Quantumlike bits and logic gates based on classical oscillators

Yuri F. Orlov
Cornell University, Ithaca, New York 14853
(Received 15 April 2002; published 27 November 2002)

It is shown how a classical system can possess such quantum properties as indeterminism, interference of probabilities, unitary transformations, wave functions, and noncommuting operators, and be used in quantumlike computations.

DOI: 10.1103/PhysRevA.66.052324 PACS number(s): 03.67.Lx, 03.65.−w

The theoretical advantage of quantum over digital computation [1] lies mostly in the use of quantum interference. The idea is to eliminate a large number of incorrect outcomes through destructive interference while at the same time creating a high probability of correct outcomes through constructive interference. Both cases involve a relatively small set of operations. While the probabilistic nature of quantum mechanics may be a disadvantage, its combination with interference may accelerate some computations. Here we show that this combination can be reproduced in classical systems of linear oscillators and used for quantumlike computations.

The central idea is the following. Subjecting some identical *numbered* linear oscillators to a set of sequential perturbations gives us an interference effect, but not of a quantum type. This problem can be solved by introducing a device programmed to decide what number (we can also say, what oscillator) must be chosen at the end of the set of perturbations, when the number chosen before those perturbations is known. The resulting decision can then be described in a formalism mathematically indistinguishable from that of quantum computing. This decision-making device plays the role of a quantum measurement apparatus. Although the probabilities of measurement results are regulated by nature in the quantum case, whereas in our system it is by a program, this difference is irrelevant to results of computations since both measurements and choices are made only at the ends of deterministic parts of computations. Moreover, if the probability distribution itself must be an outcome of the computations, then our classical system has an advantage over a quantum one: being free from quantum collapse due to measurement, it does not need multiple repetitions of the same computations; the probabilities are directly measurable relative energies of oscillations. Another advantage of this system is that it is easier to realize. Finally, we might add that if macroscopic natural intelligence possesses some quantumlike properties, as some physicists suspect [2], then our quantumlike classical system may be a more realistic model for this phenomenon than a purely quantum system.

The correspondence between this system and quantum systems is in fact deeper than it might seem. As will be shown in the course of our discussion, the system possesses not only the *algorithmic* indeterminism typical of quantum systems [3], but also unitary transformations, wave functions, and noncommuting operators, as well as the interference of probabilities already noted. But it does not possess the property of quantum nonlocality, which means that it can be used for quantumlike computations but not for quantumlike communication.

We demonstrate first how the major building blocks of quantum computing—qubits and quantum logic gates—can be constructed and implemented in our system as quantumlike bits (which we call *Qbits*) and quantumlike logic gates. Then we show how combinations of these building blocks operate in our system by working through several initial steps of a Shor quantum algorithm [4],

The main elements of our system are the following:

(a) Numbered linear oscillators.

(b) Perturbation devices to change and exchange oscillator amplitudes and phases.

(c) Devices to ''multiply'' oscillators, imitating the multiplication of quantum states.

(d) A decision-making device producing random numbers (choices) in accordance with the relative energies of the oscillations of different oscillators.

(e) A device to establish either maximal or zero amplitudes of the oscillators.

(f) Devices to measure amplitudes, phases, and energies of oscillators. There can also be digital computing devices, etc.

The simplest element of our quantumlike system is the Qbit. A single Qbit is formed by two identical linear oscillators numbered 0 and 1. Oscillator k, $k=0,1$, oscillates as $q_k=A[c_k\exp(i\omega t)+c_k^*\exp(-i\omega t)]$, where c_k is the dimensionless complex amplitude, $|c_k|\leqslant 1$, and $2A$ is the maximal amplitude of the q oscillations. The energy of the noninteracting oscillators

$$H=\sum H_k=2m\omega^2A^2\sum |c_k|^2. \tag{1}$$

By design, $|c_0|^2+|c_1|^2=1$, so $|c_k|^2$ is the relative energy of the kth oscillator. Consider all possible unitary transformations of the complex amplitudes made with the help of *perturbations* between times t and t' such that the full energy H is conserved, $H(t)=H(t')$. For a given transformation U, $c_k(t)\to c_k'(t')$,

$$c_k'=u_{kl}c_l, \quad u_{kl}=u_{lk}^*, \quad \sum |c_k|^2=\sum |c_l^*|^2=1. \tag{2}$$

