www.nature.com/scientificreports

scientific reports

Check for updates

OPEN

Demonstration of Shor's factoring algorithm for N = 21 on IBM quantum processors

Unathi Skosana & Mark Tame

We report a proof-of-concept demonstration of a quantum order-finding algorithm for factoring the integer 21. Our demonstration involves the use of a compiled version of the quantum phase estimation routine, and builds upon a previous demonstration. We go beyond this work by using a configuration of approximate Toffoli gates with residual phase shifts, which preserves the functional correctness and allows us to achieve a complete factoring of $N = 21$. We implemented the algorithm on IBM quantum processors using only five qubits and successfully verified the presence of entanglement between the control and work register qubits, which is a necessary condition for the algorithm's speedup in general. The techniques we employ may be useful in carrying out Shor's algorithm for larger integers, or other algorithms in systems with a limited number of noisy qubits.

Shor's algorithm[1] is a quantum algorithm that provides a way of finding the nontrivial factors of an L-bit odd composite integer $N = pq$ in polynomial time with high probability. The crux of Shor's algorithm rests upon Quantum Phase Estimation (QPE)[2], which is a quantum routine that estimates the phase φ_u of an eigenvalue $e^{2\pi i\varphi_u}$ corresponding to an eigenvector $|u\rangle$ for some unitary matrix $\hat{U}$. QPE efficiently solves a problem related to factoring, known as the order-finding problem, in polynomial time in the number of bits needed to specify the problem, which in this case is $L = \lceil \log_2 N \rceil$. By solving the order-finding problem using QPE and carrying out a few extra steps, one can factor the integer N. There is no known classical algorithm that can solve the same problem in polynomial time[2,3].

A large corpus of work has been done with regards to the experimental realization of Shor's algorithm over the years. The pioneering work was performed with liquid-state nuclear magnetic resonance, factoring 15 on a 7-qubit quantum computer[4]. The considerable resource demands of Shor's original algorithm were circumvented by using various approaches, including adiabatic quantum computing[5] and in the standard network model using techniques of compilation[6] that reduced the demands to within the reach of single-photon architectures[7–9] and a super-conducting phase qubit system[10]. In 2012, a proof-of-concept demonstration of the order-finding algorithm for the integer 21 was carried out with photonic qubits using, in addition to the aforementioned compilation technique, an iterative scheme[11], where the control register is reduced to one qubit and this qubit is reset and reused[12,13]. However, factoring was not possible in this demonstration due to the low number of iterations. Later, the iterative scheme was demonstrated for factoring 15, 21 and 35 on an IBM quantum processor by splitting up the iterations and combining the outcomes[14]. Recently, building on previous schemes of hybrid factorization[15,16], a quantum-classical hybrid scheme has been implemented on IBM's quantum processors for the prime factorization of 35. This hybrid scheme of factorization alleviates the resource requirements of the algorithm at the expense of performing part of the factoring classically[17].

In this paper, we build on the order-finding routine of Ref.[11] and implement a version of Shor's algorithm for factoring 21 using only 5 qubits—the work register contains 2 qubits and the control register contains 3 qubits, each providing 1-bit of accuracy in the resolution of the peaks in the output probability distribution used to find the order. This approach is in contrast to the iterative version[18] used in Refs.[11] and[14], which employs a single qubit that is recycled through measurement and feed-forward, giving 1-bit of accuracy each time it is recycled. The advantage of the iterative approach lies in this very reason; through mid-circuit measurement and real-time conditional feed-forward operations, the total number of qubits required by the algorithm is significantly reduced. At the time of writing, IBM's quantum processors do not yet support real-time conditionals necessary for the implementation of the iterative approach, so we use 3 qubits for the control register, one for each effective iteration. Thus, our compact approach is completely equivalent to the iterative approach. In future, once the capability of performing real-time conditionals is added, a further reduction in resources will be possible for

Department of Physics, Stellenbosch University, Matieland 7602, South Africa. email: ukskosana@gmail.com

our implementation, potentially improving the quality of the results even more and opening up the possibility of factoring larger integers.

As it stands, the controlled-NOT (CX) gate count of the standard algorithm[19] exceeds 40 and in preliminary tests we have found that the output probability distribution is indistinguishable from a uniform probability distribution (noise) on the IBM quantum processors. Our improved version reduces the CX gate count through the use of relative phase Toffoli gates, reducing the CX gate count by half while leaving the overall operation of the circuit unchanged and we suspect this technique may extend beyond the case considered here. We have gone further than the work in Ref.[11], where full factorization of 21 was not achieved as with only two bits of accuracy for the peaks of the output probability distribution continued fractions would fail to extract the correct order. On the other hand, in the work in Ref.[14], where 21 was factored on an IBM processor, a larger number of 6 qubits was required and the iterations were split into three separate circuits, with the need to re-initialise the work register into specific quantum states for each iteration. Our approach is thus more efficient and compact, enabling algorithm outcomes with reduced noise. To support our claims, we successfully carry out continued fractions and evaluate the performance of the algorithm by (i) quantitatively comparing the measured probability distribution with the ideal distribution and noise via the Kolmogorov distance, (ii) performing state tomography experiments on the control register, and (iii) verifying the presence of entanglement across both registers.

The paper is organized as follows. In “Background”, we give a brief review of the order-finding problem and its relation to Shor’s algorithm. In “Compiled Shor’s algorithm” we expound on the compiled version of Shor’s algorithm, where we consider the specific case of the factorization of $N = 21$. We construct the quantum circuits that realize the required modular exponentiation unitaries and proceed to optimize their CX gate count through the introduction of relative phase Toffoli gates. We report our results from executing our compact construction of the algorithm on IBM’s quantum computers in “Experiments”. Finally, we provide concluding remarks of our study in “Concluding remarks”. Supplementary information is included.

Background

Order finding.

The order-finding problem is typically stated as follows. Given positive integers N and $a \in \{0, 1, \ldots, N-1\}$ that share no common factors, we seek to find the least positive integer $r \in \{0, 1, \ldots N\}$ such that $a^r \bmod N = 1$. The integer r is said to be the *order* of a and N, and the order-finding problem is that of finding r for a particular a and N. There exists no classical algorithm that can solve the order-finding problem efficiently, that is, with operations (elementary gates) that scale polynomially in the number of bits needed to specify N, *i.e.* $L \equiv \lceil \log_2 N \rceil$[2,3].

Shor’s algorithm.

The order-finding problem can be efficiently solved on a quantum computer with $\mathcal{O}(L^3)$ operations; the cost being mostly due to the *modular exponentiation* operation which requires $\mathcal{O}(L^3)$ quantum gates[2]. The problem of prime factorization is the subject of Shor’s algorithm, which is equivalent to the order-finding problem: for an L-bit positive odd integer $N = pq$ and randomly chosen positive integer $a \leq N$ co-prime to N, the order r of a and N can be used to find the non-trivial factors of N. The algorithm is probabilistically guaranteed, with probability greater than a half that the greatest common divisor $\gcd(a^{r/2} \pm 1, N)$ gives the prime factors of N[2]. Shor’s algorithm uses two quantum registers; a control register and a work register. The control register contains n qubits, each for one bit of precision in the algorithmic output. The work register contains $m = \lceil \log_2 N \rceil$ qubits where m is the number of qubits to encode N. The measurement of the control register outputs a probability distribution peaked at approximately the values of $2^n s/r$, where s is associated with the outcome of the measurement and thus randomly assigned. The details of how the peaked probability distribution comes about are given in the order-finding routine outlined below. One can determine the order r from the peak values of the distribution using *continued fractions*, with a number of operations that scales polynomially in $\lceil \log_2 N \rceil$. The procedure, or routine, for order finding is summarized below.

