DEVELOPMENT OF A QUANTUM ALGORITHM THAT USES QUANTUM PARALLELISM FOR FINDING THE SHORTEST PATH IN A GRAPH

Cesar Borisovich Pronin

Andrey Vladimirovich Ostroukh

Moscow Automobile and Road Construction State Technical University (MADI)., 64, Leningradsky prospect, Moscow, Russian Federation, postal code: 125319

Boris Vasil'evich Pronin

Russian State Agrarian University-Moscow Timiryazev Agricultural Academy (RSAU-MTAA)., Ul. Timiryazevskaya, 49, Moscow, Russian Federation, postal code: 127550

Abstract: The following research highlights the development of a quantum algorithm designed to use quantum parallelism for performing parallel quantum calculations. This concept was used to make a quantum algorithm for finding the shortest path in a graph.

For developing quantum algorithms within 16 qubits, the Quirk quantum circuit emulator was chosen, which has an advanced set of sensors, that allow to visualize the transformations performed in quantum circuits, and a large number of complex quantum gates, which greatly simplifies the construction of new quantum circuits.

Key words: Dijkstra's algorithm, finding the shortest path in a graph, quantum informatics, qubit, superposition, quantum gate, graph, control qubit.

Introduction

Dijkstra's algorithm is classical algorithm for finding the shortest path between nodes in a graph, that was invented by a Dutch scientist Edsger Wybe Dijkstra in 1959. The algorithm is based on the principle of sequentially finding the shortest paths from one (source) node to every other [1-5] and only works for graphs with non-negative paths.

This algorithm is widely used in logistics and can be easily adopted for tasks such as finding the lowest transportation cost or highest average speed limit, by changing the path length values on a given graph [1-5].

Interpretations of Dijkstra's algorithm normally use linear search for finding the shortest paths to the nodes, so the number of linear search iterations equates the number of analyzed paths. Using the unique properties of quantum computer architectures, in theory it is possible to get a drastic decrease in runtime by evading linear search and some conditional operations, thus creating a quantum algorithm for finding the shortest path in a graph would be relevant.

Developing a quantum algorithm for finding the shortest path in a graph

One of the most important steps in creating this algorithm, is creating such a representation for its input data, that changes made to it won't affect the key structure of the quantum circuit, this is vital to assure that the developed algorithm can reach its maximum efficiency. In the algorithm which is developed and shown below all interactions between nodes are constructed based on provided input data, which is normally represented with a graph. Changing the values, that represent lengths of paths in the input graph doesn't affect the given key structure of the quantum circuit [6 … 8].

By applying a Hadamart gate to every qubit in a quantum register, we can use uniform superposition and amplitude properties of states in a quantum register, [9 … 13] for analyzing all possible paths to a certain point in a graph simultaneously.

To visualize the concept of creating such a quantum circuit, let's use a triangular input graph with nodes A, B and C, along with paths between them. It's required to find the minimum path from node A to C (fig. 1).

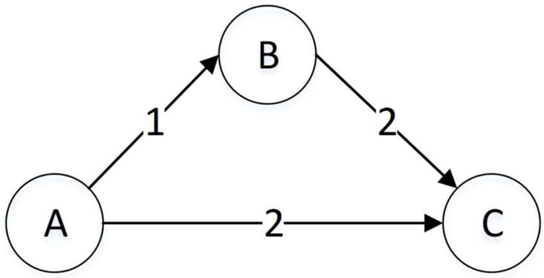

Fig. 1. Graph «ABC»

Based on the given input graph we can estimate, that there can be 2 paths for getting to node C from A:

1) A-B + B-C
2) A-C

It is possible to use classical logic elements to get analyze the lengths of these two paths, by making a classical logic circuit as shown on fig. 2, 3 and 4. This circuit won't give us the possibility of parallel (simultaneous) analysis, but it can be useful for establishing the key relations between the given paths, which can be used to form a similar quantum algorithm.

