Quantum Stud.: Math. Found. (2022) 9:155–164
https://doi.org/10.1007/s40509-021-00262-w

REGULAR PAPER

Integer numeric multiplication using quantum Fourier transform

Joseph L Pachuau · Arnab Roy · Anish Kumar Saha

Received: 26 November 2021 / Accepted: 6 December 2021 / Published online: 27 December 2021

Abstract Quantum computing is a computation process that exploits the theory of quantum physics. Quantum algorithms have the power to perform tasks with fewer queries than classical computing. To realise the advantages of quantum algorithms, arithmetic operations are required. Among them, multiplication operation is a hot topic for research. In this paper, we have proposed a generic structure for the multiplication of any two integers using quantum Fourier transform. This approach of multiplication is applicable for different quantum applications. The generic pattern and the various merits of the proposed quantum circuits are explained and analysed.

Keywords Qubit · Quantum gate · Quantum Fourier transform · Quantum phase · Integer multiplication

1 Introduction

Quantum computing solves various polynomial and non-polynomial problems with the help of superposition and quantum parallelism. Quantum logic gates are the building blocks that form the computational circuit and the gates perform some unitary operations on a two-state qubit. Unitary operations are reversible in nature, and for which, many classical circuits can not be easily converted into quantum circuits [1]. The quantum circuits give a unique output vector for a unique input vector and vice versa and for performing various merits like ancilla inputs, delay, quantum cost, garbage outputs are used. In a quantum arithmetic circuit components like reversible adders, multipliers, subtractors are the essential components used to form a circuit [2,3] and there are different design methodologies and framework for synthesizing these circuits, such as Conventional Ripple Borrow Approach, Reversible Adder without Input Carry and Reversible Adder with Input Carry [4]. There are various arithmetic operations performed in quantum circuits and a survey on it is shown in [5]. An arithmetic quantum circuit is designed in [6], where the author states that microprocessors consisting of several components including loops of data are allowed to confine only one SQF. In [7], a binary adder is designed, where the author considers Quantum Dot Cellular Automata (QCA). A technique named windowing is performed in [8], that optimizes the arithmetic quantum circuits, where small table lookups are used to reduce the circuit. It allows the control qubits used in circuits to get integrated making the circuit compact. Among most of the arithmetic operations performed in a quantum circuit, Quantum Fourier transform (QFT) is mainly used. QFT is defined as a module for arithmetic operation

J. L. Pachuau · A. Roy · A. K. Saha (✉)
Department of Computer Science and Engineering, National Institute of Technology, Silchar 788010, India
e-mail: anishkumarsaha@gmail.com

where inputs are the basic integers (qubits) in the form of basis. The addition of two integers using QFT is possible, where the inputs are 2q qubits. The first q qubits will be constant acting as a control line whereas the second will be represented in the QFT format using rotation gates. For multiplication, the same two q qubits integers are used, where the first q qubits are the control lines and the second q qubits are constant and they are transformed into QFT format using generalized controlled adders [9]. Quantum circuits consist of a unit called reversible Arithmetic Logical Unit (ALU), which performs various arithmetic and logical operations such as modular addition, subtraction, negative subtraction, bitwise exclusive-or and no-operation. In ALU only 6n reversible gates are required, for performing these basic operations [10,11] and gates like V-shaped binary adder, Feynman gates and Controlled-NOT gates are used. Using QCA, a 4 bit ALU is designed in [12]. Here the QCA is compared with CMOS, as QCA performs well in comparison to CMOS both in speed, power and area. This ALU performs all OR, XOR and AND operation.

An arithmetic operation performed using quantum computing, makes it easy, flexible and fast by parallel processing. QFT comes up as a model for performing an arithmetic operation using quantum circuits. In this paper, we show, how to perform a multiplication operation on two integers belonging to different qubit sizes. We aimed to compute multiplication of any two integer numbers, and for which the computational steps are performed in phase level of qubits using QFT and various R_k gate. The rest of the paper followed as: in Sect. 2. some of the related works on QFT performing the arithmetic operation are discussed. In Sect. 3. the quantum circuit represents the generic format for multiplication. Finally in Sect. 4 analysis and discussion are done on the proposed quantum circuit.

2 Related Works

The arithmetic calculation is carried out using the QFT, as shown in [13]. Here a quantum-based multiply-accumulate circuit (QMAC) is modeled, where two integers are multiplied, then added with the third integer, expressing as $z + y * x$. In this circuit two operators namely, phase shift gate (R_k) and a single unitary operator ($M(y, x)$) are used. In a quantum circuit, QFT is implemented using carbon-13 spins, an alanine sample, consisting of resonance frequency, coupling constants, frequency difference and relaxation time. QFT implements a 3 qubit nuclear magnetic resonance (NMR) input as shown in [14] to extract periodicity from the wave function. QFT are of two types i.e., modular and non-modular and different types of operations are performed in it. In [15], for example, an arithmetic operation used in both QFT types is performed. Apart from that, operations like the calculation of absolute value and two's complement are also performed. An operation performed on two signed integer having different qubit size are required for multiplication and division and the circuits are of fewer ancillary qubits which result in low resources and less time complexity. Arithmetic operation on a signed integer is also performed in [16]. Here on a series of inputs i.e. in QFT transform format, a method on quantum computing, a weighted sum is computed and an extension of a quantum adder is used to compute the average weight. Alternatively, quantum phase estimation is also used to compute addition. Multiplication and summation are performed securely in [17], where authors considered the novel approach of quantum computing. Here QFT is used to securely compute the operation. The summation or multiplication are computed without revealing the private inputs. One of the non-zero problems is the factoring problem, where the classical computer is said to be unable to calculate efficiently. But when in [18] this operation is performed by Shor's algorithm using quantum circuits, the problem is resolved efficiently. In Shor's algorithm, order-finding is the main problem and to perform order-finding, two registers are used. The first register consists of 2n qubits that are set to 0 and the second one consists of n qubits. The main factor for order-finding are modular exponentiation operation $ME(a, N)$ and QFT [19]. An ADD quantum circuit is built-in [20], where the addition is performed using QFT taking $n + 1$ qubits as inputs. There are no ancillary qubits in this circuit. This circuit does not apply a ripple-carry method where the carry bit is stored after addition. If QFT uses the Coppersmith's circuit having no ancillary qubits, the overall circuit size becomes larger to $O(n^2)$. In a QFT, the Fourier coefficient is an important part and encoding such Fourier coefficient along with the size of quantum becomes scares sometimes. Considering such problem authors in [21], proposed a scheme for encoding of Fourier coefficient. Along with encoding, the approximation of QFT to modulus 2^n, the circuit has an upper bound in-depth complexity of $O(\log n log(1/\epsilon))$ and

