17

FUNDAMENTAL ISSUES IN ATOMIC/NANOELECTRONIC COMPUTATION

M. P. Anantram and Vwani Roychowdhury

Dept. of Electrical Engineering
University of California
Los Angeles, CA 90095

Dedicated to Professor Thomas Kailath with great admiration and gratitude.

ABSTRACT

The past decade has seen tremendous experimental and theoretical progress in the field of mesoscopic devices and molecular self assembly techniques, leading to laboratory demonstration of many new device concepts. While these studies have been important from a fundamental physics perspective, it has been recognized by many that they may offer new insights into building a future generation of computing machines. This has recently led to a number of proposals for computing machines which use these new and novel device concepts. In this chapter, we explain the physical principles behind the operation of some of these proposals, primarily those based on conventional architectures. We then show that these proposals do not satisfy some fundamental requirements of a good computing model. The specific schemes discussed fall into the categories of **ground state** and **atomic** computing models. Finally, we propose a new scheme for atomic computing.

1 INTRODUCTION

The past decade has seen much experimental and theoretical study related to mesoscopic and atomic sized devices [1, 2, 3, 4]. Several groups in different countries have successfully built nanoscale devices, such as single-electron transistors and quantum dots, using both advanced lithographic techniques, as well as novel self-assembly methodologies, and have investigated the physics of these devices. The feature sizes of these devices are in the $1 - 100nm$ range.

An important question is : Can these new device concepts be used to build a future generation of computing machines? There have recently been many papers which address this question and they deal with proposals for both conventional and non conventional computing models. It is worth noting that architectures

A. Paulraj et al. (eds.), *Communications, Computation, Control, and Signal Processing*
© Springer Science+Business Media New York 1997

based on even the conventional models have remained rather elusive because it is widely believed in the solid state electronics community that the present strategies for integrating devices on a chip will be impractical for nanoelectronic devices due to their small size, low power handling capacity and low gain. The novel proposals which have recently been advanced, articulating primarily, visions of future atomic/nanoelectronic systems utilize unconventional architectures [5, 6, 7, 8, 9] and possess several characteristics suited to nanoelectronic/atomic implementation such as the absence of explicit interconnects and the convenience of loading (reading) the input (output) from the edges of the computing system. Some examples of the recently advanced non conventional computing models include neuromorphic [10] and quantum computing [8, 11, 12].

In this chapter, we mainly focus on the realization of conventional computing models using novel architectures involving atomic/nanoelectronic devices. We will begin with an explanation of the basic physics underlying two of the recently proposed paradigms, **ground state** and **atomic computing**. We will then show that these models do not fulfill some fundamental requirements of a good computing model. We will make concrete proposals for improvements, where possible.

Amongst the many features a good computing model should possess, the following three can be regarded as fundamental :

(i) the ability to interchange bit values between the basic units (bistable devices) in a manner independent of either the bit values being interchanged or the remaining bit values.

(ii) time efficient, unidirectional flow of information along the computational trajectory.

(iii) the ability to correct errors (we do not address issues related to error correction in this chapter).

In the novel paradigm of ground state computing [5, 6, 9], the computing architecture comprises locally-coupled arrays of basic units (Fig. 17.1). The basic units are bistable devices which represent binary values of 0 and 1 (section 2), and they interact with one another by Coulomb (or some such) interaction to form larger devices such as wires, logic gates and cellular automata (Fig. 17.1). Two features which distinguish their operation from conventional architectures are :

(a) They are edge driven. That is, the basic units on the edges form the input and output (Fig. 17.1). The interior units do not interact with the outside world.

(b) A computation is complete only when the computing machine relaxes to its ground state. The ground state depends only on the values of the input cells and not on the initial configuration of the remaining cells. The input cells are held fixed throughout the computation. The output to the

computation is read from the cells marked 'output' only after the system has settled in its ground state (Fig. 17.1).

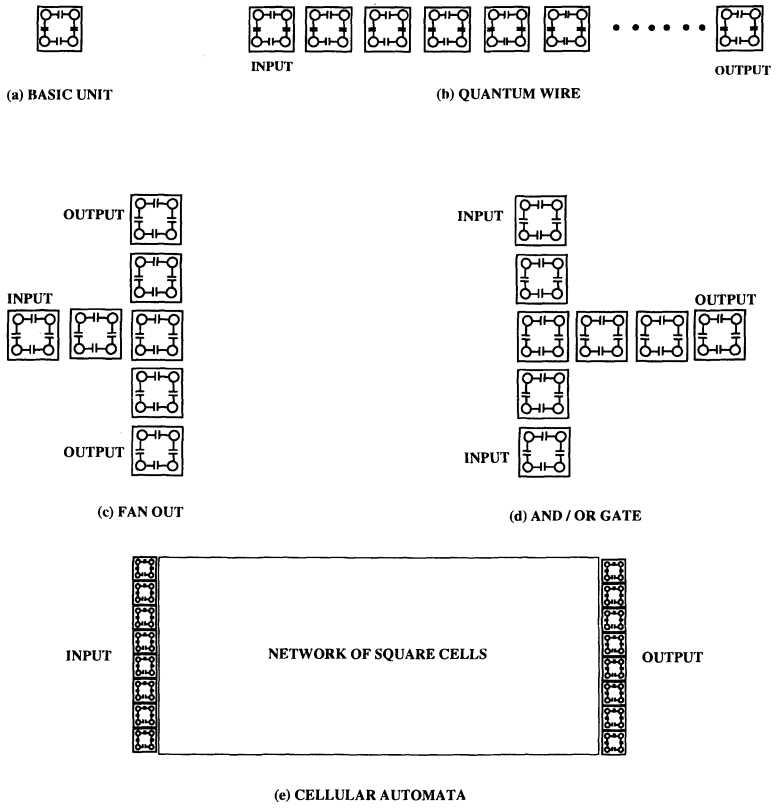


Figure 17.1 (a) Pictorial representation of the basic bistable unit. Construction of a (b) wire (c) fan out (d) AND / OR gates (d) using the basic unit (Ref. 4). The basic units interact with each other by Coulomb or some such interaction.

The requirement of evolution of the computing system to its ground state raises many important questions :

- Is a time efficient evolution of the computational trajectory to the ground state of the physical system involved feasible (requirement (ii) of page 310)? Or will the system be trapped by metastable states, thus delaying time efficient evolution to the ground state?
- Will the system remain in the ground state for a long enough time to be of computational use, before being excited to higher lying states?

