2004

TR-2004010: Optimal Reversible Quantum Circuit for Multiplication

Anh Quoc Nguyen

Optimal Reversible Quantum Circuit for Multiplication

Anh Quoc Nguyen
(communicated by Michael Anshel)

Quantum Computation can solve many problems that are intractable with classical computers. Development of quantum algorithms requires quantum circuits for elementary arithmetic. Reversible requirement makes quantum circuit design more difficult than classical circuit design. The number of qubits must be kept minimal. In this paper we present an optimal, in term of qubit usage, reversible quantum circuit for multiplication.

Since Shor developed an algorithm for factoring that runs in polynomial time using a quantum computer [9] and Grover developed algorithm for searching unsorted database that has quadratic improvement over optimal classical algorithm [10] scientists have begun intensive development of quantum circuits (a.k.a. quantum networks) for elementary arithmetic operations. V. Vedral, A. Barenco, and A. Ekert developed quantum circuits for addition, modular addition, controlled modular multiplication and modular exponentiation that are sufficient to implement Shor's algorithm, [2]. Many other authors tried to improve those circuits, either by reducing execution time or the number of required qubits.

The Shor's factoring algorithm was experimentally realized by L.Vandersypen, M. Steffen, G. Breyta, C. Yannoni, M. Sherwood, I.L. Chuang,[7] and [8]. In [8] authors expressed the importance of effective quantum circuits with minimal number of actual gates and qubits. With current experimental facility the number of qubits are more crucial. In this paper we are proposing a universal quantum circuit for multiplication that is optimal in term of qubit usage.

1. A Multiplication Circuit

The circuit for multiplication of two n-bits numbers, *a* and *b*, can be constructed from controlled addition circuits. We use the basic addition circuit, or plain adder, from the article [2]. Figure 1a describes the circuit.

