INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2024, VOL. 70, NO. 2, PP. 501-508
Manuscript received May 13, 2024; revised June, 2024. doi: 10.24425/ijet.2024.149572

Designing of Reversible Functions in Classical and Quantum Domains

Andrzej Skorupski, and Ryszard Romaniuk

Abstract—**First sections of the paper contain some considerations relevant to the reversibility of quantum gates. The Solovay-Kitayev theorem shows that using proper set of quantum gates one can build a quantum version of the non-deterministic Turing machine. On the other hand the Gottesmann-Knill theorem shows the possibility to simulate the quantum machine consisting of only Clifford/Pauli group of gates. This paper presents also an original method of designing the reversible functions. This method is intended for the most popular gate set with three types of gates CNT (Control, NOT and Toffoli). The presented algorithm leads to cascade with minimal number CNT gates. This solution is called optimal reversible circuits. The paper is organized as follows. Section 5 recalls basic concepts of reversible logic. Section 6 contain short description of CNT set of the reversible gates. In Section 7 is presented form of result of designing as the cascade of gates. Section 8 describes the algorithm and section 9 simple example.**

Keywords—**reversible logic; reversible circuits; reversible gate; CNT set of the gates**

I. Reversibility Transfer to Quantum Domain

DESIGNING of reversibility functions, as presented below, is a basic requirement for quantum gates and circuits [23]. The asked questions concerning the transfer between classical and quantum domains are: what are similarities and differences, how reversibility transfers between the domains, how to manage non-reversibility in quantum, simple and complex gate equivalence, gate synthesis and decomposition, gate redundance and ancilla, gate agnostics to technology, Hermitian self-inverse and non-self-inverse gates, skew-Hermitian, extension of classical universal CNT set to quantum, Pauli/Clifford quantum gates, non-Clifford gates, etc. The questions finally aim at realization of reversible unitary quantum gates and circuits which cannot be simulated by classical computation [24].

All quantum gates are reversible. The mapping of a gate's input to its output is a bijection. All multi-gate systems and scalar and Kronecker products of unitary gates are unitary and reversible. Quantum logic functions can be synthesized as complex gates or sequences of gates. Irreversible functions necessary to implement a specific quantum system can be transformed into reversible ones by adding ancilla qubits to the input, output or both sides. After performing the calculations of the required function, the ancilla redundant qubits are un-calculated (reverse calculated) or left in the system. Leaving ancilla entangled qubits in the system can lead to measurement errors. A function operating on n qubits has dimension $2^n x 2^n$. A qubyte, a register of 8 qubits, is represented by a matrix of $2^8 x 2^8$ elements.

Here we take a closer look at quantum gates and ideal systems, but going a little deeper into the detailed properties of matrix gate operators and their equivalences, such as unitarity, reversibility, self-reversibility and non-self-reversibility, also known as Hermitian and non-Hermitian, indempotent, involutive, separable and non-separable operators. multi-qubit, in other words pseudo-quantity and true quantum, in other words locality and non-locality (skew Hermitian), technological agnosticity, etc. Some of these properties (in some sense treated as types of symmetry) are absolutely required for classical quantum gates (for a quantum Turing machine) such as unitarity and reversibility, and some not necessarily, such as Hermitian. Most allowed quantum gates, unitary and reversible, are non-Hermitian [25]. The more symmetry a gate has, the more fundamental it is, such as the Pauli group and some of its transformations and extensions to the Clifford group.

Quantum gates are operators that act on qubits. Qubits meet certain criteria, sometimes called the DiVincenzo criteria, to enable the gate construction of a quantum computer (or rather processor) with their help. A qubit is a well-defined two-level quantum object. Qubits are addressable, meaning it is possible to set any very simple, often fundamental, quantum state before using them to conduct functional activities. Qubits are controllable, which means that it is possible to program their quantum states with the help of dedicated external interactions. Qubits are decoherently isolated from unwanted interactions, i.e. they have a decoherence time long enough for computational needs. Qubits are organizable, which means they can be set into various types of registers containing many separate qubits.

Quantum gates build the ideal logical structure of a quantum computer, provided they form a sufficient group, called a universal set. Not all quantum gates form universal sets. The significant natural symmetry of a particular quantum gate allows for a greater number of equivalences with related gates. The symmetries of individual gates in a quantum system have a significant impact on the final properties of the system, e.g. the possibility of grouping basic gates into logical qubits and higher order logic gates, resistance to decoherence, resistance to the generation and uncontrolled multiplication of

The authors are from Warsaw University of Technology, Institute of Computer Science, and Institute of Electronic Systems, Poland (e-mail: Andrzej.Skorupski@pw.edu.pl, Ryszard.Romaniuk@pw.edu.pl).

errors, etc. Some gate symmetries facilitate the multiplication of quantum errors , hence the systems use a mixture of gates with various degrees of symmetry that have preventive features. Quantum gates also differ in the type of quantum resources they build in the form of superposition and entanglement. Superposition and entanglement can be stronger or weaker. Some gates reach maximum values of quantum resources, such as those determined by Bell states. Different types of entanglement may be transferred or not transferred between each other, e.g. between W and GHZ gates. Quantum gates may be more or less resistant to decoherence. A quantum system cannot be built from a completely arbitrary set of gates. It must contain a minimum universal set, or only such a mandatory set may be supplemented with permitted arbitrary goals.