There are no clear answers to these questions in the papers proposing ground state computing [5, 6, 9]. These papers only verified that the ground states of various structures correspond to logical operations. While there have been some hypotheses in literature that relaxation to the ground state will take place by a domino-type effect [6, 13], there have also been suggestions that a computing machine which relies on evolution to the ground state might not be feasible if metastable states are encountered in the computational trajectory (Fig. 17.2) [13, 14]. It is also of relevance to note that in other energy-based models of computation, such as the Ising spin-glass systems and the Hopfield Networks, meta stable states exist and play a definite role. The ground state in these schemes corresponds to an optimal solution for the problem under consideration and metastable states correspond to approximate solutions. In contrast, in the proposed models of ground state computing, metastable states (if they exist) do not have any useful computational interpretation.

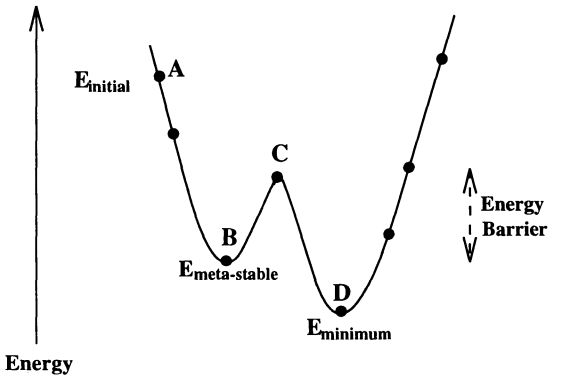


Figure 17.2 A system in the initial state (A) can reach the ground state (D) only by the series of single electron transitions, A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D. State B is a meta stable state as an energy barrier has to be surmounted to reach state C.

We have *resolved* the controversy involving the role of metastable states in ground state computing and have answered the above questions for the first time, by considering a specific model for ground state computing [15]. The salient features are discussed in section 2. We show that metastable states play an important role in determining the evolution of the computational trajectory, resulting in the time required for evolution to the ground state to be too large to be of computational use, even for the most advantageous choice of parameters.

Atomic Computing (Computing on an atomic chain) :

There has been excitement over the recent technological prospect of using an array of two level atoms as a computing machine, with the atoms serving as bistable devices [7, 8]. These proposals exploit the fact that the ground ($|0>$)

and excited ($|1>$) states of a two level atom can be used to represent binary values 0 and 1 as shown in Fig. 17.3 (a). ϵ^0 and ϵ^1 denote the ground and excited state energies. Switching between the two binary states ($|0>$ and $|1>$) can be accomplished by shining a π pulse. A π pulse has a resonant frequency equal to $\epsilon^1 - \epsilon^0$ and has an electric field profile shaped so as to cause an exact transition between states $|0>$ and $|1>$. Fig 17.3 (b) shows how to flip the state

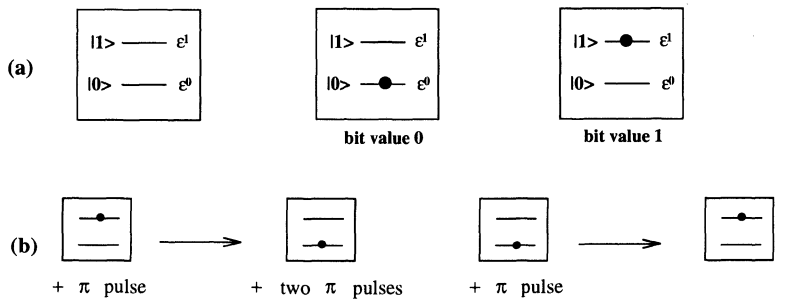


Figure 17.3 (a) A two level system. The ground and excited states states of the system can be used to represent binary bit values of 0 and 1. (b) By applying a π pulse with a resonant frequency equal to the resonant energy of the system, precise switching between binary numbers 0 and 1 can be accomplished.

of an isolated bistable device. To realize a computing machine, it is however necessary to have the ability to flip the state of a bistable device in a manner which depends on the bit values of a few other bistable devices. The interaction between atoms provides an adequate environment to achieve this. For example, consider two atoms A and B with distinct resonant frequencies interacting with each other. Now π pulses with distinct frequencies, $\omega^A_{B=0}$ and $\omega^A_{B=1}$ are required to flip the state of atom A, depending on the bit value stored in atom B. Infact, it can be shown that the basic logic operations such as AND, EX-OR and FAN OUT can be achieved using an array of three atoms with distinct resonant frequencies (section 3).

This line of thought was used in Ref. [8], which proposed that a polymer chain $A\ B\ C\ A\ B\ C\ \ldots\ A\ B\ C$ and a repertoire of π pulses comprises a computing machine (Fig. 17.4 (a)). Atoms A, B and C have distinct resonant frequencies. Such a system exhibits the following parallel feature : a π pulse would flip all atoms of a given kind with identical neighbors, throughout the chain. For example, in Fig. 17.4, if the bit values of A_2 and C_2 are identical to those of A_4 and C_4, then a π pulse which flips the bit value of atom B_2 will also flip the bit value of B_3. While the parallel feature can be used as an advantage in some applications, it is not a desirable feature in others (section 3). Specifically, the parallel feature proves to be the nemesis when it comes to satisfying requirement **(i)** discussed in page 310.

We propose a new computing model composed of two atomic chains which interact with each another (Fig. 17.4). This system does not suffer from the drawback discussed in the previous paragraph (section 3). Also, the parallel feature can be exploited when necessary in this model.

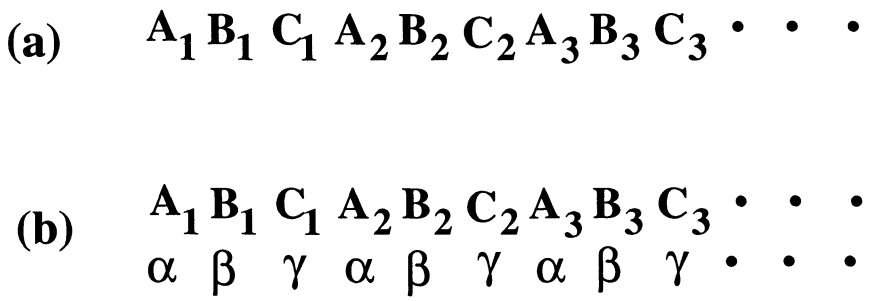


Figure 17.4 (a) A single polymer chain (b) Two interacting polymer chains : The various atoms of the chain on the top represent bistable devices. Only one atom of the chain on the bottom has a bit value of 1, the rest have a bit value of 0. This feature is used to uniquely mark one atom in the top chain. Note that the single polymer chain lacks this feature as all atoms of a given type (say B) are marked only by its neighbors (A and C) throughout the chain.

