Qubit Minimization of Boolean Functions

Vladimir Hahanov, Ka Lok Man, Mykhailo Liubarskyi, Ivan Hahanov, Svetlana Chumachenko

***Abstract*— The memory-driven innovative architecture of logic-free quantum computing is presented, which is characterized by the use of photon read-write transactions in the structure of electrons associated with superposition and entanglement of states. The quantum methods for parallel minimization of Boolean functions and solving the coverage problem based on the use of qubit data structures are proposed.**

***Index Terms*— Boolean function minimization, cloud-driven computing, quantum memory-driven computing, minimum coverage.**

I. Innovation for the Architecture of Quantum Computing

The physical basis of classical quantum computing (Fig. 1) is the leverage of superposition and entanglement operations over electron states, which are quite sufficient for the computational process [1-3]. The electron performs a memory function for storing a bit of information. The low and high electron orbits correspond to the values of zero and one. A functionally complete basis for creating a quantum computing architecture is represented by the operations of superposition and entanglement, which can be put into correspondence with the traditional logical basis or-not. In set theory, an isomorphism in the form of a "union-complement" pair is put in the given basis.

The superposition operation in quantum physics is isomorphic to logical inversion or set-theoretic complement in the algebra of logic. Therefore, it is natural that each state of a binary digit (an electron), after applying this operation to a bit of information, knows everything about each other wherever they are, up to the inversion $a_i / \overline{a}_i$.

Further, based on this pair of primitives (and, or), a more complex system of logical elements is constructed to organize and optimize the computational processes. Disadvantages of quantum classical computing are: 1) The high cost of maintaining the temperature conditions for the operation of quantum atomic structures at a level of -270 degrees Celsius; 2) Observability of the results of computational processes, leading to data destruction after reading.

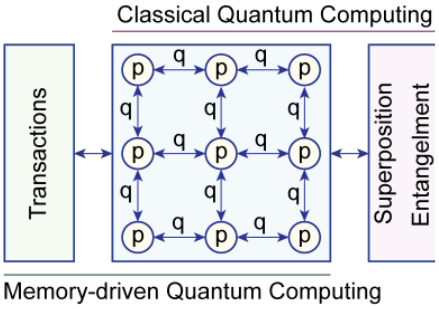


Fig. 1. Two types of quantum computing

Innovation in the architecture of quantum computing is determined by the elimination of logic, associated with superposition and entanglement. The analogy can be the memory-driven architecture of the classic computer, free of reusable logic. In such a computer, there is nothing but a memory, where a transaction (write-read operation) is performed on the address memory. Transactions are sufficient for organizing any computational process by using a unique characteristic equation [2, 3]:

$$M_i = Q_i[M(X_i)].$$

Here is the memory for the: vector-state of the computational process; vector-qubit of a logical primitive; vector-address of the logic Q-coverage cell. Q-logic is implemented on addressable memory, where all primitives are also integrated under M-state vector, which forms the binary addresses of M(X) from the array of variable X.

An innovative proposal is to create a quantum memory-driven computing without superposition and entanglement operations based on the use of the above characteristic equation, which realizes read-write transactions on the electron structure (see Fig. 1). To exclude two mentioned operations from quantum computing means to significantly simplify the architecture based on the memory of electrons to perform transactions between them using quanta or photons.

A confirmation of the validity of the proposed innovative quantum architecture can serve several recent publications that focus stable trends to create quantum computing, based on the atomic structure of memory with the transmission of information by means of photons or quanta [4-11].

Scientists from the California Institute of Technology created an optical quantum memory [5], in which information is transmitted by encoding data leveraging the quantum state of photons. Memory is realized on rare-earth elements and is capable of storing photon states with the help of intermediate resonators between the atom and light.

Vladimir Hahanov is with the Kharkov National University of Radioelectronics, Kharkov, 61166, Ukraine (phone: 380577021326; fax: 380577021326; e-mail: *hahanov@icloud.com*).

Ka Lok Man is with the Xi'an Jiaotong-Liverpool University, China. (e-mail: *kalok2006@gmail.com*).

Mykhailo Liubarskyi is with the Kharkov National University of Radioelectronics, Kharkov, 61166, Ukraine (e-mail: *mlyubarskyy@gmail.com*).

Ivan Hahanov is with the Kharkov National University of Radioelectronics, Kharkov, 61166, Ukraine (phone: 380577021326; fax: 380577021326; e-mail: *ivanhahanov@icloud.com*).

