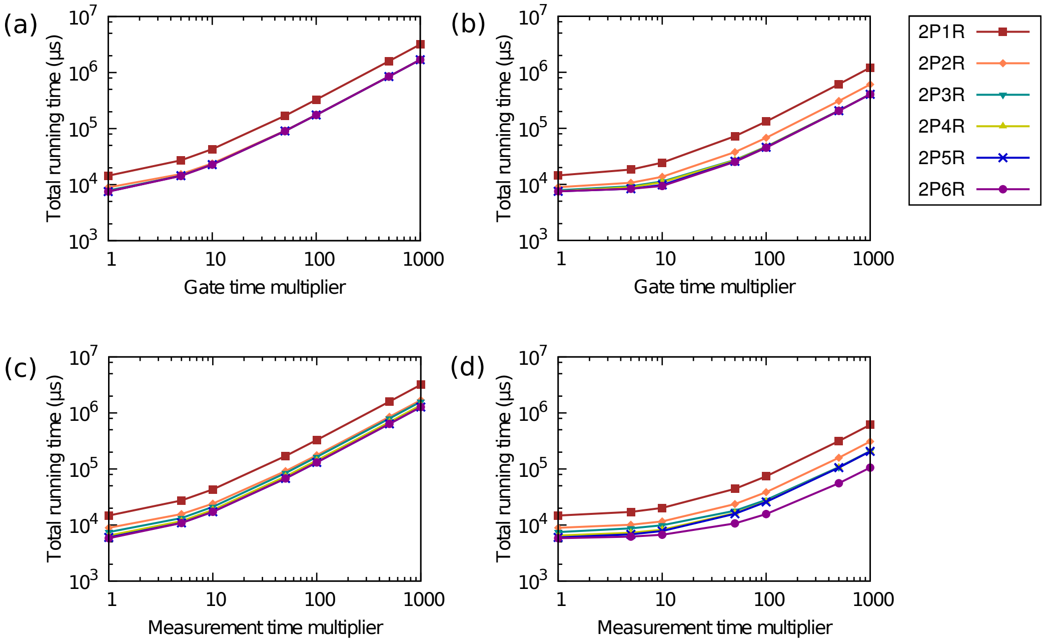

FIG. 9. Top: Total latencies for the (a) Steane-Shor and (b) Steane-DiVincenzo-Aliferis algorithms as a function of gate-time multiplier for various numbers of ancilla sets. Bottom: Total latencies for the (c) Steane-Shor and (d) Steane-DiVincenzo-Aliferis algorithms as a function of measurement-time multiplier for various numbers of ancilla sets. In all cases, preparation is limited to the top two ancilla rows.

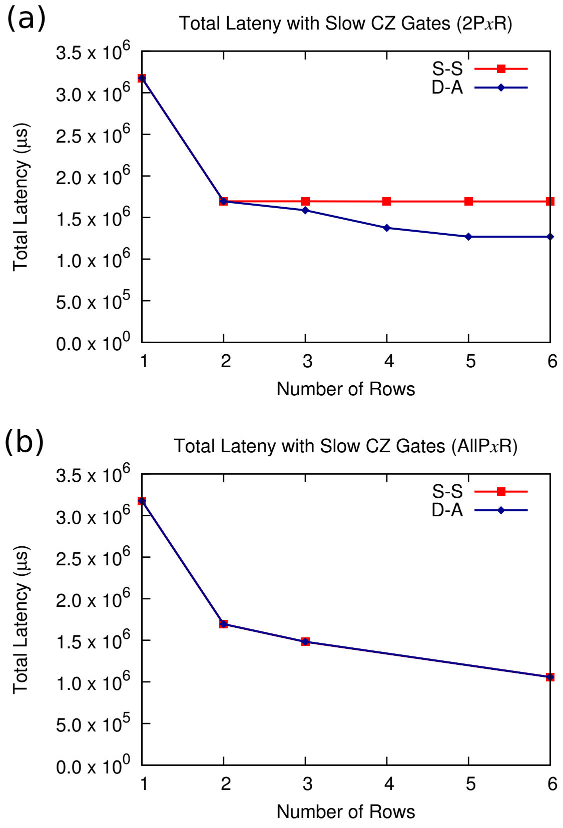

FIG. 10. Latency for Steane-Shor(S-S) and Steane DiVincenzo-Aliferis(D-A) circuits as a function of total number of ancilla rows for the case of CNOT execution time 1000 times longer than the default time. (a) Two preparation rows (2PxR); (b) ancilla can be prepared on all rows (AllPxR).

same number of CNOT stages for any given number of ancilla sets. Just as in the case of all-row preparation for default gate times described above, Steane-Shor has slightly lower latency due to parallelizing transport and verifier measurement. As the gate time is increased, this latency difference becomes vanishingly small. This behavior can be seen in Figure 10(b).