2 GROUND STATE COMPUTING

The previous studies of ground state computing only verified that the ground state of suitably designed coupled arrays of quantum dots corresponds to the basic logic operations. However, to resolve the controversy surrounding the role of metastable state in ground state computing models, it is necessary to study the dynamical evolution of the computational trajectory. In this section, we study this in the simplest network of quantum dots, a wire (Fig. 17.1). The quantum dots in our study are metallic type (continuum of energy levels) and have been successfully modeled by the orthodox theory of single electron tunneling [16]. The previous proposals of ground state computing considered dots with a single energy level using a Hubbard-type Hamiltonian and calculated only the energies of the ground state configurations [5, 6, 9]. These studies did not address the dynamics probably because the electron relaxation process in quantum dots with discrete energy levels is not well understood. Some recent studies however seem to indicate that the electron relaxation rate in quantum dots with discrete levels is orders of magnitude larger than in metallic type dots [17]. In which case, we expect our results to be indicative of dots with discrete levels also. The following specific issues are studied in this section :

- We first discuss the model used to study the dynamics of the computing system.
- Then, the existence of metastable states in the computational trajectory is established at zero temperature.
- By increasing the temperature, the ground state of the computing machine can always be accessed. However, for ground state computing to be feasible, it is necessary for the system to remain in the ground state for a long enough time period. Hence, we determine if there exists a region in the parameter space of the device dimensions and the temperature, where the system both relaxes to the ground state and remains there for a sufficiently long time period.
- Finally, it is verified if the time required to relax to the ground state is acceptable from a computational viewpoint, at least for the most advantageous choice of parameters.

2.1 The Model

Basic unit : The basic unit representing a bistable device, a *cell*, is shown in Fig. 17.5. It consists of four quantum dots along the edges of a square, which share two excess electrons. The four dots are separated by tunnel barriers whose resistances are much smaller than $\frac{h}{e^2}$ and in this limit, the dots have an integer number of electrons at any time. Two or more bistable devices interact via Coulomb interaction and this is modeled by capacitances. The non zero capacitances between the dots are shown in Fig. 17.1 (b). All other capacitances are zero. The intra and inter cell capacitances C and C', and the capacitance between the dots and the ground C_0 are identical in every cell of the wire. The tunnel resistances between the two dots along the sides of a cell are represented by R, and the tunnel resistances between dots along the diagonals of a cell and between dots of different cells are infinite.



Figure 17.5 Representation of a single binary bit. The square cell consists of four quantum dots which share two electrons. Electrons occupying diagonally opposite dots represent binary values of 0 and 1.

The lowest energy (ground) state of a single cell is two fold degenerate and it consists of electrons aligned either along the left are right diagonals. The two

degenerate ground states can be used to represent binary values 0 and 1 (Fig. 17.5). The ground state of a wire is also two fold degenerate and is shown in Fig. 17.6. Here the two excess electrons are aligned along the right (or left) diagonals in every cell.

System : The specific device studied here is the most fundamental one, a wire, which consists of an array of square cells (Fig. 17.1). The ground state is two fold degenerate and is shown in Fig. 17.6. Here the two excess electrons are aligned along the right (or left) diagonals in every cell.

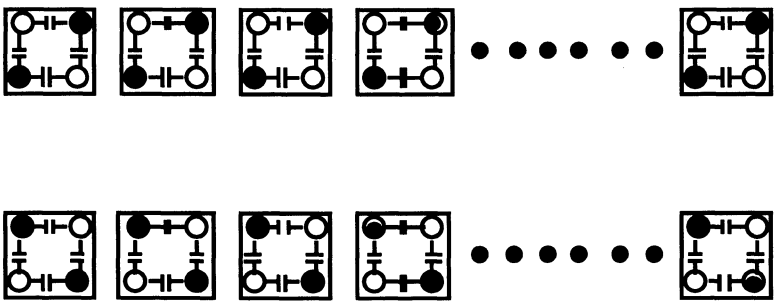


Figure 17.6 The ground state of a quantum wire is two fold degenerate. The two electrons are aligned either along the left or right diagonals as shown.

Initial state : Two excess electrons can be distributed in $4C_2$ ways in every cell. If there are N cells in the wire there are 6^N possible initial configurations. Our monte carlo simulator can handle any of these initial configurations. Here we will only consider the following initial state which is physical interesting. Consider the system to be in the ground state shown in Fig. 17.6 (a). In the initial state considered, the polarization of the left most cell alone is flipped (Fig. 17.7) and held fixed. The left most cell which represents the input is held fixed. If the wire relaxes to the ground state shown in Fig. 17.6 (b), then information is transmitted from the left most cell (the input) to the right most cell (the output).

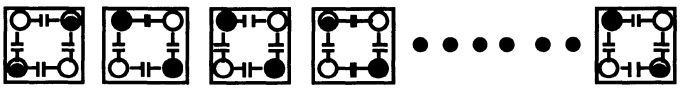


Figure 17.7 The initial state considered in this chapter.

Time evolution and the model

The system is set up in the initial state (Fig. 17.7). For the computation to be complete, the system should evolve into the ground state shown in Fig. 17.6.

The output is measured from the cell marked so after the system settles in the ground state.

The time evolution is modeled by the orthodox theory of single electron tunneling. [18] Here, a vector $Q(t)$ represents the charge in the quantum dots at time t. The charge configuration at the next time step $t + \delta t$ is computed using the monte carlo method which is briefly explained now. The transition probability for a single electron tunneling event where an electron tunnels from dot i to dot j is given by, [18]

$$\Gamma_{ij} = \frac{\Delta E_{ij} G}{C'} \frac{1}{exp(\frac{\Delta E_{ij}}{kT}) - 1}, \tag{17.1}$$

where, G and C' are the tunnel conductance and capacitance of the barrier between dots i and j respectively. $\Delta E_{ij} = E^a - E^b$, where E_a and E_b are total energies of the system after and before the tunneling event. Then a cumulative vector of all the tunneling probabilities,

$$S^p = \sum_{1}^{p} \Gamma_p, \tag{17.2}$$

is calculated. p labels the various tunneling events and can take values 1, 2, 3, .. ., M, where M represents the total number of tunneling events. A random number r is chosen between 0 and S^M. The charge vector Q in the next time step of the monte carlo simulation is determined by the single electron transition labeled n which satisfies $S^n < r < S^{n+1}$. The time taken for a single electron tunneling event is calculated using the expression $-\frac{\ln(r')}{S^M}$, where r' is a random number between 0 and 1 and S^M is the sum of all possible single electron transition probabilities from the initial state.

