A Majority-Logic Nanodevice Using a Balanced Pair of Single-Electron Boxes

Takahide Oya,[a,*] Tetsuya Asai,[a] Takashi Fukui,[b] and Yoshihito Amemiya[a]

[a] *Department of Electrical Engineering, Hokkaido University*

[b] *Research Center for Integrated Quantum Electronics, Hokkaido University, Sapporo 060-8628, Japan*

This paper describes a majority-logic gate device that will be useful in developing single-electron integrated circuits. The gate device consists of two identical single-electron boxes combined to form a balanced pair. It accepts three inputs and produces a majority-logic output by using imbalances caused by the input signals; it produces a 1 output if two or three inputs are 1, and a 0 output if two or three inputs are 0. We combine these gate devices into two subsystems, a shift register and an adder, and demonstrate their operation by computer simulation. We also propose a method of fabricating the unit element of the gate device, a minute dot with four coupling arms. We demonstrate by experiments that it is possible to arrange these unit elements on a GaAs substrate, in a self-organizing manner, by means of a process technology that is based on selective-area metalorganic vapor-phase epitaxy.

Keywords: Majority Logic, Nanodevice, Single Electron, Circuit, Adder, Vapor Phase Epitaxy.

1. INTRODUCTION

One of the challenges for nanoelectronics is the development of integrated circuits on the basis of single-electron circuit technology. For this purpose, we must develop single-electron logic devices that are useful in the construction of large digital systems. This paper proposes one such device, a majority-logic device that consists of a balanced pair of single-electron boxes.

The single-electron circuit is an electronic circuit made up of tunneling junctions and capacitors. Such a circuit is designed to implement electronic functions by controlling the transport of individual electrons (for a detailed explanation, see Gravert and Devoret's text).[1] It has been receiving increasing attention because it can be used to produce LSIs that combine huge scales of integration with ultralow levels of power dissipation. To take steps toward this goal, we must develop digital logic devices that manipulate individual electrons to perform complex and large-scale logic operations. In this paper, we will propose a structure for one such logic device and describe its implementation. The device is a single-electron circuit based on the concept of majority logic.

Majority logic is a way of implementing digital operations that is different from Boolean logic. Instead of using the Boolean logic operators (AND, OR, and their complements), the basis of majority logic is the principle of *majority decision*. Majority logic provides logic processes that are far more sophisticated than those of Boolean logic; consequently, majority logic is more powerful in terms of implementing a given digital function with a smaller number of logic devices (Amarel et al.[2] have given details).

The prospects for the practical application of majority logic are wholly dependent on the feasibility of devices that are suitable for majority logic. In the late 1950s, several computer systems based on the majority-logic architecture were developed and constructed for practical use. Their basic element was a nonlinear-reactance device called the *parametron*—a majority-logic device based on the phenomenon of parametric phase-locked subharmonic oscillation. After that, however, majority logic had to leave the stage because the transistor gate circuit—a Boolean logic device by nature—came to be the dominant device in digital electronics. Majority logic, however, can be expected to make a comeback with the recent development of nanotechnology and quantum devices. This is so because quantum devices fabricated by nanotechnology provide functional properties that are excellent in terms of implementing majority-logic operations. The leading examples are the quantum-flux parametron, which is composed of Josephson junction circuits,[3] the quantum cellular automaton consisting of quantum dot arrays,[4] and, in the area of single-electron circuits, the majority logic gate based on Tucker's single-electron inverter.[5]

In this paper we propose a novel single-electron majority gate device that is simpler and is therefore more suitable for LSI applications. The first of the following sections is an outline of the unit function required for majority logic (Section 2). We then propose a logic-gate

*Author to whom correspondence should be addressed.

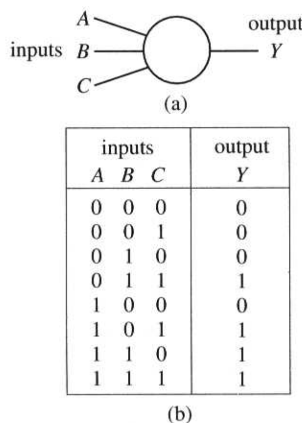
RESEARCH ARTICLE

 © 2002 by American Scientific Publishers 1533-4880/2002/03–04/333/010/$17.00+.25 doi:10.1166/jnn.2002.108

device that implements this unit function. The device consists of a simple single-electron circuit, that is, a balanced pair of single-electron boxes. Its majority-logic operations utilize imbalances caused by the input signals (Section 3). We go on to design two sample subsystems, a shift register and an adder, by combining a number of the gate devices. We also demonstrate the operation of these subsystems by computer simulation (Section 4). After that, we propose a method for fabricating actual gate devices in integrated circuits. The unit element of the gate is a minute dot with four coupling arms. We hope to fabricate unit elements integrated on a substrate by using a process technology based on selective-area metalorganic vapor-phase epitaxy and have taken the first steps in this direction (Section 5). Our fabrication technology is as yet imperfect, and we have yet to fabricate a complete device. We will persevere with improving our technology, however, and we expect to produce prototype single-electron majority-logic ICs in the near future. We hope that this paper will be helpful to readers who are aiming to create novel nanoelectronic devices.

2. UNIT FUNCTION OF MAJORITY LOGIC

The basic operation or unit function of majority logic is to determine the output state that reflects the majority vote of the input states. The logic element, a majority gate, has an odd number of binary inputs and a binary output; the logic symbol for a three-input gate is shown in Figure 1a. The output is a logical 1 when the majority of the inputs is logical 1 and a logical 0 when the majority of inputs is logical 0. The operation of the three-input majority gate is shown in Figure 1b. When, for instance, the three inputs are 0, 1, and 0 (third row in the table), the output is 0, and when the inputs are 1, 0, and 1 (sixth row in the table), the output is 1. Any digital function can be implemented by a combination of majority gates and binary inverters. Further details on majority logic are explained in Ref. 2. Three-input gates suffice for the construction



inputs			output
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Fig. 1. A three-input majority gate. (a) Symbol. (b) Truth table.