Order-finding routine.

1. Initialization

   Prepare $|0\rangle^{\otimes n}|0\rangle^{\otimes m}$ and apply $H^{\otimes n}$ on the control register and X on the m^{th} qubit in the work register to create a superposition of 2^n states in the control register and $|1\rangle$ in the work register:

$$|0\rangle^{\otimes n}|0\rangle^{\otimes m} \rightarrow \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle|1\rangle.$$

2. Modular exponentiation function (MEF)

   Conditionally apply the unitary operation $\hat{U}$ that implements the modular exponentiation function $a^x \bmod N$ on the work register whenever the control register is in state $|x\rangle$:

$$\begin{aligned}\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle|1\rangle &\rightarrow \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle|a^x \bmod N\rangle \\ &= \frac{1}{\sqrt{r2^n}} \sum_{s=0}^{r-1} \sum_{x=0}^{2^n-1} e^{2\pi i s x/r} |x\rangle|u_s\rangle.\end{aligned}$$

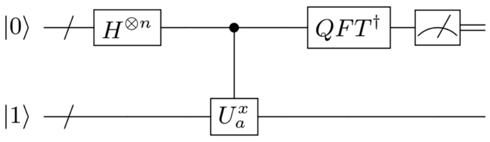


Figure 1. Schematic of the routine used for the period finding part of Shor's algorithm. The first (control) register has n qubits. The number of qubits in the control register determines the bit-accuracy of the value of $2^n s/r$. The bottom (work) register has the m qubits required to encode N. First, the control and work registers are initialized, then conditional modular exponentiation is performed, indicated by the controlled unitary and an inverse quantum Fourier transform is applied to the control register followed by a standard computational basis measurement. The circuit is essentially the QPE algorithm applied to the unitary matrix $\hat{U}_a$—see text for details.

In the second line, $|u_s\rangle$ is the eigenstate of $\hat{U}$: $\hat{U}|u_s\rangle = e^{2\pi is/r}|u_s\rangle$ and $\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle$ has been used for the work register. The MEF operation is equivalent to applying $\hat{U}^x$ to the work register when the state $|x\rangle$ is in the control register, as shown in Fig. 1, with $\hat{U}|y\rangle = |ay \bmod N\rangle$ for a given state $|y\rangle$ (the subscript a in $\hat{U}$ is suppressed for notational convenience). This provides an alternative way to write the output state and allows a connection between the MEF operation and the QPE algorithm for the unitary operation $\hat{U}$.

3. Inverse Quantum Fourier Transform (QFT)

   Apply the inverse quantum Fourier transform on the control register:

$$\frac{1}{\sqrt{r2^n}}\sum_{s=0}^{r-1}\sum_{x=0}^{2^n-1} e^{2\pi isx/r}|x\rangle|u_s\rangle \rightarrow \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|\varphi_s\rangle|u_s\rangle.$$

4. Measurements

   Measure the control register in the computational basis, yielding peaks in the probability for states where $\varphi_s \simeq 2^n s/r$ due to the inverse QFT. Thus, the outcome of the algorithm is probabilistic, however, there is a high probability of obtaining the location of the φ_s peaks after only a few runs. The accuracy of φ_s to $2^n s/r$ is determined by the number of qubits in the control register.

5. Continued fractions

   Apply continued fractions to $\varphi = \varphi_s/2^n$ (the approximation of s/r) to extract out r from the convergents (see Supplementary information VII for details).

Compiled Shor's algorithm

A full-scale implementation of Shor's algorithm to factor an L-bit number would require a quantum circuit with $72L^3$ quantum gates acting on $5L + 1$ qubits for the order-finding routine[20], i.e. factoring $N = 21$ would require 9000 elementary quantum gates acting on 26 qubits. The overhead in quantum gates comes from the modular exponentiation function part of the algorithm, while the overhead in qubits comes from the level of accuracy needed to successfully carry out the continued fractions part of the algorithm. Such an overhead obviously puts a full-scale implementation beyond the reach of current devices. However, compilation techniques such as the one described in Ref.[20], bridge this gap and allow for small-scale proof-of-concept demonstrations, where the quantum circuit is tailored around properties of the number to be factored. This significantly simplifies the controlled-operations that realize the MEF operation (see previous section), which is the most resource-intensive part of the order-finding routine. The resource demands of the compiled quantum circuit are significantly reduced, making it suitable for quantum devices with low connectivity.

From Ref.[11], we extend the compiled quantum order-finding routine for the particular case of factoring $N = 21$ with $a = 4$ to accommodate another iteration for better precision in the resolution of the peaks for the value of $2^n s/r$. For the case of $N = 21$, other choices of a give 2, 4 or 6 for r. The cases for $r = 2$ or $r = 4$ have been demonstrated for $N = 15$[4,7–10] and would bear a similar circuit structure in the present case. With only three iterations, $r = 6$ would be out of reach as continued fractions would fail. For $a = 4$ we have $r = 3$, which is a choice that does not suffer from the aforementioned reasons. Despite r being an odd integer, the algorithm is successful in finding it from $a = 4$. This is the case for certain choices of perfect square a and odd r, and $a = 4$ and $r = 3$ is such a case[11].

In contrast to Ref.[11], our implementation is not iterative and uses three qubits for the control register rather than one qubit recycled on every iteration. The iterative version is based on the recursive phase estimation, made possible by the use of the semi-classical QFT[12]. However, we have used the traditional QFT because mid-circuit measurements with real-time conditionals are not possible yet on IBM's quantum processors. The traditional QFT for 3 qubits (see[2]—Box 5.1) that we implemented is equivalent to Fig. 1A and Fig. 1B in Ref.[13]. The latter is the semi-classical QFT that makes possible the implementation of the iterative version of Shor. If mid-circuit measurements with real-time conditionals were possible, the 3-qubit semi-classical QFT would be possible and may improve the quality of the results we present here through the use of only 1 qubit for the control register, as in Ref.[11]. IBM has suggested that the behaviour of real-time conditionals can be reproduced through post selection of the mid-circuit measurements. However, in the present case the speed up gained would be lost using this post selection method (see Supplementary information I).

