Quantum multidimensional operators with many controls

Nikolay Raychev

Abstract - In this report is proposed an approach for constructing a quantum circuit, containing *O(n^2)* Toffoli gates, CNOT gates and single qubit gates that implements a *C^n(X)* gate for n > 3, without using work qubits. For solving the problem is constructed a $C^n\ NOT$ gate from linear number of Toffoli gates and single qubit gates, without using ancilla bits..

Keywords: Quantum computing, diffraction, simulator, operators, gates

1. INTRODUCTION

The controlled single qubit operators are an essential element in the multidimensional quantum calculations. Because the controlled single qubit operators with many controls are not directly available for synthesis, the application of these operators effectively involves the use of numerous single qubit operators with only one control, this is a fundamental problem for the multidimensional quantum calculations. There are many applications for single qubit operators with many controls, including the application of unitary arithmetical operations [8, 5], synthesizing multidimensional quantum circuits [6, 4] and the operators for the Grover's algorithm in a multidimensional quantum logic [3]. The problem related to the controlled single qubit operators is solved by Barenco et al. [1] for binary quantum logic, using $\Theta(n^2)$ single qubit operators with one control and without ancillary qubits. Muthukrishnan and Stroud [7] developed a quantum array that can be used to control single qubit operations in a radical R> 2 by n ≥ 2 controls using $\Theta(n)$ single qubit operators with one control and $\left\lceil\frac{n-1}{r-2}\right\rceil$ ancilla qubits. The quantum array of Barenco et al. [1] was extended by Brennen, Bullock and O'Leary [2] for multidimensional quantum calculations using $O(n^{\log_2 r+2})$ single qubit operators with one control without using any ancilla qubits where r is a radical and n is the number of controls. However, this quantum array requires taking small radicals of the operation that is being controlled which is not practical since these radicals correspond to rotations by small angles on the Bloch sphere. A new quantum array for implementing hermitian operations in an odd radical R > 2 with n controls will be shown that uses $O(n^{\log_2 r+2})$ single qubit operators with one control and without ancillary qubits, but does not require taking small radicals, as the case is with the existing quantum multidimensional operators that do not use ancilla qubits. Another quantum array will be shown that requires only $O(n^{\log_{r-1} 1+2})$ single qubit operators with one control and can be used to control every single qubit operator in an arbitrary radical r, but requires additionally $\lceil\log_{r-1} n\rceil$ ancillary qubits. These ancillary qubits may be used later, their states are restored to |0>. It must be taken into account that the bases for the logarithmic expressions are $r-1$, this second quantum multidimensional operator requires several operator and ancilla qubits for higher radicals.

This article describes the solution in three parts. The first part reveals the construction of $C^n\ NOT$, when there is an ancillary bit. The goal of part two is to be used this ancillary bit again, but for the construction of increment gates. Finally, part three is for setting up an ancilla bit out of nothing.

2. REVERSIBILITY

The reversible circuits are interesting for several reasons.

First, the reversible circuits bypass the Landauer limit, one of the lower limits on the energy, necessary to perform calculations. In practice, if energy is not needed to be spent for elimination of errors, a reversible computation could be done for free (i.e. without consuming negative entropy; without converting an energy into waste heat (on the other hand, we're more than six orders away

IJSER

from the limit, so this is a rather distant into the future hypothetically useful practicality).

Second, the reversible circuits are a source of torturing problems and unique issues, whose solutions are applicable to quantum computing (where all the operations are reversible). For example, the reversible gates can be classified in classes for equivalency according to "how universal" they are. And, of course, the fact that can not be used NAND, AND, NOR or OR gates, makes the problems upon constructing the circuits even more tangled.

Despite the loss of "standard" universal logical gates, there are still universal gates for reversible computations. Unfortunately these gates always come with stipulations on their universality. The Fredkin gate (controlled swap) can be used for universal calculations, but preserves the number of the ON bits. As a result is required a linear number of ancilla bits, in order to make something useful only with the Fredkin gates.

In this article is used the much more flexible Toffoli gate, but it also has some conditionalities. In fact there is no reversible gate, which can build all reversible operations.