Unitary transformations of complex amplitudes provide the possibility of using interference in the computations. But doing so inevitably requires a probabilistic approach, as in quantum mechanics. To understand this, let us try to solve the following problem. Imagine for a moment that some computation involves only a single Qbit. At time t, one of the two oscillators' numbers, 0 or 1—let it be 0—is chosen as the initial condition for this computation. Since our numbers are identified with the corresponding oscillators, this choice means by definition that $c_0(t)=1$, while $c_1(t)=0$. These initial amplitudes can be established with the help of device (e). Then, in the course of the computation, the corresponding perturbations between times t and t' change the amplitudes and phases of both oscillators, as in Eq. (2) where the U is defined by these perturbations. In accordance with the very concept of computation we want to know the *certain* result at the end, at time t', namely, a new number, 0 or 1. This is possible in some cases. For example, if at the end $c_0(t')=0$ and $c_1(t')=1$, then obviously the result equals 1. In the more complicated general case, the final amplitudes equal neither 0 nor 1, so we want a program that assigns some of all the possible parameters of matrix U to the number 0, and the rest to the number 1; these assignments can be different for two different initial conditions, 0 and 1. However, *the problem of finding an assignment consistent with the properties of unitary transformations is algorithmically unsolvable*. For there exists an *infinite* set of transformation parameters that cannot be divided into two subsets assigned to 0 and 1, respectively, without logical contradiction. [Within the group of transformations (2) there is a subgroup isomorphic to the spin 1/2 rotations on a plane, when this spin is placed on the same plane. The algorithmic unsolvability for such a case was proved in Ref. [3].]

Our system is thus algorithmically indeterministic with respect to getting final certain numbers. So we need to introduce a probability distribution and use a probabilistic program according to which decision-making device (d) can *choose* random numbers as the results of computations (which themselves can be parts of larger computations); these results are similar to random *measurement* results of quantum computations. The only consistent way to introduce probabilities of choices w_k, $k=0,1$, in the frame of our system is to combine Eqs. (1) and (2). This gives $w_k=|c_k|^2$.

Under certain conventions, transformation U can be considered a rotation in a vector Hilbert space whose eigenvectors—we will call them "number eigenstates"—are $|x\rangle$, where x is integer, and state vectors ("number states") are

$$|\psi(x)\rangle=\sum_x c_x|x\rangle. \tag{3}$$

$|x\rangle$ corresponds to oscillator x having maximal amplitude and a phase defined as the zero phase. $c_x|x\rangle$ corresponds to the same oscillator having complex amplitude c_x. As for superposition (3), it does not correspond to any physical sum of oscillators having different numbers x and amplitudes c_x. Nevertheless, $\psi(x)$, which can be written also as a column of the c_x values, as in quantum mechanics, plays the role of the wave function because c_x are now probability amplitudes. There exists, of course, a fundamental difference between our system and quantum systems. In quantum computers, all but one possible numerical outcomes of a measurement of the x value are destroyed by that measurement. In our system, *nothing is destroyed by any measurement or choice* as long as device (e) is not used.

Quantumlike logic gates, which are reversible, are mathematically unitary operators (mostly not commuting with one another) and as such can always be represented by unitary transformations of the sets of our oscillators. The matrices U of unitary transformations of complex amplitudes determine the twice bigger matrices M of canonical transformations of coordinates and momenta. Let $m=\omega=A=1$. For the two oscillators of a single Qbit, in the case of an instant perturbation,

$$\begin{pmatrix} q_0' \\ \dot q_0' \\ q_1' \\ \dot q_1' \end{pmatrix}=\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ -m_{12} & m_{11} & -m_{14} & m_{13} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ -m_{32} & m_{31} & -m_{34} & m_{33} \end{pmatrix}\begin{pmatrix} q_0 \\ \dot q_0 \\ q_1 \\ \dot q_1 \end{pmatrix},$$

$$\begin{aligned} 2m_{11}&=u_{00}+u_{00}^*, & 2m_{31}&=u_{10}+u_{10}^* \\ 2m_{12}&=-i(u_{00}-u_{00}^*), & 2m_{32}&=-i(u_{10}-u_{10}^*), \\ 2m_{13}&=u_{01}+u_{01}^*, & 2m_{33}&=u_{11}+u_{11}^* \\ 2m_{14}&=-i(u_{01}-u_{01}^*), & 2m_{34}&=-i(u_{11}-u_{11}^*). \end{aligned} \tag{4}$$

