FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 24, no. 1, April 2011, 73-89

Synthesis of quantum circuits in Linear Nearest Neighbor model using Positive Davio Lattices

Marek Perkowski, Martin Lukac, Dipal Shah, and Michitaka Kameyama

Abstract: We present a logic synthesis method based on lattices that realize quantum arrays in One-Dimensional Ion Trap technology. This means that all gates are built from 2x2 quantum primitives that are located only on neighbor qubits in a one-dimensional space (called also vector of qubits or Linear Nearest Neighbor (LNN) architecture). The Logic circuits designed by the proposed method are realized only with 3*3 Toffoli, Feynman and NOT quantum gates and the usage of the commonly used multi-input Toffoli gates is avoided. This realization method of quantum circuits is different from most of reversible circuits synthesis methods from the literature that use only high level quantum cost based on the number of quantum gates. Our synthesis approach applies to both standard and LNN quantum cost models. It leads to entirely new CAD algorithms for circuit synthesis and substantially decreases the quantum cost for LNN quantum circuits. The drawback of synthesizing circuits in the presented LNN architecture is the addition of ancilla qubits.

Keywords: Reversible logic synthesis, lattice, leinear nearest neighbor model.

1 Introduction: Standard versus Linear Nearest Neighbor quantum cost models

Most papers in the literature about automated synthesis of quantum and reversible (permutative) circuits are not related to any particular quantum realization technology [1, 2, 3, 4, 5, 6]. Their models assume that a gate can be realized on any subset of qubits. The model used in most of the previous permutative quantum

Manuscript received on February 2, 2011.

M. Perkowski and D. Shah are with Department of Electrical and Computer Engineering, Portland State University, Portland, OR, USA, (email: mperkows@ee.pdx.edu). M. Lukac and M. Kameyama are with Graduate School of Information Sciences, Tohoku University, Sendai, Japan, (email: lukacm@kameyama.ecei.tohoku.ac.jp).

circuit synthesis papers assumes that there can exist a gate located between any two qubits, even if these qubits are located far away in physical space (in vector) one from another. This (very approximate) assumption may be sufficient to calculate quantum costs for very small circuits. This assumption was accepted in a theoretical framework but from a practical point of view and with respect to particular technologies (such as Ion trap in this case) creating gates on arbitrary qubits is not only extremely difficult but also cost ineffective; each gate has to be properly converted and realized in an LNN architecture. Thus, in general architecture independent synthesis models are sufficient to approximate the real cost of small circuits. For larger quantum circuits realized in the future as well as for currently realizable circuits with about 12 qubits architecture dependent cost models and synthesis methods are required. For instance in quantum optics [7, 8] such architectural models require more development to take into account more complex constraints such as time propagation and physical size.

There exists no single technology for which this model is valid. In contrast, for various realization technologies there exist different neighborhoods of qubits [9, 10]. For instance, in the One-Dimensional Ion Trap technology [11] the qubits create a linear, one-dimensional (1D) vector, the Linear Nearest Neighbor (LNN) model (architecture). In quantum optics, qubits also interact by proximity using optical wires or crystals [12, 13, 7]. Therefore, it is safe to assume that the LNN cost model is currently one of the most appropriate models for current technologies. Circuits realized in LNN use quantum gates defined only on neighbor qubits and the gates are built from 1 x 1 and 2x2 quantum primitives. We believe that LNN model should be used for Ion Trap and similar technologies and new quantum cost models should be developed for other specific technologies.

With respect to general quantum circuits the LNN Model was introduced by Fowler et al [14] for designing a Quantum Fourier transform circuit. Their work was improved in [15]. Paper [10] considers theoretical aspects of techniques for translating quantum circuits between various architectures. The first paper about permutative quantum circuits design with the LNN model was written by Cuccaro et al [16] and they designed a ripple-carry addition circuit. Automated synthesis of general quantum circuits with LNN model was first introduced briefly in [17] but no specific method was presented and results analyzed. Chakrabarti and Sur-Kolay [18, 19] presented analysis of costs of single-output FPRM-based reversible circuits. Methods for general quantum circuit for the LNN model were discussed by Hirata et al [20] and other authors [21, 22, 23, 24] created various methods to synthesize reversible quantum circuits in the LNN model. These methods are called ”nearest neighbor quantum synthesis”. For instance, Hirata’s method [20] starts from an arbitrary quantum array and modifies it to the LNN Architecture by

inserting SWAP gates and minimizing their number. The advantage of this method is that it can be applied to an arbitrary quantum circuit. The number of added gates is however excessive and the properties of permutative circuits being the special case of quantum circuits are not taken into account. Thus developing an algorithm specifically designed for the LNN architecture can significantly improve the cost of circuits realized in the LNN model because the circuits are specifically crafted to match the architecture rather than designing a reversible circuits with arbitrary gates and then modifying it to match the architecture.

Moreover, most of the methods whether using the LNN model or not do not evaluate and compare the differences between the used cost and the LNN model. This means that despite claiming minimal results while using multi-controlled Toffoli (MCT) gates the same results can be shown to be non optimal when using the LNN model.

We present here a new approach based on lattices, which applies to only permutative (reversible quantum) circuits. The method does not only exploit the local minimization of SWAP gates as in the previous works such as [20, 24] but, by synthesizing circuits as a lattice the method uses the lattice structure to design circuits that are less costly when designed for the LNN model. We start from an arbitrary (non-reversible) Boolean function and realize it as a reversible quantum circuit; the method presented here converts a non-reversible function to a reversible circuit by adding ancilla qubits. The proposed approach presents for the first time a conversion of an arbitrary Boolean function to a circuit with a quantum cost model that takes technology-related considerations into account in the logic synthesis algorithm.

