# Indexed cluster of controlled computational operators

Nikolay Raychev

**Abstract** - In this article is considered how to turn a n-bit increment operator into an O(n) cluster of controlled indexed computational operators CNOT, NOT and Toffoli. The incrementing cluster computational operators are an extension of the work of the author on the construction of controlled cluster computing NOT-s and the expansion of a NOT operator with many controls into a linear number of NOT-s with two controls. In order to reach the final goal, namely construction of NOT-s with many controls without an ancilla bit, is required the ability to perform large incrementations.

**Key words:** Quantum computing, diffraction, simulator, operators, gates

## 1. INTRODUCTION

Similarly to the construction of an operator with controlled NOT is necessary an ancilla bit, in order to make the construction work, considering that the obstacle with the parity of the permutations applies again. But in this case it is not necessary to be used quantum elements, but normal, classical, reversible circuits.

The cluster models of quantum computations are important both practically as well as conceptually. On the one hand, they lead to new experimental methods, on the other they offer further insight for undiscovered until now properties of the quantum information. The cluster models of quantum computations [1] not only provide a framework for description of interacting quantum fields [2], but they also offer additional practical models for realizations in the quantum computers, when a suitable circuit for qubit encoding [3, 4] is defined. Meanwhile, the cluster models of quantum computations [5] show that the implementation of very difficult to compute wave equations can be avoided only by applying single qubit measurements on suitably prepared multiple entangled states of the resources. The cluster models of quantum computations are a synthesis of these protocols [6.7]. In addition to its inherent conceptual circuit, the formalism represents a potential alternative for implementation of a quantum computer. The optical cluster models of quantum computations have different advantages over the discrete analogues [8]. Any such cluster state may be generated deterministically by offline extraction and passive linear optics [9]. In addition, through alternative methods, large cluster models of quantum computations can be generated simultaneously, using only one optical parametric oscillator (OPO), and not an interferometer [10]; certain such suggestions also have significant potential [11, 12]. These particularities of the cluster models of quantum computations show that they offer useful experimental model for the principles of computations on the basis of measurement [13]. The cluster models of quantum computations involving four optical modes are demonstrated experimentally [14,15,16]. In this article are upgraded the results achieved so far.

## 2. INCREMENTATION

The controlled cluster increment operator increases a value into an additional code, represented by a group of lines. For example, if a 3-bit cluster increment operator is applied to a 3-bit circuit, then the state of the circuit will be cycled from [Off, Off, Off] to [On, Off, Off] to [Off, On, Off] to [On, On, Off] to [Off, Off, On] to [On, Off, On] to [Off, On, On] to [On, On, On] and then back to [Off, Off, Off].

The implementation of a cluster increment operator out of NOT operators is easy, if the presence of a large number of controls is allowed. The increase is only spread, as carrying, through the bits, until it encounters an Off bit. Each bit flips only if all lower bits were On before the start of the incrementation. In other words, each line receives a NOT, which is controlled by all previous lines.

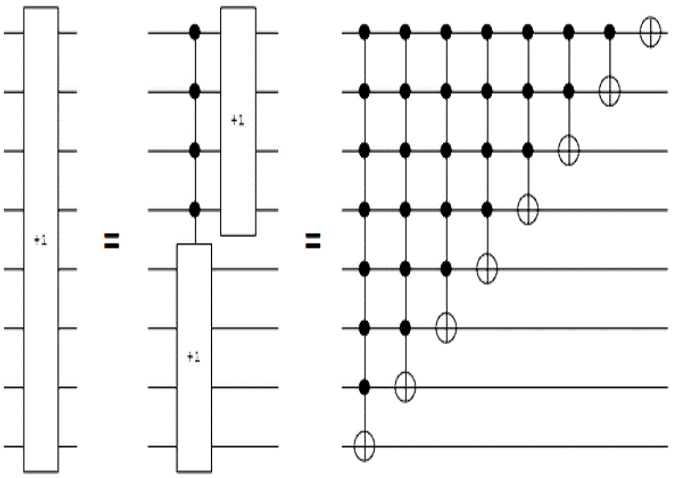


*Figure 1*

The goal is to achieve the same, which make the above circuits, but without using more than two controls on any NOT and without using more than O(n) NOT-s.