Svetlana Chumachenko is with the Kharkov National University of Radioelectronics, Kharkov, 61166, Ukraine (phone: 380577021326; fax: 380577021326; e-mail: *svetachumachenko@icloud.com*).

ISBN: 978-988-14048-8-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The dimension of quantum memory is 1000 times smaller than traditional classical solutions. It is implemented in a nanotube, which allows storing information in a very small volume.

Practical realization of the idea of replacing electrons with photons leads to the creation of computing with a performance close to the speed of light [6, 7]. Korean researchers have taken one more step toward quantum-optical computing. They created a photon-triggered nano-wire transistor based on crystal and porous silicon, where the switching and amplifying of the current is carried out under the influence of a photon. The use of photons in logical AND, OR and NAND gates leads to the ultracompact nanoprocessors and nanoscale photodetectors for high-resolution imaging.

Scientists from Columbia University drive on the way to create a transistor from one atom in molecular electronics [8, 9]. They created a geometrically ordered cluster of inorganic atoms with a central nucleus consisting of 14 atoms, which was connected to gold electrodes, which allowed controlling the transistor under the influence of one electron at room temperature.

The transmission of digital signals between molecules was achieved for the first time, which is a significant achievement of molecular computation development [10, 11]. The creation of electronic components from individual molecules is focused on miniaturization and integration of electronic devices. However, the practical implementation of molecular devices and circuits for the transmission and processing of signals at room temperature is a complex problem, which was solved by placing SnCl2Pc molecules on the copper (Cu) surface. The planar orientation of molecules in intermolecular interaction can be considered as a carrier of information. In connected molecular arrays, the signal is transmitted from one molecule to another along previously specified routes, which implement logical operations. The phenomena of planar orientation allow the use of molecules with internal bistable states to create complex molecular devices and circuits.

The theoretical similarity of classical computing with quantum lies in a general model of the computing architecture that uses memory for data storage and a functionally complete basis of primitive elements (or, not) = (superposition, entanglement) for the implementation of arithmetic logic operations on data.

What are the formal differences between classical and quantum computing? The first of them sequentially processes addressable or ordered heterogeneous data, spending on the procedure Q = n cycles. It is also capable of processing homogeneous data in parallel and in one automatic cycle. If the data are not ordered and are sets, then the limiting computational complexity of their processing on a classical computer depends on the power of the two sets and is defined as $Q = n \times m$. For example, to intersect two sets:

$$M_1 \cap M_2 = \{Q, E, H\} \cap \{E, H, J\} = \{E, H\}$$

it is necessary to spend 6 automatic cycles. Quantum computing eliminates this drawback associated with the quadratic or multiplicative computational complexity of the intersection procedure on a classical computer. He solves the problem of simultaneous and parallel processing of set-theoretical data. An example of this is the parallel execution of the above-mentioned operation of intersection over sets in one automatic cycle. To do this, the operation of superposition (union) of primitive symbols, entering into sets:

$$M_1 \cap M_2 = \{V\} \cap \{C\} = \{P\} = \{E, H\},$$

taking into account a closed set-theoretic alphabet [14]: B*(Y)={Q, E, H, J, O={Q, H}, I={E, J}, A={Q, E}, B={H, J}, S={Q, J}, P={E, H}, C={E, H, J}, F={Q, H, J}, L={Q, E, J}, V={Q, E, H}, Y={Q, E, H, J}, $U = Æ$}.

The symbols of the alphabet represent the set of all subsets on the universe Y, which are composed by superposition of primitives. Quantum superposition makes it possible to concentrate several discrete states at one point in the Hilbert space. Similarly, the join operation also creates a symbolic image in a single point in the discrete space containing several states.

Based on the mentioned, it's enough to just use a multi-valued closed alphabet to simulate quantum computing on a classical one. But for this it is necessary to first create a symbolic system (set theory algebra) for the encoding of states. The simplest is the Cantor algebra, which operates with two discrete states and creates 4 symbols: $A^k = \{0, 1, X = \{0, 1\}, Æ\}$. The symbols of this alphabet are a set-theoretic interpretation and isomorphism of the qubit. Otherwise, the superposition of two states of one qubit creates 4 symbols. Naturally, two qubits are capable to generate 16 states, three qubits – 64 states. In the general case, the number of states Q has a dependence on the number of qubits n, which is represented by the following formula: $Q = 2^{2^n}$.

