Received: 17 October 2022 | Revised: 22 June 2023 | Accepted: 2 August 2023

IET Quantum Communication

DOI: 10.1049/qtc2.12065

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Modular quantum circuits for secure communication

Andrea Ceschini | Antonello Rosato | Massimo Panella

Department of Information Engineering, Electronics and Telecommunications, University of Rome "La Sapienza", Rome, Italy

Correspondence

Massimo Panella, Via Eudossiana 18, Rome 00184, Italy.
Email: massimo.panella@uniroma1.it

Funding information

European Union, Grant/Award Numbers: NextGenerationEU, CN00000013, CUPB83C22002940006; National Centre For HPC, Big Data and Quantum Computing, Grant/Award Numbers: CN1, Spoke10; Italian Piano Nazionale di Ripresa e Resilienza (PNRR), Grant/Award Numbers: Mission4, Component2, Investment1.4

Abstract

Quasi-chaotic generators are used for producing a pseudorandom behaviour that can be used for encryption/decryption and secure communications, introducing an implementation of them based on quantum technology. Namely, the authors propose a quasi-chaotic generator based on quantum modular addition and quantum modular multiplication and they prove that quantum computing allows the parallel processing of data, paving the way for a fast and robust multi-channel encryption/decryption scheme. The resulting structure is validated by means of several experiments, which assessed the performance with respect to the original VLSI solution and ascertained the desired noise-like behaviour.

KEYWORDS

quantum computing, telecommunication security

1 | INTRODUCTION

Quasi-chaotic (QC) generators represent a particular class of pseudorandom number generators (PRNGs) with a range of implementations in different sectors. They aim at generating a pseudorandom behaviour of some produced digital sequences in order to mask the information to be processed or transmitted in a secure way [1–5]. In particular, QC generators are extremely suitable for encryption and, more in general, for encoding/decoding signals for secure communication [6–8]. Therefore, QC generators are considered particularly suitable to exploit the potentiality of discrete-time circuits in the area of secure and covert data transmission. In the past, Residue Number System (RNS) architectures have been proposed to implement QC generators [9], as they make use of modular arithmetic by which the pseudorandom behaviour can be obtained in a straightforward manner and with interesting properties regarding Very Large Scale Integration (VLSI) deploying, modularity, speed, fault tolerance and low-power consumption [10].

In this paper, we focus on the use of modular arithmetic, not necessarily based on RNS, in order to obtain a flexible implementation of a QC generator that can be successively mapped into a quantum digital circuit. To this end, a QC generator can be implemented by means of the nonlinear Infinite Impulse Response (IIR) filter shown in Figure 1, which is characterised by the following difference equation:

$$x[k] = \left\langle u[k] + \sum_{i=1}^{N} w_i \, x[k-i] \right\rangle_M , \tag{1}$$

where all algebraic operations are defined modulo M and all elements of input and output time series, as well as the filter coefficients w_i, $i = 1 \ldots N$, belong to the ring $R(M)$ of the integers modulo M. Usually, $u[k]$ is the input time series to be encrypted, while $x[k]$ is the resulting encrypted version; the N-order IIR QC generator is characterised by the set of coefficients w_i, which can be considered as the encryption key for secure communication [11].

The inverse Finite Impulse Response (FIR) system that allows the decoding of the modulated input $u[k]$ from the received sequence $\tilde{x}[k]$ is obtained by the following difference equation:

$$\tilde{u}[k] = \left\langle \tilde{x}[k] - \sum_{i=1}^{N} w_i \, \tilde{x}[k-i] \right\rangle_M , \tag{2}$$

where $\tilde{u}[k]$ is the decoder output, that is, the decoded time series relative to $u[k]$, while the decoder input $\tilde{x}[k]$ is the

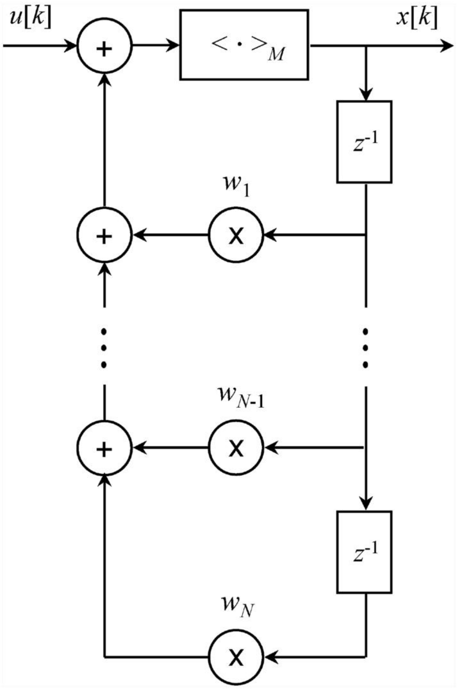


FIGURE 1 Schematic diagram of a modular quasi-chaotic (QC) generator. Despite only one modulo operation is reported, the latter can be distributed over all sums and multiplications in the filter.

sequence received on the other side of a communication channel. For the sake of simplicity, given the symmetric nature of any encoder/decoder pair, only the encoder system is considered in the present paper. In other words, the involved algebraic operations are the same and hence, the quantum implementation of both systems follows the same main rules.