Metastable states : At zero temperature, a metastable state is reached if all the tunneling probabilities Γ_{ij} are zero (which implies that all $\Delta E_{ij} \geq 0$). At a finite temperature, we define the system to be trapped in a metastable state for a long time if S^M is smaller than $1e^{-8}\ (RC)^{-1}$.

2.2 Zero Temperature Results

We first discuss the simple cases of wires with two and three cells. It is useful to remember that when (i) $C' >> C$, the interaction energy is large for electrons in neighboring cells to be adjacent to each other and (ii) $C' << C$, the interaction energy is large for electrons in the same cell to be on adjacent dots.

Two cell case : The initial state is shown in Fig. 17.8(a). The ground state energy is the two cell case of Fig. 17.6. When $C' >> C$, it is energetically favorable for the electron in dot 5 to tunnel to dot 7 (state B of Fig. 17.8 (b)) because the large interaction energy between the electrons in dots 2 and 5 can be avoided. The ground state is now easily reached by an electron tunneling

from dots 8 to dot 6. When $C' << C$, the single electron tunneling events all lead to states where two electrons occupy adjacent dots of the right cell. All such states are energetically unfavorable in the $C' << C$ limit (the electrons like to align along the diagonals of a cell) and the system remains trapped in the initial state, which is a metastable state. So, there is a threshold value for $\frac{C'}{C}$ above which the system can reach the ground state and below which the system remains trapped in the initial state.

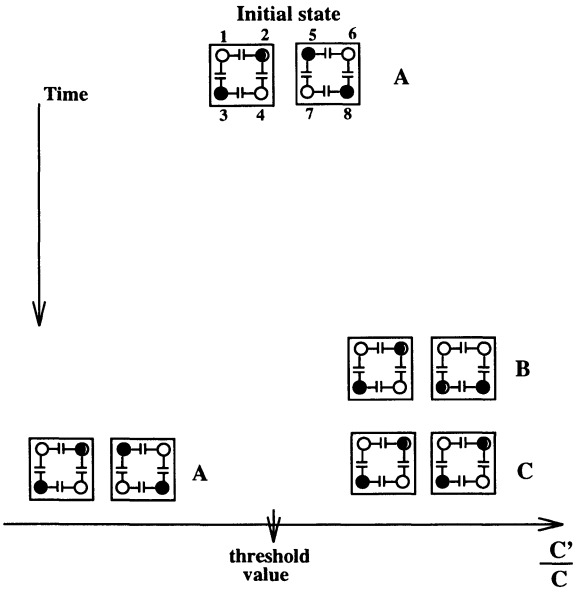


Figure 17.8 Evolution of the initial state as a function of $\frac{C'}{C}$.

Three cell case : Here, we have evaluated the total energies of various states to reach the following conclusions. For $C' >> C$, the interaction energy is very large for states where electrons are in adjacent dots of different cells. This interaction energy can be avoided only by the tunneling of an electron from dot 5 to dot 7 (Fig. 17.9). The remaining seven single electron tunneling events lead to total energies which are larger than that of the state A. Tunneling of an electron from dot 12 to dot 10 further reduces the total energy because it lowers the interaction between dots 8 and 12. The system is now trapped in state C as all the possible single electron tunneling events now only raises the total energy. So state C is a metastable state for $\frac{C'}{C}$ larger than threshold 2 (Fig. 17.9). In the other limit, where $\frac{C'}{C}$ is smaller than threshold value 1, as in the two cell case, the initial state is a metastable state. This is because all the eight possible single electron tunneling events lead to a state where electrons in either the second or the third cells are on adjacent dots. Such a configuration

has a large energy in the $C' << C$ limit. For the parameter space in between thresholds 1 and 2, we find that the system is trapped in state B which is a metastable state (Fig. 17.9).

For cells of longer lengths (up to sixty cells), we have verified by monte carlo simulation that a linear wire is trapped in a metastable states at zero temperature.

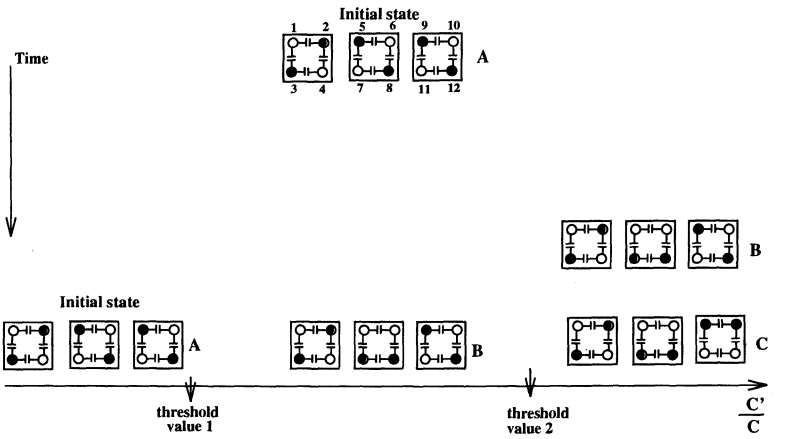


Figure 17.9 A plot of the various metastable states reached as a function of $\frac{C'}{C}$. The energies of various states have been calculated to verify the behavior.

2.3 Finite Temperature Results

We have so far shown that at zero temperature, ground state computing using a coupled array of quantum dots will suffer from problems associated with metastable states. We now ask, can the computing system escape from the metastable state, relax to the ground state and remain there for a long time period, at a finite temperature? At a high enough temperature, the system will fluctuate between various configurations including the ground state. For ground state computation to be feasible, it is however necessary for the system to remain in the ground state for a long enough time period. For this it is important that the temperature be large enough to *shake* the system out of a metastable state but not so large as to cause the system to leave the ground state in a short time period. We study the dynamics with the aim of determining if there is a regime of device parameters and temperature where this would happen. Then, we determine if the time required to relax to the ground state is acceptable from a computational viewpoint, at least for the most advantageous choice of parameters. This question can be addressed by a monte carlo

simulation of the electron dynamics [18]. A detailed discussion of the dynamics will be given elsewhere [15] and we only summarize our main results.

