PAPER Special Issue on Integrated Electronics and New System Paradigms

A Multiple-Valued Hopfield Network Device Using Single-Electron Circuits

Takashi YAMADA[†a)], ***Student Member and*** **Yoshihito AMEMIYA**[†], ***Member***

SUMMARY We developd a method of *implementing a multiple-valued Hopfield network on electronic circuits by using single-electron circuit technology.* The single-electron circuit shows quantized behavior in its operation because of the discrete tunnel transport of electrons. It can therefore be successfully used for implementing neuron operation of the multiple-valued Hopfield network. The authors developed a single-electron neuron circuit that can produce the staircase transfer function required for the multiple-valued neuron. A method for constructing a multiple-valued Hopfield network by combining the neuron circuits was also developed. A sample network was designed that solves an example of the quadratic integer-programming problem. And a computer simulation demonstrated that the sample network can converge to its optimal state that represents the correct solution to the problem.

key words: *single electron, Hopfield network, multiple valued*

1. Introduction

One of the challenges in microelectronics is the development of novel electronic devices that can perform neural computing by utilizing functional properties of quantum phenomena. We have developed one such computation device: *a multiple-valued Hopfield network device that utilizes the quantized transport properties in single-electron tunneling.*

The Hopfield network is a kind of recurrent neural network for solving combinatorial optimization problems (Refs. [1] and [2]). It consists of a large network of processing elements (neurons) interconnected bidirectionally with signal connections having various connection weights. Each neuron receives an input signal from every other neuron and sends an output signal to every other neuron. The neuron has a binary output state, 0 or 1, and changes its state in response to the inputs, according to a given transition rule. All neurons operate in parallel and each adjusts its own state to the states of all the others; in consequence, the whole network converges to a final configuration. The structure of combinatorial optimization problems can be mapped onto the structure of a Hopfield network by deciding the connection weights between the neurons. In this way, we can find the solution to a problem simply by observing the final configuration that the Hopfield network reaches. The unique and important feature of the Hopfield network is that a given problem is solved through concurrent operation of all neurons in the network; therefore, the Hopfield network can be used to produce computation devices that can solve optimization problems in a short time regardless of the size of the problem.

A promising subject of research on the Hopfield network is to develop a multiple-valued extension to ordinary Hopfield networks. Ordinary Hopfield networks employ binary-state neurons for processing, so the kind of optimization problem that can be accepted is limited to examples with binary variables (0 and 1). If we can develop an improved Hopfield network that manipulates multiple-valued variables (i.e., integer variables 0, 1, 2, 3, etc.), then we will be able to construct a novel computation device that can deal with optimization problems from a wide range of subjects. To meet this requirement, Aiyoshi and Yoshikawa developed the concept of the multiple-valued Hopfield network and showed that the neuron element for their network has to have a *multistep* or *staircase* transfer function instead of a simple threshold function used in ordinary Hopfield networks (Refs. [3] and [4]). They also demonstrated as an example that quadratic integer-programming problems can be solved by using the multiple-valued Hopfield network.

The crutial problem in developing actual devices or LSIs for the multiple-valued Hopfield network is how to implement the staircase neuron function on an electronic circuit. Presently available electronic circuits for generating a staircase function, such as a comparator array combined with a multitude of reference voltage sources, consist of many device elements and, consequently, require a large volume of space. They therefore cannot be used for LSI implementation. So we must develop a novel neuron device for constructing multiple-valued Hopfield network LSIs.

Presumably, this requirement can be met by using single-electron circuits, a quantum electronic circuit based on the Coulomb blockade effect in electron tunneling. The single-electron circuit changes its internal state through a discrete event of electron tunneling in response to the inputs and, thereby, changes its output voltage as a discrete function of the inputs. The authors therefore expect that the single-electron circuit will concisely produce the staircase neuron function re-

Manuscript received January 20, 1999.
Manuscript revised April 5, 1999.

[†]The authors are with the Department of Electrical Engineering, Hokkaido University, Sapporo-shi, 060-8628 Japan.
a) E-mail: yamada@sapiens-ei.eng.hokudai.ac.jp

quired for the multiple-valued Hopfield network. To give a practical form to this idea, we propose in this paper an actual circuit construction for single-electron multistep neuron devices.

In the following sections, first, the concept of the multiple-valued Hopfield network is outlined. The multiple-valued Hopfield network can solve optimization problems that manipulate multiple-valued variables. On the other hand, it requires a special neuron element with a staircase transfer function (Sect. 2). Next, we present an idea for implementing a staircase neuron function by using the single-electron circuit. The single-electron circuit has inherently multiple-valued properties in its operation owing to the discrete tunnel transport of electrons. We develop a circuit structure for neuron elements and demonstrate by computer simulation that the developed neuron circuit can successfully produce a multistep output voltage in response to the input (Sect. 3). By combining a number of the neuron circuits, we construct the multiple-valued Hopfield network; as an example, a sample network, which represents a problem instance of the quadratic integer programming, is designed. The problem-solving behavior of the network is then demonstrated by computer simulation. It is shown that the network does converge successfully to its optimal configuration that corresponds to the correct solution (Sect. 4). The authors hope that this paper will stimulate the thinking of readers who are aiming to create neural computing devices that utilize quantum phenomena.