In order to measure the strength of the QC modulation obtained by Equation (1), the noise-like behaviour of the zero-input response $x[k]$, $k \geq 0$, is measured when $u[k]$ is set to zero and the IIR filter is used as a QC oscillator with given initial conditions on $x[-1]$, $x[-2]$, up to $x[-N]$:

$$x[k] = \left\langle \sum_{i=1}^{N} w_i \, x[k-i] \right\rangle_M . \tag{3}$$

The advent of quantum communication can become an enabler for the development of new cryptography techniques. In particular, QC generators have the potential to be at the core of this development. To the best of our knowledge, nobody has so far implemented or discussed a quantum implementation of a nonlinear IIR filter acting as a QC generator. The original contribution of this work pertains to the design of a quantum QC oscillator through modular quantum arithmetic operations. To the best of our knowledge, it is impossible to find direct comparisons with other literature due to the novelty of our work in implementing a quantum version of a nonlinear IIR digital filter; in fact, apart from Quantum Fourier Transform [12], there are no consistent works pertaining to the implementation of quantum digital filters for signal processing. There are instead some approaches based on the generation of pseudorandom quantum states [13, 14]. Comparative studies in this regard are out of the scope of this paper, which is devoted to the introduction of the aforementioned approach. Nonetheless, despite the technological issues that still affect real quantum systems [15], we provide a preliminary circuit implementation of a QC system in the quantum framework.

QC generators could benefit from quantum computing in several ways, by using parallel encryption/decryption schemes [12] or even adopting computationally efficient neural networks via quantum superposition [16, 17]. As a matter of fact, while classical computers are based on bits as elementary units of information, with mutually exclusive values of 0 and 1, quantum devices use qubits as building blocks of the computation: the 0 and 1 states coexist simultaneously in a probabilistic superposition. The latter properly allows quantum computers to process data in parallel in a high-dimensional form with very few qubits [18], with the goal of adopting the quantum QC generators proposed herein for encryption/decryption in high-speed secure communication channels.

The primary contribution of this paper is the introduction of a novel approach to replicate the modular operations in Equation (1) with suited quantum gates, in order to obtain a QC behaviour given an initial condition. The correct functioning of the proposed quantum QC circuit is experimentally validated with the help of Qiskit$^{\text{TM}}$, which is a framework for quantum circuits' simulation developed in Python. Taking a single section of the filter, all of the possible input-output pairs obtained with the quantum circuit are compared to the desired result. Successively, the zero-input response of the whole quantum time series is validated against the classical counterpart and the effects of bit-flip measurement errors are studied for different noise levels. Finally, the autocorrelation function of the zero-input response is calculated to verify its similarity to an uncorrelated noise sequence.

2 | QUANTUM IMPLEMENTATION OF QC OSCILLATORS

From a structural perspective, there are no substantial modifications to be made to the VLSI architecture based on Complementary Metal-Oxide Semiconductor (CMOS) technology, as the one of a quantum QC oscillator should be equivalent to the classical counterpart. It can be designed via a series of modular quantum arithmetic functions such as quantum addition and quantum multiplication. Recalling the architecture in Figure 1, the design of a single section of a quantum QC filter is illustrated in Figure 2.

Given a sequence of P samples, each ith section of the quantum filter, $i = 1 \ldots N$, is composed as follows: $x_{i-1}[k]$ is the input to the ith section, w_i is the ith corresponding coefficient, $x_i[k-1]$ is a first-order delay and $x_i[k]$ is the output of the filter. Since quantum computation must be reversible, for each iteration i relative to sample k of the filter, the inputs w_i, $x_{i-1}[k]$ and $x_i[k-1]$ are also present in the output. Moreover, some ancillas initialised to $|0\rangle$ act as temporary registers to store intermediate steps of the algorithm. A part of the ancillas is used to encode the output $x_i[k]$, while the rest is restored to zero via uncomputing [19], in order to use it for further calculations. In fact, modern quantum systems are able to provide just a small number of logical qubits, therefore it is of paramount importance to save and reuse as many qubits as possible during the computation.

Considering that we are dealing with binary representations of integer numbers, working with n-qubit binary strings directly leads to a modular arithmetic representation with an $M = 2^n$ modulo; thus, the range of possible values is $[0, 2^n - 1]$. In addition, given two numbers a and b, the following property of modulo n arithmetic is exploited:

$$(a \,\mathrm{op}\, b) \bmod n = ((a \bmod n) \,\mathrm{op}\, (b \bmod n)) \bmod n, \qquad (4)$$

for any operation op $\in \{+,-,*\}$ and where $(\cdot) \bmod n$ denotes modulo n residue extraction. Binary arithmetic intrinsically leverages this property to handle the overflow with a 2^n modulus reduction.

The elementary operations computed inside a quantum QC generator are analogous to the ones in the classical circuit. A quantum version of modular addition, bit shift and modular multiplication is hereafter presented.

2.1 | Quantum modular addition

Quantum addition between two numbers a and b can be thought as a bit-wise increment of b controlled by the corresponding qubits of a, as depicted in Figure 3. This operation does not need ancillary registers to store the result, because the value of the first addend is added to the second one in an addition assignment fashion. The sum is computed modulo n due to the possible overflow. In order to deal with quantum principles, the $+=$ operator is implemented instead; it directly adds one number onto another. This is mandatory to make the computation reversible [12].

2.2 | Quantum bit shift

Bit shift is used to multiply or divide a binary number by powers of 2. In particular, a single left shift corresponds to a multiplication by 2: all the bits are shifted off to the left, so that the most significant bit is discarded and a 0 bit is inserted at the end of the binary string. In a quantum framework, left shift can be carried out with a series of 'swap' gates in cascade and an ancilla qubit initialised to $|0\rangle$, as shown in Figure 4. The swap gates are responsible for moving the qubits left. The ancilla ends up storing the most significant qubit, while the $|0\rangle$ previously contained in the ancilla is inserted at the end of the sequence. The unused output qubit at the end of the computation is a 'garbage', that is, it is an unintended byproduct of the quantum computation process, representing residual quantum information that does not contribute to the desired result but is still necessary to make the computation reversible [12].