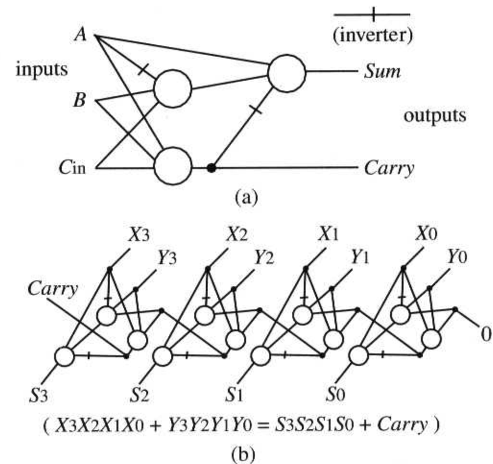


Fig. 2. Majority-logic circuits. (a) Full adder. (b) 4-bit ripple carry adder.

of any logic system, but a larger number of inputs permits a more concise structure for many logic systems. In general, however, only three-input and five-input gates are used.

Majority logic provides a complete and concise implementation of most digital functions encountered in logic-design applications. As an example, the implementation of a single-bit full adder and a 4-bit ripple-carry adder is illustrated in Figure 2. Following the notation for majority-logic circuits, we represent inverters as segments lying at right angles across connecting branches. A full adder is composed of only three gates with two inverters; in contrast, the Boolean-based implementation requires a larger structure, which consists of seven or eight gate elements.

3. USING SINGLE-ELECTRON CIRCUITS TO CONSTRUCT A MAJORITY GATE

3.1. The Single-Electron Box

The main component of the majority-gate device we are proposing is the single-electron box. A single-electron box is the circuit illustrated in Figure 3a. It consists of a tunneling junction C_j in series with a bias capacitor C_L and a bias voltage V_{dd}. It has an island node 1 at which extra electrons are stored ("extra electrons" means electrons that are not canceled out by the background of positive ions in the node material). At the low temperatures at which the Coulomb-blockade effect is established, the number n of extra electrons takes a value such that the electrostatic energy in the circuit (including the bias-voltage source) is minimized. The value of n is 0 at $V_{dd} = 0$, and it increases with V_{dd} because electrons tunnel from the ground to node 1 through junction C_j. Thus, n is a staircase function of V_{dd} (Fig. 3b) and changes discontinuously in one-valued steps at $V_{dd} = (n \pm 1)e/(2C_L)$ (C_L is the bias capacitance, e is the elementary charge).

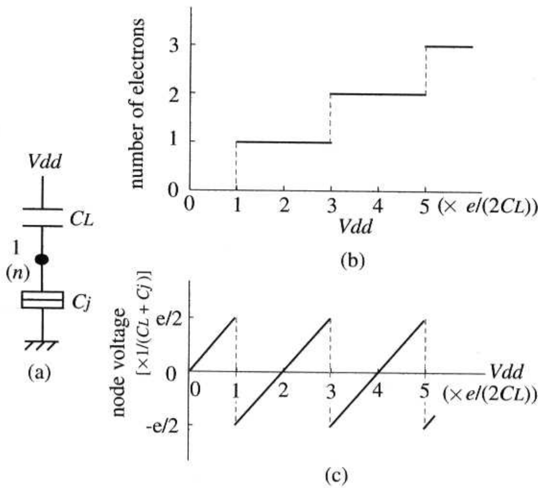


Fig. 3. The single-electron box. (a) Circuit configuration. (b) The static number n of extra electrons on node 1 as a function of bias voltage V_{dd}. (c) The voltage at node 1 as a function of V_{dd}.

This discontinuity means that the voltage at node 1 is a sawtooth function of V_{dd}, as shown in Figure 3c.

3.2. A Balanced Pair of Single-Electron Boxes

The single-electron box is a monostable device, and its internal state is completely determined by the bias voltage V_{dd}; therefore, it is not in itself applicable to binary logic applications. To create a majority-gate device, we develop a bistable device from the single-electron box by combining two single-electron boxes to form a balanced pair, as shown in Figure 4a.

The balanced pair consists of two identical single-electron boxes (two C_L-C_j's) that are connected to each other through a coupling capacitor C_0. This device is a single-electron analogue of the quantum-flux parametron. The balanced pair has two island nodes 1 and 2, and its internal state is expressed by the pair of numbers (n_1, n_2) of excess electrons on the respective nodes. When an electron tunnels from the ground to one node of the balanced pair, that node carries a negative charge, and this suppresses electron tunneling from the ground to the other node. The state of the balanced pair is therefore either (1, 0) or (0, 1) for a given bias voltage V_{dd}.

The balanced pair changes its state with increasing V_{dd}, as shown in Figure 4b. It takes the state (0, 0) at $V_{dd} = 0$. When V_{dd} is increased to the first threshold, V_1, an electron tunnels from the ground to one of the island nodes (the probability of tunneling is equal for the two nodes). The figure shows the situation where this electron tunnels to node 1, and, consequently, the state of the balanced pair changes from (0, 0) to (1, 0). Once an electron has tunneled to one node, tunneling to the other node is suppressed, and the circuit maintains its state until V_{dd} is further increased to reach the second threshold, V_2. The balanced pair therefore shows bistability in the bias range

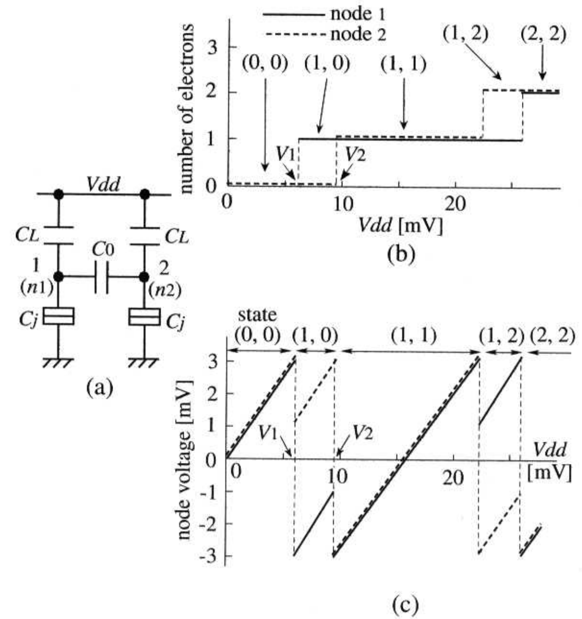


Fig. 4. A balanced pair of single-electron boxes. (a) Circuit configuration. (b) Static numbers n_1 and n_2 of extra electrons on nodes 1 and 2. (c) Voltages at nodes 1 and 2 as functions of V_{dd}.