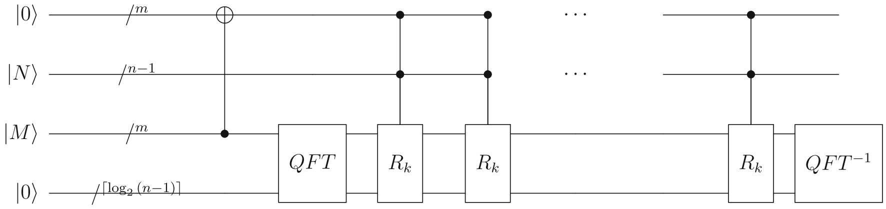


Fig. 1 Simplified symbolic structure of quantum multiplication

an error bounded by ϵ [22]. As an example, they demonstrate that while performing Shor's factoring algorithm, the depth of the circuit is found to be $O(\log n)$. They also showed that there is a lower bound in-depth complexity, $\Omega(\log n)$ in the QFT approximation. Using this approximation circuit, it is showed that QFT with an arbitrary modulus m, have a depth complexity, $O((\log\log m)(\log\log 1/\epsilon))$ and size complexity, $(\log m + \log(1/\epsilon))$.

Multiplication of two N-digit binary numbers, are computed in [23]. This is implemented in two ways: multi-tape turing machines and logical networks. For an arbitrary Z_p, the approximation algorithm is performed in quantum Fourier transform [24]. It takes only $O(n\log n)$ steps. It improves the method of Kitaev. This leads to an efficient Fourier sampling technique. It is efficient for a large class of periodic functions. For arbitrary orders, authors in [25], showed that the Quantum Fast Fourier transform (QFFT) could be precise with arbitrary orders. The QFT can be exact for any order with amplitude amplification. Kitaev is used for QFT construction. This reveals how exact the discrete quantum algorithm of Shor is rendered by QFT. For the QFT a generic framework is presented in [26]. An efficient quantum circuits for the QFT is provided having both finite Abelian and non-Abelian groups. For an efficient QFTs all families of groups as well as new families are included. Understanding the higher-form QFT symmetries is mathematically described in [27]. It is a special case of two or more groups. It discusses the generalized module spaces and cosmological defects in QFT. An interpretation of QFT anomalies is proposed. Quantum factoring is performed on an application, using an approximate version of Fourier Transform [28]. In the prospect of QFT, authors in [29], show that there exist an quantum circuits for integer multiplication of $O(n\log n\log n\log n)$ size smaller than the Schonhage-Strassen bound if there exists circuits of $O(n\log n\log n\log n)$ size performing QFT.

3 Representing QFT for multiplication

In a QFT for an input $|X\rangle \equiv |x_1x_2\ldots x_n\rangle$ of n qubits input, the generalised output in tensor product representation is expressed as (Figs. 1, 2):

$$\begin{aligned} QFT|x_1x_2\ldots x_n\rangle =& \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i0.x_n}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i0.x_{n-1}.x_n}|1\rangle)\cdots \\ & \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i0.x_2\ldots x_{n-1}.x_n}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i0.x_1.x_2\ldots x_{n-1}.x_n}|1\rangle) \end{aligned} \tag{1}$$

Lets taking an small example of input $|x\rangle = |a.b.c.d\rangle$ and the output after applying QFT is shown below and the same in Fig. 3:

$$\begin{aligned} QFT|abcd\rangle =& \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i0.d}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i0.c.d}|1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i0.b.c.d}|1\rangle) \\ & \otimes \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i0.a.b.c.d}|1\rangle) \end{aligned} \tag{2}$$