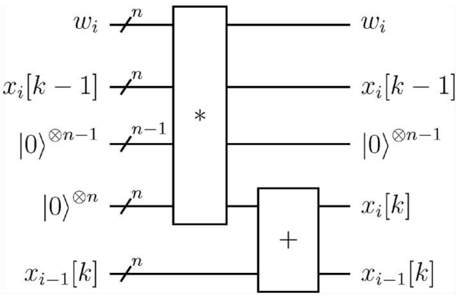


FIGURE 2 A snapshot of the ith section of the quantum quasi-chaotic (QC) filter associated with the ith delay of the IIR structure.

2.3 | Quantum modular multiplication

A quantum modular multiplier can be constructed using a combination of controlled modular adders and left shifters, as proposed in ref. [20]. The sequence of controlled additions accumulates into a product register, so that the final product ends up being the already reduced modulo n. Given two binary numbers a and b of length n, each partial product is of the form $a_i(2^i b) \bmod n$, $i = 0 \ldots (n-1)$. Each addition in the multiplier uses a value that is twice the previous value, therefore we just need to shift the value by one position for each addition. An example of quantum modular multiplication circuit is represented in Figure 5. In order to free some ancilla

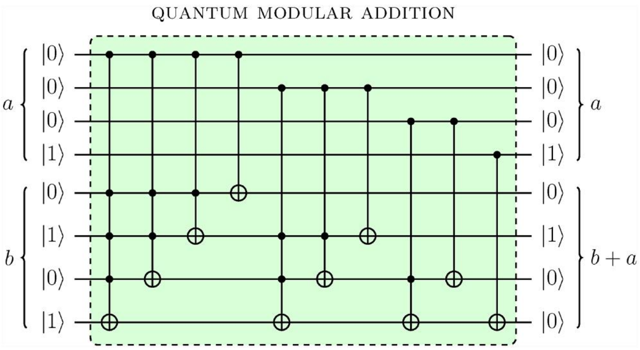


FIGURE 3 Quantum modular addition between two numbers a and b in modulo $2^4 = 16$. The least significant qubit is on top.

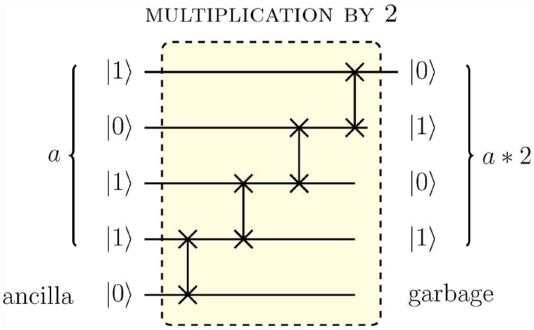


FIGURE 4 Quantum multiplication by 2 using left shifting.

qubits at the end of the computation and to retrieve all the factors of the multiplication to their original state, uncomputing is employed. It reverses the operations that entangled such qubits by applying the same quantum gates in a reversed order. This way, by reversing the left shifts transformations, the registers at the output which do not contain the multiplication result are restored to their initial value.

2.4 | Unrolled quantum QC oscillator

The architecture of a quantum QC oscillator is depicted in Figure 6. It is an unrolled concatenation of quantum circuits from Figure 2, with a cascade of N first-order quantum modular sections as described before. The output $x_i[k]$ of the ith section becomes the input of the next $(i + 1)$th section. The signals are not stored in external registers due to the no-cloning theorem, but they are processed in a forward manner. Input qubits could be put into superposition to initialise the filter with every possible combination of inputs; successive processing of the resulting output may be performed to gain quantum advantage. Such a quantum QC oscillator architecture may be employed as part of a larger algorithm, where a quantum oracle function selects desired results among the possible outputs in superposition based on arbitrary criteria.

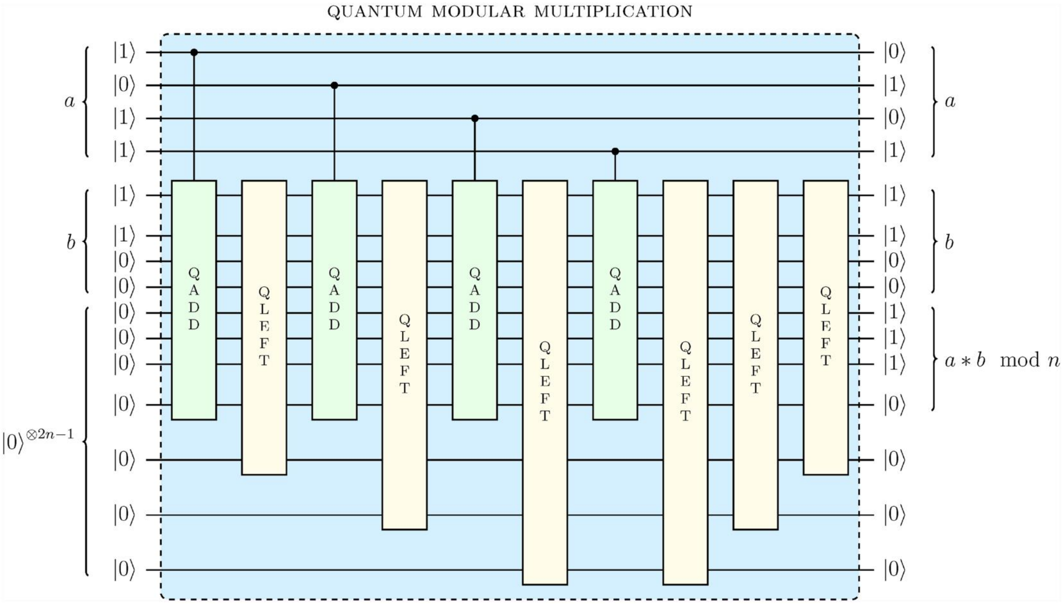


FIGURE 5 An example of 4-qubit quantum multiplier based on a sequence of controlled additions and left shifts. The QADD and QLEFT transformations correspond to quantum modular addition and left shift, respectively.