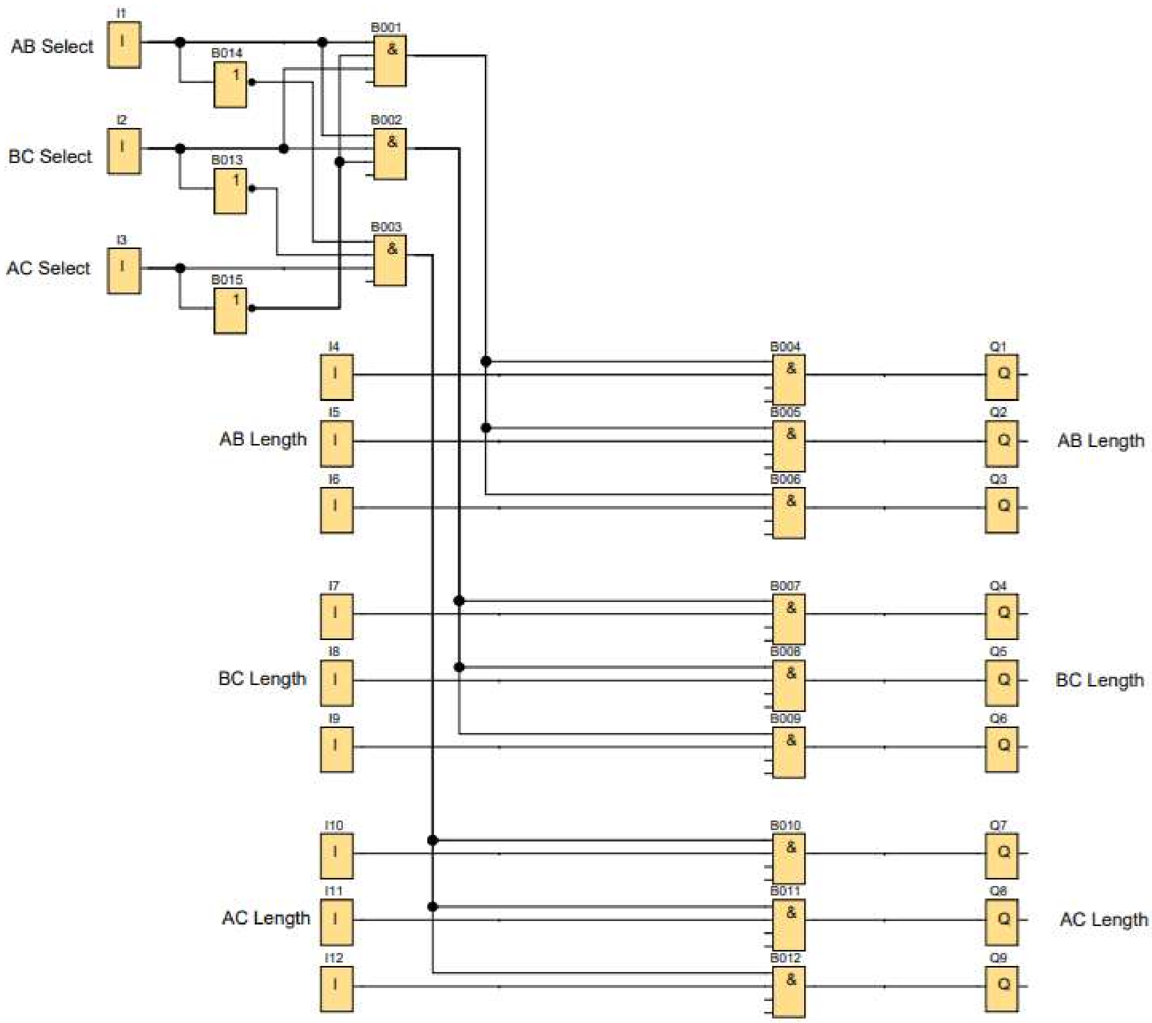

Fig. 2. Classical logic circuit for sequential path selection, modeled in Siemens LOGO!