In Ref.[11], a step that is unique among the compilation steps of previous demonstrations, and central to their demonstration is mapping the three levels $|1\rangle, |4\rangle$ and $|16\rangle$ accessed by the possible $2^L = 2^5$ levels of the work-register to only a single qutrit system. In our demonstration we also use this step, however IBM processors consist of qubits and so we represent the work register by three basis states from a two-qubit system and discard the fourth basis state as a null state. The states encoding the three possible levels of the work register; $|1\rangle, |4\rangle$ and $|16\rangle$ are mapped to $|q_0 q_1\rangle$ according to

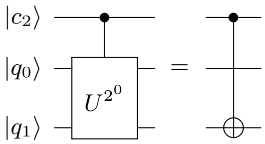
$$\begin{aligned} |1\rangle &\mapsto |\log_4 1\rangle = |00\rangle, \\ |4\rangle &\mapsto |\log_4 4\rangle = |01\rangle, \\ |16\rangle &\mapsto |\log_4 16\rangle = |10\rangle. \end{aligned} \tag{1}$$

Therefore instead of evaluating $4^x \bmod 21$ in the work register as described in step 2 of "Shor's algorithm", the compiled version of Shor's algorithm effectively evaluates $\log_4[4^x \bmod 21]$ in its place for $x = 0, 1 \dots 2^3 - 1$[20], which reduces the size of the work register to 2 qubits in comparison to the 5 qubits required in the standard construction. Note the ordering of quantum bits in the work register is $|q\rangle = |q_0\rangle|q_1\rangle$, where the rightmost qubit is associated with the least significant bit. Similarly, with the control register we have $|c\rangle = |c_0\rangle|c_1\rangle|c_2\rangle$. In total the algorithm requires 5 qubits: 3 for the control register and 2 for the work register. Implementing the controlled unitaries $\hat{U}^x$ that perform the modular exponentiation $|x\rangle|y\rangle \to |x\rangle\hat{U}^x|y\rangle = |x\rangle|a^x y \bmod N\rangle$ reduces to effectively swapping around the states $|1\rangle, |4\rangle$ and $|16\rangle$ in the work register controlled by the corresponding bit of the integer x in the control register, which is given by $x = c_2 2^0 + c_1 2^1 + c_0 2^2$. In other words, $\hat{U}^x = \hat{U}^{c_0 2^2}\hat{U}^{c_1 2^1}\hat{U}^{c_2 2^0}$. Thus, depending on the control qubit c_i, one of the following maps is applied:

$$\begin{aligned} \hat{U}^1 &: \{|1\rangle \mapsto |4\rangle, |4\rangle \mapsto |16\rangle, |16\rangle \mapsto |1\rangle\}, \\ \hat{U}^2 &: \{|1\rangle \mapsto |16\rangle, |4\rangle \mapsto |1\rangle, |16\rangle \mapsto |4\rangle\}, \\ \hat{U}^4 &: \{|1\rangle \mapsto |4\rangle, |4\rangle \mapsto |16\rangle, |16\rangle \mapsto |1\rangle\}. \end{aligned} \tag{2}$$

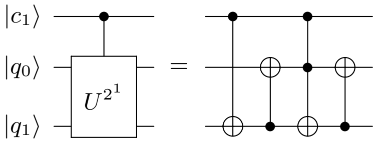
The next simplification step comes from the fact that these operations on the work register need not be controlled *SWAP* (Fredkin) gates, they can be as simple as *CX* gates, as we show next.

Modular exponentiation. Implementing $\hat{U}^1$ on the two-qubit work register is simplified considerably by noting that the states $|4\rangle$ and $|16\rangle$ initially have zero amplitude, and thus the operation $|1\rangle \mapsto |4\rangle$ alone is sufficient. This operation can realized with a *CX* gate controlled by $|c_2\rangle$ targeting the second work qubit $|q_1\rangle$.



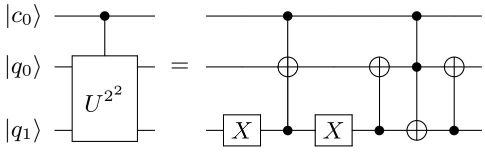
(3)

Similarly, the implementation of $\hat{U}^2$ can be simplified by noting that the states $|1\rangle$ and $|4\rangle$ are the only non-zero amplitude states in the work register after $\hat{U}^1$ may have been applied, thus prompting us to only consider $|1\rangle \mapsto |4\rangle$ and $|4\rangle \mapsto |16\rangle$. A *CX* gate controlled by $|c_1\rangle$ targeting $|q_1\rangle$ followed by a Fredkin gate, swapping $|q_0\rangle$ and $|q_1\rangle$ realizes this simplified $\hat{U}^2$.



(4)

In the above, the Fredkin gate has been decomposed into a Toffoli gate (*CCX*) and two *CX* gates. The subsequent implementation of $\hat{U}^4$ admits no simplifications as all the possible states in the work register may have non-zero amplitude at this point. This operation is implemented with a Toffoli and a Fredkin gate with single-qubit *X* gates.



(5)

The full circuit diagram is shown in Fig. 2—note that before simplification the order of application of the controlled unitaries is interchangeable, $\hat{U}^{2^{(n-1)}}$ or $\hat{U}^{2^1}$ could be applied first. Interchanging the order only has the

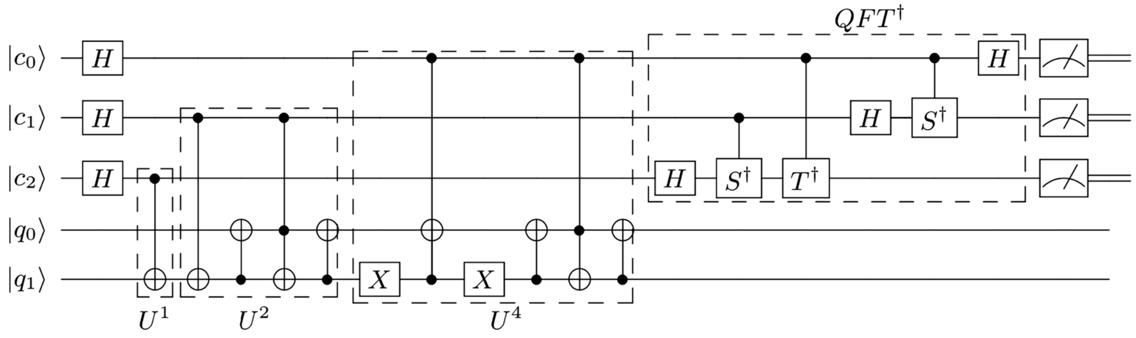
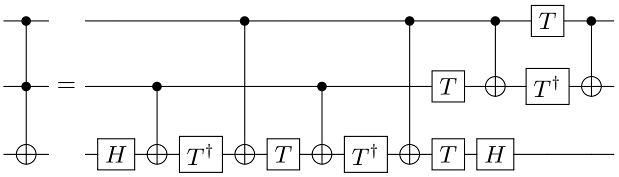


Figure 2. Compiled quantum order-finding routine for $N = 21$ and $a = 4$. This circuit uses five qubits in total; 3 for the control register and 2 for the work register. The above circuit determines $2^n s/r$ to three bits of accuracy, from which the order can be extracted. Here, up to a global phase, $S = R_z(\frac{\pi}{2})$ and $T = R_z(\frac{\pi}{4})$ are phase and $\pi/8$ gates, respectively.

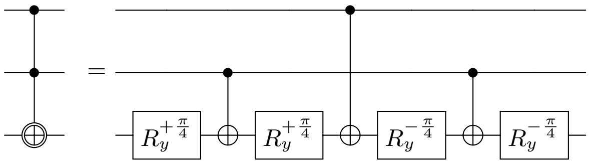
effect of interchanging the order of the outcome bits at the end of the computation. This is the reason the order of application of the controlled unitaries here is in reverse order to that in Ref.[11].

