2004

TR-2004007: 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 and Grover developed algorithm for searching unsorted database that has quadratic improvement over classical algorithm scientists began intensive development of quantum circuits (or quantum networks as some authors call) 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. Multiplication circuit without overhead in multiplying part

The circuit for multiplication of two n-bits numbers, c and d, can be constructed from controlled addition circuits. The basic addition circuit, or plain

adder, stores values c and d in registers r_c and r_d, and the result is stored in register r_d is presented in figure 1. The registers r_c and r_d 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 addition.

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 is added to the product.

Figure 2 describes a quantum realization of the classical design. This circuit uses 4n to store important information and they are assigned as the following: qubits $0 \div n-1$ are used for the multiplicand c, qubits $n \div 2n-1$ are used for the multiplier d, qubits $2n \div 4n-1$ are used for the product, or product register. Inside the circuit qubits $0 \div n-1$ are used as control bits for all adders and qubits $n \div 2n-1$ are connected to their source register. Result registers for 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 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.

2. Addition circuit using Quantum Fourier Transform

It's easy to see that the plain adder circuit is not optimal in term of qubit usage, because it use n-1 extra qubits for temporary carries. In [1] Draper developed a circuit for addition that utilizes a property of Quantum Fourier Transform (QFT) to avoid using extra qubits. The original design is represented by figure 3. It is interesting to notice that in the design author denotes the state of first qubit in n-qubit register after QFT by $|\varphi_{n-1}(a)\rangle$, and the state of nth 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 4 represents modified version. We call this circuit Fourier Addition (FA).

The FA circuit has two n-qubit registers, first register contains value *c*, second register contains QFT of value *d*, and that register contains QFT of the value *c*+*d* 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 d. The inverse circuit, or subtraction, is not reversing gates' order in addition circuit, since it will produce same result, i.e. addition. 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 without any 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 plains adders in the multiplication circuit described above. The reason lies in the QFT process. If we want to use the FA circuits for design in previous circuit to compute the product, the product register requires QFT for 2n qubits, QFT_{2n}, because the product register contains QFT_{2n} of $c*d$ after calculation; while FA circuit for addition of two n-qubit numbers uses QFT for n qubits, QFT_n.

In figure 2 the adders in all stages are identical and they work on two n-qubits operands. At j^{th} stage the partial product is added to the product by adding n-bit binary representation of multiplicand to appropriate part of the product if c_j is 1. The size of the product is 2n. Therefore the actual addition is adding two 2n-bit numbers. We can use n-bit adders in this design because one operand, the partial product, has at least n zeros. But we have to modify FA circuits to allow addition multiplicand to a portion of product register that is in QFT state.

The fact that product register has 2n qubits and it is in QFT state forces us to use FA for two 2n-qubits in every stage. The FA circuits for multiplication are gotten from regular FA circuits for 2n-qubit operands by removing all gates that connect to zero qubits outside the multiplicand for each stage. In principle we can retain all those gates, they just add nothing to the product register because one of control bits is zero. Removing those gates makes our circuit more compact and it will run faster. One of such circuit, circuit for second stage, is described in figure 5. By comparing figure 4 and figure 5 we can see that the circuit for first stage has most gates, and the circuit for last stage has least gates.

The complete circuit for multiplication is presented in figure 6. In the figure FA_j is the FA circuit for j^{th} stage. The output of the circuit is QFT_{2n} of the product cd. It is important to notice that although the original circuit in figure 3 for addition in QFT state doesn't preserve carry, our modified controlled FA circuits preserve carry because they use more qubits in result register.

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. Rather, the inverse circuit for multiplication uses inverse FA circuits.

4. Analysis

According to rules for quantum circuit designs that are described in [3] and other articles the circuits must be reversible and have no loop. From that point of view the multiplication of two numbers, each of them is stored in an n-qubit register, requires at least 4n qubits, because we have to preserve both input values and put 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 higher operational accuracy, because it uses rotations with small phases; therefore it is more error prone. The multiplication circuit with regular adder has higher operational accuracy, since it uses only CNOT and TOFFOLI gates, which have much smaller error rate.