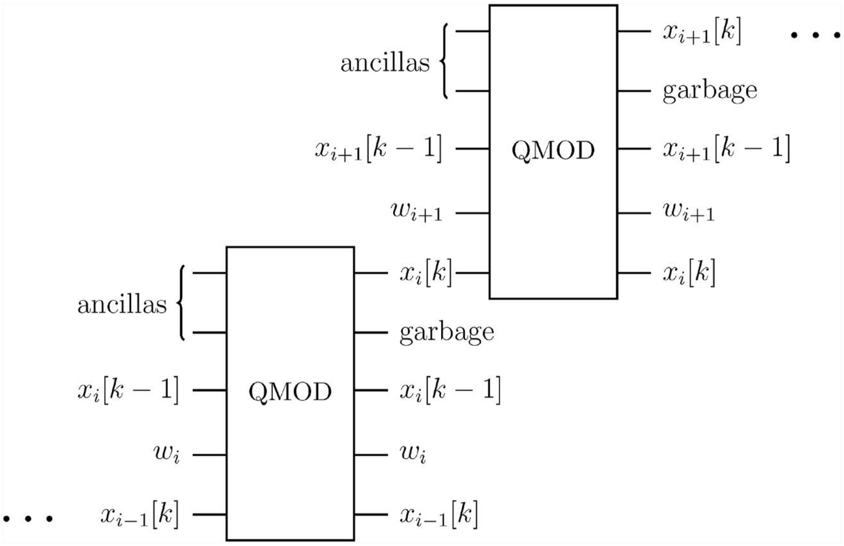


FIGURE 6 Unrolled scheme of a quantum quasi-chaotic (QC) oscillator. Each single section is denoted as QMOD.

2.5 | Complexity analysis of the quantum QC circuit

An in-depth analysis of the proposed quantum QC generator is hereafter illustrated. First, a qubit count is discussed to analyse the spatial complexity of the circuit. Every ith QMOD section of the filter receives at the input n qubit strings for the coefficient w_i, n qubit strings for the input $x_{i-1}[k]$, n qubit strings for the first-order delay $x_i[k-1]$ and $2n-1$ ancillary qubits. In total, $5n-1$ qubits are needed to perform a QMOD operation. However, at the output, $n-1$ ancillary qubits can be reused in the computation since they are restored via uncomputing. Moreover, the filter response $x_i[k]$ is also reused in the next $i+1$ section of the filter. Based on these considerations, the subsequent QMOD sections in a pipeline architecture will only need $3n$ qubits each. Therefore, the final qubit count $C(k)$ for k samples in a quantum QC generator is given by the following function:

$$C(k) = \begin{cases} 5n-1, & k=1 \\ 5n-1+3nk, & k \geq 2 \end{cases} \tag{5}$$

and hence, the complexity is linear with respect to n.

As for the spatial complexity, the quantum cost in terms of primitive gates is strictly related to the size n of the input strings. Every QMOD section has the same number of gates and is composed of a quantum modular addition QADD and a quantum modular multiplication QMUL. For the sake of simplicity, let us consider all the qubits connected to each other and a depth of 1 for a generic n-Toffoli gate. The QADD operation has a depth proportional to $\mathcal{O}(n^2)$. The QMUL operation is a sequence of n QADD and $2n-1$ QLEFT operations, therefore its depth is in the order of $\mathcal{O}(n^3)$. As a result, the overall depth of a single QMOD section of the filter is in the order of $\mathcal{O}(n^3)$.

3 | EXPERIMENTAL ANALYSIS

We performed three different experiments to validate our proposed quantum approach. All the experiments were conducted using Python 3.8 on a computer equipped with an AMD® Ryzen 7™ 5800X 8-Core CPU at 3.80 GHz and with 64 GB of RAM. The tests were conducted using Qiskit's 32-qubit simulator called `qasm_simulator`, that is, a general-purpose quantum simulator backend for testing quantum circuits both ideally and subject to noise modelling. Each input string was represented in the range [0, 15] using 4 qubits in modulo $M = 16$ arithmetic. The choice of using only 4 qubits to encode binary numbers was due to actual technological constraints. This was also the rationale behind the choice of adopting a 2^n modulo for each section of the filter, which is easy to compute through simple overflow. Further experiments with an arbitrary modulo may be performed in future works, where the number of available qubits will no more be a constraint and fault-tolerance quantum computers will be more widely available.

3.1 | Experiment A

A single section of the quantum QC oscillator was evaluated. The behaviour of such quantum circuit exactly reflected the response obtained by its classical counterpart for every possible combination of input string. Without loss of generality, only a test with a fixed coefficient $w_i = |0011\rangle = 3$ and a constant input $x_{i-1}[k] = |1101\rangle = 13$ is hereafter reported. The only parameter that varied was the delay $x_i[k-1]$, which ranged between 0 and 15. The correct functioning of such single QMOD section is presented in Figure 7.