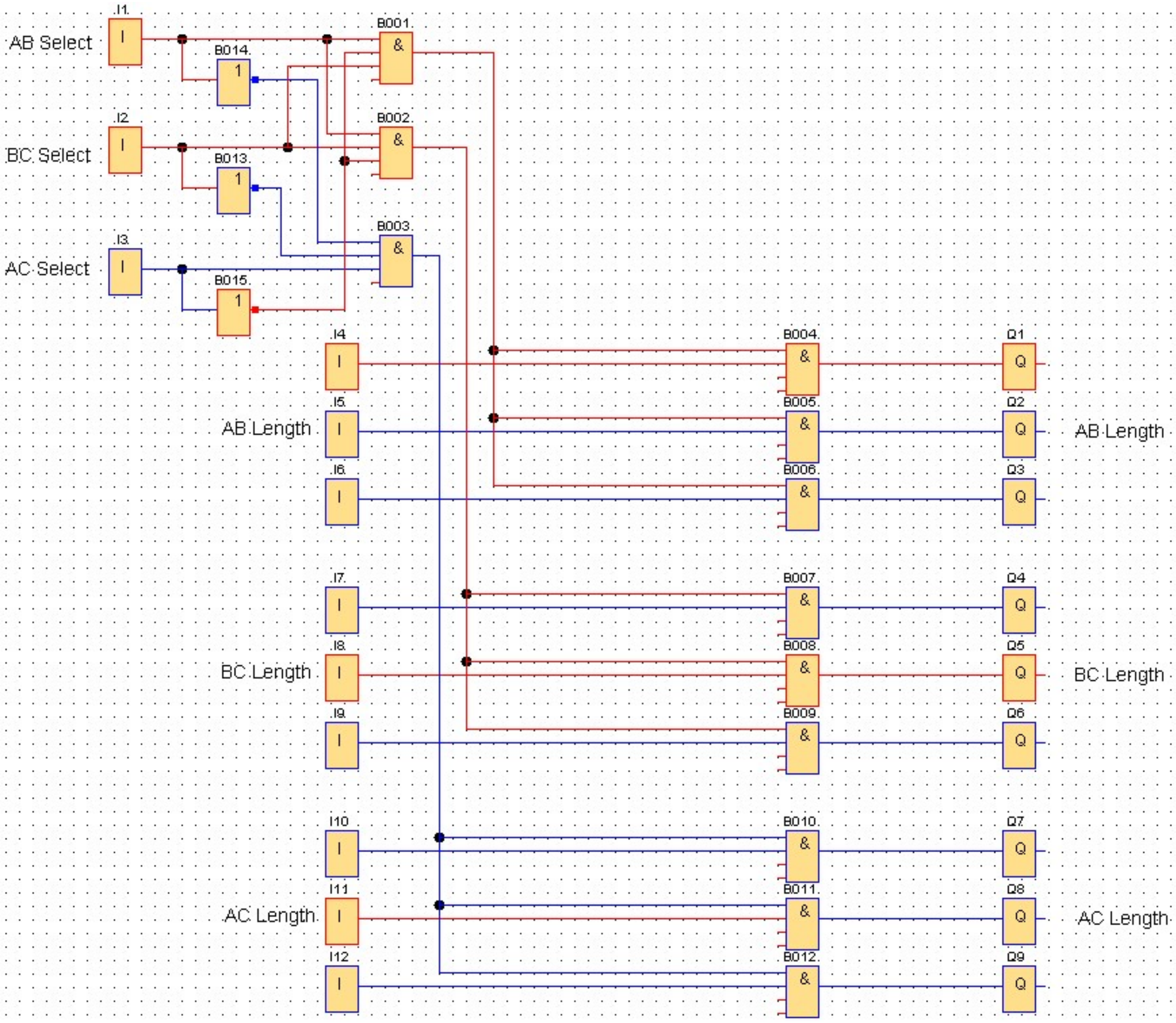

Fig. 3. Classical logic circuit for sequential path selection: path from A to C, mode AB+BC

