Integer Factoring with Unoperations

Paul Kohl[1]

[1]School of Computation, Information and Technology, Technical University of Munich, Theresienstr. 90, 80333 Munich, Bavaria, Germany.

Contributing authors: paul.kohl@tum.de;

Abstract

This work introduces the notion of unoperation $\mathfrak{Un}(\hat{O})$ of some operation $\hat{O}$. Given a valid output of $\hat{O}$, the corresponding unoperation produces a set of all valid inputs to $\hat{O}$ that produce the given output. Further, the working principle of unoperations is illustrated using the example of addition. A device providing that functionality is constructed utilising a quantum circuit performing the unoperation of addition – referred to as unaddition. To highlight the potential of the approach the unaddition quantum circuit is employed to construct a device for factoring integer numbers N, which is then called unmultiplier. This approach requires only a number of qubits $\in \mathcal{O}((\log N)^2)$, rivalling the best known factoring algorithms to date.

Keywords: Quantum Computing, Quantum Algorithms, Factoring, Digital Circuits

MSC Classification: 11A51 , 68Q12 , 81P65 , 81P68

1 Introduction

In the past, it seemingly was hard to construct quantum algorithms for general problems. Because of this, the best known algorithms in quantum computing are probably still Grover's search algorithm [1] and Shor's factoring algorithm [2, 3]. Nevertheless, other approaches trying to utilise quantum resources via variational approaches have been explored and can even be tested on physical devices already [4]. But none with the same critical impact as the aforementioned algorithms, as those variational approaches tend to be intended for Noisy Intermediate-Scale Quantum (NISQ) devices. Some other applications of quantum technology include the well established quantum key distribution protocols based on superposition [5] or on entanglement [6]. There also exist proposals for quantum (identity) authentication protocols [7, 8, 9] and

quantum message authentication codes [10], but some of them have weaknesses just as classical approaches [11, 9, 12, 13]. This demonstrates that there are certain fields that might take advantage of the exploitation of quantum mechanical principles. Yet, a general framework to construct quantum algorithms that have an advantage over classical algorithms for different problems remains elusive.

One group of problems that would be of interest for novel algorithms in practice is the following. Problems that have an efficient classical algorithm, however, the inverse of the problem does not have any efficient algorithm (yet). In the context of cryptography such a function is called a trapdoor one-way function [14]. If one would be able to construct a device that can efficiently reverse any efficiently computable operation, this hypothetical device would obviously open up the possibility of constructing a computationally easy inversion of such trapdoor functions.

The most famous example for a trapdoor function is of course multiplication and factoring. RSA, still one of the most widely used public-key cryptography schemes is based on the computational hardness of factoring [15]. For this trapdoor function Shor's algorithm provides an efficient algorithm on a quantum computer, thus this problem is a good candidate for comparison with the main contribution of this work, which aims at the construction of a quantum algorithm based on the concept of unoperations. This novel approach might enable a more systematic approach to constructing quantum algorithms for a certain class of problems, making it interesting for the whole field of quantum computing.

2 Methods

The striking fact is, that quantum circuits are inherently reversible due to the unitarity of quantum mechanical transformations. Therefore, in quantum computation the inversion of a computation on a quantum state to get back the original quantum state is called uncomputation. However, this implies that there is already a quantum state produced by a preceding computation which can be uncomputed. In general, one is not in the fortunate situation to have an intact quantum state produced by the algorithm to be inverted.

2.1 Unoperations

Because a readily uncomputable state is rarely the starting point I propose the notion of unoperation $\mathfrak{Un}(\cdot)$. That is, the so-called unoperation $\mathfrak{Un}(\hat{O})$ of some operation $\hat{O}$, where

$$\hat{O}(x) := \tilde{x} \tag{1}$$

is an operation on input x producing output $\tilde{x}$. Given a valid output $\tilde{x}$ of operation $\hat{O}$ the unoperation is then defined as the function $\mathfrak{Un}(\hat{O})$ that produces a set of all valid inputs x to $\hat{O}$ that produces the given output $\tilde{x}$. At the first glance, the proposed notion is not really that different from inversion of a function or uncomputation of a quantum state. At a second glance however, there is a subtle difference. The unoperation coincides with the inverse of a function in case the function is bijective, because then there is a well-defined unique pairing of any input with its (single) corresponding output. Reformulated in the framework of quantum mechanics that corresponds to the statement that they coincide if the

operator is unitary and thus has a conjugate transpose that can be applied to the state to uncompute it. Now, the unoperation is not just limited to this case. The unoperation $\mathfrak{Un}(\hat{O})$ should produce *all* valid inputs x to $\hat{O}$ given $\tilde{x}$. That is, it is defined as

$$\mathfrak{Un}(\hat{O})(\tilde{x}) := \{x \mid \hat{O}(x) = \tilde{x}\}. \quad (2)$$

This means, we want to get a set of inputs and not just a single input producing $\tilde{x}$, which can be easily illustrated for a surjective operation like addition.