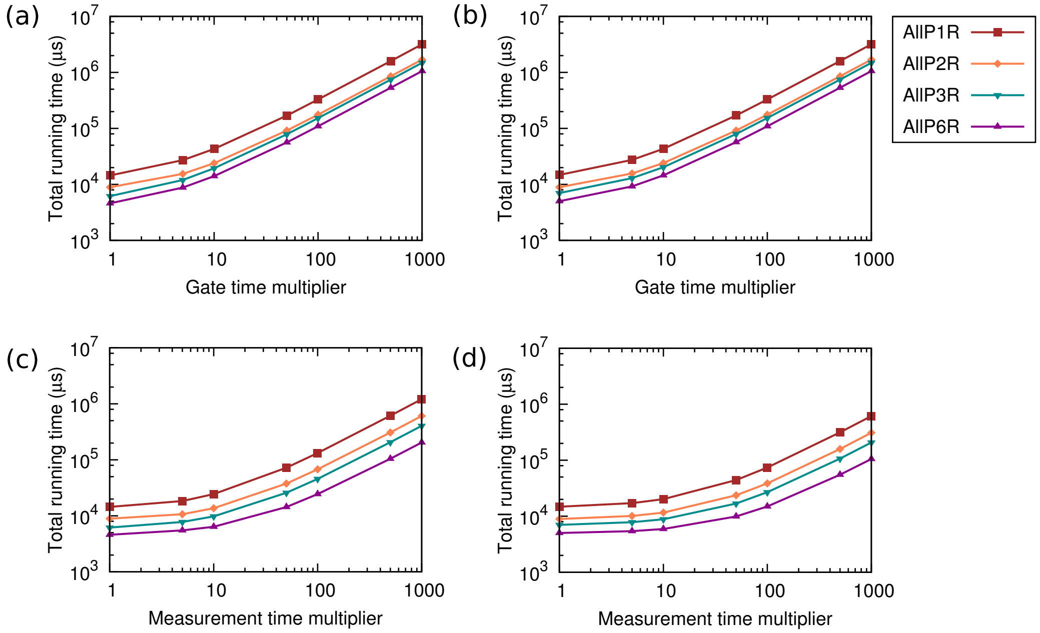

FIG. 11. Top: Total latency for (a) the Steane-Shor and (b) Steane-DiVincenzo-Aliferis algorithms as a function of gate-time multiplier for various numbers of ancilla sets. Bottom: Total latency for (c) the Steane-Shor and (d) Steane-DiVincenzo-Aliferis algorithms as a function of measurement-time multiplier for various numbers of ancilla sets. In all cases, the ancilla sets are prepared at-once (no limit on preparation).

D. Measurement time variation

The total latency and the effectiveness of using spare ancilla sets to parallelize the algorithms also vary with the measurement time. The results for the two-row-preparation Steane-Shor and Steane-DiVincenzo-Aliferis algorithms as a function of a measurement time multiplier for various numbers of ancilla sets are shown in Figure 9(c) and (d) respectively. Results for all-row-preparation are shown in Figure 11(c) and (d). As with increased gate time, the latency is dominated by measurement time for large measurement times. This is expected because the time spent on moves and gates becomes negligible compared to the time spent on measurements.

For both preparation types, the latency is reduced consistently as more sets of ancilla are used for each algorithm. As with the gate time variation, this is because more stabilizer measurements can be performed in parallel with more sets. The time saved also increases consistently as the measurement time increases. This increase indicates that more measurements are performed in parallel.

As in all other cases, Steane-Shor sees a saturation effect as the number of rows are increased. Increasing the measurement time puts more emphasis on the verification measurement bottleneck, since the ancilla-data controlled gate is not allowed to occur until after the verification measurement is completed. For extremely long measurement times and one ancilla set, Steane-Shor is effectively 12 measurement stages- one verification measurement and one ancilla measurement per stabilizer. For two ancilla sets, Steane-Shor is effectively 7 measurement stages. For three or more ancilla sets, it saturates at 4 measurement stages. For shorter measurement times, this behavior is moderated by transport and gate times, but the latency reduction still appears.