We compute the time taken to reach the ground state as a function of the temperature for wires of various lengths. The time taken to reach the ground state decreases exponentially with increase in temperature (Fig. 17.10). For temperatures larger than the threshold kT_0, the transition probability to leave the ground state is larger than the threshold rate of $1e^{-8}$ $(RC)^{-1}$. This means that the system stays in the ground state only for a time smaller than $1e^8$ RC. This happens because the transition probabilities increase exponentially with temperature (Eq. (17.1)). We also find that the threshold temperature kT_0 does not depend significantly on the length of the wire [15]. The wires relax to the ground state and remain there for an average time which is larger than $1e^8$ RC only in the narrow parameter range, $\frac{C'}{C}$ from approximately 1.1 and 1.35, for wires having from three to sixty cells.

¿From Fig. 17.10 (a), we see that a wire having 45 cells reaches the ground state in a time of 10^8 RC. For an RC of about 0.1 pico seconds, this corresponds to a time of 10 micro seconds. This time is too long to be useful from a computational viewpoint.

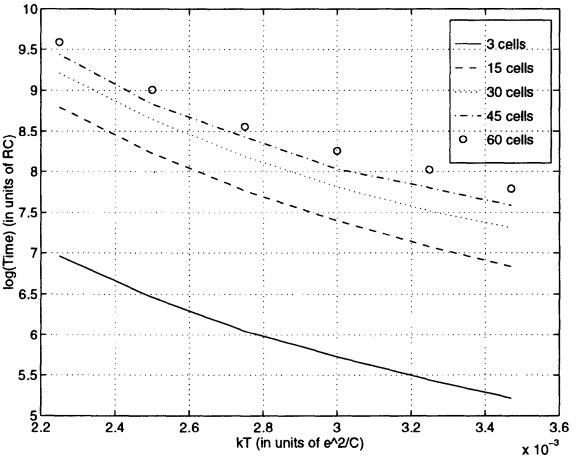


Figure 17.10 A plot of the time taken to reach the ground state versus temperature for wires of different lengths. The initial state is shown in Fig. 7. The value of $\frac{C'}{C}$ was chosen to be 1.3.

3 COMPUTING ON AN ATOMIC CHAIN

In the introduction, we explained the representation of binary values 0 and 1 using a two level atom. We will now address the following important issues related to building a computing machine using an array of atoms :

- **Construction of a logic gate from a set of interacting atoms.**
- **Where are the wires to connect to connect the 'atomic' gates?**
- **Is a single polymer chain $A\,B\,C\,A\,B\,C\,...\,A\,B\,C$ a general purpose computing machine? What are its shortcomings?**
- **A new proposal to build an 'atomic' computer is suggested.**

3.1 A Three Atom Logic Gate

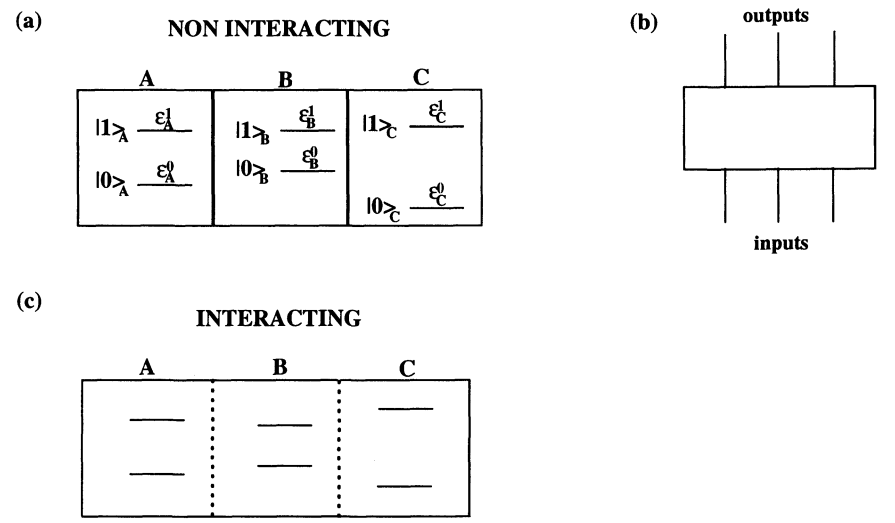


Figure 17.11 (a) A system of isolated atoms A, B and C. The resonant frequency of a atoms does not depend on the state of its neighbors. (b) A pictorial representation of a logic gate where the three output and input terminals are spatially separated from each other. (c) If atoms A, B and C interact via say the nearest neighbor interaction, the resonant frequency of an atom depends on the state of its nearest neighbors.

First, consider three two level atoms A, B and C isolated from one another (Fig. 17.11(a)). The (wave function, energy) of the ground and excited state of the individual atoms are represented by ($|0>_i$, ϵ_i^0) and ($|1>_i$, ϵ_i^1) respectively. The bit value of these atoms can be flipped using π pulses with resonant frequency, $\omega^i = \epsilon_i^1 - \epsilon_i^0$, where $i \in A,\ B,\ C$. Any 3-bit binary number can be represented

by this system in much the same way as they are represented by three devices in a conventional logic gate. Three wires in a conventional logic gate can however be fed as the 'input' to a black box which performs a logic operation, giving an output (Fig. 17.11(b)). For the output to be different from the input, it is important that the input terminals interact inside the black box. Also, the different logic gates are realized by using different interactions (different circuits) in the black box. All this can be emulated by a series of two level atoms if they are allowed to interact (Fig. 17.11(c)). A good starting point to include interactions between the different atoms is the much studied *nearest neighbor interaction* of solid state physics. Here, it is assumed that every atom interacts with only its immediate left and right neighbors except for the end atoms which have only a single nearest neighbor. As a result of the nearest neighbor interaction, the resonant frequency required to flip the bit value of atom i depends on the bit values of its nearest neighbors, atoms $i-1$ and $i+1$,

$$\begin{aligned}
\omega_A &- - - - > \omega_0^A,\ \omega_1^A \\
\omega_B &- - - - > \omega_{00}^B,\ \omega_{10}^B,\ \omega_{01}^B,\ \omega_{11}^B \\
\omega_C &- - - - > \omega_0^C,\ \omega_1^C
\end{aligned}$$

Here, ω_A, ω_B, and ω_C are the resonant frequencies of the isolated atoms. The subscripts on the right hand side refer to the state (binary bit value) of the nearest neighbors. For example, ω_0^A is the resonant frequency required to flip the bit value of atom B when atom C has a bit value of 0. Similarly, ω_{01}^B is the resonant frequency required to flip the bit value of atom B when atoms A and C have bit values of 0 and 1 respectively.

