Quantum Data Structures for SoC Design

Svetlana Chumachenko, Alexander Shkil, Anastasiya Hahanova, Artur Ziarmand, Aleksey Pryimak

Design Automation Department, Kharkov National University of Radioelectronics, UKRAINE, Kharkov, Lenin ave. 14,

E-mail: hahanov@kture.kharkov.ua

Abstract – **A qubit-vector model of computing automaton is proposed; it is characterized by the transactional interaction of memory components, which represent the combinational and sequential elements and are implemented in the form of a qubit or "quantum" primitives needed to create parallel virtual computers and cloud-focused processors.**

Keywords - **Qubit, Quantum Data Structure, Cloud, SoC Design.**

I. Introduction

The goal is to develop quantum-like data structures, which are characterized by the properties of superposition, entanglement and parallelism, for creating high-performance deterministic computers focused on high-speed analysis of processes and phenomena in cyberspace. The objectives are the following: 1) Development of the power set data structures isomorphic to the qubit structures in non-deterministic quantum computers. 2) Using the power set data structures for a compact description of computational processes based on the use of logical operations. 3) Examples of compact description and analysis of digital components.

Market feasibility of using emulation of quantum computing methods when creating virtual (cloud) computer (VC) in cyberspace is based on the use of qubit data models, focused on parallel solving the discrete optimization problems with a significant increase in the cost of memory [1-6]. Without considering the physical foundations of quantum mechanics related to non-deterministic interaction of atomic particles [10-13], we use the notion of the qubit as a binary or multi-valued vector for joint and simultaneous defining the power set of states of a discrete cyberspace area based on linear superposition of unitary codes, focused on parallel usage of methods for analyzing and synthesizing components of cyberspace. In the rapidly developing theory of quantum computing the state vectors form a quantum register of n qubits, forming a unitary or Hilbert space H, the dimension of which has power-law dependence on the number of qubits $\text{Dim } H = 2^n$.

The motivation of a new approach for the design of the VC is governed by the advent of cloud services, which are dedicated virtual computer systems, distributed in space and implemented in hardware or software. The programmer is not always interested in how and where are implemented codes of software applications, as well as he is no need to know the theory of designing digital automaton on gate level, register transfer level and use specific components of computer devices (flip-flops, registers, counters, multiplexers). Any component can be functionally represented in vector form of the truth table, implemented by using memory. Traditionally implemented logic functions are not considered. It partially reduces performance, but considering that 94% of SoC is the memory the remaining 6% is also can be implemented in memory components that will not be critical for the majority of software and hardware applications. Therefore, for programming effective virtual computers a theory based on two components of a higher abstraction level (memory and transaction) can be used.

II. Quantum Data Structures

The feature of data organization in a classical computer is that each bit, byte or other component has its own address. Therefore, there is a problem of effective processing an association (finite symbol alphabet) of equal elements, which do not have the address order by definition, e.g., the set of all subsets. A solution could be based on a processor, where the unit cell is the image or pattern of the universe of n unitary coded primitives using superposition to form a power set $|B(A)| = 2^n$ of all possible states of a cell in the form of the set of all subsets [4,6]. In fact, it turns out that for the efficient processing the set of all subsets the vector of the addressable cells or bits is introduced; the power set of states is formed from its zeros and ones. Otherwise, the set-theoretic culture should be reduced to the address-focused data structures of a modern computer to significantly improve the performance when executing the set-theoretic operations. Thus, the power set-vector is the addressable representation of the power set of states as the set of all subsets of unitary encoded elements for executing logic operations in parallel instead of coordinate-wise set-theoretic procedures.

