npj | Quantum Information

www.nature.com/npjqi

ARTICLE OPEN

Approximate quantum Fourier transform with $O(n\log(n))$ T gates

Yunseong Nam[1]✉, Yuan Su[2]✉ and Dmitri Maslov[3]✉

The ability to implement the Quantum Fourier Transform (QFT) efficiently on a quantum computer facilitates the advantages offered by a variety of fundamental quantum algorithms, such as those for integer factoring, computing discrete logarithm over Abelian groups, solving systems of linear equations, and phase estimation, to name a few. The standard fault-tolerant implementation of an n-qubit unitary QFT approximates the desired transformation by removing small-angle controlled rotations and synthesizing the remaining ones into Clifford+T gates, incurring the T-count complexity of $O(n\log^2(n))$. In this paper, we show how to obtain approximate QFT with the T-count of $O(n\log(n))$. For brevity, the above figures omit the dependence on the approximation error ε, assuming the error is fixed. Our approach relies on quantum circuits with measurements and feedforward, and on reusing a special quantum state that induces the phase gradient transformation. We report asymptotic analysis as well as concrete circuits, demonstrating significant advantages in both theory and practice.

npj Quantum Information (2020)6:26 ; https://doi.org/10.1038/s41534-020-0257-5

INTRODUCTION

Quantum Fourier Transform (QFT) is one of the most important operations in quantum computing. It can extract the periodicity encoded in the amplitudes of a quantum state, which is employed by an efficient algorithm for integer number factoring, widely known as Shor's algorithm[1]. Shor's integer factoring algorithm can be generalized (while still relying on the QFT) into a polynomial-time algorithm for the discrete logarithm problem over Abelian groups[1]. The importance of the above is witnessed through the threat such algorithms pose to modern public-key cryptosystems, such as the RSA or the ECC. Using the QFT as a subroutine, the eigenphase of a black-box unitary can be estimated up to an arbitrary precision[2], which may be used to estimate quantum amplitudes[3,4], simulate quantum chemistry/dynamics[5], find the ground state/energy of a Hamiltonian[6], compute Hessian to optimize molecular geometry[7], exponentiate unitaries[8], construct fractional powers of the QFT using constantly many copies of the controlled-QFT[8,9], extract features of the solution of linear systems[10], and more. QFT has also been used in quantum arithmetics[11,12] and quantum cryptography[13].

QFT can be implemented approximately by removing all rotation gates with angles smaller than a certain threshold value, resulting in the Approximate QFT (AQFT). In practice, it was shown that it suffices to apply AQFT with ~5.3×10^4 controlled rotation gates to factor 2048-digit numbers (reflecting the de facto key size for today's standard[14]) with a high expected algorithmic accuracy ($\gtrsim$99.992%)[15]. AQFT has been studied extensively in the literature. The robustness of the quantum computer equipped with the AQFT was investigated in detail[16–20]. A study of the optimal level of the approximation of the AQFT in the presence of certain errors may be found in ref. [21]. Implementation of the QFT and its approximate version over restricted architectures was addressed in refs [22,23]. An efficient approximate implementation of the AQFT that harnesses certain quantum hardware features was also investigated[24].

Quantum information is fragile, and it is generally accepted that the implementation of large quantum algorithms must rely on the fault-tolerant computations. Fault tolerance suppresses the errors at the cost of using multiple physical qubits to encode a single logical qubit. Fault-tolerant computations must furthermore rely on a quantum gate library consisting of those gates that are constructible fault tolerantly. A standard choice for such a computationally universal gate library is Clifford+T. Within known fault tolerance approaches, Clifford gates can generally be implemented with the relative ease, frequently transversally. On the other hand, a non-Clifford gate typically does not admit such an implementation; for instance, a T gate may be implemented fault tolerantly by distilling a certain quantum state and then teleporting it into the gate[25]. A T gate is indeed far more costly than any of the Clifford gates, and therefore efficient fault-tolerant circuits must minimize the T-count.

To implement an n-qubit AQFT to within a certain fixed error fault-tolerantly, the standard approach is to approximate the desired transformation by removing small-angle controlled rotations to bring down the gate count from $O(n^2)$ [ref. [26], page 219] to $O(n\log(n))$, and then replace the remaining $O(n\log(n))$ controlled rotations with their Clifford+T implementations. The resulting circuit has the T-count of $O(n\log^2(n))$. Only in the special case of the semiclassical version of AQFT[27], where the AQFT transform is followed by the measurement, the T-count of $O(n\ \log(n))$ implementation is known[28]. In contrast, in this paper, we focus on the fully coherent AQFT.