$V_1 < V_{dd} < V_2$, and we use the balanced pair under this condition. When V_{dd} exceeds V_2, electron tunneling to the other node (node 2) occurs, and the state of the balanced pair changes from (1, 0) to (1, 1). When V_{dd} is further increased, the state changes are in this order: (1, 1), (1, 2) (or (2, 1)), (2, 2),....

Owing to the discrete changes of the state, the voltages at nodes 1 and 2 are a sawtooth function of V_{dd}, as is shown in Figure 4c. With increasing V_{dd}, the voltage at each node increases to a maximum V_m, drops to a minimum $-V_m$ because of electron tunneling, then increases again to repeat the same cycle. In the bistable region ($V_1 < V_{dd} < V_2$), the voltage at node 2 is positive for the state (1, 0) and negative for the state (0, 1). This is utilized in the majority-gate device we will propose in the next subsection.

The expressions for the thresholds and for the node voltages are complicated, so Figure 4b and c, shows the numerical results simulated for a sample set of parameters $C_L = C_j = C_0 = 10$ aF and zero temperature. We used a modified Monte Carlo method in simulation. Kuwamura et al.[6] and the Appendix give details of this method.

3.3. Constructing a Majority-Gate Device

We then used the balanced pair to construct the majority-gate device shown in Figure 5a; the figure illustrates an example of a three-input configuration. The majority gate consists of a balanced pair (two C_L-C_j units connected via C_0), input capacitors (C, connected to node 1), and the *same* number of output capacitors (C, connected to

RESEARCH ARTICLE

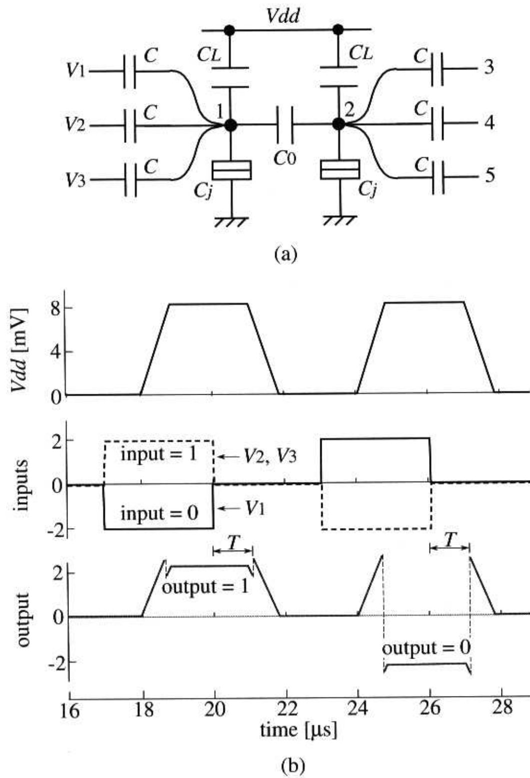


Fig. 5. A majority-gate device. (a) Circuit configuration. (b) Logic operation (simulated).

node 2). Node 1 is the input node and node 2 is the output node. Three input voltages, V_1, V_2, and V_3, are applied to node 1 through the input capacitors. The input capacitors form a voltage-summing network and produce the mean of their inputs on node 1. The balanced pair produces the corresponding logic output on node 2 as illustrated later, and the output is retrieved through output terminals 3, 4, and 5. In operating the majority gate, we use a positive voltage and a negative voltage of equal amplitude to represent the binary logic values, 1 and 0.

The majority gate works in the following way. We start by grounding the output terminals and then gradually increase the bias voltage, V_{dd}, to a value that establishes bistability for the balanced pair. Electron tunneling occurs as V_{dd} increases, and the balanced pair enters either the (1, 0) or (0, 1) state. The state it actually takes is now determined by the polarity on a majority of the inputs. When two or three inputs carry a 1 (the positive voltage), the potential is higher at node 1 than at node 2, so the tunneling junction that is connected to node 1 (the tunneling junction on the left) will be the first to reach its tunneling threshold; consequently, the balanced pair takes the state (1, 0) and produces a positive output voltage on node 2. When two or three inputs carry a 0 (a negative voltage), the balanced pair takes the state (0, 1) and produces a negative output voltage on node 2.

We confirmed the operation of the gate for all input combinations by computer simulation. Figure 5b shows some of the results simulated with this set of parameters: $C_{\mathrm{L}} = C_{\mathrm{j}} = C_0 = 10$ aF, $C = 2$ aF, tunneling-junction conductance $=1$ μS, and zero temperature. The bias voltage, V_{dd}, is the trapezoidal clock pulse shown in the upper plot of Figure 5b (in the simulation, we approximated the linear rise and fall of the trapezoid as staircase voltage waves, in which the duration of each step was 10 ns and the difference between two sequential steps was 0.1 mV). Three inputs (V_1, V_2, V_3) are applied synchronously with this bias clock, V_{dd}. They are rectangular pulses, positive for logical 1 and negative for 0 (the middle plot in Fig. 5b); the figure shows the inputs (0, 1, 1) and (1, 0, 0) being applied in sequence. Depending on the majority of the inputs, the voltage at the output node (node 2) changes from 0 to become positive (1-valued) or negative (0-valued) (the bottom plot of Fig. 5b). With a 0 output, the output voltage initially increases with the increase in the bias voltage, V_{dd}, and then turns negative as electron tunneling takes place. The output established in each clock cycle is maintained after the input pulses have been turned off, until the bias voltage has fallen below the threshold for tunneling (duration T in the bottom plot of Fig. 5b).