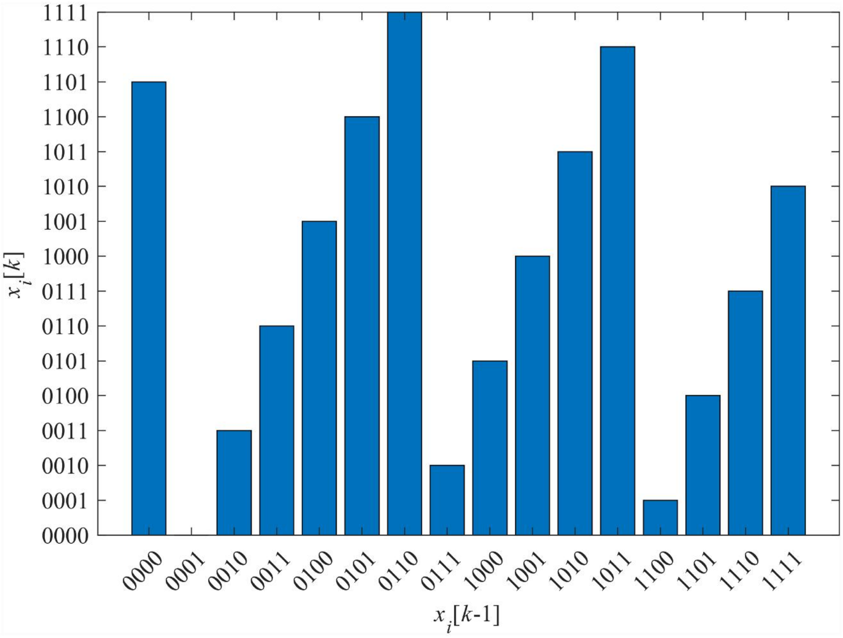
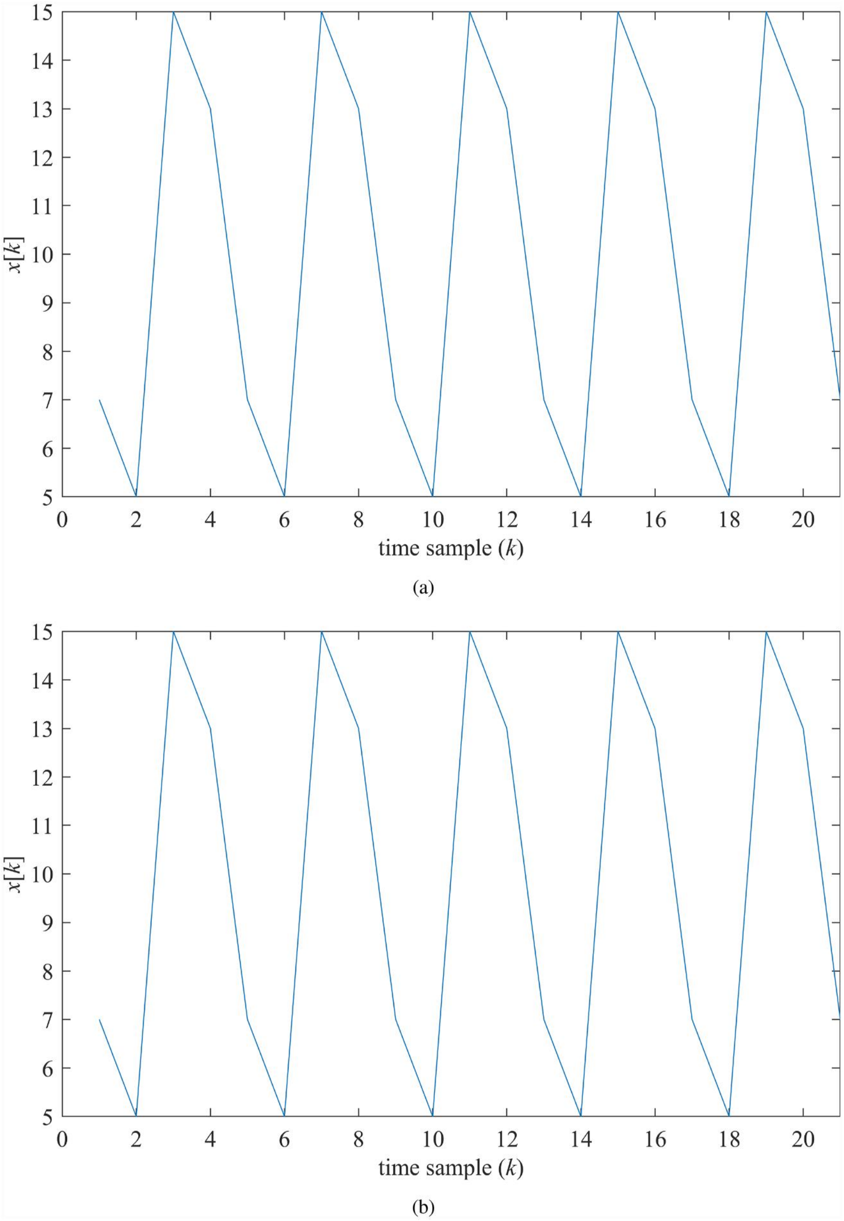


FIGURE 7 The output $x_i[k]$ of a single section of the quantum circuit with respect to different delays $x_i[k-1]$.

3.2 | Experiment B

A quantum QC generator with filter order $N = 1$ was realised to calculate the zero-input response for $k = 1\ldots50$ samples. Given the initial conditions of $w_1 = |0011\rangle = 3$, $x_1[0] = |1101\rangle = 13$, $x_0[1] = |0000\rangle = 0$, $M = 16$, the output $x_1[k]$ of the circuit was calculated and validated against the classical counterpart, calculated in Python instead. To overcome the constraints on the number of qubits in the simulator, a single QMOD section was implemented at each step rather than the entire architecture, which was too demanding in terms of resources. For every new sample k, the input parameters were adjusted accordingly with appropriate quantum gates. At the end of the QMOD computation, the qubits were reset to reuse them in the next step of the algorithm. As demonstrated by Figure 8, the behaviour of the quantum QC generator in an ideal setting, that is, without considering noise effects, perfectly overlaps to the curve originated by the classical circuit.