By contrast, for long measurement times, Steane-DiVincenzo-Aliferis sees a superior latency for even a single ancilla row, and a staggered decrease in latency for additional ancilla sets. This is not surprising, since handling long measurement times was the motivation for this scheme. For a single set and extremely long measurement times, Steane-DiVincenzo-Aliferis is effectively 6 measurement stages (one measurement per stabilizer). Using the two-row preparation strategy with two ancilla sets, measurement is reduced to 3 stages. Using three, four, or five sets reduces this to 2 measurement stages, and finally six sets reduces the latency to 1 measurement stage. This behavior simply comes from dividing the total number of measurement stages (six) by the number of ancilla sets, rounding up. Figure 12(a) shows the behavior of both encodings for long measurement times (1000 times longer than the default value).

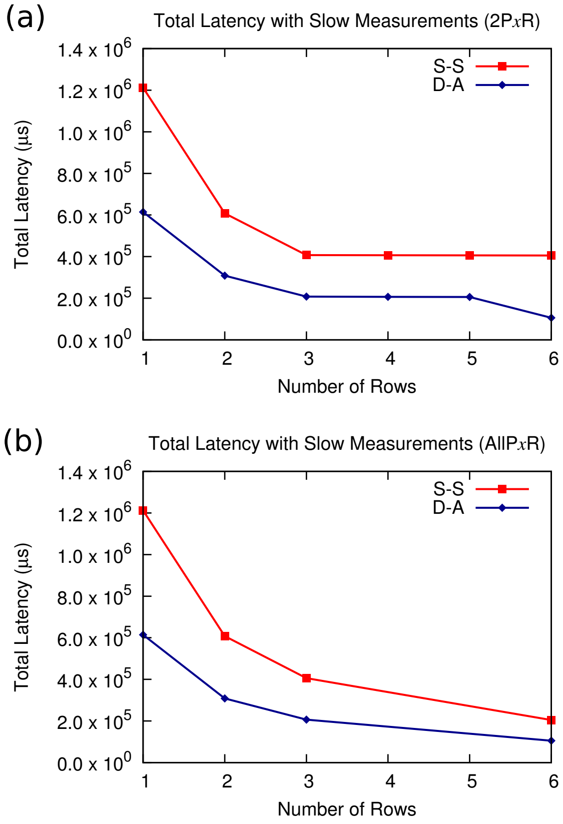

FIG. 12. Latency for Steane-Shor(S-S) and Steane-DiVincenzo-Aliferis(D-A) circuits as a function of total number of ancilla rows for the case of measurement execution time 1000 times longer than the default time. (a) Two preparation rows (2PxR); (b) ancilla can be prepared on all rows (AllPxR).

For the case of long measurement and all-row preparation, both Steane-Shor and Steane-DiVincenzo-Aliferis follow the same latency reduction: the algorithm is reduced to twelve or six measurement stages, respectively, which is divided by the number of number of ancilla sets rounded up. In all cases, DiVincenzo-Aliferis is superior to Shor method in terms of latency. This is shown in Figure 12(b).

E. Scheduled error correction: Two-row versus all-row preparation

Figure 13 shows the total execution time and logical error rates of Steane QECC with different numbers of ancilla qubits and their scheduling scheme. In order to to reduce the errors in syndrome measurements, we assume that we run the whole QECC circuits three times and the final syndromes are determined by the majority vote. This enables us to ignore one measurement error on a set of syndrome extraction.

As expected, it takes the longest to execute the whole QECC scheme when we keep only one set of ancilla in both Steane-Shor and Steane-DiVincenzo-Aliferis circuits. For Steane-Shor circuits, the number of preparation rows has the most influence on the execution time. Adding four more ancilla sets with 16 qubits only reduced the time by an additional 5% (2P2R → 2P6R), while adding more preparation rows with the same numbers of qubits reduced the execution time an additional 15 to 20% (2P3R → AllP3R and 2P6R → AllP6R). The execution time of the Steane-DiVincenzo-Aliferis circuits are more susceptible to numbers of available ancilla qubits. Adding four ancilla sets with the same number of preparation rows reduces the total time by 20% (2P2R → 2P6R).