Now a Toffoli gate[1], with atoms A and C as controls can be realized by applying a π pulse with frequency ω_{11}^B. The state of atom B is then inverted only if both atoms A and C have a bit value of 1.

An important difference between conventional and 'atomic' logic gates : There is an important difference between the spatial location of the input and output terminals of conventional logic gates and the system under consideration. In a conventional representation of a logic gate (Fig. 17.11(b)), the input and output terminals are spatially separated. On the other hand for the system in Fig. 17.11 (c), both the input and output terminals are represented by the same set of atoms. The bit values of the atoms at two different times, prior to and after shining the desired sequence of pulses represent the input and output.

[1] A Toffoli gate is a universal gate with equal number of input and output terminals. Other logical gates such as the AND, NAND etc. can be constructed from copies of it as discussed in [19].

3.2 The Wiring Paradigm in Atomic Computers

A general purpose computing machine can be viewed as a circuit of a large number of Toffoli gates. A 'wiring' scheme for a computing machine should possess the following rather obvious features :

(a) It must be possible to perform local operations on a few terminals at a time without changing the state of the remaining terminals

(b) It must be possible to 'wire' together terminals far away from each other so that a controlled operation such as the Toffoli gate can be realized using these terminals (Fig. 17.12) (requirement (i) of page 310).

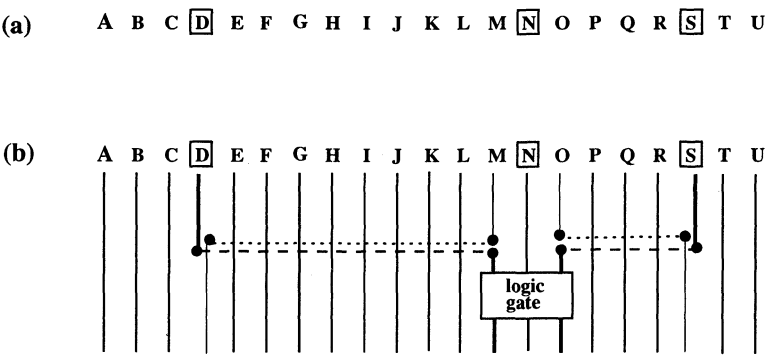


Figure 17.12 The terminals in this figures are labeled A through U. (a) The three 'boxed' terminals form the input to a logic gate in the next stage of the computation. How can the terminals be 'wired' in an array of interacting atoms. (b) In a conventional VLSI circuit this is done by routing the wires appropriately. The solid, dashed and dotted wires are patterned on three different layers of the circuit. So they are non intersecting except for vertical connection between the layers where necessary.

We have shown that the Toffoli operation can be performed on terminal i with terminals $i-1$ and $i+1$ as controls. So, point (a) above is satisfied and we will now discuss point (b). In conventional VLSI circuits, physical wires are 'routed' in and between different layers of the circuit to realize an arbitrary 'wiring' scheme. For example, in Fig. 17.12(b) wires **D**, **N** and **S** can be routed as shown to form the inputs to a logic gate. The solid, dotted and dashed lines are wires in three different layers of the circuit and they only intersect at the vertical connections between the layers, which are located at the solid dots. In an array of atoms, physical wires connecting the atoms does not make sense. However, the role of physical wires can be emulated, if the bit values of *any* two terminals can be interchanged in a systematic manner, which does not either change or depend on the detailed information of the bit values of the

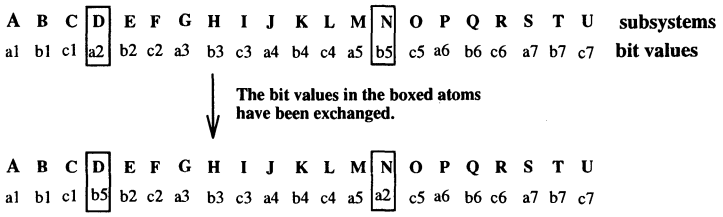


Figure 17.13 To accomplish a wiring scheme such as the one in the previous figure, it is necessary to be able to interchange the bit values of any two terminals far apart from each other (terminals D and N in this figure) without using and changing the bit values of the remaining terminals.

other atoms in the array (Fig. 17.13). Going a step further, we assert that this would be possible if the bit values of *adjacent* terminals can be interchanged in a systematic manner which does not either change or depend on the bit values of the remaining terminals. The interchange of bit values of a pair of terminals far away from each other can then be realized by a series interchanges of the bit values of adjacent atoms.

4 IS A POLYMER CHAIN $A\,B\,C\,A\,B\,C\,..\,A\,B\,C$ A GENERAL PURPOSE COMPUTING MACHINE?

We now consider an array of atoms $A\,B\,C\,A\,B\,C\,..\,A\,B\,C$, where the constituent atoms A, B and C have different resonant frequencies (the subscripts in Fig. 17.4 (a) are only used to label the positions of the atoms). Ref. [8] proposed that such an atomic array and a repertoire of π pulses with appropriate resonant frequencies comprises a computing machine. We will show here that this cannot be the case because point (b) discussed above is not satisfied. If light pulses can be localized to the size of a single atom, the bit values of any two adjacent terminals can be interchanged as required by point (b), by shining the appropriate sequence of pulses. This follows because atoms A, B and C have different resonant frequencies and so distinct pulses are required to interchange the bit values of atoms i and $i+1$, and atoms i and $i-1$. Unfortunately, the typical sizes of individual atoms is in the range of tens of Angstroms and their excitation frequencies are usually in the meV range. At these frequencies, the spatial extent of a light pulse is of the size of many atoms put together. Then, it is relevant to consider schemes where the light pulses shine over the entire atomic array. Due to this unavoidable restriction, point (b) cannot be satisfied. This is because a series of pulses which purports to interchange the bit values of B_i and C_i will also interchange the bit values of

all other B_j and C_j atoms, where j is a number labeling the various triplets of the array. The main drawback of the single chain model is that it lacks the ability to target (or uniquely mark) a single atom in the chain.