We develop a more efficient implementation with the T-count complexity of $O(n\log(n))$ for the general case of fully coherent AQFT, improving over the standard construction by a factor of $O(\log(n))$. Including the dependence on the approximating error ε results in the reduction of complexity from $O(n\log(n/\varepsilon)\log(\frac{n\log(n/\varepsilon)}{\varepsilon}))$, assuming the error budget is split equally between the approximation of the QFT itself and the approximation by Clifford+T library, and evenly across gates

[1] IonQ, College Park, MD 20740, USA. [2] Department of Computer Science, Institute for Advanced Computer Studies, and Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD 20740, USA. [3] IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA. ✉email: nam@ionq.co; buptsuyuan@gmail.com; dmitri.maslov@gmail.com

Published in partnership with The University of New South Wales

needing the decomposition into Clifford+T, to $O(n\log(n/\varepsilon)+\log(n/\varepsilon)\log(\frac{\log(n/\varepsilon)}{\varepsilon}))$. We drop the dependence on ε in most discussions to improve the readability. Our results show that, in general and regardless of the amenability to the semiclassical approach, the AQFT may be implemented with $O(n\log(n))$ T gates. This allows for the efficient implementation of the AQFT in any quantum algorithm, including those that use the AQFT as subroutines in the midst of the quantum computation[5,7,10,12,13]. Since our implementation is more involved compared to the standard, we also make a separate effort to show that the constant factor and small-order additive terms missing in the asymptotic analyses but otherwise present in our construction do not prevent it from achieving a significant practical advantage.

RESULTS AND DISCUSSION

We start with a high-level description of our result, and delay the detailed discussion of algorithmic advantages and further low-level optimizations offered by the final circuits to the following subsections.

The entry point for our construction is the standard textbook implementation of the QFT circuit [ref. [26], page 219] using $O(n^2)$ parametrized controlled-Z^a rotations, where $a\in\{1/2, 1/4, \ldots, 1/2^{n-1}\}$, and n Hadamard gates. Recall that the AQFT may be obtained from the textbook circuit by simply discarding the rotations with parameter a below a certain threshold, keeping only b controlled rotations per layer, with parameter b scaling logarithmically with n (see Fig. 1 for an illustration). A standard fault-tolerant implementation of AQFT with $\sim n\log(n)$ (removing lower order terms, and for simplicity furthermore assuming n stages of $\log(n)$ gates) parametrized controlled rotations, choosing $b=\log(n)$ for simplicity and to remove the dependence on the approximation error, uses $\sim 24n\log^2(n)$ T gates since 8 T gates are employed to map controlled rotations into uncontrolled ones [ref. [30], Fig. 10], and $\sim 3\log(n)$ T gates are needed to approximate the uncontrolled rotations[35].

We optimize the above implementation by first noting that mapping controlled rotations into uncontrolled ones may be done using only 4 T gates. This reduces the T gate count to $\sim 12n\log^2(n)$. We next notice that the uncontrolled rotations come in layers, and thus can be induced via adder, given access to a $\log(n)$-qubit gradient state[32]. Using an efficient b-bit integer adder[31] with ~$4b$ T gates allows to reduce the T gate requirement from $\sim 12n\log^2(n)$ to $\sim 8n\log(n)+3\log^2(n)$, where $8=4+4$ T gates are employed to remove the control (4 T gates) and integer-add the target (4 T gates) per each controlled rotation, and $3\log(n)$ T gates[35] are used on each of $\log(n)$ qubits to synthesize the $\log(n)$-qubit gradient state, that is then reused. This is the most significant reduction giving improvements in both asymptotic analysis and gate counts. We next apply RUS circuits to reduce the cost of state generation by a factor of about 2.5[33], leading to $\sim 8n\log(n)+1.2\log^2(n)$ T gates and find local optimizations worth of ~$8n$ T gates further bringing down the T gate cost to the final figure of $\sim 8n(\log(n)-2)+1.2\log^2(n)$, compared to the original $\sim 24n\log^2(n)$.

Details of the construction

We start with an n-qubit AQFT whose construction relies on $O(nb)$ controlled-Z^a gates with

$$Z^a := \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi a} \end{bmatrix},$$

where $a\in\{1/2, 1/4, \ldots, 1/2^b\}$, for $b:=\lceil\log n\rceil$, and n Hadamard (H) gates (see Fig. 1 for an illustration with $n=6$ and $b=3$). Such a choice of b implies a very specific approximation error ε, whose analysis will be detailed in the next section. We unite the individual controlled rotations into $n-1$ sets separated by the H gates, such as illustrated in Fig. 1.