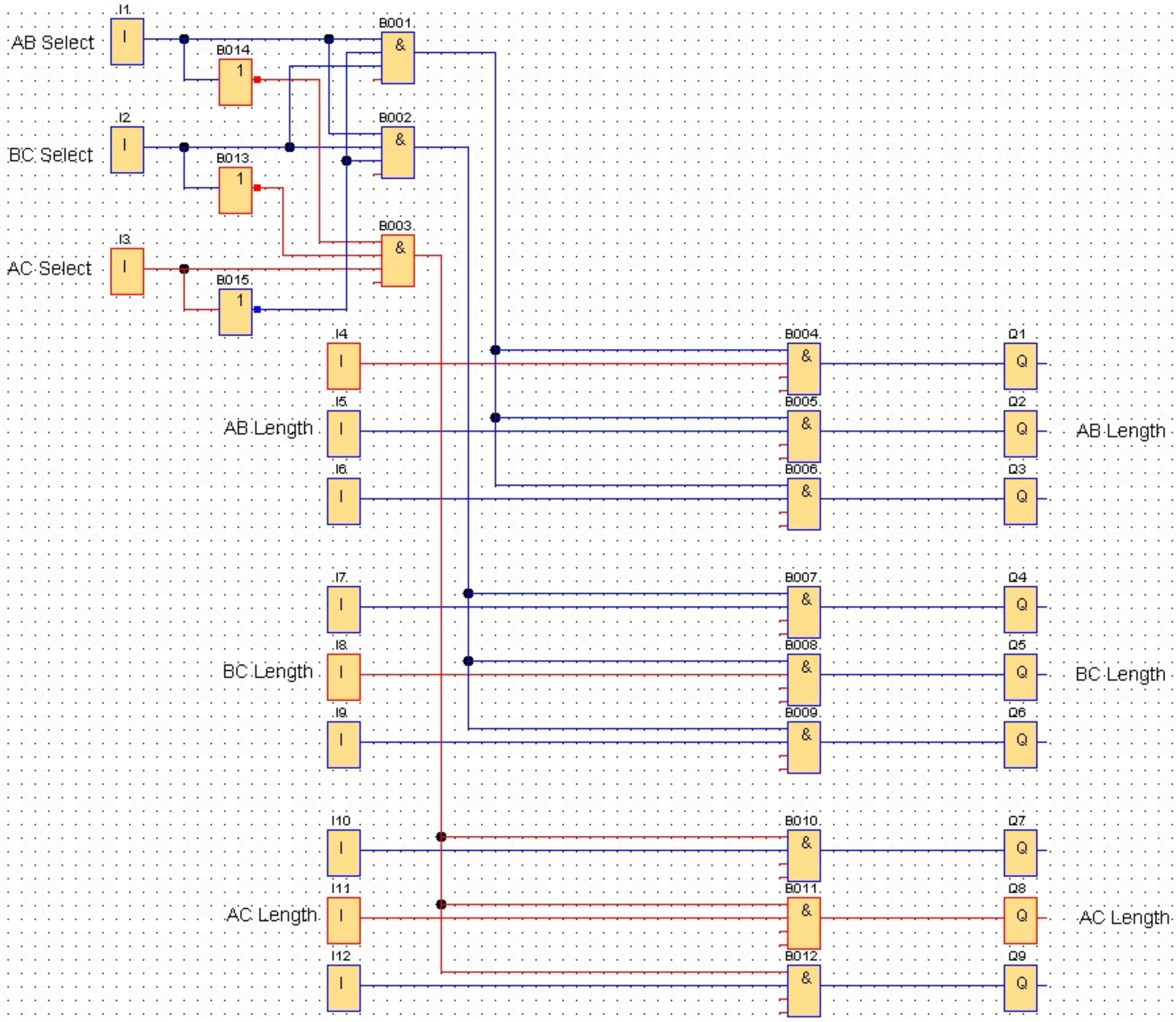

Fig. 4. Classical logic circuit for sequential path selection: path from A to C, mode AC