What is the most important gate for quantum computing? Those using public quantum clouds know this well. At the beginning of a quantum system, which we read from left to right and which connects the gates with quantum wires representing qubits, it is necessary to set the input state. It is good to have as much quantum resources as possible from the beginning in the form of coherence and entanglement. The immediate choice is, for example, parallel one-qubit Hadamard gates that set the same ket probability on all required qubit lines, and then two-qubit quantum XOR (CX) gates that strongly entangle two selected lines to the two-qubit Bell state.

The simplest system for obtaining a strongly and uniformly entangled Bell state consists of one Hadamard gate on the control line and then a CX gate. Such a system of gates in the signal line encodes a qubit from the base state to a qubit with quantum resources of superposition and real entanglement, or, in the reverse order, decodes the encoded qubit to the initial or simpler quantum state. At the same time, such a system can correct quantum errors arising during processing. Using a bit of quantum programming lingo, we can also say that the Hadamard gate is very compatible with the S phase gate (or its inverse S†), which is an extension of the H gate in order to obtain complex superposition. The S gate maps the X→Y axes and is a rotation about the Z axis by π/2. For the inverse S-1=S†, due to Hermitian self-adjoint, the mapping is X→-Y and the rotation is -π/2.

A reversible XOR gate is a individually controlled NOT gate, also known as CNOT, or a controlled Pauli gate CX, with a fixed rotation around the X axis by an angle π in the unitary Bloch sphere. The CX gate generates and manipulates entanglement, and is used very often in various computational systems, as well as, for example, quantum error correction systems. The CX gate combined with single-qubit gates creates the simplest universal set, one of the possible universal sets. In such a universal set, it plays a fundamental, irreplaceable role. In Toffoli notation, the top a qubit is the source (or control) and the bottom b is the purpose (signal). The CX gate transforms the base state of the two-qubit input ket into the output |a,b>→|a,a⊕b>, where ⊕ is the XOR operation. When the input qubits are not in a state of superposition between |0> and |1>, i.e. they behave like classical bits, the quantum XOR gate has a truth table identical to the truth table of the classical I/O gate: 00-00, 01-11,10 -10,11-01. The 4x4 matrix gate operator CX=[1,0,0,0/0,0,0,1/0,0,1,0/0,1,0,0] acts on a columnar four-amplitude state vector [00,01, 10,11] giving the output state vector [00,11,10,01].

If the control qubit of the CX gate is not in the base state, but in the uniform superposed state |+>=(1/√2)(|0>+|1>), i.e. after passing through the H|0> gate, then at the gate input CX we have, for example, the state |0>⊗|+>=|0+>=(1/√2)(|00>+|01>). This state, after passing through the CX gate, gives an entangled state, one of the Bell states: CX|0+>=(1/√2)(|00>+|11>). The probability of states after measurement |00> and |11> is 50%, and the states |01> and |10> are 0%. This entangled quantum state is not separable.

Unitary multi-qubit gates have operators in the form of frame matrices, where some frames are functional and the remaining parts are permutational, e.g. in multiply controlled and other complex gates. The single-qubit gate G, i.e. the unitary operator bra, affects qubit k written in the form of ket <G|k>. The two-qubit gate G2 acts on two qubits written in the form of a ket of the product of these qubits <G2|k1k2>. In the equivalent matrix form, it is the scalar product of a matrix unitary gate operator of dimensions 2x2 and 4x4 respectively and a column vector of one qubit and two qubits. Among two-qubit (also multi-qubit) gates, we distinguish separable (separable) into two qubits (different groups of qubits) and non-separable (not separable), due to the sufficiently strong entanglement of the qubits.

II. Unitary, reversible operators in Quantum

A quantum gate is a unitary linear transformation of U. There are uncountable many quantum gates. Basic quantum gates are described by very simple unitary matrices (operators) U. The determinant of a unitary matrix is a complex number with norm 1. Unitarity ensures that if U exists, then U-1 also exists, so quantum gates are always reversible. Unitarity requires reversibility, but does not require self-reversibility, i.e. the $U=U^{-1}$ condition. So quantum gates can be self-reversible (Hermitian) and non-self-reversible (non-Hermitian). The unitary operator U satisfies the condition: $U^{-1}=U^{\dagger}$. The Hermitian operator h satisfies the condition: $h=h^{\dagger}$. Hence it follows that not all unitary operators are Hermitian, and not all Hermitian operators are unitary. If the unitary operator U is Hermitian then $U^{-1}=U^{\dagger}=U$, or $U=U^{-1}$is self-reversible. CX gate is self-reversible and therefore Hermitian.

Among the classical gates, there is a group of reversible gates, which in a sense precede quantum systems and belong to one classical-quantum group of reversible gates. Classical reversible gates are essentially permutation gates. Quantum gates are unitary operators with respect to a certain basis in Hilbert space. Hilbert space is a linear, unitary, complete space, here with respect to qubits, over the field of complex numbers. A Hilbert space is a metric space with the metric given by the dot product. The basis in linear space is an extension of the idea of the Cartesian coordinate system in Euclidean space. In relation to qubits, a computational basis is used, referring to a certain coordinate system or labelling orthogonal basis vectors, of which there are d-1 for a d-level qubit. Unitary symmetry is a type of symmetry related to the group of unitary matrices. The unitary group of degree n, denoted as U(n), is a group of unitary matrices of dimension nxn with the group operation being matrix multiplication. The unitary group U(n) is a real Lie group of dimension n2. The group of unitary matrices with determinant 1 is denoted as SU(n). In the case of a group of 2x2 quantum gates, we talk about SU(2) symmetry.