2. The Concept of the Hopfield Network

2.1 An Ordinary Hopfield Network

The Hopfield network is a computation device for solving combinatorial optimization problems and employs the operation of a specific recurrent network. (For detailed explanations, see Refs. [1] and [2].) The configuration of the network is illustrated in Fig. 1. The network consists of neurons (denoted by triangles) and connections (denoted by solid circles). The output of each neuron feeds back into inputs of other neurons. Denoted by Wij is the connection weight to neuron i from neuron j, θi is the bias-connection weight to neuron i from a bias that is fixed at the value of 1, and xi is the output of neuron i. The connection weights Wij and θi can be given any desired value under the restrictions that $Wij = Wji$ and $Wii = 0$. The neuron output xi is binary (i.e., $xi = 1$ or 0) in ordinary Hopfield networks. A set of neuron outputs xi is called the state of the network.

In this network, each neuron i takes a weighted sum of its inputs according to the following equation:

$$si = \sum_j Wij\, xj + \theta i. \tag{1}$$

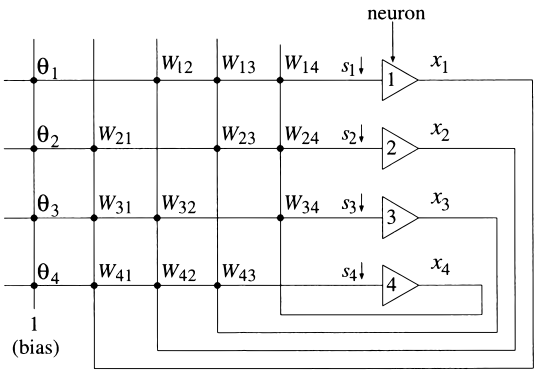


Fig. 1 Concept of the Hopfield network. Denoted by Wij is the weight of connection between neuron i and neuron j ($Wij = Wji$), θi is the weight of connection between neuron i and a bias of 1, si is the input for neuron i given by Eq. (1) in the text, and xi is the output of neuron i.

And each neuron generates the corresponding output xi (= 1 or 0), following a given transfer function (a function for determining output xi from the weighted sum si). In the network, all neurons operate in parallel to update their outputs continuously, and the whole network converges to an optimal configuration through the updating process. By choosing the connection weights and using an appropriate transfer function for neurons, we can map a given optimization problem onto the network structure. In this way, finding the solution to the problem can be reduced to finding the optimal configuration of the network. In problem solving, we set the network in an initial state (any state will do), then allow it to change its state without restraint. After some transition time the network converges to a final state. If convergence has been successful, the solution to the problem is obtained from the final state of the network (see Appendix A).

As an example, we here take up the unconstrained quadratic-programming problem that is stated as:

Given a set of coefficients Aij and Bi ($Aij = Aji$, $Aii \neq 0$; $i, j = 1, 2, \ldots, n$), minimize the objective function

$$\frac{1}{2}\sum_i^n \sum_j^n Aij\ xi\ xj + \sum_i^n Bi\ xi. \tag{2}$$

To solve this problem, we prepare a Hopfield network such that the output of each neuron represents each variable xi. For an ordinary Hopfield network with binary-output neurons, an acceptable problem is limited to examples in which the variables are binary (i.e., $xi \in \{0, 1\}$). Under such 0-1 restraint, a square term xi^2 can be replaced with xi and, consequently, Eq. (2) can be rewritten as

$$\frac{1}{2}\sum_i^n \sum_{j\neq i}^n Aij\ xi\ xj + \sum_i^n (Bi + \frac{1}{2}Aii)\ xi. \tag{3}$$

It has been proved that this 0-1 programming can

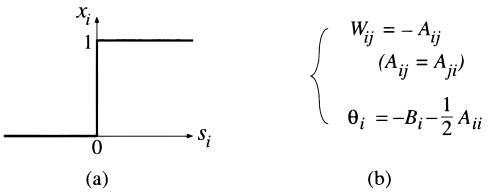


Fig. 2 Elements for an ordinary (binary) Hopfield network: (a) the transfer function for neurons, and (b) the connection weights for mapping the quadratic 0-1 programming given by Eq. (2) in the text.

be mapped onto the Hopfield network, by using the single-threshold transfer function for neurons illustrated in Fig. 2(a), with the connection weights shown in Fig. 2(b). The value of solution xi that minimizes the objective function of Eq. (3) can be read from the final state of the network; that is, the value of solution xi is equal to the output of neuron i.