4.1 A New Proposal For an Atomic Computing Machine

Are there any schemes which would not suffer from the drawbacks of the $A\ B\ C\ A\ B\ C\ ...\ A\ B\ C$ chain? A system which does not suffer from the drawback of the previous section is a chain where all the constituent atoms are different from each other. The atoms all have distinct resonant frequencies and a pulse or sequence of pulses with a large spatial extent can be designed to satisfy points (a) and (b) of the previous subsection. This scheme does not suffer from any logical problems but there are two drawbacks. Firstly it is difficult to build a system where the constituent atoms are all different. And secondly, a repertoire of light pulses with too many resonant frequencies will be required.

The desirable features (a) and (b) (of the previous sub section) can be realized in a model consisting of two chains $A\ B\ C\ A\ B\ C\ ...\ A\ B\ C$ and $\alpha\ \beta\ \gamma\ \alpha\ \beta\ \gamma\ \ldots\ \alpha\ \beta\ \gamma$, which interact with each other (Fig. 17.4). Each atom in the top (bottom) chain interacts with only the atoms to its immediate left, right and bottom (top). The atoms of the first chain ($A\ B\ C\ A\ B\ C\ ...\ A\ B\ C$) comprise the terminals of the computer. The second chain ($\alpha\ \beta\ \gamma\ \alpha\ \beta\ \gamma\ \ldots\ \alpha\ \beta\ \gamma$) is used to mark a single atom in the first chain, a feature which the single chain model lacked. This is done by allowing only one of the atoms in the second chain to have the bit value of 1, the remaining atoms have a bit value of 0. This system has the following resonant frequencies :

$$\omega^A \text{ - - - - > } \omega^{A:end}_{B;\alpha} \text{ for the left most A of the array}$$
$$\omega^{A}_{CB;\alpha} \text{ for all other A}$$
$$\omega^B \text{ - - - - > } \omega^{B}_{AC;\beta}$$
$$\omega^C \text{ - - - - > } \omega^{C:end}_{B;\gamma} \text{ for the right most C of the array}$$
$$\text{ - - - - > } \omega^{C}_{BA;\gamma} \text{ for all other C}$$
$$\Omega^\alpha \text{ - - - - > } \Omega^{\alpha:end}_{\beta;A} \text{ for the left most } \alpha \text{ of the array}$$
$$\Omega^{\alpha}_{\gamma\beta;A} \text{ for all other } \alpha$$
$$\Omega^\beta \text{ - - - - > } \Omega^{\beta}_{\alpha\gamma;B}$$
$$\Omega^\gamma \text{ - - - - > } \Omega^{\gamma:end}_{\beta;C} \text{ for the right most } \gamma \text{ of the array}$$
$$\text{ - - - - > } \Omega^{\gamma}_{\beta\alpha;C} \text{ for all other } \gamma$$

As the left most A and α atoms and the right most C and γ atoms have only one nearest neighbor on the same chain, their resonant frequencies are different from the A, α, C and γ atoms in the interior of the chain.

Any computation can be done on this array by : (i) initializing the array to correspond to the input and (ii) applying a series of π pulses to perform the computation. The first step corresponds to initializing the input terminals of a computing machine and the second step is equivalent to passing these input values through a sequence of logical gates.

We end this discussion by clearly giving the rules for exchanging the bit values of any two atoms (requirement (i) of page 310) without using or changing the bit values of the remaining atoms.

Rules for exchanging the bit values of any two atoms :

Rule 1 (Marker Manipulation) Only one of the atoms in the second row is in the 1 state. This atom constitutes the marker. The marker can be moved to the right (left) by applying a sequence of pulses which will interchange the 1 state of the atom currently serving as the marker with the 0 state of the atom to its right (left). Further, what is important is that the marker can be moved independent of the bit values in Row 1.

The *pulse coding* for exchange of bit information between atoms i and $i+1$ is

$$[\ \omega^{i+1}_{11;0}\ \omega^{i+1}_{10;0}\ \omega^{i}_{01;0}\ \omega^{i}_{11;0}\ ][\ \omega^{i+1}_{11;0}\ \omega^{i+1}_{10;0}\ \omega^{i}_{01;1}\ \omega^{i}_{11;1}\ ]$$
$$[\ \omega^{i+1}_{11;1}\ \omega^{i+1}_{10;1}\ \omega^{i}_{01;0}\ \omega^{i}_{11;0}\ ][\ \omega^{i+1}_{11;1}\ \omega^{i+1}_{10;1}\ \omega^{i}_{01;1}\ \omega^{i}_{11;1}\ ]$$

$\omega^{i}_{xy;z}$ is used to represent the resonant frequency of atom i in Row 2 if its left, right and top (in Row 1) neighbors have bit values of x, y and z respectively. The pulse sequences in the first, second, third, and fourth bracket move the marker to the right if the atoms in Row 1 located right on top of atoms i and $i+1$ in Row 2 have a bit values of (0 and 0), (1 and 0), (0 and 1), and (1 and 1) respectively. By applying all four pulse sequences, the position of the marker will move one unit to the right, independent of the bit values of the atoms in Row 1. ω is used to represent the the excitation frequencies of atoms in Row 2. A similar sequence of pulses can move the marker to the left by one position.

Rule 2 (Exchange of any two bits - 'wiring') :

Bit values in Row 1 ... v_1 v_2 v_3 X_1 y_1 y_2 ... y_{N-1} y_N Z_1 w_1 w_2 w_3 ...
Bit values in Row 2 ... 0 0 0 1 0 0 ... 0 0 0 0 0 0 ...

can be changed to

Bit values in Row 1 ... v_1 v_2 v_3 y_N y_1 y_2 ... y_{N-1} X_1 Z_1 w_1 w_2 w_3 ...
Bit values in Row 2 ... 0 0 0 0 0 0 ... 0 1 0 0 0 0 ...

by the following sequence of pulses : (a) Place the marker at the location of X_1 using Rule 1. Interchange x_i with x_{i+1} in Row 1 if the marker is at location i and then move the marker to $i+1$ (using Rule 1). Do this N times and then move the marker one unit to the left. This leaves the marker at y_N. (b) Interchange x_i and x_{i-1} in Row 1 if the marker is at location i and then move

the marker to $i-1$. Do this N times and then move the marker N steps to the right.