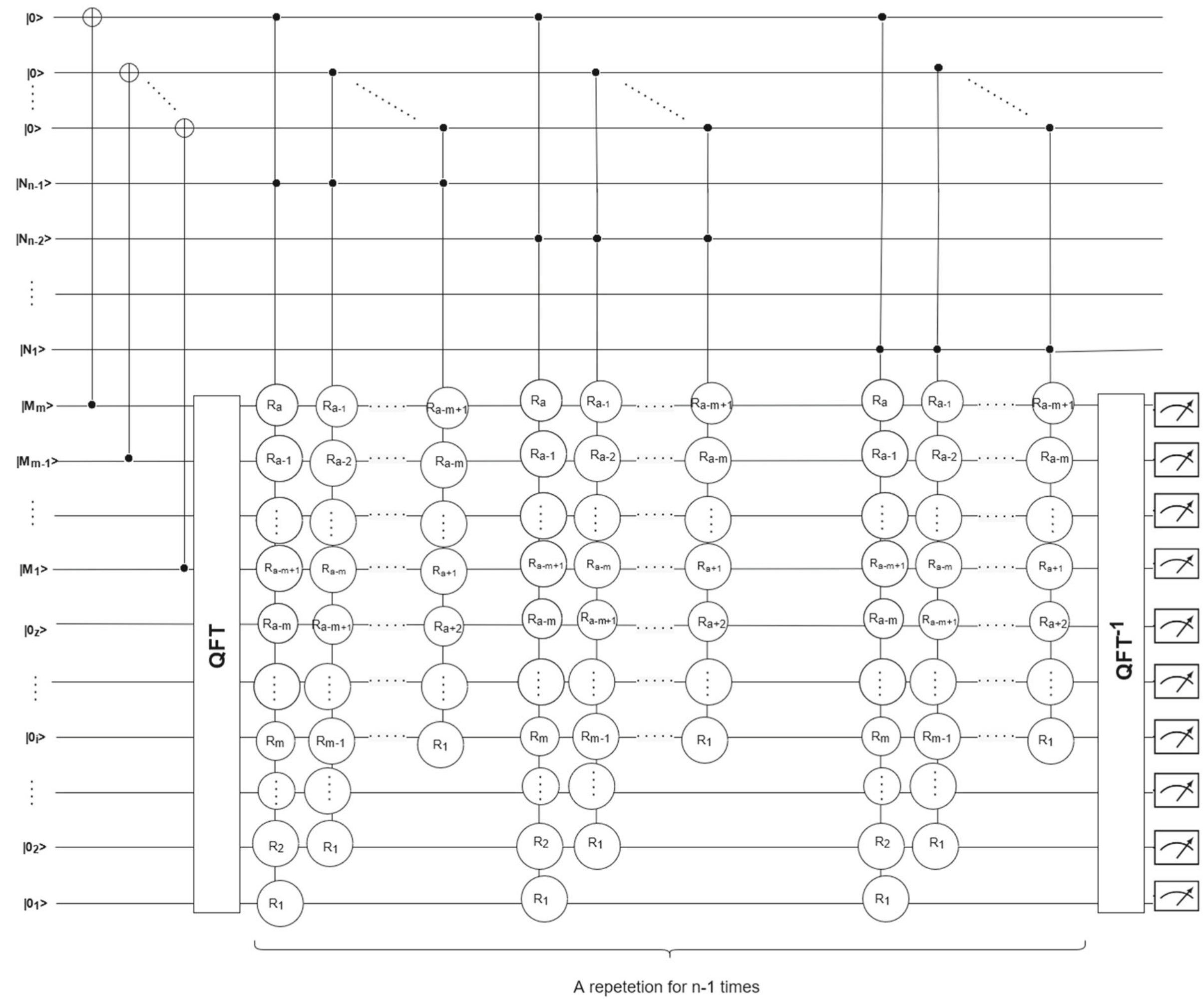


Fig. 2 Generic circuit diagram for quantum multiplication

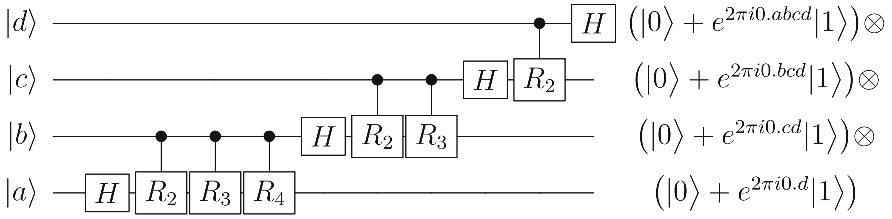

$|d\rangle$
$|c\rangle$
$|b\rangle$
$|a\rangle$
H
R_2
R_3
R_4
$\big(|0\rangle + e^{2\pi i 0.abcd}|1\rangle\big)\otimes$
$\big(|0\rangle + e^{2\pi i 0.bcd}|1\rangle\big)\otimes$
$\big(|0\rangle + e^{2\pi i 0.cd}|1\rangle\big)\otimes$
$\big(|0\rangle + e^{2\pi i 0.d}|1\rangle\big)$

Fig. 3 Quantum circuit and outputs on 4-qubit QFT

In this paper, our aim is to obtain the multiplication of any two integer numbers. For this, a Table 2 is shown for the required quantum states multiplied by various numbers. Two basic quantum gates are frequently used here, namely CNOT and R_k, and the matrix formation is given below,

