IRSTI 27.35.57 https://doi.org/10.26577/JPCSIT.2023.v1.i1.09

A.A. Mukhanbet*, M.T. Nakibayeva, B.S. Daribayev

Al-Farabi Kazakh National University, Almaty, Kazakhstan
*e-mail: mukhanbetaksultan0414@gmail.com

IMPLEMENTATION OF QUANTUM ARITHMETIC OPERATIONS WITH INTEGER CHARACTERS USING THE QUANTUM FOURIER TRANSFORM

Abstract. It has been proven that a quantum computer is superior to an electronic computer in solving some NP problems. Based on quantum operations, this article proposes a new quantum sum and quantum multiplication, and then the floating point quantum multiplier and quantum sum are created on the basis of fixed point number operations. These studies lay the foundation for the quantum implementation of digital filters. This article provides a new way to calculate the summator on a quantum computer. This method uses the quantum Fourier transform (QFT) and reduces the number of qubits needed to be added by eliminating the need to temporarily transfer bits. This approach also allows you to add a classical number to a quantum superposition without encoding a classical number in a quantum register. This method also allows for mass parallelization during its execution. Adding and multiplying capabilities based on QFT are improved with some changes. The proposed operations are compared with the operations of close quantum arithmetic.

Key words: quantum computer, quantum operations, quantum Fourier transform, arithmetic operations.

1 Introduction

In recent years, quantum computing has gained attention for its ease of solving complex problems of classical computing. After Shor proposed the quantum factoring algorithm, researchers' interest in quantum arithmetic operations increased. Quantum arithmetic operations are necessary in many studies, such as quantum signal processing, quantum machine learning. In particular, in the processing of quantum images, arithmetic operations are used in many processes, such as steganography, boundary detection and pattern recognition.

Since Feynman's time, many scientists have believed that the computing power of a quantum computer is unparalleled in an electronic computer [1]. In 1992, Deutsch proposed a quantum computer model that confirmed that the Deutsch problem could be solved in a quantum computer faster than in an electronic computer [2]. Both shore's quantum factorization algorithm and Grover's search algorithm have shown that quantum computers have great computational advantages over traditional electronic computers in some aspects [3]. Quantum computing and quantum computers have attracted the broad attention of scientists [4,5]. Two important applications of quantum computing are quantum cryptography [6,7] and quantum image processing [8]. Since quantum information is not subject to cloning, quantum integration is an important physical guarantee of safe communication [9]. Some chaotic systems implemented by quantum ventilation circuits can be used as pseudo-random sequence generators in image cryptosystems to realize the encryption and decryption of quantum images. a method of using the product and divisor of a regular comma number using basic arithmetic devices, especially quantum valve circuits, using quantum valve circuits. The research work paves the way for the implementation of digital operations of a fixed point using a quantum computer.

To improve the time and memory complexity of shore's quantum factoring algorithm, the first elementary operations of quantum arithmetic are represented by modular sum, modular multiplier and modular construction operations. Gossett showed how to design modular arithmetic elements from quantum valves using the transport-storage method in a classical computer. Draper proposed a new method for calculating amounts on a quantum computer. This method uses the quantum Fourier transform (QFT) and reduces the number of transport cubes. Proposed a new scheme of quantum addition by transmitting pulses of linear depth. This new connection scheme uses only one auxiliary qubit instead of the many auxiliary qubits used in previous connection circuits. Takahashi and Kunihiro proposed a quantum sequence based on the pulsation transfer method to add two N-bit binary numbers that do not use auxiliary qubits. They focused on reducing depth and

created a fast quantum circuit for inclusion using the classical transfer prediction technique. Takahashi and Kunihiro combined a modified version of the quantum switch with a non-uniform Takahashi transport using the quantum switch in parallel. This modified Switch used multiple qubits and retained auxiliary qubits.

Traditionally, additional algorithms for a quantum computer copied their classical counterparts with the necessary extensions for reversible computing. Fast quantum addition algorithms use transport-storage methods, but follow the classical model. However, a great additional algorithm for a quantum computer may not look like its classic counterpart. This article presents a new paradigm of addition in a quantum computer. The addition method used takes two values a and b, calculates the Quantum Fourier Transform (QFT) of F(A), and then uses b to transform F(A) to F(A + B). An inverse quantized Fourier transform can then be applied and the sum recovered. Since the calculation of the conversion before and after the accumulation is associated with certain costs, the maximum number of calculations must be performed within the conversion range before it is removed from it. A number of articles devoted to the implementation of the application in the classical quantum computer have been published [2]. All implementations use at least 3N qubits to add two N-bit numbers. The method presented here follows the scheme shown in [7]. The sum consists of two main unitary computing units.