At first glance the most easy thing is to apply the construction with large controlled cluster NOT-s [16]. Unfortunately, since this construction requires O(n) operators per each controlled cluster NOT, in the end a quadratic number of operators is needed (because $1+2+3+4+\ldots+n \in \Theta(n^2)$).

Similarly to the last time, the problem will be solved for different types of ancilla bits. There will be considered recordable ancilla bits (initially zero, allowed to end up as non-zero), zeroed ancilla bits (initially zero, required to end up as zero), as well as borrowed ancilla bits (with an unknown initial value, obliged to end up with the same value). The garbage ancilla bits will not be considered, because in this case they do not offer more benefits compared to the borrowed bits.

Let's first focus on the cases with a single ancilla bit.

**Single ancilla bit**

If there is a circuit with n+1 lines with n incrementing lines and one ancilla line, the goal is the incrementation to be broken up into smaller operations. In this section is not necessary to get all the way to the operators of Toffoli. Instead, the size of the operations simply have to be reduced. Once the operations are small enough, possibilities open up because bits, not used by an operation can be borrowed, as ancillary for the relevant operation.

The first case for consideration is when the single ancilla bit is recordable. The top lines, which store the low bits of the number for incrementation may be incremented without depending from the bottom lines in any way. But the bottom lines, which store the high bits must be incremented only when all top lines are On. The bottom lines, depending from n/2 top lines, can be avoided, by storing the intersection of these lines in the ancilla bit. In this way the bottom incrementation needs only one control:

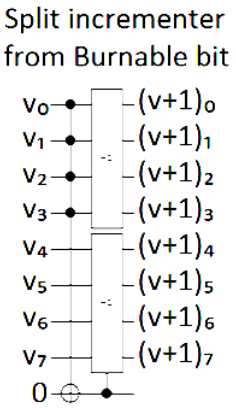


*Figure 2*

It is possible to absorb the single additional control into the increment cluster operator. The controlled cluster increment operator is equivalent to an increment operator with a control line as the new lowest bit, with the exception that the final NOT on the low bit is missing. Such an absorbing control is a matter of subsequent switching of the former control line:

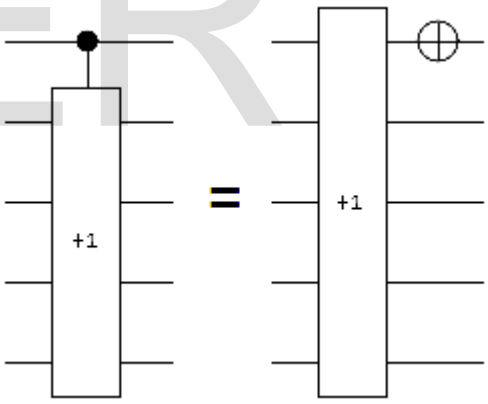


*Figure 3*

It should be noted that the absorbed control bit is treated as the low bit, even if the absorbed line is in "wrong" position. Either the control bit must be swapped in the correct position, or a custom restructured cluster increment operator will be needed.

The next event for consideration is with a single zeroed bit. Everything that is needed here, is to take the solution from the case with the recordable bit and to be canceled the effects on the ancilla bit. This is a simple addition to the circuit, because in reality there is only one effect, and it is easily reversible:

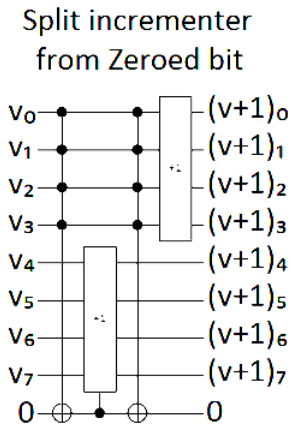


*Figure 4*

The last case with a single ancilla bit is the case with the borrowed bit. This time the solution is much more complicated.