Permutations and parity

Each reversible operation must pair inputs of different outputs, such that each output comes from exactly one input. More specifically, the operation must be equivalent to a permutation of the space of the states.

The permutations have a parity. If an odd number of swaps is necessary, in order to perform a permutation, then it has an odd parity. On the other hand, an even number of swaps means that the permutation has an even parity. When permutations are chained together (i.e. one reversible operation is applied followed by another), the parity of the resulting overall net permutation is the sum of the parts of the two chained permutations. When two even permutations or two odd permutations are chained, the result is an even permutation. When circuiting one even and one odd permutation (regardless of the order) the result is an odd permutation.

The parity is useful very often, when it must be proven, that something is impossible. For example, it is an integral part of the proof that can be created configurations of a sliding puzzle. It should be noted that the rules for adding a parity imply that an odd permutation can not be created by circuiting even permutations. This limitation will be used to show that some reversible operations can not be constructed out of smaller such.

Let's examine a *NOT* gate with many controls, enough to affect all lines of a circuit. For example, let's have a 12-bit circuit and the last bit must be toggled, when the first nine are ON (i.e. we have $C^{11}\,NOT$). The permutation, corresponding to that $C^{11}\,NOT$ reverses the state 111111111110 with the state 111111111111, but leaves all other states unaffected. Since only one swap is performed and this is an odd number, $C^{11}\,NOT$ is an operation with odd parity (when it is applied to a 12-bit circuit).

Let's now consider any operation that does not affect all lines of a circuit. There must be some bit b, from which the operation does not depend on or affect. When examining the swaps, carried out by this operation, each swap at b = 0 must have an equivalent such at b = 1. In other words, the presence of an unaffected bit doubles the number of the swaps (since the swap must be done once at b = 0 and once at b = 1). Therefore the number of the swaps of the space of the states, carried out by this operation, must be even, so the operation has to have an even parity.

Because a controlled NOT that affects each line, has an odd parity, and each operation, affecting on several lines, has an even parity, and the chaining of even operations can not create an odd operation, then it is impossible to decompose an operation with a controlled NOT, which affects all the lines, in smaller operations.

The barrier of the parity sounds like a huge problem, but in fact is more about lack of working space rather than anything else. As soon as the controlled NOT stops to affect every single bit, the argument stops working. Although the chained permutations retain the total number of swaps when working on Module 2, this is not true for other modules. When there is even one indifferent bit, additionally the limitations of the parity can be bypassed.

Ancillary bits

IJSER

The ancilla bits are extra bits, not included in the performed logical operation, which give to the constructions of the circuit a "space to move". In addition to making the constructions possible in the first place, the ancilla bits allow for simpler and more effective constructions.

The ancilla bits are several different types. Sometimes their initial value is known, and sometimes it is not. In some cases, their initial value must be restored, and in others - not. Usually for the ancilla bits it is assumed that they start in OFF or ON state and the constructions must return them to that state before completion (so later the gates can reuse the ancillary bit). In this article are described four types of ancilla bits.

To avoid ambiguity and confusion, let's name and define the four types of ancilla bits:

- **Recordable bits**: They initially are OFF, but they do not have restrictions for the state afterwards. Mainly the recordable bits are (a small part of) negative entropy, which can be consumed to perform some irreversible computations.

- **Zeroed bits** : Initially they are OFF and must be ensured that they will remain OFF upon completion. The zeroed bits are usually used just as the recordable bits, with the exception that the effects are cleaned up before continuing. Circuits, using zeroed bits, are with a constant coefficient larger than circuits, using recordable bits, because of the uncomputation tax.

- **Garbage bits**: They can be in any state initially and it is possible to add more and more garbage in the state (the initial value should not be restored). This type of bits are more complex to use in comparison with the recordable and zeroed bits, because a logic is necessary around the detection of toggling. The detection of toggling usually includes repeating of self-removable operation twice, conditioned on potentially toggled garbage bit. When no toggling occurs the operation either doesn't happen or happens twice. Circuits, using garbage bits, are with a constant coefficient larger than circuits, using recordable bits, because of the tax for toggle-detection.