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}$$

Table 1 Sequence of operation for multiplication using QFT and R_k

Number 1 in qubits	Multiplied By 1		Multiplied By 2		Multiplied By 3	
	Multiplied by 1 using QFT	Required step	Required quantum state	Required rotation	Required quantum state	Required rotation
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.00010}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.00100}\|1\rangle)\otimes$	R_4	$(\|0\rangle + e^{2\pi i0.00101}\|1\rangle)\otimes$	R_4
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.0010}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.0100}\|1\rangle)\otimes$	R_3	$(\|0\rangle + e^{2\pi i0.0101}\|1\rangle)\otimes$	R_3
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.010}\|1\rangle)\otimes$	QFT	$(\|0\rangle + e^{2\pi i0.100}\|1\rangle)\otimes$	R_2	$(\|0\rangle + e^{2\pi i0.101}\|1\rangle)\otimes$	R_2
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.10}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.00}\|1\rangle)\otimes$	R_1	$(\|0\rangle + e^{2\pi i0.01}\|1\rangle)\otimes$	R_1
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$	R_0	$(\|0\rangle + e^{2\pi i0.1}\|1\rangle)\otimes$	R_0
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.00011}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.00110}\|1\rangle)\otimes$	R_4R_5	$(\|0\rangle + e^{2\pi i0.01001}\|1\rangle)\otimes$	R_4R_5
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.0011}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.0110}\|1\rangle)\otimes$	R_3R_4	$(\|0\rangle + e^{2\pi i0.1001}\|1\rangle)\otimes$	R_3R_4
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.011}\|1\rangle)\otimes$	QFT	$(\|0\rangle + e^{2\pi i0.110}\|1\rangle)\otimes$	R_2R_3	$(\|0\rangle + e^{2\pi i0.001}\|1\rangle)\otimes$	R_2R_3
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.11}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.10}\|1\rangle)\otimes$	R_1R_2	$(\|0\rangle + e^{2\pi i0.01}\|1\rangle)\otimes$	R_1R_2
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.1}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$	R_0R_1	$(\|0\rangle + e^{2\pi i0.1}\|1\rangle)\otimes$	R_0R_1
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.00100}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.01000}\|1\rangle)\otimes$	R_3	$(\|0\rangle + e^{2\pi i0.01100}\|1\rangle)\otimes$	R_3
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.0100}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.1000}\|1\rangle)\otimes$	R_2	$(\|0\rangle + e^{2\pi i0.1100}\|1\rangle)\otimes$	R_2
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.100}\|1\rangle)\otimes$	QFT	$(\|0\rangle + e^{2\pi i0.000}\|1\rangle)\otimes$	R_1	$(\|0\rangle + e^{2\pi i0.100}\|1\rangle)\otimes$	R_1
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.00}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.00}\|1\rangle)\otimes$	R_0	$(\|0\rangle + e^{2\pi i0.00}\|1\rangle)\otimes$	R_0
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$	R_0	$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$	R_0
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.00101}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.01010}\|1\rangle)\otimes$	R_3R_5	$(\|0\rangle + e^{2\pi i0.01111}\|1\rangle)\otimes$	R_3R_5
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.0101}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.1010}\|1\rangle)\otimes$	R_2R_4	$(\|0\rangle + e^{2\pi i0.1111}\|1\rangle)\otimes$	R_2R_4
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.101}\|1\rangle)\otimes$	QFT	$(\|0\rangle + e^{2\pi i0.010}\|1\rangle)\otimes$	R_1R_3	$(\|0\rangle + e^{2\pi i0.111}\|1\rangle)\otimes$	R_1R_3
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.01}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.10}\|1\rangle)\otimes$	R_0R_2	$(\|0\rangle + e^{2\pi i0.11}\|1\rangle)\otimes$	R_0R_2
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.1}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$	R_0R_1	$(\|0\rangle + e^{2\pi i0.1}\|1\rangle)\otimes$	R_0R_1
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.00110}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.01100}\|1\rangle)\otimes$	R_3R_4	$(\|0\rangle + e^{2\pi i0.10010}\|1\rangle)\otimes$	R_3R_4
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.0110}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.1100}\|1\rangle)\otimes$	R_2R_3	$(\|0\rangle + e^{2\pi i0.0010}\|1\rangle)\otimes$	R_2R_3
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.110}\|1\rangle)\otimes$	QFT	$(\|0\rangle + e^{2\pi i0.100}\|1\rangle)\otimes$	R_1R_2	$(\|0\rangle + e^{2\pi i0.010}\|1\rangle)\otimes$	R_1R_2
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.10}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.00}\|1\rangle)\otimes$	R_0R_1	$(\|0\rangle + e^{2\pi i0.10}\|1\rangle)\otimes$	R_0R_1
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$	R_0R_0	$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$	R_0R_0
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.00111}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.01110}\|1\rangle)\otimes$	$R_3R_4R_5$	$(\|0\rangle + e^{2\pi i0.10101}\|1\rangle)\otimes$	$R_3R_4R_5$
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.0111}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.1110}\|1\rangle)\otimes$	$R_2R_3R_4$	$(\|0\rangle + e^{2\pi i0.0101}\|1\rangle)\otimes$	$R_2R_3R_4$
$\|1\rangle$	$(\|0\rangle + e^{2\pi i0.111}\|1\rangle)\otimes$	QFT	$(\|0\rangle + e^{2\pi i0.110}\|1\rangle)\otimes$	$R_1R_2R_3$	$(\|0\rangle + e^{2\pi i0.101}\|1\rangle)\otimes$	$R_1R_2R_3$
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.11}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.10}\|1\rangle)\otimes$	$R_0R_1R_2$	$(\|0\rangle + e^{2\pi i0.01}\|1\rangle)\otimes$	$R_0R_1R_2$
$\|0\rangle$	$(\|0\rangle + e^{2\pi i0.1}\|1\rangle)\otimes$		$(\|0\rangle + e^{2\pi i0.0}\|1\rangle)\otimes$	$R_0R_0R_1$	$(\|0\rangle + e^{2\pi i0.1}\|1\rangle)\otimes$	$R_0R_0R_1$