Our design uses 3-qubit controlled-controlled rotation gates. As [6] and other articles proved that two-qubit gates are universal, the gate can be realized as combination of two-qubits gates. The use of 3-qubit gates 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 regrouped to allow parallel execution, as the author of [1] pointed 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

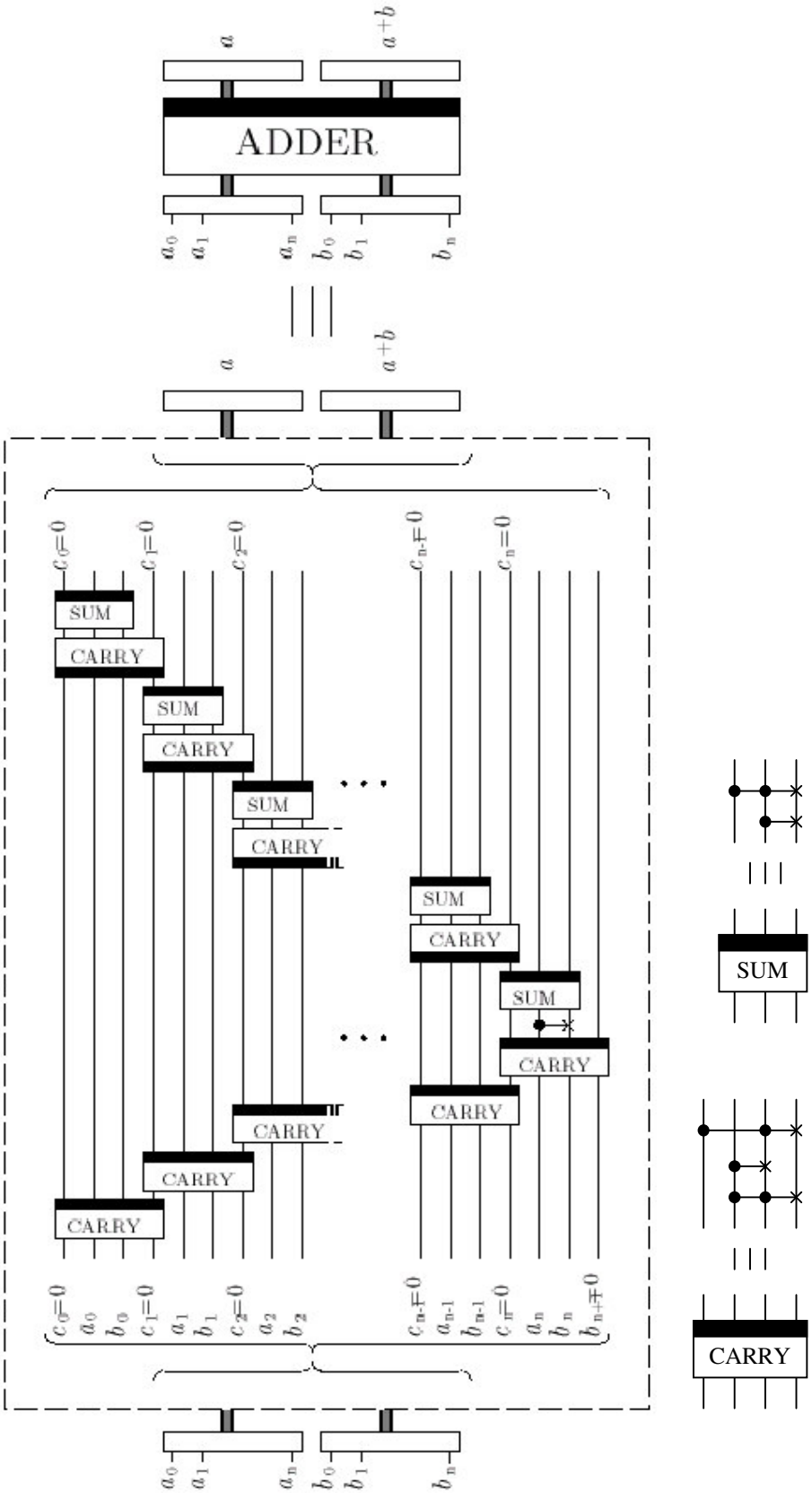

Figure 1. Plain adder as described in article [2]