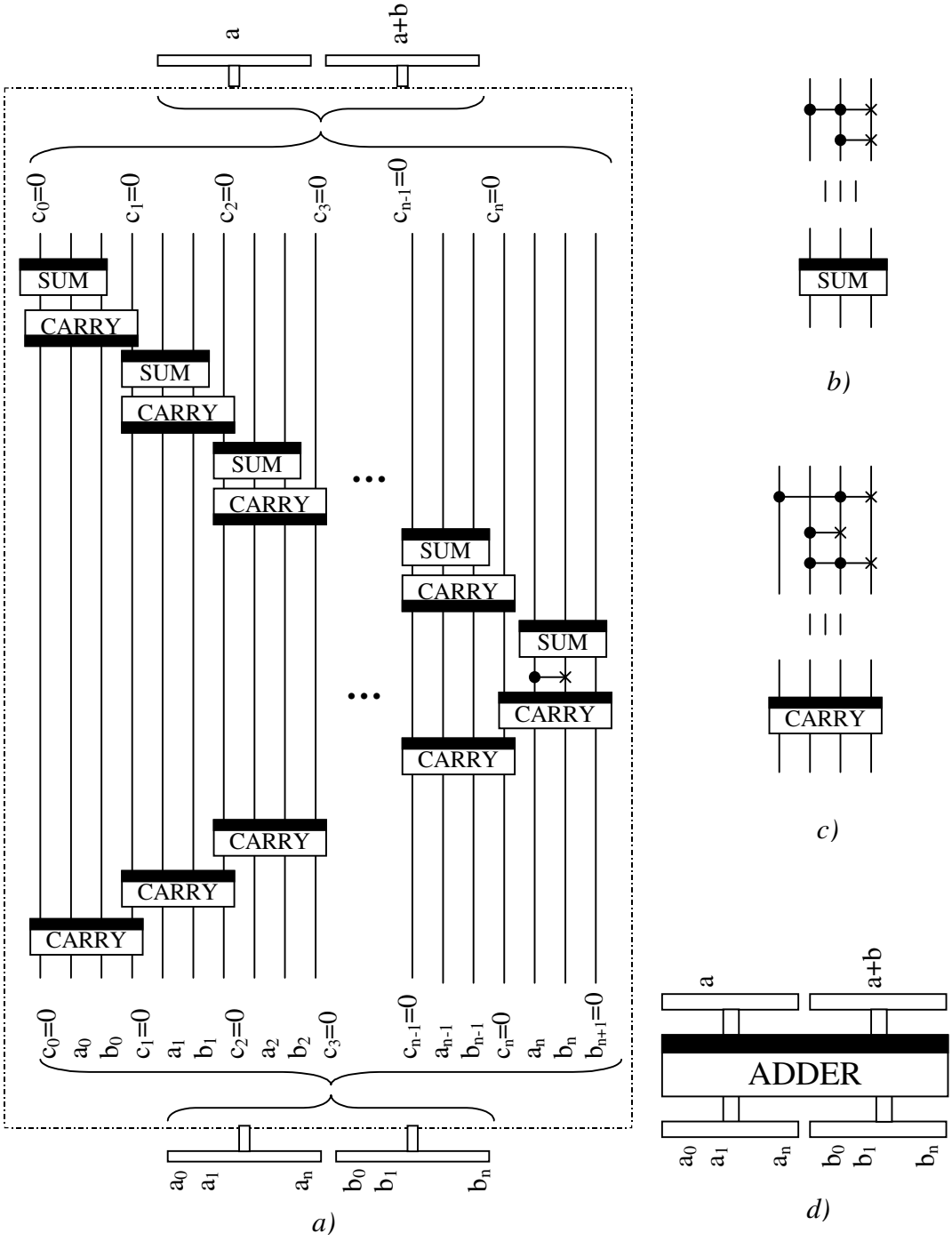

Figure 1. Plain adder as described in article [2]

The registers for values *a* and *b* can be called *source* and *result registers,* respectively. Note that the carry from addition is treated as highest bit of result register. The bit must be zero before the addition. Figure 1d is the block representation of figure 1a for using in more complicated circuits.

By adding a control qubit to every CNOT or TOFFOLI gate we have a controlled addition circuit. If we denote the binary form of *c* as $c_0c_1\ldots c_{n-1}$ then the product *c*d* can be represented as:

$$c*d = c_0(2^0*d) + c_1*(2^1*d) + \ldots + c_{n-1}*(2^{n-1}*d)$$

In classical circuit design each term $c_j*(2^j*d)$ is called partial product, and the product is set to 0 at beginning. At j^{th} step partial product $c_j*(2^j*d)$ is added to the product.

Figure 2 describes a quantum realization of the classical design. This circuit uses 4n qubits to store important information and they are assigned as the following: qubits q_0 to q_{n-1} are used for the multiplicand *c*, qubits q_n to q_{2n-1} are used for the multiplier *d*, qubits q_{2n} to q_{4n-1} are used for the product. Inside the circuit qubits q_0 to q_{n-1} are used as control bits for all adders and qubits q_n to q_{2n-1} are connected to their source register. Result registers from those adders are connected to different portions of the product register. All adders share n-1 working qubits for temporary carries, which are not shown in figure 2, because they don't play important role. The use c_j bit of at j^{th} stage ensures that partial product is added to the product only if c_j is 1. Only connection scheme will be used for our final design. As we mentioned earlier the carry for every adder is treated as highest bit of the result register, therefore in figure 2 we supply n+1 qubits to result register.

In this design we use no extra qubits in the multiplying part to compute the product. All n-1 working qubits come form plain adders, or addition part.