It uses two quantum cost functions; standard quantum cost and Linear Nearest Neighbor model (LNN). The LNN model assumes that circuit is designed or modified in such a way that it is composed of only 1*1 gates and 2*2 gates on neighboring qubits. To evaluate our approach we compare the two costs methods as well as we compare the obtained costs with various previous algorithms.

The main contributions of the proposed algorithm is in the fact that it allows to generate reversible logic circuits in the LNN model with a practically achieved minimum of SWAP gates. This is verified with other algorithms the generate circuits with higher number of not only SWAP gates but a higher number of gates in general.

The paper is organized as follows. Section 2 explains the standard and LNN models of calculating quantum costs. Section 3 presents how one type of the previously introduced Lattice Diagrams, the Positive Davio Lattices, can be adapted to regular realizations of quantum circuits for the standard and LNN model models. Section 4 presents our experimental results with both cost models and Sec-

tion 5 concludes the paper. The paper assumes that the reader is familiar with basic quantum gates and reversible logic concepts and with previous works on lattice algorithms.

2 Motivation for the LNN model for quantum arrays realized in Ion Trap

A gate between any two qubits would mean an immediate direct interaction between any two ions in the Ion Trap, which is physically impossible in this technology due to space separation [11, 25]. In the simplest (but practical as of 2011) case, all ions in Ion Trap are placed linearly (as a One-Dimensional vector). Every ion (qubit) can interact with at most one neighbor above and one neighbor below. This physical constraint of "2-neighbor" quantum layout of the substrate has much influence on practical designs. As an example of problems with LNN circuit model, consider the very simple 4x4 Toffoli gate shown as a unit in Fig. 1(a). Other authors [26, 27, 28, 29, 30, 31, 32] calculate the quantum cost of the gate as a function of number of inputs regardless of what is the distance of the qubits used in this gate. This is not accurate when the circuit is realized in linear Ion Trap technology. Nor is it good for quantum optics or NMR technology that is currently in use. To realize this circuit in the LNN model, one ancilla bit should be added as in Fig. 1(b). Next, each of the 3x3 (standard) Toffoli gates from Fig. 1(b) are macro-generated to the Barenco's realization of this gate [33], thus creating the quantum array in Fig. 1(c). This would be fine if every two qubits can interact directly: but they cannot. So transformations from Fig. 1(d) to create 2-neighbor-only type of circuits are required. The final circuit for the gate from Fig. 1(a) is then shown in Fig. 1(e). It has 27 2x2 gates in 2-neighbors-only topology after the minimization of certain gates. There are other ways to realize this gate in layout, even without ancilla bit. They are however even more expensive when realized in linear Ion Trap. The number would be even higher if the gate would be realized on five qubits that are not neighbors.

Based on the above example, the quantum circuits in the LNN architecture should have short connections inside gates. As discussed in [34, 35, 36] short connections require regular structures such as Lattices [34, 37, 38, 39] created by adaptation and generalization of Akers Arrays [40]. The method proposed here uses Positive Davio Lattices (PDL) [41, 42]. The reason for using PDL comes from the fact that after we analyzed the mapping of Lattice diagrams to LNN architecture circuits, we found that the internal connections of the Lattice can be mapped well, i.e. with small distances. There is however a big trouble with connecting all Toffoli gates to input variables: this involves very many SWAP gates. This is illustrated

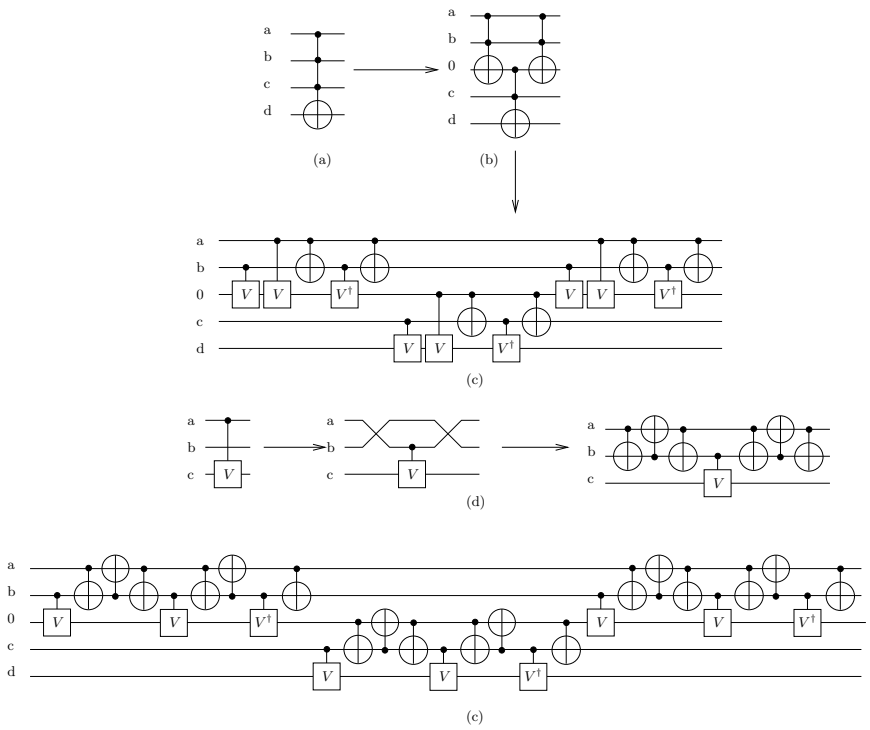

Fig. 1. This example illustrates the nature of a problem with linear Ion Trap. A 4x4 Toffoli gate that looks a cheap gate which is however quite expensive when mapped to linear-neighborhood quantum array. (a) symbol of a gate as used by other authors, (b) decomposed Toffoli gate, (c) the final circuit with 2-qubit quantum primitives, but not-realizable in linear neighborhood as it has wires going over gates, (d) steps to realize the gate with a wire going over it, (e) the final circuit in linear neighborhood Ion Trap.

in Fig. 4 and Fig. 5. Fig. 4 shows standard quantum array with ancilla bit for function FX2 realized on a PLA-like structure using only two-controlled qubits Toffoli gates and Feynman Gates. Fig. 3 shows the same function rewritten to our 1D neighborhood model by adding SWAP gates. This example illustrates the big cost of SWAP gates when they are added to calculate a realistic quantum cost for LNN model of qubits required in Ion Trap. The same property can be shown on any published circuit for well-known benchmarks.