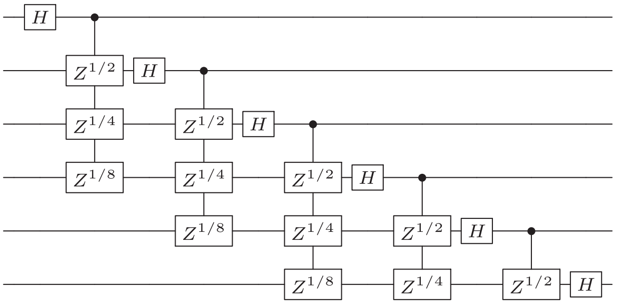

Fig. 1 AQFT with $n=6$ and $b=3$. Note that each of the $n-1$ sets of controlled-z^a gates are separated by the H gates.

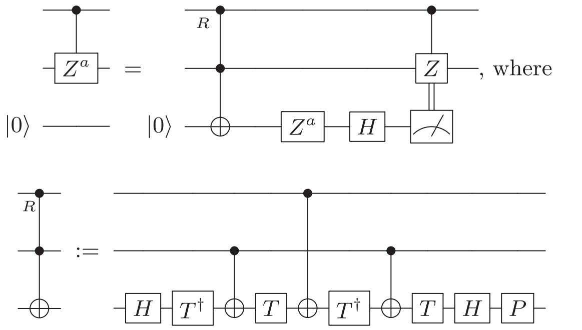

Fig. 2 Ancilla-aided, measurement/feedforward-based fault-tolerant controlled-Z^a gate. This construction improves on the known state-of-the-art in the quantum resource requirement (see main text for detail), while enabling to decouple the control from the target, important for further optimization.

To implement a given controlled-Z^a rotation, we map its real-valued degree of freedom into that of the uncontrolled power of Pauli-Z, such as shown in Fig. 2. This implementation was developed by combining Kitaev's trick[2] with Toffoli-measurement construction of Jones [29] with our own choice of the relative the phase Toffoli gate, and custom circuit simplifications. Our circuit improves over the one reported in [ref. [30], Fig. 10] (note that the middle T gate in [ref. [30], Fig. 10] can be replaced with the Z^a gate) by 4 T gates (8 ↦ 4), 9 CNOT gates (12 ↦ 3), 1 H gate (4 ↦ 3), and 1 Phase (P) gate (2 ↦ 1) at the cost of introducing 1 measurement and 1 classically-controlled controlled-Z operation. Note that the fault-tolerant cost of those operations introduced is significantly lower than that of a single T gate, as the construction of the T gate itself requires both a measurement and a classically controlled quantum correction[25].

We now group the uncontrolled Z^a rotations into one layer (time slice), as shown in Fig. 3. This layer applies the transformation that was coined the phase gradient operation in[31], the induction of which by the addition circuit was first reported in ref. [32]. Such a transformation can be implemented by a b-bit adder at the cost of $4b+O(1)$ T gates[31], so long as one has access to a special quantum state $|\psi_{b+1}\rangle := \frac{1}{\sqrt{2^{b+1}}}\sum_{j=0}^{2^{b+1}-1} e^{-2\pi ij/2^{b+1}}|j\rangle$. The quantum state $|\psi_{b+1}\rangle$ can be reused to induce phase gradient transformations in all $n-1$ sets of controlled-Z^a rotations. A schematic circuit diagram of our AQFT implementation is shown in Fig. 4.

To construct the special $(b+1)$-qubit state $|\psi_{b+1}\rangle$, we first apply H gates to the quantum register $|00\ldots0\rangle$ and then exercise the gates Z, $Z^{-1/2}$, …, $Z^{-1/2^b}$. The latter step is accomplished via approximating each Z^a by RUS circuits[33]. Specifically, we approximate complex number $e^{i\pi a}$ by z^*/z, where $z\in\mathbb{Z}[\omega]$ with

$\omega := e^{i\pi/4}$ being the cyclotomic integer obtained from the PSLQ Algorithm[34]. We choose $r \in \mathbb{Z}[\sqrt{2}]$ randomly and search the solution $y \in \mathbb{Z}[\omega]$ of the norm equation $|y|^2 = 2^L - |rz|^2$ with $L = \lceil \log(|\mathrm{rz}|^2) \rceil$[35], such that $V := \frac{1}{\sqrt{2}^L}\begin{pmatrix} rz & y \\ -y^* & rz^* \end{pmatrix}$ is a unitary. We exactly synthesize the two-qubit gate $\begin{pmatrix} V & 0 \\ 0 & V^\dagger \end{pmatrix}$into a Clifford+T circuit[33,36]. Upon measuring the second qubit and obtaining 0, the gate Z^a is successfully implemented. Otherwise, a Z error takes place and can be reversed at zero cost in the T gate count. The expected number of repetitions until success is $2^L/|rz|^2$. We resorted to using this more complex algorithm as opposed to the simpler one given by refs [35,36], as we already use quantum circuits with measurements and feedforward elsewhere in our constructions, and the RUS approach results in about 2.5-fold improvement[33] in the number of the T gates required to obtain the desired Z^a.

