Quantum Simulation of Several-Particle Systems

P. Douglas Tougaw, Craig S. Lent, and Wolfgang Porod

Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556 U.S.A.

Abstract

We study the simulation of quantum cellular automata and how such a simulation is simplified by the features of FORTRAN-90. We demonstrate the use of second-quantized operators to write the cell Hamiltonian and explain the Hartree self-consistent method for simulating a many-cell system. Finally, several examples of simulated QCA devices are shown.

I. Introduction

The particular quantum system we simulate consists of several of the quantum cells shown in Fig. (1a). We determine the ground state of the system by solving the time-independent Schrödinger equation. Each cell consists of five coupled quantum dots which contain a total of two electrons. The cells only interact with each other Coulombically; no tunneling is allowed between cells. Since the state of each cell is affected by its nearest neighbors, we call such a system a quantum cellular automaton (QCA).

Because of Coulombic repulsion between the two electrons in each cell, the charge density exhibits strongly bistable behavior. The ground state of the cell is therefore in one of the two states shown in Fig. (1b). Because of this bistable nature, we can use the state of each cell to encode binary information. We define a cell polarization which measures to what extent the cell is in one of the two stable states shown in Fig. (1b):

$$P \equiv \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{\rho_0 + \rho_1 + \rho_2 + \rho_3 + \rho_4} \tag{1}$$

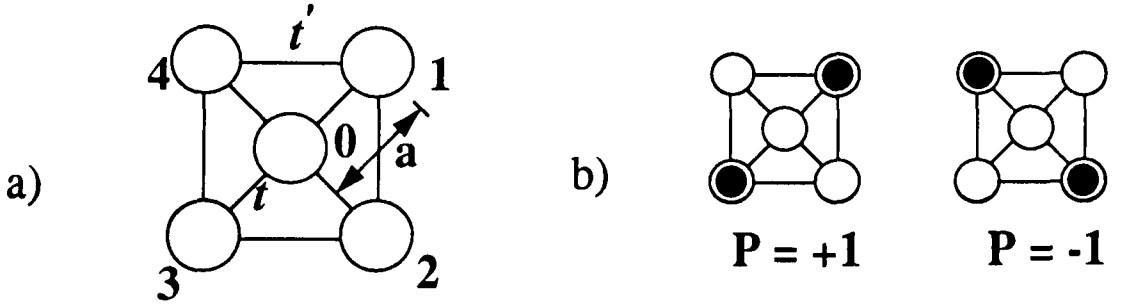

Figure 1. Schematic of the basic five-site cell. (a) The geometry of the cell with t=0.3meV, t'=t/10, and a=20 nm. (b) Coulombic repulsion causes the electrons to occupy antipodal sites within the cell. The two bistable states have cell polarizations of P=+1 and P=-1(See equation (1)).

II. Second-Quantized Hamiltonian

We define a second-quantized annihilation operator, $\hat{a}_{i,\sigma}$, which destroys an electron on site i with spin σ, and a creation operator, $\hat{a}^{\dagger}{}_{i,\sigma}$, which creates a particle on site i with spin σ. The product of these two operators, $n_{i,\sigma} = \hat{a}_{i,\sigma}\hat{a}^{\dagger}{}_{i,\sigma}$, is the number operator for that site and spin.

Using these operators we can compactly write the Hubbard-type tight-binding Hamiltonian of a single isolated cell:

$$H_0^{cell} = \sum_{i,\sigma} E_0 n_{i,\sigma} + \sum_{i>j,\sigma} t_{i,j}\left(a_{i,\sigma}{}^{\dagger} a_{j,\sigma} + a_{j,\sigma}{}^{\dagger} a_{i,\sigma}\right) +$$

$$\sum_{i} E_Q n_{i,\ } n_{i,\ } + \sum_{i>j,\sigma,\sigma'} V_Q \frac{n_{i,\sigma} n_{j,\sigma'}}{|\vec{R}_i - \vec{R}_j|} \tag{2}$$

This Hamiltonian includes on-site energies, tunneling between sites, on-site charging costs, and Coulombic repulsion between each pair of sites. The interaction with neighboring cells alters the on-site energies in the first term.

III. Unique Features of Fortran-90

Fortran-90 supports a level of data abstraction sufficient to allow direct implementation of these second-quantized operators and the related state vectors. We have created user-defined types representing creation and annihilation operators and many-electron site kets and bras. We also provide functions to convert between these types and to define the effect of each operator on all other data types.

The second useful feature of Fortran-90 for our purpose is operator overloading. This allows us to use an operator without regard to the data types it acts upon. We then provide an interface that invokes the appropriate function based on the data types involved. In this case, the action of a creation or annihilation operator on a Dirac ket in the site representation is specified. The operation is "overloaded" onto the normal multiplication symbol. A similar overloading specifies that multiplication of a Dirac bra and ket be interpreted as the inner product of the state vector.