Finally, one can observe that the number of the SWAP gates required for an arbitrary reversible circuit be mapped to the LNN model can be approximated analytically.

Lemma 2.1. *An arbitrary multi-controlled Toffoli (MCT) gate with k-1 control bits and 1 target bit (together having k bits) that is defined over p wires ($p \geq k$, including*

skipped wires) requires at maximum:%and realized in using the CNOT, CV/CV† gates requires at minimum

$$\hat{s} = 2 * (p-k) * (k-1) + (p-1) * 2 \tag{1}$$

with $2 * (p-k) * (k-1)$ representing the number of swap gates required to bring the control bits to the LNN proximity of the target bit and the $(p-1) * 2$ term represents the number of SWAP gates required to bring the qubits inside of the Toffoli gate itself to the LNN neighborhood. This means that for a reversible circuit realized by only Toffoli gates a maximum of $\hat{s}$ gates will make the circuit into a LNN compatible circuit. Note that equation 1 does only specify how many SWAP gates are required to group the controls in a LNN model.

Proof. Consider the Toffoli gate shown in Figure 2(a). The gate is defined over 6 qubits and has 2 control bits and one target. In this particular case, the distance of the control qubits and of the target is maximal and thus using formula from eq. 1 we obtain the correct result 12 SWAP gates for outside of the MCT gate (Figure 2(b)) and 14 as the total number of SWAP gates(Figure 2(c)).

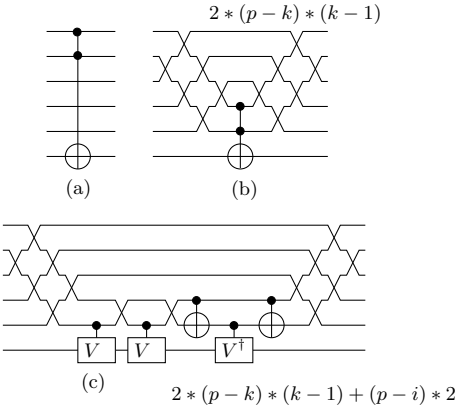

Fig. 2. MCT gate defined over 6 qubits. (a) realized as standard MCT, (b) realized in the LNN model, (c) realized in the LNN model also within the gate itself.

Because the gate in Figure 2(a) has maximal distance between the control and the target qubits, any other configuration of the same gate will require less or equal to 14 SWAP gates. □

Now let's look at a more complex example with two MCT gates. The circuit in Figure 3(a) shows two MCT gates connected in series. Observe that when the circuit is built using standard methods of synthesis (Figures 3(b)) - by building the circuit from MCT gates and then converting it to the LNN model - the cost of the

SWAP gates is much higher than when the circuit is built using synthesis methods for the LNN architecture (Figure 3(c)). This is because in the algorithm that builds circuits for the LNN model one can directly predict which lines should be returned to their initial position right after being used and which not.

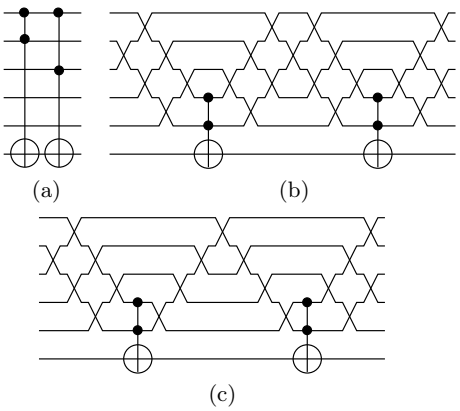

Fig. 3. Two MCT gates defined over 6 qubits. (a) realized as standard MCT, (b) realized as two MCT gates transformed to the LNN model, (c) realized as two MCT designed for the LNN model.

Finally, we believe that LNN cost model should replace the standard cost model for Ion Trap technology. New cost models should be also created along these lines also for other quantum technologies, rather than using a "general" cost which has no relation to any technology that we are aware of. Even for small circuits costs calculated with the LNN model differ much from standard quantum costs and different types of quantum synthesis method show better cost minimization abilities. Our new CAD tool for standard and LNN models of quantum costs is called QULASYN (QUantum LAttices SYNthesizer).

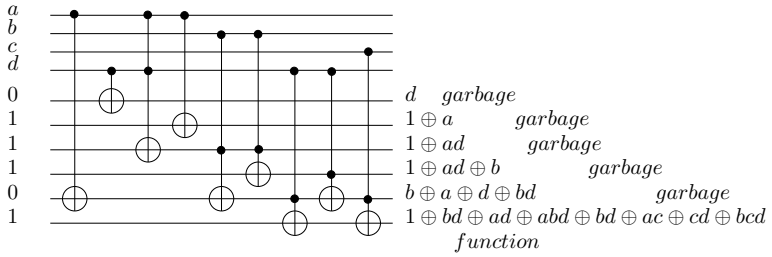

Fig. 4. Circuit for function $FX2(a,b,c,d) = 1 \oplus bd \oplus ad \oplus abd \oplus bc \oplus ac \oplus cd \oplus bcd$ created with our method for traditional quantum cost function calculation that does not take Ion Trap technology constraints into account.

