Computer-Aided Design of Single-Electron Boltzmann Machine Neuron Circuit

Masamichi Akazawa, Takashi Yamada, and Yoshihito Amemiya

Faculty of Engineering, Hokkaido University

Kita 13, Nishi 8, Sapporo 060, Japan

Abstract–We present a computer-aided-design method for constructing a circuit for a Boltzmann-machine neuron, utilizing single-electron tunneling (SET). We have found, through computer simulation, that a stochastic response unit circuit can be made in a simple configuration using SET junctions, and the probability for an output of 1 can be controlled by the input voltages.

I. INTRODUCTION

Computer simulation is an essential and useful tool for developing functional devices for next-generation nanoelectronics.[1]

This paper presents one instance of this – a computer-aided design of single-electron Boltzmann machine neuron circuits. The Boltzmann machine is a neural network that uses stochastic or probabilistic operation of neurons. Ordinarily, it is difficult to implement using conventional electronic devices. But we have found, through computer simulation, that the stochastic neuron operation can be easily achieved by means of single-electron circuits.

II. STABILITY DIAGRAM OF THE SINGLE-ELECTRON-TUNNELING CIRCUIT

The single-electron-tunneling (SET) circuit [2][3] is an electronic circuit that consists of tunnel junctions and capacitors. A SET circuit has a number of nodes that are interconnected by means of tunnel junctions. Its internal state is determined by the configuration of electrons (i.e., the pattern in which the excess electrons are distributed among the nodes). The circuit varies its electron configuration through tunneling in response to the input, and thereby changes its output voltage as a function of the input.

A SET circuit changes its state to decrease its free energy; hence the circuit operates as an organic whole. Therefore any SET circuit has to be designed taking into consideration the global stability of the whole circuit. Because a SET circuit has complex internal states, a "guide map" is needed to grasp the overall situation of the circuit. The guide map or tool for this purpose is known as the *stability diagram*, the concept of which was first introduced by Likharev [3]. It is a diagram that illustrates the internal states of a SET circuit in a multidimensional space of *circuit variables* (namely, the voltages of powers and inputs, and the capacitances of tunnel junctions and capacitors). Looking at a stability diagram, we can see the internal state of the circuit as a function of the circuit variables. Given the stability diagram of a SET circuit, we can determine optimum values of circuit parameters so that the circuit will produce required functions.

The stability diagram can be calculated analytically for a simple SET circuit composed of a few junctions. But a circuit of greater complexity is difficult to calculate on paper, so computer simulation is needed. We have therefore developed a simulator to draw a stability diagram for a given circuit.

III. STABILITY-DIAGRAM SIMULATOR

The simulator that we developed calculates the stability diagram for a given circuit as follows. We first assume a current state for the circuit and then a set of values of circuit variables. After that, we calculate the energy change of the circuit for each possible tunneling. If all the energy change is incremental, we can consider the current state to be stable under this set of circuit variables. If the energy is reduced for one or more tunnelings, then we can consider the current state to be unstable under this set of circuit variables. We call this procedure a *trial* here. After ascertaining the stability of the state for a trial, we change each circuit variable slightly, and then repeat the trial calculation for a new set of circuit variables. By scanning the entire space defined by the variables, we can draw the stability diagram for the assumed state of the circuit. The same sequence is then repeated for the other states. Through these iterations, the complete stability diagram can be developed.

IV. BOLTZMANN MACHINE

The Boltzmann machine is a kind of feedback neural network that can solve various problems in subjects such as combinatorial optimization, classification, and association. Figure 1(a) presents a schematic diagram of a Boltzmann machine neural network. This consists of a large network of neurons that are interconnected bidirectionally with signal connections having various connection strengths. Each neuron receives input signals from other neurons and sends output signals to other neurons. The neuron has two output states, either 1 or 0, and changes its state according to the inputs, following a stochastic transition rule; i.e., the output is a random 1-0 bit stream. All neurons operate in parallel and each adjusts its own state to those of all the others. After some processing time, all the neurons finally reach maximal consensus about their individual states, and the whole network then stabilizes in a global configuration. For details, see [4] and [5].