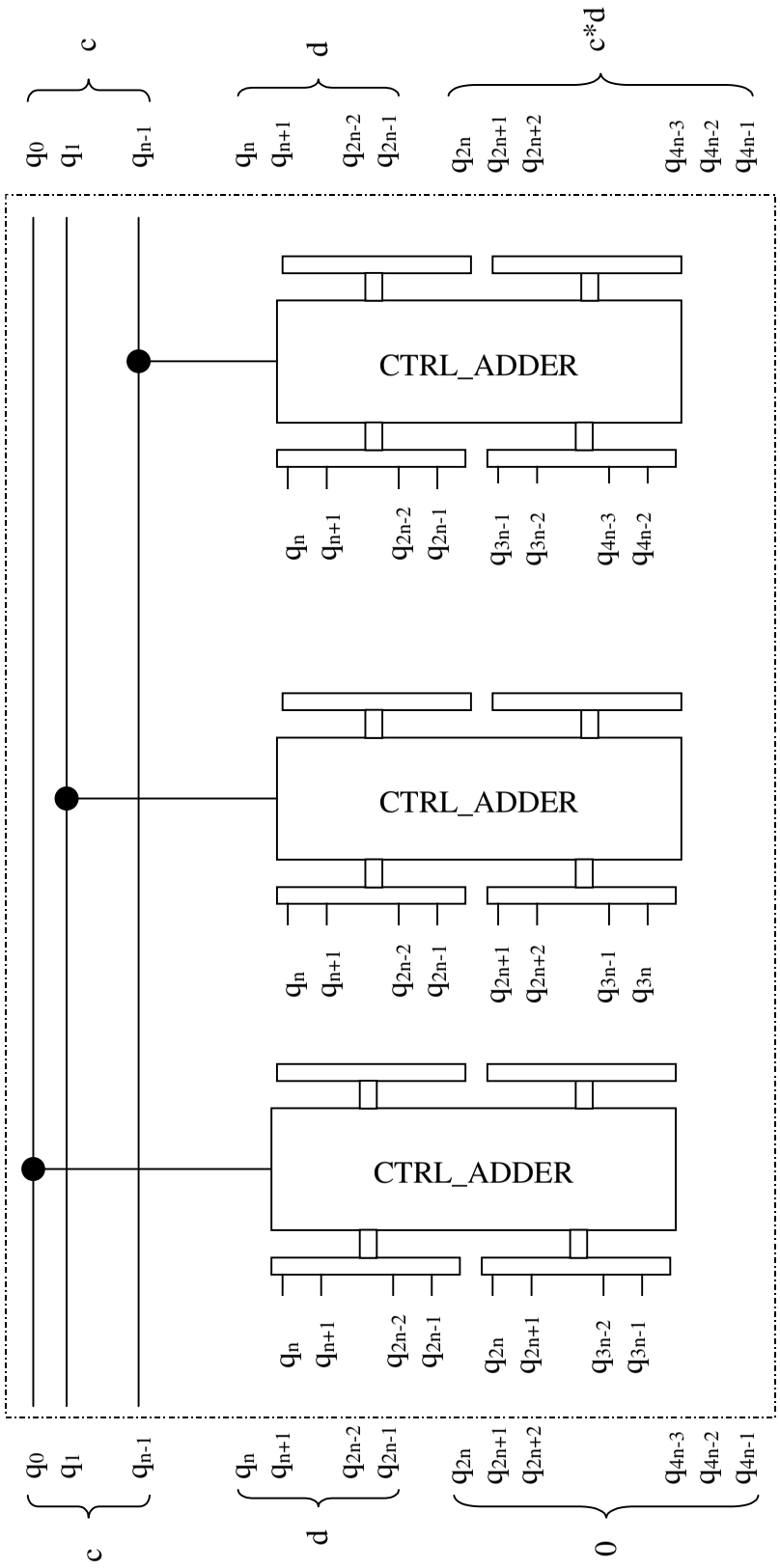

Figure 2. Quantum circuit for multiplication using controlled plain adders. Qubits q_n to q_{2n-1} play role of partial product in every stage.

2. Addition circuit using Quantum Fourier Transform

For reference purpose we describe Quantum Fourier Transform (QFT) circuit in figure 3a, which realizes the design by Coppersmith [11]. The swap circuit, figure 3b, which can be implemented with three CNOT gates, swaps states of two qubits. The controlled rotate gate, figure 3c, is controlled version of one of the basic quantum gate.