2.2 The Multiple-Valued Hopfield Network

In the problem expressed by Eq. (2), we here relax the 0-1 restraint on variables and assume that each variable xi can take an integer from 0 to N. The problem is therefore expressed as:

minimize

$$\frac{1}{2}\sum_{i}^{n}\sum_{j}^{n} Aij\ xi\ xj + \sum_{i}^{n} Bi\ xi,$$
$$\text{where}\quad xi \in \{0, 1, 2, ..., N\}\ (i = 1, 2, ..., n). \qquad (4)$$

This extended problem (the quadratic interger-programming problem) is beyond the range of ordinary Hopfield networks. For problem solving, a more sophisticated architecture is required. (Strictly speaking, we can manage to solve this problem with an ordinary Hopfield network if we use a large number of neurons. But this is impracticable in LSI implementation.) To meet this requirement, Aiyoshi and Yoshikawa developed an improved Hopfield network that can manipulate multiple-valued variables (see Refs. [3] and [4]). This network is called the multiple-valued Hopfield network.

The multiple-valued Hopfield network has the same configuration as that of its ordinary relatives (see Fig. 1) and is under the same restrictions that $Wij = Wji$ and $Wii = 0$. The difference between them is in the operation of neurons. A neuron of the multiple-valued Hopfield network produces a quantized output according to a staircase transfer function with a multiple threshold. The tranfer function required for neuron i is illustrated in Fig. 3(a). Using this multiple-valued neuron, the quadratic integer programming described by Eq. (4) can be mapped onto a network that has the connection weights given in Fig. 3(b). As in the

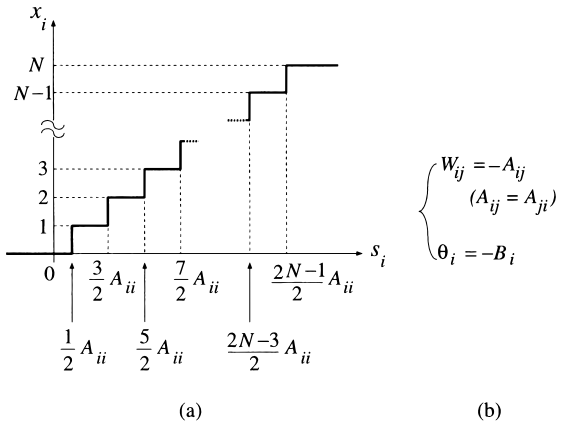


Fig. 3 Elements for the multiple-valued Hopfield network: (a) the staircase transfer function for neuron i, and (b) the connection weights for mapping the quadratic integer programming given by Eq. (4) in the text.

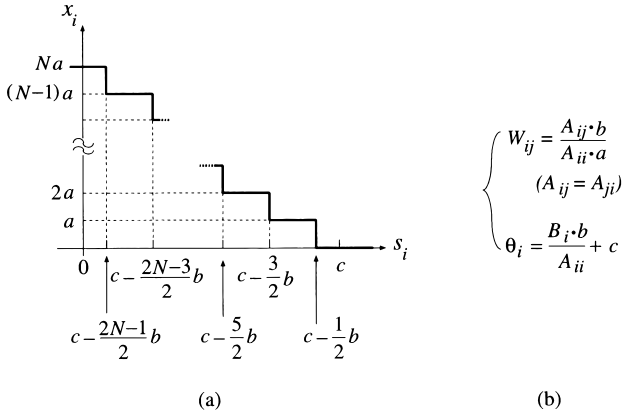


Fig. 4 Modified elements for the multiple-valued Hopfield network: (a) the modified staircase transfer function for neuron i, and (b) the connection weights for mapping the quadratic integer programming given by Eq. (4) in the text.

ordinary Hopfield network, the solution to the problem is obtained from the final state of the network; that is, the value of solution xi is equal to the output of neuron i. (Strictly speaking, the shape of the transfer function for neuron i depends on a sign of coefficient Aii. The function illustrated in Fig. 3(a) is for $Aii > 0$. For $Aii < 0$ and $Aii = 0$, see Refs. [3] and [4]).

Next, we assume that a given neuron has a transfer function like that illustrated in Fig. 4(a) instead of that in Fig. 3(a) (a, b, and c in Fig. 4(a) are the *characteristic coefficients* that determine the shape of the transfer function). This is the characteristic of the neuron device that will be developed in the next section. The problem described by Eq. (4) can also be mapped successfully onto the network provided the modified connection weights given in Fig. 4(b) are used.

3. Constructing the Multiple-Valued Neuron Device Using Single-Electron Circuits

3.1 The Concept of the Single-Electron Circuit

The single-electron circuit is an electronic circuit consisting of tunnel junctions and capacitors that is designed for manipulating electronic functions by controlling the transport of individual electrons. (For a detailed explanation, see Ref. [5].) A single-electron circuit has a number of nodes that are interconnected by the tunnel junctions. Electrons in each node can tunnel to another node through the tunnel junction. The internal state of the circuit is determined by the configuration of its electrons (i.e., the pattern in which the electrons are distributed among the nodes). The circuit changes its electron configuration through electron tunnelings in response to the inputs and, thereby, changes its output voltage as a function of the inputs. A change of the electron configuration caused by a tunneling event can occur, at low temperatures, only when the free energy of the circuit decreases in the tunneling event. This phenomenon is called the *Coulomb blockade*. Utilizing this phenomenon, the single-electron circuit controls the transport of individual electrons to produce various functions such as digital logic and memory operations.