- **Borrowed bits**: They can be in any state initially and must be restored to this state afterwards. The borrowed bits have the disadvantages of the zeroed and garbage bits and pay their taxes. However the borrowed bits are much more easy-to-find, because they can be borrowed from themselves. Each operation, which does not affect the entire circuit, can use unaffected lines as real bits for borrowing.

With these four types of ancilla bits can be constructed some large controlled NOT-s. It was already discussed, why the case without ancilla bits is impossible, but what will happen in case of a single ancillary bit?

Single ancillary bit

If there is a circuit with n+2 lines with n control lines, one target line and a secondary line, the goal is to be broken up $C^n\,NOT$ into smaller operations. The idea is to be taken:

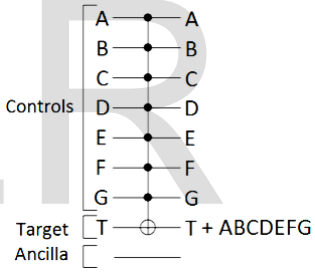

Figure 1

and to be broken up into controlled NOT-s with less controls. It is not necessary to reach up to 2 controls, but simply to reduce the maximum number of controls per operation.

The simplest case is when the single ancillary bit is recordable. The ancillary bit can be toggled to ON when half of the controls are ON, and then to be used a single control on the ancilla bit, which to play the role of this half of the controls:

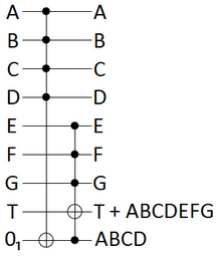

Figure 2

IJSER

If the ancillary bit is zeroed instead of recordable, the effects on it must be cleared before finishing. The effects are very simple and can be removed by repeating the previous actions in the first place:

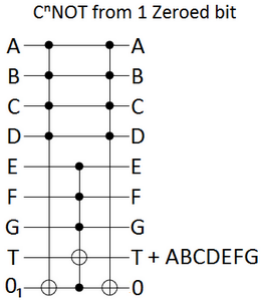

Figure 3

When the ancillary bit is garbage, shall be carried out detection of toggling. Conditionally T is toggled on both sides of the possible toggling of the ancillary bit, so that the T-toggling is canceled by itself, unless the ancillary bit was toggled:

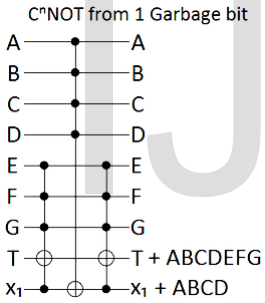

Figure 4

Finally, the case with the borrowed bit is just a combination of the tricks with garbage and zeroed bit:

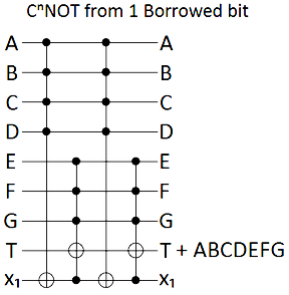

Figure 5

Each of the above constructions uses one ancillary bit, in order for $C^n\ NOT$ to become constant number of $C^{\frac{n}{2}}\ NOT$ (more specifically are used $C^{\frac{n}{2}}\ NOT$-s and $C^{\frac{n+1}{2}}\ NOT$-s).

This construction can be applied iteratively, turning $C^n\ NOT$-s into $C^{\frac{n}{2}}\ NOT$-s into $C^{\frac{n}{4}}\ NOT$-s and so on p times, until reaching the basic case of the Toffoli gates when $\frac{n}{2^p} \approx 2$. Unfortunately this may require more than linear number of Toffoli gates.

For a bit, which can be borrowed, the differential equation as a result of iteration to the base case would be $T(n) = 4T(\frac{n}{2})$, which means an $O(n^2)$ function.

If there is a recordable bit, the differential equation can be expected to be $T(n) = 2T(\frac{n}{2}) \in O(n)$, but in reality, the bit can be recorded only once, so it must be toggled to garbage bit after the first iteration.