In addition, new operations of subtraction, division and raising to a power are proposed based on the QFT. The presented arithmetic operations can perform non-modular operations with all numbers without restrictions, using few resources. In addition, new quantum schemes of two, absolute and complementary operations for comparison are proposed using addition and subtraction operations proposed on the basis of QFT.

2 Methodology

Classical and quantum addition. A full adder is a logic circuit used by classical computers to perform addition of up to 3 bits.

The circuit of the full adder has 3 inputs: A, B and Cin (the English word “Carry in” is shortened because it is taken from the previous full adder, because they can be connected together)

It also has 2 outputs called Sum and Cout (abbreviation of “Carry out” because it gives a bit to Cin of the next adder). The classic scheme of the adder is shown in figure 1.

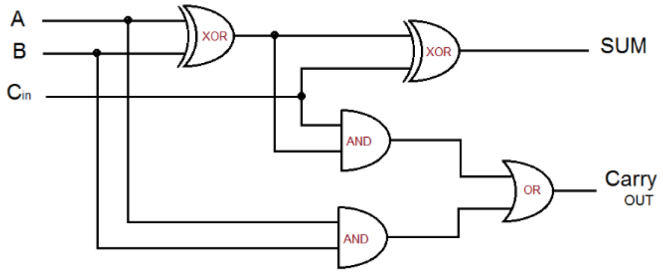

Figure 1 – Classic adder scheme

Accordingly, it is possible to construct a truth table of 2-bit values. The result can be seen in table 1.

Table 1 – Classic summator reality table

A	B	Cin	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

To implement a full summator in a quantum computer, we need 4 qubits (i.e. 1 for each input and output of the full summator).

• Zero Q0 qubit: Required to enter A;
• First Q1 qubit: required to enter B;
• Second Q2 qubit: needed to input Cin;
• First qubit Q1: Required for Sum, the first qubit is used;
• Fourth qubit Q4: Required for Cout.

To construct the addition operator, we need a few basic quantum logic valves, which we use to perform the step-by-step part of the addition process:

• Quantum valve X behaves non-classical, translating the qubit from state / 0 > to state / 1 > and vice versa;

• The CX valve (controlled – X) acts on two qubits at once, one of which is called a control qubit, and the other is called a target qubit. This valve applies the X valve to the target qubit if the control qubit is in the / 1>position;

• CCX (Controlled – Controlled – X) valve affects three qubits simultaneously, two control qubits and one target qubit. It applies valve x to the target qubit if both control qubits are in state |1>.

To calculate the carry bit, we need a valve that receives three input qubits: a carry qubit from the previous column and one qubit from each term. If at least two input qubits are in state |1>, the valve performs the operation of switching the qubit to state |1>. The truth table for the transfer valve can be seen in Table 2.

Table 2 – Truth table for transfer valve

A	B	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

This activity can be divided into several steps:

• First, qubits Cin and A are compared, and if both are in the |1> state, the Cout bit is reversed.

• If the qubit from A is in state |1>, the qubit of register B is changed.

• Cin and B qubits compare, if both are in |1> state, the output bit Cout is inverted.

This cannot be done by using only one of the X, CX or CCX valves, so we can achieve a circuit like Figure 2 by using several valves:

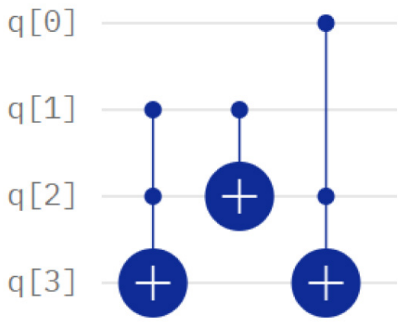

Figure 2 – Visualization of Cout transfer valve using CX and CCX valves

The crosses represent the target qubit, and the colored circles represent the control qubits.

Lines with two colored circles and one cross circle represent the CCX valve, and a line with one colored circle and one cross circle represents the CX valve.