4. USING THE MAJORITY GATE CIRCUITS IN SUBSYSTEM DESIGN

4.1. Interstage Coupling

Any logic function can be implemented by combining identical gates into a cascade, with the output capacitors of one gate acting as the input capacitors of the succeeding gates. An example is illustrated in Figure 6a. The majority gate that we propose is bilateral, so we control

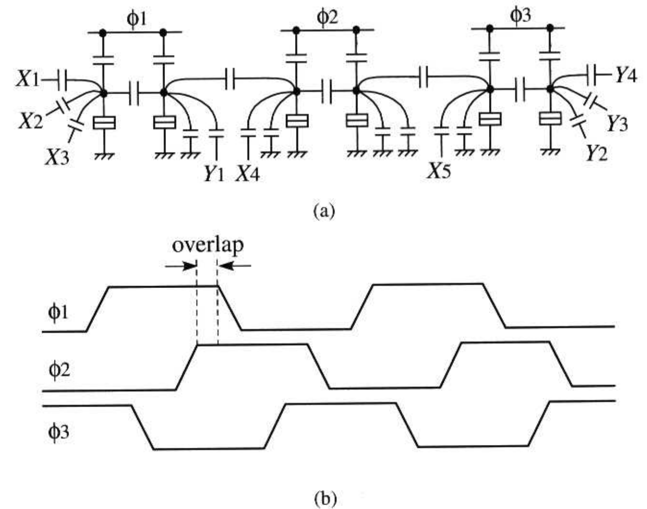


Fig. 6. Gating with a three-phase clock. (a) Configuration of interstage coupling. X_1 to X_5 denote inputs from other gates, and Y_1 to Y_4 denote outputs to other gates. (b) The three-phase clock pulses used to excite the gates.

RESEARCH ARTICLE

the direction of signal flow by gating it with a three-phase clock. This is analogous to an Esaki-diode pair circuit and the quantum-flux-parametron circuit. We divide the gate circuits into three groups and excite each group in turn by one phase of the three clock signals, ϕ_1 to ϕ_3, as shown in Figure 6b. For instance, in Figure 6a, the first or left-most gate (and the fourth gate, etc.) belongs to the first group and is exited by the ϕ_1-phase clock; the middle gate (and the fifth gate, etc.) belongs to the second group and is exited by the ϕ_2-phase clock; the rightmost gate (and the sixth gate, etc.) belongs to the third group and is exited by the ϕ_3-phase clock. The phases of the three clock signals overlap so that the output of a stage will be established while the preceding stage is maintaining its output; signals are thus transmitted from one gate to the next. The direction of signal flow is determined by the relative timing of the three phases; in Figure 6a, it is rightward.

In each majority gate, the numbers of capacitors connected to the input and output nodes must be the same to maintain the balance between two single-electron boxes in each pair. The middle gate of Figure 6b has two inputs and one output, so one dummy capacitor is connected between the input node and ground, and two dummy capacitors are connected between the output node and ground.

4.2. Inversion Coupling

Logical inversion is obtained simply by applying the input signal to be inverted to an output node of the gate. An example is given in Figure 7. In this circuit, applying voltage signal A to output node 2 is equivalent to applying an opposite-polarity signal to input node 1, so the gate operates as if it were receiving the logical inverse of A. Inversion is thus achieved without a special device.

4.3. Shift Register

A shift register is constructed by connecting a number of the gates to form a chain. A sample configuration with nine gates (1 through 9) is shown in Figure 8a; the code (ϕ_1 through ϕ_3) above each gate indicates the clock phase

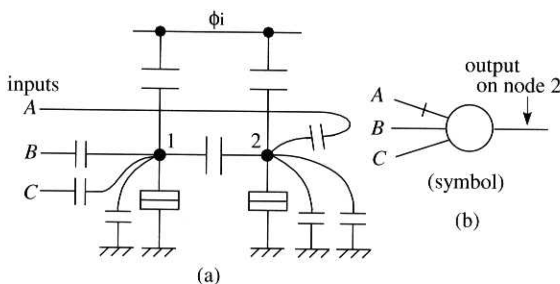
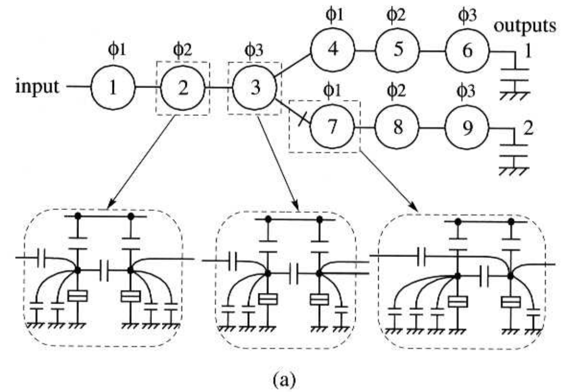


Fig. 7. Inversion coupling. (a) A majority gate with an input applied to the output node. (b) The symbolic representation of this gate.



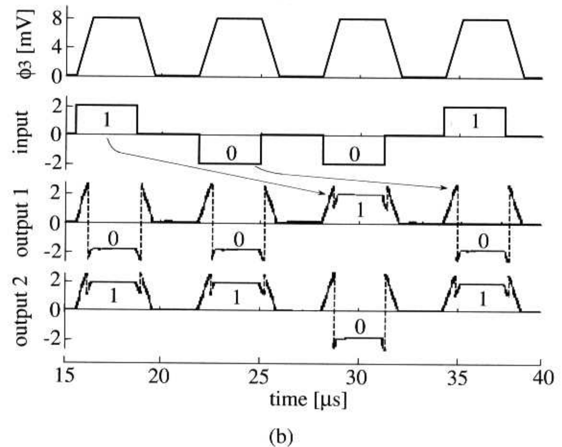


Fig. 8. Shift register. (a) Circuit configuration. (b) Output waveforms on nodes 1 and 2 (simulated). The waveforms of the ϕ_3-clock signal and the input are also shown. Clocks ϕ_1 and ϕ_2 are omitted.

with which the gate is driven (the clocks have the timing shown in Fig. 6b). The shift register has a fork after gate 3. Gate 3 is inversion-coupled to gate 7, so the output signal on node 2 is the inverse of that on node 1.