For zeroed and garbage bits the things become more interesting. At first glance it seems that their differential equation must be just $T(n) = 3T(\frac{n}{2}) \in O(n^{\log_3 2}) \approx O(n^{1.585})$ However an even separation should not be used between the sizes of the sub-operations. Because only one of the sub-operations happens two times, can be gained efficiency by giving proportionately less controls. Therefore, instead of this, must be analyzed the differential equation $T(n) = 2T(c_n \cdot n) + T\big((1 - c_n) \cdot n\big)$, where c_n is a parameter for optimization, which determines the asymmetry of the separation. This is a very interesting differential equation, which unfortunately is difficult to solve.

We have a recursive relation, which looks like this:
$T(n) = 2T(c_n \cdot n) + T\big((1 - c_n) \cdot n\big) + O(1)$
The basic case is $T(a) = 1$, when $a \leq 1$. The task is to find the optimum value of $c \in (0,1)$, in order to minimize the speed of the asymptotic increase.

If $c = 1/2$ is used after the recursion it can be simplified to $3T(\frac{n}{2}) + O(1)$ and the increase can be described with the following relationship $O(n^{\log_3 2}) \approx O(n^{1.585})$.

The alternation of c_1 and $c_0 = \frac{1}{3}, \frac{1}{2}$ achieves resultant complexity $O\big(n^{\log_5 2}\big) \approx O(n^{1.165})$. In the general case, if it is cycled through $\frac{1}{p}, \frac{1}{p-1} \ldots \frac{1}{3}, \frac{1}{2}$ is obtained $O\big(n^{\log_p 2p-1}\big) \in O\big(n^{\log_p 2p}\big) = O\big(n^{1+1/\log_p 2}\big) \in O(n^{1+\epsilon})$.

IJSER

This suggests that, with the help of a single parameter c we can achieve approximate linear approximation, however we just can not set $\mathrm{p} = \Gamma^{-1}(n)$. And perhaps the optimum value of c reaches $0(n\ \log(n))$, which is better than $0(n^{1+\epsilon})$. Since the problems are caused by the multiplication, transforming the problem into $T(n) = T(\mathrm{p}) = 2T(\mathrm{p} - \log(\mathrm{c})) + T(\mathrm{p} - \log(1 - \mathrm{c})) + 2\mathrm{p}$ could make things more easy.

However, it is possible to circumvent the problem with the complicated differential equation. It must be noted that the constructions with a single bit always create sub-operations, which are quite smaller. It is ensured that there are at least $\left\lceil \frac{n}{2} \right\rceil$ unaffected bits and the operations will have size at most $C^{\left\lceil \frac{n+1}{2} \right\rceil}\ NOT$

And all these unaffected bits may be borrowed!

n - 2 ancilla bits

If there is a circuit with 2n - 1 lines, with n control lines, n - 2 ancillary lines and one target line, the goal is to break $C^n\ NOT$ to a linear number of Toffoli gates. The ancillary bits will be dispersed throughout the circuit instead of being placed at the bottom, to make the constructions look more simple:

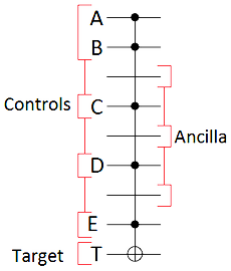

Figure 6

The case with the recordable bits is again the most simpler. The Toffoli gates may be used to intercept controls, and the ancilla bits will store the gradually accumulating intersection of all controls. Finally, each recordable bit will be affected twice, and each control bit - once:

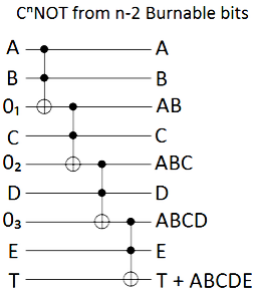

Figure 7

At the zeroed bits must be eliminated the garbage, which is added to the ancilla bits. The elimination is just a matter of applying the same operations in reverse order, by skipping only the operations that affected the target. This creates a circuit that looks like it's pointing towards the target:

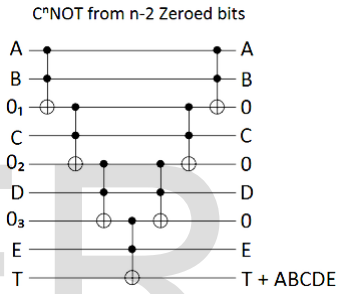

Figure 8

The construction with a garbage bit again is based on detection of toggling, but this time the construction must be nested. The nested detectors of toggling will distribute the toggling, until one of them fails to be activated, so the nesting should be continued, while the target is conditionally toggled. The resulting circuit appears as an arrow, pointing out of the target:

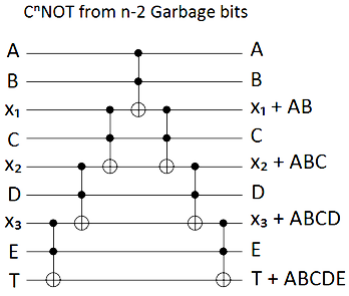

Figure 9

The solution with borrowed bits again combines the detection of toggling and the elimination. Let us first take the solution with the garbage bits, then

IJSER

eliminate the operations which do not affect the target:

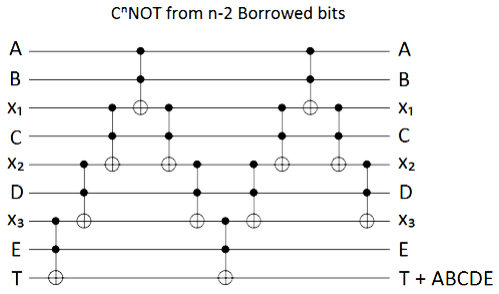

Figure 10

Each of the above constructions uses n - 2 ancilla bits, in order to turn a $C^n\,NOT$ into *O(n)* Toffoli gate.

Putting it all together

The construction with a single ancillary bit is not effective enough to be applied iteratively. Since there are already effective constructions with n - 2 ancilla bits, the efficiency problem can be solved, by switching to a construction with n - 2 borrowed bits after applying the relevant construction with a single ancillary bit.

For example, the circuit, obtained after using one borrowed bit, brakes up $C^7\,NOT$ into four $C^4\,NOT$-s, then brakes up each of these $C^4\,NOT$-s into eight Toffoli gates by borrowing two unaffected bits:

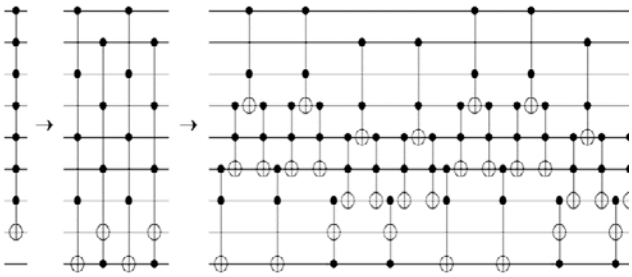

Figure 11

This construction uses ≈ 16n Toffoli gates, achieving the limit O(n). This is asymptotically optimal, because without Ω(n) gates it is not possible to be included enough Toffoli gates, which to affect all involved lines.

The constant factor of the number of the necessary gates depends on the type and number of the ancilla bits. If it is started with a single recordable bit instead of a borrowed bit, then the final number of Toffoli gates is shortened from ≈ 16n to ≈ 8n. If it is started with n bits for borrowing instead of only one, are achieved ≈ 4n. If it is started with n zeroed or garbage bits, giving an advantage in quality and quantity, this will lead to ≈ 2n. The best scenario, n recordable bits, requires only ≈ n gates.

Summary

Without ancilla bits it is impossible to be built operators with many controls from small operations. Multi-dimensional quantum NOT operators can be build with n controls from Θ(n) Toffoli gates with one ancillary bit, even if that bit is in an important unknown state that must be preserved. The presence of a greater quantity or better quality of ancilla bits improves the efficiency of the construction with constant coefficients.