2.2 Unaddition

Accordingly, we now turn to the unoperation of addition – consequently called unaddition and denoted $\mathfrak{Un}+$ as the unoperation of the infix operator $+$ for addition. Addition is a surjective operation and also a binary operation. As such it has two inputs and one output. We concentrate only on sums on field $\mathbb{N}_0$ here. For example, if the output of the addition is $\tilde{x} = 3$, then the unaddition should produce

$$\mathfrak{Un}+(3) = \{(0,3),(1,2),(2,1),(3,0)\}. \quad (3)$$

2.3 Construction of quantum circuits

To construct a quantum circuit that achieves this functionality, we first impose the following: The result of the quantum mechanical unaddition operator should be a superposition of all the elements in the set that should be produced by unaddition. The device will be constructed utilising principles from classical circuit design adapted to the intended devices and taking into account capabilities of quantum circuits. The results of the construction of the different quantum circuits, quantum gates and a full quantum algorithm for integer factoring will be given in Sec. 3 and discussed in Sec. 4.

2.4 Simulation of quantum circuits

The devices were simulated with the classical quantum circuit simulator framework from `qiskit` [16, 17] on a compute server with 64 CPU threads at 3,2 GHz maximum boost clock and 128 GiB of DDR4-3200 RAM.

3 Results

The device for unaddition, the so-called unadder was constructed in analogy to a classical digital circuit for addition. The following section clarifies the concept of this digital circuit.

3.1 Classical Ripple-Carry-Addition

The Ripple-Carry-Adder (RCA) construction of the digital adder is a divide-and-conquer approach that breaks down the $n$bit numbers a and b to be added into single bits a_i and b_i with $i \in [0, n[$. These are then added with a full-adder device for single bit addition with carry. [18, 19]

Each full-adder performs following calculation:

$$a_i + b_i + c_{\text{in},i} = sum_i + c_{\text{out},i}, \quad (4)$$

where a_i, b_i, $c_{\text{in},i}$, sum_i, and $c_{\text{out},i}$ are single bits [18, pp.188-189]. For the RCA, the carry $c_{\text{in},n-1} = 0$ for the bitwise addition of the Least Significant Bit (LSB). Additionally, the carry bit $c_{\text{out},i}$ of the addition of a less significant bit is fed into the next bitwise addition as the

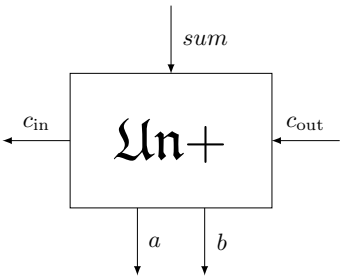

Fig. 1 Black box circuit of a full-unadder. It is the unoperation device corresponding to the full-adder used in classical circuit design construction for RCAs. It is capable of the unaddition of a 1 bit sum with the carry c_{out}, which will produce valid bit combinations for a, b, and c_{in} satisfying Eq. 4.

new $c_{\text{in},i-1}$. Like this the sum in the end can be constructed from the resulting bits sum_i in the correct order and the final carry $c_{\text{out},0}$ is the Most Significant Bit (MSB) of the resulting sum. [18, 19]

3.2 Ripple-Carry-Unadder

The analogous Ripple-Carry-Unadder (RCU) circuit utilises the same overall circuit design, but reverses the flow of information. To be able to do this, one needs to construct the unoperation of the bitwise unaddition, consequentially termed full-unadder. Its black-box circuit is shown in Fig. 1.

The RCU then takes as input the $n$bit sum in binary, such that its bits are sum_i with $i \in [0, n[$. The final carry $c_{\text{out},0}$ – which is an input now – is assumed to be 0. The output then should be the set of all n bit numbers a, b, and the carry bit $c_{\text{in},n-1}$ that satisfy

$$a + b + c_{\text{in},n-1} = sum. \tag{5}$$

In its quantum form the RCU should produce the superposition of all the different elements of the set. Thus, I impose on the quantum full-unadder that it produces a uniform distribution of the valid output combinations in case an input of $(c_{\text{out},i}, sum_i)$ can be produced by multiple combinations. This is a reasonable assumption, because a priori there is no knowledge that would suggest favouring one combination over the other. In consequence, the quantum operator of the full-unadder should satisfy the truth table given in Tab. 1.