For parallel execution, but already logical operations on qubits, it is necessary to encode the primitive symbols of the alphabet with a unary binary code. The remaining symbols are obtained by superposition of primitive codes. The exception is the character code of the empty set, which is obtained by applying the logical and-operation. For the Cantor algebra the correspondence table "Symbol-Code" has the following form:

$a_i \in A^k$	0	1	X	Æ
$C(a_i)$	10	01	11	00

.

The payment for the parallelism of logical operations on sets in a classical computer is a significant increase in the bit length (register, memory) for encoding symbols of the alphabet. A similar correspondence table for encoding the hexadecimal alphabet B*(Y) has the form:

$a_i \in B^*$	Q	E	H	J	O	I	A	B	S	P	C	F	L	V	Y	Æ
$C(a_i)$	1	0	0	0	1	0	1	0	1	0	0	1	1	1	1	0
	0	1	0	0	0	1	1	0	0	1	1	0	1	1	1	0
	0	0	1	0	1	0	0	1	0	1	1	1	0	1	1	0
	0	0	0	1	0	1	0	1	1	0	1	1	1	0	1	0

.

Thus, the expansion of the power of the set-theoretical alphabet can be matched with the build-up of qubits in a classical quantum computer. This makes it possible to

ISBN: 978-988-14048-8-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

perform computational procedures in parallel and in one automatic clock cycle based on the leverage of logical (set-theoretic) operations.

As for the tabular model of a logical element, it is initially represented by a set of rows or by a set of discrete relations between input and output variables. Instead of such set amount, a qubit vector of output states is proposed, oriented to addressable parallel simulation of digital logic circuits. Replacement of an unordered row set of a truth table by an ordered vector of addressable states makes it possible to create parallel computation on classical computers by increasing the memory for unary encoding of each state. Otherwise, the superposition of n elements of a finite set in a quantum (Q) computer has a one-to-one correspondence to the n-dimensional vector in the classical (C) addressable computer, Fig. 2. This vector is obtained by performing or-operation on unary codes of primitive elements of the original set. Naturally, any intersection (and), union (or), addition (not) in C-computer of unary data codes is performed in parallel in one automatic cycle, as in a Q-computer. The pay for the speed is the increase in the memory (the number of bits) for unary encoding of symbols, relative to positional encoding, which is determined by the following expression: Q = n / log n.

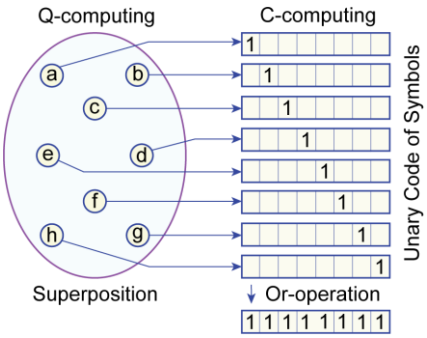


Fig. 2. Superposition of primitive elements and logical union of vectors

Thus, the expansion of the power of the set-theoretical alphabet can be matched with the build up of qubits in a quantum computer. This makes it possible to perform computational processes in parallel and in one automaton clock cycle on the basis of the use of logical (set-theoretic) operations.

The superposition of n elements of the finite-set in quantum (Q) computing corresponds uniquely to the n-dimensional vector in the classical (C) addressable computing, which is obtained by applying the or-operation to the unary codes of the symbols of the original set executed in parallel in one automaton clock cycle.

An optimal solution of the coverage problem [12] is obtained by using the Quantum Coverage Processor (QCP), which creates all possible combinations of input vectors in the form of qubit data structures represented by a set of all subsets, Fig. 3. The circuit has a logic analyzer at each register output, which determines the completeness of the coverage by executing and-operation on all bits of the register variable. The number of such vector-bit converter functions corresponds to the number of elements and is equal to $Q = 2^n - 1$. The bit results of executing and-operations are integrated into the register of RG analysis, which identifies the obtained coverage by unit bits. The last digit of the register, equal to 1, indicates the existence of a positive result obtained during the search for coverage.