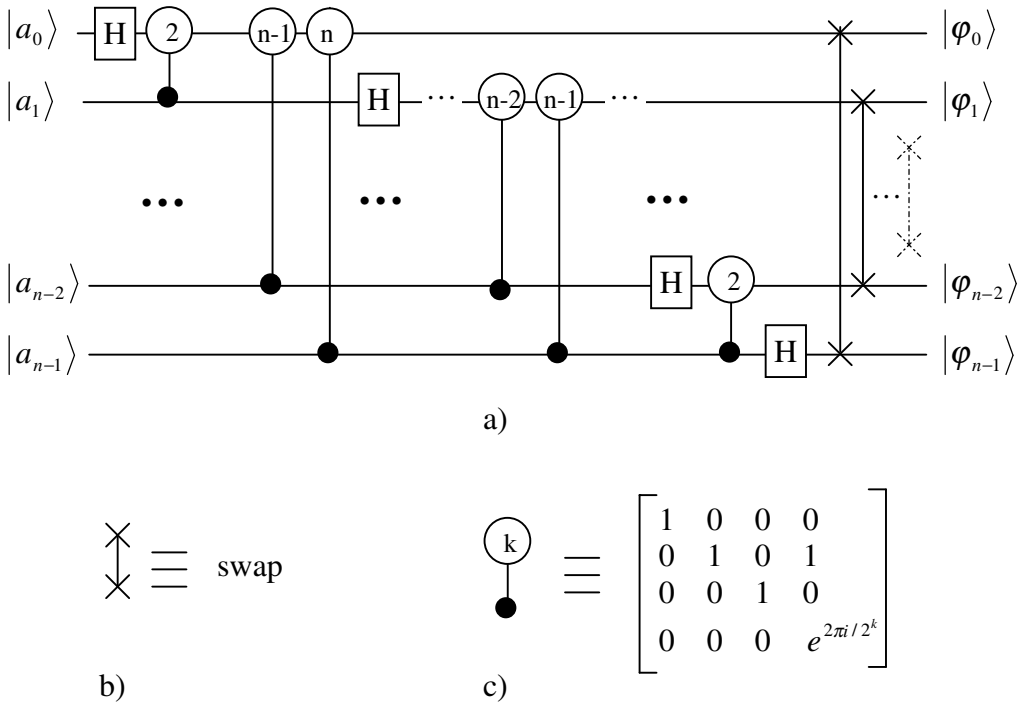

Figure 3. Circuit for Quantum Fourier Transform.

It's easy to see that the plain adder circuit that is described in figure 1 is not optimal in term of qubit usage, because it uses n-1 extra qubits for temporary carries. In article [1] T. Draper developed a circuit for addition that utilizes a property of Quantum Fourier Transform (QFT) to avoid using extra qubits for temporary carries. The original design is represented in figure 4.