Local optimization

Here we describe a local optimization of the AQFT circuit developed above, exploiting the fact that controlled-P and controlled-T gates have a special implementation, due to both P and T gates being a part of the Clifford+T library.

We start by noting that the controlled-P gate may be implemented by two CNOT gates and three T gates (including inverses) as shown in Fig. 5. We know from our construction above that each controlled-Z^a gate in the AQFT is implemented using 8 T gates (of which 4 are used to remove the control, and 4 to implement the target via the adder). Therefore, instead of relying on inducing the gradient operation through the adder, we implement controlled-P gates directly, according to Fig. 5.

Next, we consider controlled-T gates. As per Fig. 1, we see that each controlled-T gate in the AQFT neighbors a controlled-P gate in the following layer of controlled-Z^a gates in the target qubit line. Since we implement controlled-P gates according to Fig. 5, we may obtain T-count savings via gate cancellation ($\mathrm{TT}^\dagger = Id$) by rewriting the controlled-T gate as the controlled-$\mathrm{Z}^{3/4}$ gate followed by the controlled-$\mathrm{Z}^{-1/2}$, where the controlled-$\mathrm{Z}^{-1/2}$ gate is implemented according to Fig. 5, inducing T-count reduction by 2 on the 'target' of controlled-$\mathrm{Z}^{-1/2}$ and controlled-T gates, and by another 2 for each layer of controlled-Z^a gates by cancellations on the 'control' line, and the controlled-$\mathrm{Z}^{3/4}$ gate is implemented directly as per the top panel of Fig. 2, which costs 5 T gates.

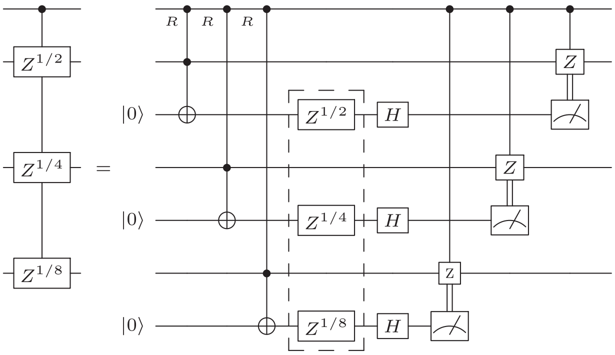

Fig. 3 A 4-qubit example of the layer of controlled-z^a gates. The uncontrolled rotations are grouped together to induce the phase gradient operation[31,32].

Altogether, the above implementation of the controlled-T and controlled-P gate pair requires $7(= 5 + 3 + 3 - 2 - 2)$ T gates. This is in comparison to 16 T gates that would otherwise have been used by the implementation based on the adder. What remains to be investigated at this point is the modification that needs to be made to the gradient operation so as to induce a partial gradient operation, i.e., $|k\rangle|\psi_{d+1,b+1}\rangle \mapsto e^{2\pi ik/2^{b+1}}|k\rangle|\psi_{d+1,b+1}\rangle$, where $k < 2^{b-d}$, $d \le b$, and $|\psi_{d+1,b+1}\rangle$ is the state $|\psi_{b+1}\rangle$ without first $d+1$ qubits, to implement the remaining Z^a gates in a layer.

To obtain the partial gradient operation, we analyze how the gradient operation works. Firstly, we formally define the state $|\psi_{d+1,b+1}\rangle := \frac{1}{\sqrt{2^{b-d}}}\sum_{j=0}^{2^{b-d}-1} e^{-2\pi ij/2^{b+1}}|j\rangle$. The application of $(b-d)$-bit addition (see ref. [31]) to $|k\rangle|\psi_{d+1,b+1}\rangle$ results in two cases: $k + j < 2^{b-d}$ and $k + j \ge 2^{b-d}$. In order for the partial gradient operation to work, we need $k + j \mapsto k + j \bmod 2^{b-d}$. This may be achieved by applying $\mathrm{Z}^{1/2^d}$ gate to the most significant bit of the modular addition circuit. Since in our case $d = 2$, this amounts to applying a T gate for each gradient operation. This means that the overall result of our optimization detailed in this section is by about $8(n-2)$ T gates.