Table 1 Truth table of a full-unadder.

Input		Output			Probability
c_{out}	sum	c_{in}	b	a	
0	0	0	0	0	1
0	1	1	0	0	1/3
		0	1	0	1/3
		0	0	1	1/3
1	0	0	1	1	1/3
		1	0	1	1/3
		1	1	0	1/3
1	1	1	1	1	1

The full-unadder acts on single bits sum and c_{out} to produce the three bits a, b, and c_{in} or a superposition of said bits in case the pair (c_{out}, sum) can be produced by addition of different combinations of these bits. The probabilities for the output states are given in the last column for the given combination.

3.2.1 Quantum circuit for full-unadders

Thus, the full-unadder can be translated to a quantum circuit acting on the initial state

$$|c_{\text{out}}\rangle\,|sum\rangle\,|000\rangle \in \mathbb{C}^{2\otimes 5}. \tag{6}$$

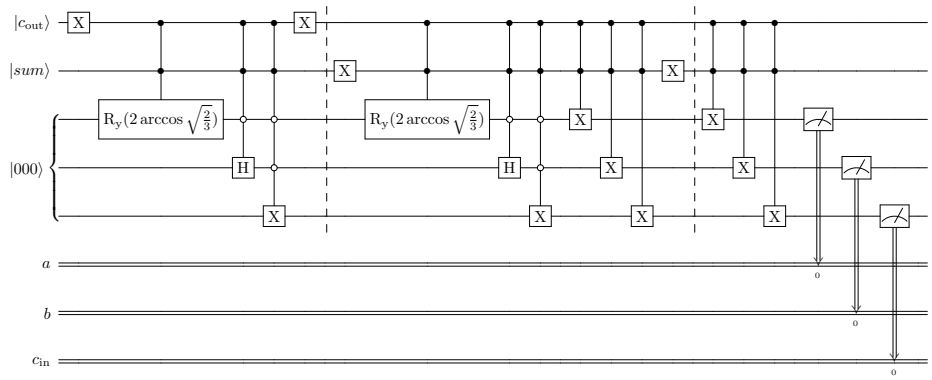

Fig. 2 A quantum circuit of a full-unadder. The action on $|c_{\text{out}}\rangle\,|sum\rangle = |00\rangle$ is implicit and the actions on $|c_{\text{out}}\rangle\,|sum\rangle \in \{|01\rangle, |10\rangle, |11\rangle\}$ (cf. Tab. 1) are described by the sections of the circuit indicated by the separation lines, respectively.

The last 3 qubits are then transformed into a, b, and c_{in} according to Tab. 1. A quantum circuit implementing this transformation is given in Fig. 2. Filled controls are activated by $|1\rangle$, while empty controls are activated by $|0\rangle$. The $\mathrm{R_y}$-gate denotes a rotation around the y-axis of the Bloch sphere describing the qubit the gate acts on. I.e.

$$\mathrm{R_y}\left(2\arccos\sqrt{\frac{2}{3}}\right) = \begin{bmatrix} \frac{\sqrt{2}}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} \end{bmatrix}. \quad (7)$$

The H- and X-gates denote the standard Hadamard- and `NOT`-gates. This circuit uses 5 qubits independently of the input and is thus $\in \mathcal{O}(1)$.

3.2.2 Optimised quantum gate for full-unadders

It is also possible to construct a quantum gate for the single-bit full-unadder acting on only 3 qubits instead. This gate then acts on the input state

$$|c_{\text{out}}\rangle\,|sum\rangle\,|0\rangle \in \mathbb{C}^{2\otimes 3}, \quad (8)$$

which are then transformed to the state

$$|c_{\text{in}}\rangle\,|b\rangle\,|a\rangle \in \mathbb{C}^{2\otimes 3}. \quad (9)$$

An explicit unitary matrix representation of such a gate is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{6}} & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{6}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{6}} \\ 0 & 0 & \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{6}} \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{3}} & 0 & 0 & \frac{2}{\sqrt{6}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (10)$$

In the following, only this optimised gate is used, as it is not only enables simulation of bigger circuits due to its smaller circuit footprint, but also due to the simpler bookkeeping of the different qubits which are to be connected for more complex quantum devices. It is denoted $\mathfrak{Un}+_{\text{opt}}$ in the circuits.