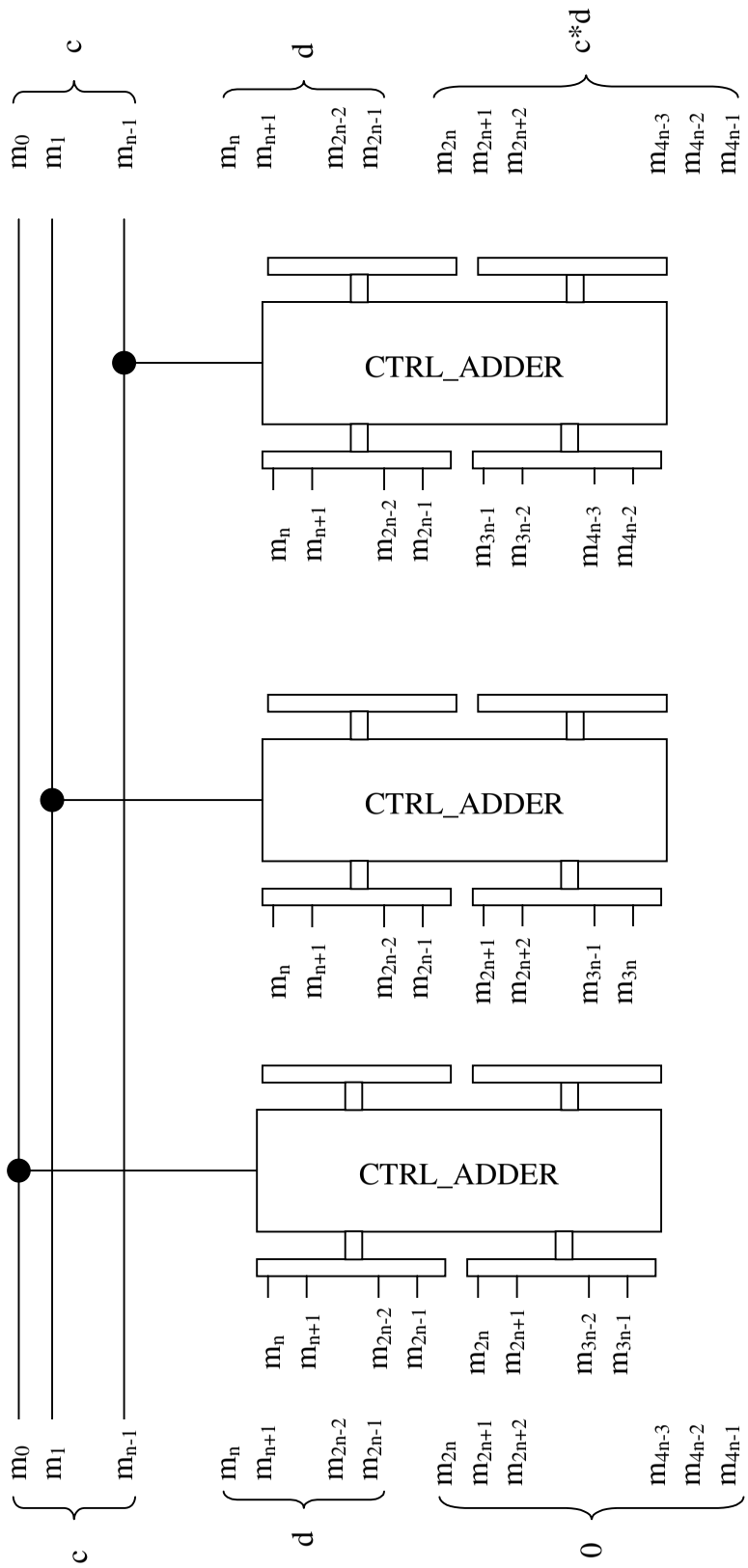

Figure 2. Quantum Circuit for Multiplication using controlled plain adders.