Comparisons to prior work

Our improved implementation of AQFT_n with $n > b > 2$ requires the qubit count of $n_q = n + 3b - 4$, the CNOT-gate count of $7.5n - 13 + \sum_{l=3}^{n-1}(16\min(b-2, l-2) - 5) + \sum_{b'=3}^{\min(b,n-1)} C_{\mathrm{CNOT}}(\mathrm{RUS}_{b'})/p_{b'}$, and the T-count of $7n - 11 + \sum_{l=3}^{n-1}(8\min(b-2, l-2) + 1) + \sum_{b'=3}^{\min(b,n-1)} C_{\mathrm{T}}(\mathrm{RUS}_{b'})/p_{b'}$, where $C_g(\mathrm{RUS}_{b'})$ denotes the count of the fault-tolerant gate g in the RUS circuit synthesizing $z^{-1/2^{b'}}$, and $p_{b'}$ denotes the success probability of the RUS circuit. As follows from our constructions, the T gate count can be fairly accurately approximated by the simple formula $8n(b-1)$. This may be compared to the previous state of the art that uses a variant of [ref. [30], Fig. 10] to implement the controlled-Z^a, which requires $n_q = n + 1$ qubits, the CNOT gate count of $12 \cdot \sum_{l=0}^{n-1}\min(b, l)$, and the T-count of $3(n-1) + \sum_{b'=2}^{\min(b,n-1)}(n-b')[C_{\mathrm{T}}(\mathrm{Gridsynth}_{b'}) + 8]$, where C_{T}(Gridsynth) is the T-count of the Gridsynth algorithm [35] synthesizing $z^{1/2^{b'}}$ and $C_{\mathrm{T}} = 1$ when considering $z^{\pm 1/4}$ gate.

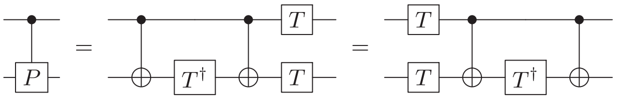

Fig. 5 Direct implementation of the controlled-P gate. These constructions also work when all Z-axis gates are replaced by their complex conjugates.

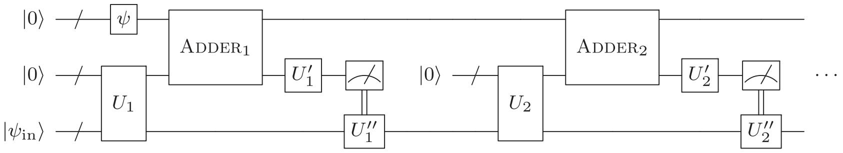

Fig. 4 A schematic diagram of the full implementation of the fault-tolerant AQFT. ψ denotes the preparation of the special state $|\psi_{b+1}\rangle$. U_i illustrate the operations that precede the ith adder, including H gates and the relative phase Toffoli gates used to map controlled-Z^a into uncontrolled Z^a rotations. U_i' denotes the operations that follow the adder up to the in-circuit measurements. ADDER$_i$ denotes the ith adder. U_i'' are the classically controlled controlled-Z gates, applied at the ith step.

Table 1. Quantum resource counts for implementing an n-qubit AQFT with $b=13$. n_q denotes the number of qubits required to execute the corresponding circuit. Columns CNOT and T report the number of respective gates in the circuits. All circuits are available in ref. [38].

	Our AQFT implementation			AQFT with controlled-z^a per[30] (Fig. 10)			Optimized AQFT[37]		
Circuit	n_q	CNOT	T	n_q	CNOT	T	n_q	CNOT	T
AQFT_8	25	390	303	9	336	1083	8	56	1821
AQFT_{16}	51	1798	1162	17	1404	6309	16	234	7815
AQFT_{32}	67	4654	2698	33	3900	19,261	32	650	22,683
AQFT_{64}	99	10,366	5770	65	8892	47,099	64	1482	54,269
AQFT_{128}	163	21,790	11,914	129	18,876	106,631	128	3146	123,333
AQFT_{256}	291	44,638	24,202	257	38,844	229,729	256	6474	267,007
AQFT_{512}	547	90,334	48,778	513	78,780	476,873	512	13,130	553,277
AQFT_{1024}	1059	181,726	97,930	1025	158,652	993,727	1024	26,442	1,148,497
AQFT_{2048}	2083	364,510	196,234	2049	318,396	2,084,983	2048	53,066	2,427,081
AQFT_{4096}	4131	730,078	392,842	4097	637,884	4,316,993	4096	106,314	4,993,035
Complexity	$O(n)$	$O(n\log(n))$	$O(n\log(n))$	$O(n)$	$O(n\log(n))$	$O(n\log^2(n))$	$O(n)$	$O(n\log(n))$	$O(n\log^2(n))$