The logical error rates are, however, almost constant regardless of number of ancilla qubits and preparation rows. We see that the errors on the measurements and the two-qubit gates dominate over the error on the idle and moving qubits. For example, the memory error rate on a qubit being idle for the total execution time of baseline Steane-Shor

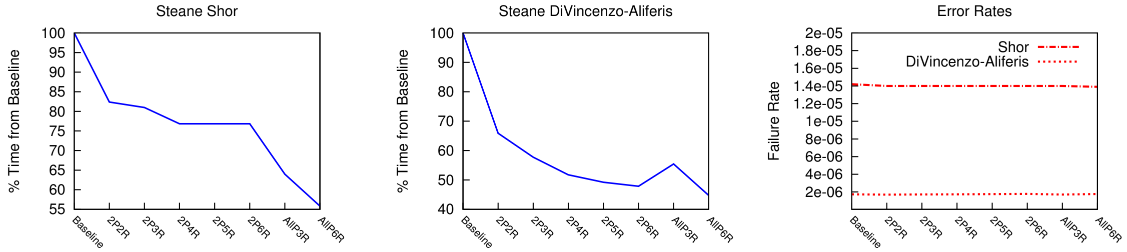

FIG. 13. Steane Shor and Steane-DiVincenzo execution time and error rates. Baseline schedule keeps only one set of ancillae and re-uses them by preparing it six times. We assume that the whole set of syndrome extraction is repeated three times to reduce the measurement errors. Execution times are shown relative to the baseline which are $5.1 \times 10^4 \mu s$ for the Steane-Shor circuit and $5.2 \times 10^4 \mu s$ for the Steane-DiVincenzo-Aliferis circuit.

schedule would be 2.75×10^{-5}. This is comparable to the error rate of a single CNOT gate, and each qubit in a Steane QECC encounters multiple CNOT gates.

Taken in total, the results suggest that increasing the number of simultaneous ancilla preparations has the greatest impact on QEC run times, without adversely affecting the QEC error rate. It also shows that the Steane-DiVincenzo-Aliferis algorithm is equivalent or superior to the Steane-Shor algorithm, particularly for long measurement times, given the error model presented in this paper. This is attributable to reducing (for simultaneous preparation) or removing (for DiVincenzo-Aliferis) the ancilla verification bottleneck after preparation.

V. CONCLUSION

We examined changes in execution time and logical error rates of Steane QECC by varying the number of ancilla qubits and how they are scheduled on the ion trap architecture. We identified possible resource bottlenecks and opportunities for parallelism in preparing blocks of Shor ancilla. After studying both standard Shor and DiVincenzo-Aliferis ancilla for a variety of multiple ancilla set preparations, we found that one-time ancilla preparation was superior to on-demand preparation. This is attributed to the time-intensive process of ancilla preparation driving QEC latency. On-demand ancilla preparation limits the speed of the QEC round to the speed of sequential ancilla preparations, particular for verification schemes like Steane-Shor. In comparing the Steane-Shor and Steane-DiVincenzo-Aliferis latencies, we found that for the case of a single ancilla set with roughly equivalent gate, measurement, and transport times, Steane-Shor has slightly lower latency. This is due to the ability to perform parallel operations on the verification and ancilla qubit. This also holds true for multiple ancilla sets with one-time preparation. For on-demand preparation, verification becomes a bottleneck significantly slowing down Steane-Shor compared to Steane-DiVincenzo-Aliferis as ancilla sets are added. When gate times are increased, Steane-Shor and Steane-DiVincenzo-Aliferis become effectively identical, as they both have the same number of parallel CNOT operations. As measurement times are increased, Steane-DiVincenzo-Aliferis shows a much lower latency, as expected.

The results presented are based on an ion trap description with optimistic error rates, but pessimistic gate and movement times. The long times required for error correction in this paper could be improved in a number of ways. For example by using ultrafast lasers, single qubit gate times as fast as 50 ps have been achieved [36] and two-qubit gate times can in principle be considerably improved [37, 38]. Transport in the model is limited to 1 m/s but recent experiments have shown that a transport speed of 40-80 m/s can be achieved while still controlling the quantum states of the ion motion [39, 40].