3 Lattice Diagrams with various types of expansion gates and their mapping to LNN model

As already introduced, in this paper we adapt Positive Davio Lattice Diagrams to quantum circuits [36]. Unlike in the standard Shannon Lattice Diagrams that uses multiplexers we restrict ourselves to build quantum lattice equivalents for only Positive Davio Lattices, using only 3*3 Toffoli, Feynman and NOT gates.

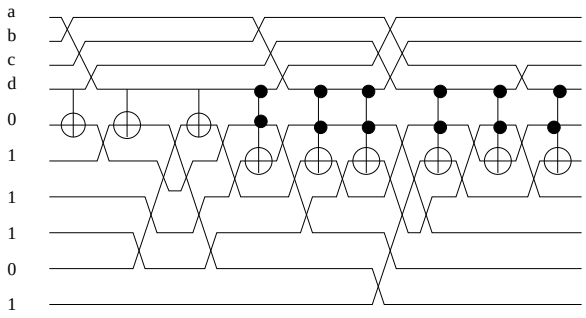

Fig. 5. Circuit from Figure 4 modified with adding SWAP gates for new cost function calculation that does take Ion Trap technology constraints into account, with 36 SWAP gates added. It has 36 SWAP gates added to realize LNN model quantum cost, obviously increased.

As an example, consider the classical Positive Davio Lattice for function FX2(a,b,c,d) illustrated by a diagram shown in Fig. 6. It is designed using software presented in [37, 36]. The algorithm for the lattice starts from a logic Exclusive-Sum-of-Products (ESOP) equation describing the desired function. Initially a variable is selected and both the f_x, $f_{\overline{x}}$ and $f_x \oplus f_{\overline{x}}$ is calculated. This is repeated for every variable until all available variables are constant. During the process of the lattice construction redundant nodes are removed, merged with the goal of minimizing the size of the lattice. Such lattice then can be further explored using for instance sifting or variable repetition to obtain the most desirable lattice.

The Positive Davio Lattice is next transformed to a standard form of a quantum array. For instance, to help the reader, the lattice from function F3(a,b,c) is presented in Fig. 7(a) in a form that is intermediate between a Lattice Diagram and a Quantum Array. This intermediary form is transformed as in Fig. 7(b), where every intersection of wires from Fig. 7(a) is replaced by a SWAP gate in Fig. 7(b). This way, a new type of regular structure realized in quantum array with regular connections is obtained and the long connections typical for standard Toffoli gates are avoided. Figure 7 illustrates the essence of our transformation method from lattice diagrams to quantum arrays of permutative circuits. It explains also why we use non-standard notation for intermediate stages. The whole trick was to twist the diagram and replace intersections of lines with SWAP gates. This graphical

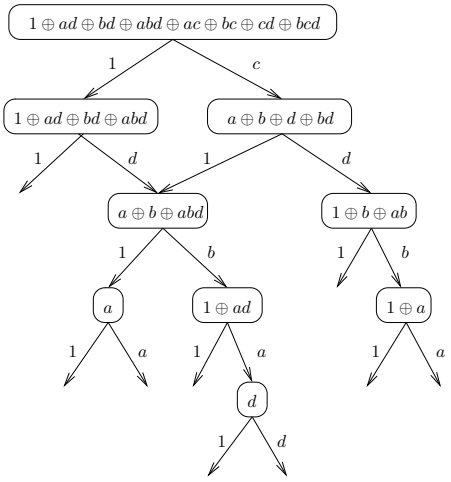

Fig. 6. Example of Positive Davio Lattice from [37]. Positive Davio Expansion is applied in each node. Variable d is repeated.

method explains also our SWAP insertion algorithm.. The number of SWAP gates in our method is however smaller because of regularity of the new structure from Fig. 7(a). We do not present here the detailed algorithms to create Positive Davio Lattices as they are discussed in full detail in previous papers [37, 35, 36]. but we provide a high level description for the sake of reader's understanding.

As can be seen the proposed method generates additional qubits. The number of the garbage qubits is the same as in standard quantum array however there are additional SWAP gates required to realize the LNN model. This results in circuits where there are no Toffoli gates realized on non-neighbor qubits. The cost of added SWAP gates is relatively low as each such gate can be realized with 3 Feynman gates [9], or 11 EM pulses [43] after optimization.

Fig. 8 presents the transformation of standard Positive Davio Lattice from Fig. 7(a) drawn in another way to a regular quantum array with addition of SWAP gates. Fig. 8 shows the transformation from the macro-level to the CNOT/CV gates as well as the transformation to the closest-neighbor model applicable for Ion trap technology. These Controlled-Square-Root-of-Not (CV) gates and their hermitians are explained in detail in [9]. They are a good approximation of the quantum cost in the Electromagnetic (EM) pulses.

Using the transformations to pulses as shown in Fig. 9, the final circuit cost can be calculated as follows. Each of the blocks of gates shown in Fig. 8 has the cost shown in Fig. 9(c). The cost is 20x1 + 8x2 = 36. Because in between the blocks some of the gates can be combined, it can be shown that when the circuit

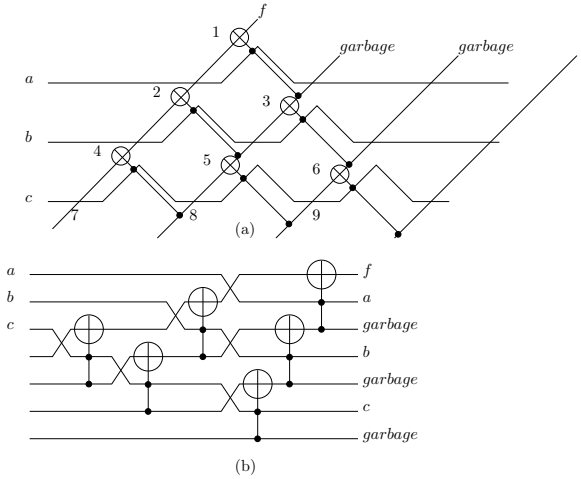

Fig. 7. Transformation of function F3(a,b,c) from classical positive Davio Lattice to a Quantum Array with Toffoli and SWAP gates. Each SWAP gate is next replaced with 3 Feynman gates. (a) intermediate form, (b) final Quantum Array.