A simulated result for the signal transmission along the shift register is illustrated in Figure 8b; the device parameters were as given in Section 3.3. The input applied to the shift register is the sequence "100100100...." After a delay of two clock periods, the sequence "100100100..." appears on node 1 and the sequence "011011011..." on node 2. Four cycles of the sequence of operations are shown in the figure.

4.4. Full Adder

Figure 2a illustrates an implementation of the full adder shown in Figure 9a. The core of the adder consists of majority gates 5, 6, and 7. Gates 1 to 3 are input buffers that accept the inputs to drive the succeeding gates. Gate 4 is a delay buffer used to transfer the signal from gate 1 to gate 7 with the correct clock timing. The adder accepts three inputs, the augend A, addend B, and carry input C_{in}, and produces the corresponding carry and sum outputs C_o and S_o. The inputs are taken in while ϕ_3 is high; the carry output C_o is produced when ϕ_2 goes high, and the

RESEARCH ARTICLE

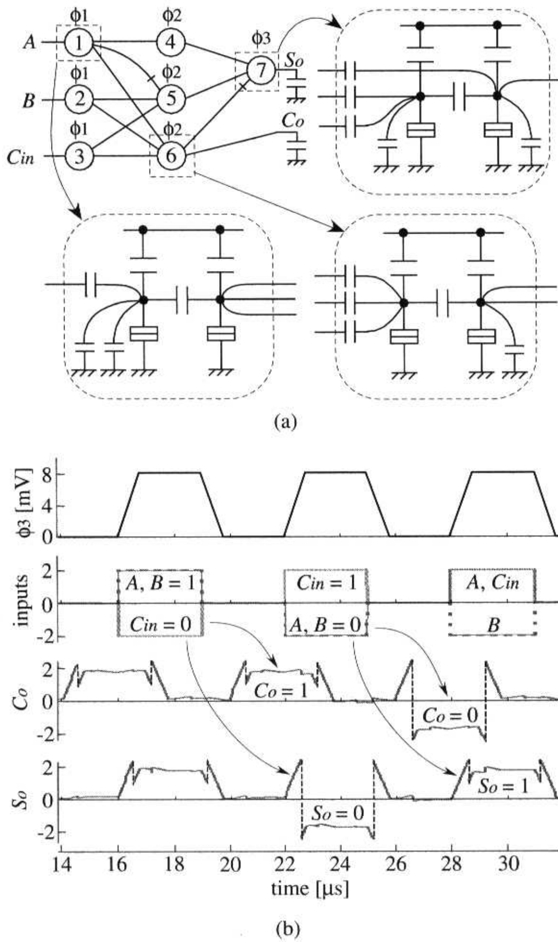


Fig. 9. Full adder. (a) Circuit configuration. (b) Output waveforms of carry C_o and sum S_o, the ϕ_3-clock, and the inputs (simulated). Clocks ϕ_1 and ϕ_2 are not shown.

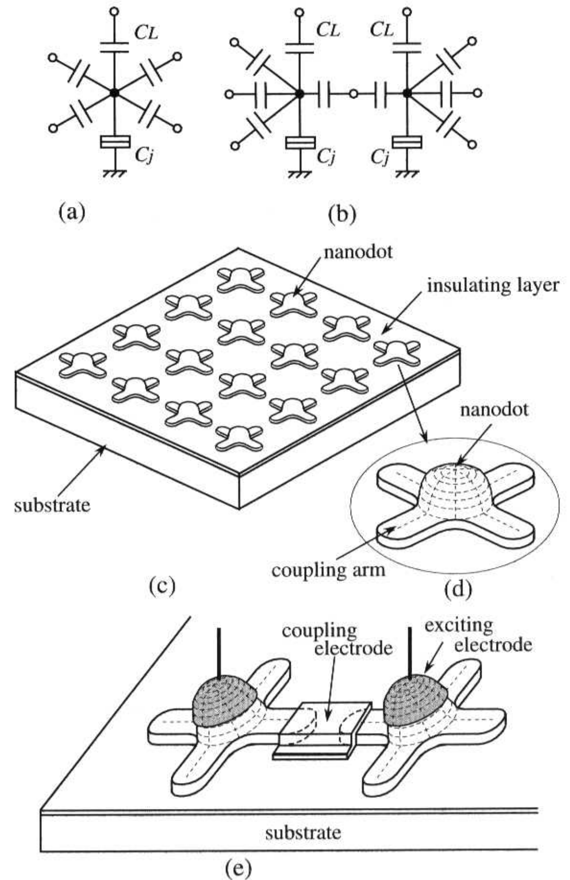


Fig. 10. The unit element of the majority-gate device—(a) single-electron box with four coupling terminals. (b) A majority gate consisting of two elements. (c) A regular arrangement of nanodots on a substrate. (d) Nanodots, each with four coupling arms, over an insulating layer. (e) Using an electrode to capacitively couple two nanodots and form a majority gate. The exciting electrodes are also shown.

sum output S_o is produced when ϕ_3 again goes high. The respective delays between the inputs and the carry output and the sum output are two-thirds of a clock period and one clock period.

We simulated the add operation and confirmed correct operation for all input combinations. Three clock cycles of results are shown in Figure 9b; the device parameters were as given in Section 3.3. In the figure, two sets of inputs $(A, B, C_{in}) = (0, 0, 1)$ and $(1, 1, 0)$ are sequentially entered, and the correct outputs $(C_o, S_o) = (0, 1)$ and $(1, 0)$ are produced in response.