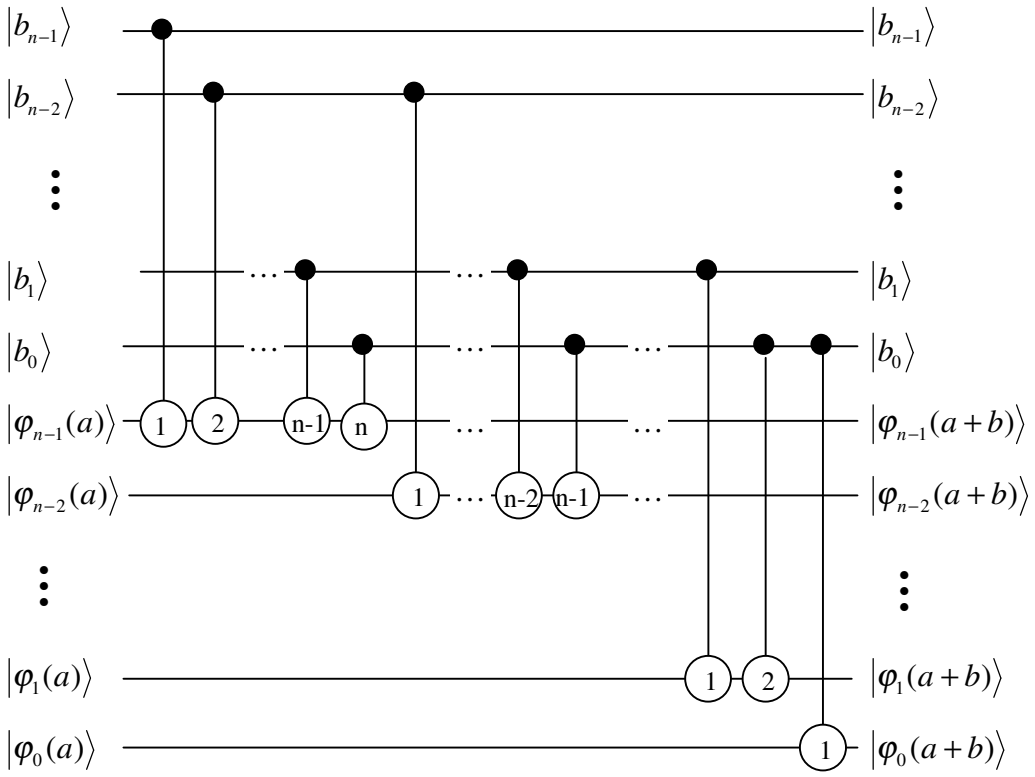

Figure 4. Original Draper circuit for addition in QFT state.
In this design the author uses notation $|\varphi_0(a)\rangle$ as state of n^{th} qubit after QFT

It is interesting to notice that in this design author denotes the state of first qubit in n-qubit register after QFT by $|\varphi_{n-1}(a)\rangle$, and the state of n^{th} qubit in n-qubit register after QFT by $|\varphi_0(a)\rangle$; this creates some confusion in applying the circuit in particular design. We reorganize gates so that we can use the notation $|\varphi_0(a)\rangle$ as state for first qubit after QFT. Figure 5 represents modified version. We call this circuit Fourier Addition (FA). As in the case of plain adder we call two registers in FA circuit by source and result.

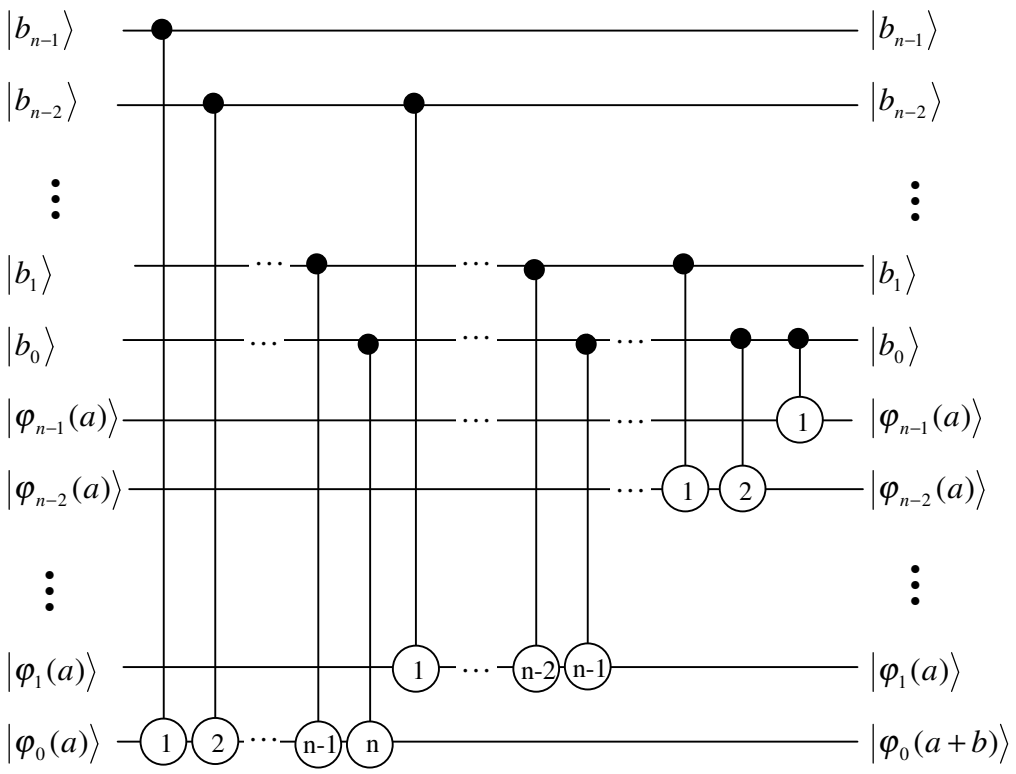

Figure 5. Modified circuit for addition in QFT state, or Fourier Addition (FA)