In the future, we plan to extend this analysis to other quantum error-correcting codes including non-CSS type codes. We can also apply the same methods to study performance of various quantum architectures and to determine whether there exists affinity between certain devices and types of codes. The QMP method is flexible enough to handle a wide array of qubit architectures and couplings. Future improvements to the QCFT method will allow us to approximate error rates for circuits beyond those generated by Clifford operators.

ACKNOWLEDGMENTS

This work was supported by the Office of the Director of National Intelligence - Intelligence Advanced Research Projects Activity through Department of Interior contract D11PC20167.

[1] K. M. Svore, D. P. DiVincenzo, and B. M. Terhal, Quantum Information & Computation **7**, 297 (2007).

[2] R. Raussendorf and J. Harrington, Phys. Rev. Lett. **98**, 190504 (2007), URL `http://link.aps.org/doi/10.1103/PhysRevLett.98.190504`.

[3] A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A **80**, 052312 (2009), URL `http://link.aps.org/doi/10.1103/PhysRevA.80.052312`.

[4] A. Cross, Master's thesis, Massachusettes Institute of Technology, Cambridge, MA (2005).

[5] D. Aharonov, M. Ben-or, R. Impagliazzo, and N. Nisan, arXiv:quantph/9611028v1 (1996).

[6] A. M. Steane, Phys. Rev. Lett. **77**, 793 (1996), URL `http://link.aps.org/doi/10.1103/PhysRevLett.77.793`.

[7] D. P. DiVincenzo and P. W. Shor, Phys. Rev. Lett. **77**, 3260 (1996), URL `http://link.aps.org/doi/10.1103/PhysRevLett.77.3260`.

[8] Y. S. Weinstein, Phys. Rev. A **84**, 012323 (2011), URL `http://link.aps.org/doi/10.1103/PhysRevA.84.012323`.

[9] Y. S. Weinstein and S. D. Buchbinder, Phys. Rev. A **86**, 052336 (2012), URL `http://link.aps.org/doi/10.1103/PhysRevA.86.052336`.

[10] D. P. DiVincenzo and P. Aliferis, Phys. Rev. Lett. **98**, 020501 (2007), URL `http://link.aps.org/doi/10.1103/PhysRevLett.98.020501`.

[11] D. Kielpinski, C. R. Monroe, and D. J. Wineland, Nature **417**, 709 (2002).

[12] S. A. Schulz, U. Poschinger, F. Ziesel, and F. Schmidt-Kaler, New Journal of Physics **10**, 045007 (2008), URL `http://stacks.iop.org/1367-2630/10/i=4/a=045007`.

[13] F. Splatt, M. Harlander, M. Brownnutt, F. Zähringer, R. Blatt, and W. Hänsel, New Journal of Physics **11**, 103008 (2009), URL `http://stacks.iop.org/1367-2630/11/i=10/a=103008`.

[14] J. M. Amini, H. Uys, J. H. Wesenberg, S. Seidelin, J. Britton, J. J. Bollinger, D. Leibfried, C. Ospelkaus, A. P. VanDevender, and D. J. Wineland, New Journal of Physics **12**, 033031 (2010), URL `http://stacks.iop.org/1367-2630/12/i=3/a=033031`.

[15] M. D. Hughes, B. Lekitsch, J. A. Broersma, and W. K. Hensinger, Contemporary Physics **52**, 505 (2011), URL `http://www.tandfonline.com/doi/abs/10.1080/00107514.2011.601918`.

[16] D. L. Moehring, C. Highstrete, D. Stick, K. M. Fortier, R. Haltli, C. Tigges, and M. G. Blain, New Journal of Physics **13**, 075018 (2011).

[17] R. B. Blakestad, C. Ospelkaus, A. P. VanDevender, J. H. Wesenberg, M. J. Biercuk, D. Leibfried, and D. J. Wineland, Phys. Rev. A **84**, 032314 (2011).

[18] J. T. Merrill, C. Volin, D. Landgren, J. M. Amini, K. Wright, S. C. Doret, C.-S. Pai, H. Hayden, T. Killian, D. Faircloth, et al., New Journal of Physics **13**, 103005 (2011), URL `http://stacks.iop.org/1367-2630/13/i=10/a=103005`.

[19] S. C. Doret, J. M. Amini, K. Wright, C. Volin, T. Killian, A. Ozakin, D. Denison, H. Hayden, C.-S. Pai, R. E. Slusher, et al., New Journal of Physics **14**, 073012 (2012), URL `http://stacks.iop.org/1367-2630/14/i=7/a=073012`.