Modular exponentiation with relative phase Toffolis. In total, the modular exponentiation routine requires three Toffoli gates; traditionally a single Toffoli gate can be decomposed into six CX gates and several single-qubit gates[2] as follows



(6)

Taking into account a given processor's topology and the constraints it poses, as well as other parts of the circuit (the inverse QFT), further increases the tally of CX gates. This becomes undesirable as it is understood that there is an upper limit on the number of CX gates that can be in a circuit with the guarantee of a successful computation. The number of CX gates from the decomposition of the Toffoli gate can be cut in half if we permit the operation to be correct up to relative phase shifts. Margolus constructed a gate that implements the Toffoli gate up to a relative phase shift of $|101\rangle \mapsto -|101\rangle$ that only uses three CX gates and four single qubit gates[21]. This construction has been shown to be optimal[22].



(7)

Maslov showed the advantages of using a relative phase Toffoli gate when the gate is applied last or when relative phases do not matter for certain configurations of Toffolis, resulting in no overall change to the functionality in any significant way[23]. The configuration in the circuit shown in Fig. 2 is one such configuration that permits a replacement of Toffoli gates with Margolus gates without changing the overall functionality. All the Margolus gates in the circuit in Fig. 3 (which is the circuit in Fig. 2 with the Toffoli gates replaced by Margolus gates) never encounter the basis state $|101\rangle$, thus leaving the operation of the circuit unchanged. See Supplementary information II for details. This further compacting reduces the number of CX gates considerably and puts the algorithm within reach of current IBM processors with a limited number of noisy qubits.

Experiments

Physical qubit mapping. The proposed compiled circuit in Fig. 3 was mapped onto 5 physical qubits (3 control qubits and 2 work qubits) and executed on a sub-processor of IBM's 7-qubit quantum processor **ibmq_casablanca** and 27-qubit quantum processor **ibmq_toronto**, which we will refer to as 7Q and 27Q, and whose topologies are shown in Fig. 4. When mapping the compiled circuit a few considerations can be taken into account. First, as can be seen from Eq. (7), the Margolus gate can be implemented on a collinear set of qubits, as the first control qubit need not be connected to the second control qubit. On the other hand, mapping the three-qubit inverse QFT onto physical qubits without incurring additional $SWAP$ gates is not possible, as the three controlled-phase gates require all three qubits to be interconnected in a triangle and the aforementioned quantum processors do not have such a topology. Additionally, more $SWAP$ gates are introduced to the transpiled circuit, as the processor topologies do not permit the topology required by the compiled circuit, as shown in Fig. 5.

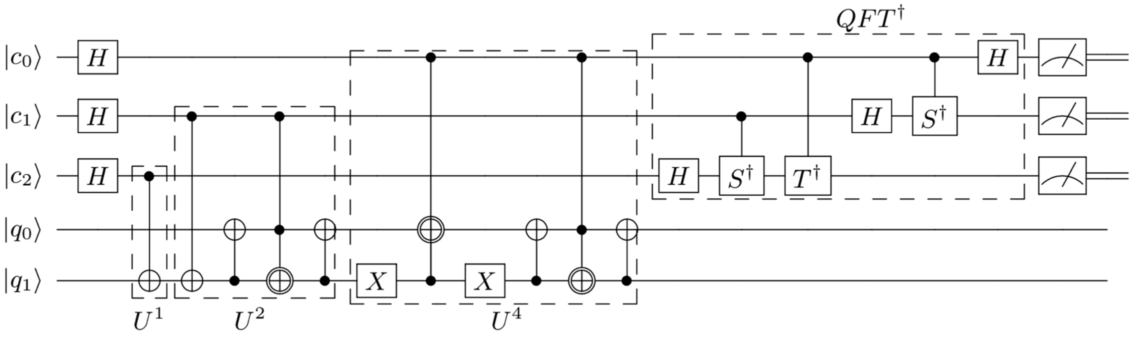
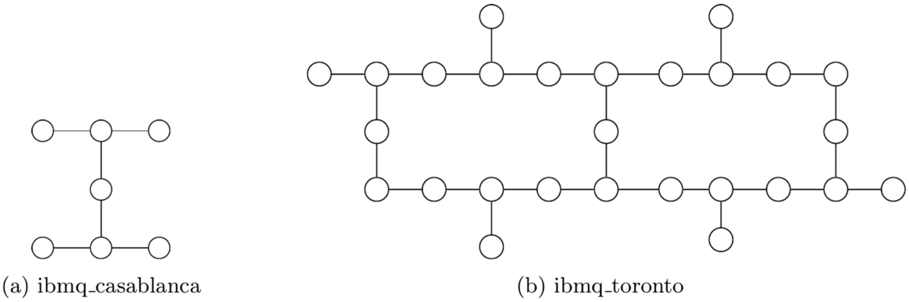


Figure 3. Approximate compiled quantum order-finding routine implemented with Margolus gates in place of Toffoli gates in the construction in Fig. 2.


(a) ibmq_casablanca

(b) ibmq_toronto

Figure 4. Qubit topology of IBM Q experience processors.

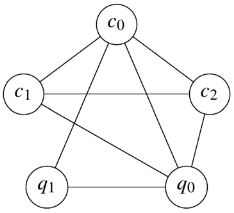
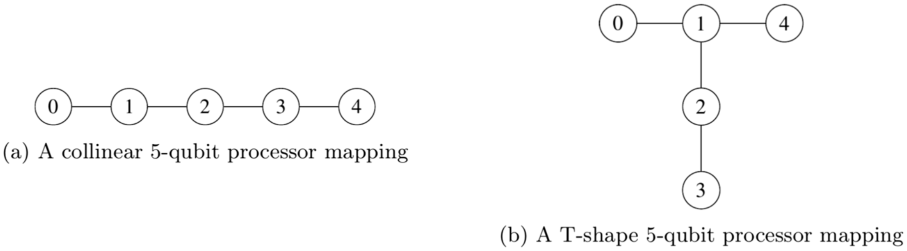


Figure 5. Qubit connections required by the compiled circuit in Fig. 3.



(a) A collinear 5-qubit processor mapping

(b) A T-shape 5-qubit processor mapping

Figure 6. The two possible 5-qubit processor mappings on the architectures shown in Fig. 4.

The only possible five-qubit mappings on the quantum processors are all isomorphic to either a collinear set of qubits or a T-shaped set of qubits, as shown in Fig. 6a,b. Choosing the mapping in Fig. 6b over the one in Fig. 6a is motivated by the fact that the former is slightly more connected than latter and thus in effect would reduce the number of *SWAP* gates in the mapped and transpiled circuit.