The unitary operator is a generalization of the unitary matrix. The unitary operator is a normal operator whose composition with its conjugate operator is identity. The unitary operator preserves geometric quantities (norms, metrics, angles, orthogonality, lengths) and the dot product (inner product) in the unitary space. A unitary space is a linear space with the dot product of vectors defined in it, which is alternatively denoted as <x,y>, (x,y), <·|·>, etc. The dot product is semi-linear, linear due to one and anti-linear because of the second argument. Formally, the choice is free. In the Dirac bracket notation, popularly used in quantum computing, antilinearity is assumed with respect to the first argument, i.e. bra denoting a continuous linear functional in the Hilbert complex space (and in relation to a qubit, a unitary operator, i.e. a quantum gate). Linearity is assumed towards ket, i.e. the unitary column vector denoting the quantum state of the qubit.

Not all gates are Hermitian, i.e. described by the Hermitian operator (self-adjoint, self-reversible). Most unitary gates are non-self-reversible, i.e. non-Hermitian. Hermitian gates are the basic, elementary group of quantum gates. A Hermitian (self-adjoint) matrix is a complex square matrix equal to its transposed complex conjugate. Hermitian matrices are normal matrices. The Hermitian matrix is a complex extension of the symmetric matrix. Pauli matrices are Hermitian matrices. The 3x3 Gell-Man matrices are traceless Hermitian matrices. Most of the most commonly used quantum gates such as Pauli XYZ, H,CX, SWAP, Toffoli, Fredkin are Hermitian. Most of the single-qubit and two-qubit gates used are Hermitian. Most allowed quantum gates are non-Hermitian.

As the quantum gate dimension increases, fewer and fewer gates are Hermitian. Gate hermiticity plays a significant role in reversible gates and quantum systems. In classical logic systems, hermiticity does not play such a role and most of the classical gates used in practice are non-Hermitian, even in reversible systems. Square Hermitian matrices are unitary diagonalizable with real eigenvalues. All square unitary and Hermitian matrices are normal matrices. The normality of the U matrix is equivalent to the existence of an orthonormal basis in which PUP^{-1} is a diagonal matrix and P is a transformation matrix to the orthonormal basis. A generalization of a normal matrix is a normal, linear operator, limited in Hilbert space, which commutates (is commutative) with its conjugate. Any normal matrix can be diagonalized (spectral theorem). These properties of operators translate into the characteristics of quantum gates.

The unitary operator is a surjective bounded linear operator and represents an isomorphic linear transformation between topological normalized Hilbert vector spaces. The product of the unitary operator with its Hermitian conjugate is commutative and is an identity operator. The unitary operator is both isometry and coisometry, or in other words surjective isometry. Unitary operators are automorphisms of Hilbert spaces. The Hermitian coupling of a bounded linear operator is involutional, reversible in the case of operator reversibility, isomorphic in the case of operator isomorphicity, injective in the case of surjectivity of the operator, dense in the case of operator injectivity, antilinear, and anti-separable.

Hermitian operators build observables in quantum computing in terms of real eigenvalues. The sum of Hermitian operators is a Hermitian operator. The product of the Hermitian operator by a real number is the Hermitian operator. Every Hermitian operator defined on the entire Hilbert space is bounded. Hermitian operators unconstrained as observables, e.g. operators of physical quantities, are defined on some Hilbert subspace. The properties of the unitary and Hermitian operators collected and recalled here translate into the properties of quantum gates and their behavior in quantum systems.

A special place in quantum computing is occupied by the Hermitian operator and the 2x2 one-qubit Hadamard gate with a unitary and involutional matrix analogous to Pauli gates $H=(1/\sqrt{2})[1,1/1,-1]$ and having no classical reference, and therefore not having truth table. This is a 'pure' quantum gate. In general, the Hadamard matrix is an nxn square matrix filled only with ±1 numbers, in which the rows are mutually orthogonal. Each pair of rows represents perpendicular vectors. Finding the H matrix of significant dimensions is a computationally difficult problem. A single-qubit H gate maps the underlying starting kets |0> and |1> into a uniform superposition of states |0> and |1> with equal probability $(1/\sqrt{2})(|0>\pm|1>)$. Such states are marked as kets |+> and |->, and then the gate H is written as the sum of the inner products ket and bra H=|+><0|+|-><1|. The H gate rotates the qubit state by an angle of π radians with respect to the diagonal axis in the x-z plane, i.e. $(x+z)/\sqrt{2}$, which is equivalent to assembling a Pauli gate X and then rotating by π/2 with respect to the Y axis, therefore the following conditions are satisfied the following identities decompositions: identity HH=I, $H=XR_y(\pi/2)=X\sqrt{Y}$, and $H=R_y(\pi/2)Z=\sqrt{Y}Z$, $H=Z/\sqrt{Y}=\sqrt{}^{-1}YX$. The H gate can be treated as a unitary transformation mapping qubit operations in the Z axis to the X axis and vice versa, therefore the following dependencies, decompositions, factorizations are met: X=HRZ, $S=\sqrt{Z}=H\sqrt{X}H$, $R_Z(\varphi)=HR_Z(\varphi)H$. The Hadamard gate can be represented, factorized using a phase gate π/2, S=P=P(π/2) and a rotation gate R_x relative to the X axis by an angle π/2 as follows: $H=PR_x(-\pi/2)P$.