0-7803-3775-1/97/$10.00

The structure of mathematical problems such as combinatorial optimization can be mapped onto the structure of a Boltzmann machine by deciding the connection pattern and connection strengths of the neurons. In this way, finding the optimal solution to a problem can be reduced to finding the optimal configuration of the Boltzmann machine. The unique and important feature of the Boltzmann machine is its method of operation, which uses stochastic neuron-state transition and simulated annealing algorithms. This allows the Boltzmann machine to reach a configuration that is globally optimal (and thereby an optimal solution) without falling into configurations that are only locally optimal. (This is a problem with other neural network models.) Because of this, the stochastic output of the neuron is the most important feature of the Boltzmann machine.

The basic concept of the Boltzmann machine neuron is illustrated in Fig. 1(b). It has two constituents, a sum-of-product unit and a stochastic-response unit. The sum-of-product unit has a number of input connections and local memory that stores connection strengths w_i (positive or negative analog values). Also, it receives input signals x_i (1 or 0) (and bias input that controls the threshold of the neuron) from other neurons and produces the weighted sum of inputs s ($= \sum w_i \cdot x_i + w_0$). The stochastic-response unit is peculiar to the Boltzmann-machine neuron. It generates an output, 1 or 0, updating the output state every moment, following a given probability that depends on the input value of s. The probability function for a state 1 is usually chosen to be the sigmoid function, expressed as

$$f(s) = \frac{1}{1 + \exp(-s/T)} \text{, or } f(s) = \frac{1}{1 + \exp(s/T)} \text{, } \quad (1)$$

where T (temperature) is the control parameter that slowly decreases from a large value to zero during the simulated annealing process. (Here the "temperature" need not be thermal temperature; any factor that can change the dependence of $f(s)$ on s can be used.) The shape of the function is illustrated in Fig. 2 for $f(s) = 1/(1 + \exp(s/T))$, with the value of T as a parameter. Convergence of the network systems requires the capability of varying "temperature T" with continuity by means of a control signal. The probability function need not necessarily be this function; any monotonic nonlinear function can be used, provided that it becomes 1 (or 0) at large positive values of s and becomes 0 (or 1) at large negative values of s.

A Boltzmann-machine LSI for practical use must integrate thousands of neurons on a chip. The crucial problem in developing such LSIs is how to implement the generation of randomness for the stochastic operation. Every neuron has to have its own randomness generator because stochastic independence between the neurons is required. But presently available circuits for generating randomness, such as the

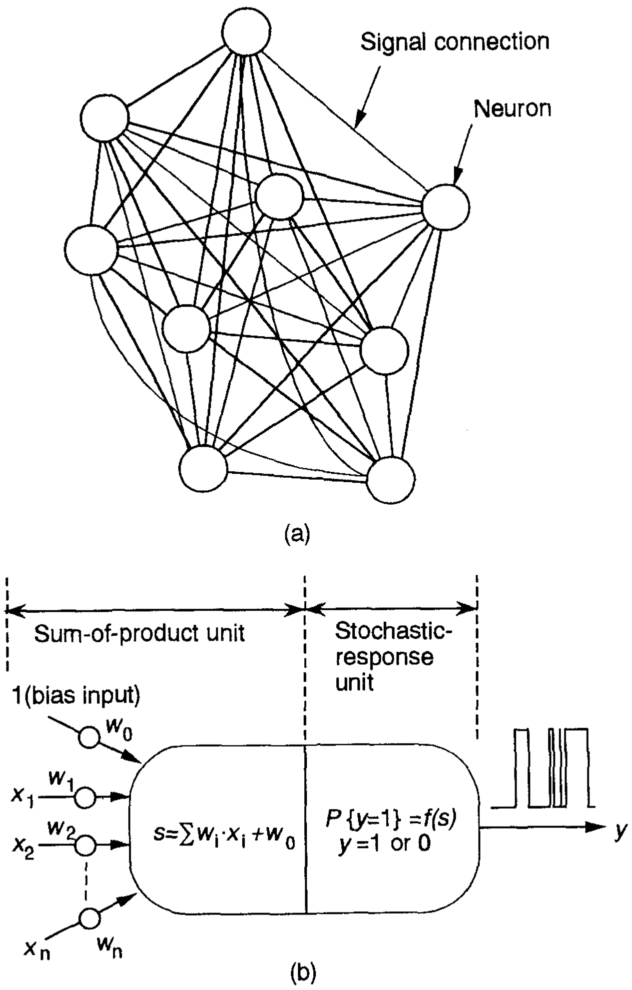

Fig. 1 Boltzmann-machine neural network and its neuron. (a) Concept of the network, (b) function of a Boltzmann-machine neuron.