REFFERENCES

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 1st ed. (Cambridge University Press, Cambridge, UK, 2000).
[2] P. W. Shor, SIAM Journal on Computing 26, 1484 (1997).
[3] L. K. Grover, Physical Review Letters 79, 325 (1997).
[4] C. H. Bennett and G. Brassard, in *Proceedings of IEEE international Conference on Computers, Systems and Signal Processing, Bangalore, India* (IEEE Press, New York, 1984), p. 175.
[5] A. K. Ekert, Phys. Rev. Lett. 67, 661 (1991).
[6] C. Elliott, New Journal of Physics 4, 46 (2002). 12
[7] C. Elliott, D. Pearson, and G. Troxel, in *Proceedings of the ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 25-29, 2003, Karlsruhe, Germany.* (PUBLISHER, ADDRESS, 2003), pp. 227–238.
[8] C. Elliott, IEEE Security & Privacy 2, 57 (2004).
[9] C. Elliott *et al.*, in *Current status of the DARPA quantum network (Invited Paper)*, edited by E. J. Donkor, A. R. Pirich, and H. E. Brandt (SPIE, ADDRESS, 2005), No. 1, pp. 138–149.
[10] J. H. Shapiro, New Journal of Physics 4, 47 (2002).
[11] B. Yen and J. H. Shapiro, IEEE Journal of Selected Topics in Quantum Electronics 9, 1483 (2003).
[12] S. Lloyd *et al.*, SIGCOMM Comput. Commun. Rev. 34, 9 (2004).
[13] I.-M. Tsai and S.-Y. Kuo, IEEE Transactions on Nanotechnology 1, 154 (2002).