5. USING SEMICONDUCTOR PROCESS TECHNOLOGY TO CONSTRUCT ACTUAL DEVICES

5.1. The Unit Element of the Majority-Gate Device

The unit element of our majority device is a single-electron box with four terminals for capacitive coupling, illustrated in Figure 10a. Joining two of these unit elements produces a majority gate with three input and three output terminals (Fig. 10b). We propose the following steps 1 to 4 as a process for forming many unit elements on a substrate to build logic circuits. We start with a conductive substrate.

1. Grow an insulating layer over the whole surface of the substrate.
2. Form minute conductive dots (hereafter called "nanodots") in a two-dimensionally regular arrangement on the insulating layer that covers the substrate (Fig. 10c). For use as unit elements, we form the nanodots such that
   i. each nanodot has four arms for coupling with other nanodots (Fig. 10d), and
   ii. a tunneling junction runs between the nanodot and the substrate beneath the nanodot.
3. Grow an insulating layer to cover all of the nanodots and their arms.
4. Use electrodes to interconnect the nanodots into majority gates and thus produce the desired logic

RESEARCH ARTICLE

functions. Capacitive coupling between two nanodots can be achieved by simply forming an electrode to connect two arms of a pair of nanodots (Fig. 10e). The bias capacitor through which the gates are excited can be made by forming an electrode on each nanodot. We can thus fabricate various logic circuits by simply designing appropriate patterns of electrodes.

The key point in this process is to form the regular arrangement of nanodots with their coupling arms and tunneling junctions. For this purpose, we are developing a fabrication technology that is based on the self-organized crystal growth that takes place in selective-area metalorganic vapor-phase epitaxy. Although our technology is not yet perfect, we have succeeded in forming regular arrays of GaAs nanodots on an insulating layer that covers a conductive GaAs substrate. The technology we have developed is discussed in Section 5.2. At the present stage, tunneling junctions are not formed beneath the nanodots. We will propose a method for forming tunneling junctions in Section 5.3.

5.2. Fabricating Nanodot Arrays by Means of Selective-Area Metalorganic Vapor-Phase Epitaxy

Our fabrication technology is based on selective-area metalorganic vapor phase epitaxy (SA-MOVPE). This is used to form, on an n-type GaAs substrate, an arrangement of n-type nanodots with arms buried in insulating layers of AlGaAs. This process makes use of the dependence on orientation of crystal growth rates in SA-MOVPE. A detailed explanation of SA-MOVPE is given in Refs. 7 and 8. The sequence of processes we used was as follows.

5.2.1. Preparing the Substrate. The initial substrate was an n-type wafer of GaAs with the crystal surface in the ⟨001⟩ orientation. Plasma deposition was used to cover this surface with a 40-nm-thick layer of silicon nitride (SiN_x).

5.2.2. Forming SiN_x Masks. The SiN_x layer was formed into a regularly arranged pattern of squares, shown in Figure 11, by electron-beam lithography and wet chemical etching. Each SiN_x square was 200 nm on a side, and the rows and columns of the squares were arranged in the ⟨010⟩ and ⟨100⟩ directions, each with a pitch of 400 nm. The squares of SiN_x act as a mask for the succeeding steps of crystal growth. The GaAs surface is exposed in the regions where the SiN_x has been etched away ("intersections" and "streets" in the figure). In the succeeding steps, GaAs and $Al_xGa_{1-x}As$ crystals are selectively grown on this exposed GaAs surface.

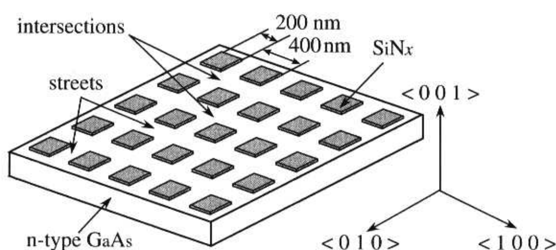


Fig. 11. GaAs substrate with SiN_x mask patterns. The GaAs surface is exposed in the "intersections" and "streets."

5.2.3. Crystal Growth. The process conditions used in growing the GaAs and $Al_xGa_{1-x}As$ were as follows. We used an RF-heated, horizontal quartz reactor. The carrier gas was hydrogen (H_2). The source materials, mixed with the carrier gas, were trimethylgallium ($(CH_3)_3Ga$), triethylaluminum ($(C_2H_5)_3Al$), and arsine (AsH_3). The partial pressures were 1.9×10^{-6} atm for $(CH_3)_3Ga$ and 6.7×10^{-7} atm for $(C_2H_5)_3Al$; the partial pressure of AsH_3 was varied during the process (described later). The total pressure (i.e., including the carrier gas) was 0.1 atm, and the growth temperature was 750 °C. The growth rates were 0.5 μm/h for GaAs layers and 0.8 μm/h for AlGaAs layers (these values are for growth on the (001) plane with a large area).

5.2.4. Growing a Buffer Layer. With the SiN_x masks in place, a thin layer of GaAs layer (about 70 nm thick) was grown as a buffer layer on the substrate. The partial pressure of AsH_3 was set at 6.7×10^{-5} atm. The GaAs layer grown was n-type and included no dopant.

5.2.5. Growing the First Insulating Layer. An insulating layer of $Al_{0.35}Ga_{0.65}As$ was grown on the GaAs buffer. The partial pressure of AsH_3 was set at 6.7×10^{-5} atm. A trace of oxygen was added to the carrier gas to make the GaAlAs layer insulating. The growth time was 20 min, and the thickness of the layer was 130 nm in the center of the intersection on the substrate (for "intersection," see Fig. 11).

5.2.6. Forming the Nanodots. An n-type layer of GaAs was grown on the insulating layer of AlGaAs. The partial pressure of AsH_3 was set to 5.0×10^{-4} atm. The growth time was 5 min, and the layer was 30 nm thick in the center of the intersection. As will be detailed in the next subsection, the nanodots of GaAs, each cross-shaped and with four arms, were automatically formed on the AlGaAs layer because of the orientation dependence of the growth rate.