The single-electron circuit has been receiving increasing attention because it can be used to produce LSIs that combine large integration and ultralow power dissipation. The key point for constructing single-electron circuit devices is to fabricate the circuit elements (tunnel junctions and capacitors) in very minute dimensions (50 nm or less) because the Coulomb blockade effect emerges only when the capacitance of each element is reduced to femto farads or less. The technology for such nanofabrication is still immature but has been making steady progress. Several elemental devices, such as logic gates and memory cells, have been produced in recent years, and a prototype of single-electron LSI can be expected in the near future.

3.2 Multiple-Valued Characteristic of Single-Electron Circuits

A conspicuous property of the single-electron circuit is that the circuit shows "quantized behavior" in its operation. This quantization is caused by the fact that the charge on each node of the single-electron circuit is changed only through electron tunnelings. Because one electron is transferred through a tunneling event, the variation of the node charge is necessarily quantized in units of the elementary charge; therefore, the node charge is a discrete variable, so the node potential is also discrete. In consequence, the output voltage of a single-electron circuit changes as a discontinuous function of the input voltage. These properties can be well utilized for the present purpose; that is, creating a multiple-valued neuron device in a compact construction.

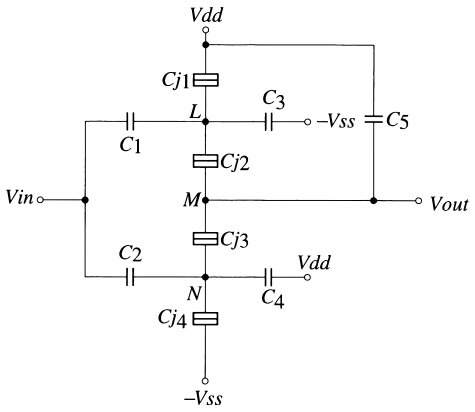


Fig. 5 Configuration of the neuron circuit for the multiple-valued Hopfield network.

3.3 Creating the Multiple-Valued Neuron Device Using a Single-Electron Circuit

Our purpose is to construct a *single-electron circuit that changes the charge on its output node (therefore its output voltage) as a staircase function of the input voltage.* For this purpose, we take *Tucker's single-electron inverter* and modify its circuit parameters to create a staircase transfer function (for the details of Tucker's original circuit, see Ref. [6]).

The circuit we use for the neuron device is illustrated in Fig. 5. It consists of four tunnel junctions ($Cj1$ through $Cj4$), two input capacitors ($C1$ and $C2$), two bias capacitors ($C3$ and $C4$), a load capacitor ($C5$), and two voltage sources Vdd and $-Vss$. The circuit accepts a voltage input Vin and produces the corresponding voltage output $Vout$. The circuit has three island nodes (L, M, and N), and its internal state is expressed by a set of numbers (l, m, n) of excess electrons on the three nodes.

Depending on capacitance parameters, this circuit shows complex internal states and therefore shows various input-output characteristics. To create the multiple-valued neuron device, we set the circuit parameters so that the circuit will operate with a staircase transfer function. In determining the optimum parameters, we used the *stability diagram* of the circuit as a guide map. (The stability diagram illustrates the internal states of a single-electron circuit in a multidimensional space of circuit variables—namely, the voltages of powers and inputs and the capacitances of tunnel junctions and capacitors. For details of the stability diagram, see Ref. [7].) The optimum set of parameters depends on what kind of staircase function we require (i.e., what values are needed for the characteristic co-

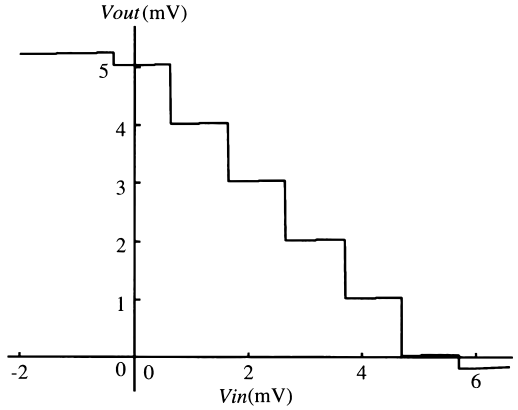


Fig. 6 The tranfer function (input-output curve) of the multiple-valued neuron circuit of Fig. 5. Simulated using the Monte Carlo method given in Ref. [8]. The temperature is 0 K.

efficients a, b and c in Fig. 4(a)). An example set of parameters is:

$$Cj1 = Cj4 = 3 \text{ aF},\ Cj2 = Cj3 = 5 \text{ aF},$$

$$C1 = C2 = 5 \text{ aF},\ C3 = C4 = 10 \text{ aF},\ C5 = 150 \text{ aF},$$

$$Vdd = 5.20 \text{ mV},\ -Vss = 0 \text{ mV}. \quad (5)$$

The corresponding transfer function is illustrated in Fig. 6. The output voltage discontinuously jumps by a constant decrement as the input voltage increases and, in consequence, a staircase transfer function is obtained. (The authors simulated the operation of single-electron circuits by using the Monte Carlo method combined with the basic equations for electric-charge distribution, charging energy, and tunneling probability. For details on this method, see Ref. [8]).