The FA circuit has two n-qubit registers, first register contains value *a*, second register contains QFT of value *b*, and that register contains QFT of the value *a*+*b* after calculation. To make the real use of a FA circuit we need a QFT circuit to provide input to second register, and an inverse QFT circuit, or QFT^{-1}, to transform the content of second register back to a basis state for measuring. The FA circuit has very interesting feature: *its gates are completely independent, and they can appear in any order*. The FA circuit is reversible, we can un-compute the result register to get back value *b*. The inverse circuit, or subtraction, is not reversing gates' order in addition circuit, since it will produce same result, i.e. addition. Rather, the inverse circuit uses controlled rotation gates with opposite, or negative, phase.

3. Quantum circuit for multiplication using addition in QFT state

Since FA circuit requires exactly 2n qubits for addition of two n-qubit numbers, we can use those adders to built quantum circuit for multiplication to avoid using extra qubits. The original FA circuit can be modified to become controlled FA, by adding a control bit to every controlled rotation gate. But such controlled FA circuits are not ready to substitute controlled plain adders in the multiplication circuit described above. The reason lies in the QFT process. The original FA circuit for addition of two n-qubit numbers uses QFT for n qubits, QFT_n, while the result from multiplication of two n-qubit numbers has 2n qubits. If we want to use the FA circuits for adding partial products to the total product in multiplication circuit we should provide QFT_{2n} of 0 to product register and we expect its final state to be QFT_{2n} of the value c*d.

Regular controlled FA circuit that operates with QFT_{2n} states would require a control qubit along with two registers of size 2n qubits and it adds two 2n-qubit numbers. In multiplication circuit we just want to add n-qubit partial product to appropriate portion of 2n-qubits register; therefore we have to build such circuit based upon FA logic.

Recall that in multiplication circuit described in figure 2 the partial product in each stage is the multiplicand that is shifted to appropriate portion of total product register. Using that logic we build FA circuits for using in multiplication from regular controlled FA circuits for 2n-qubit operands by removing all gates that connect to zero qubits outside the multiplicand for each stage. For example in first stage, stage 0^{th}, partial product corresponds to qubits q_0 to q_{n-1} of product register therefore we remove all gates that connect to qubit q_n, q_{n+1},.., q_{2n-1} in source register for regular controlled FA to get addition part for that stage. We repeat this process for every stage. One of such circuit, circuit for adding partial product to the product at second stage, FOURIER_ADDER_1, is described in figure 6. The number of actual qubits in the product register that are used varies at each stage.

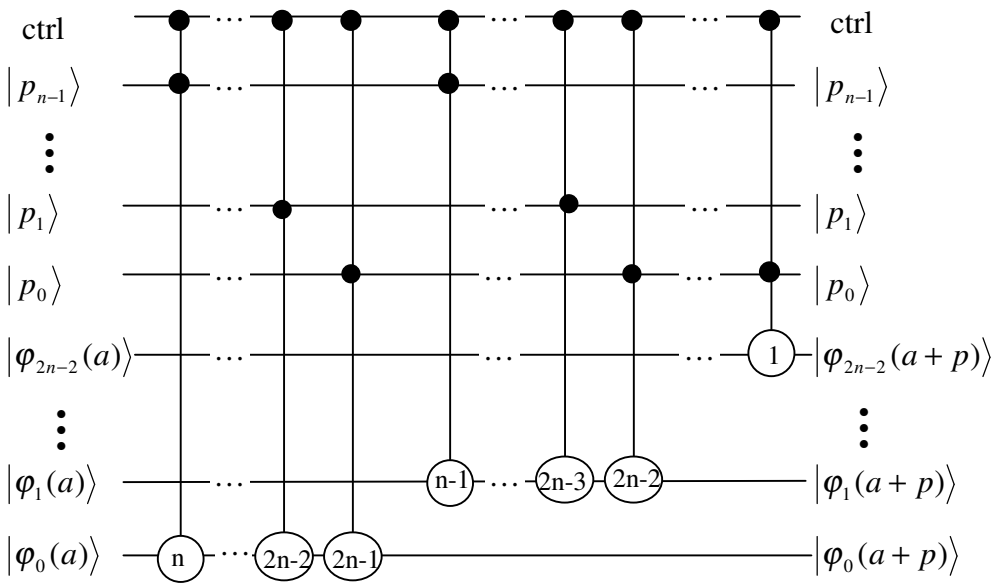

*Figure 6. FA circuit for second stage in multiplication circuit. Value **a** is current content of product register, **p** is partial product at this stage*