For the summation valve, we need a valve that takes three input qubits: an output transport qubit and one bouquet from each term. The valve performs the same operation as the truth table in Table 3:

Table 3 – Truth table for Sum valve

A	B	Cin	Sum
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

The output is a qubit from the register, so you can save space and operations by rewriting the register and amount instead of storing it in another register. Looking at the truth table, you can break down the operation into smaller steps:

• If the input transport qubit is in the |1>state, the qubit is converted to register B.

• If qubit A is in the |1>position, it is rotated again.

After that, we can access the CX gateways as shown in Figure 3:

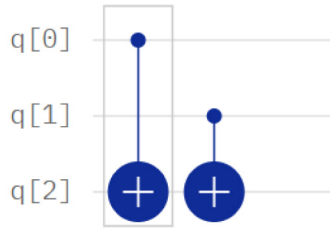

Figure 3 – Visualization of the sum valve using the CX valve

Crosses represent the target bit, and painted circles represent the control bits.

The method described here, although simple and convenient, is not the most effective, since it reproduces the processes used by classical computers in a quantum computer. Another quick way is to add a quantum Fourier transform (QFT)on a quantum computer.

The n-bit reversible summator design is a direct extension of the 3-bit connector. Additional N qubits are used as temporary transfer bits. These qubits return to zero after using them, so they can be used for subsequent calculations. Therefore, although the input and output data can only be stored using 2n qubits, 3N qubits must be used for calculation.

3 Results

Advanced quantum summator. Based on Vedral's work, three Toffoli Gates and two unmanaged gates are used to create the quantum complete summator. The Switch and subtraction built using quantum circuits are shown in Figure 4. According to figures 5 and 6, there is no reduction scheme on the computer. An electronic computer combines addition and subtraction into an addition operation using the addition form. In a quantum computer, addition and subtraction can be combined into an addition operation using the complement form, as in an electronic computer. The complement of a natural

integer is the same as the original integer, and the complement of a negative integer is the inverse of each bit, except for the character bit after 1. A quantum scheme for converting a character integer to a complement type is shown. In Figure 7, if the input source code is a, the output will be a complement of a; if the input has a complement of a, the output will be the source code of A. In Figure 8, the addition and subtraction operations are combined into addition operations with a complement processing unit.

As mentioned above, the first quantum addition algorithms showed their classical counterparts with necessary extensions for reversible computations. Subsequent quantum addition algorithms were based on the use of additional qubit transport, but still followed the classical model. The ideal addition algorithm for a quantum computer may not resemble its classical counterpart, so a quantum adder based on the quantum Fourier transform (QFT) was invented (Figure 4).

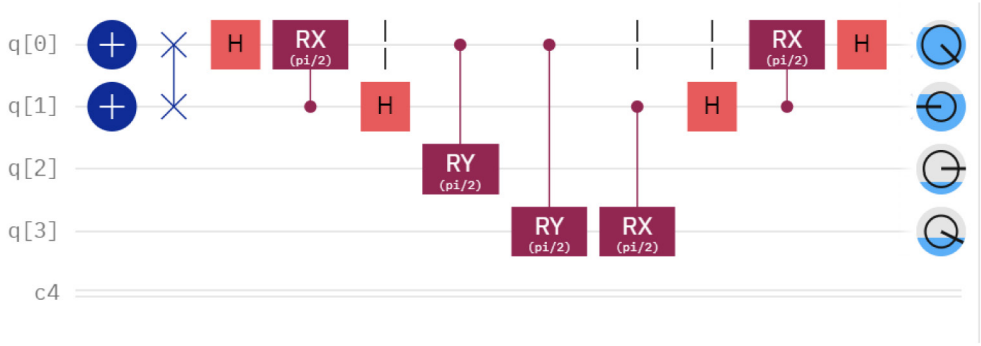

Figure 4 – A quantum switching scheme based on the quantum Fourier transform.

The starting block in Figure 4 represents the QFT. Its mathematical formula is shown in Figure 5.

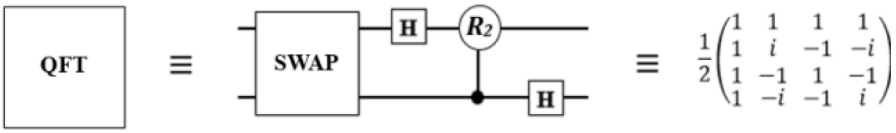

Figure 5 – Fourier quantum transform scheme and its mathematical form.