If the perturbation is not instantaneous but continues from t to t', then the phase shift, $u_{kl}\to u_{kl}\exp[i\omega(t'-t)]$, must be included in formula (4) for M, while in Eq. (2) U is unchanged. We have $q_k'(t')=A[c_k'(t')e^{i\omega t'}+c_k'^*(t')e^{-i\omega t'}]$ if $q_k(t)=A[c_k(t)e^{i\omega t}+c_k^*(t)e^{-i\omega t}]$. The implementation of logic gates in our system does not require (though it does not exclude) the use of auxiliary oscillators, because to arbitrarily transform a single Qbit whose two complex amplitudes have three independent parameters, we need only the following three operations.

$R^{(k_L)}(\phi)$, a phase rotation of a single oscillator k_L of Qbit L (if the Qbit is placed inside some numbered set of Qbits). There are two possible rotations, $k_L=0$ or 1. A parametric adiabatic perturbation $\delta\omega^2(t)$, where $t_1<t<t_2$, $\delta\omega^2(t_1)=\delta\omega^2(t_2)=0$, and $d(\delta\omega^2)/dt\ll\omega^3/\pi$, produces such a rotation (we omit subindex L),

$$\ddot q_k+[\omega^2+\delta\omega^2(t)]q_k=0,\quad u_{kk}=e^{i\phi},$$

$$u_{ij}=\delta_{ij}\quad \text{otherwise},$$

$$\phi=\int_{t_1}^{t_2}\delta\omega^2(t)dt/2\omega. \tag{5}$$

$C^{(k_l i_L)}(\varphi)$, connection of the two different oscillators, k_L and l_L of the Qbit L with the help of perturbation $\delta v^2(t)$ analogous to $\delta\omega^2$, with $u_{kk}=u_{ll}=\cos\varphi$, $u_{kl}=u_{lk}=i\sin\varphi$,

$$\ddot q_k+\omega^2 q_k+\delta v^2(t)q_l=0,\quad \ddot q_l+\omega^2 q_l+\delta v^2(t)q_k=0,$$

$$\varphi=\int_{t_1}^{t_2}\delta v^2 dt/2\omega. \tag{6}$$

For example, the Pauli operators and the frequently used Hadamard H gates (L omitted):

$$\sigma_x = R^{(l)}(-\pi/2)R^{(k)}(-\pi/2)C^{(kl)}(\pi/2),$$

$$\sigma_y = \sigma_z C^{(kl)}(\pi/2), \quad \sigma_z = R^{(l)}(\pi)R^{(k)}(0),$$

$$H \equiv \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = R^{(l)}(\pi/2)\sigma_z C^{(kl)}(\pi/4)\sigma_z R^{(l)}(\pi/2),$$

$$k=0, \quad l=1. \tag{7}$$

In concluding this description of single Qbits we should note that the identical oscillators represent only the simplest and most convenient way to construct an analog of a quantum bit and to demonstrate our main idea. In a more general conception of Qbits, oscillators may have different frequencies but the same $m\omega^2A^2$, and resonance perturbations and auxiliary oscillators (with the same $m\omega^2A^2$) can be used.

Looking at two Qbits, now, the combination of J and K, $J>K$, is not trivial only when the entangled states of these two Qbits are involved. If the Qbits have not interacted at the earlier stages of a given computation, then by definition their common state is the tensor product of the two corresponding states (3), $|\psi(x^{(J)})\rangle \otimes |\psi(x^{(K)})\rangle$. To implement a two-Qbit gate in the case of these two Qbits means to connect all four of their constituent oscillators, thus creating an entangled state. Mathematically, we first algebraically multiply our two one-Qbit states. Then we apply the needed 4×4 unitary matrix to the four initial complex amplitudes of the four eigenstates $|x^{(J)}\rangle \otimes |x^{(K)}\rangle$ produced by this multiplication, $x^{(J)}=0$ or 1, $x^{(K)}=0$ or 1. Physically in our system, we "multiply" the oscillators within each of the four pairs of oscillators by using device (c). (The necessity of such multiplication is a disadvantage of our system compared with quantum systems.) The multiplication is as follows. Suppose there are two oscillators, $x^{(J)}$ at the left and $x^{(K)}$ at the right, $J>K$, with the corresponding complex amplitudes $c_{x^{(J)}}$ and $c_{x^{(K)}}$. Then the product is the single oscillator having amplitude $c_{x^{(J)}x^{(K)}} = c_{x^{(J)}}c_{x^{(K)}}$ and the ordinal number $x^{(J)}x^{(K)}$, in which all integers are written in the binary system. When the two original oscillators represent number eigenstates, say, $|k\rangle$ at the left and $|l\rangle$ at the right, k, $l=0$ or 1, whose complex amplitudes therefore equal 1, the oscillator product also represents the number eigenstate, namely, $|kl\rangle$. So as a result of the "multiplication" we get four *single* oscillators now numbered as $0^{(J)}0^{(K)}$, $0^{(J)}1^{(K)}$, $1^{(J)}0^{(K)}$, and $1^{(J)}1^{(K)}$. (Upper indices are retained when it is important to remember where our two Qbits, J and K, are placed inside some bigger set of Qbits.) Now the number state