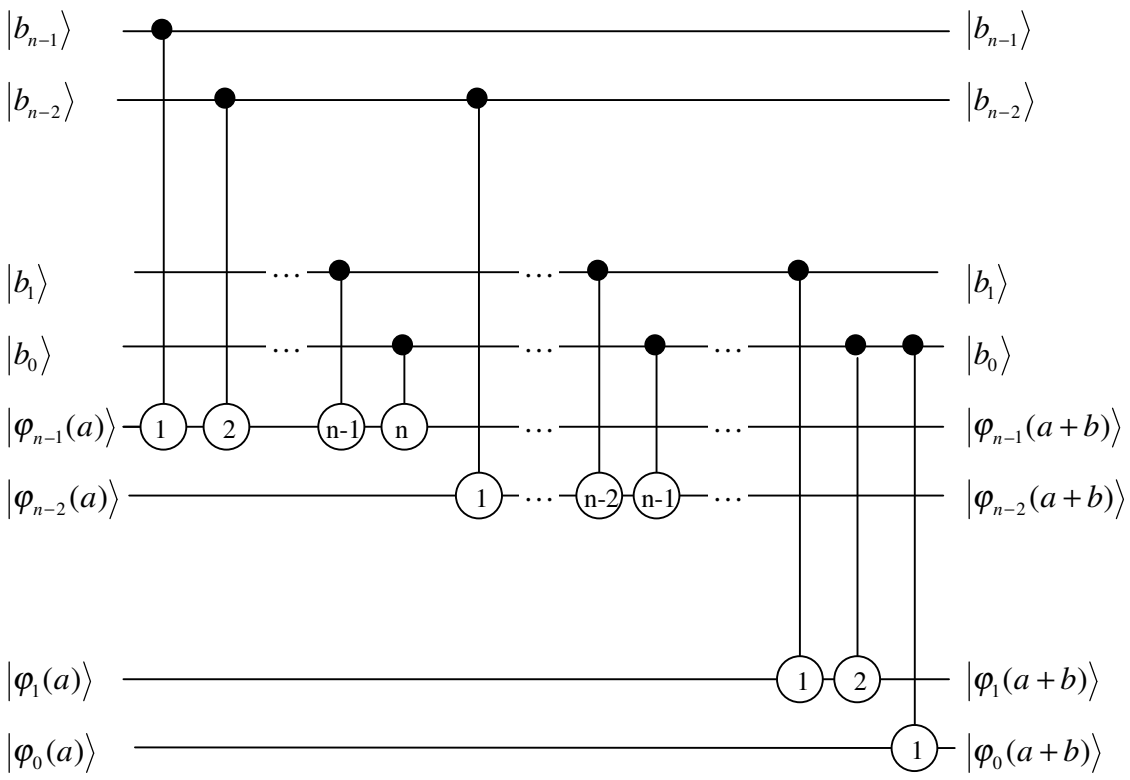

Figure 3. Original Draper circuit for addition in QFT state.

In this design the author uses notation $|\varphi_0(a)\rangle$ as state of n^{th} qubit after QFT