Clifford quantum gates are elements of the broader Clifford group, which is the set of discussed mathematical transformations of Pauli operators, in the form of permutations and other unitarity-preserving transformations. The Clifford group contains single-qubit Pauli gates, their unitary transformations, arbitrary rotations, phase shifts, two-qubit permutations in the form of controlled Pauli gates CX, CY, CZ, CS, etc. such as the controlled Hadamard gate CH, multi-qubit permutations in the form of higher-order controlled gates, e.g. .C^2X=CCX (Toffoli gate), C^nX, C^nY, C^nZ, etc. The general expression for the unitary matrix U(2) is used here, 2x2 $U=[a,b/-e^{i\varphi}b^*,e^{i\varphi}a^*]$, $|a|^2+|b|^2=1$, $\det(U)=e^{i\varphi}$,, from which some Clifford gates are recruited. If additionally det(U) of such a matrix is 1, then the matrix and the gate belong to a special unitary group, the Lie group, SU(2).

III. Multivariable Quantum Gates

Two-qubit (and multi-qubit) gates are required to handle entangled quantum states. Single-qubit gates cannot handle such states. Not all two-qubit gates result in appropriate entangled states (not classically simulated). Basic two-qubit gates include controlled gates. Controlled gates CX,CY,CZ have operators in the form of frame matrices and contain an operator frame, e.g. X,Y.Z, and a permutation part. Control can be extended to any number of qubits in the gate. The control function of a quantum gate plays a different role when analyzed from the point of view of measuring the quantum state. Before measurement, it acts on all superposed states that meet the operating conditions of the control line. After the measurement, we assume that the control function was performed or not depending on which base state was measured. The group of controlled gates includes the controlled phase shift gate $P(\varphi)=[1,0/0,e^{i\varphi}]$ denoted as $CP(\varphi)$=CPHASE. The 4x4 CP frame matrix operator contains a P frame in the lower right corner and a permutation in the upper left corner (always relative to the standard computational basis). The CP gate shifts the phase by φ only if it acts on the |11> state. A special case of the general gate $CP(\varphi)$ is $CZ=CP(\varphi=\pi)$. Notation of the controlled gate operator is a CU with a frame matrix with U frames and a permutation $CU=[1000/0100/00u_{00}u_{01}/00u_{10}u_{11}]$.

The dot product of the gates C=XY=iZ means their series connection. Similarly, the Kroneker (tensor) product of two quantum gates means their parallel connection C=X Y. If we denote a one-qubit Hadamard gate as H, then in principle it can and should be denoted as H1, because H_2=H H is Hadamard transform. The series connection HH=I is identity. The nth-order Hadamard transform H_n=H n performs a transformation operation on a register of n-qubits. If H_n is applied to a register of n qubits initialized to |0>, i.e. H|0>= H|0>=(H)(|0>), then the quantum register is in a state of uniform superposition with equal probabilities in each of its 2^n possible quantum states. The condition of uniform superposition of quantum states of qubits at the input of a quantum gate system is required in many quantum algorithms and calculations such as amplitude amplification, quantum phase estimation, etc. Such a state is obtained and often used as the first step in many complex quantum algorithms.

Another form of converting the H gate into a two-qubit form (i.e. also supporting quantum entangled states) is H_I=H I, where I is the (quantum) identity gate (quantum wire). So it is an H gate connected in parallel with the quantum wire. The H_I gate has a 4x4 matrix operator of the form $H_I=(1/\sqrt{2})[1010/0101/10\text{-}10/010\text{-}1]$. The H_I gate can be applied to one of the Bell states $H_I[(\sqrt{}^{-1}2)(|00>+|11>]$ generating a base probability distribution of states, e.g. (1/2)[|00>+|01>+ |10>-|11>. The unitary inversion operation of the probably most popular system of input gates H_I and CX is as follows: $(CX\cdot H_I)\dagger=H_I^{\dagger}\cdot CX^{\dagger}=H_I\cdot CX$.

A SWAP gate switches two qubits. It is sometimes denoted as S, which can be easily confused with a phase gate $P(\varphi=\pi/2)=S$, unless the context of application in a specific gate system is analysed more closely. The SWAP frame operator includes the X operator placed symmetrically and permutations up and down the diagonal (of course with respect to the computational base |00>,|01>,|10>,|11>). The square root √SWAP gate performs half of the SWAP function on two qubits. The √SWAP gate is universal, i.e. together with single-qubit gates, it creates a universal set of gates from which any set of other gates, including multi-qubit gates, can be built. The √SWAP gate is not maximally entangled. Generation of the Bell state requires the use of at least two √SWAP gates. A related SWAP gate is the imaginary SWAP gate, denoted as iSWAP. The difference in the frame operator of the iSWAP gate compared to the SWAP gate is the exchange of real ones 1 in frame The iSWAP gate has its square root version √iSWAP with a cage matrix $[\sqrt{}^{-1}2,i\sqrt{}^{-1}2/i\sqrt{2},\sqrt{}^{-1}2]$. The iSWAP gate has equivalent relationships with the Ising R_{xx} and R_{yy} coupling gates.