$$|\psi(x)\rangle = \sum c_x|x\rangle, \quad x=00,\ 01,\ 10,\ 11, \tag{8}$$

is represented by the ordered set of the four numbered oscillators to which we can apply the needed perturbation. It is sufficient to use four single phase rotations and six connections between any two single oscillators in Eq. (8). Consider, for example, the controlled-NOT (CNOT) gate, whose matrix elements are $u_{00,00}=u_{01,01}=u_{10,11}=u_{11,10}=1$, with other elements equal to zero. In our classical system, we can exchange oscillations of the two oscillators 10 and 11 (in binary notation) by applying a one-Qbit operation, as if these oscillators belong to the same single Qbit. (This is impossible in quantum systems because the mere selection of states inside a superposition may destroy the superposition.) By using perturbation (6), with $k=10$, $l=11$, and $\varphi=\pi/2$, and then individual rotations, we get in fact the σ_x operation,

$$X_{\text{CNOT}} = R^{(10)}(-\pi/2)R^{(11)}(-\pi/2)C^{(10,11)}(\pi/2). \tag{9}$$

which produces an entangled state because it exchanges oscillations of oscillators taken from different Qbits.

Next we will show how to implement one and two Qbit logical gates in the case of an entangled number-state $|\psi(x)\rangle = \Sigma c_x|x\rangle$ corresponding to some big number N of ordered Qbits, numbered as 0, 1,..., $(N-1)$ in decimal notation. Every number eigenstate $|x\rangle$ here is the tensor product of N number eigenstates $|x_I\rangle$, $x_I=0$ or 1, each taken from N different Qbits, and is represented by a single oscillator x, the result of multiplying the N corresponding oscillators:

$$|x\rangle = |x_{N-1},\ x_{N-2},\ldots x_J,\ldots,\ x_K,\ldots\ x_2,\ x_1,\ x_0\rangle. \tag{10}$$

Number $x=x_{N-1},\ldots,x_0$, expressed in the binary system, is a set of 0's and 1's. Their indices are the Qbits' own decimal numbers. Suppose, for example, that we have to rotate the phase of oscillator 1_K, that is, of oscillator 1 of Qbit number K. Exactly half of all 2^N possible x values in Eq. (10) have 1 at the K place in the binary notation, so half of the possible states in the superposition have $x_K=1$. In our system, we can easily select 2^{N-1} oscillators containing $x_K=1$ in their numbers and perform the one-Qbit rotations of their phases simultaneously while keeping other oscillators untouched. Consider now the two-Qbit gate $B^{(KJ)}$, for Qbits J and K, whose matrix elements are $u_{0_J0_K,0_J0_K}=u_{0_J1_K,0_J1_K}=u_{1_J0_K,1_J0_K}=1$; $u_{1_J1_K,1_J1_K}=\exp(i\pi/2^{J-K})$. This gate equally rotates phases of all those 2^{N-2} oscillators which correspond to the number-eigenstates of numbers x possessing l's at both J and K places in Eq. (10). We can implement this gate as one-Qbit rotations of all relevant 2^{N-2} oscillators simultaneously, while keeping other oscillators untouched.

To show how different elements of our system work together, let us go through several implementation steps of Shor's factorization algorithm [4]. We use a simple textbook example of factorizing the number 4, without any comments on the algorithm itself. In this case there are two registers—register X, RX, and register Y, RY—with four Qbits in the RX and two Qbits in the RY, with the prepared initial state $|0000,00\rangle$, that means, initially, $x=0$, $y=0$. In this case it is better *not* to multiply the original oscillators, which have amplitudes equal to 1, $x_0^{(3)}$, $x_0^{(2)}$, $x_0^{(1)}$, $x_0^{(0)}$ in the RX and $y_0^{(1)}$, $y_0^{(0)}$ in the RY.