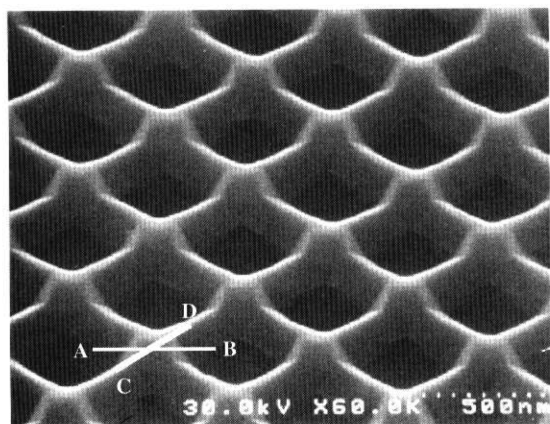
5.2.7. Growing the Second Insulating Layer. An insulating layer of $Al_{0.35}Ga_{0.65}As$ was grown on the GaAs nanodots. The partial pressure of AsH_3 was set at 5.0×10^{-4} atm. A trace of oxygen was added to the carrier gas. The growth time was 15 min, and the layer was 100 nm thick in the center of the intersection.

5.3. The Resulting Structure—A Regular Arrangement of Nanodots with Arms

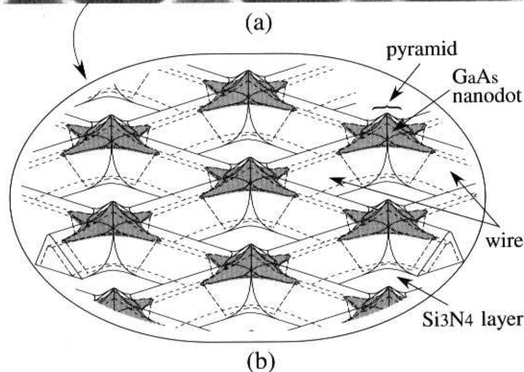
Figure 12a shows the structure produced by our process. This image is from a scanning electron microscope (SEM). Figure 12b is a schematic enlargement. The pattern is of parallel crossing lines and consists of regularly arranged "pyramids" and "wires" that connect the pyramids. The pyramids and the wires grow on the intersections and streets, respectively, that is, where the GaAs substrate is exposed. The structure consists mainly of insulating AlGaAs; each GaAs nanodot is buried in a pyramid of AlGaAs and stretches its four arms into the four wires as depicted in Figure 12b.

Figure 13a is a cross-sectional view of the pyramid (a section along line AB in Fig. 12a), and Figure 13b shows an actual image, taken by SEM, of this section. In the growth of AlGaAs, facet sidewalls in the {111} orientation preferentially appear at the bottom of the pyramid, while {114} facets appear in the upper regions. In the growth of GaAs, {111} facets again appear at the bottom, but {113} facets appear in the upper regions. This difference results in the formation of a GaAs nanodot buried in an AlGaAs pyramid, as shown in the figures.

Figure 13c illustrates a longitudinal section through the pyramid and wires (a section along line CD in Fig. 12a).



(a)



(b)

Fig. 12. A three-dimensional view of the fabricated structure with its regular arrangement of nanodots that have arms. (a) An image of the surface via an SEM. (b) A schematic enlargement, with the GaAs nanodots shaded in gray.

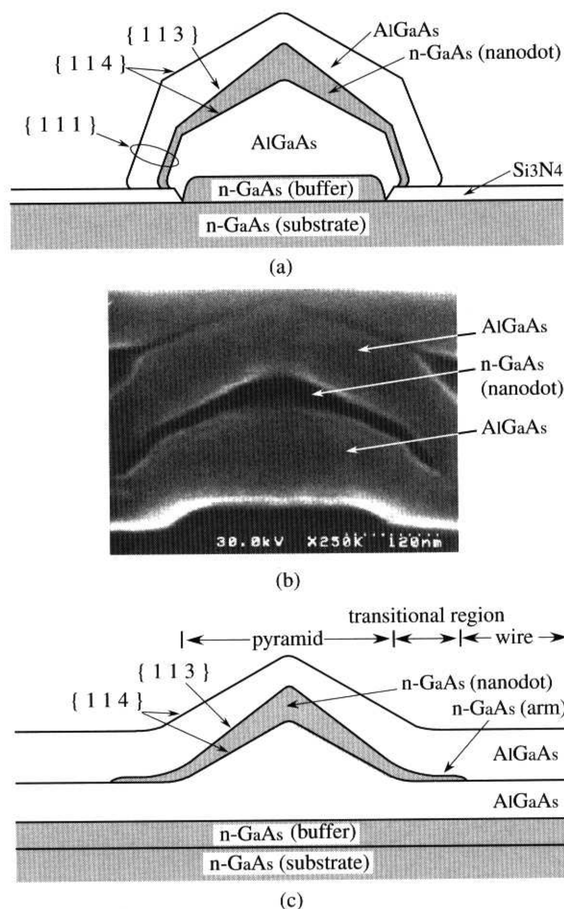


Fig. 13. Sections through a pyramid and the attached wires. (a) Cross section along line AB of Figure 12a. (b) A SEM image of a section divided along line AB of Figure 12a. (c) A longitudinal section along line CD of Figure 12a.

The outline of the wires is formed by the first stage of AlGaAs growth. Each wire is in the shape of a triangular prism that lies on the substrate (on the street), and its sidewalls are {011} facets. While the wire has {011} facets, the pyramid has {114} facets, so a transitional region (region of competition) appears in the region where the wire meets the pyramid. In the succeeding stage of GaAs growth, a layer of GaAs is formed on the AlGaAs pyramid and on the transitional regions, but it is not formed on the greater portion of the wire because GaAs does not grow on {011} planes under the process conditions we used. As a result, we obtain the cross-shaped GaAs nanodot with its four coupling arms outflung in four directions, as depicted in Figure 12b.