For a concrete comparison with the previous state of the art[30,37] at the gate-by-gate level, we implemented our improved fault-tolerant construction as described in Section II B in software. We synthesized the RUS circuits for z^a gates with $a\in\{-1/2^3,-1/2^4,\ldots,-1/2^{13}\}$, motivating the choice of the smallest angle $\pi/2^b$ by that sufficient to launch a quantum attack on the classically-infeasible instance of the integer factoring problem corresponding to cracking the RSA-2048. We also chose the overall fault-tolerance error that arises from the gate synthesis to be below 1.1×10^{-4} for all sizes of the AQFT ($n\leq4096$ and $b=13$) we considered. In particular, we chose the error 10^{-5} per z^a gate approximation for our improved construction. This amounts to the gate-synthesis error budget of ~$10^{-5}/n$ per rotation for the previous state-of-the-art AQFT circuit. The improvement of the accuracy per Z^a gate is justified by the fact that our implementation of the AQFT requires the approximation of only $O(b)$ rotations instead of $O(nb)$ in the previous constructions.

Summary of the resulting quantum resource cost of our improved AQFT implementation is shown in Table 1. We included a comparison of the gate costs of our implementation to those circuits known previously: first set relying on [ref. [30], Fig. 10] to implement controlled-Z^a gates in the AQFT and the second set resulting from an automated AQFT circuit optimization[37]. For both implementations, we used Gridsynth algorithm[35] to synthesize Z^a gates. Note that our implementation carries a significant practical advantage, saving quantum resource cost in the form of the T-count by a factor of as large as 12 (AQFT_{4096} with $b=13$). The slight increase in n_q and the CNOT gate counts are completely offset by the savings in the T-count in the fault-tolerant regime.

Complexity analysis

The total T-count in our AQFT circuit is $8n(b-1)+O(b\log(b/\varepsilon))$. This is because each of the $nb-b(b+1)/2=nb+O(b^2)$ controlled-Z^a gates consumes 4 T gates to be first mapped into an uncontrolled Z^a and another 4 T gates for the Z^a to be implemented as a part of the adder circuit, except for controlled-$Z^{1/2}$ and controlled-$Z^{1/4}$ gates; the two require 7 T gates to implement and 1 T gate to correct for the phase in the partial gradient operation. The construction of the special state $|\psi_{d+1,b+1}\rangle$ requires implementation of $O(b)$ Z^a rotations, and we approximate each rotation with $O(\log(b/\varepsilon))$ T gates[33] to achieve accuracy ε/b per rotation.

There are two sources of approximation errors in our construction. Our circuit differs from the ideal AQFT circuit only in the preparation of the special state $|\psi_{d+1,b+1}\rangle$. Therefore, the spectral norm distance between our AQFT circuit and the ideal AQFT is $O(b\cdot\varepsilon/b)=O(\varepsilon)$. This ensures that, with $1-O(\varepsilon^2)$ probability, regardless of how many operations to follow from the $|\psi_{d+1,b+1}\rangle$ state preparation stage, our circuit implements the ideal AQFT. If we choose $b=O(\log(n/\varepsilon))$, the spectral norm error of the ideal AQFT circuit will be $O(\varepsilon)$. Due to the triangle inequality, the total error can be upper bounded by adding the error of the Clifford+T synthesis and the error of AQFT, which is still $O(\varepsilon)$.

The above error analysis shows that for all effective purposes (specifically, when $\varepsilon>n/2^n$) we can drop the dependence on the approximation error ε, resulting in the claimed T-count of $O(n\log n)$.

Future work

Future lines of inquiry may include laying out our circuit in restricted architectures and the optimization of depth. To address former, both the basic QFT[22,23] and the adder[31] we rely on (being the long adder) can be laid out in the Linear Nearest Neighbor architecture with a constant SWAP overhead. Thus, the increase in the CNOT gate count due to SWAP operations will remain under control, and the overall cost of the implementation is expected to continue being dominated by the cost of the T gates (note that the introduction of SWAP gates does not increase the number of T gates), although the cost of the CNOTS will start to matter more. To address depth, we first note that everything but the adder is already parallelized. To optimize depth, one may choose to rely on a fast logarithmic-depth adder and lay it out in 2D Square Lattice (a natural architecture for superconducting circuit quantum information processors) using the H-tree – an H-tree, popular in VLSI design, is a fractal tree, embedded in a 2D square lattice, constructed from a repeating pattern that resembles the letter H. This will introduce additional gates and require more space, but it may reduce the depth. Note that for small numbers such as those used in our result ($b=13$) the H-tree remains compact and requires few SWAP operations.