The *pulse coding* for exchange of bit information between atoms i and $i+1$ in Row 1 conditional to the fact that the marker is at position i in Row 2 is:

$$[\ \Omega^i_{01;1}\ \Omega^i_{11;1}\ (\text{Rule 1 (1R)})\ \Omega^{i+1}_{11;1}\ \Omega^{i+1}_{10;1}\ (\text{Rule 1 (1L)})\ \Omega^i_{01;1}\ \Omega^i_{11;1}\ ].$$

$\Omega^i_{xy;z}$ represents the resonant frequency of atom i in Row 1 if its left, right and bottom (in Row 2) neighbors have bit values of x, y and z respectively. (Rule 1 (1R [L])) means apply Rule 1 to move the marker one unit to the right [left].

5 CONCLUDING REMARKS

A general purpose atomic / nanoelectronic computing machine must have the following characteristics : **(i)** the ability to interchange bit values between the basic units in a manner independent of either the bit values being interchanged or the remaining bit values. **(ii)** time efficient, unidirectional flow of information along the computational trajectory. **(iii)** the ability to correct errors.

In this chapter, we resolved some controversial issues surrounding points (i) and (ii) above in the recently proposed paradigms of **ground state** and **atomic computing**. We have also proposed a new scheme for atomic computing which does not suffer from the drawbacks of a single chain model.

Conclusion 1 : The controversy surrounding the role of metastable states in ground state computation has been resolved for a particular model involving an array of metallic type dots. We find that ground state computing models can be plagued by metastable states. We show that, at zero temperature, the computing machine can be trapped by metastable states for very long time periods, thus hindering relaxation to the ground state of the machine. In terms of computation, this implies inefficient flow of information from the input to the output. By raising the temperature, the ground state can be reached. However, for ground state computing to be feasible, it is necessary for both the computing machine to remain in the ground state for a long time period and for the computational trajectory to lead to the ground state in a short time period. For the model considered, we have shown that there is a narrow parameter regime of the device dimensions and the temperature, where relaxation to the ground state is possible for long time periods. Unfortunately, the physical time required to relax to the ground state is too large to be of any computational use.

There have been some studies indicating that the relaxation time for electrons in quantum dots with discrete energy levels is orders of magnitude larger than for metallic type dots with a continuum of energy levels [17]. So, even though we study a model with metallic type dots, we expect our results to be valid in the limit of dots with discrete energy levels also.

Conclusion 2 : The recently proposed computing paradigms, which use two level atoms as bistable devices, presents us with a future direction in building 'atomic' computers [8]. We showed that a single polymer chain of the form $A\ B\ C\ A\ B\ C\ ...\ A\ B\ C$, does not comprise a general purpose computing machine because an arbitrary 'wiring' scheme cannot be realized in this system. The reason for this is related to the inability to interchange the bit values of two adjacent bistable devices (atoms) in a manner consistent with requirement (i). We propose a new model for 'atomic' computers consisting of two arrays of atoms interacting with one another, and show that requirement (i) is satisfied by this model. We present explicit rules using π pulses, to interchange bit values of two atoms in a manner consistent with requirement (i).

Acknowledgements

This work was supported in part by NSF Grants No. ECS-9308814 and No. ECS-9523423.

REFERENCES

[1] C. W. J. Beenakker and H. van Houten. *Quantum Transport in Semiconductor Nanostructures, Solid State Physics, vol. 44, Eds. H. Ehrenreich and D. Turnbull.* Academic Press, Inc., U S A, 1991.

[2] Claude Weisbuch and Borge Vinter. *Quantum Semiconductor Structures, Fundamentals and Applications.* Academic Press, Inc., San Diego, U.S.A, 1991.

[3] Hermann Grabert and Michel H. Devoret. *Single Charge Tunneling, Coloumb Blockade Phenomena in Nanostructures.* Plenum Press, New York, U.S.A, 1992.

[4] In Ari Aviram, editor, *Molecular electronics – science and technology : St. Thomas, Virgin Islands 1991*, New York, U.S.A, 1992. American Institute of Physics.

[5] P. Bakshi, D. A. Broido, and K. Kempa. *J. Appl. Phys.*, 70:5150, 1991.

[6] Craig Lent, P. Douglas Tougaw, Wolfgang Porod, and Gary H. Bernstein. Quantum cellular automata. *Nanotech.*, 4:49, 1993.

[7] H. Korner and G. Mahler. Optically driven quantum networks : Applications in molecular electronics. *Phys. Rev. B*, 48(4):2335, 1993.

[8] S. Llyod. A potentially realizable quantum computer. *Science*, 261:1569, 1993.

[9] S. Bandyopadhyay, B. Das, and A. E. Miller. Supercomputing with spin polarized single electrons in a quantum coupled architecture. *Nanotechnology*, 5:113, 1994.

[10] Vwani Roychowdhury, D. B. Janes, S. Bandyopadhyay, and Xiaodong Wang. Collective computational acitivity in in self-assembled arrays of quantum dots: A novel neuromorphic architecture for nanoelectronics. *To Appear in IEEE transactions on electron devices.*

[11] Proceedings of the workshop on physics and computation physcomp '94. In *IEEE International Electron Devices Meeting*, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[12] J. I. Cirac and P. Zoller. Quantum computation with cold trapped ions. *Phys. Rev. Lett.*, 74(20):4091, 1995.

[13] S. Bandyopadhyay. *Preprint.*

[14] Rolf Landauer. *Phil. Trans. Royal Soc. London, Ser. A.* To Appear in.

[15] M. P. Anantram and V. Roychowdhury. Preprint.

[16] D. V. Averin and K. K. Likharev. Single electronics : A correlated transfer of single electrons and cooper pairs in systems of small tunnel junctions. In B. L. Altshuler, P. A. Lee, and R. A. Webb, editors, *Mesoscopic Phenomena in Solids*, U. S. A, 1991. Elsevier Science Publishers.

In B. L. Altshuler, P. A. Lee, and R. A. Webb, editors, *Mesoscopic Phenomena in Solids*, U. S. A, 1991. Elsevier Science Publishers.

[17] H. Benisty. Reduced electron-phonon relaxation rates in quantum-box systems: Theoretical analysis. *Phys. Rev. B*, 51(19):13281, 1995.

[18] N. S. Bakhalov, G. S. Kazacha, K. K. Likharev, and S. I. Serdyokova. *Sov. Phys. JETP*, 68(3):581, 1989.

[19] Richard P. Feynman. Quantum mechanical computers. *Foundations of Physics*, 16(6):507, 1986.