An example of summing two numbers using QT is considered. Figure 6 shows a quantum circuit for adding two numbers using QFT based on IBM Composer. IBM Quantum Composer is a graphical quantum programming tool that allows you to drag and drop operations to build quantum circuits and run them on real quantum hardware or simulators.

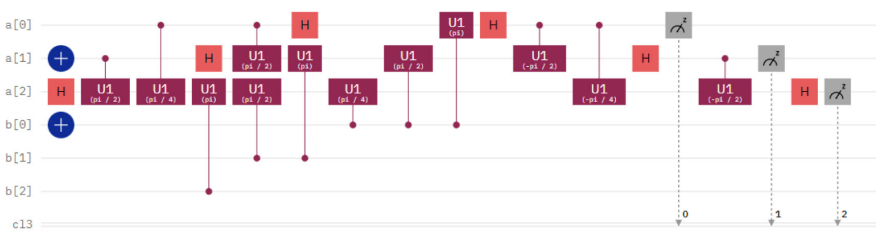

Figure 6 – A quantum scheme for adding two numbers using QFT based on IBM Composer

The real-time visualizations come from a single state vector simulator, which is different from the system specified in the launch parameters, which can be fired multiple times. The simulator creates randomness by generating results based on the original number. A seed value is an initial value fed into an algorithm that generates pseudorandom numbers and simulates quantum randomness.

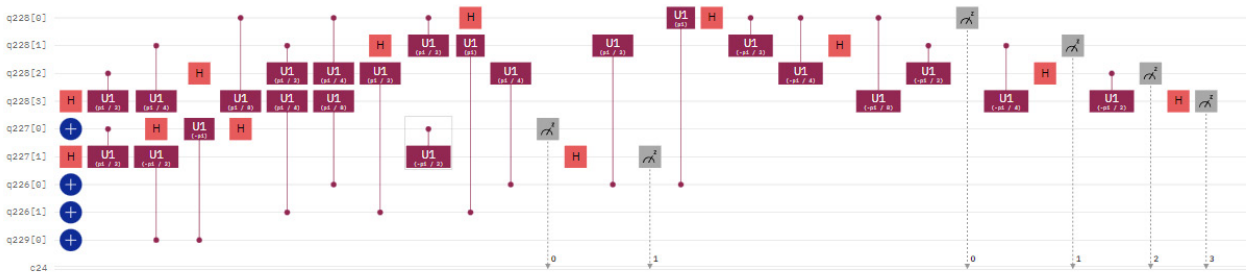

Figure 7 – Visualization of the result and the statevector and the 5-qubit constraint

The statevector method provided as a back option is the simulation method Aer uses to simulate the circuit, but it is still a qasm simulator, which is expected to return measurement counters rather than a state vector.

Addition and multiplication of two numbers were performed in various quantum simulators and the computation time was calculated.

Table 4 – Adding two numbers

Quantum simulators	The qubits	Add time
ibmq_qasm_simulator	32	3.6816170215
simulator_statevector	32	4.4676377218
simulator_extended_stabilizer	63	4.1823140211

Quantum multiplication of a floating point number. A method for multiplying decimals with fixed points on a quantum computer is to use a calibration method for fixed-point processing of decimals. The calibration method is denoted by Qm, where m + n + 1 is the capacity of the computer. For example, -0.52 is used to calibrate, 1.851E takes a fixed point number (in hexadecimal), and 0.68 uses Q0.15 to calibrate to get a fixed point number of 0.Ae14 (in hexadecimal). These two Q0s are the product of two decimal digits (n-1) of the provided source code, a decimal number (2n-1) scaled by Q0

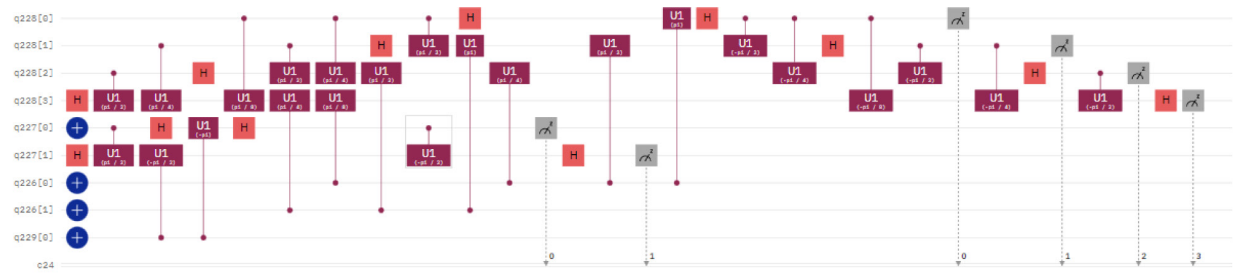

Figure 8 – A quantum scheme for multiplying two numbers using QFT based on IBM Composer

Figure 8 calculates the quantum Fourier transform one cube at a time. Using a single Hadamard valve on the nth qubit of the quantum register, then repeated phase shifts with parameters equal to Pi are applied, which are divided by increasing levels.