5.4. Forming Tunneling Junctions beneath the Nanodots

The nanodot structure we formed does not include a tunneling junction. To form a tunneling junction between each nanodot and the substrate, we are developing the

RESEARCH ARTICLE

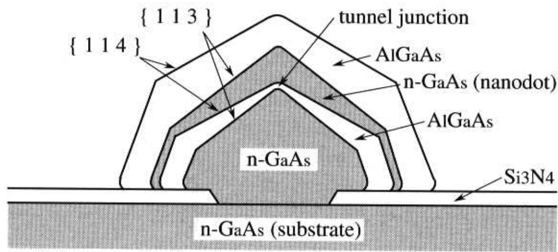


Fig. 14. The modified structure, with a tunnel junction between the nanodot and the substrate.

modified structure illustrated in Figure 14, a cross-sectional view of the new type of pyramid. This differs from the structure shown in Figure 13a in that an n-type GaAs pyramid is grown on the substrate before the first stage of AlGaAs growth. We can form this structure by growing GaAs pyramids, instead of the thin buffer layer of GaAs, in step 4 in Section 5.2. The subsequent steps are the same as those described in Section 5.2. The upper regions of the GaAs pyramid will have {113} facets, whereas the first layer of AlGaAs will have {114} facets. Consequently, the first layer of AlGaAs, sandwiched between the n-GaAs pyramid and the n-GaAs nanodot, will be very thin at the point of the GaAs pyramid; a tunneling junction between the GaAs nanodot and the GaAs pyramid (therefore the GaAs substrate) will thus be formed. With this modified structure, we will be able to fabricate majority-gate devices and proceed to develop majority-logic LSIs.

6. SUMMARY

We proposed a majority-gate device that is useful in developing single-electron integrated circuits. The gate device consists of two identical single-electron boxes combined to form a balanced pair. It produces a majority-logic output by using imbalances caused by the input signals. We combined these gate devices into two sample subsystems, a shift register and an adder, and confirmed their operation by computer simulation. We also proposed a method of fabricating the unit element of the gate device, a nanodot with four coupling arms. We demonstrated by experiments that a regular arrangement of GaAs nanodots can be formed on a substrate, in a self-organizing manner, by means of a process technology that is based on the selective-area metalorganic vapor phase epitaxy. With these results, we will be able to develop single-electron LSIs that are based on the majority-logic architecture.

7. APPENDIX

The following is the procedure for the Monte Carlo simulation for single-electron circuits that we used in this paper. This is an excerpt from the work by Kuwamura et al.[6] For details, see the reference. This method of simulation calculates the time-dependent behavior of single-electron circuits operated with rectangular-voltage inputs. The effect of the cotunneling phenomenon is ignored in the calculations.

Consider a lumped-parameter circuit that consists of tunneling junctions (tunneling-junction capacitors), ordinary or nontunneling capacitors, and input voltage sources (power and clock supplies, and signals). The internal state of the circuit is expressed by a set of numbers that represent the number of excess electrons on the nodes of the circuit (in the following description, "state" refers to this set of electron numbers). Choose a starting state for the circuit, and set time $= 0$. Electron tunneling, or the time-dependent behavior of the circuit, for the given values of source voltages, is simulated in the following way.

Step 1. Compute the electrostatic energy E_0 (the sum of electrostatic energy on the tunneling junctions and ordinary capacitors) in the current state. Then enumerate all possible subsequent states and compute the electrostatic energy E_{i1} for each subsequent state i (a subsequent state means a state into which the current state can be transformed by tunneling of a single electron; if the number of tunneling junctions is N, there are $2N$ possible tunneling events and therefore $2N$ subsequent states). Also compute the energy E_{i2} that the voltage sources will supply in transforming the circuit from the current state to each subsequent state i.

Step 2. Compute the energy difference ΔE_i $(=E_0 - E_{i1} + E_{i2})$ for each subsequent state. From the value of ΔE_i, calculate the waiting time for each of the $2N$ possible tunneling events. The waiting time τ_i is given as

$$\tau_i = (1/\Gamma_i)\ln(1/\gamma)$$

where γ is a uniform random number $(0 < \gamma < 1)$ and Γ_i is the mean tunneling rate. For γ, we use a pseudorandom number that is generated by computer for each tunneling event. The mean tunneling rate Γ_i is the mean number of electrons that tunnel in 1 s and is given by

$$\Gamma_i = G_t \Delta E_i / \{e^2(1 - \exp(-\Delta E_i/(k_B T)))\}$$

where G_t is the tunneling conductance of the tunnel-junction capacitor, k_B is the Boltzmann constant, e is the elementary charge, and T is the temperature (at zero temperature, $\Gamma_i = G_t \Delta E_i / e^2$ for $\Delta E_i > 0$, and $\Gamma_i = 0$ for $\Delta E_i < 0$).

Step 3. After calculating the waiting time τ_i for each of the $2N$ possible tunneling events, take the tunneling event that has the shortest waiting time, and accept the corresponding subsequent state as the current state. Then, put the time forward by τ_i and return to step 1 and carry out the next iteration.

References and Notes

1. H. Gravert and M. H. Devoret, *Single Charge Tunneling—Coulomb Blockade Phenomena in Nanostructures*, Plenum, New York **(1992)**.
2. S. Amarel, G. Cooke, and R. O. Winder, *IEEE Trans. Electron. Comput.* 13, 4 **(1964)**.

RESEARCH ARTICLE

3. K. F. Loe and E. Goto, *IEEE Trans. Magn.* MAG-21, 884 (**1985**).
4. C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, *Nanotechnology* 4, 49 (**1993**).
5. H. Iwamura, M. Akazawa, and Y. Amemiya, *IEICE Trans. Electron.* E81-C, 42 (**1998**).
6. N. Kuwamura, K. Taniguchi, and C. Hamakawa, *IEICE Trans. Electron.* J77-C-II, 221 (**1994**).
7. K. Kumakura, K. Nakakoshi, J. Motohisa, T. Fukui, and H. Hasegawa, *Jpn. J. Appl. Phys.* 34, 4387 (**1995**).
8. K. Kumakura, J. Motohisa, and T. Fukui, *J. Crystal Growth* 170, 700 (**1997**).

Received: 4 December 2001. Revised/Accepted: 26 March 2002.