3.4 The Stability Diagram of the Neuron Circuit

We explain the operation of the neuron circuit by using the stability diagram. The stability diagram for this circuit configuration with given capacitance parameters is plotted in Fig. 7 on a plane of two voltage variables, input voltage Vin and power voltage Vdd (the value of $-Vss$ is fixed at 0 mV). Plain-colored regions marked at (l, m, n) are stable regions in which the circuit stabilizes in an internal state (l, m, n). The shaded region is an unstable zone in which electron tunneling frequently occurs and the circuit consequently alternates between two or more internal states. The stable regions take on various configurations and most regions overlap with one another.

We operate the circuit on an operating line (corresponding to $Vdd = 5.20$ mV (see Eq. (5)) illustrated by PQ in Fig. 7. There is no overlap of the stable regions on this operating line, so the internal state of the circuit can definitely be determined by input voltage Vin; e.g., in the figure, the circuit takes internal state (0, 0, 0) at point P and (0, 3, 0) at point R. As input voltage Vin increases, the internal state changes in the order: $(0,0,0), (0,1,0), (0,2,0), (0,3,0), (0,4,0), (0,5,0)$. As the input voltage Vin increases, electrons are added one-by-one to the output node. The output voltage therefore varies as a staircase function of the input, as illustrated in Fig. 6.

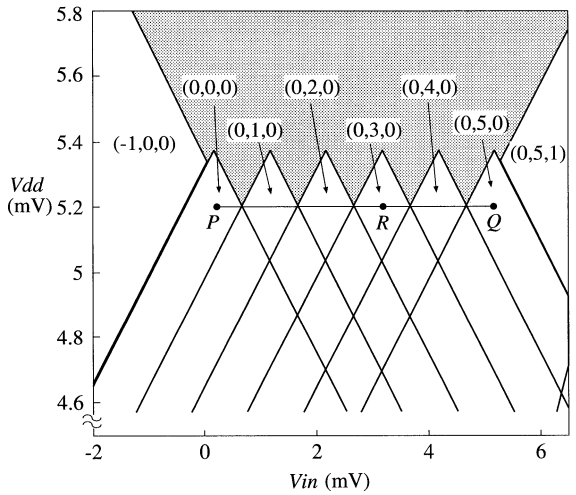


Fig. 7 Stability diagram for the multiple-valued neuron circuit. For the capacitance parameters, see the text. The value of $-Vss$ is set at 0 mV. The circuit is operated on line PQ.

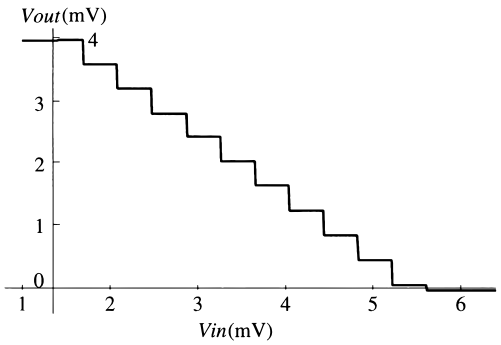


Fig. 8 Another instance of the staircase transfer function. The set of paramerers is: $Cj1 = Cj4 = 3$ aF, $Cj2 = Cj3 = 5$ aF, $C1 = C2 = 5$ aF, $C3 = C4 = 10$ aF, $C5 = 400$ aF, $Vdd = 5.20$ mV, $-Vss = 0$ mV. The temperature is 0 K.

By designing the circuit parameters, we can control the number of steps in the transfer function. Figure 8 illustrates the transfer function of neuron circuit with a different set of parameters.

4. Operation of a Multiple-Valued Hopfield Network

4.1 Constructing a Multiple-Valued Hopfield Network

The multiple-valued Hopfield network can be constructed by combining the developed neuron circuits into a network. We illustrate here a sample net-

x_3= 0	x_1					
	0	1	2	3	4	5
0	0	–17	–28	–33	–32	–25
1	–25	–38	–45	–46	–41	–30
x_2 2	–38	–47	–50	–47	–38	–23
3	–39	–44	–43	–36	–23	–4
4	–28	–29	–24	–13	4	27
5	–5	–2	7	22	43	70

x_3= 3	x_1					
	0	1	2	3	4	5
0	57	43	35	33	37	47
1	26	16	12	14	22	36
x_2 2	7	1	1	7	19	37
3	0	–2	2	12	28	50
4	5	7	15	29	49	75
5	22	28	40	58	82	112