A controlled CSWAP gate, called a Fredkin gate or abbreviated as F or CS, is a three-qubit gate and performs a controlled SWAP operation. In classical computing, it is a universal gate and has a character analogous to the quantum version, i.e. both have the same truth tables (for the base state of the quantum version). The Fredkin quantum gate has a cagey 8x8 matrix operator. A Fredkin gate maintains the number of I/O states |0> and |1>. The Fredkin gate was implemented in a photonic version and is a candidate for implementation in the form of a PIC integrated circuit. Fredkin gates are an element of the implementation of, for example, Shor's algorithm.

The Toffoli gate is a three-qubit, double-controlled NOT=X gate, i.e. Toff=CCX=CCNOT. It is equivalent to the Deutsch gate for the angle π/2, i.e. Toff=D(π/2). The Toffoli gate is universal for classical computations, but not for quantum ones. The classical and quantum Toffoli gates have the same structure, they are analogous, they both have the same truth table, except for the 8x8 framed matrix operator containing the CX frame and the permutation residue. The Toffoli gate achieves universality in combination with a single-qubit H gate. In the simplest system of input states |0> and |1>, the Toffoli gate operates on the third qubit when both control qubits are |1>.

The three-qubit unitary Deutsch gate D(φ), is a quantum generalization of the Toffoli gate, transfers the state $|a,b,c>\rightarrow i\cos(\varphi)|a,b,c>+\sin(\varphi)|a,b,1-c>$ only for a=b=1. The D gate is a universal quantum gate. Operator D is a generic U(2) 8x8 matrix with a double-controlled cage structure with the cage matrix [u11,u12/u21,u22] in the lower right corner and the rest in permutation form. The D gate, as well as other multi-qubit gates, are subject to scalar product decomposition into simpler gates, e.g. two-qubit individually controlled gates and one-qubit gates.

Two-qubit and multi-qubit gates are either separable, i.e. decomposable into single-qubit gates, or entangled and cannot be separated into single-qubit gates. Two-qubit gates that cannot be constructed as the tensor product of two one-qubit gates are entangled gates, and are called true two-qubit or true two-qubit gates. True two-qubit gates are important because only one of these gates is needed, e.g. the frequently used CX gate, which, when combined with single-qubit gates, creates a universal set of gates. Functional completeness in the Boolean logical domain is created, for example, by a set of gates: X, CX and Toffoli. Of the possible basic 24 two-qubit gates in the computational database, 4 are separable and 20 are non-separable.

IV. Multigate sets

The Clifford set alone is not a universal quantum gate set. It can be simulated according to GKT. Quantum systems consisting of Clifford gates entering the universal set, called the primitive gate set, can be efficiently simulated in polynomial time by a probabilistic classical computer, as proven by the Gottesmann-Knill GKT theorem. Other unitary operators, discussed above, that are outside the primitive set can be synthesized or approximated by combining primitive gates into a quantum system. The required additional unitary gate can be factorized into a scalar or tensor product of the available primitive gates from the universal set. The U(2n) group is the symmetry group for quantum gates acting on n qubits. Factorization is equivalent to the problem of finding a path in the U(2n) group from the set generating primitive gates.

The Solovay-Kitayev SKT theorem shows that given a universal set of primitive gates, there is an efficient approximation to every quantum gate. Formally, if a set of single-qubit quantum gates (such as Clifford) generates a dense subset of the SU(2) group, then this set fills the SU(2) group quickly, which means that any additionally required gate can be approximated by a fairly short sequence of gates generating this set. However, for large numbers of qubits in a quantum system, the problem of synthesizing any gate becomes impossible. A quantum system with n qubit gates can be approximated to an accuracy of e, in the operator norm, by a quantum system with $O(m \log^c(m/e))$ gates from the required finite set of universal gates. The SKT theorem guarantees that a quantum processor requires the implementation of a relatively small number of gates to perform virtually all quantum operations.

The GKT theorem states that stabilizer systems can be effectively simulated classically in $O(n \log n)$ time. The Clifford group is a normalizer of the Pauli qubit group. The centralizer (commutator) and the normalizer (stabilizer), as well as the idealizer, are special subgroups of a given group, having universal applications in its study. The normalizer meets weaker conditions than the centralizer. Centralizers and normalizers, as well as idealizers, also apply to Lie algebra and Lie groups (unitary group U(n)). The Clifford group can only be generated by using CX, H, S gates and stabilizer circuits can be built using these gates. The GKT theorem proves that quantum calculations based on the type of entanglement generated by stabilizer systems do not provide a computational advantage over classical calculations. This also means that even some highly entangled quantum states can be simulated classically. Many of the fundamental quantum algorithms use only Clifford gates, including entanglement distillation and quantum error correction algorithms. It is necessary to know exactly what type of entanglement we are dealing with in a specific gate system in a quantum processor [25, 27].

V. Introduction to design a reversible functions

The reversible functions are implemented by digital circuits with the same number of input and output bits (Fig. 1). These circuits could be lossless information circuits as well as reversible ones if mapping of the vector input into the vector output is mutually unambiguous.