In order to evaluate the robustness of the proposed quantum circuit in presence of several noise levels, we also performed the experiment after injecting bit-flip measurement errors to all the qubits, which is a common case study in quantum information theory research. Bit-flip measurement errors consist in flipping the state of a qubit with probability p during a measurement. For $p = 0.02$, the behaviour of the quantum circuit still follows the expected output, as shown in Figure 9. For increasing values of probability p, the noise level progressively impairs the circuit's output; in the case $p = 0.16$, the final outcome becomes



(a)

(b)

FIGURE 8 Comparison between zero-input responses: (a) Ideal quantum quasi-chaotic (QC) oscillator; (b) classical QC oscillator based on VLSI technology. Only 21 out of 50 samples are reported for the sake of illustration.

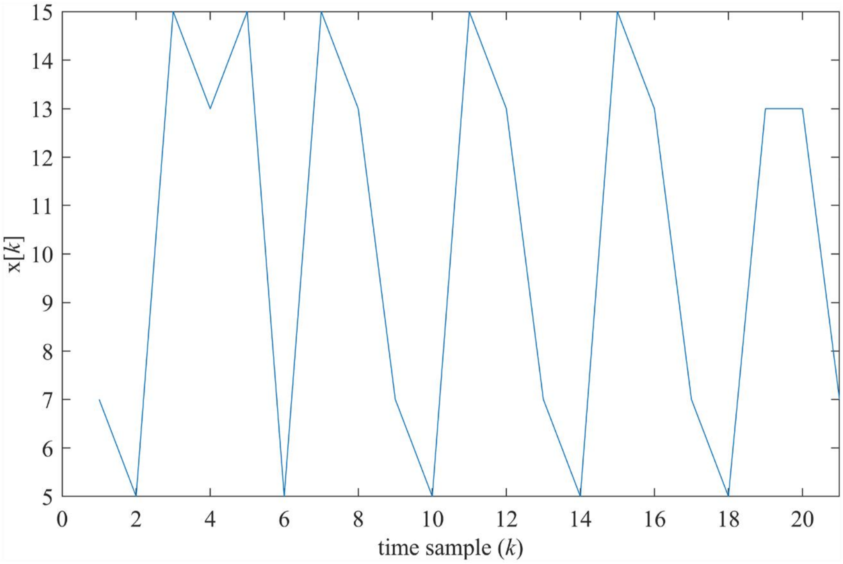


FIGURE 9 Zero-input response with a bit-flip measurement error having probability $p = 0.02$.

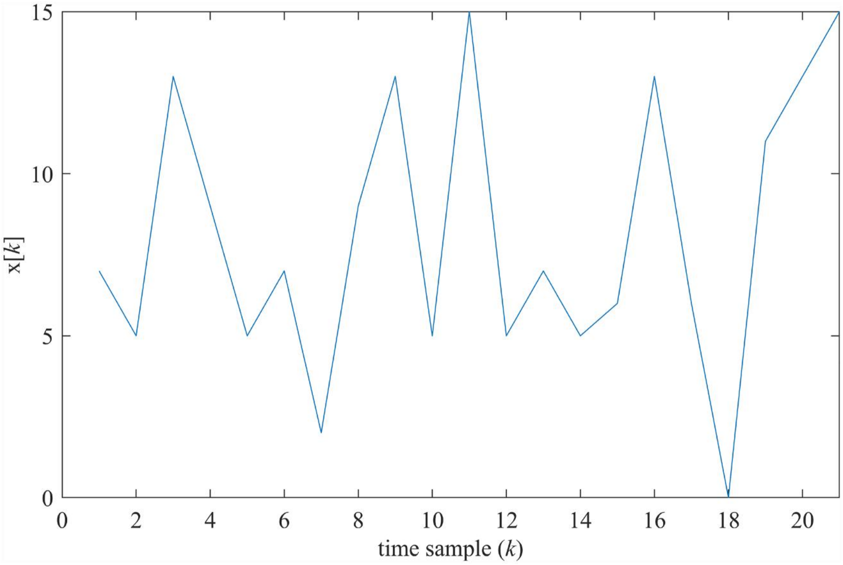


FIGURE 10 Zero-input response with a bit-flip measurement error having probability $p = 0.16$.

completely dissimilar from the desired result as illustrated in Figure 10. An overview of the Root Mean Squared Error (RMSE) with respect the ideal behaviour, averaged over 10 different runs, is represented in Figure 11 against the noise level p. As the probability of occurring in bit-flip measurement errors increases, the RMSE increases accordingly and almost linearly.

The level of quantum noise that can be tolerated in a practical quantum circuit is highly dependent on the specific algorithm being run, the error correction techniques being used and the quantum device used for the experiment. As a general rule, researchers often target a physical qubit error rate on the order of 10^{-3} or less for practical quantum computation [21]; this includes both bit-flip and phase-flip errors. The 10^{-3} level is seen as a threshold because error correction codes like the surface code, which is a leading quantum error correction scheme, is believed to effectively correct errors at this level in the near future [22, 23]. Currently, the measurement error probability is between 10^{-1} and 10^{-2} for superconducting qubits [22–24], which are the most common and widespread types of quantum devices.

Recently, researchers at Google™ claimed to be able to measure an error rate of 3.028% and 2.914% for a distance-3 qubit array and a distance-5 qubit array respectively using a fault tolerant surface code, with a readout error of 1.9% [25]. These error rates fall within the same order of magnitude as our bit-flip noise level parameter $p = 0.02$ in Sect. 3.2, under which our quantum circuit continues to operate with a well-defined behaviour. This suggests that our approach is robust against a degree of quantum noise that is comparable to what is currently achievable in state-of-the-art quantum computing hardware. Moreover, since quantum error correction seems to improve performance with increasing qubit number according to ref. [25], it provides a strong indication that our quantum circuit may perform increasingly well as the number of qubits scales in future quantum systems. This expected scalability aligns with the predicted growth in quantum computing power, and gives confidence in the practical applicability of our quantum algorithm.