In the first step, a QFT is applied to make the state in the quantum phase. QFT^{-1} will result in the same number, giving the multiplication of $m \times 1$. For the multiplication other than 1, a different combination of R_k is required to be applied in various qubits (Fig. 4). For example, if multiplier $m = 2_{10}$ or 00010_2 and multiplicand $n = 2_{10}$ then R_0, R_1, R_2, R_3 and R_4 is required to apply in each qubit to obtain the desired phase of states. The state after applying QFT^{-1}, will provide the result of $m = 2_{10} \times 2_{10} = 00100_2$. For $m = 2_{10}$ and n changed to 3_{10}, then same pattern of R_k is required to applied once more, giving result of $m = 2_{10} \times 3_{10} = 00110_2$. The value of multiplicand (n) decides the number of times, the patterned of R_k's will be repeated. For another example, multiplier m=3_{10} or

Table 2 Table representing the sequences of repeated R_k for multiplication of generic input size, where $a \equiv m + \lceil \log_2 (n-1) \rceil$

Multiplier (m)	Multiplied by 1 using QFT	Required Step	Required Quantum State for further multiplication by 2	Required R_k gates in different qubits
$\|A_a\rangle$	$(\|0\rangle + e^{2\pi i 0.A_1 \ldots A_a}\|1\rangle)\otimes$			$R_{a-m+1}.R_{a-m+2}\ldots R_{a-1}.R_a$
$\|A_{a-1}\rangle$	$(\|0\rangle + e^{2\pi i 0.A_2 \ldots A_a}\|1\rangle)\otimes$			$R_{a-m}.R_{a-m+1}\ldots R_{a-2}.R_{a-1}$
$\vdots$	$\vdots$	QFT	$QFT(2 \times A_1 A_2 \ldots A_a)$	$\cdots\ \cdots$
$\|A_2\rangle$	$(\|0\rangle + e^{2\pi i 0.A_{a-1}A_a}\|1\rangle)\otimes$			$R_0.R_0\ldots R_1 R_2$
$\|A_1\rangle$	$(\|0\rangle + e^{2\pi i 0.A_a}\|1\rangle)\otimes$			$R_0.R_0\ldots R_0 R_1$

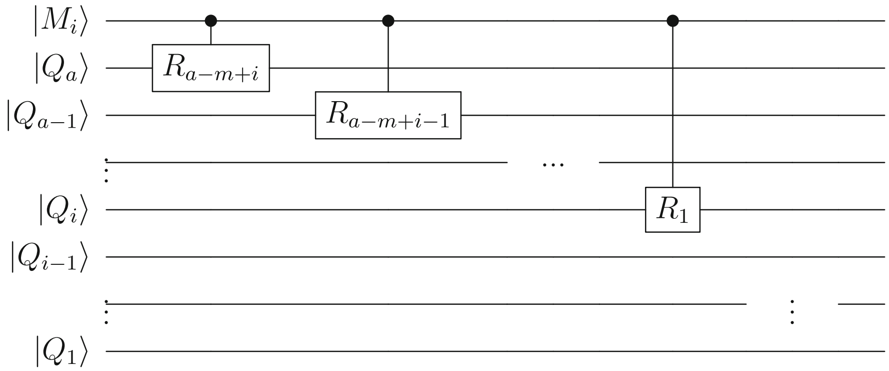


Fig. 4 Relationship of $|M_i\rangle$'s with Sequenced R_k's for each multiplicand

Table 3 Dependency of Control line with $|M_i\rangle$ position where $|M\rangle = |M_1 M_2 \ldots M_m\rangle$ and $a = m + [\log_2 (n-1)]$

Positions wise M_i acted as control line								
	M_1	M_2	M_3	$\cdots$	M_i	$\cdots$	M_{m-1}	M_m
Applied R_k's	R_{a-m+1}	R_{a-m+2}	R_{a-m+3}	$\cdots$	R_{a-m+i}	$\cdots$	R_{a-1}	R_a
	R_{a-m}	R_{a-m+1}	R_{a-m+2}	$\cdots$	$R_{a-m+i-1}$	$\cdots$	R_{a-2}	R_{a-1}
	R_0	R_0	R_{a-m+1}	$\cdots$	$R_{a-m+i-2}$	$\cdots$	R_{a-3}	R_{a-2}
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
	R_1	R_2	R_3	$\cdots$	R_i	$\cdots$	R_{m-1}	R_m
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
	R_0	R_0	R_0	$\cdots$	R_0	$\cdots$	R_1	R_2
	R_0	R_0	R_0	$\cdots$	R_0	$\cdots$	R_0	R_1

00011_2 then $R_0.R_1$, $R_1.R_2$, $R_2.R_3$, $R_3.R_4$ and $R_4.R_5$ is required to apply in various sequenced qubits to obtain the desired phases of states. For other values of m=4,5,6,7 also, the sequence of R_k is possible to obtain for the desired output. If the same process continued for any generic value of multiplier (m), the sequence of gates for various qubits are shown in Table 2. There is a connection between $R_k's$ with M_i and the corresponding positioned value of M_i acts as control line for the operation. The same is showing in Table 3. Suppose, for i^{th} position $|M_i\rangle$ in $|M\rangle$, the sequence of applied R_k over all qubits are shown in Fig. 5. $|M_i\rangle$ is acted as control qubit for all applied R_k's. An example is given for obtaining result upto 12_{10} in Fig. 5 . As $N - 1 = 3$, the same pattern of R_k's is repeated three times along with control line of $|M_1\rangle$ and $|M_2\rangle$ respectively.