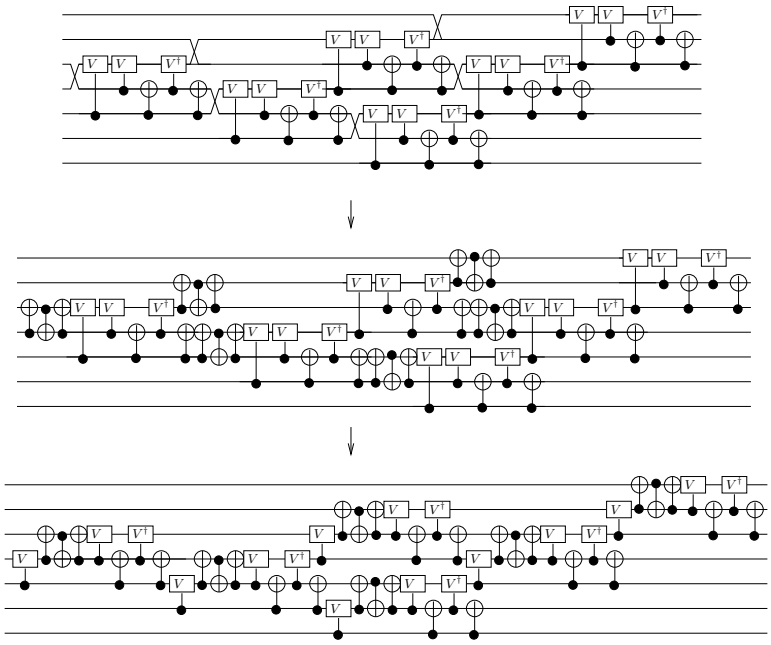

Fig. 8. Transformation of the circuit realized in Fig. 7 using Toffoli gate. Each Toffoli and SWAP gates are replaced by quantum CNOT and CV/CV† quantum gates and rearranged to satisfy the neighborhood requirements of Ion trap.

from Fig. 8 is built in EM pulses its total cost is 36x6 - 3 = 213. This way, we can use the regularity of quantum array on the lowest implementation level (quantum rotations level [9]) to further reduce the number of EM pulses. We used the method from[43] to reduce the number of EM pulses.

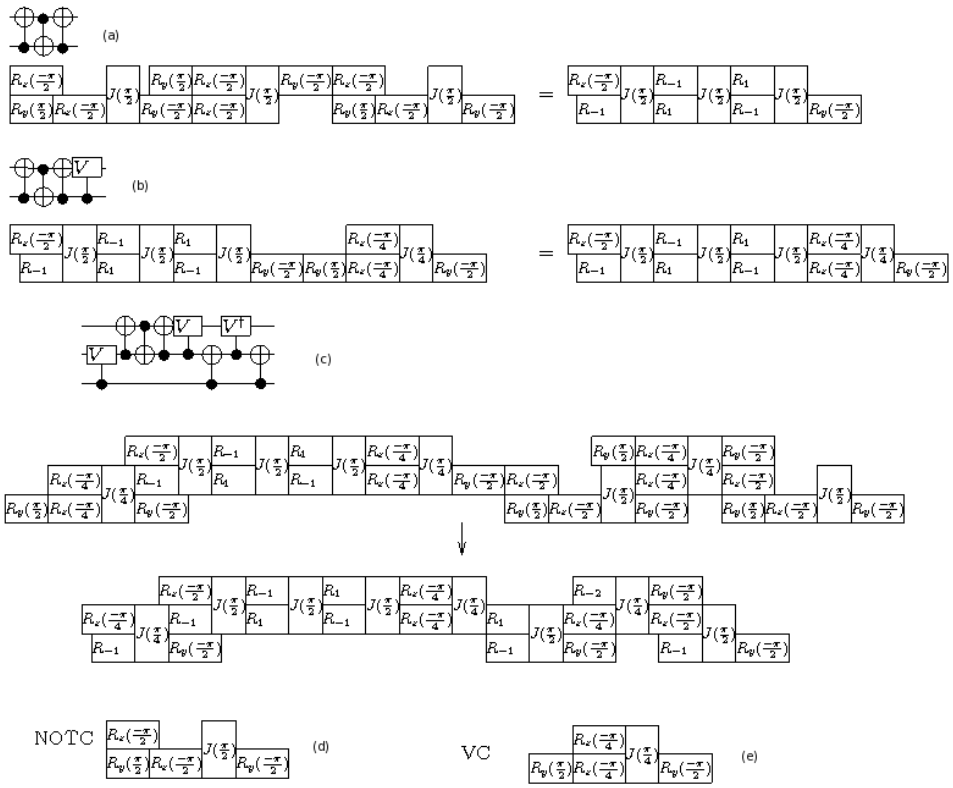

Fig. 9. The transformations of blocks of quantum gates to the pulses level.