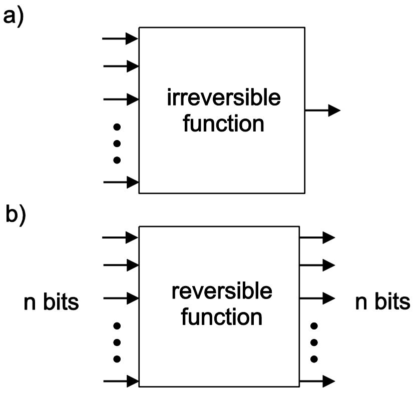


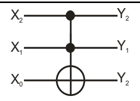
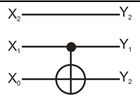
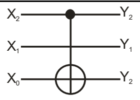
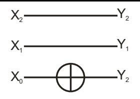
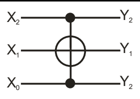
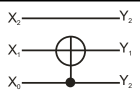
Fig. 1.a) Irreversible circuit, b) reversible circuit

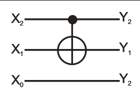
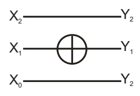
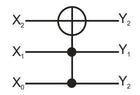
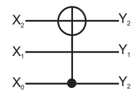
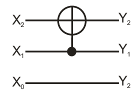
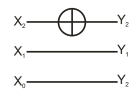
Landauer showed that the loss of information implies energy loss [1]. The result of this theorem is the possibility of construction of low energy circuits using reversible logic. The one of conditions of reversibility is a mutual unambiguity, i.e. for each input pattern is assigned different an output pattern, and vice versa [21]. Each of the Boolean functions included in the reversible function has the same number of the 0-s and 1-s minterms. The other conditions of reversibility are: no fan-outs and no feed-backs [2]. Depend of the number of variables there are different number of reversible functions. In general case there are $2^n!$ different function, where n is the number of variables. There are 24 reversible functions for 2 variables, 40320 functions for 3 variables and more than $20x10^{12}$ for 4 variables. The base of the synthesis are the types of the gates used in this procedure. There are many types of the gates as: NOT, Controlled NOT, Toffoli, Fredkin, Kerntopf gates and others [7,8,9,10]. One of the most popular set of the reversible gates is the CNT set (Controlled NOT, NOT and Toffoli gates).

VI. Reversible gates

The three variables CNT set of the gates containing 12 gates. All of them are presented in Table I where are the names of the gates, assigned number and diagrams.

TABLE I
THE CNT SET OF THREE VARIABLE REVERSIBLE GATES

Gate numbers	
T0 - 8	
C0-1 – 7	
C0-2 – 6	
N0 – 9	
T1 – 5	
C1-0 – 4	

C1-2 – 3	
N1 – a	
T2 – 2	
C2-0 – 1	
C2-1 – 0	
N2 - bb	

It easy to show the logic gate operation. The operation of each gate consists of swapping of the pairs minterms in true table. For example if on the inputs of the gate T0 is identical function than on the outputs is the function $Y_2Y_1Y_0$ presented in Table II.

TABLE II
EXAMPLE OF THE GATE T0 OPERATION

Row No.	$X_2X_1X_0$	$Y_2Y_1Y_0$
0	000	000
1	001	001
2	010	010
3	011	011
4	100	100
5	101	101
6	110	111
7	111	110

The operation of the gate T0 consists of swapping of the pairs minterms 6 with 7. In this case $Y_2=X_2$, $Y_1=X_1$, $Y_0=X_0 \oplus X_2 \cdot X_1$.

TABLE III
SWAPPED MINTERMS FOR REVERSIBLE GATES

Gate	Swapped rows/values
T0	6,7
C0-1	2,3 & 6,7
C0-2	4,5 & 6,7
N0	0,1 & 2,3 & 4,5 & 6,7
T1	5,7
C1-0	1,3 & 5,7
C1-2	4,6 & 5,7
N1	0,2 & 1,3 & 4,6 & 5,7
T2	3,7
C2-0	1,5 & 3,7
C2-1	2,6 & 3,7
N2	0,4 & 1,5 & 2,6 & 3,7

In Table III are presented operations (the pairs of minterm swapped by proper gate) for all gates. As we can see each gate swap minterm 7. The gates with XOR on line X_2 swap minterm 7 with minterm 3. The gates with XOR on line X_1 swap minterm 7 with minterm 5. The gates with XOR on line X_0 swap minterm 7 with minterm 6.

VII. REVERSIBLE CIRCUITS

To implement any reversible function the designer must find the string of the reversible gates which transform the identical function I into given function F. The solution of this problem is a cascade of reversible gates (Fig. 2) containing a minimal number of gates.

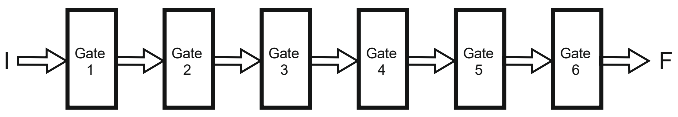


Fig. 2. Cascade with 6 reversible gates

Usually exist more then one solution of this problem. To implement the best solution of the synthesis other criteria need to be added. In this paper we miss it.

VIII. ALGORITHM

The algorithm of the synthesis the given reversible function consists in searching for subsequent gates that replace appropriate rows. In each step are selected the gates swapping these rows. The end of algorithm is when on the output of the gate in subsequent step appear the given function.