A certain analogy between the vector data structures of the power set and the qubit of the quantum computer is there. Quantum qubit concept as an elementary structure for storing information in a quantum computer that enables the superposition of states $|\psi\rangle = \alpha\cdot|0\rangle + \beta\cdot|1\rangle$ can be put in one-to-one correspondence with the power set of states in a classical computer [4-6], forming, for example, the Cantor's alphabet $A^k = \{0,1,X,\varnothing\}$, $X = \{0,1\}$. Here two first primitive of the alphabet are unitary encoded by the following vectors (ordered binary sequences): 0 = (10) и 1 = (01). The codes of other two symbols can be obtained as derivatives by using the superposition or union operations (logical adding) (10) $\vee$ (01) = (11), as well as the intersection operation (logical multiplication) (10) $\wedge$ (01) = (00), which generates four states of the power set: {(10),(01),(11),(00)} [4-6] in the form of binary vectors or tuples.

To prove the correctness of the use of the adjective "qubit" for models of digital units, it is necessary to compare the

algebra of set theory and linear algebra by the example of Cantor's algebra $A^k = \{0,1,X,\varnothing\}$. The first two symbols are primitive or primary elements which are not decomposed into components. The third symbol is derived in the set theory and defined by the superposition or union of symbols-primitives: $X = 0 \cup 1$. In a Hilbert space, where the linear algebra operates by the primitives $\alpha\cdot|0\rangle$ и $\beta\cdot|1\rangle$ of a qubit, the third symbol is formed by superposition of two components of the state vector: $|\psi\rangle = \alpha\cdot|0\rangle + \beta\cdot|1\rangle$. Complete analogy is obvious; it can be applied to form the fourth symbol (the empty set) that is absent in a qubit (linear) algebra by using the function inverse of the superposition. In the set theory it is the intersection operation, which forms an empty set by the primitive symbols: $\varnothing = 0 \cap 1$. In a Hilbert space such operation is scalar (inner) product or Dirac function (symbol) $< a \mid b >$ [1,5], which has a geometric interpretation in the following form $< a \mid b > \approx \mid a \mid \times \mid b \mid \times \cos\angle(a,b)$. If the projection a and b of a quantum state vector are orthogonal (this is true), the following result is appeared $< \alpha \mid \beta > = \mid a \mid \times \mid b \mid \times \cos\angle(a,b) = \mid a \mid \times \mid b \mid \times \cos\angle 90^o = 0$. The scalar product of orthogonal vectors is zero. This result is an analogue of the empty set symbol when performing the intersection operation of the algebra of sets. Thus, the correspondence table of the set algebra and linear algebra presented below confirm the property of isomorphism between the symbols of the power set and states of the qubit vector obtained using isomorphic operations of union - superposition and intersection - the scalar product:

Boolean $A^k =$	0	1	$X = 0 \cup 1$	$\varnothing = 0 \cap 1$
Qubit $\mid\psi\rangle =$	$\mid 0\rangle$	$\mid 1\rangle$	$\alpha \mid 0\rangle + \beta \mid 1\rangle$	$\alpha \mid 0\rangle \mid \beta \mid 1\rangle$

So, the data structure "power set" can be considered as a deterministic (not-probability) image or analogue of quantum qubit in the algebra of sets (logic), the primitive elements of which are unitarily encoded by binary vectors (tuples) and characterized by the properties of superposition, parallelism and entanglement. This enables the use of the proposed model for improving the performance of the analysis of digital units using classic computers, as well as quantum computers without substantial modification, which will appear for 5-10 years in the market of electronic technology.

In [4] the possibility of emulating the classic computing processes on quantum computers is considered. But given the "reversibility" of conformity of these algebras, it is further proposed the inverse transformation - emulation of some advantages (superposition, parallelism, uncertainty) of quantum computing based on classical processors using the power set as the base model of data structures for interpretative describing and modeling digital systems.

In the market of electronic technology the competition exists between the variants of idea implementation [1-6]:

1) software implementation of the project is associated with the synthesis of the interpretative model of software functionality or hardware programmable logic devices based on FPGA, CPLD; the advantage is technological design modification, the disadvantage is low performance of the digital system;

2) hardware implementation is focused on the use of compiling models in the development of software applications or the implementation of the project as VLSI chips [6]. Advantages and disadvantages of a hardware implementation are inverse in regard to the software implementation: high performance and the inability to modify.