The circuits in this paper are designed using a CAD tool QULASYN (QUantum LAttices SYNthesizer) that uses both the standard and LNN models of calculating quantum cost functions. More details on the implementation of the QULASYN, on the variable ordering and repetition as well as on synthesis with new gates can be found in [36, 44]. The algorithm uses various optimization techniques and accelerations in order to perform an efficient KFDD such as preorder search, swapping of Davio nodes, sifting or exploitation of the symmetry of the KFDD. However these techniques are not described in this paper as they are not proper to the KFDD applied to reversible synthesis. The particularity of the proposed approach is the fact of using only Toffoli, CNOT and the Not gate.

4 Experimental results

Experimental results for calculating Quantum arrays with traditional quantum costs done by our QULASYN tool are given in Table 1. We compare our Lattice tool with MMD and Agrawal/Jha software. MMD stands for Miller, Maslov and Dueck's algorithm and AJ stands for Agrawal and Jha algorithm. This table shows advantage of applying lattice based quantum synthesis even for traditional cost functions with gates in standard quantum array. The best quantum costs are bold, italic and underlined. The same costs for more than one method are bold and italic. Our tool created the best result in 9 cases and in 8 cases the same cost results were found. Dashes are for results that we have no access to.

Table 1. The transformations of blocks of quantum gates to the pulses level.

Benchmark	#Real inputs	#Garbage inputs	#Gares Lattice	Cost Lattice	CPU time Lattice	#Gates DMM	Cost DMM	#Gates AJ	Cost AJ
2to5	3	4	31	107	0.12	15	107	20	10
rd32	5	1	4	8	< 0.01	4	8	4	8
rd53	5	5	11	39	< 0.01	16	75	13	16
rd84	8	7	20	68	< 0.01	28	98	- -	- -
5bitadder	10	5	29	55	< 0.01	29	55	- -	- -
8bitadder	16	8	122	322	0.10	122	322	- -	- -
3_17	3	1	10	21	< 0.01	6	12	6	14
6sym	11	4	19	75	0.37	20	62	NA	NA
9sym	15	5	25	101	0.40	28	94	NA	NA
5mod5	5	1	14	58	< 0.01	10	90	11	91
4mod5	4	1	6	18	< 0.01	5	13	5	13
ham3	3	0	3	7	< 0.01	5	7	5	9
ham7	7	4	21	61	< 0.01	25	49	23	81
ham15	7	15	9	47	0.10	191	205	- -	- -
xor5	5	0	4	4	< 0.01	4	4	4	4
xor20	20	0	19	20	< 0.01	19	19	19	19
xnor5	5	1	5	5	< 0.01	- -	- -	- -	- -
dcod24	4	2	10	30	< 0.01	- -	- -	11	31
cycle0_2	12	6	180	860	29.7	19	1198	- -	- -
cycle7_3	20	10	920	4160	40.10	48	6057	- -	- -
graycode6	5	5	5	5	< 0.01	5	5	5	5
graycode10	10	9	9	9	< 0.01	9	9	9	9
graycode20	10	19	19	19	< 0.01	19	19	19	19
nth_prime3_inc	3	4	4	6	< 0.01	4	6	- -	- -
nth_prime4_inc	4	5	16	48	< 0.01	12	58	- -	- -
nth_prime5_inc	5	5	29	91	0.22	26	78	- -	- -
nth_prime6_inc	6	6	148	586	0.36	55	667	- -	- -
Alu	5	2	5	17	< 0.01	- -	- -	18	114
4_49	4	4	16	52	0.04	16	58	13	51
hwb4	4	4	12	28	< 0.01	17	63	15	35
hwb5	5	5	24	96	1.20	24	104	- -	- -
hwb6	6	6	32	128	2.00	42	140	- -	- -
hwb7	7	6	49	185	0.10	35	203	- -	- -
pprm1	4	4	9	33	< 0.01	- -	- -	- -	- -
pprm2	10	6	55	235	0.50	- -	- -	- -	- -
pprm3	15	12	29	540	0.50	- -	- -	- -	- -

Table 2. The transformations of blocks of quantum gates to the pulses level.

Benchmark	#Gates Lattice	Cost Lattice	#Gates with SWAP insertion for Lattice	Cost with SWAP gates for Lattice	#Gates DMM	Cost DMM	#Gates with SWAP insertion for MMD	Cost with SWAP gates for MMD
2to5	31	107	61	197	15	107	31	155
rd32	4	8	8	20	4	8	6	14
rd53	11	39	44	138	16	75	72	273
rd84	20	68	52	164	28	98	241	311
5bitadder	29	55	68	94	29	55	68	94
8bitadder	122	322	497	697	122	322	497	697
3_17	10	21	14	33	6	12	8	18
6sym	19	75	39	135	20	62	78	236
9sym	25	101	55	191	28	94	98	304
5mod5	14	58	17	67	10	90	48	204
4mod5	6	18	10	30	5	13	11	31
ham3	3	7	3	7	5	7	7	13
ham7	21	61	49	145	25	49	79	249
ham15	47	191	87	311	109	206	189	446
Xor5	4	4	4	4	4	4	4	4
Xor20	19	19	19	19	19	19	19	19
Xnor5	5	5	5	5	5	5	5	5
decod24	10	30	14	42	- -	- -	- -	- -
Cycle10_2	180	860	306	1238	19	1198	199	1738
Cycle17_3	920	4160	- -	- -	48	6057	- -	- -
Graycode6	5	5	5	5	5	5	5	5
Graycode10	9	9	9	9	9		9	9
Graycode20	19	19	19	19	19	19	19	19
Nth_prime3in c	4	6	5	9	4	6	6	12
Nth_prime4in c	16	48	20	60	12	58	18	76
Nth_prime5in c	29	91	39	121	26	78	128	384
Nth_prime6in c	148	586	- -	- -	55	667	- -	- -
Alu	5	17	7	23	- -	- -	- -	- -
4_49	16	52	41	127	16	58	40	130
hwb4	12	28	15	40	17	63	39	129
hwb5	24	96	44	156	24	104	64	224
hwb6	32	128	72	248	42	140	144	446
hwb7	49	185	129	425	35	203	- -	- -
pprm1	9	33	27	87	- -	- -	- -	- -
pprm2	55	235	90	370	- -	- -	- -	- -
pprm3	29	540	73	669	- -	- -	- -	- -