Step 1. Apply four independent H gates [see Eq. (7)] to the RX. As a result, the amplitudes of every oscillator inside every RX Qbit become $c=1/\sqrt{2}$. Every single Qbit state in this register becomes a superposition, $|0\rangle \to (|0\rangle + |1\rangle)/\sqrt{2}$, with the full X state as the tensor product of four such superpositions.

Step 2. Apply the controlled-NOT two-Qbit gate [see Eq. (9)] to Qbits 0 at the *RX* and 1 at the *RY*. As explained above, to do so we must first use device (c) to multiply the number states of these Qbits, that is, the oscillators representing these number states. This produces two oscillator-products, $(1/\sqrt{2})|0^{(0)},0^{(1)}\rangle$ and $(1/\sqrt{2})|1^{(0)},0^{(1)}\rangle$. (The commas separate the *X* and *Y* registers.) The two-Qbit controlled-NOT gate applied to the four oscillators, $0^{(0)},0^{(1)}$; $0^{(0)},1^{(1)}$; $1^{(0)},0^{(1)}$; $1^{(0)},1^{(1)}$—two of which have zero amplitudes—exchanges the amplitudes of the oscillators $1^{(0)},0^{(1)}$ and $1^{(0)},1^{(1)}$. Since $1^{(0)},1^{(1)}$ has zero initial amplitude, we now get $c_{1^{(0)}0^{(1)}}=0$, $c_{1^{(0)}1^{(1)}}=1/\sqrt{2}$.

Step 3. Apply the σ_x operator [see Eqs. (6) and (7)] to Qbit 0 of the *RY*, to exchange the amplitudes of its two oscillators, 0 and 1. At this stage the change of the state is

$$|0000,00\rangle\rightarrow\tfrac{1}{4}(|0\rangle+|1\rangle)^{(3)}(|0\rangle+|1\rangle)^{(2)}(|0\rangle+|1\rangle)^{(1)}\times(|0^0,0^1\rangle+1^01^1\rangle)|1^0\rangle. \tag{11}$$

Step 4. Apply $H^{(3)}$ [see Eq. (7)], to the two oscillators of the *RX* Qbit 3. This leads to $(1/\sqrt{2})(|0\rangle+|1\rangle)^{(3)}\rightarrow\sqrt{2}|0^3\rangle$.

Step 5. Apply the $B^{(23)}$ gate to transformed Qbit 3 and Qbit 2 of the RX. For this we must first ''multiply'' their oscillators, producing $\sqrt{2}(|0^{(3)}0^{(2)}\rangle+|0^{(3)}1^{(2)}\rangle)$. The $B^{(23)}$ gate does not change this product because $|1^{(3)}1^{(2)}\rangle$ is absent (has zero amplitude) in this superposition.

After several more similar operations, we get the final state

$$|\psi(x,y)\rangle=\tfrac{1}{2}(|0000,01\rangle+|0001,01\rangle+|0000,11\rangle-|0001,11\rangle). \tag{12}$$

Only these four states survive the destructive interference caused by the perturbations following one after another. At this stage, Shor's algorithm requires *repeated* measurements of *x* by repeatedly preparing and then perturbing the same state. As we have already noted, our classical system does not require repeated reconstructions of the same initial states, since neither choices nor measurements destroy our number states. In Eq. (12), there are four oscillator products whose relative intensities (probabilities) equal 1/4, and two possible *x* values each with the probability 1/4+1/4=1/2. The use of this result to work through Schor's algorithm further is beyond the scope of this paper.

The author thanks the Open Society Institute for supporting this work, and Sidney Oriov for useful discussions.

[1] Michael A. Nielsen and Isaac L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press Cambridge, England, 2000).

[2] T. D. Lee (private communication); Roger Penrose *et al.*, *The Large, the Small and the Human Mind* (Cambridge University Press, Cambridge, England, 2000).

[3] Yuri F. Orlov, Phys. Rev. Lett. **82**, 243 (1999); Phys. Rev. A **65**, 042106 (2002).

[4] P. W. Shor, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE Press, Los Alamitos, CA, 1994).