In the last article of the author [16] was used the detection of switching, when working with garbage and borrowed bits. There a self-undoing operation was repeated twice, stated by the ancilla bit, so that the operation would undo itself, unless the bit is not switched. This does not work for the incrementation because the incrementing operation is not inverse to itself.

The trick here is to use a bit-wise addition. When the bits of a number in an additional code X are switched, they toggle from storing of X to storing of $\bar{X} = -X - 1 (mod\ 2^n)$
If the complemented value is incremented, after which the complement is taken again, then finally is obtained $\overline{\bar{X}+1} = \overline{-X-1+1} = -(-X) - 1 = X - 1$
In other words, the surrounding of an increment operator with NOT-s turns it into a decrementing! (and vice versa.)

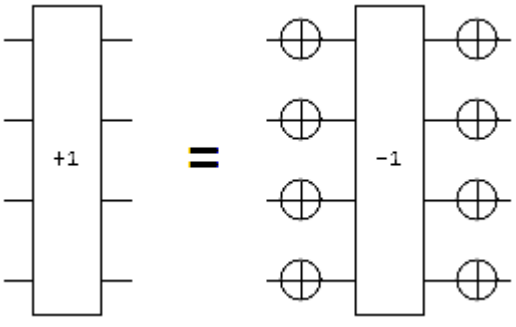


*Figure 5*

The conversion from increment to decrement is useful because now an increment can be turned into the opposite of increment and this can be done conditionally. In this way can be made the detection of switching to work in this case:

- Let's apply an incrementing and decrementing cluster operator to the high bits of the number.
- Both operations are determined by the borrowed ancilla bit.

- Each time the low bits are On, the ancilla bit and high bits are switched, before and after the decrement operator.
    - If the low bits are not On, nothing happens with the high bits. Either the borrowed ancilla bit is Off, which means that neither the increment nor the decrement cluster operator has an effect, or the borrowed ancilla bit is On and the increment and decrement operator are applied, by undoing one another. In both cases the network effect is not an effect.

    - If the low bits are On, then the NOT-s around the decrement cluster operator trigger and transform the decrementation into incrementation. If the borrowed ancilla bit is On, the increment cluster operator is triggered, and the operator, transformed from decrementing into incrementing, is not. Otherwise the borrowed ancilla bit is Off and only the operator, transformed from decrementing into incrementing, is triggered.

So according to the above plan the high bits are incremented exactly once, when the low bits are On, but nothing happens with the high bits, if any of the low bits is Off. That is the demanded logic for the high bits.

Below is represented the construction with a single borrowed bit for 8-bit numbers:

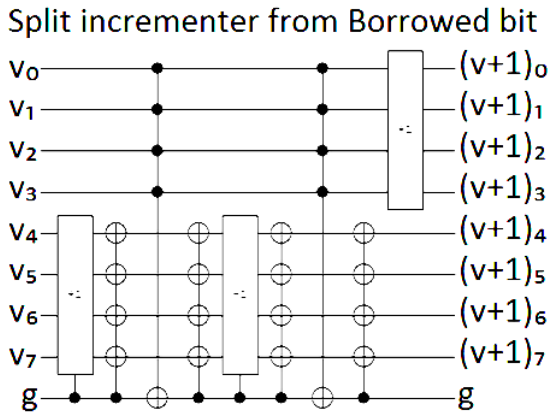


*Figure 6*

The above circuit uses another method, which was not mentioned. Because the switching of as many bits as possible is requested, when the first n/2 bits are On, but without paying the quadratic price for having n/2

controlled NOT-s with size n/2 is used a detection of switching, to spread the ancilla bit to all target bits.