The complete circuit for multiplication is presented in figure 7. In the figure FOURIER_ADDER_j is the partial product addition circuit for j^{th} stage. Qubits q_n to q_{2n-1} play role of partial product in every stage as in the multiplication circuit with plain adders. The output of the circuit is QFT_{2n} of the product *cd*. It is important to notice that although the circuit in figure 5 for addition in QFT state doesn't have storage for carry, i.e. its result register after inverse QFT may contain value that is smaller than one of the input numbers, that doesn't affect our design, because in the multiplication circuit additions of partial products to the product register never produce a carry from the highest bit.

It's easy to see that the circuit is universal, with the meaning that it can compute product of any two n-bit numbers. The circuit is reversible and we can un-compute the result to get back zero bits in product register. Like FA circuit, the inverse circuit for multiplication is not a simple reversing of gates in forward circuit. The inverse circuit for multiplication uses inverse FA circuits to archive the result.

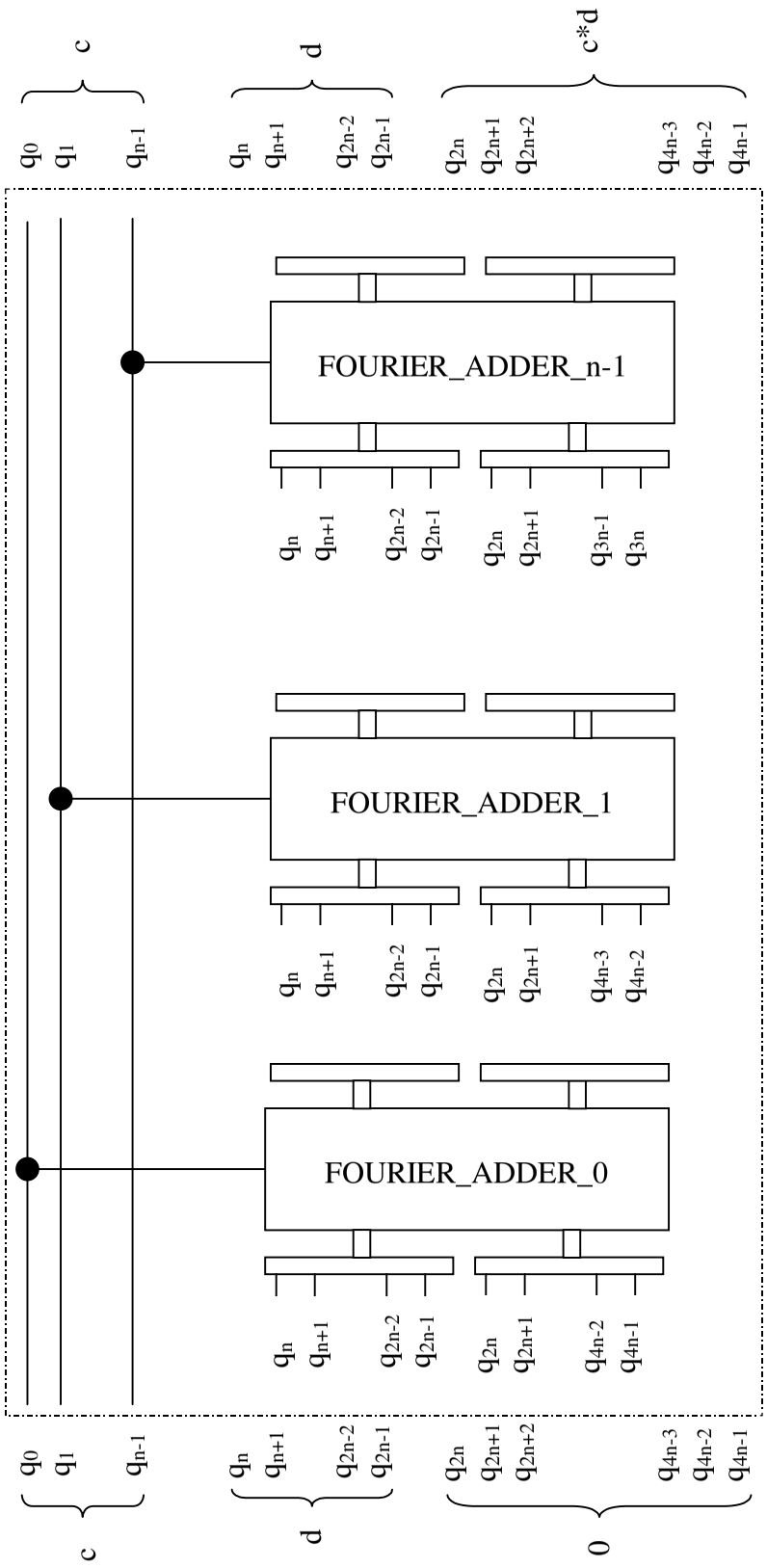

Figure 7. Quantum circuit for multiplication using controlled Fourier Adders.

4. Analysis

According to rules for quantum circuit design that are described in [3] and other articles the circuits must be reversible and have no loop. From that point of view the multiplication circuit for computing the product of two numbers, each of them has n-qubit register, requires at least 4n qubits, because we have to preserve both input values and store the product in a separate register. In general product of two n-bit numbers will occupy 2n bits, therefore any reversible circuit for multiplication requires 4n qubits. Our circuit uses exactly that amount of qubits and therefore is optimal in term of qubit usage.

It is worth to notice that FA circuit requires high operational accuracy, because it uses rotations with small phases; therefore it is more error prone. All circuits with regular adders have higher operational accuracy, since regular adder uses only CNOT and TOFFOLI gates, which have much smaller error rate.