Taking into account the basic variants of the idea implementation the quantum data structures focused on improving the performance of flexible models of software or hardware implementation of the project are proposed.

Quantum description of the structure of digital systems is represented below. Qubit (n-qubit) is the vector form of the unitary (unary) coding of the universe of n primitives to define the power set of states 2^{2^n} using 2^n binary variables. For example, if n=2 then 2-qubit defines 16 states by using four variables. If n=1 then qubit defines four states on the universe of two primitives (10) and (01) by using two binary variables (00,01,10,11) [4]. At that the superposition (the simultaneous existence) is allowed in the vector of 2^n states, designated as primitives. Qubit (n-qubit) allows the usage of parallel logical operations instead of element-wise set-theoretic operations to significantly accelerate the processes of analysis of discrete systems.

Further qubit is identified with n-qubit or a binary vector, if it does not interfere with the understanding the presented material. Since quantum computation is related to the analysis of qubit data structures, we will continue to apply the definition of "quantum" to identify technology that uses three properties of quantum mechanics: parallelism of processing binary vectors, a superposition of states and their entanglement. Qubit synonymous when specifying a binary vector of logic function is Q-coverage (Q-vector) [4] as a unified form of defining (superposition) output states, corresponding to the address codes of the input variables of the logic element.

Qubit in a digital system is used as a form of defining a structural primitive invariant to the implementation technology of the functionality (hardware, software). Moreover, the synthesis of digital systems based on qubit structures not tied rigidly to the Post's theorem, which defines five conditions (classes) of the existence of a functionally complete basis [5]. At the proposed abstraction level n-qubit gives more opportunities for vector defining any n-input function of the power set (the number of subsets of it is $|B(A)| = 2^{2^n}$), which certainly contains all the functionalities, meet the five classes of the Post's theorem [5]. The format of the structural qubit component of the digital circuit $Q^* = (X, Q, Y)$ includes an interface (input and output variables), as well as qubit-vector Q defining the function $Y = Q(X)$, the dimension of which is determined by the exponential function of the number of input lines $k = 2^n$.

Practically focused novelty of qubit simulation is to replace the truth tables of components of digital units by vectors of output states. Applying such transformations to the logic gates can be simply demonstrated. Let functional primitive has the following binary coverage (truth table)

$P =$

X_1	X_2	Y
0	0	1
0	1	1
1	0	1
1	1	0

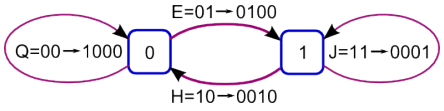
,

which can be transformed by the unitary encoding of input vectors through the use of two-stroke alphabet [6], originally designed for a compact description of all the possible automaton variables (Fig. 1).

Here there are symbols, their unitary and binary codes (for example, Q = 00 – 1000) for describing two adjacent states of automaton variables. The proposed alphabet structurally defines the power set (the set of all subsets) of states by using the universe of four primitives Y={Q,E,H,J}. A unitary code corresponds to the format of vector comprising two qubits, which form 16 symbols of two-stroke alphabet. Using the last one, any coverage of two-input logic primitive can be represented by two cubes, or even one, given that they are mutually inverse:

$P =$ | 00 1 / 01 1 / 10 1 / 11 0 | = | Q 1 / E 1 / H 1 / J 0 | = | V 1 / J 0 | = | 1110 1 / 0001 0 | → 1 1 1 0

Qubit vector is used to simulate the components of digital units, and two mutually inverse cubes - for the synthesis of tests for one-dimensional activation of circuit variables.