3.3 Quantum Ripple-Carry-Unadder

Constructing a quantum RCU with this gate is as simple as the construction of a classical RCA from full-adder circuit components. In this construction the $c_{\text{out},0}$ is set to $|0\rangle$. It is the first input of the first full-unadder. The second input of the first unadder is the the MSB of the n bit sum to be unadded – i.e. sum_0. The third qubit input of the first full-unadder is an ancillary qubit set to $|0\rangle$. The first full-unadder thus produces $|c_{\text{in},0}\rangle$ (which equals $c_{\text{out},1}$ used as the input for the second full-unadder) and the two MSBs of the n bit numbers a and b – i.e. a_0 and b_0. Then, the next bit of the sum sum_1, $c_{\text{out},1}$, and a fresh ancilla set to $|0\rangle$ are fed into the second full-unadder. This procedure repeats until all bits of the sum sum_i are fed into the device for $i \in [0, n[$. The final measurement on the corresponding qubits collapses the superposition to one of the possible input combinations satisfying Eq. 5. An example of 3 bit RCU with input sum 6 can be found in Appendix A Fig. 4.

3.4 Simulation of unaddition

Now follows the description of the simulations of unaddition and their results.

3.4.1 Simulation of full-unadder

The simulation of the circuit from Fig. 2 with the `QasmSimulatorPy` simulation backend from `qiskit 0.43.3` was done with 10^9 shots. The probabilities of a given measurement outcome are taken to be their relative frequencies. The simulated results can be seen in Tab. 2.

Similarly, the simulation of the optimised gate version of the full-unadder given by Eq. 10 was done with the `AerSimulator` backend from `qiskit 0.43.3` for 10^9 shots with probabilities of each outcome taken to be their relative frequencies again. These result is presented in Tab. 3. The comparison of these results quite clearly show the identical behaviour for the two devices within a reasonable margin of error.

Table 2 Simulation results for the full-unadder.

Input		Output			Probability [%]
c_{out}	sum	c_{in}	b	a	
0	0	0	0	0	100,000 000 0
0	1	1	0	0	33,332 631 9
		0	1	0	33,333 518 2
		0	0	1	33,333 849 9
1	0	0	1	1	33,334 131 1
		1	0	1	33,333 547 6
		1	1	0	33,332 321 3
1	1	1	1	1	100,000 000 0

Simulation results for the full-unadder quantum circuit (cf. Fig. 2) for all possible input bit combinations. The simulation was done for 10^9 shots and the probabilities were calculated as the relative frequency of a given output.

3.4.2 Simulation of Ripple-Carry-Unadder

The transpiled circuit if the RCU using the optimised full-unadder gate was simulated with the `AerSimulator` backend and `qiskit 0.43.3`, because the `QasmSimulatorPy` backend is limited to maximally 24 qubits. This enables the simulation of bigger unadditions. Thus, it was possible to to simulate the unaddition of input number N in the range 0 to 524 287, that is, up to $2^{19} - 1$ on the available classical hardware. The simulations results in the set of the triples (a, b, c_{in}) for n bit a, b, and 1 bit c_{in} according to Eq. 5, provided that the simulation was

Table 3 Simulation results for the optimised full-unadder.

Input		Output			Probability [%]
c_{out}	sum	c_{in}	b	a	
0	0	0	0	0	100,000 000 0
0	1	1	0	0	33,334 268 9
		0	1	0	33,331 028 9
		0	0	1	33,334 702 2
1	0	0	1	1	33,336 539 2
		1	0	1	33,332 220 2
		1	1	0	33,331 240 6
1	1	1	1	1	100,000 000 0

Simulation results for the optimised full-unadder quantum gate (cf. Eq. 10) for all possible input bit combinations. The simulation was done for 10^9 shots and the probabilities were calculated as the relative frequency of a given output.

done with a sufficient number of shots to observe all $2N + 1$ expected triples. This confirms with very solid data basis that the RCU works as expected.

3.5 Unmultiplication

The preceding elaborations are a proof of principle, yet the unoperation provided for addition does not really solve any computationally interesting problem since both directions of summation are computationally unproblematic. The problem of factoring integer numbers on the other hand is considered hard and utilisable for computational security [15, 14], which is still true classically [20], albeit not quantum [2, 3]. This problem can also be approached by the unoperation of multiplication which will return the set of all factor pairs that can be multiplied to get the input number. The unmultiplication is denoted $\mathfrak{Un}\times$. The construction of a corresponding