[20] K. Wright, J. Amini, D. Faircloth, C. Volin, S. Doret, H. Hayden, C.-S. Pai, D. Landgren, D. Denison, T. Killian, et al., arXiv:quantph/1210.3655 (2012).

[21] N. Isailovic, M. Whitney, Y. Patel, and J. Kubiatowicz, SIGARCH Comput. Archit. News **36**, 177 (2008), ISSN 0163-5964, URL `http://doi.acm.org/10.1145/1394608.1382137`.

[22] T. Metodi, D. Thaker, A. Cross, F. Chong, and I. Chuang, in *MICRO-38: Proc. 38TH Annual IEEE/ACM Int. Symp. on Microarchitecture* (2005), pp. 305–316.

[23] C. R. Clark, T. S. Metodi, S. D. Gasster, and K. R. Brown, Phys. Rev. A **79**, 062314 (2009), URL `http://link.aps.org/doi/10.1103/PhysRevA.79.062314`.

[24] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, arXiv:quant-ph/1208.0391 (2012).

[25] P. J. Salas and A. L. Sanz, Phys. Rev. A **69**, 052322 (2004), URL `http://link.aps.org/doi/10.1103/PhysRevA.69.052322`.

[26] A. M. Steane and B. Ibinson, Phys. Rev. A **72**, 052335 (2005), URL `http://link.aps.org/doi/10.1103/PhysRevA.72.052335`.

[27] K. Svore, A. Aho, A. Cross, I. Chuang, and I. Markov, Computer **39**, 74 (2006).

[28] T. S. Metodi, D. Thaker, A. W. Cross, F. T. Chong, and I. L. Chuang, in *Proceedings of the SPIE Defense and Security Symposium* (2005).

[29] A. M. Steane, Phys. Rev. A **68**, 042322 (2003), URL `http://link.aps.org/doi/10.1103/PhysRevA.68.042322`.

[30] The ion trap PMD is part of the government furnished information that IARPA has provided for official use only. The necessary data to reproduce the results are contained in this paper.

[31] M. A. Rowej, A. Ben-Kish, B. Demarco, D. Leibfried, V. Meyer, J. Beall, J. Britton, J. Hughes, W. M. Itano, B. Jelenkovic, et al., Quantum Information & Computation **2**, 257 (2002).

[32] J. P. Home and A. M. Steane, Quantum Information & Computation **6**, 289 (2006).

[33] N. Nilsson, *Problem Solving Methods in Artificial Intelligence* (McGraw Hill, 1971).

[34] J. Pearl, *Heurestics: Intelligent Search Strategies for Computer Problem Solving* (Addison Wesley, 1984).

[35] P. Aliferis, D. Gottesman, and J. Preskill, Quantum Information & Computation **6**, 97 (2006).

[36] W. C. Campbell, J. Mizrahi, Q. Quraishi, C. Senko, D. Hayes, D. Hucul, D. N. Matsukevich, P. Maunz, and C. Monroe, Phys. Rev. Lett. **105**, 090502 (2010), URL `http://link.aps.org/doi/10.1103/PhysRevLett.105.090502`.

[37] J. J. Garcia-Ripoll, P. Zoller, and J. I. Cirac, Phys. Rev. Lett. **91**, 157901 (2003), URL `http://link.aps.org/doi/10.1103/PhysRevLett.91.157901`.

[38] C. D. B. Bentley, A. R. R. Carvalho, D. Kielpinski, and J. J. Hope, New Journal of Physics **15**, 043006 (2013), URL `http://stacks.iop.org/1367-2630/15/i=4/a=043006`.

[39] R. Bowler, J. Gaebler, Y. Lin, T. R. Tan, D. Hanneke, J. D. Jost, J. P. Home, D. Leibfried, and D. J. Wineland, Phys. Rev. Lett. **109**, 080502 (2012), URL `http://link.aps.org/doi/10.1103/PhysRevLett.109.080502`.

[40] A. Walther, F. Ziesel, T. Ruster, S. T. Dawkins, K. Ott, M. Hettrich, K. Singer, F. Schmidt-Kaler, and U. Poschinger, Phys. Rev. Lett. **109**, 080501 (2012), URL `http://link.aps.org/doi/10.1103/PhysRevLett.109.080501`.