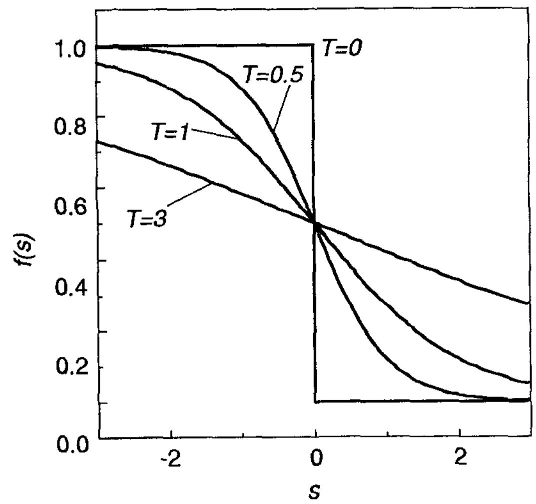

Fig. 2 Probability function $f(s)$ as a function of a weighted sum of inputs s. Illustrated is $f(s) = 1 / (1 + \exp(s/T))$, with T as a parameter.

thermal noise amplifier and the random bit generator, consist of many devices and consequently require a large volume of space; hence, they cannot be used for LSI implementation.

To overcome this problem, we have presented the idea that the inherent stochastic character of SET can be used for implementing the stochastic-response unit of the Boltzmann-machine neuron [6]. We will describe in the next section a single-electron neuron circuit that gives practical form to this idea. The point is to operate a SET circuit in unstable regions to produce stochastic output. A SET circuit in unstable regions varies its internal state between two more states, so an output of a random 1-0 bit stream can be expected. If the probability for an output 1 (or 0) can be changed in response to an input, then this phenomenon will be useful for the stochastic-response unit of the Boltzmann-machine neuron.

V. COMPUTER-AIDED DESIGN OF SINGLE-ELECTRON NEURON CIRCUIT

The stochastic-response unit has to be designed in a such configuration that the "temperature T" of the probability function can be changed by a control voltage. For this purpose, we modified the SET inverter circuit proposed by Tucker[7]. The circuit we propose for a stochastic response unit is illustrated in Fig. 3. The circuit has three island nodes (L, M, and N), and its internal state is expressed by the numbers of excess electrons (l, m, n) stored on the three nodes respectively. The circuit receives a voltage input s from a sum-of-product unit to generate its internal state and produces the corresponding voltage output y. The bias voltage V_b adjusts the threshold of the circuit by adding an offset to the input, and the value of the "temperature T" is changed by the control voltage V_{dd}.

For this circuit configuration, we designed the stability diagram for operating the circuit under unstable conditions around zero input. A desirable set of the capacitance parameters is:

$$C_{j1} = 1 \text{ aF}, C_{j2} = 2 \text{ aF}, C_1 = 3 \text{ aF}, C_2 = 9 \text{ aF}, C_{out} = 24 \text{ aF}. \quad (2)$$

Assuming this capacitance set, we drew the stability diagram in a three-dimensional space of three voltage variables (s, V_b, and V_{dd}). In Figs. 4 (a) and (b), a part of the diagram is illustrated on a plane of the two voltage variables; s and V_b. Four plain-colored regions, are stable regions, in which the circuit stabilizes at internal state (-1, -1, 0), (0, -1, 0), (0, 0, 0), and (0, 0, 1); the former two states produce high positive output voltage (an output '1'), while the latter two produce low output voltage (an output '0'). The approximate output voltage for each state is illustrated by putting a letter (H or L) before the electron-number set. The shaded region is an unstable region in which electron tunneling frequently occurs, and the circuit consequently alternates two or more internal states.

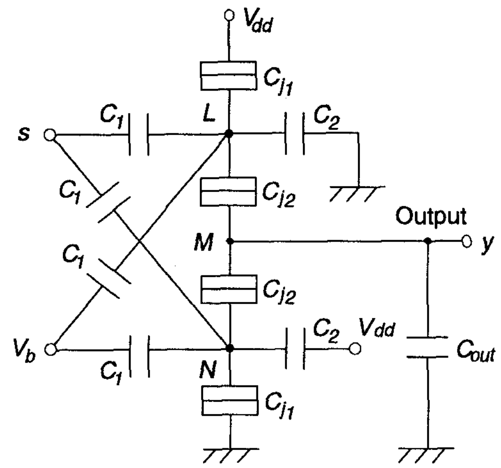

Fig. 3 Single-electron circuit for the stochastic response unit.