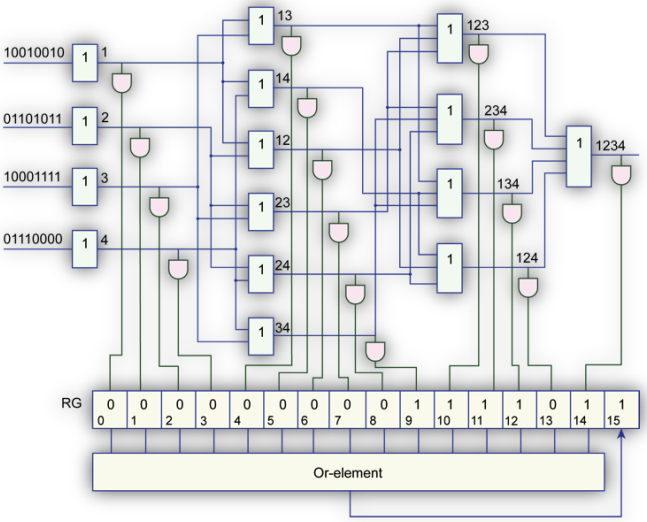


Fig. 3. QC-Processor for searching an optimal coverage

In this case, the leftmost unit in the RG register-analyzer determines the minimum coverage. The circuit is also designed to determine the primitivism or uniqueness of input vectors that is identified by zero values in all digits of the register-analyzer, except for the last one that is equal to 1 in this case. The hardware complexity of the register digital circuit for searching the optimal coverage, where n is the number of rows, m is the length of the register, is determined by the following analytical expression:

$$Q = [2^{n+1} + 2^n]' \; m + 2^n.$$

The computational complexity of cumulative procedures for searching the optimal coverage is n automatic cycles in the worst case. Xor-functions can be used in logic elements instead of or-operations, which allows solving problems of identification and recognition of cyber-objects represented in vector form. The following axiom is used here: the xor-operation of two vectors-primitives creates their logical union or superposition. Thus, n objects of discrete vector space are recognized, if all xor-combinations (except the last one) form zero values in the output of and-analyzers.

Example 1. Determine the minimum coverage of the eight bits of the following table by unit values:

X - Inputs	0	1	2	3	4	5	6	7
1	1	0	0	1	0	0	1	0
2	0	1	1	0	1	0	1	1
3	1	0	0	0	1	1	1	1
4	0	1	1	1	0	0	0	0

Four register variables X = (1,2,3,4) of the circuit forms the following state of the register-analyzer:

RG = 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1

Single values of the register show the existence of four possible solutions of the problem: C = {3,4}, {1,2,3}, {2,3,4}, {1,3,4}, {1,2,3, 4}. The minimum coverage is provided by two input vectors: C = {3,4}, which is identified by the leftmost unit in the register-analyzer RG.

The QCP structure can be simplified by removing the register, which stores the search results for the optimal

ISBN: 978-988-14048-8-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

coverage, Fig. 4. In this case, the QCP circuit becomes strictly logical, where the positive result of the search is determined in a single automaton clock by the unit value of the output state of the integral or-element. The optimal coverage will be identified by the unit value of the output of and-element, which is topologically closer to the external inputs of the circuit.

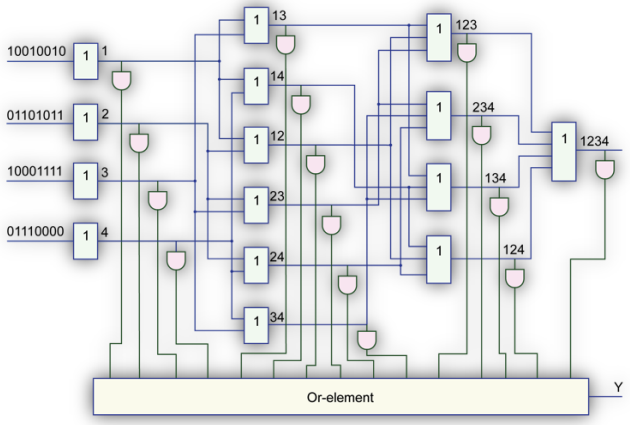


Fig. 4. Combinational QC-Processor for searching coverage

Example 2. A structure of xor-elements is presented that solves the problem of determining the set of vectors-primitives fed to the inputs of the digital circuit, Fig. 5. The positive result of the solution is defined by the presence of two rightmost unit coordinates of the vector-analyzer RG.