3.3 | Experiment C

Strong properties of QC generators are proved when data belong to a Galois field GF(M) defined by a prime modulus M [9]. In case of even moduli of type $M = 2^n$, weakened properties pertaining to a shorter periodicity of time series and to their dependence on initial conditions are balanced by the increased efficiency on hardware implementations. However, the autocorrelation of the zero-input response for the considered quantum QC oscillators results similar to that of an uncorrelated noise sequence, as shown in Figure 12 in the case $M = 2^4$ and $N = 4$.

Remark. The choice of the modulus M is constrained by the characteristics of the input-output signals when the inherent system is applied to a digital signal processing application. For instance, it is related to the resolution (i.e., the number of levels) by which digital signals are represented. Usually, input-output signals are considered in a signed form onto a dynamic range of the modular representation: $[-(M-1)/2, (M-1)/2]$ if M is odd; $[-M/2, M/2-1]$ if M is even. Each integer in the dynamic range is mapped onto the legitimate range $[0, M-1]$, which represents the actual

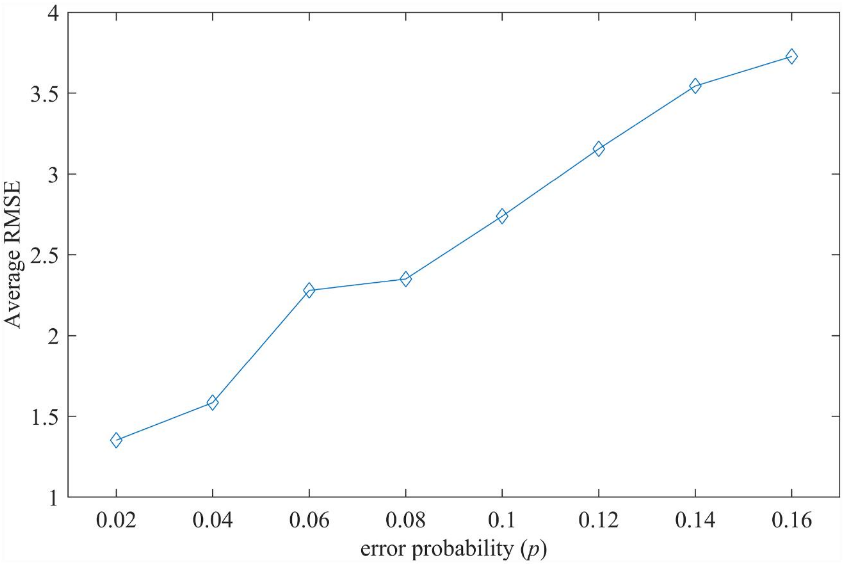


FIGURE 11 Average RMSE of zero-input response versus different noise levels.

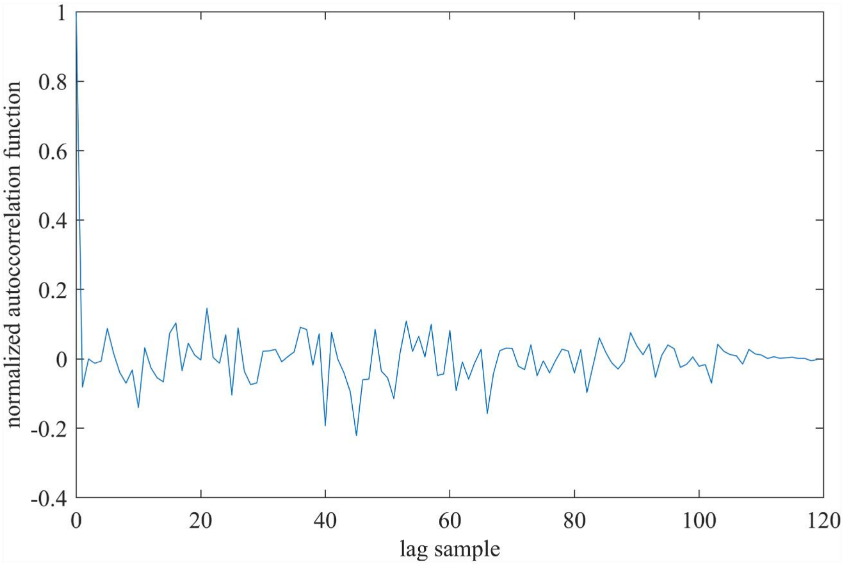


FIGURE 12 Normalised autocorrelation of the zero-input response with period 120 samples, obtained with $M = 16$ and filter coefficients $w_1 = 13$, $w_2 = w_3 = 0$, $w_4 = 1$.

computational range of the modular arithmetic. For instance, $[-(M-1)/2, -1]$ maps onto $[(M+1)/2, M-1]$ if M is odd, whereas $[-M/2, -1]$ maps onto $[M/2, M-1]$ if M is even.

4 | CONCLUSION

In this work, we presented a study on quasi-chaotic generation based on quantum modular arithmetic. We were able to experimentally validate the proposed approach with different simulations, assessing the performance with respect to the original VLSI solution. As for practical applications, we argue that quantum superposition would allow to parallelise the execution of a QC generator for different encryption keys, providing an exponential number of filter responses simultaneously. Accordingly, quantum QC oscillators should be employed as part of a larger algorithm, selecting just one result among the possible outputs based on arbitrary criteria or an oracle function could be applied to the outputs in superposition while adopting some amplitude amplification procedures.

Rather than a specific solution for all the inputs in superposition, some derived properties of the overlapped output may be considered, such as its sum, the minimum value or the presence of a certain element. Future works might also investigate on extended implementations dealing with modulo M arithmetic, where M is a prime number ensuring a stronger noise-like behaviour. In this regard, further experiments would benefit from running on actual quantum hardware to underpin the practical advantages of a quantum QC oscillator.