If any other combination of paths gets selected on inputs *AB Select*, *BC Select* and *AC Select*, this circuit will interrupt all output signals, assigned to path lengths: *AB Length*, *BC Length*, *AC Length*.

Based on the key relations between path selections shown on fig. 2-4, it is possible to make a quantum circuit, shown on fig. 5.

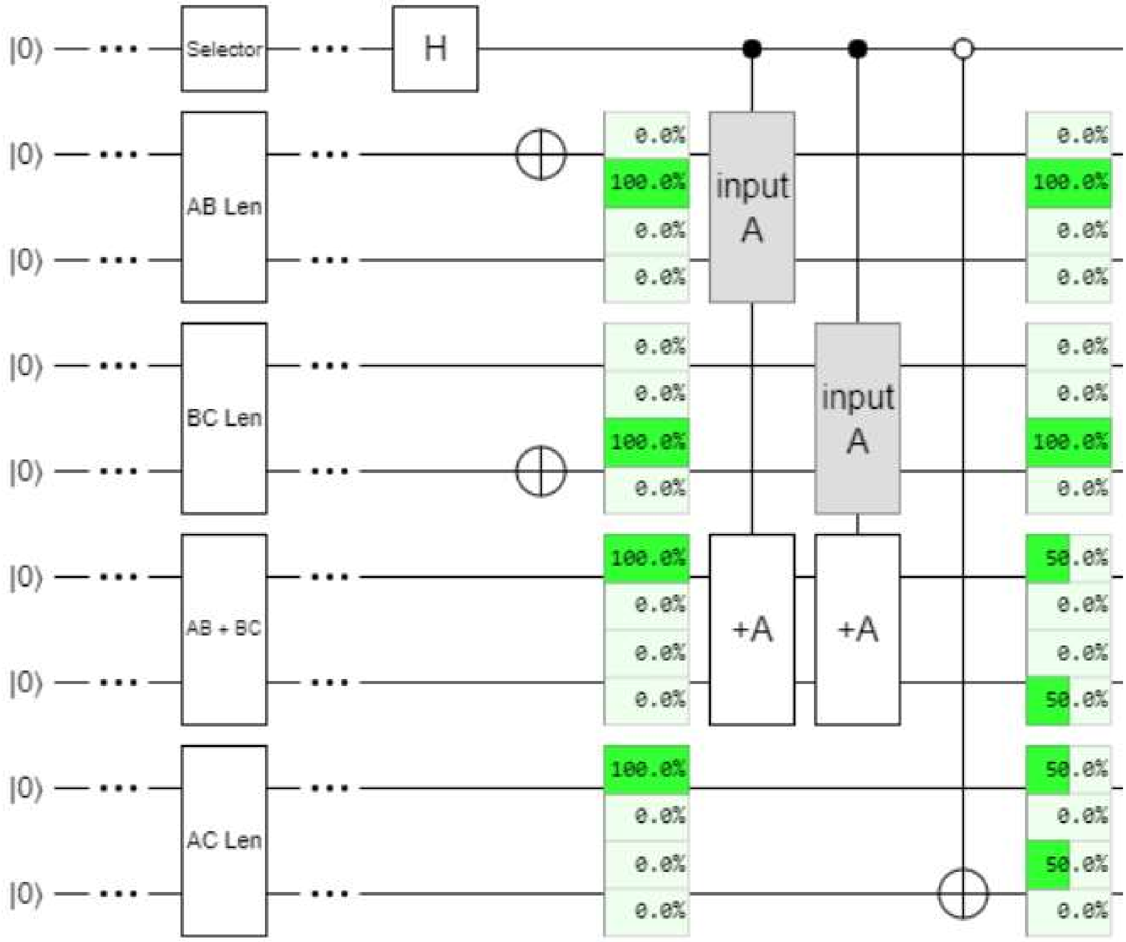

Fig. 5. Quantum circuit, made using the key relations, shown by the input graph

After applying a Hadamart gate to the top qubit in this circuit, the addition of path lengths for AB and BC, is happening at the same time as the selection of path AC, because those tasks are controlled by different control elements, one of which is called "control" (is triggered by the |1> state) and it controls the addition of path lengths for AB and BC, the other element is called "anti-control" (is triggered by the |0> state), it is opposite to "control" and controls the selection of path AC. Creating larger algorithms for more complex graphs, using this principle, can lead to solving the problem of finding the shortest paths in those graphs, while only demanding more qubits for the circuit. Elements named "Input A" and "Input B" receive the values of qubits they are placed on, and act as variables for operators they are connected to.

To compare the final path lengths, it is possible to use Grover's algorithm for removing null states from result qubits: *AC+BC* and *AC Len*. In the circuit on fig. 6, only one Grover iteration is needed, but even with long numbers, there is no real need in increasing the number of those iterations, while the superposition assigned to the resulting states is much higher, than that of other states [6, 7].

Quantum circuit emulator «Quirk» was used for building and debugging the algorithms [13 … 16]. It offers a wide variety of sensors, that allow detailed visualization for transformations, that are performed in quantum circuits. All qubits have $|0>$ value by default [7].