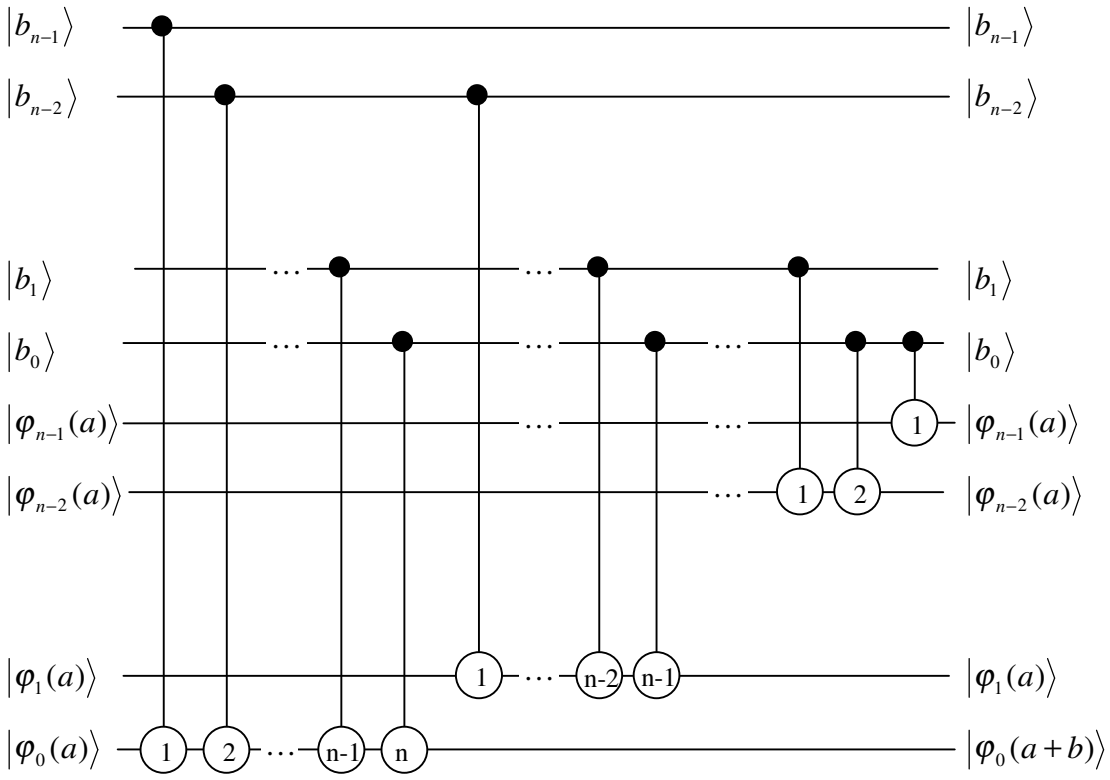

Figure 4. Modified circuit for addition in QFT state, or Fourier Addition (FA)

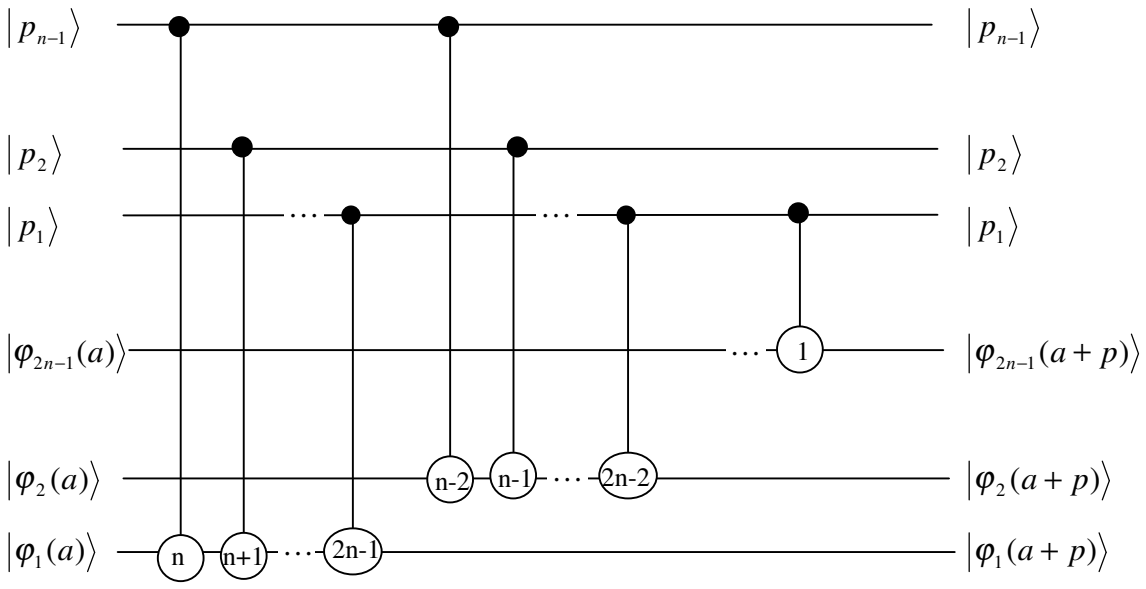

*Figure 5. FA circuit for second stage in multiplication circuit. Value **a** is current content of product register, **p** is partial product at this stage*