Explanations. Xor-operation for n primitives (vectors) always creates their union. Here we are talking about such coding of n objects in discrete cyberspace, when the simultaneous existence or superposition of all objects becomes possible within the framework of the form represented by one vector. This means that proper unary encoding n objects results in the situation when any superposition except for the last one (when all n vectors are used) will be incomplete; this results in the appearance of zero coordinates, which form the zero value of the outputs of and-analyzers.

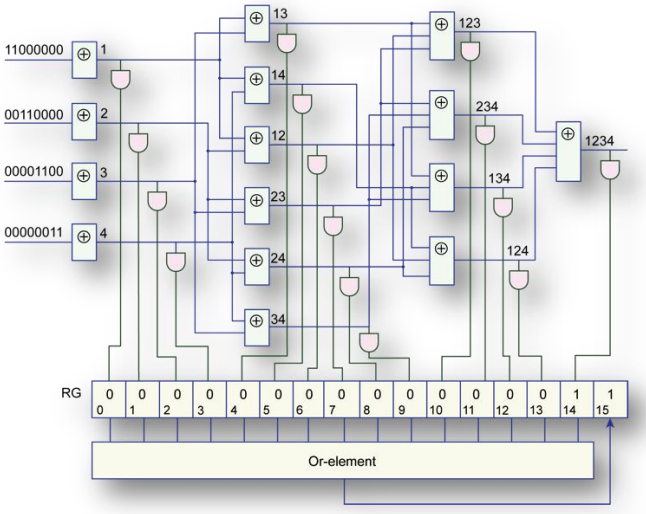


Fig. 5. A circuit for generating a set of vectors-primitives

The combinational QC-processor is proposed for parallel solving the coverage problem, which is characterized by simultaneous calculation of all possible combinatorial variants of coverage through hardware implementation of superposition operation, which makes it possible to increase the performance of procedures in several times in searching the optimal solution.

II. The Method of Quantum Minimization of Boolean Functions

The quantum representation of data in the form of a superposition of unary codes can be used to significantly simplify the method of undetermined coefficients while minimizing Boolean functions [13].

Statement. Any complex truth table of a discrete object can be represented by no more than two vectors of quantum coverage under unitary encoding (UC) of input states. The procedure that illustrates this statement is shown in Fig. 6. Zero and unit truth table cubes are shown here (input states, which are unary coded and logically combined, VUC). As a result, two quantum-coverage vectors are obtained, each of them can represent a logical function in the form of a qubit coverage.

Suppose there is a table $T = T_{ij},\ i = \overline{1,m};\ j = \overline{1,n}$. In the worst case, the number of states in each column of the table (matrix) is m. This number of states can be represented by m bits of a unitary code. (For example, a single line (11) consisting of two binary digits can represent two symbols of the Cantor alphabet:

$$A^k = \{0 \to 10;\ 1 \to 01;\ X \to 11;\ \varnothing \to 00\}.$$

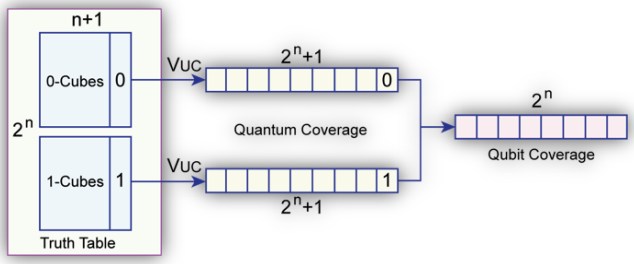


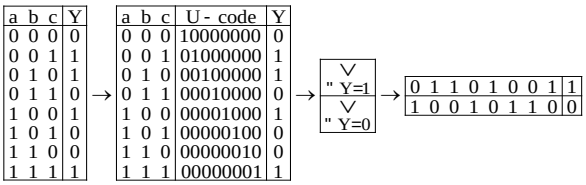
Fig. 6. The model for generating qubit coverage

As a result, we obtain a matrix of dimension m by n, where each cell contains m-bit vector. Applying a superposition operation or a logical union to all rows of a table through unitary encoding creates one row that forms in a compact form the relations previously represented by the source table. For example, all binary-decimal codes of input states of a three variable logical element are represented by a vector (11111111), if the following encoding is applied: 000 – 10000000, 001 – 01000000, 010 – 00100000, 011 – 00010000, 100 – 00001000, 101 – 00000100, 110 – 00000010, 111 – 00000001. However, the truth table is a functional match

$$Y = f(X),\ X \to Y,\ X = \{x_1, x_2, ..., x_i, ..., x_n\},\ Y = \{0,1\}.$$

Taking into account that the function is defined on two discrete values {0,1}, the binary truth table can always be represented by two rows; each of them is a result of superposition or union 0 or 1 unitary code of input actions. For example, after the logical union of unary codes of input states by unit and zero output value the truth table of a three input xor element has the following form:

ISBN: 978-988-14048-8-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



Thus, any truth table of a digital device can be represented in an explicit form by two rows-cubes of quantum coverage, which unite through superposition of unit and zero (in output state) input values of the original table. The cubes of a quantum coverage are always mutually inverse, so to define the functionality it is enough to leverage one of them, assuming that the second one can be quickly and simply completed through the inversion operation, if necessary.

Taking into account the possibility of defining the truth table by two cubes of quantum coverage, we further suggest improvement of the known method of undetermined coefficients for minimization of the logical functions. Let there be a source table of undetermined coefficients to minimize the Boolean function of three variables [14]:

T_{ij}	$x_1x_2x_3$	k_1	k_2	k_3	k_{12}	k_{13}	k_{23}	k_{123}	f_i
0	000	k_1^0	k_2^0	k_3^0	k_{12}^{00}	k_{13}^{00}	k_{23}^{00}	k_{123}^{000}	1
1	001	k_1^0	k_2^0	k_3^1	k_{12}^{00}	k_{13}^{01}	k_{23}^{01}	k_{123}^{001}	0
2	010	k_1^0	k_2^1	k_3^0	k_{12}^{01}	k_{13}^{00}	k_{23}^{10}	k_{123}^{010}	1
3	011	k_1^0	k_2^1	k_3^1	k_{12}^{01}	k_{13}^{01}	k_{23}^{11}	k_{123}^{011}	0
4	100	k_1^0	k_2^0	k_3^0	k_{12}^{10}	k_{13}^{10}	k_{23}^{00}	k_{123}^{100}	1
5	101	k_1^1	k_2^0	k_3^1	k_{12}^{10}	k_{13}^{11}	k_{23}^{01}	k_{123}^{101}	0
6	110	k_1^1	k_2^1	k_3^0	k_{12}^{11}	k_{13}^{10}	k_{23}^{10}	k_{123}^{110}	0
7	111	k_1^1	k_2^1	k_3^1	k_{12}^{11}	k_{13}^{11}	k_{23}^{11}	k_{123}^{111}	1

This table is converted to the binary form of all possible combinatorial combinations of the input variable states, which can be used to form the output values of the function represented in the last column:

T_{ij}	$x_1x_2x_3$	1	2	3	12	13	23	123	f
0	000	0	0	0	00	00	00	000	1
1	001	0	0	1	00	01	01	001	0
2	010	0	1	0	01	00	10	010	1
3	011	0	1	1	01	01	11	011	0
4	100	1	0	0	10	10	00	100	1
5	101	1	0	1	10	11	01	101	0
6	110	1	1	0	11	10	10	110	0
7	111	1	1	1	11	11	11	111	1

Naturally, the combinations of input values obtained in the cells of the table: 0,1; 00, 01, 10, 11; 000, 001, 010, 011, 100, 101, 110, 111 are trivially transformed into unitary codes of binary states 10, 01; 1000, 0100, 0010, 0001; 10000000, 01000000, 00100000, 00010000, 00001000, 00000100, 00000010, 00000001, respectively:

T_{ij}	$x_1x_2x_3$	1	2	3	12	13	23	123	f
0	000	10	10	10	1000	1000	1000	10000000	1
1	001	10	10	01	1000	0100	0100	01000000	0
2	010	10	01	10	0100	1000	0010	00100000	1
3	011	10	01	01	0100	0100	0001	00010000	0
4	100	01	10	10	0010	0010	1000	00001000	1
5	101	01	10	01	0010	0001	0100	00000100	0
6	110	01	01	10	0001	0010	0010	00000010	0
7	111	01	01	01	0001	0001	0001	00000001	1

Next, a separate logical union of all unit and zero rows of the table in two integrating vectors is performed. As a result, all possible combinations of variables are obtained, which form unit and zero values of the function.