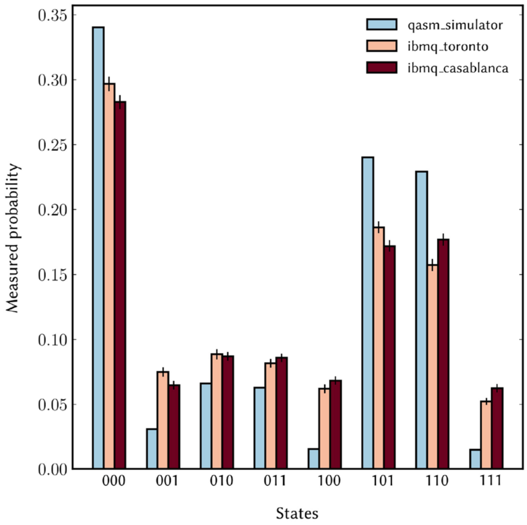


Figure 7. Results of the complete quantum order-finding routine for $N = 21$ and $a = 4$. On each processor, the circuit was executed 8192 × 100 times with measurement error mitigation. The error bars represent 95% confidence intervals around the mean value of each histogram bin (see Supplementary information IV). The simulator probabilities show the ideal case.

Performance. To evaluate the performance of the algorithm, we first transpiled the circuit in Fig. 3 down to the chosen quantum processor with the mapping below

$$\begin{aligned} 0 &\mapsto c_0, \\ 1 &\mapsto c_2, \\ 4 &\mapsto c_1, \\ 2 &\mapsto q_1, \\ 3 &\mapsto q_0. \end{aligned} \tag{8}$$

Through the transpiler's optimization, with the mapping above it is possible to have a circuit that has 25 CX gates and a circuit depth of 35. Figure 7 shows the results of measurements on the control register qubits from the two processors, where measurement error mitigation has been applied to results and mitigates the effect of measurement errors on the raw results (see Supplementary information III). The outcomes $|101\rangle$ and $|110\rangle$ occur with probability $\sim$ 14% and $\sim$ 16% respectively. The theoretical ideal probability is $\sim$ 25%, as can be seen from the simulator results in Fig. 7. However, the amplification of the peaks $|000\rangle$, $|101\rangle$ and $|110\rangle$ is clearly visible from the processor outcomes.

We quantify the successful performance of the algorithm by comparing the experimental and ideal probability distributions via the trace distance or Kolmogorov distance[2], which measures the closeness of two discrete probability distributions P and Q and is defined by the equation $D(P, Q) \equiv \sum_{x\in\mathcal{X}} |P(x) - Q(x)|/2$, where $\mathcal{X}$ represents all possible outcomes. This measure shows an agreement between measured and ideal results—the trace distance between the measured distribution and the ideal distribution is 0.1694 and 0.1784 for **ibmq_toronto** and **ibmq_casablanca**, respectively. On the other hand, the trace distance between the ideal distribution and a candidate random uniform distribution is 0.4347. Furthermore, we evaluate the performance of the algorithm by characterizing the measured output state in the control register, this is achieved via state tomography yielding the density matrix of the measured state. The measured state and ideal state on the output register are quantitatively compared using the fidelity for two quantum states ρ and σ, and is defined to be $F(\rho, \sigma) \equiv \mathrm{tr}\sqrt{\rho^{1/2}\sigma\rho^{1/2}}^2$. We measured a fidelity of $F(\rho_{\mathrm{id}}, \rho_{27Q}) = 0.6948 \pm 00650$ and $F(\rho_{\mathrm{id}}, \rho_{7Q}) = 0.70 \pm 0.0275$ on the 27 qubit and 7 qubit quantum processors respectively, as shown in Fig. 8. In Fig. 9 we show the estimated density matrices in the computational basis for each respective device.

Factoring *N* = 21. The measured probability distributions in Fig. 7 are peaked in probability for the outcomes 000 ($\varphi_s = 0$), 101 ($\varphi_s = 5$) and 110 ($\varphi_s = 6$), with ideal probabilities of 0.35, 0.25 and 0.25, respectively. Here we are using the integer representation of the binary outcomes. The outcome 000 corresponds to a failure of the algorithm[11]. For the outcome 101, computing the continued fraction expansion of $\varphi = \varphi_s/2^n = 5/2^3 = 5/8$ gives the convergents $\{0, 1, 1/2, 2/3, 5/8\}$ (see Supplementary information VII for details), so that the third convergent 2/3 in the expansion can be identified as s/r and correctly gives $r = 3$ as the order when tested with the relation $a^r \bmod N = 1$, while the other convergents do not give an r that passes the test. On the other hand, the continued fraction expansion of $\varphi = 6/8$ gives $\{0, 1, 3/4\}$ and incorrectly gives $r = 4$ as the order (see Supplementary information VII for details). This failure can be avoided in principle by adding further qubits to the control register so that the peak in the probability distribution becomes narrower and more well

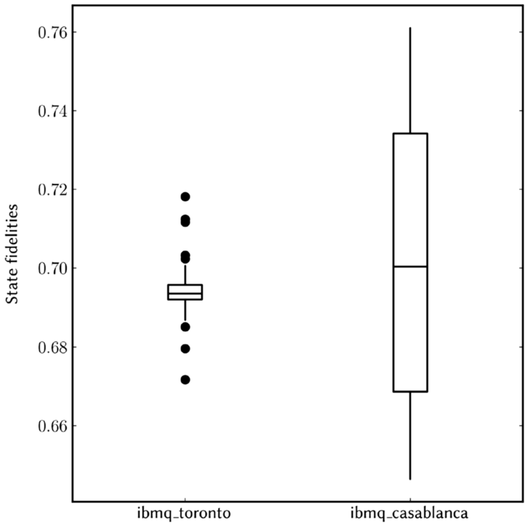


Figure 8. Boxplot of a sample ($\nu = 50$) of state fidelities from the respective two devices showing the spread of the values around the sample mean and 95% confidence intervals.