Fig. (2) shows a segment of Fortran-90 code that demonstrates how easily quantum mechanical expressions can be written as Fortran code once these definitions are in place. The code is easily understandable because the level of data abstraction matches the level of quantum mechanical abstraction.

$$\langle i_1 | \sum_{i,j,\sigma} t(a^\dagger_{i,\sigma} a_{j,\sigma} + a^\dagger_{j,\sigma} a_{i,\sigma}) | i_2 \rangle$$

⇩

```
! -- Hopping terms --
 do ist=1,NSTATES
   do jst=1,NSTATES
    do ispl=1,2
  H(il,i2)=H(il,i2)+ket2bra(baseket(il))*(T0(ist)* &
   &ad(ist,ispl)*a(jst,ispl)*baseket(i2))
  H(il,i2)=H(il,i2)+ket2bra(baseket(il))*(T0(ist)* &
   &ad(jst,ispl)*a(ist,ispl)*baseket(i2))
     end do
   end do
 end do
```

Figure 2. Conversion from second-quantized quantum mechanical expression to FORTRAN-90 code using data abstraction techniques.

IV. Hartree Self-Consistent Calculations

Since electrons are not allowed to tunnel between cells, we can solve for the ground state of each cell separately. Such intracellular calculation includes exchange and correlation effects exactly. The interaction between cells is included using a Hartree self-consistent technique. Once the iterative solution of the system has converged, the system is in an eigenstate. Use of several different initial conditions and comparison of the eigen-energies allows us to determine the overall ground state of the system.

V. Application: Quantum Cellular Automata

We have used this scheme to simulate many arrangements of quantum cells. The most fundamental of these calculations is shown in Fig. (3a). The system consists of two cells as shown in the inset. The charge density of cell 2 is fixed, and the induced polarization in cell 1 is then calculated. This is repeated for many values of P_2 in the range [-1,+1] and the induced polarization P_1 can then be calculated as a function of P_2. This cell-cell response function demonstrates the highly nonlinear and bistable nature of the cell's response to its neighbors.

Fig. (3b) shows a similar cell-cell response function calculated at several non-zero temperatures. This requires calculating the thermal expectation value of the polarization over the canonical ensemble. As seen in the figure, the nonlinearity of the response degrades as the temperature increases.

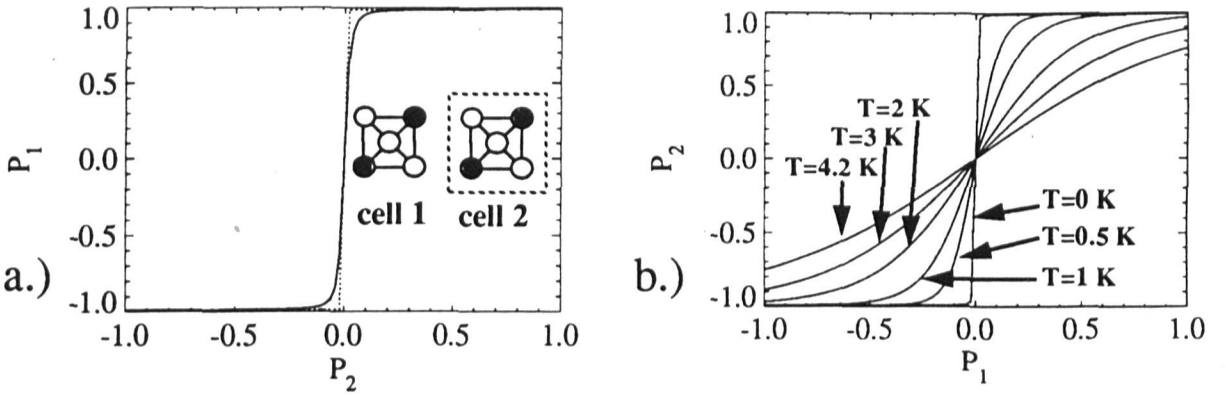

Figure 3. Cell-cell response functions. (a) The cell-cell response function for the basic five-site cell at zero temperature. The solid line corresponds to the antisymmetric case, and the dotted line to the symmetric case. (b) The temperature dependence of the cell-cell response function.

Fig. (4) shows three of the basic QCA devices. Fig. (4a) demonstrates that even a weakly polarized cell can drive a line of similar cells and that the bistable saturation of the cell-cell response will return the signal to maximum polarization in subsequent cells. Fig. (4b) demonstrates that a signal will propagate through a right-angle turn without degradation, and Fig. (4c) shows that a single line of cells can fan out to multiple lines and maintain signal integrity.