x_3= 1	x_1					
	0	1	2	3	4	5
0	7	–9	–19	–23	–21	–13
1	–20	–32	–38	–38	–32	–20
x_2 2	–35	–43	–45	–41	–31	–15
3	–38	–42	–40	–32	–18	2
4	–29	–29	–23	–11	4	31
5	–8	–4	6	22	44	72

x_3= 4	x_1					
	0	1	2	3	4	5
0	100	87	80	79	84	95
1	67	58	55	58	67	82
x_2 2	46	41	42	49	62	81
3	37	36	41	52	69	92
4	40	43	52	67	88	115
5	55	62	75	94	119	150

x_3= 2	x_1					
	0	1	2	3	4	5
0	26	11	2	–1	2	11
1	–3	–14	–19	–18	–11	2
x_2 2	–20	–27	–28	–23	–12	5
3	–25	–28	–25	–16	–1	20
4	–18	–17	–10	3	22	47
5	1	6	17	34	57	86

x_3= 5	x_1					
	0	1	2	3	4	5
0	155	143	137	137	143	155
1	120	112	110	114	124	140
x_2 2	97	93	95	103	117	137
3	86	86	92	104	122	146
4	87	91	101	117	139	167
5	100	108	122	142	168	200

Fig. 9 The value of the problem function given by Eq. (6) in the text, tabled for all possible combinations of variables x_1, x_2, and x_3. The function takes the minimum, -50, for the variable set $x_1 = 2$, $x_2 = 2$, and $x_3 = 0$.

work that solves an instance of the quadratic integer-programming problems. In the following, we determine, first, a set of connection weights for a problem instance. Then, using the determined set of connection weights, we demonstrate by computer simulation the problem-solving operation of the network.

Consider the following quadratic integer-programming problem:

minimize

$$3x_1^2 + 6x_2^2 + 6x_3^2 + 4x_1x_2 - 2x_2x_3 + x_3x_1 - 20x_1 - 31x_2 + x_3, \qquad (6)$$

where $xi \in \{0, 1, 2, 3, 4, 5\}$.

In advance of problem solving with the Hopfield network, we calculated the value of this function for all possible combinations of variables xi. The results are tabled in Fig. 9. The function takes the minimum, -50, for the variable set $x_1 = 2$, $x_2 = 2$, and $x_3 = 0$.

To solve this problem instance, we prepared a network with three neurons to represent problem variables xi by the output of i-th neurons ($i = 1, 2, 3$). We used the neuron circuit whose characteristic is illustrated in Fig. 6; (characteristic coefficients are $a = 1.02\,\mathrm{mV}$, $b = 1.02\,\mathrm{mV}$, and $c = 5.23\,\mathrm{mV}$). The required connection weights between the neurons can be determined by using the relation given in Fig. 4(b). The overall configuration of the network, together with the determined connection weights is given in Fig. 10.

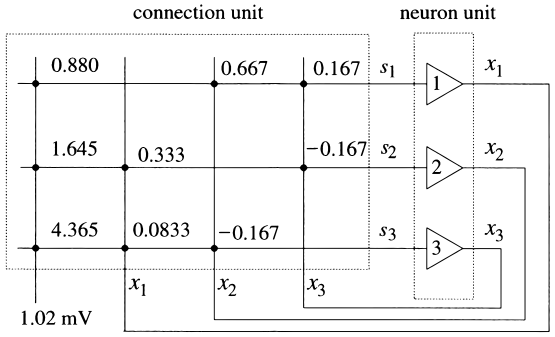


Fig. 10 The configuration of the multiple-valued Hopfield network together with the values of the connection weights. This circuit is for the quadratic integer-programming problem given by Eq. (6) in the text.

4.2 Problem Solving Operation of the Network

For problem solving, it is essential that, starting with a given initial state, the network circuit should converge to its minimum energy states. To observe the behavior of the sample network, we simulated the state transition of the network.

The network consists of a neuron unit and a connection unit (see Fig. 10). Each unit was simulated by using the following method.

(i) *Neuron unit*: We used the Monte Carlo method for simulating the neuron unit because it is essential to use the Monte Carlo method for simulating the discrete tunnel events in single-electron circuits. In simulation, we used the same parameter set as that given in Sect. 3.3. The tunnel resistance was assumed to be 3 MΩ for all junctions, and the temperature was assumed to be 0 K.

(ii) *Connection unit*: We considered the connection unit as a blackbox that produced output si in response to input xi without time delay, and we calculated its input-output characteristic simply by using Eq. (1). The reason we used the simple calculation instead of the Monte Carlo method is as follows. The connection unit has to generate both of positive analog weights and negative ones, as shown in Fig. 10. A *linear* amplifier is therefore required for actual implementation. Such amplifiers cannot be constructed by single-electron circuits, so we will have to use CMOS circuits for implementing the connection unit. (This means that an LSI chip for the multiple-valued Hopfield network will have a hybrid structure of single-electron neuron circuits and CMOS connection-weight circuits: see Appendix B) For simulating CMOS circuits, we do not need to use the Monte Carlo method; we can grasp the circuit operation through the simple calculation. (It will take enormous computing time if we use the Monte Carlo method to simulate CMOS circuits because the number of electrons concerned is too large.)