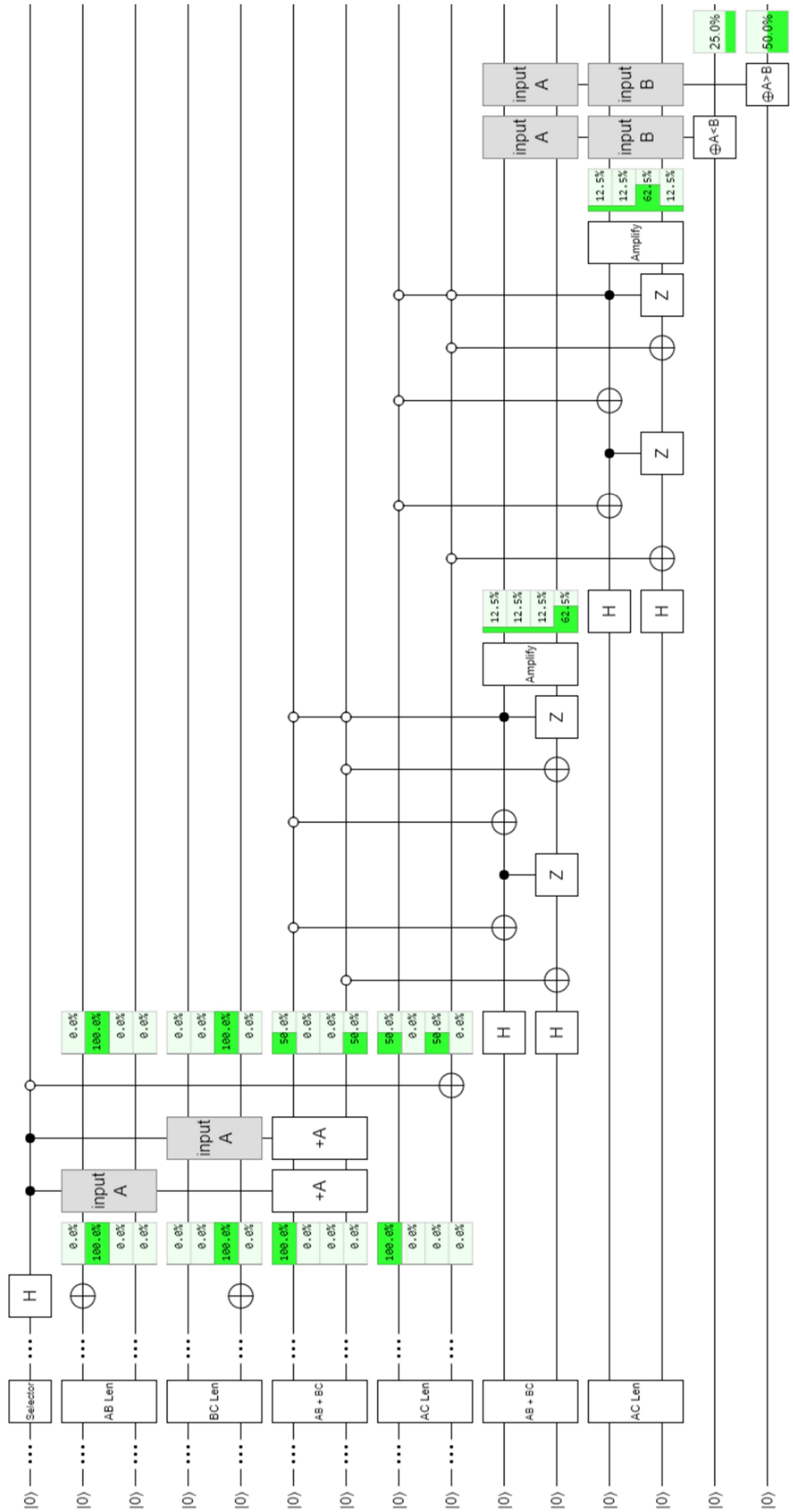

Fig. 6. Quantum circuit, for finding the shortest path in graph "ABC" after dissolving null states

Researching Dijkstra’s algorithm and comparing it to the developed quantum algorithm for finding the shortest path in a graph

For comparing all different paths, using Dijkstra’s algorithm the maximum number of linear search iterations N_D is equal to the number of paths N_{pi} to each of the target nodes N_{tn}.

$$N_D = \sum_{i}^{N_{tn}} N_{\pi i}$$

For the example graph “ABC” on fig. 2.1:

$$N_{D\ for\ ABC} = A \to B + (B \to C + A \to C),$$

Where B is target node 1, and C is target node 2:

$$N_{D\ for\ ABC} = 1 + (1 + 1) = 3$$

To get all possible path lengths from start to finish, using the developed quantum algorithm, only one iteration is needed, due to simultaneous state analysis, which is possible only on quantum computers or their emulators. To receive the shortest path (result), for now, the required number of comparing operators $N_{num.comp.}$ equals:

$$N_{num.comp.} = N_{num.} - 1,$$

where $N_{num.}$ is the quantity of compared numbers.

Since, right now, there are no cycles in quantum emulators, a number of comparing operators, required to provide the end result, was given instead.

Conclusion

The following research has shown, that it is possible to develop a quantum algorithm, that can utilize quantum parallelism for finding the shortest path in a graph. Using the unique properties of quantum computer architectures, we managed to get a drastic decrease in the number of iterations needed to get the result, by evading linear search and some conditional operations, thus creating a quantum algorithm for finding that can find the shortest path in a graph.

For further researches in the field of quantum informatics, creating functions that could be effectively used for finding a minimum or maximum element, from many given numbers, could prove to be quite relevant. This could sufficiently decrease the quantity of qubits used for this task and the number of iterations needed to solve these types of tasks.