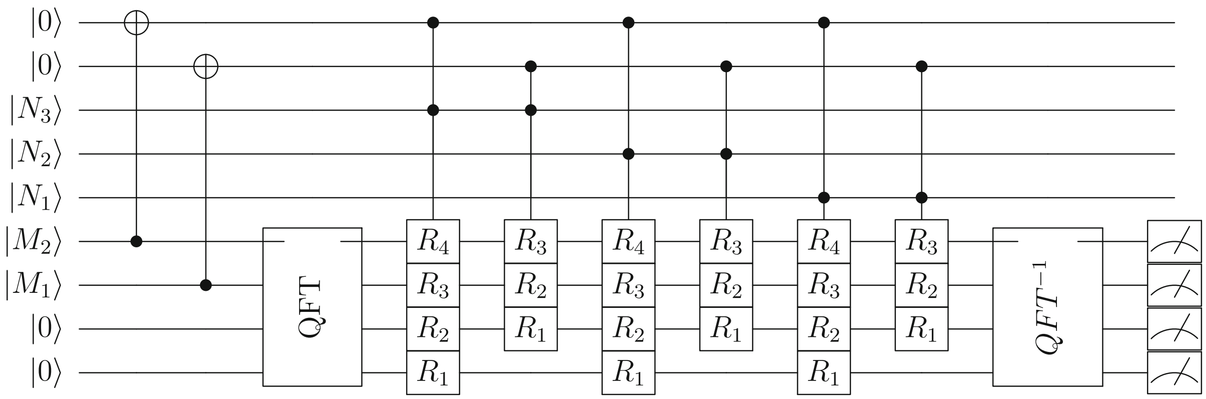


Fig. 5 An example of multiplication for obtaining result upto 12_{10} value

Algorithm 1: Quantum_multiplication

```
Inputs: $AncillaInputs|0\rangle^{\otimes z} \otimes FirstNumber|M_1 \dots M_m\rangle \otimes SecondNumber|N_1 \dots N_{n-1}\rangle \otimes AncillaInput|0\rangle^{\otimes m}$
where $z = [\log_2 (n-1)]$ and two input values are M and N-1 respectively. Ancilla Input $|0\rangle^{\otimes z}$ are additional qubits for carrying
the whole results after $QFT^{-1}$. Maximum length of the result is $m + \lceil \log_2 (n-1) \rceil$
Outputs: $|0_1 \dots 0_z M_1 \dots M_m\rangle$ Qubits associating after $QFT^{-1}$ contains multiplication result.
Procedure:
$|Q_{initial}\rangle \equiv |0\rangle^{\otimes z} \otimes |M_1 \dots M_m\rangle \otimes |N_1 \dots N_{n-1}\rangle \otimes |0\rangle^{\otimes m}$
$\equiv |0_1 \dots .0_z\rangle \otimes |M_1 \dots M_m\rangle \otimes |N_1 \dots N_{n-1}\rangle \otimes |0_1^{'} \dots 0_m^{'}\rangle$
for i=1 To m do
    $|Q_i\rangle \equiv |0\rangle^{\otimes z} \otimes |M_1 \dots M_m\rangle \otimes |N_1 \dots N_{n-1}\rangle \otimes |M_1 \dots M_{i-1}\rangle \otimes CNOT(|M_i\rangle, |0\rangle) \otimes |0\rangle^{\otimes(m-i-1)}$
    $\equiv |0\rangle^{\otimes z} \otimes |M_1 \dots M_m\rangle \otimes |N_1 \dots N_{n-1}\rangle \otimes |M_1 \dots M_i\rangle \otimes |0\rangle^{\otimes(m-i-1)}$
$|Q_{m+1}\rangle \equiv QFT(|0\rangle^{\otimes z}|M_1 \dots M_m\rangle) \otimes |N_1 \dots N_{n-1}\rangle \otimes |M_1 \dots M_m\rangle$
for i=2 To n do
    $|Q_{m+i}\rangle \equiv QFT(|i \times (0000 \dots 0M_1 \dots M_m)\rangle) \otimes |N_1 \dots N_{n-1}\rangle \otimes |M_1 \dots M_m\rangle$
$|Q_{m+n}\rangle \equiv QFT^{-1}.QFT(|n \times (0000 \dots 0M_1 \dots M_m)\rangle) \otimes |N_1 \dots N_{n-1}\rangle \otimes |M_1 \dots M_m\rangle$
$\equiv |n \times (0000 \dots 0M_1 \dots M_m)\rangle \otimes |N_1 \dots N_{n-1}\rangle \otimes |M_1 \dots M_m\rangle$
Measurement Qubits $|n \times (0000 \dots 0M_1 \dots M_m)\rangle$
```

4 Analysis and discussion

A QFT circuit consists of Hadamard, rotation and swap gates. As quantum circuits are reversible in nature, QFT^{-1} consists of the same numbers of quantum gates. The total number of quantum gates in together QFT and QFT^{-1} is as follows,

$$= 2 \times \text{Number of gates in QFT}$$

$$= 2 \times (\text{Numbers of Hadamard gates} + \text{Swap gates} + \text{Controlled R\_k's})$$

$$= 2 \times (a + a/2 + [(a-1) + (a-2) + \cdots + 3 + 2 + 1])$$

$$= a.(a+2)$$

Now, another part for calculating the total number of sequenced patterned controlled R_k gates for each multiplicand is to be,

$$[a + (a-1) + \cdots + (a-m+1)] = m.a - [1 + 2 + 3 + \cdots + (m-1)]$$

$$= m.a - \frac{m.(m-1)}{2}$$

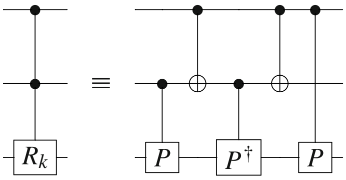
The sequenced pattern of controlled R_k gates for a multiplicand n are repeated $n-1$ times in the circuit, thus making the total number of controlled-R_k's after placing $a = m + \lceil \log_2 (n-1) \rceil$ to be as below,