Conclusion

Before our contribution, the best known coherent approximation of the n-qubit QFT to an error ε by a quantum fault-tolerant Clifford+T circuit featured the T-count of $O(n\log(n/\varepsilon)\log(\frac{n\log(n/\varepsilon)}{\varepsilon}))$, with the term $O(n\log(n/\varepsilon))$ originating from the standard AQFT construction using controlled rotations, and term $O(\log(\frac{n}{\log(n/\varepsilon)}\varepsilon))$ coming from the fault-tolerance overhead. In this paper we reported an improved approximation of the QFT by a quantum Clifford+T circuit with the T-count of $O(n\log(n/\varepsilon)+\log(n/\varepsilon)\log(\frac{\log(n/\varepsilon)}{\varepsilon}))$. Our improvement is twofold: first, we reduce the dependence on n from $O(n\log^2(n))$ to $O(n\log(n))$, and second, we moved the dependence on ε from the leading term into a lower order additive term. This means that the smaller the desired approximation error the more efficient our construction is compared to those known previously.

Our implementation includes constant factor improvements that are not captured by the asymptotics. We report significant practical advantages from applying our construction, as is evidenced by the numbers in Table 1, showing the improvement by a factor of 10 to 12 in the T-count for values of n of the size expected in practical applications of quantum computers. This shows that our result carries both theoretical and practical value.

METHODS

Descriptions of the methods used to construct the AQFT circuit, the central result of our paper, are available in Section II. See Section II A for the detailed methods of the circuit construction. See Section II B for further circuit optimization methods used to improve the T-gate counts.

DATA AVAILABILITY

The AQFT circuits that use our improved circuit design are available in the online repository[38] https://github.com/y-nam/QFT.

CODE AVAILABILITY

Our improved AQFT circuits, which are the quantum programs, are available in the online repository[38] https://github.com/y-nam/QFT.

Received: 18 July 2019; Accepted: 13 February 2020;
Published online: 13 March 2020