1. The indictor of the appropriate rows which must be swapped is the XOR function $S_i = Y_i \oplus X_i$. This function contain value 1 if the bits Y_i and X_i in any row are different. Than this row must be swaps.
2. On each line X_i must be selected the gate (gates) which swapped the minterms indicated by function S_i.
3. For each selected gates calculate the output functions and repeat points 1 and 2.
4. The algorithm terminate when all $S_i = 0$.

IX. EXAMPLE FOR 3 VARIABLE FUNCTION

Let be given the reversible function specified by true table from Table IV.

TABLE IV
EXAMPLE OF THREE VARIABLE REVERSIBLE FUNCTION

No.	$X_2X_1X_0$	$Y_2Y_1Y_0$
0	000	000
1	001	100
2	010	010
3	011	001
4	100	110
5	101	011
6	110	111
7	111	101

Due to first point of algorithm we designate the three functions Y_2, Y_1 and Y_0. ,

$$Y_2 = X_2 \oplus X_0 \oplus X_1 X_0$$
$$Y_1 = X_2 \oplus X_1 \oplus X_1 X_0 \oplus X_2 X_1$$
$$Y_0 = X_2 X_0 \oplus X_2 X_1 \oplus X_1 X_0$$

1. Now is easy to designated three functions S_i.

$$S_2 = X_0 \oplus X_1 X_0,\ S_1 = X_2 \oplus X_1 X_0 \oplus X_2 X_1$$
$$S_0 = X_2 X_0 \oplus X_2 X_1 \oplus X_1 X_0$$

2. The function S_2 has ones in rows 1 and 5. To swap these minterms using gate with XOR on line X_2 is needed gate C2-0.
   The function S_1 has ones in rows 3, 4, 5 and 7. To swap minterms 3 and 4 using gate with XOR on line X_1 is needed gate C1-0 or C1-2.
   The function S_0 has ones in rows 1 and 6. To swap minterm 1 using gate with XOR on line X_0 is needed gate N0. To swap minterm 6 using gate with XOR on line X_0 is needed gate T0.

These 5 functions are the result of first step of our algorithm. For each of these cases we need to execute the second step. In order to simplify our considerations we select only the gate C1-0 and it output function F1 for second step.

1. The function S_2 has ones in rows 3 and 7. To swap these minterms using gate with XOR on line X_2 is needed gate T2.
   The function S_1 has ones in rows 3 and 4. To swap these minterm the gates using gate with XOR on line X_1 is needed gate C1-2 or C2-0.
   The function S_0 has ones in rows 3 and 6. To swap these minterms using gate with XOR on line X_0 is needed gate C2-0.
2. In second step for first selected gate C1-0 exist four gates needed for swapping appropriate rows. It easy to see that on the line X_2 the gate T2 ordering function Y_2 (the function $S_2 = 0$).

The double gates string (C1-0 i T2) has on the output function F2 for the third step..

1. The function $S_2 = 0$.
   The function S_1 has ones in rows 4 and 7. To swap these minterm the gates using gate with XOR on line X_1 is needed gate C1-2.
   The function S_0 has ones in rows 6 and 7. To swap these minterms using gate with XOR on line X_0 is needed gate T0.
2. In third step we use the gate T0 ordering function Y_0.

In fourth step:

1. The function $S_2 = 0$.
   The function S_1 has ones in rows 4 and 6. To swap these minterm the gates using gate with XOR on line X_1 is needed gate C1-2.
   The function $S_0 = 0$.
2. In third step we use the gate T0 ordering function Y_0.

In fifth step:

1. The function $S_2 = 0$.
   The function S_1 has ones in rows 6 and 7. To swap these minterm the gates using gate with XOR on line X_1 is needed gate T1.
   The function $S_0 = 0$.
2. In third step we use the gate T1 ordering function Y_0.
   This step terminate the algorithm. The result of synthesis is show in the Fig. 3.

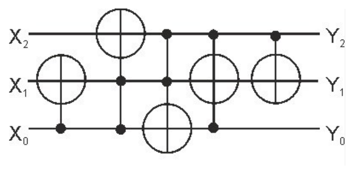


Fig. 3. The solution of the algorithm using 5 reversible gates

X. CONCLUSIONS

The most important issue in the area of comparisons between classical and quantum computations is probably the search for a reduction in the required computing resources between both classes of computers with a logical gate architecture in the computable area. In the non-deterministic polynomial subclass of NP, strict evidence for resource reduction is sought, usually expressed in asymptotic form and the capital O notation, i.e. $NP=O(n^x)$. In the exponential subclass, e.g. $EXPTIME=O(X^n)$, for conventional calculations, problems reduced to NP in the quantum class are sought. The search is to find suitable quantum algorithms and prove asymptotic reduction in the demand for functional resources. Reductions of this type found arouse a lot of interest and are one of the main driving forces behind the development of quantum computing. The practical implementation of the most famous of them in the form of Shor, Groover, HHL algorithms, etc. is still a promise of the future and requires quantum computers with millions of logic gates.

The other aim of this paper is to present the design of optimal reversible cascade which enables implementation of the given reversible functions. The presented examples illustrate the algorithm for the synthesis of the reversible functions of the three variables. Was showed how to find the reversible function implemented by the given cascade. This algorithm is scalable for more variables. In this case presented algorithm could be aided by computer program.