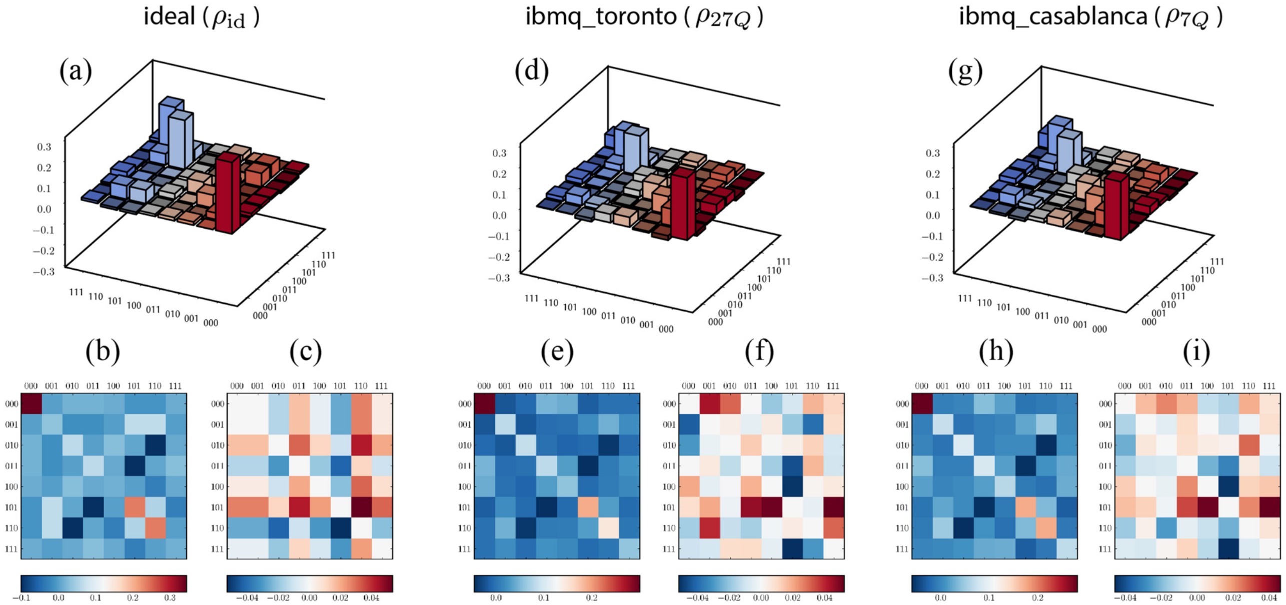


Figure 9. Ideal and measured density matrices after the inverse QFT, estimated via a maximum-likelihood reconstruction from measurement results in the Pauli-basis. **(a)** The ideal state $|\Psi\rangle\langle\Psi|$ (only the real parts are shown, imaginary parts are less than 0.04). **(b)** A matrix plot of the real part of $|\Psi\rangle\langle\Psi|$. **(c)** A matrix plot of the imaginary part of $|\Psi\rangle\langle\Psi|$. These plots are compared with the measured states ρ_{27Q} and ρ_{7Q} in panels **(d,g)**, and the corresponding matrix plot of their real parts in panels **(e,h)**, and imaginary parts in panels **(f,i)**, respectively. We observe there is a resemblance between the ideal state and the measured states, but noise in both real and imaginary parts is notable.

defined[11]. Another option is to simply apply continued fractions to all peaked outcomes and test if the value of r found satisfies the order relation for a and N. It is interesting to note that from the results of Ref.[11], successfully finding the order $r = 3$ was not possible to achieve, as with only two bits of accuracy in the experiment the continued fractions would always fail due to the peaked outcomes of 10 (2) and 11 (3) giving the convergents of $\{0, 1/2\}$ and $\{0, 1, 3/4\}$, respectively. In our case, we successfully find $r = 3$, from which we obtain $\mathrm{gcd}(\mathrm{a}^{\mathrm{r}/2} \pm 1, \mathrm{N}) = \mathrm{gcd}(8 \pm 1, 21) = 3$ and 7. Thus, with our demonstration, extending the number of outcome bits to three has allowed us to fully perform the quantum factoring of $N = 21$.

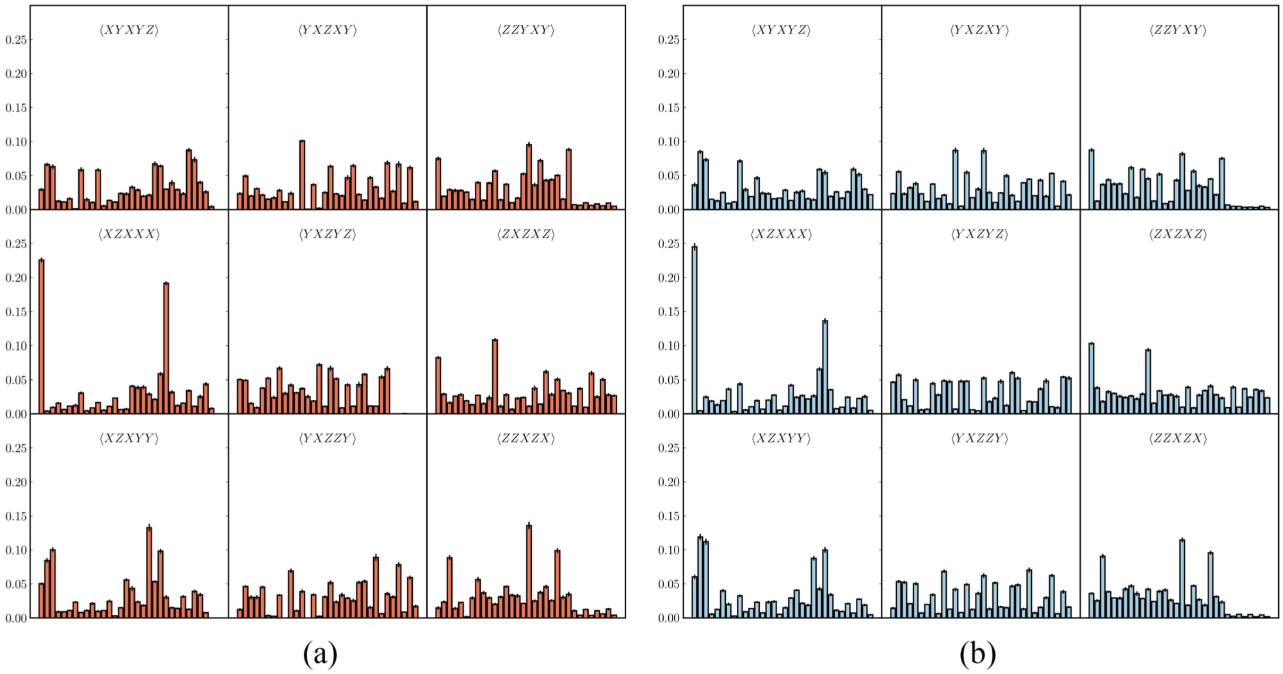


Figure 10. A subset of 9 of the 79 measurement settings required for each term in: **(a)** $\mathrm{tr}(|\Psi\rangle\langle\Psi|\rho_{7Q})$ and **(b)** $\mathrm{tr}(|\Psi\rangle\langle\Psi|\rho_{27Q})$. The x-axis from left to right shows the labels from p_{00000} to p_{11111}.

Verification of entanglement. The presence of entanglement between the control and work registers is known to be a requirement for the algorithm to gain any advantageous speedup over its classical counterparts in general[6,24,25]. For detecting genuine multipartite entanglement around the vicinity of an ideal state $|\psi\rangle$, one can construct a projector-based witness such as the one below:

$$\hat{\mathcal{W}}_{\psi} = \alpha\mathbb{I} - |\psi\rangle\langle\psi|, \tag{9}$$

where α is the square of the maximum overlap between $|\psi\rangle$ and all biseparable states. In other words, $\mathrm{tr}(\hat{\mathcal{W}}_{\psi}\rho) \geq 0$ for biseparable states and $\mathrm{tr}(\hat{\mathcal{W}}_{\psi}\rho) < 0$ for states with genuine multipartite entanglement in the vicinity of $|\psi\rangle$ [26]. For the ideal state after modular exponentiation (but before the inverse QFT) in both the control and work registers, $\alpha = 0.75$ was found using the method described in the appendix of Ref.[26]. This was implemented using the software package QUBIT4MATLAB[27]. Therefore ideally the state in both registers after modular exponentiation has genuine multipartite entanglement.