The most complex gates in our design are 3-qubit controlled-controlled rotation gates as shown in figure 6. As [6] and other articles prove that two-qubit gates are universal, the gate can be realized as combination of two-qubits gates. The use of 3-qubit gates in this paper allows us to focus on the design of the circuit and makes diagrams easier to understand.

As we mentioned earlier all gates in FA circuit are independent, they can be re-grouped to allow parallel execution, as the author of [1] points out. Our design preserves this feature. We hope this design will encourage scientist to use multiplication in their design.

Acknowledgement

Author deeply appreciates the guidance and the support from professor Michael Anshel.

Literature:

1. T.G. Draper. Addition on a Quantum Computer. Online Archive quant-ph/00083033.
2. V. Vedral, A. Barenco, A. Ekert. Quantum Networks for Elementary Arithmetic Operations. Phys Rev A, 1995
3. M.A. Nielson and I.L. Chuang. Quantum Computation and Quantum Information, Cambridge Univ. Press, 2000.
4. David P. DiVincenzo. The Physical Implementation of Quantum Computation. Fortschritte der Physik, 48(9-11), 2000, pp.771-783
5. Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John Smolin, Harald Weinfurter. Elementary gates for quantum computation. Phys. Review A, March 1995
6. David P. DiVincenzo. Two-bit gates are universal for quantum computation. Phys. Rev. A, 1994
7. Lieven M.K. Vandersypen, Miathias Steffen, Gregory Breyta, Costantino S. Yannoni, Mark H. Sherwood, Isaac L. Chuang. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature, Vol. 414, Dec. 2001, pp. 883-887
8. Lieven M.K. Vandersypen. Experimental Quantum Computation with Nuclear Spins in liquid solution. Ph.D dissertation, Dept. of Elec. Eng., Stanford University, Stanford, California, USA, July 2001
9. Peter Shor. Algorithm for Quantum Computation: Discrete Logarithm and Factoring. Proceedings, 35th Annual Symposium on Foundation of Computer Science, pp. 124-134.
10. L. K. Grover, in Proceedings of the 28th Annual Symposium on the Theory of Computing (ACM Press, New York, 1996), pp. 212–219.
11. D. Coppersmith. An approximate Fourier transform useful in quantum factoring", IBM Research Report No. RC19642, 1994.