The results of the simulation are illustrated in

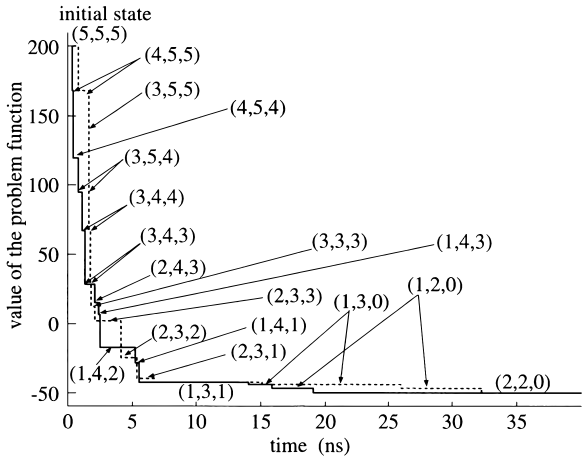


Fig. 11 State transition in the sample network. The results of two trials are plotted (solid curve and dashed curve). The network can successfully reach the minimum state that corresponds to the correct solution to the problem.

Fig. 11. In the figure, the state of the sample network is expressed by a normalized set of three neuron outputs $(u1, u2, u3)$, where ui is the output of i-th neuron normalized by the characteristic coefficient a (i.e., $ui = x\ i/a$). The circuit was initially set at state (5, 5, 5), then it was allowed to change its state without restraint. After some transition time, the circuit stabilized in the final state (2, 2, 0) that corresponds to the correct solution to the problem. This procedure, a trial, was repeated many times using a different series of random numbers; the results of two trials are illustrated in the figure. It can be seen that the circuit successfully reaches the global minimum state, (2, 2, 0). We repeated the same trial many times and confirmed that every trial resulted in successful convergence. In this way, we can find the minimum state of the network and, thus, the correct solution to the problem.

5. Conclusion

We developed a method of implementing the multiple-valued Hopfield network on electronic circuits by using the single-electron circuit technology. The single-electron circuit shows quantized behavior in its operation. This behavior results from the discrete tunnel transport of electrons, so the operation of the multiple-valued Hopfield network can be implemented easily by using single-electron circuits. We developed a single-electron neuron circuit that can produce the staircase transfer function required for the multiple-valued neuron; the neuron circuit produces a multistep output voltage in response to the weighted sum of its inputs. By combining the developed neuron circuits, we designed a sample network that implemented a problem instance of quadratic integer programming. Computer simulation showed that the sample network can converge to its minimum energy state, which represents the correct solution to the problem. Our results show that multiple-valued Hopfield network LSIs can be fabricated by using single-electron circuits.

(This work was supported by a Grant-in-Aid for Scientific Research on Priority Areas "Single-electron devices" from the Ministry of Education, Science, Sports, and Culture and by CREST of JST (Japan Science and Technology).)

References

[1] J.J. Hopfield and D.W. Tank, "Neural computation of decisions in optimization problems," Biological Cybernetics, vol.52, pp.141–152, 1985.
[2] D.W. Tank and J.J. Hopfield, "Simple neural optimization networks: An A/D converter, signal decision circuit, and linear programming circuit," IEEE Trans. Circuits & Systems, vol.CAS-33, no.5, pp.533–541, 1986.
[3] E. Aiyoshi and A. Yoshikawa, "New perspective on optimization—Combinatorial optimization by neural networks," Trans. IEE vol.112C, no.9, pp.533–540, 1992.
[4] A. Yoshikawa, "Integer programming using multiple-valued neural networks," Papers of Technical Meeting on Information Processing, IP-95-4, IEE Japan, 1995.
[5] H. Grabert and M.H. Devoret, Single Charge Tunneling—Coulomb Blockade Phenomena in Nanostructures, Plenum Pless, 1992.
[6] J.R. Tucker, "Complementary digital logic based on the Coulomb blockade," J. Appl. Phys., vol.72, no.9, pp.4399–4413, 1992.
[7] M. Akazawa and Y. Amemiya, "Eliciting the potential functions of single-electron circuits," IEICE Trans. Electron., vol.E80-C, no.7, pp.849–858, 1997.
[8] N. Kuwamura, K. Taniguchi, and C. Hamakawa, "Simulation of single-electron logic circuits," IEICE Trans. Electron., vol.J77-C-II, no.5, pp.221–228, 1994.

Appendix A

Strictly speaking, it is not possible to be certain, in the Hopfield network, that convergence to the global minimum can always be obtained. This is because many local minima can exist in the function of the network state. To avoid the network becoming stuck in the local minima, several sophisticated techniques are employed in operating the network, but we will not discuss them here.