Q	Operations	1	2	3	12	13	23	123	f
1	$Q^1 = \bigcup_{f_i=1} T_{ij}$	11	11	11	1111	1011	1011	10101001	1
2	$Q^0 = \bigcup_{f_i=0} T_{ij}$	11	11	11	1111	0111	0111	01010110	0
3	$Q = (\bigcup_{f_i=1} T_{ij}) \cap \overline{(\bigcup_{f_i=0} T_{ij})}$	00	00	00	0000	1000	1000	10101001	Y

In order to obtain a disjunctive form of minimizing dimension (row 3 in the above Q-table), it is necessary to subtract the zero cube from the unit cube of the quantum coverage by the rule represented in the following formula:

$$Q = (\bigcup_{f_i=1} T_{ij}) \setminus (\bigcup_{f_i=0} T_{ij}) = (\bigcup_{f_i=1} T_{ij}) \cap \overline{(\bigcup_{f_i=0} T_{ij})}.$$

Decrypting the resulting quantum cube Q into a disjunctive normal form gives the following result:

$$Y = \bar{x}_1\bar{x}_3 \cup x_2\bar{x}_3 \cup \bar{x}_1\bar{x}_2\bar{x}_3 \cup \bar{x}_1x_2\bar{x}_3 \cup x_1\bar{x}_2\bar{x}_3 \cup x_1x_2x_3.$$

This form is not minimal, and therefore requires solving the coverage problem of the simplest initial unit terms (000, 010, 100, 111) by the solutions obtained. For the function Y, it is obvious that the first two terms (0x0, x00) cover the logical summands 3,4,5 or (000, 010, 100), which are redundant in accordance with the absorption rule $a \cup ab = a$, this makes it possible to obtain a minimal disjunctive normal form in the following form:

$$Y = \bar{x}_1\bar{x}_3 \cup x_2\bar{x}_3 \cup x_1x_2x_3.$$

Another solution of the coverage problem is associated with the use of a QC processor having 6 register inputs, which allows determining the minimum DNF by simulation of the binary codes-rows $X_i \in X$ of the coverage table:

T	000	010	100	111	$X_i \in X$
0x0	1	1	.	.	1100
x00	1	.	1	.	1010
000	1	.	.	.	1000
010	.	1	.	.	0100
100	.	.	1	.	0010
111	.	.	.	1	0001

The result of code simulation defines a minimum coverage as three code rows, which create a minimal function:

$$Y = 1100 \vee 1010 \vee 0001 \rightarrow 0x0 \vee x00 \vee 111 \rightarrow \bar{x}_1\bar{x}_3 \vee x_2\bar{x}_3 \vee x_1x_2x_3.$$

This method can be used to obtain the minimum DNF or CNF by using truth tables, where the number of zero and unit cube rows does not differ much from each other. Another application of the method is associated with a significant minimization of the faulty area when diagnosing digital systems.

The computational complexity Q of the quantum method of undetermined coefficients is defined by the expression that forms the time for unary coding of states of the truth table (for comparison, Q^b is the complexity of the basic minimization method):

$$Q = 2^n \cdot 3^n;$$
$$Q^b = 2^n \cdot 2^n \cdot 2^n = 2^n \cdot 2^{n+n};$$
$$R = \frac{Q}{Q^b} = \frac{2^n \cdot 3^n}{2^n \cdot 2^{n+n}} = \frac{3^n}{2^{2n}}.$$

ISBN: 978-988-14048-8-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Thus, the computational complexity Q of obtaining compact quantum coverage for minimizing Boolean functions is significantly smaller compared to the base method Q^b of undetermined coefficients using a special form of the truth table.

Excluding the preprocessing of the truth table, which consists in unary coding of states, the computational complexity of the Boolean function minimization method is defined by only three vector parallel operations.

The memory costs H for storing data structures are formed by the dimension of the table, necessary for superpositioning two quantum-coverage vectors, where table cells are represented by unary state codes:

$$H = 2^n \times 3^n;$$
$$H^b = 2^n \times 2^n = 2^{2n};$$
$$S = \frac{H}{H^b} = \frac{2^n \times 3^n}{2^n \times 2^n} = \frac{3^n}{2^n}.$$

Thus, in order to obtain the compact quantum coverage, it is necessary to leverage a table H, which has substantially larger dimension than the original truth table H^b.

A quantum method for Boolean functions minimization is proposed, which differs from the method of undetermined coefficients by parallel execution of the superposition operation over 0 and 1 states of input variables represented by unitary codes, which makes it possible to significantly improve performance due to redundant memory.