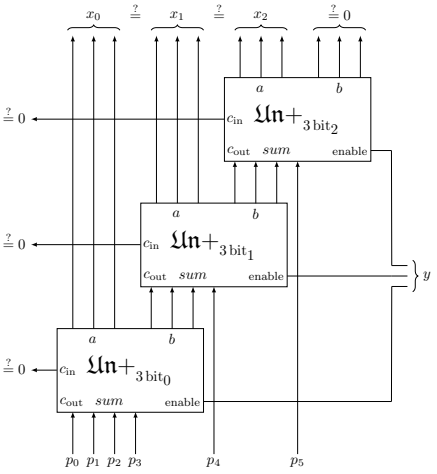

Fig. 3 Black box circuit of a 3 bit unmultiplier. It is the unoperation device corresponding to the classical circuit for the multiplier built from RCAs and in turn full-adders (cf. [19, Example 12.21]). It is capable of the unmultiplication of numbers whose factors fit into the 3 bit registers. The "enable" describes the binary multiplication with 1 or 0, i.e. if the enable signal is 0 (1) the corresponding RCU is off (on).

device again utilises the divide et impera approach which breaks down classical multiplication into a series of additions [19, pp.405 et seqq.]. The unmultiplication device now reverses the information flow and, consequently, exchanges the RCAs with RCUs. The black box construction of such a device can be seen in Fig. 3.

First, let us look at the classical binary multiplier that acts on inputs x and y to produce product p, in our example both x and y fit into 3 bit and thus p fits into 6 bit. It would then take the input x and add it onto an empty register conditioned on y – i.e. if the LSB of y is 0 the register containing the result p remains empty, if it is 1 then x is added, this 3 bit addition of course having $c_{\text{in}} = 0$ for its LSB-addition. The output of this

first addition is a 4 bit number, where c_{out} houses the MSB and the remaining bits are saved in sum. The same is then performed for the remaining bits in y with the input of the next RCA being the 3 MSBs of the previous addition, while its LSB is fixed as a resulting bit in the register of p. [19, pp.405 et seqq.]

Because the multiplier of such a construction just employs controlled additions, it is possible to unoperate it using the unadder as follows. The black-box construction of the unmultiplier reverses the flow of information to construct a superposition of all possible inputs to the corresponding classical binary multiplier with the analogous construction described above. This can be seen in Fig. 3 for the 3 bit unmultiplier. It takes the 4 MSBs (p_0, p_1, p_2, p_3) of product p as inputs to the first 3 bit RCU (constructed as in Fig. 4). Conditioned on the output of this RCU the MSB of y is reconstructed. Refer to the Sec. 3.6 for how this is done specifically.

Outputs a (as register x_1) and c_{in} of the first RCU are used for later classical post-processing, while b (3 bit) is fed into the second RCU as the 3 MSBs of the sum input and the next bit of the product (p_4) is used as the LSB. The outputs c_{in} and a (as register x_2) of this RCU are again used in post-processing and b as the 3 MSBs for the third RCU. The procedure is repeated for the third – and last – RCU, the only difference being that its b output is not fed into further devices but is also used for post-processing (cf. Sec. 3.6).

Accordingly, for an n bit unmultiplier this procedure would be repeated with n separate n bit RCUs.

3.6 Quantum circuit for unmultiplication

It is relatively straightforward to translate the black-box view (Fig. 3) to a quantum circuit. The corresponding quantum circuit that implements the unmultiplication for 3 bit with the described construction is shown in Appendix A Fig. 5 for an input product $p = 6$ (binary 000110). The register p is fed to the 3 bit RCU-gates as described in Sec. 3.5, while registers x_i with $i \in \{0, 1, 2\}$ are used as the ancilla inputs which are transformed to the a-outputs of the corresponding RCU $\mathfrak{Un}+_{\text{opt,3 bit}_i}$.

The reconstruction of the factor y of the product $p = x \cdot y$ is done with the procedure termed Quantum Feedback (QFB) for y. This y-QFB means, that the qubit of y corresponding to $\mathfrak{Un}+_{\text{opt,3 bit}_i}$ (y_i) is set to 1 with the X-gate and then activates the RCU. After the RCU, if its a-output (x_i) is 0 on all bits, y_i is set to 0, as this means that the RCU did not act, which is the equivalent of the conditional enable signal in the classical binary multiplier being 0. This is implemented with the multi-0-controlled X-gate as shown in Fig. 5 after each RCU-gate.

After the 3 RCU- and y-QFB procedures on the relevant qubits the state is measured to the corresponding classical registers c_{in,x_i}, x_i, y and $const0$ and post-processing can begin.