IJSER

[14] S.-T. Cheng and C.-Y. Wang, IEEE Transactions on Circuits and Systems I: Regular Papers 53, 316 (2006).
[15] J. C. Garcia-Escartin and P. Chamorro-Posada, Phys. Rev. Lett. 97, 110502 (2006).
[16] M. Oskin, F. T. Chong, and I. L. Chuang, Computer 35, 79 (2002).
[17] D. Copsey *et al.*, IEEE Journal of Selected Topics in Quantum Electronics 9, 1552 (2003).
[18] C. H. Bennett and S. J. Wiesner, Physical Review Letters 69, 2881 (1992).
[19] X. S. Liu, G. L. Long, D. M. Tong, and F. Li, Phys. Rev. A 65, 022304 (2002).
[20] A. Grudka and A. W´ojcik, Phys. Rev. A 66, 014301 (2002).
[21] C.-B. Fu *et al.*, JOURNAL OF THE KOREAN PHYSICAL SOCIETY 48, 888891 (2006).
[22] A. Winter, IEEE Transactions on Information Theory 47, 3059 (2001).
[23] H. Concha, J.I.; Poor, IEEE Transactions on Information Theory 50, 725 (2004).
[24] M. Fujiwara, M. Takeoka, J. Mizuno, and M. Sasaki, Physical Review Letters 90, 167906 (2003).
[25] J. R. Buck, S. J. van Enk, and C. A. Fuchs, Phys. Rev. A 61, 032309 (2000).
[26] M. Huang, Y. Zhang, and G. Hou, Phys. Rev. A 62, 052106 (2000).
[27] B. J. Yen and J. H. Shapiro, in *Two Problems in Multiple Access Quantum Communication*, edited by S. M. Barnett *et al.* (AIP, ADDRESS, 2004), No. 1, pp. 25–28.
[28] B. J. Yen and J. H. Shapiro, Physical Review A (Atomic, Molecular, and Optical Physics) 72, 062312 (2005).
[29] B. Sklar, IEEE Communications Magazine 21, 6 (1983).
[30] B. Sklar, *Digital Communications*, 2nd ed. (Prentice Hall, Upper Saddle River, New Jersey 07458, 2000).
[31] P. D. Townsend, Nature 385, 47 (1997).
[32] V. Fernandez *et al.*, in *Quantum key distribution in a multi-user network at gigahertz clock rates*, edited by G. Badenes, D. Abbott, and A. Serpenguzel (SPIE, ADDRESS, 2005), No. 1, pp. 720–727.
[33] Nikolay Raychev. Dynamic simulation of quantum stochastic walk. In International jubilee congress (TU), 2012.
[34] Nikolay Raychev. Classical simulation of quantum algorithms. In International jubilee congress (TU), 2012.
[35] Nikolay Raychev. Interactive environment for implementation and simulation of quantum algorithms. CompSysTech'15, DOI: 10.13140/RG.2.1.2984.3362, 2015
[36] Nikolay Raychev. Unitary combinations of formalized classes in qubit space. International Journal of Scientific and Engineering Research 04/2015; 6(4):395-398. DOI: 10.14299/ijser.2015.04.003, 2015.
[37] Nikolay Raychev. Functional composition of quantum functions. International Journal of Scientific and Engineering Research 04/2015; 6(4):413-415. DOI:10.14299/ijser.2015.04.004, 2015.
[38] Nikolay Raychev. Logical sets of quantum operators. International Journal of Scientific and Engineering Research 04/2015; 6(4):391-394. DOI:10.14299/ijser.2015.04.002, 2015.
[39] Nikolay Raychev. Controlled formalized operators. In International Journal of Scientific and Engineering Research 05/2015; 6(5):1467-1469, 2015.
[40] Nikolay Raychev. Controlled formalized operators with multiple control bits. In International Journal of Scientific and Engineering Research 05/2015; 6(5):1470-1473, 2015.
[41] Nikolay Raychev. Connecting sets of formalized operators. In International Journal of Scientific and Engineering Research 05/2015; 6(5):1474-1476, 2015.
[42] Nikolay Raychev. Indexed formalized operators for n-bit circuits. International Journal of Scientific and Engineering Research 05/2015; 6(5):1477-1480, 2015.
[43] Nikolay Raychev. Converting the transitions between quantum gates into rotations. International Journal of Scientific and Engineering Research 06/2015; 6(6): 1352-1354. DOI:10.14299/ijser.2015.06.001, 2015.
[44] Nikolay Raychev. Quantum algorithm for non-local coordination. International Journal of Scientific and Engineering Research 06/2015; 6(6):1360-1364. DOI:10.14299/ijser.2015.06.003, 2015.
[45] Nikolay Raychev. Universal quantum operators. International Journal of Scientific and Engineering Research 06/2015; 6(6):1369-1371. DOI:10.14299/ijser.2015.06.005, 2015.
[46] Nikolay Raychev. Ensuring a spare quantum traffic. International Journal of Scientific and Engineering Research 06/2015; 6(6):1355-1359. DOI:10.14299/ijser.2015.06.002, 2015.
[47] Nikolay Raychev. Quantum circuit for spatial optimization. International Journal of Scientific and Engineering Research 06/2015; 6(6):1365-1368. DOI:10.14299/ijser.2015.06.004, 2015.
[48] Nikolay Raychev. Encoding and decoding of additional logic in the phase space of all operators. International Journal of Scientific and Engineering

Research 07/2015; 6(7): 1356-1366. DOI:10.14299/ijser.2015.07.003, 2015.
[49] Nikolay Raychev. Measure of entanglement by Singular Value decomposition. International Journal of Scientific and Engineering Research 07/2015; 6(7): 1350-1355. DOI:10.14299/ijser.2015.07.004, 2015.
[50] Nikolay Raychev. Quantum algorithm for spectral diffraction of probability distributions. International Journal of Scientific and Engineering Research 08/2015; 6(7): 1346--1349. DOI:10.14299/ijser.2015.07.005, 2015.
[51] Nikolay Raychev. Reply to "The classical-quantum boundary for correlations: Discord and related measures". Abstract and Applied Analysis 11/2014; 94(4): 1455-1465, 2015.
[52] Nikolay Raychev. Reply to "Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions". Expert Systems; 25(12): 98-105, 2015.
[53] Nikolay Raychev. Classical cryptography in quantum context. Proceedings of the IEEE 10/2012, 2015.

IJSER