III. Conclusion

1. The memory-driven innovation architecture of quantum computing is presented, which is determined by the ability to eliminate logic associated with superposition and entanglement of states, based on the use of the characteristic equation that realizes read-write transactions on the structure of electrons. Eliminating logical operations from quantum computing will greatly simplify the architecture to the level of the memory structure of electrons to perform transactions between them using quanta or photons. The formal difference between quantum computing and classical is shown, which consists in the possibility of parallel and simultaneous execution of logical operations on sets.

2. A quantum method for Boolean functions minimization is proposed, which differs from the method of undetermined coefficients by parallel execution of the superposition operation over 0 and 1 states of input variables represented by unitary codes, which makes it possible to significantly improve performance due to redundant memory. A quantum method for coverage problem solving is also proposed, which leverages a digital logic scheme for quasi-parallel search for the optimal coverage.

References

[1] V. Hahanov. *Cyber Physical Computing for IoT-driven Services*. New York: Springer, 2017.

[2] V.I. Hahanov, T. Bani Amer, S.V. Chumachenko, E.I. Litvinova. (2015) “Qubit technology for analysis and diagnosis of digital devices”, *Electronic modeling*. J 37 (3), pp.17-40.

[3] V. Hahanov, W. Gharibi, E. Litvinova, M. Liubarskyi, A. Hahanova. “Quantum memory-driven computing for test synthesis”, in Proc. IEEE East-West Design and Test Symposium, Novi Sad, Serbia, 2017, pp. 123-128.

[4] P. Roushan, C. Neill, J. Tangpanitanon, V. M. Bastidas, A. Megrant, R. Barends, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. Fowler, B. Foxen, M. Giustina, E. Jeffrey, J. Kelly, E. Lucero, J. Mutus, M. Neeley, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. White, H. Neven, D. G. Angelakis, J. Martinis. “Spectroscopic signatures of localization with interacting photons in superconducting qubits”, *Science*, 01 Dec 2017, vol. 358, issue 6367, pp. 1175-1179.

[5] T. Zhong, J. M. Kindem, J. G. Bartholomew at all. “Nanophotonic rare-earth quantum memory with optically controlled retrieval”, *Science*. 29 Sep 2017, Vol. 357, Issue 6358, pp. 1392-1395.

[6] J. Kim, Hoo-Cheol Lee, Kyoung-Ho Kim at all. “Photon-triggered nanowire transistors”, *Nature Nanotechnology* 12, 2017, pp. 963–968 .

[7] D. Johnson. “Photon-Triggered Nanowire Transistors a Step Toward Optical Computing”. August 2017. [Online]. Available: https://spectrum.ieee.org/nanoclast/semiconductors/devices/nanowire-transisitors-triggered-by-photons-combine-photonics-and-electronics

[8] D. Johnson. “Single-Molecule Transistors Get Reproducibility and Room-Temperature Operation “. August 2017. [Online]. Available: https://spectrum.ieee.org/nanoclast/semiconductors/devices/single-molecule-transistors-get-reproducibility-and-roomtemperature-operation

[9] G. Lovat, B. Choi, D. W. Paley at all. “Room-temperature current blockade in atomically defined single-cluster junctions”, *Nature Nanotechnology 12(11)*, pp.1050.

[10] D. Johnson. “Quantum Optical Memory Device One Thousand Times Smaller Than Previous Options “. August 2017. [Online]. Available: https://spectrum.ieee.org/nanoclast/semiconductors/nanotechnology/quantum-optical-memory-device-one-thousand-times-smaller-than-previous-options

[11] C. Li, Z. Wang, Y. Lu, X. Liu, L. Wang. “Conformation-based signal transfer and processing at the single-molecule level”. *Nature Nanotechnology 12(11)*. November 2017, pp. 1071-1076.

[12] P.T. Hester, K. Adams. *Systemic Decision Making. Fundamentals for Addressing Problems and Messes*, Springer. 2017.

[13] V. Hahanov, A. Barkalov, M. Adamski. *Infrastructure intellectual property for SoC simulation and diagnosis service*, Springer, Germany, 2011.

[14] A.Ya. Savelyev. Applied theory of digital automata. M: Vischaya. schkola, 1987.

ISBN: 978-988-14048-8-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)