REFERENCES

[1] R. Landauer, Irreversibility and heat generation in the computing process. IBM Journal of Research and Development , 5(3):183–191, July 1961. https://doi.org/10.1147/rd.53.0183

[2] M. Nielsen, I. Chuang, Quantum Computation and Quantum Information. Cambridge University Press, 2000. https://doi.org/10.1017/CBO9780511976667

[3] B. Desoete, A. De Vos, M. Sibinski, T. Widerski, Feynman's reversible gates implemented in silicon, 6th International Conference MIXDES, pages 496–502, 1999.

[4] M. Veldhorst, C. H. Yang, J. C. C. Hwang, W. Huang, J. P. Dehollain, J. T. Muhonen, S. Simmons, A. Laucht, F. E. Hudson, K. M. Itoh, A. Morello, A. S. Dzurak, A two-qubit logic gate in silicon, Nature, 526, 410–414, October 2015

[5] P. Picton, Opoelectronic, multivalued, conservative logic, International Journal of Optical Computing, 2:19–29, 1991.

[6] R. C. Merkle, K. E. Drexler, Helical logic, Nanotechnology, 7:325–339, 1996. https://doi.org/10.1088/0957-4484/7/4/004

[7] E. Fredkin T. Toffoli. Conservative logic. International Journal of Theoretical Physics, 21:219–253, 1982.

[8] R. Feynman. Quantum mechanical computers. Optic News, 11:11–20, 1985.

[9] T. Toffoli. Reversible computing. Tech memo MIT/LCS/TM-151, MIT Lab for Comp. Sci, 1980.

[10] P. Kerntopf, Maximally efficient binary and multi-valued reversible gates, International Workshop on Post-Binary ULSI Systems, pp. 55–58, Warsaw, Poland, May 2001.

[11] K. Iwama, Y. Kambayashi, S. Yamashita, Transformation rules for designing CNOT-based quantum circuits, Design Automation

Conference, New Orleans, Louisiana, USA, June 10-14 2002. https://doi.org/10.1109/DAC.2002.1012662

[12] D. M. Miller, D. Maslov, W. Dueck, A transformation based algorithm for reversible logic synthesis, Proceedings of the Design Automation Conference, pages 318–323, June 2003. https://doi.org/10.1145/775832.775915

[13] K. Fazel, M. A. Thornton, J. E. Rice, ESOP-based Toffoli Gate Cascade Generation, Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp. 206 –209, 2007. https://doi.org/10.1109/PACRIM.2007.4313212

[14] M. H. A. Khan, M. A. Perkowski, Multi-output ESOP Synthesis with Cascades of New Reversible Gate Family, Proc Int. Symp. On Representations and Methodology of Future Comp. Technology, pp.43-55,2003.

[15] R. Wille, R. Drechsler, BDD-based synthesis of reversible logic for large functions, Design Automation Conf. , pp. 270–275, 2009. https://doi.org/10.1145/1629911.1629984

[16] M. Hawash, M. Perkowski, N. Alhagi, Synthesis of Reversible Circuits with No Ancilla Bits for Large Reversible Functions, Proc. ISMVL, 2010, p. 1-7. https://doi.org/10.1109/ISMVL.2010.16

[17] D. Wang, S. Sun, H. Chen, Matrix-based algorithm for 4-qubit reversible logic circuits synthesis, Energy Procedia, vol. 13, pp. 365-371, 2011. https://doi.org/10.1016/j.egypro.2011.11.052

[18] O. Golubitsky, D. Maslov, A study of optimal 4-bit reversible Toffoli circuits and their synthesis, IEEE Transactions on Computers, vol. 61, no. 9, 2012,. pp. 1341-1353. https://doi.org/10.1109/TC.2011.144

[19] A. Khlopotine, M. Perkowski, P. Kerntopf, Reversible logic synthesis by iterative compositions, International Workshop on Logic Synthesis*,* 2002.

[20] M. Soeken, N. Abdessaied, G. De Micheli, Enumeration of reversible functions and its application to circuit complexity, Conference on Reversible Computation, 2016, 255–270.

[21] P. Gupta, A. Agrawal, N. K. Jha. An algorithm for synthesis of reversible logic circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, pp. 2317-2330, 2006. https://doi.org/10.1109/TCAD.2006.871622

[22] P. Kerntopf. A new heuristic algorithm for reversible logic synthesis. *ACM/*IEEE *DAC*, pages 834-837, 2004. https://doi.org/10.1145/996566.996789

[23] A.Barenco, et al., 1995, Elementary gates for quantum computation, https://doi.org/10.48550/arXiv.quant-ph/9503016

[24] R.Romaniuk, 2021, Quantum gates, Elektronika 62(12): 17-25, https://doi.org/10.15199?13.2021.4 (in Polish)

[25] A.Pathak, 2013, Non-Hermitian quantum gates are more common than Hermitian quantum gates. https://doi.org/10.48550/arXiv.1309.4037

[26] D.Deutsch, P.Hayden, 1999, Information flow in entangled quantum systems, https://doi.org/10.48550/arXiv.quant-ph/9906007

[27] D.Gross, et al., 2009, Most quantum states are too entangled to be useful as computational resources, https://doi.org/10.48550/arXiv.0810.4331