The state |F(ψ(reg_a))> |F(ψ(reg_a+reg_b))> transforms the qubit using a reg_b-dependent quantum Fourier transform. After constructing the quantum circuit, the entire algorithm was implemented in Python at the IBM Quantum Lab. Qiskit libraries were used to write the quantum code. Qiskit is an open SDK for working with quantum computers at the level of pulses, circuits and applica tion modules. Qiskit includes a complete set of quantum valves and many ready-made circuits, so you can use Qiskit for research and application development. It can be run and scheduled on real quantum processors and uses Qiskit Runtime to run quantum programs on cloud processors, CPUs and GPUs.

```
Enter the multiplicand. 100
Enter the multiplier. 101

/tmp/ipykernel_59/2096073847.py:17:
er than 3 months after the release
  qc.cu1(pie / float(2**(i + 1)), ı
/tmp/ipykernel_59/2096073847.py:32:
er than 3 months after the release
  qc.cu1(factor*pie / float(2**(i))
/tmp/ipykernel_59/2096073847.py:43:
er than 3 months after the release
  qc.cu1(-1 * pie / float(2**(n - i

{'010100': 2}
Время: 4.1626787185668945
```

Figure 9 – Result of multiplying two numbers using QT based on IBM Quantum Labs

As shown in Figure 9, the numbers 4 and 5 were obtained in binary to multiply two numbers. As a result, it produced the number 010100 within 4 seconds. If we change it to the 16-digit system, it gives the number 20.

4 Conclusion

This article discusses the arithmetic operations implemented in a quantum computer and, accordingly, the arithmetic operations performed in a quantum computer. First, a quantum summation and an additional quantum summation were performed. Then the methods of representation and calculation of fixed point numbers were discussed. After that, a general quantum multiplier and a fixed-point digital divider were created. Based on the advanced blocks of quantum arithmetic, it is possible to perform high-precision decimal multiplication and division. As a result of the work, a new method of calculating the adder in a quantum computer was presented. This technique uses the quantum Fourier transform (QFT) and reduces the number of qubits needed to connect by eliminating the need to temporarily transmit bits. This approach allows adding a classical number to a quantum superposition without encoding a classical number in a quantum register. As a result of the work, the operations of adding and multiplying two numbers were implemented in quantum simulators at the IBM Quantum Lab. Quantum circuits were built in IBM Composer. Based on this work, we plan to develop quantum versions of floating-point digital operations and implement quantum versions of digital systems used in simple electronic computers in the future.

Acknowledgments

The research work was financed by the grant of the Science Committee of the Ministry of Education and science of the Republic of Kazakhstan under the project № AP09260564.

References

1. P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J.Sci.Statist.Comput. 26(5) (1997) 1484–1509.
2. V. Vedral et al., Quantum networks for elementary arithmetic operations, Phys. Rev. A 54(1) (1996) 147–153
3. P. Gossett, Quantum carry-save arithmetic, arXiv:quant-ph/9808061v2 (1998) 1–12
4. T.G. Draper, Addition on a quantum computer, arXiv:quant-ph/0008033v1 (2000) 1–8.
5. S.A. Cuccaro et al., A new quantum ripple-carry addition circuit, arXiv:quant-ph/0410184v1 (2004) 1–9
6. Y. Takahashi and N. Kunihiro, A linear-size quantum circuit for addition with no ancillary qubits, Quantum Info. Comput. 5(6) (2005) 440–448
7. T.G. Draper et al., A logarithmic-depth quantum carry-lookahead adder, Quantum Info. Comput. 6(4) (2006) 351–369
8. Y. Takahashi and N. Kunihiro, A fast quantum circuit for addition with few qubits, Quantum Info. Comput. 8(6) (2008) 636–649.
9. J.J. Alvarez-Sanchez et al., A quantum architecture for multiplying signed integers, J. Phys.: Conf. Ser. 128(1) (2008) 12-13.