REFERENCES

1. Shor, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**, 1484–1509 (1997).
2. Kitaev, A. Quantum measurements and the Abelian Stabilizer Problem. Preprint at https://arxiv.org/abs/quant-ph/9511026 (1995).
3. Brassard G. & Hoyer P. An exact quantum polynomial-time algorithm for Simon's Problem. In *Proc. of Fifth Israeli Symposium on Theory of Computing and Systems*, 12–23 (IEEE, Ramat-Gan, Israel, 1997). https://arxiv.org/quant-ph/9704027.
4. Grover, L. Quantum computers can search rapidly by using almost any transformation. *Phys. Rev. Lett.* **80**, 4329 (1998).
5. Kassal, I., Whitfield, J. D., Perdomo-Ortiz, A., Yung, M.-H. & Aspuru-Guzik, A. Simulating chemistry using quantum computers. *Annu. Rev. Phys. Chem.* **62**, 185 (2011).
6. Abrams, D. S. & Lloyd, S. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.* **83**, 5162 (1999).
7. Kassal, I. & Aspuru-Guzik, A. Quantum algorithm for molecular properties and geometry optimization. *J. Chem. Phys.* **131**, 224102 (2009).
8. Sheridan, L., Maslov, D. & Mosca, M. Approximating fractional time quantum evolution. *J. Phys. A* **42**, 185302 (2009).
9. Klappenecker, A. & Roetteler, M. Quantum software reusability. *Int. J. Found. Comput. Sci.* **14**, 777–796 (2003).
10. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
11. Draper T. G. Addition on a quantum computer. Preprint at https://arxiv.org/abs/quant-ph/0008033 (2000).
12. Ruiz-Perez, L. & Garcia-Escartin, J. C. Quantum arithmetic with the quantum Fourier transform. *Quantum Inf. Process.* **16**, 152 (2017).
13. Yang, Y.-G., Jia, X., Sun, S.-J. & Pan, Q.-X. Quantum cryptographic algorithm for color images using quantum Fourier transform and double random-phase encoding. *Inf. Sci.* **277**, 445–457 (2014).
14. Barker E. & Roginsky A. NIST Special Publication 800-131A Revision 1(NIST, Gaithersburg, MD, 2015).
15. Nam, Y. S. & Blümel, R. Scaling laws for Shor's algorithm with a banded quantum Fourier transform. *Phys. Rev. A* **87**, 032333 (2013).
16. Coppersmith D. An approximate Fourier transform useful in quantum factoring. Preprint at https://arxiv.org/abs/quant-ph/0201067 (2002).
17. Barenco, A., Ekert, A., Suominen, K.-A. & Törmä, P. Approximate quantum Fourier transform and decoherence. *Phys. Rev. A* **54**, 139 (1996).
18. Niwa, J., Matsumoto, K. & Imai, H. General-purpose parallel simulator for quantum computing. *Phys. Rev. A* **66**, 062317 (2002).
19. Fowler, A. & Hollenberg, L. C. L. Scalability of Shor's algorithm with a limited set of rotation gates. *Phys. Rev. A* **70**, 032329 (2004).
20. Nam, Y. S. & Blümel, R. Robustness and performance scaling of a quantum computer with respect to a class of static defects. *Phys. Rev. A* **88**, 062310 (2013).
21. Nam, Y. S. & Blümel, R. Analytical formulas for the performance scaling of quantum processors with a large number of defective gates. *Phys. Rev. A* **92**, 042301 (2015).
22. Takahashi, Y., Kunihiro, N. & Ohta, K. The quantum Fourier transform on a linear nearest neighbor architecture. *Quantum Inf. Comput.* **7**, 383–391 (2007).
23. Maslov, D. Linear depth stabilizer and quantum Fourier transformation circuits with no auxiliary qubits in finite neighbor quantum architectures. *Phys. Rev. A* **76**, 052310 (2007).
24. Maslov, D. & Nam, Y. S. Use of global interactions in efficient quantum circuit constructions. *New J. Phys.* **20**, 033018 (2018).
25. Bravyi, S. & Kitaev, A. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phy. Rev. A* **71**, 022316 (2005).
26. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information*. (Cambridge University Press, New York, 2000).
27. Griffiths, R. B. & Niu, C. S. Semiclassical Fourier transform for quantum computation. *Phys. Rev. Lett.* **76**, 3228 (1996).
28. Goto, H. Resource requirements for a fault-tolerant quantum Fourier transform. *Phys. Rev. A* **90**, 052318 (2014).
29. Jones, C. Novel constructions for the fault-tolerant Toffoli gate. *Phys. Rev. A* **87**, 022328 (2013).
30. Amy, M., Maslov, D., Mosca, M. & Roetteler, M. A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. *IEEE Trans. CAD* **32**, 818–830 (2013).
31. Gidney, C. Halving the cost of quantum addition. *Quantum* **2**, 74 (2018).
32. Yu., A., Kitaev, Shen, A. H. & Vyalyi, M. N. *Classical and Quantum Computation*. (American Mathematical Society, Providence, RI, 2002).
33. Bocharov, A., Roetteler, M. & Svore, K. M. Efficient synthesis of universal Repeat-Until-Success circuits. *Phys. Rev. Lett.* **114**, 080502 (2015).
34. Bertok P. *PSLQ Integer Relation Algorithm Implementation*. http://library.wolfram.com/infocenter/MathSource/4263/ (2004).
35. Ross, N. J. & Selinger, P. Optimal ancilla-free Clifford+T approximation of z-rotations. *Quantum Inf. Comput.* **16**, 901–953 (2016).
36. Kliuchnikov, V., Maslov, D. & Mosca, M. Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates. *Quantum Inf. Comput.* **13**, 607–630 (2013).
37. Nam, Y. S., Ross, N. J., Su, Y., Childs, A. M. & Maslov, D. Automated optimization of large quantum circuits with continuous parameters. *npj Quantum Inf.* **4**, 23 (2018).
38. Nam Y. S., Su Y. & Maslov Maslov D. https://github.com/y-nam/QFT (2018).

ACKNOWLEDGEMENTS

Y.S. thanks Andrew Glaudell for helpful discussions. Authors thank Craig Gidney for his helpful comments that inspired us to further improve our results. We note that Craig Gidney independently developed $6n + Const$ improvement through the use of reduced gradient operation ($8n + Const$ in our version). Y.S. was supported in part by the Army Research Office (MURI award W911NF-16-1-0349), the National Science Foundation (grant 1526380), and the Google Ph.D. Fellowship program. This material was based on work supported by the National Science Foundation, while D.M. working at the Foundation. Any opinion, finding, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

AUTHOR CONTRIBUTIONS

D.M. designed research. Y.N. and D.M. designed the quantum circuits. Y.S. and Y.N. implemented R.U.S. and Y.N. synthesized the QFT circuits.

COMPETING INTERESTS

The authors declare that there are no competing interests.

ADDITIONAL INFORMATION

Correspondence and requests for materials should be addressed to Y.N., Y.S. or D.M.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.