Because there is the initial state of the classical binary multiplier which is fixed for all inputs – that is, it uses 0 as all c_{in} of all RCAs, 0 in the b-register of the first RCA, and it has only a single x which is the a-input for all RCAs – the unmultiplier has to take care of this. The unmultiplier has outputs which will be part of the superposition constructed by the device, yet are not valid inputs to a classical binary multiplier, e.g. some

$c_{\text{in}} \neq 0$, the b-output of the last RCU $\neq 0$ or multiple different x_i. These have to be sorted out by classical post-processing of the measured outcomes after the quantum circuit. I.e. on the observed outcomes one has to perform the classical checks if

$$const0 \stackrel{?}{=} 0, \tag{11}$$

$$c_{\text{in},x_i} \stackrel{?}{=} 0 \ \forall\, i, \tag{12}$$

$$(x_i \stackrel{?}{=} x_j) \vee (x_i \stackrel{?}{=} 0) \ \forall\, i \neq j. \tag{13}$$

After that we get all possible factor pairs (x, y) that produce $p = x \cdot y$.

With the construction from Fig. 5 generalised to n bit on the available classical hardware it was possible to simulate the inputs $p \in [0, 16[$. For all simulated products the unmultiplier produces all possible factor pairs that produce the given p.

Regarding complexity, this unmultiplier construction for n bit input number N using the optimised full-unadder gate is

$$\in \mathcal{O}(n^2 + 3n) \subseteq \mathcal{O}(n^2) = \mathcal{O}(\lceil \log_2 N \rceil^2)$$

in the number of qubits. Asymptotically, one can omit the rounding $\lceil \cdot \rceil$ and the base of the logarithm, thus this unmultiplier is $\in \mathcal{O}((\log N)^2)$.

4 Discussion

In conclusion, this work shows that it is possible at least for some operations to construct the corresponding unoperation within the quantum circuit framework, which may be utilised to build an efficient device for factoring. It not only shows the existence of such an algorithm, but provides a constructive approach to implement such an algorithm for the unoperation of addition and multiplication. The proposed devices are easily implemented and simulatable on reasonably powerful classical hardware for nontrivial examples to show their working principle.

Aside from presented optimisations, other schemes for improving the performance and the resource requirements can be implemented. These could include techniques to modify the devices in order to decrease the amount of possible outcomes to be post-processed classically that do not correspond to factor pairs directly, or to recover factor pairs from such an outcome.

Also different different classical multiplication algorithms can be explored for applicability of the unoperation scheme Even other trapdoor functions where no efficient algorithm for computing their reverse direction may be unoperated, opening up the possibility for different quantum cryptanalytic algorithms.

Acknowledgements. The author acknowledges the financial support by the Federal Ministry of Research, Technology and Space of Germany (BMFTR) via the Q-TREX project with identification number 16KISR026.

Declarations

Funding

PK: Federal Ministry of Research, Technology and Space of Germany (BMFTR), 16KISR026.

Data availability

All data that support the findings of this study are available from the author upon reasonable request.

Code availability

The code for simulations is available from the author upon reasonable request.

Author contributions

PK wrote all sections, constructed all figures and tables, wrote and performed simulations, and reviewed the manuscript.

Competing Interests

The author declares no conflict of interest.

A Quantum circuits

The quantum circuit of the 3 bit RCU with input 6 can be found in Fig. 4. The unmultiplier circuit can be found in Fig. 5.

References

[1] L. K. Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery (ACM), 1996, pp. 212–219. ISBN: 978-0-89791-785-8. DOI: 10.1145/237814.237866. arXiv: quant - ph / 9605043v3 [quant-ph].

[2] P. W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. ISSN: 1095-7111. DOI: 10 . 1137 / s0097539795293172.

[3] P. W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 1994, pp. 124–134. DOI: 10 . 1109 / SFCS.1994.365700.

[4] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien. "A variational eigenvalue solver on a photonic quantum processor". In: *Nature Communications* 5.1 (July 2014), p. 4213. ISSN: 2041-1723. DOI: 10 . 1038 / ncomms5213. arXiv: 1304 . 3061v1 [quant-ph].

[5] C. H. Bennett and G. Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theoretical Computer Science* 560 (2014). Theoretical Aspects of Quantum Cryptography – celebrating 30 years of BB84, pp. 7–11. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.05.025.

[6] A. K. Ekert. "Quantum cryptography based on Bell's theorem". In: *Phys. Rev. Lett.* 67 (6 Aug. 1991), pp. 661–663. DOI: 10.1103/PhysRevLett.67.661.