In order to check whether the output state from the IBM processors is close to the ideal state and has genuine multipartite entanglement, full state tomography would normally be needed to characterize the state ρ_{exp} in both the control and work registers. This would require 3^5 measurements, making it impractical to gather a sufficiently large data set within a meaningful time frame. However, we need not measure the full density matrix, the quantity $\mathrm{tr}(|\Psi\rangle\langle\Psi|\rho_{\mathrm{exp}})$ suffices. To measure this, we can decompose $\rho = |\Psi\rangle\langle\Psi|$ into 293 Pauli expectations as

$$|\Psi\rangle\langle\Psi| = \sum_{ijklm} p_{ijklm}\sigma_i^{(1)}\sigma_j^{(2)}\sigma_k^{(3)}\sigma_l^{(4)}\sigma_m^{(5)}, \tag{10}$$

where $\sigma_i = \{I, X, Y, Z\}$ are the usual Pauli matrices plus the identity. However, the number of measurements needed to obtain all 293 expectation values can be reduced[28]. This is because the measured probabilities from a measurement of a single Pauli expectation value, i.e. $\langle ZZZZZ\rangle$, can be summed in various combinations to derive other Pauli expectations values, i.e. $\langle ZIZZZ\rangle$, $\langle IZZZZ\rangle$, etc. The values derived are nothing but the marginalization of the measured probabilities over the outcome space of some set of qubits (see Supplementary information V for details). We can do the same for each term in the set of terms from the Pauli decomposition of ρ, calling it $\mathcal{S}_d$, forming a set of other Pauli terms that can be derived from the same counts. Taking the union of these sets to be $\mathcal{S}_u$, the complement $\mathcal{S}_d\backslash\mathcal{S}_u$ gives the 79 terms we only need to measure (see Supplementary information V). We measure the 79 Pauli expectation values of the terms above with respect to the state in both registers after modular exponentiation and from this we compute/derive the 293 terms in $\mathcal{S}_d$ and therefore $\mathrm{tr}(|\Psi\rangle\langle\Psi|\rho_{\mathrm{exp}})$. The measured probabilities for each term, some of them shown in Fig. 10, result in an expectation value of $\mathrm{tr}(|\Psi\rangle\langle\Psi|\rho_{7Q}) = 0.677 \pm 0.00365$ and $\mathrm{tr}(|\Psi\rangle\langle\Psi|\rho_{27Q}) = 0.626 \pm 0.00304$, which leads to

$$\begin{aligned} \mathrm{tr}(\hat{\mathcal{W}}_{\Psi}\rho_{7Q}) &= 0.0729 \pm 0.00365, \\ \mathrm{tr}(\hat{\mathcal{W}}_{\Psi}\rho_{27Q}) &= 0.124 \pm 0.00304. \end{aligned} \tag{11}$$

The results obviously fail to detect genuine multipartite entanglement, however, this does not mean entanglement is entirely absent. Consider the square of the maximum overlap between the ideal state $|\Psi\rangle$ and all pure states $|\theta\rangle$ that are unentangled product states with respect to some bipartite partition (bipartition) $\mathcal{B}$ of the qubits,

$$\max_{\theta \in \mathcal{B}} |\langle\theta|\Psi\rangle|^2 = \beta_\Psi. \tag{12}$$

Thus, any other state $|\xi\rangle$ for which

$$|\langle\xi|\Psi\rangle|^2 > \beta_\Psi \tag{13}$$

cannot be a product state with respect to the bipartition $\mathcal{B}$, implying that there is non-separability, or entanglement, across this bipartition. The above result extends to mixed states ρ_ξ due to the convex sum nature of mixed quantum states[27]. We compute Eq. (12) for all possible bipartitions of our ideal state $|\Psi\rangle$ (see Supplementary information VI for more details).

For the experimental state ρ_{7Q} we find, with the exception of the bipartition $\mathcal{B} = (c_0c_1c_2q_1)(q_0)$, that it is non-separable with respect to all other bipartitions, *i.e.* the square of the overlap between ρ_{7Q} and $|\Psi\rangle$ (~ 0.677) is greater than the maximal square overlap between $|\Psi\rangle$ and all product states in each of these bipartitions. Similarly for ρ_{27Q}, with the exception of bipartitions $\mathcal{B} = (c_0c_1c_2q_1)(q_0)$ and $\mathcal{B} = (c_0c_1c_2q_0)(q_1)$, the state is non-separable with respect to all other bipartitions. Most notably, both ρ_{7Q} and ρ_{27Q} are non-separable with respect to the bipartition $\mathcal{B} = (c_0c_1c_2)(q_0q_1)$, which is a bipartition between the control and work registers. This implies that non-separability or entanglement is present between the registers, as required for the algorithm's speedup in general[6,24,25]. Furthermore, the maximum (not necessarily global but a good proxy of it) expectation value of the operator $|\Psi\rangle\langle\Psi|$ for product states, is found via a greedy search algorithm[27] to be around 0.30, further asserting that indeed the qubits are entangled with each other in some way.

Concluding remarks

In summary, we have implemented a compiled version of Shor's algorithm on IBM's quantum processors for the prime factorization of 21. By using relative phase shift Toffoli gates, we were able to reduce the resource demands that would have been required in the standard compiled and non-iterative construction of Shor's algorithm (with regular Toffoli gates), and still preserve its functional correctness. The use of relative phase shift Toffoli gates has also allowed us to extend the implementation in Ref.[11] to an increased resolution. Moreover, while the latter implementation used only 1 recycled qubit for the control register, in contrast to our three qubits, it falls one iteration short of achieving full factoring for the reasons already mentioned. It is not clear what additional resource overheads (single and two-qubit gates) would be needed in implementing another iteration in their scheme and it is likely that these overheads are what prevented the full factoring of 21 in the photonic setup used. Furthermore, we note that in principle there is no real advantage in using three qubits for the control register as we have done here instead of one qubit recycled, as in Ref.[11]. However, in practice it is not possible at present to recycle qubits on the IBM processors and so we used three qubits instead. In future, once this capability is added, a further reduction in resources will be possible for our implementation, potentially improving the quality of the results even more.

We have verified, via state tomography, the output state in the control register for the algorithm, achieving a fidelity of around 0.70. For the verification of entanglement generated during the algorithm's operation, the resource demands of state tomography were circumvented by measuring a much reduced number of Pauli measurements to uniquely identify a quantum state[28]. However, this method is quite specialized and cannot be easily generalized to larger systems. In scaling up Shor's algorithm to higher integers beyond 21 using larger quantum systems, other methods of quantum tomography can be used to characterize the performance. These include compressed sensing[29] and classical shadows[30], which give theoretical guarantees, and improved scaling in the number of Pauli measurements and classical post-processing than standard methods. In the case when the state belongs to a class of states with certain symmetries, such as stabilizer states, only a few measurements are required for measuring the fidelity and detecting multipartite entanglement[31]. However, not all entangled states are neatly housed within these well-studied classes. Ref.[32] introduces a device-independent method for multipartite entanglement detection which scales polynomially with the system size by relaxing some constraints. Another scheme constructs witnesses that require a constant number of measurements of the system size at the cost of robustness against white noise. This provides a fast and simple procedure for entanglement detection[33]. Many fundamental questions on the subjects of quantum tomography and multipartite entanglement still remain to be answered[34] and advances will help in efficiently quantifying the performance of algorithms in larger quantum processors.