Table 2 compares QULASYN with other methods for the LNN Model. Column 1 is the name of benchmark, column 2 is the number of gates calculated for the standard model, and column 3 is the quantum cost for the standard model. Column 4 is the number of gates after insertion of SWAP gates to the lattice circuit. Column 5 is the respective quantum cost with SWAP gates inserted to lattice. Columns 6 to 9 give respective results for MMD. The results for MMD method were recalculated by inserting the necessary SWAP gates (algorithms to insert SWAP gates are given in [20, 18] and other papers). To compare thus quantum costs of Lattice method with MMD one has to compare columns 5 and 9. Bold italic numbers should help

the comparison.The new method is better in 14 cases, and worse in 3 cases. In 4 cases the quantum costs are the same. In some functions like Ham7, rd53, and hwb6 the improvements of our method for LNN model cost are dramatic. The reason that modified MMD is better in some instances is perhaps caused on the fact that our tool is not finding the optimal order of variables in lattice, but this should be an area of further research.

5 Conclusions

We presented a new synthesis method of permutative quantum circuits with two quantum cost functions: standard and LNN model. Tables 1 and 2 demonstrate strong improvements that are brought by our method in both variants. It should be however remembered that our method increases the number of ancilla qubits, so the same criticism can apply to it as to other algorithms that introduce ancilla qubits. The numbers of these ancilla qubits can be found in Table 1. We do not claim in this paper to replace the standard quantum costs with the LNN model, we advocate only to create CAD tools that will use several technology-related quantum costs.

One of the most interesting aspects of the presented approach is the natural consequence of reducing the number of SWAP gates by simply mapping the reversible circuit on a lattice. This means that simply representing the reversible circuit in the lattice has for consequence of mapping the circuit in the physical space in such a manner that optimizes the reversible layout with respect to the LNN model. This approach will be more explored in the future extensions of this work with respect to the presented lattice as well as with respect to other structures.

References

[1] A. Mischenko and M. Perkowski, "Logic synthesis of reversible wave cascades," in *Proceedings of IWLS*, pp. 197–202, 2002.

[2] A. Khlopotine, M. Perkowski, and P. Kerntopf, "Reversible logic synthesis by gate composition," in *Proceedings of IWLS*, pp. 261–266, 2002.

[3] D. M. Miller, D. Maslov, and G. W. Dueck, "Synthesis of quantum multiple-valued circuits," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 12, no. 5-6, pp. 431–450, 2006.

[4] G. Yang, W. Hung, X. Song, and M. Perkowski, "Majority-based reversible logic gates," *Theoretical Computer Science*, vol. 334, no. 1-3, 2005.

[5] W. N. N. Hung, X. Song, G. Yang, J. Yang, and M. Perkowski, "Optimal synthesis of multiple output boolean functions using a set of quantum gates by symbolic reachability analysis," *IEEE Transaction on Computer-Aided Design of Integrated Circuits and systems*, vol. 25, no. 9, pp. 1652–1663, 2006.

[6] N. Alhagi, M. Hawash, and M. Perkowski, " Synthesis of Reversible Circuits for Large Reversible Functions" *Facta Universitais, Series: Electronics and Energetics*, vol. 24, no. 3, pp. 273–289, December 2010.

[7] J. Fiurášek, "Linear optical fredkin gate based on partial-swap gate," *Phys. Rev. A*, vol. 78, p. 032317, Sep 2008.

[8] Y.-X. Gong, G.-C. Guo, and T. C. Ralph, "Methods for a linear optical quantum fredkin gate," *Phys. Rev. A*, vol. 78, p. 012305, Jul 2008.

[9] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[10] D. Chang, D. Maslov, and S. Severini, "Translation techniques between quantum circuits architecture." http://www.iqc.ca/severin/qipabs.pdf.

[11] J. Cirac and P. Zoller, "Quantum computation with cold trapped ions," *Physical Review letters*, vol. 74, no. 20, p. 4091, 1995.

[12] J. L. O'Brien, G. J. Pryde, A. G. White, T. C. Ralph, and D. Branning, "Demonstration of an all-optical quantum controlled-not gate," *Nature*, vol. 426, pp. 264–267, 2004.

[13] D. E. Browne and T. Rudolph, "Resource-efficient linear optical quantum computation," *Phys. Rev. Lett.*, vol. 95, p. 010501, Jun 2005.

[14] A. Fowler, S. Devitt, and L. Hollenberg, "Implementation of shor's algorithm on a linear nearest neighbor qubit array," *Quantum Information and Computation*, vol. 4, no. 4, pp. 237–251, 2004.

[15] Y. Takahashi, N. Kunihiro, and K. Ohta, "The quantum fourier transform on a linear nearest neighbor architecture," *Quantum Information and Quantum Computation*, vol. 7, no. 4, pp. 383–391, 2007.

[16] S. Cuccaro, T. Draper, S. Kutin, and D. Moutlon, "A new quantum ripple-carry addition circuit." quant-ph/0410184, 2004.

[17] V. V. Shende, S. S. Bullock, and I. L. Markov, "A practical top-down approach to quantum circuit synthesis," in *Proceedings of Asia Pacific DAC*, 2005.

[18] A. Chakrabarti and S. Sur-Kolay, "Nearest neighbor based synthesis of quantum boolean circuits," *Engineering Letters*, vol. 15:2, p. 26, 2007.