[7] B.-S. Shi, J. Li, J.-M. Liu, X.-F. Fan, and G.-C. Guo. "Quantum key distribution and quantum authentication based on entangled state". In: *Physics Letters A* 281.2–3 (Mar. 2001), pp. 83–87. ISSN: 0375-9601. DOI: 10.1016/s0375-9601(01)00129-3. arXiv: quant - ph / 0102058v1 [quant-ph].

[8] C. Hong, J. Heo, J. Jang, and D. Kwon. "Quantum identity authentication with single photon". In: *Quantum Information Processing* 16 (10 Aug. 2017), p. 236. ISSN:

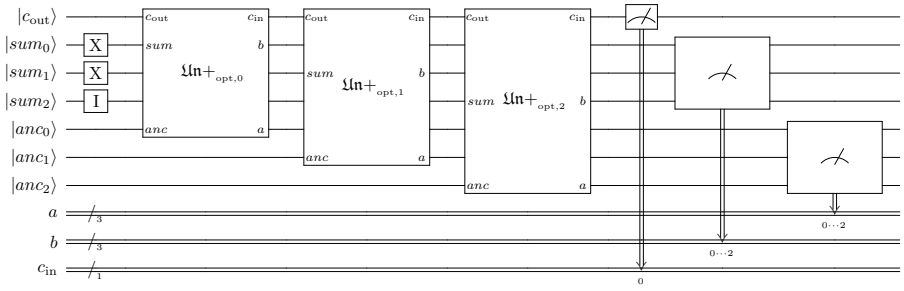

Fig. 4 Quantum circuit of a 3 bit RCU with input sum 6 (binary 110). It uses the optimised full-unadder gate $\mathfrak{Un}+_{opt}$ implemented with the unitary matrix from Eq. 10.

1573-1332. DOI: 10.1007/s11128-017-1681-0.

[9] P. Zawadzki. "Quantum identity authentication without entanglement". In: *Quantum Information Processing* 18 (1 Nov. 2018), p. 7. ISSN: 1573-1332. DOI: 10.1007/s11128-018-2124-2.

[10] M. Curty and D. J. Santos. "Quantum authentication of classical messages". In: *Physical Review A* 64.6 (6 Nov. 2001), p. 062309. ISSN: 1094-1622. DOI: 10.1103/PhysRevA.64.062309. arXiv: quant-ph/0103122v2 [quant-ph].

[11] D. Li Calsi and P. Kohl. "Comment on "quantum identity authentication with single photon"". In: *Quantum Information Processing* 23.10 (Oct. 2024), p. 357. ISSN: 1573-1332. DOI: 10.1007/s11128-024-04564-x.

[12] T.-S. Wei, C.-W. Tsai, and T. Hwang. "Comment on "Quantum Key Distribution and Quantum Authentication Based on Entangled State"". In: *International Journal of Theoretical Physics* 50 (9 2011), pp. 2703–2707. ISSN: 1572-9575. DOI: 10.1007/s10773-011-0768-0.

[13] C. E. González-Guillén, M. I. González Vasco, F. Johnson, and Á. L. Pérez del Pozo. "An Attack on Zawadzki's Quantum Authentication Scheme". In: *Entropy* 23.4 (2021). ISSN: 1099-4300. DOI: 10.3390/e23040389.

[14] W. Diffie and M. E. Hellman. "New Directions in Cryptography". In: *IEEE Transactions on Information Theory* 22.6 (Nov. 1976), pp. 644–654. ISSN: 1557-9654. DOI: 10.1109/TIT.1976.1055638.

[15] R. L. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342.

[16] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta. *Quantum computing with Qiskit*. 2024. DOI: 10.48550/arXiv.2405.08810. arXiv: 2405.08810v3 [quant-ph].