References

1. Nikolaev A.B., Ostroukh A.V. (2006) Informative technologies in management and transport logistics – Saint-Louis, MO, USA: Publishing House Science and Innovation Center, 2013 – 254 p. – ISBN 978-0-615-67110-9.
2. Ostroukh A.V., Vorob'yeva A.V., Surkova N.E. Automation of motor transport management. New approach based on intelligent multi-agent systems – Saarbrucken, Germany: LAP LAMBERT Academic Publishing, 2015 – 117 p. – ISBN 978-3-659-47576-4.
3. Ostroukh A.V., Pronin C.B., Researching the possibilities of creating mathematical oracle functions for the Grover's quantum search algorithm // Industrial ACS and controllers. – 2018. – № 9. – P. 3-10. (in Russ.)
4. Pronin C.B., Ostroukh A.V., Quantum logic elements, results and analysis of their influence on quantum circuits // BEST STUDENT ARTICLE 2017: Volume of articles of the XI International Scientific and Practical Competition. In 3 volumes., vol. 1. – Penza: ICSP «Science and Education», p.73 -78, 2017. (in Russ.)
5. Nielsen M, Chang I. Quantum computation and quantum information: Translation from eng. By Vyaliy M.N and Ostrovsky P.M. & Valiev K.A. – Issue: Mir. 2006. 824 p. (in Russ.)
6. Valiev K.A., Kokin A.A. Quantum Computers: Hopes and Reality. 2-nd issue, Izhevsk: SPC RHD, 2002. 320 p. (in Russ.)

7. Baumeister D., Eckert A., Zeilinger A., Physics of quantum information. Quantum cryptography. Quantum teleportation. Quantum computing // Translation from eng. By Kulik S.P. and Shmaonov T.A., Issue: Postmarket - 2002. 376 p. (in Russ.)
8. Preskill J. Quantum Information and Quantum Computing. Vol.1. Izhevsk. RHD. 2008. 464 p. (in Russ.)
9. Valiev K.A. Quantum computers and quantum computing // Successes of Physical Sciences. 2005. Isuue.175. №1. P.3-39. (in Russ.)
10. Ozhigov Yu. I. Quantum computing, Moscow, MGU, 2003. (in Russ.)
11. IBM Quantum Experience // IBM Composer. URL: https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide (date: 10.04.2018)
12. Grover L.K.: A fast quantum mechanical algorithm for database search // Cornell University Library. URL: https://arxiv.org/abs/quant-ph/9605043
13. Quirk – online quantum computer simulator. URL: http://algassert.com/quirk (дата обращения: 21.04.2018)
14. Quantum Teleportation // Wikipedia – the free encyclopedia. URL: https://en.wikipedia.org/wiki/Quantum_teleportation (date: 10.04.2018)
15. Nielsen, Michael A.; Chuang, Isaac. Quantum Computation and Quantum Information: 10th Anniversary Edition. // Cambridge University Press (9 December 2010): p. 97. - 2.3 Application: superdense coding, ISBN 978-1-139-49548-6.
16. Pronin B.V. Atomic physics and the basics of Quantum physics., Moscow, RSAU-MTAA, 2015, 178 p. ISBN 978-5-9675-1207-0

Author details

Andrey Vladimirovich Ostroukh

Russian Federation, full member RAE, Doctor of Technical Sciences, Professor, Department «Automated Control Systems».

State Technical University – MADI, 125319, Russian Federation, Moscow, Leningradsky prospekt, 64. Tel.: +7 (499) 151-64-12. http://www.madi.ru

ostroukh@mail.ru

Cesar Borisovich Pronin

Russian Federation, Postgraduate Student, Department «Automated Control Systems».

State Technical University – MADI, 125319, Russian Federation, Moscow, Leningradsky prospekt, 64. Tel.: +7 (499) 151-64-12. http://www.madi.ru

caesarpr12@gmail.com

Boris Vasil’evich Pronin

Russian Federation, Candidate of Technical Sciences, Professor, Department «Physics».

Russian State Agrarian University-Moscow Timiryazev Agricultural

Academy – RSAU-MTAA, 127550, Russian Federation, Moscow, Ul. Timiryazevskaya, 49. Tel.: +7 (499) 976-04-80. http://www.inter.timacad.ru

borpronin@gmail.com