[19] A. Chakrabarti and S. Sur-Kolay, "Rules for synthesizing quantum boolean circuits using minimized nearest-neighborhood templates," in *Proceedings of the 15th International Conference on Advanced Computing and Communications*, pp. 183–189, 2007.

[20] Y. Hirata, M. Nakanishi, and S. Yamashita, "An efficient method to convert arbitrary quantum circuits to ones on a linear nearest neighbor architecture," in *3rd International conference on Quantum, Nano, and Micro Technologies*, 2009.

[21] M. Mozammel H A Khan, N. Siddika, and M. Perkowski, "Minimization of Quaternary Galois Field Sum of Products Expression for Multi-Output Quaternary Logic Function using Quaternary Galois Field Decision Diagram," in *Proceedings of the International Symposium on Multiple-Valued Logic 2008*, 2008.

[22] M. Mottonen and J. Vartiainen, "Decomposition of general quantum gates," in *Trends in Quantum computing Research*, NOVA, New York, 2006.

[23] P. Mudbhari, "General logic linear nearest neighbor (llnn) architecture for fault-tolerant quantum computation," Master's thesis, Wichita State University, 2008.

[24] R. Wille, M. Saeedi, and R. Drechsler, "Synthesis of reversible functions beyond gate count and quantum cost," in *Proceedings of the IWLS*, 2009.

[25] D. Wineland and T. Heinrichs, "Ion trap approaches to quantum information processing and quantum computing," *A Quantum Information Science and Technology Roadmap*, vol. N/A, 2004.

[26] D. Maslov and G. Dueck, "Improved quantum cost for n-bit Toffoli gates," *IEE Electronic Letters*, vol. 39, no. 25, pp. 1790–1791, 2003.

[27] D. Maslov and G. Dueck, "Garbage in reversible design of multiple output functions," in *proceedings of the 6th International Symposium on Representations and Methodology of Future Computing Technologies*, pp. 162–170, 2003.

[28] M. Lukac, M. Perkowski, H. Goi, M. Pivtoraiko, C. H. Yu, K. Chung, H. Jee, B.-G. Kim, and Y.-D. Kim, "Evolutionary approach to quantum reversible circuit synthesis," *Artif. Intell. Review.*, vol. 20(3-4), pp. 361–417, 2003.

[29] D. Maslov, G. W. Dueck, and D. M. Miller, "Synthesis of Fredkin-Toffoli reversible networks," *IEEE Transactions on VLSI*, vol. 13, no. 6, pp. 765–769, 2005.

[30] A. Agrawal and N. Jha, "Synthesis of reversible logic," in *Proceedings of DATE*, pp. 710–722, 2004.

[31] R. Wille and R. Drechsler, "Bdd-based synthesis of reversible logic for large functions," in *Proceedings of the 46th Annual Design Conference*, 2009.

[32] M. Lukac, M. Perkowski, and M. Kameyama, "Evolutionary quantum logic synthesis of boolean reversible logic circuits embedded in ternary quantum space using structural restrictions," in *Proceedings of the WCCI 2010*, 2010.

[33] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and W. H., "Elementary gates for quantum computation," *Physical Review A*, vol. 52, pp. 3457–3467, 1995.

[34] M. Perkowski, L. Jozwiak, and R. Drechsler, "A canonical and/exor form that includes both the generalized reed-muller forms and kronecker reed-muller forms," in *Proc. of the Reed-Muller 1997 Conference*, pp. 219–233, 1997.

[35] m. Chrzanowska-Jeske, Z. Wang, and Y. Xu, "Regular representation for mapping to fine-grain, locally-connected fpgas," in *Proceedings of ISCAS*, 1997.

[36] D. Shah and M. Perkowski, "Synthesis of quantum arrays with low quantum costs from kronecker functional lattice diagrams," in *IEEE Congress on Evolutionary Computation*, pp. 1–7, 2010.

[37] M. Perkowski, M. Chrzanowska-Jeske, and Y. Xu, "Lattice diagrams using reed-muller logic," in *Proceedings of RM*, 1997.

[38] M. Perkowski, A. Al-Rabadi, P. Kerntopf, A. Buller, M. Chrzanowska-Jeske, A. Mishchenko, M. M. Azad Khan, A. Coppola, S. Yanushkevich, V. Shmerko, and L. Jozwiak, "A general decomposition for reversible logic," in *Proc. RM'2001*, August 2001.

[39] M. Perkowski, P. Kerntopf, A. Buller, M. Chrzanowska-Jeske, A. Mischenko, X. Song, A. Al-Rabadi, L. Jozwiak, A. Coppola, and B. Massey, "Regular realization of symmetric functions using reversible logic," in *Proceedings of EUROMICRO Symposium on Digital System*, 2001.

[40] S. Akers, "A rectangular logic array," *IEEE Transactions on Computers*, vol. C-21, pp. 848–857, 1972.

[41] M. A. Perkowski, M. Chrzanowska-Jeske, and Y. Xu, "Lattice diagrams using reed-muller logic," in *IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design*, pp. 85–102, 1997.

[42] M. Chrzanowska-Jeske, Z. Wang, and Y. Xu, "A regular representation for mapping to fine-grain, locally-connected fpgas," in *Circuits and Systems, 1997. ISCAS '97.*,

Proceedings of 1997 IEEE International Symposium on, vol. 4, pp. 2749 –2752 vol.4, June 1997.

[43] S. Lee, S. Kim, J. Biamonte, and M. Perkowski, "The cost of quantum gate primitives," *Journal of Multiple-Valued Logic and Soft Computing*, vol. 12, no. 5-6, pp. 561–574, 2006.

[44] M. Lukac, "Variable ordering using ga for lattice diagrams and y gates circuits." technical report, 2010.