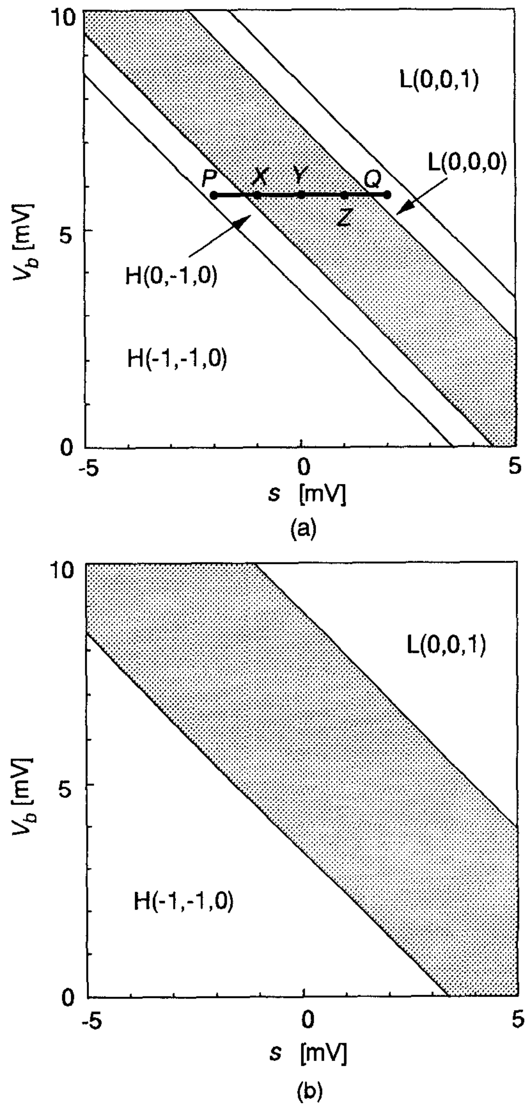

Fig. 4 A stability diagram of the circuit of Fig. 2. Capacitance parameters are: $Cj1$ = 1 aF, $Cj2$ = 2 aF, $C1$ = 3 aF, $C2$ = 9 aF, $Cout$ = 24 aF. The value of Vdd is (a) 6.10 mV and (b) 6.30mV.

We operated the circuit so that the operating point moved on the segment PQ illustrated in Fig. 4 (a). It can be expected that the probability for generation of an output 1 can be changed from 1 to 0 continuously by moving the operating point from P to Q. We simulated the circuit operation by using the Monte Carlo method[8] combined with the basic equations for electric-charge distribution, charging energy, and tunneling probability. The temperature is assumed to be 0 K. A simulation result is illustrated in Fig. 5 for the condition of Fig. 4 (a) (i.e., V_{dd} = 6.10 mV). Figure 5 shows the output voltage waveform (a random 1-0 bit stream) for two instance values of the input voltage: (a) s = -1.00 mV (point X in Fig. 4 (a)) and (b) s = 1.00 mV (point Y in Fig. 4 (a)). It can be seen that the probability for an output 1 can be changed by the input s, where the state of high output is dominant for a low value of s, while the state of low output is dominant for a high value of s. Intermediate states can also be generated, but this is not a problem because their duration is always short regardless of the input voltage value. In this example, the circuit changes its internal state in a cycle of L(0, 0, 0) → L(-1, 0, 0) → H(0, -1 ,0) → H(0, -1, 1) → L(0, 0, 0). Similar operation can be observed in other V_{dd} values.

The probability for an output 1 is illustrated in Fig. 6 as a function of the input voltage, with V_{dd} (and V_b) as a parameter. It is obtained by observing the output 1-0 stream for 1 μs and measuring the total duration of an output 1. It can be seen that a probability function required for the Boltzmann-machine neuron can be obtained very easily. It should be noted that the "temperature T" of the sigmoid characteristic can be controlled by changing the value of V_{dd}. This controllability of the "temperature T" is necessary for the network system operation.