The constructions with a single bit, which were considered, allow the conversion of each n-bit increment cluster operator into two or three n/2-bit cluster increment operators, depending on the type of the used ancilla bit. These constructions can be applied over and over again, reducing the size of the remaining operators in half, until reaching simple, base cases, but that would not be asymptotically effective.

For example, differential equation for iteration of a construction with a single borrowed bit is $T(n) = 3T\left(\frac{n}{2}\right) + 0(n)$ and this means that $T(n) \in 0(n^{\log_2 3}) \approx 0(n^{1.585})$, and not O(n). The differential equation for zeroed bits is $T(n) = 2T\left(\frac{n}{2}\right) + 0(n)$, which is better, but still gives $T(n) \in 0(n\ \log(n))$ instead of O(n).

To achieve a linear number of operators is necessary to be borrowed many more bits.

**n ancilla bits**

If there is a circuit with 2n lines, with n target lines and n ancilla lines, then the target lines must be incremented. And this must be done with at most O(n) Toffoli operators or less.

The case with recordable bits needs only n-2 from the n available ancilla bits. By using the recordable bits for accumulation of the intersection of more and more controls are precisely obtained the conditions, necessary for updating each target bit.

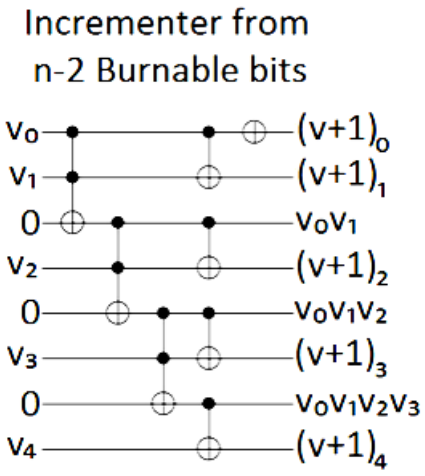


*Figure 7*

The case with zeroed bits is solved by taking the solution with recordable bits and eliminating the previous effects. In other words, it is simply cleared:

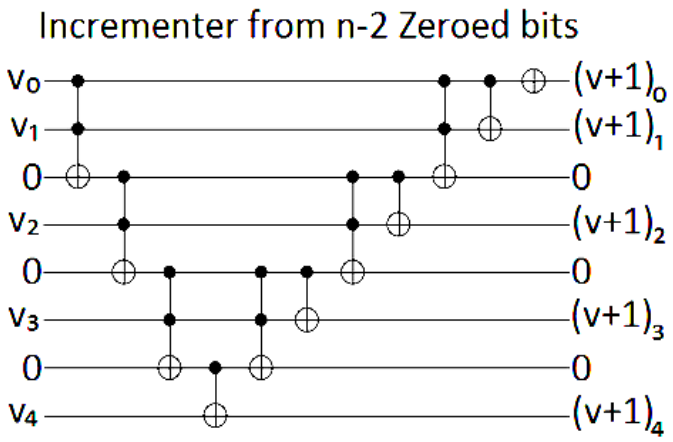


*Figure 8*

This leaves only the case with n borrowed bits.

The solution of this case is not at all easy to find. For it can be used the VanRentergem adder:

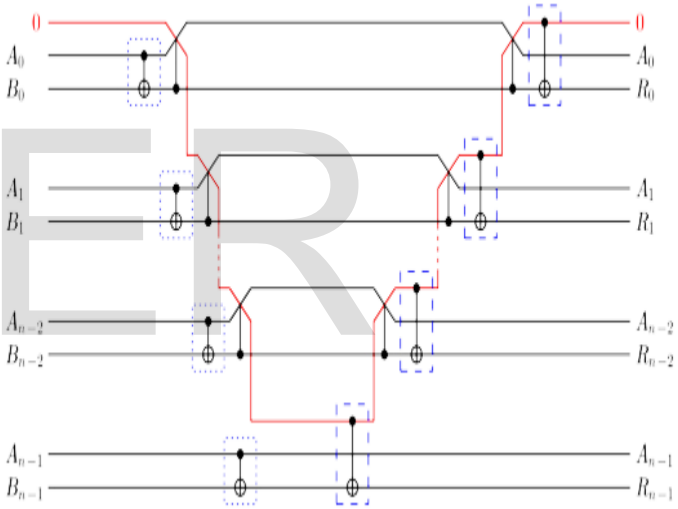

*Figure 9*

The VanRentergem adder takes a carry bit c, a value in additional code a, a value in additional code b and turns (c, a, b) into (c, a, a+b+c). Let's use a modified version of the circuit, made out of Toffoli operators instead of operators with controlled exchange, and to reverse the operators, so as to perform subtraction instead of addition:

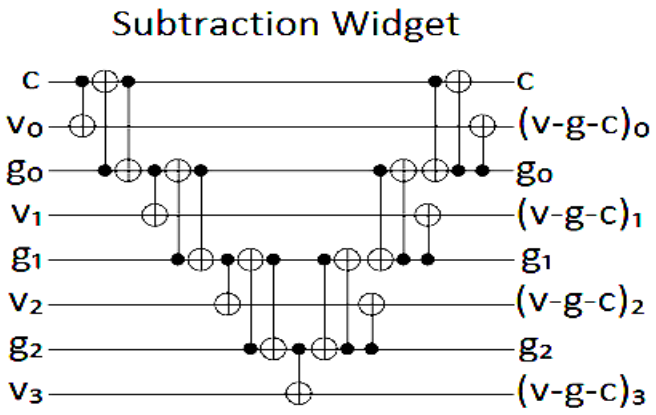


*Figure 10*

With the above widget can be subtracted a garbage carry bit c and a garbage value in additional code g out of the target value for incrementation v.

On its own the subtraction seems like the wrong action to be taken. In the end, it mixes a bunch of garbages in the target. However, the trick with bitwise complement can be used in order to solve the problem. If the widget with subtraction is applied once again, but at first are switched the bits of g, then the target bits will be relocated from storing of v to storing of v-c-g to storing of $v-c-g-(-g-1)-c=v-2c+1$. By eliminating the garbage from g, is created +1, which must be performed!

Now it is necessary to eliminate the garbage from c. Let's consider how $v-2c+1$ behaves for each possible value of c. When c is On, is obtained $v \rightarrow v-2+1=v-1$, which means that v is decremented. When c is Off, is obtained $v \rightarrow v+1$, which means that v is incremented. An increment is always wanted and a decrement can be easily turned into an increment, by surrounding it with NOT-s, but conditioned on c.

In the end are obtained two subtractions in the style of VanRentergem. The garbage with a non-carry bit is canceled out by switching before and after one of the subtractions. The garbage with a carry bit is canceled out by pre-and post-switching of the target bits, when the carry bit is On. Finally, since the target value has one more bit than the garbage value, the highest bit needs a special processing. As a whole is obtained the following:

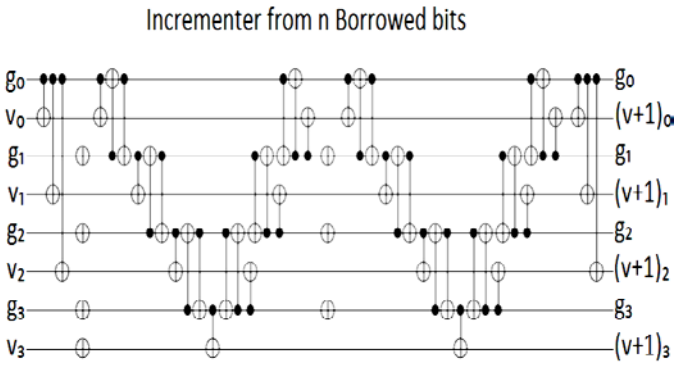


*Figure 11*

With these asymptotically efficient constructions with n ancilla bits in hand can be repaired the lack of efficiency in the constructions with a single bit.