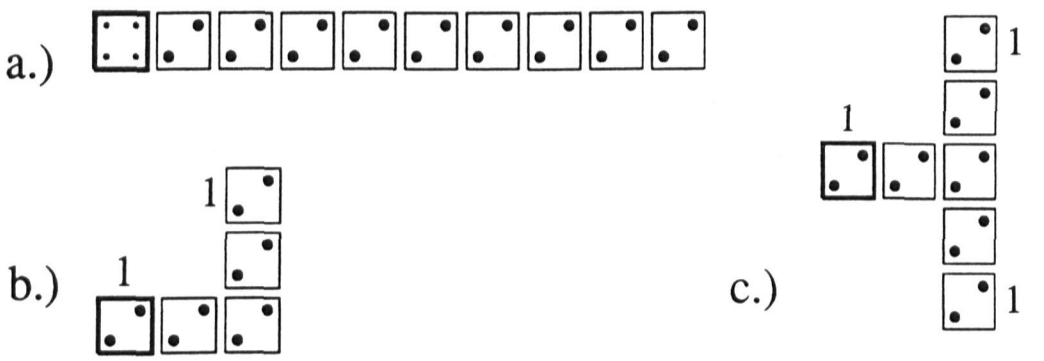

Figure 4. Three basic QCA devices. (a) A line of cells can be used to transmit information from one point to another. (b) The signal is transmitted correctly around a corner. (c) A single line can fan out correctly to multiple lines with the same signal. These are not schematic diagrams; they are plots of the actual results of the self-consistent calculation of the ground state of each system. The diameter of each dot is proportional to the charge density on the site. The cells with heavy borders are fixed; all others are free to react to the fixed charge.

Fig. (5) shows that the state of a free cell matches the majority of its fixed neighbors. This majority voting logic can provide the basis of a new computing architecture. If one of the fixed neighbors is called the "program line", such an arrangement of cells can be interpreted to be a programmable AND/OR gate. The program line determines the nature of the gate (AND vs. OR), and the other inputs are applied to the gate thus defined.

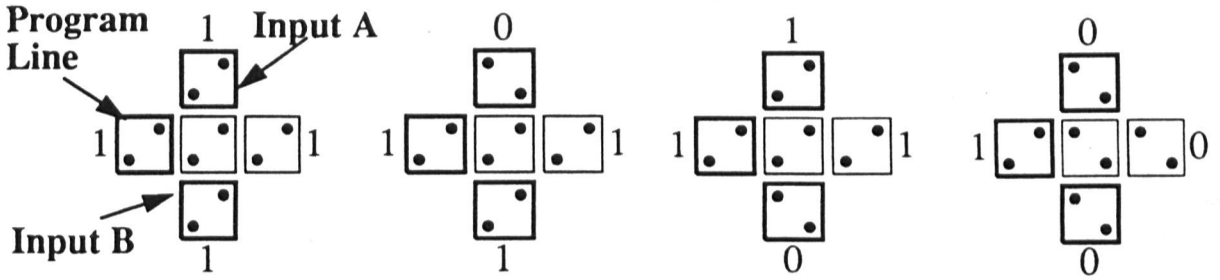

Figure 5. The programmable AND/OR gate. Here, the program line is set to one, so the other two inputs are being applied to an OR gate. In each of the four cases, the output is one if either of the two inputs are one. This is a plot of the result of a self-consistent ground state calculation.

Finally, Fig. (6) shows how to cross two lines of cells without having the signals interfere. Wire crossings are very important for implementation of devices like adders and exclusive-OR gates. Such a quasi-two-dimensional crossing is impossible with conventional devices.

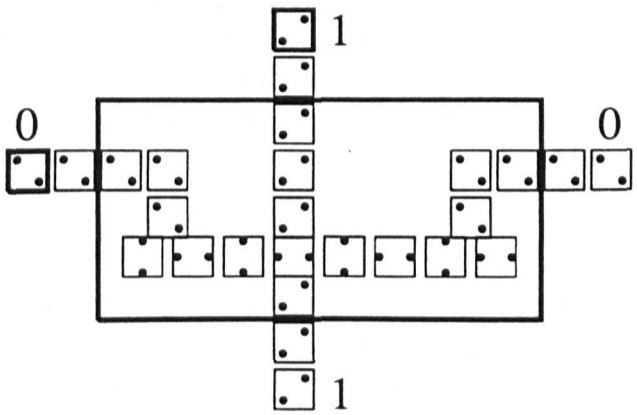

Figure 6. One way to cross two QCA wires without signal interference. The box shows the extent of the crossing, so a system designer can simply place such an arrangement of cells wherever two wires need to be crossed. This is the result of a self-consistent ground state calculation for this system.

Acknowledgment

This work was supported in part by AFOSR, ARPA, ONR, and a National Science Foundation Graduate Fellowship.

References

[1] C. S. Lent, P. Douglas Tougaw, and Wolfgang Porod, *Appl. Phys. Lett.*, **62**, 714 (1993).

[2] C. S. Lent, P. Douglas Tougaw, Wolfgang Porod and Gary H. Bernstein, *Nanotechnology* **4**, 49 (1993).

[3] P. Douglas Tougaw and C.S. Lent, to appear in Journal of Applied Physics, **74** (1993).

[4] C. S. Lent, P. Douglas Tougaw, and Wolfgang Porod, submitted to Journal of Applied Physics (unpublished).