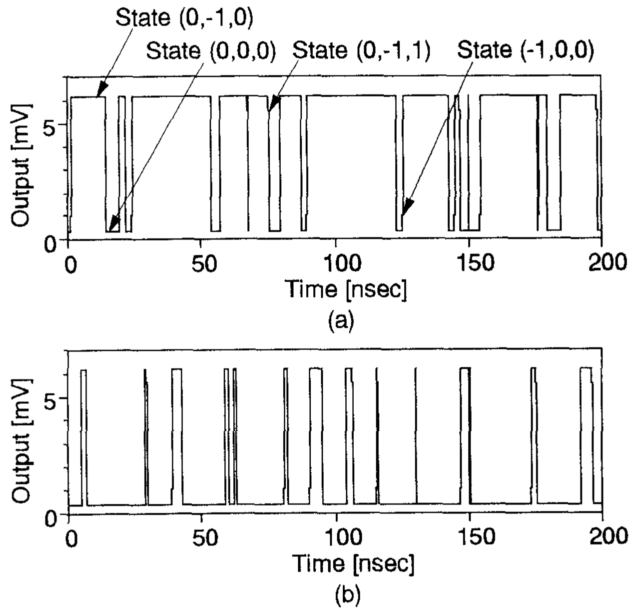

Fig. 5 Output voltage waveform simulated for the two input voltages shown in Fig. 4(a): (a) s = -1.00 mV (point X) and (b) s = 1.00 mV (point Y). Tunnel resistance is 5MΩ for three junctions.

VI. CONCLUSION

We presented a computer-aided-design method for constructing a circuit for a single-electron Boltzmann-machine neuron circuit. We developed a stability diagram simulator that plots the stable/unstable regions of a SET circuit on the circuit-variable coordinates. Utilizing the simulator, we designed the SET neuron circuit in a simple configuration. Using the single electron circuits, we will be able to fabricate very compact Boltzmann-machine LSIs.

REFERENCES

[1] T. Sugano, "Nano-electronics–A New Field for SISPAD," SISPAD'96 (Tokyo, September 2-4, 1996).

[2] H. Grabert and M. H. Devoret: *Single Charge Tunneling – Coulomb Blockade Phenomena in Nanostructures*, Plenum Press, New York, 1992.

[3] K. K. Likharev, "Single-Electron Transistors, Electrostatic Analogs on the DC Squids," *IEEE Trans. Magn.* vol. 23, pp. 1142-1145, 1987.

[4] G. E. Hinton and T. J. Sejnowski, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, edited by D. E. Rumelhart, J. L. McClelland, and the PDP Research Group (Bradford Books, Cambridge, 1986), Vol. 1, p. 282.

[5] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines* (John Wiley and Sons, Chichester, 1989), p. 117.

[6] M. Akazawa and Y. Amemiya, "Boltzmann machine neuron circuit using single-electron tunneling", J. Appl. Phys. Lett. vol. 70, no. 5, pp. 670-672, 1997.

[7] J. R. Tucker, "Complementary Digital Logic Based on the 'Coulomb Blockade'," J. Appl. Phys. vol. 72, pp. 4399-4413, 1992.

[8] N. Kuwamura, K. Taniguchi, and C. Hamaguchi, "Simulation of Single-Electron Logic Circuits," Trans. IEICE vol. J77-C-II, pp. 221-228, 1994.

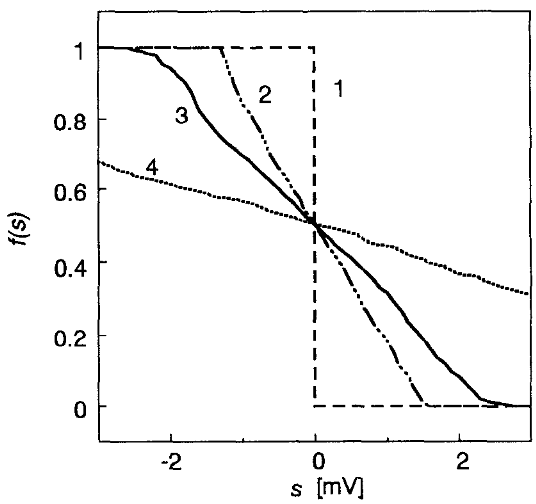

Fig. 6 The probability for generating an output 1 is illustrated as a function of the input voltage s. Curve 1 is for Vdd = 5.76mV (Vb = 5.50 mV), curve 2 for Vdd = 6.10 mV (Vb = 5.80 mV), curve 3 for Vdd = 6.30 mV (Vb = 6.10 mV), and curve 4 for Vdd = 7.50 mV (Vb = 8.50 mV).