Appendix B

In constructing the network consisting of single-electron neuron circuits and CMOS connection-weight circuits, it is essential to design the CMOS circuits so that their input capacitances will be sufficiently small. This is because the single-electron neuron circuit cannot drive a load of large capacitance. The neuron circuit given in Fig. 5 in the text, for example, produces the desired staircase function (Fig. 6) only when its load capacitance (denoted by $C5$ in Fig. 5) is set at 150 aF; therefore the neuron circuit cannot drive a CMOS circuit

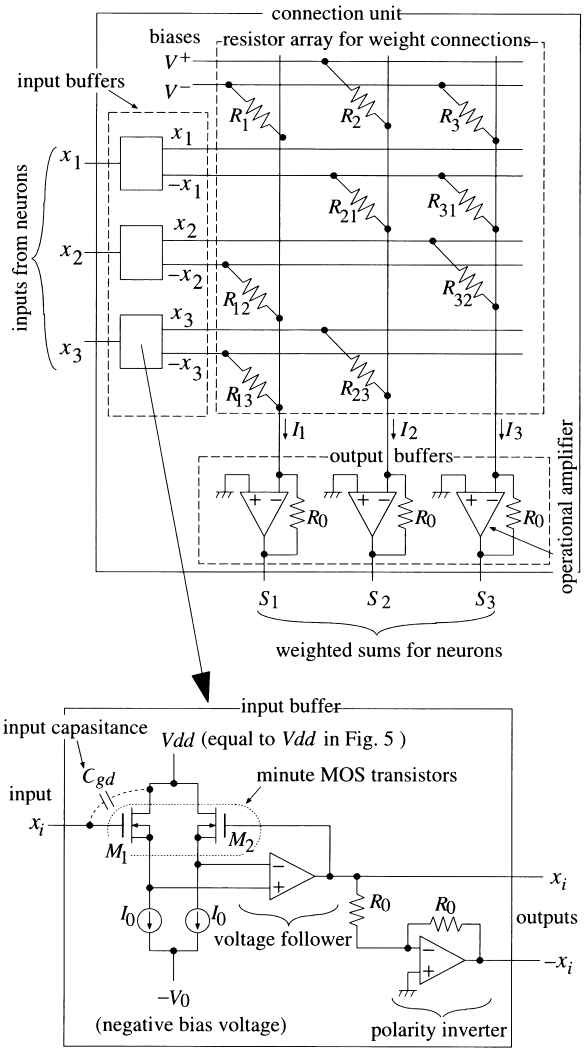


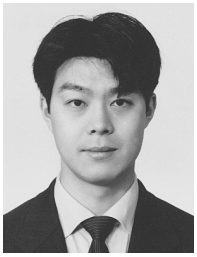
Fig. A· 1 A sample structure for the CMOS connection unit.

whose input capacitance is larger than 150 aF. It is therefore indispensable for LSI implementation of the network to develop CMOS connection-weight circuits with small input capacitances.

We here propose, in Fig. A· 1, a candidate structure for the CMOS connection unit. It consists of input buffers, a resistor array for weight connections, and output buffers. The input buffers receive voltage inputs x_i from single-electron neuron circuits and produce inverted outputs $-x_i$ and noninverted outputs x_i. From these outputs, the resistor array produces weighted-sum current signals I_i, using the technique of Kirchhoff current summation; the inverted and the noninverted outputs are used for positive- and negative-weight connections respectively. The output buffers convert current signals I_i into weighted-sum voltage outputs s_i, which are fed back to the neuron circuits.

The input buffer accepts input signal x_i with a source follower consisting of n-channel MOS transistor $M1$ and current source I_0. Then it produces output x_i by using a voltage follower combined with level-shifting transistor $M2$ biased by I_0 (identical transistors are used for $M1$ and $M2$). Inverted output $-x_i$ is produced by the use of a polarity inverter.

The input capacitance of this connection unit is equal to gate-drain capacitance Cgd of $M1$. It has been predicted that the gate-drain capacitance of a MOS transistor can be reduced to tens of attofarads by reducing the device dimensions to 0.1 μm or less. By using such a minute MOS transistor as $M1$ (and $M2$), we will be able to construct the connection unit with an ultrasmall input capacitance.



Takashi Yamada was born in Hokkaido, Japan, on August 18, 1973. He received the B.E. and M.E. degrees in Electrical Engineering from Hokkaido University in 1997 and 1999, respectively. He is currently working toward the Dr. Eng. degree in the Department of Electrical Engineering, Hokkaido University. His current research interests are in single-electron devices, neural network circuits, and intelligent LSIs.

Yoshihito Amemiya was born in Tokyo, Japan, on March 5, 1948. He received the B.E., M.E., and Dr. Eng. degrees from the Tokyo Institute of Technology in 1970, 1972, and 1975, respectively. From 1975 to 1993, he was a Member of the Research Staff at NTT LSI Laboratories, Atsugi, Japan. Since 1993 he has been a professor in the Department of Electrical Engineering at Hokkaido University. His research is in the field of semiconductor devices, LSI circuits, and digital- and analog-processing elements utilizing quantum phenomena and single-electron effects.