AUTHOR CONTRIBUTIONS

Andrea Ceschini: Conceptualisation; data curation; software; writing – original draft; writing – review and editing. **Antonello Rosato:** Formal analysis; methodology; validation; writing – original draft; writing – review and editing. **Massimo Panella:** Conceptualisation; investigation; supervision; validation; writing – original draft; writing – review and editing.

ACKNOWLEDGEMENTS

This work was in part supported by the 'National Centre For HPC, Big Data and Quantum Computing' (CN1, Spoke 10) within the Italian 'Piano Nazionale di Ripresa e Resilienza (PNRR)', Mission 4 Component 2 Investment 1.4 funded by the European Union - NextGenerationEU - CN00000013 - CUPB83C22002940006.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

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

ORCID

Massimo Panella https://orcid.org/0000-0002-9876-1494

REFERENCES

1. Eastlake, D., 3rd, Schiller, J., Crocker, S.: Randomness Requirements for Security. RFC 4086, IETF (2005)
2. Dodis, Y., et al.: Security analysis of pseudo-random number generators with input: /dev/random is not robust. In: Proc. of the ACM SIGSAC Conference on Computer & Communications Security (CCS '13), pp. 647–658 (2013)
3. Sambas, A., et al.: A novel 3D chaotic system with line equilibrium: multistability, integral sliding mode control, electronic circuit, FPGA implementation and its image encryption. IEEE Access 10, 68057–68074 (2022). https://doi.org/10.1109/ACCESS.2022.3181424
4. Benkouider, K., et al.: Dynamics, control and secure transmission electronic circuit implementation of a new 3D chaotic system in comparison with 50 reported systems. IEEE Access 9, 152150–152168 (2021). https://doi.org/10.1109/ACCESS.2021.3126655
5. Vaidyanathan, S., et al.: A new multistable jerk system with Hopf bifurcations, its electronic circuit simulation and an application to image encryption. Int. J. Comput. Appl. Technol. 67(1), 29–46 (2021). https://doi.org/10.1504/ijcat.2021.120733
6. Ananda Mohan, P.V.: Residue Number Systems: Theory and Applications. Birghauser, Mathematics, Basel (2016)
7. Alvarez, G., Li, S.: Some basic cryptographic requirements for chaos-based cryptosystems. Int. J. Bifurcation Chaos 16(8), 2129–2151 (2006). https://doi.org/10.1142/s0218127406015970
8. François, M., et al.: A new pseudo-random number generator based on two chaotic maps. Informatica 24(2), 181–197 (2013). https://doi.org/10.15388/informatica.2013.391
9. Panella, M., Martinelli, G.: An RNS architecture for quasi-chaotic oscillators. J. VLSI signal Process. Syst. signal, image video Technol. 33(1/2), 199–220 (2003). https://doi.org/10.1023/a:1021162422734
10. Panella, M., Martinelli, G.: RNS quasi-chaotic generators. Electron. Lett. 36(15), 1325–1326 (2000). https://doi.org/10.1049/el:20000952
11. Dachselt, F., Kelber, K., Schwarz, W.: Discrete-time chaotic encryption systems—part III: cryptographical analysis. IEEE Trans. Circuits Systems-I 45(9), 983–988 (1998). https://doi.org/10.1109/81.721265
12. Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, Cambridge (2010)
13. Ji, Z., Liu, Yi-K., Song, F.: Pseudorandom quantum states. In: Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference. Springer International Publishing, Santa Barbara, CA, USA (2018). August 19–23, 2018, Proceedings, Part III 38
14. Ananth, P., Qian, L., Henry, Y.: Cryptography from pseudorandom quantum states. In: Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022. Springer Nature Switzerland, Santa Barbara, CA, USA (2022). August 15–18, 2022, Proceedings, Part I Cham
15. Preskill, J.: Quantum computing 40 years later. arXiv:2106.10522v2 [quant-ph] (2021)
16. Salamat, S., et al.: RNSnet: in-memory neural network acceleration using residue number system. In: 2018 IEEE Int. Conf. on Rebooting Computing (ICRC), pp. 1–12 (2018)
17. Chervyakov, N.I., et al.: Residue number system-based solution for reducing the hardware cost of a convolutional neural network. Neurocomputing 407, 439–453 (2020). https://doi.org/10.1016/j.neucom.2020.04.018
18. Weigold, M., et al.: Expanding data encoding patterns for quantum algorithms. In: 2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C), pp. 95–101 (2021)
19. Johnston, E.R., Harrigan, N., Gimeno-Segovia, M.: Programming Quantum Computers. O'Reilly Media, Boston (2019)

20. Rines, R., Chuang, I.: High performance quantum modular multipliers. arXiv:1801.01081 [quant-ph] (2018)
21. Qin, D., Xu, X., Li, Y.: An Overview of Quantum Error Mitigation Formulas. Chinese108 Physics B (2022)
22. Fowler, A.G., et al.: Surface codes: towards practical large-scale quantum computation. Phys. Rev. 86(3), 032324 (2012). https://doi.org/10.1103/physreva.86.032324
23. Nation, P.D., et al.: Scalable mitigation of measurement errors on quantum computers. PRX Quan. 2(4), 040326 (2021). https://doi.org/10.1103/prxquantum.2.040326
24. Wu, Y., et al.: Strong quantum computational advantage using a superconducting quantum processor. Phys. Rev. Lett. 127(18), 180501 (2021). https://doi.org/10.1103/physrevlett.127.180501
25. Google Quantum, A.I., et al.: Suppressing quantum errors by scaling a surface code logical qubit. Nature 614(7949), 676–681 (2023). https://doi.org/10.1038/s41586-022-05434-1

How to cite this article: Ceschini, A., Rosato, A., Panella, M.: Modular quantum circuits for secure communication. IET Quant. Comm. 4(4), 208–217 (2023). https://doi.org/10.1049/qtc2.12065