Our demonstration involves a two-fold reduction of the resource count from the full circuit in Fig. 2 via the replacement of regular Toffoli gates with relative phase variants, which is an approach that is in the spirit of the NISQ era; tailoring quantum circuits to circumvent the shortcomings of noisy quantum processors. In addition, we suspect that we can further reduce the resource count through the use of the approximate QFT[35], while still maintaining a clear resolution of the peaks in the output probability distribution. A possible avenue of future research derived from what we have reported here is the investigation and identification of scenarios where one can replace Toffoli gates with relative phase Toffoli gates while preserving the functional correctness, in a wide range of algorithms including Shor's algorithm, as seen here. In the present case, whether such an approach is special to the case of $N = 21$ or extendable to other N is not known. Ref.[23] has performed some work in this regard, however a proper analysis and systematic composition of relative phase Toffoli gates for such purposes is still an open problem. In future, a similar approach may make possible the factorization of larger numbers with adequate accuracy in resolution of the algorithm's outcomes and their characterization.

Received: 23 March 2021; Accepted: 26 July 2021
Published online: 16 August 2021

References

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. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th edn (Cambridge University Press, 2011).
3. de Wolf, R. Quantum computing. in *Lecture Notes* (2019). arXiv:1907.09415.
4. Vandersypen, L. M. K. *et al.* Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature* **414**, 883–887 (2001).
5. Peng, X. *et al.* A quantum adiabatic algorithm for factorization and its experimental implementation. *Phys. Rev. Lett.* **101**, 220405 (2008).
6. Vidal, G. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.* **91**, 147902 (2003).
7. Lu, C.-Y., Browne, D. E., Yang, T. & Pan, J.-W. Demonstration of a compiled version of Shor's quantum factoring algorithm using photonic qubits. *Phys. Rev. Lett.* **99**, 250504 (2007).
8. Lanyon, B. P. *et al.* Experimental demonstration of a compiled version of Shor's algorithm with quantum entanglement. *Phys. Rev. Lett.* **99**, 250505 (2007).
9. Politi, A., Matthews, J. C. F. & O'Brien, J. L. Shor's quantum factoring algorithm on a photonic chip. *Science* **325**, 1221–1221 (2009).
10. Lucero, E. *et al.* Computing prime factors with a Josephson phase qubit quantum processor. *Nat. Phys.* **8**, 719–723 (2012).
11. Martín-López, E. *et al.* Experimental realization of Shor's quantum factoring algorithm using qubit recycling. *Nat. Photon.* **6**, 773–776 (2012).
12. Griffiths, R. B. & Niu, C.-S. Semiclassical Fourier transform for quantum computation. *Phys. Rev. Lett.* **76**, 3228–3231 (1996).
13. Chiaverini, J. *et al.* Implementation of the semiclassical quantum Fourier transform in a scalable system. *Science* **308**, 997–1000 (2005).
14. Amico, M., Saleem, Z. H. & Kumph, M. An experimental study of Shor's factoring algorithm on ibm q. *Phys. Rev. A* **100**, 012305 (2019).
15. Pal, S., Moitra, S., Anjusha, V. S., Kumar, A. & Mahesh, T. S. Hybrid scheme for factorisation: Factoring 551 using a 3-qubit NMR quantum adiabatic processor. *Pramana* **92**, 26 (2019).
16. Xu, N. *et al.* Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system. *Phys. Rev. Lett.* **108**, 130501 (2012).
17. Saxena, A., Shukla, A. & Pathak, A. (2020). arXiv:2009.05840.
18. Parker, S. & Plenio, M. B. Efficient factorization with a single pure qubit and log$N$ mixed qubits. *Phys. Rev. Lett.* **85**, 3049–3052 (2000).
19. Beauregard, S. Circuit for Shor's algorithm using 2n+3 qubits. *Quantum Info. Comput.* **3**, 175–185 (2003).
20. Beckman, D., Chari, A. N., Devabhaktuni, S. & Preskill, J. Efficient networks for quantum factoring. *Phys. Rev. A* **54**, 1034–1063 (1996).
21. Margolus, N. Simple quantum gates. *Unpublished manuscript (Circa 1994)* (1994).
22. Song, G. & Klappenecker, A. Optimal realizations of simplified Toffoli gates. *Quant. Inf. Comp.* **4**, 361–372 (2004).
23. Maslov, D. Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization. *Phys. Rev. A* **93**, 022311 (2016).
24. Braunstein, S. L. *et al.* Separability of very noisy mixed states and implications for NMR quantum computing. *Phys. Rev. Lett.* **83**, 1054–1057 (1999).
25. Jozsa, R. & Linden, N. On the role of entanglement in quantum-computational speed-up. *Proc. R. Soc. A* **459**, 2011–2032 (2003).
26. Bourennane, M. *et al.* Experimental detection of multipartite entanglement using witness operators. *Phys. Rev. Lett.* **92**, 087902 (2004).
27. Tóth, G. Qubit4matlab v3.0: A program package for quantum information science and quantum optics for matlab. *Comput. Phys. Commun.* **179**, 430–437 (2008).
28. Ma, X. *et al.* Pure-state tomography with the expectation value of Pauli operators. *Phys. Rev. A* **93**, 032140 (2016).
29. Gross, D., Liu, Y.-K., Flammia, S. T., Becker, S. & Eisert, J. Quantum state tomography via compressed sensing. *Phys. Rev. Lett.* **105**, 150401 (2010).
30. Huang, H.-Y., Kueng, R. & Preskill, J. Predicting many properties of a quantum system from very few measurements. *Nat. Phys.* **16**, 1050–1057 (2020).
31. Tóth, G. & Gühne, O. Entanglement detection in the stabilizer formalism. *Phys. Rev. A* **72**, 022340 (2005).
32. Baccari, F., Cavalcanti, D., Wittek, P. & Acín, A. Efficient device-independent entanglement detection for multipartite systems. *Phys. Rev. X* **7**, 021042 (2017).
33. Knips, L., Schwemmer, C., Klein, N., Wieśniak, M. & Weinfurter, H. Multipartite entanglement detection with minimal effort. *Phys. Rev. Lett.* **117**, 210504 (2016).
34. Banaszek, K., Cramer, M. & Gross, D. Focus on quantum tomography. *New J. Phys.* **15**, 125020 (2013).
35. Barenco, A., Ekert, A., Suominen, K.-A. & Törmä, P. Approximate quantum Fourier transform and decoherence. *Phys. Rev. A* **54**, 139–146 (1996).

Acknowledgements

We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. We thank Taariq Surtee and Barry Dwolatzky at the University of the Witwatersrand and Ismail Akhalwaya at IBM Research Africa for access to the IBM processors through the Q Network and African Research Universities Alliance. This research was supported by the South African National Research Foundation, the South African Council for Scientific and Industrial Research, and the South African Research Chair Initiative of the Department of Science and Technology and National Research Foundation.

Author contributions

U.S. and M.T. conceived the idea and designed the experiments. U.S. performed the experiments. Both authors analyzed the results, contributed to the discussions and interpretations. U.S. and M.T. wrote the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-95973-w.

Correspondence and requests for materials should be addressed to U.S.

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