$$(n-1).m.\left(\frac{m + 2\lceil \log_2 (n-1) \rceil + 1}{2}\right)$$

Finally, total number of gates for $m \times n$ generic multiplication circuit is,

$$\begin{aligned} &= \text{Number of CNOT gates} + \text{Number of gates in QFT\&}\ QFT^{-1} \\ &\quad + \text{Number of Sequenced Controlled} - R\_k\text{gates} \\ &= m + a.(a+2) + (n-1).m.\left(\frac{m + 2\lceil \log_2 (n-1) \rceil + 1}{2}\right) \end{aligned}$$

Another merit of a quantum circuit is quantum cost (QC). QC of well-known quantum gates, CNOT, Controlled-R_k, Hadamard and swap gate is 1. Here Two-Qubit Controlled R_k's is used and is possible to find the cost after fragmenting into [2 × 2] primitive quantum gates as below,



P is following the relation, $P^2 = R$ and from there, $P = \begin{bmatrix} 1 & 0 \\ 0 & e^{\pi i/2^k} \end{bmatrix}$. The QC for the above is 5, as this is composed of five [2 × 2] primitive gates [30].

The total QC of the generic circuit is,

$$\begin{aligned} &\text{QC of}\,[CNOT + QFT\&\ QFT^{-1} + Controlled\ R_k\ gates] \\ &\quad = m \times 1 + 2 \times (1 \times a + 1 \times a/2 + 1 \times [(a-1) + (a-2) + \cdots + 3 + 2 + 1]) \\ &\qquad + 5.(n-1).\left(m.(m + \lceil \log_2 (n-1) \rceil) - \frac{m.(m-1)}{2}\right) \\ &\quad = m + a.(a+2) + 5.(n-1).m.\left(\frac{m + 2\lceil \log_2 (n-1) \rceil + 1}{2}\right) \end{aligned}$$

Critical path delay is another parameter to measure the maximum delayed path in a circuit. The delay of a [1 × 1] or [2 × 2] primitive gates are considered as 1Δ. The well-known delay of quantum gates, Hadamard, R_k gate, and swap gates are all 1Δ respectively. The delay of a two-controlled-R_k gate is 4Δ.

The critical path delay of QFT as well as QFT^{-1} is $(2a-2)\Delta$. In this generic circuit, the R_k gates are independent of each other, but they are dependent on both QFT and CNOT gates. So, the total critical path delay of this circuit is,

$$\begin{aligned} &\text{Delay of QFT} + \text{Delay of Sequenced Controlled}\ R_k\text{gates} + \text{Delay of}\,QFT^{-1} \\ &\quad = (2a-2)\Delta + 4m(n-1)\Delta + (2a-2)\Delta \\ &\quad = (4mn - 4m + 4a - 4)\Delta \\ &\quad = (4mn + 4\lceil \log_2 (n-1) \rceil - 4)\Delta \end{aligned}$$

There are some unwanted qubits in the output, called garbage output and there are some fixed inputs called Ancilla inputs. Here, $|N_1 \ldots N_{n-1}\rangle \otimes |0^{\otimes m}\rangle$, are garbage outputs and $|0^{\otimes z}\rangle$ & $|0^{\otimes m}\rangle$ are Ancilla inputs. There is a total of $m + (n-1)$ garbage outputs and $m + z$ Ancilla inputs in the circuit.

Table 4 Various analysis for proposed quantum multiplication circuit

Merits	Value
Gate Count	$m + a.(a+2) + (n-1).m.\left(\frac{m+2\lceil\log_2(n-1)\rceil+1}{2}\right)$
QC	$m + a.(a+2) + 5.(n-1).m.\left(\frac{m+2\lceil\log_2(n-1)\rceil+1}{2}\right)$
Critical Path Delay(Worse Case)	$(4mn + 4\lceil\log_2(n-1)\rceil - 4)\Delta$
Garbage output	$m + n - 1$
Ancilla Input	$m + z$
Complexity	$O(m \cdot n)$

There are two iterations like processes performed in this circuit. Applying CNOT gates is one type of this process that repeats m times. Another is the applying of two-controlled R_k gates, where the process is iterated for $m.(n-1)$ times. So, the complexity for performing multiplication operation is calculated as,

$$
\begin{aligned}
&= O(m + m \cdot (n-1)) \\
&= O(m \cdot n)
\end{aligned}
$$

5 Conclusion

In this paper, we present a novel and efficient quantum approach to computing numeric multiplication in a reversible quantum circuit. The main goal is to present a generalized form of a quantum arithmetic circuit to compute the multiplication of any two integers. This approach basically describes the use of QFT and R_k gates, for multiplying any two integers, where all computation is performed in phase level of qubits. In this multiplication process, two iteration processes are performed, one is applying of CNOT gates and another is the applying of two-controlled R_k gates, where a different combination of R_k is required to be applied in various phase leveled qubits. Finally, the paper is concluded with the analysis of this arithmetic quantum circuit, using various parameters like the number of quantum gates, quantum cost, complexity and so on. In this paper, it is found that the complexity for performing multiplication operation is $O(m \cdot n)$.

Data Availability Statement Data sharing not applicable to this article as no data-sets were generated or analysed during the current study.