Q = 00 → 1000	E = 01 → 0100	H = 10 → 0010	J = 11 → 0001
O = {Q, H} = = {00,10} → 1010	I = {E, J} = = {01,11} → 0101	A = {Q, E} = = {00, 01} → 1100	B = {H, J} = = {01,11} → 0101
S = {Q, J} = = {00,11} → 1001	P = {E, H} = = {01,10} → 0110	C = {E, H, J} = = {01,10,11} → 0111	F = {Q, H, J} = = {00,10,11} → 1011
L = {Q, E, J} = = {00,01,11} → 1101	V = {Q, E, H} = = {00,01,10} → 1110	Y = {Q, E, H, J} = = {00,01,10,11} → 1111	□ = 00 → 0000

Fig. 1. Two-stroke alphabet of automaton variables and interpretation of the symbols

At first all couples are encoded with symbols of two-stroke alphabet, and then the union of the first three cubes are performed according to the rule of minimizing co-edge operator [5]: the vectors differ in one coordinate are combined into one. Then the resulting coverage of two cubes is encoded by qubit vectors, corresponding these symbols. To simulate fault-free behavior only one cube (zero or unit) is enough, as the second one is always the complement of the first one. Consequently, for example, a unit cube forming the output 1 allows eliminating the bit of the primitive output state, thereby reducing the dimension of the cube or the primitive model up to the number of addressable states of the element, where address is the vector composed of the binary values of the input variables; the vector defines the state of the primitive output. Due to the triviality does not make sense to show that, by analogy any truth table can be led to qubit functionality in the form of a vector of output states of the logic gates with n inputs. In addition, two-stroke alphabet is used for compact description of transition tables of finite automaton, which significantly reduces the analysis time of such models. For example, a complete directed graph of the 16 transitions on 4 nodes {00,01,10,11} is minimized in one cube:

{QQ=(00-00), QE=(00-01), EE=(00-11), EQ=(00-10), QJ=(01-01), EJ=(01-11), QH=(01-00), EH=(01-10), JJ=(11-11), HJ=(11-01), JH=(11-10), HH=(11-00), JQ=(10-10), HQ=(10-00), JE=(10-11), HE=(10-01)} = YY.

To summarize in defining n-qubit, it should be noted that its essence is different from the classical byte or bit by superpositional structuring binary vector that capable to store simultaneously n states (symbols) of the functionality in the power set with $|B(A)| = 2^n$ primitives, and perform in parallel logical operations with set-theoretic data in vector format. For example, the operation of the symmetric difference of the subsets A={a,b,c,d,e,f} and B={a,c,f,g,h,k} is performed in parallel in one clock cycle of xor-operation, if each element is defined by a unitary code, and subsets – by the corresponding vectors, which in this case are qubit operands to compute the symmetric difference:

9 – qubit	a	b	c	d	e	f	g	h	k
A =	1	1	1	1	1	1	0	0	0
B =	1	0	1	0	0	1	1	1	1
A ⊕ B	0	1	0	1	1	0	1	1	1

The above mentioned gives rise to define a structured vector as n-qubit as it is characterized by the properties of computing parallelism, superposition and uncertainty (entanglement - there are 6 units in each vector) of states, which occur in the quantum data structures.

III. Conclusion

The new model of computing discrete automaton in the form of qubit data structures is proposed. It is characterized by the transactional interaction of memory components, representing combinational and sequential elements implemented in the form of qubit or "quantum" primitives. Qubit data structures allow creating parallel virtual computers for effective solving the problems of big data analysis without arithmetic instructions and provide high performance of cloud-focused processors.

References

[1] A.G. Kurosh, *Course of Higher Algebra*, Ed.: Moskow: Nauka, 1968, 426 p.

[2] M. A. Nielsen, and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010, 676p.

[3] S. Stig, K.-A. Suominen, *Quantum approach to informatics*, John Wiley & Sons, Inc, 2005, 249 p.

[4] V.I. Hahanov, E.I. Litvinova, S.V. Chumachenko, et al., "Qubit Model for solving the coverage problem," in *Proc. of IEEE East-West Design and Test Symposium*, Kharkov. 14-17 September, 2012, pp.142 — 144.[5] V.A.

[5] Gorbatov, *The basics of Discrete Mathematic*, M.: Vysshaya Shkola, 1986, 311 p.

[6] M.F. Bondarenko, V.I. Hahanov, and E.I. Litvinova, "Structure of Logic Associative Processor," in *Automation and Remote Control*, 2012, № 10, pp. 71-92.