[17] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, J. M. Chow,

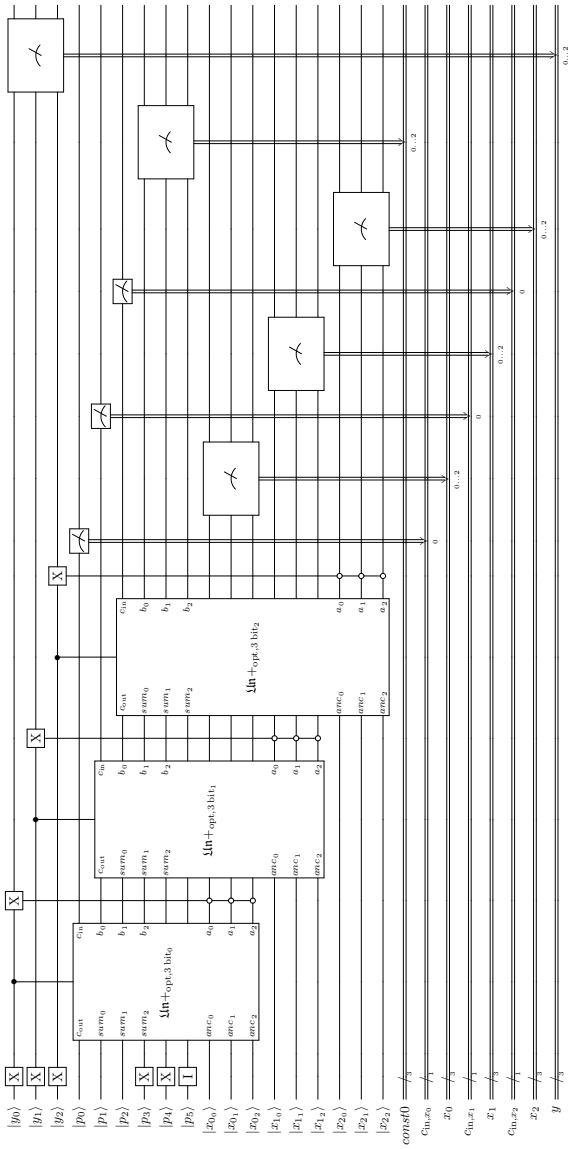

Fig. 5 Quantum circuit of a 3 bit unmultiplier with input product 6 (binary 000110, leading zeroes implicit in the circuit for clarity) with QFB to recover factor y. It uses 3 bit RCUs denoted $\mathfrak{Un}+_{\text{opt},3\,\text{bit}_i}$ for $i \in \{0, 1, 2\}$, which are implemented with optimised full-unadder gate $\mathfrak{Un}+_{\text{opt}}$ analogously to Fig. 4.

A. D. Córcoles-Gonzales, A. J. Cross, A. Cross, J. Cruz-Benito, C. Culver, S. D. L. P. González, E. D. L. Torre, D. Ding, E. Dumitrescu, I. Duran, P. Eendebak, M. Everitt, I. F. Sertage, A. Frisch, A. Fuhrer, J. Gambetta, B. G. Gago, J. Gomez-Mosquera, D. Greenberg, I. Hamamura, V. Havlicek, J. Hellmers, Ł. Herok, H. Horii, S. Hu, T. Imamichi, T. Itoko, A. Javadi-Abhari, N. Kanazawa, A. Karazeev, K. Krsulich, P. Liu, Y. Luh, Y. Maeng, M. Marques, F. J. Martín-Fernández, D. T. McClure, D. McKay, S. Meesala, A. Mezzacapo, N. Moll, D. M. Rodríguez, G. Nannicini, P. Nation, P. Ollitrault, L. J. O'Riordan, H. Paik, J. Pérez, A. Phan, M. Pistoia, V. Prutyanov, M. Reuter, J. Rice, A. R. Davila, R. H. P. Rudy, M. Ryu, N. Sathaye, C. Schnabel, E. Schoute, K. Setia, Y. Shi, A. Silva, Y. Siraichi, S. Sivarajah, J. A. Smolin, M. Soeken, H. Takahashi, I. Tavernelli, C. Taylor, P. Taylour, K. Trabing, M. Treinish, W. Turner, D. Vogt-Lee, C. Vuillot, J. A. Wildstrom, J. Wilson, E. Winston, C. Wood, S. Wood, S. Wörner, I. Y. Akhalwaya, and C. Zoufal. *Qiskit: An Open-source Framework for Quantum Computing*. Version 0.7.2. Jan. 2019. DOI: 10.5281/zenodo.2562111.

[18] W. Schiffmann and R. Schmitz. *Technische Informatik 1: Grundlagen der digitalen Elektronik*. 5th ed. Springer-Lehrbuch. Springer, 2004. ISBN: 978-3-642-18894-7. DOI: 10.1007/978-3-642-18894-7.

[19] B. J. LaMeres. "Arithmetic Circuits". In: *Introduction to Logic Circuits & Logic Design with Verilog*. 3rd ed. Springer International Publishing, 2023, pp. 389–418. ISBN: 978-3-031-43946-9. DOI: 10.1007/978-3-031-43946-9_12.

[20] A. K. Lenstra, H. W. Lenstra, M. S. Manasse, and J. M. Pollard. "The Number Field Sieve". In: *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*. STOC '90. Baltimore, Maryland, USA: Association for Computing Machinery, 1990, pp. 564–572. ISBN: 978-0-89791-361-4. DOI: 10.1145/100216.100295.