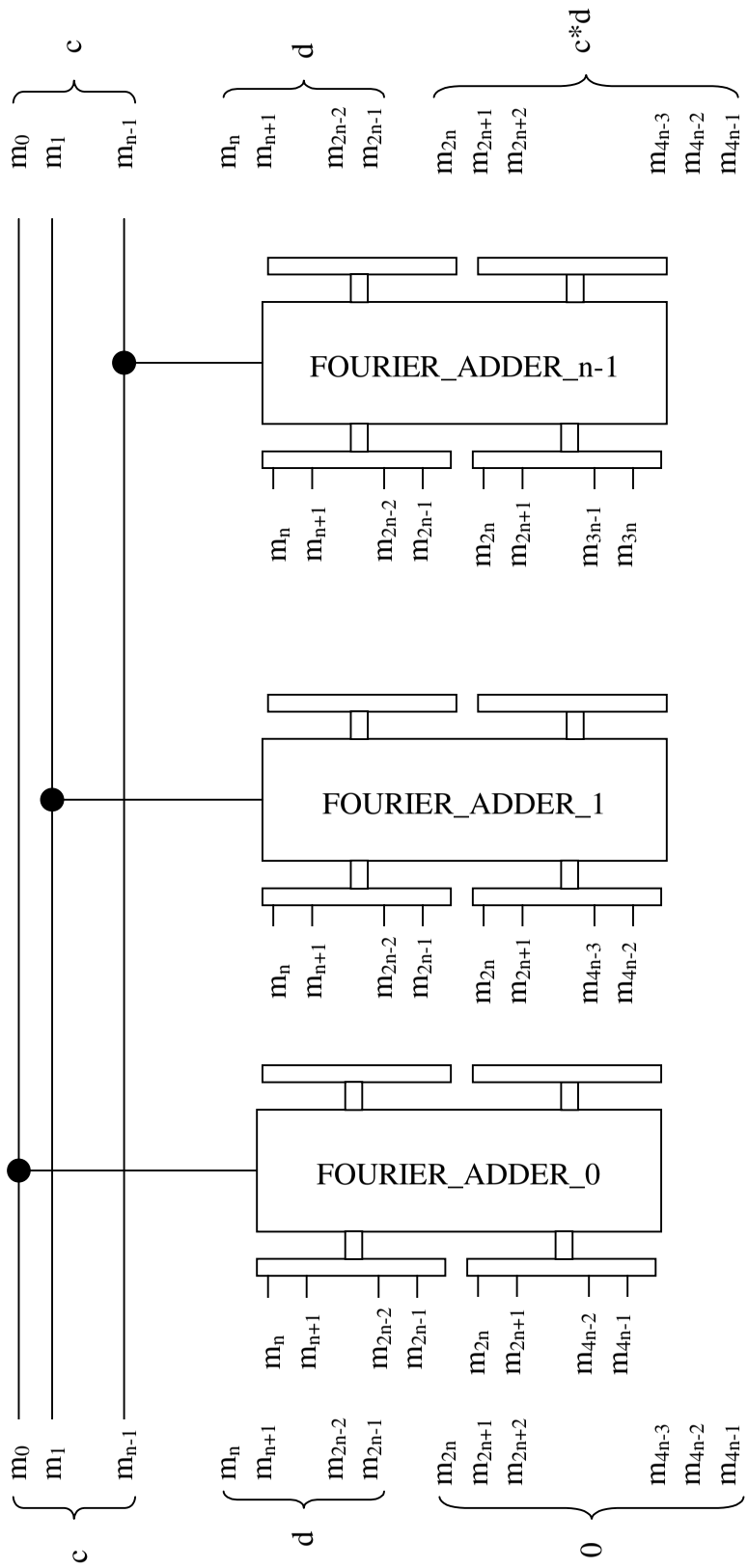

Figure 6. Quantum Circuit for Multiplication using controlled Fourier Adders.