Declarations

Conflict of interest On behalf of all authors, Anish Kumar Saha states that there is no conflict of interest.

References

1. Nielsen, M. A., Chuang, I. L.: Quantum computing and quantum information (2000)
2. Biswas, A.K., Hasan, M.M., Chowdhury, A.R., Babu, H.M.H.: Efficient approaches for designing reversible binary coded decimal adders. Microelectron. J. **39**, 1693–1703 (2008)
3. Bruce, J., Thornton, M. A., Shivakumaraiah, L., Kokate, P., Li, X.: Efficient adder circuits based on a conservative reversible logic gate. In: Proceedings IEEE Computer Society Annual Symposium on VLSI. New Paradigms for VLSI Systems Design. ISVLSI 2002, IEEE, pp. 83–88 (2002)
4. Thapliyal, H.: Mapping of subtractor and adder-subtractor circuits on reversible quantum gates. In: Transactions on Computational Science, vol. XXVII, pp. 10–34. Springer, Berlin (2016)

5. Takahashi, Y.: Quantum arithmetic circuits: a survey. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **92**, 1276–1283 (2009)
6. Tanaka, M., Yamanashi, Y., Kamiya, Y., Akimoto, A., Irie, N., Park, H.-J., Fujimaki, A., Yoshikawa, N., Terai, H., Yorozu, S.: A new design approach for high-throughput arithmetic circuits for single-flux-quantum microprocessors. IEEE Trans. Appl. Supercond. **17**, 516–519 (2007)
7. Hänninen, I., Takala, J.: Binary adders on quantum-dot cellular automata. J. Signal Process. Syst. **58**, 87–103 (2010)
8. Gidney, C.: Windowed quantum arithmetic. arXiv:1905.07682 (arXiv preprint) (2019)
9. Pavlidis, A., Floratos, E.: Arithmetic circuits for multilevel qudits based on quantum fourier transform. arXiv:1707.08834 (arXiv preprint) (2017)
10. Thomsen, M.K., Glück, R., Axelsen, H.B.: Reversible arithmetic logic unit for quantum arithmetic. J. Phys. A Math. Theor. **43**, 382002 (2010)
11. Oskin, M., Chong, F.T., Chuang, I.L.: A practical architecture for reliable quantum computers. Computer **35**, 79–87 (2002)
12. Waje, M.G., Dakhole, P., Design and implementation of 4-bit arithmetic logic unit using quantum dot cellular automata, in, : 3rd IEEE international advance computing conference (IACC). IEEE **2013**, 1022–1029 (2013)
13. Maynard, C.M., Pius, E.: A quantum multiply-accumulator. Quantum Inf. Process. **13**, 1127–1138 (2014)
14. Weinstein, Y.S., Pravia, M., Fortunato, E., Lloyd, S., Cory, D.G.: Implementation of the quantum Fourier transform. Phys. Rev. Lett. **86**, 1889 (2001)
15. Şahin, E.: Quantum arithmetic operations based on quantum Fourier transform on signed integers. arXiv:2005.00443 (arXiv preprint) (2020)
16. Ruiz-Perez, L., Garcia-Escartin, J.C.: Quantum arithmetic with the quantum Fourier transform. Quantum Inf. Process. **16**, 152 (2017)
17. Shi, R.-H., Mu, Y., Zhong, H., Cui, J., Zhang, S.: Secure multiparty quantum computation for summation and multiplication. Sci. Rep. **6**, 1–9 (2016)
18. Shor, P. W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, IEEE, pp. 124–134 (1994)
19. Beauregard, S.: Circuit for Shor's algorithm using 2n+ 3 qubits. arXiv:quant-ph/0205095 (arXiv preprint) (2002)
20. Draper, T. G.: Addition on a quantum computer. arXiv:quant-ph/0008033 (arXiv preprint) (2000)
21. Zhou, S., Loke, T., Izaac, J.A., Wang, J.B.: Quantum Fourier transform in computational basis. Quantum Inf. Process. **16**, 82 (2017)
22. Cleve, R., Watrous, J.: Fast parallel circuits for the quantum fourier transform. In: Proceedings 41st Annual Symposium on Foundations of Computer Science, IEEE, pp. 526–536 (2000)
23. Schönhage, A., Strassen, V.: Schnelle multiplikation grosser zahlen. Computing **7**, 281–292 (1971)
24. Hales, L., Hallgren, S.: An improved quantum Fourier transform algorithm and applications. In: Proceedings 41st Annual Symposium on Foundations of Computer Science, IEEE, pp. 515–525 (2000)
25. Mosca, M., Zalka, C.: Exact quantum Fourier transforms and discrete logarithm algorithms. Int. J. Quantum Inf. **2**, 91–100 (2004)
26. Moore, C., Rockmore, D., Russell, A.: Generic quantum Fourier transforms. ACM Trans. Algorithms (TALG) **2**, 707–723 (2006)
27. Sharpe, E.: Notes on generalized global symmetries in qft. Fortschr. Phys. **63**, 659–682 (2015)
28. Coppersmith, D.: An approximate Fourier transform useful in quantum factoring. arXiv:quant-ph/0201067 (arXiv preprint) (2002)
29. Ahokas, G., Cleve, R., Hales, L.: The complexity of quantum fourier transforms and integer multiplication. In: Proceedings of ERATO Conference on Quantum Information Science (EQIS2003), volume 1 (2003)
30. Mohammadi, M., Eshghi, M.: On figures of merit in reversible and quantum logic designs. Quantum Inf. Process. **8**, 297–318 (2009)

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