**Putting it all together**

Basically in order to turn a n-bit incrementation with a single ancilla bit into a linear number of Toffoli operators or smaller, must be applied the appropriate construction with a single bit and then to be applied the appropriate construction with n borrowed bits. However, there are a few stipulations.

First, after the construction with a single bit has been applied, the largest remaining operation has a size of $\left\lceil \frac{n}{2} \right\rceil$ and therefore has access to at most $\left\lceil \frac{n}{2} \right\rceil$ unaffected bits for borrowing. Since $\left\lceil \frac{n}{2} \right\rceil$ can be with one of less than $\left\lceil \frac{n}{2} \right\rceil$, sometimes it is necessary to be applied the construction with a single bit twice before the operations to be small enough in order to borrow enough bits to apply the construction with n bits. Alternatively, since the increments have only one controlled NOT, which affects all relevant lines, that operation can be subtracted from the increment (reducing the size of the bit for incrementation with 1) and to be processed separately.

Second, the constructions make (a constant number of) operators in controlled NOT-s in addition to the created increment operators. They are processed by applying the construction from the article for large controlled NOT-s.

Third, although this construction is asymptotically efficient, it has a large constant coefficient. It seems that the n-bit incrementation turns into about 32n operators of Toffoli or smaller. Probably there are solutions with better constant coefficients.

To ensure that the described construction actually works, is created a Python code for testing. The use of the code for generating the 5-bit incrementing circuit, then its breaking by hand, so to be fit on the page, gives:

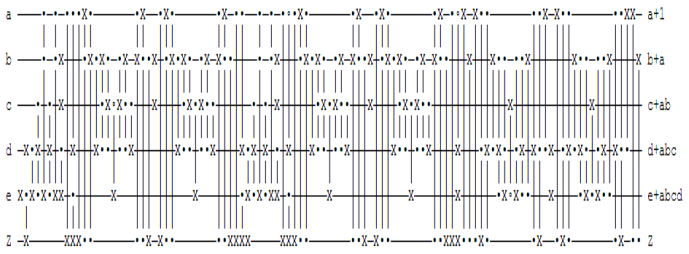


*Figure 12*

Below are given gradually larger and larger cases, which are zoomed out:

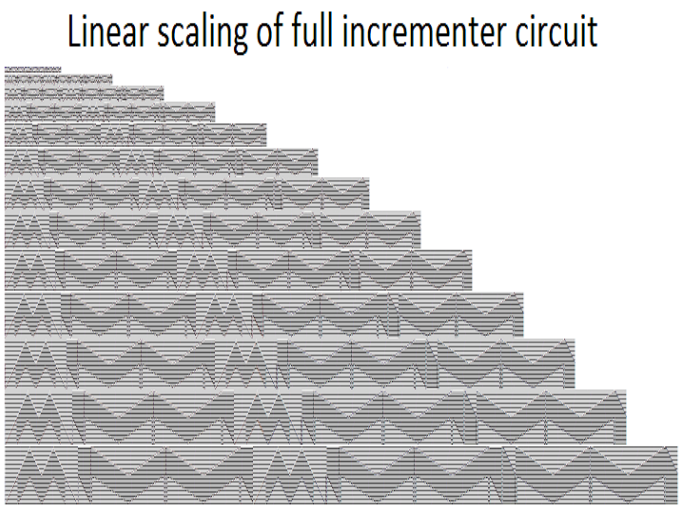


*Figure 13*

## 3. SUMMARY

Given a single ancilla bit, in an unknown state that must be preserved, can be created cluster increment operators with n lines, by using O(n) operators of Toffoli or smaller.

The key parts of the construction are the subtraction in the VanRentergem-style and the use of bitwise complements for conditional switching between addition and subtraction.



**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).
[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.