Jpn. J. Appl. Phys. Vol. 40 (2001) pp. 4485–4488
Part 1, No. 7, July 2001

Quantum-Dot Logic Circuits Based on the Shared Binary-Decision Diagram

Takashi YAMADA[1,*], Yoshitaka KINOSHITA[1], Seiya KASAI[1], Hideki HASEGAWA[1,2] and Yoshihito AMEMIYA[1]

[1]*Department of Electrical Engineering, Hokkaido University, Kita 13, Nishi 8, Sapporo 060-8628, Japan*
[2]*Research Center for Interface Quantum Electronics, Hokkaido University, Kita 13, Nishi 8, Sapporo 060-8628, Japan*

(Received September 5, 2000; accepted for publication March 26, 2001)

We propose a method of constructing quantum-dot logic circuits that can be used to develop large digital systems with ultralow power consumption. These dot circuits consist of dot arrays fabricated using a wrap-gate structure, and they perform logic operations on the basis of the binary decision diagram. Sample dot circuits such as elementary logic gates and adder subsystems are designed. The operation of the designed circuits is confirmed by computer simulation.

KEYWORDS: binary decision diagram, quantum dot logic circuit, wrap gate, dot array

1. Introduction

One promising research area in quantum-dot electronics is the development of next-generation LSIs that combine large-scale integration and low power consumption. To move toward this goal, we must first develop novel circuit technology for implementing digital logic operations using electron transport in quantum-dot arrays. We propose a method of constructing quantum-dot logic circuits that can be used to develop large digital systems. These quantum-dot circuits consist of dot arrays fabricated using a wrap-gate structure, and they perform logic operations on the basis of the binary decision diagram.

In the following sections, we first outline the wrap gate structure (§2) and the binary decision diagram (§3), and we then propose a wrap-gate dot device for implementing the unit function required for binary-decision logic operation (§4). We then describe the designs of sample logic circuits (an elementary logic gate and an adder subsystem) by combining the dot devices (§5) and confirm their logic operation by computer simulation (§6).

2. Quantum-Dot Arrays in a Wrap-gate Structure

To construct a quantum-dot circuit, we use quantum-dot arrays formed in a wrap-gate structure (WPG). The WPG is a quantum wire with two or more metal (or conductive) gates for controlling electron transport along the wire (refs. 1 and 2). Two sample WPG structures are shown in Fig. 1. A Schottky WPG [Fig. 1(a)] consists of a quantum wire of two-dimensional electron gas (2DEG) in a GaAs–GaAlAs heterostructure, with Schottky gates wrapped around the quantum wire. In contrast, a MOS WPG [Fig. 1(b)] uses an n-type silicon quantum wire, with polysilicon MOS gates wrapped around the wire. In both WPGs, depletion layers or potential barriers extend from the gates into the quantum wire. These potential barriers divide the quantum wire into many separate dots, forming a quantum-dot array [Fig. 1(c)]. In this array, electrons can be transported by tunneling through the potential barrier from one dot to adjacent dots. By changing the layout of the quantum wire and the gate electrodes, quantum-dot arrays of various configurations can be constructed.

A notable characteristic of the WPG dot array is that electron transport between two adjacent dots can be modulated by controlling the voltage of the gate to regulate the width of the potential barrier. Electrons will be transferred by tunneling from one dot to another if the gate is set to a low negative voltage, while no electrons will be transferred if the gate is set to a high negative voltage. This enables us to turn electron transport in the dot array on and off.

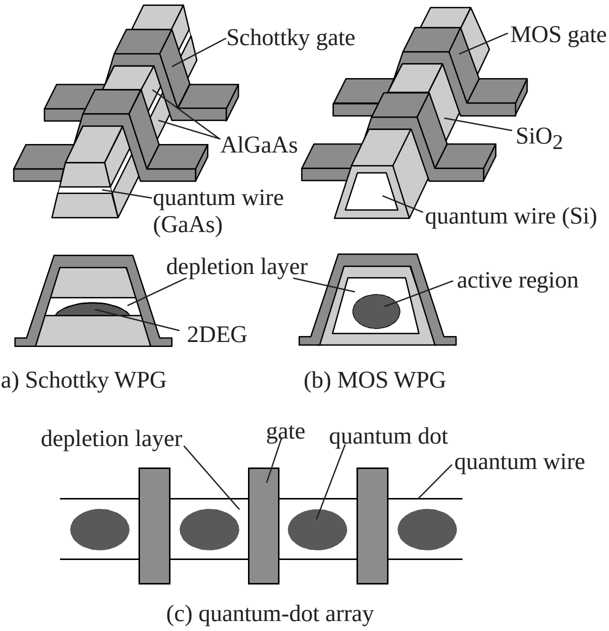


Fig. 1. WPG and a quantum-dot array.

3. Binary Decision Diagram

The binary decision diagram (BDD) is a way of representing logic functions using a directed graph (ref. 3). Simple examples of BDDs are shown in Figs. 2(a) and 2(b). A BDD is a graph consisting of many nodes and two terminals, with each node labeled by a variable (x_1, x_2, etc.). Each node has two entry branches (1-branch and 0-branch) and an exit. In each node, one branch corresponding to the value of the variable (x_i) is connected to the exit and the other branch is disconnected; *e.g.*, if variable x_i is logical 1, only the 1-branch is connected to the exit and the 0-branch is not. For a given set of variables, there exists one and only one path from either terminal to the root. To determine the value of the logic function, we check whether the 1-terminal is connected to the root; if the connection is established, the function is logical 1,

[*] E-mail: yamada@sapiens-ei.eng.hokudai.ac.jp

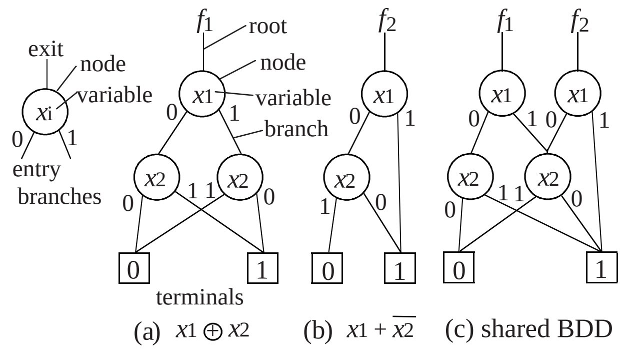


Fig. 2. Examples of the BDD.

and if not, the function is logical 0.

Most digital systems contain multiple output functions that are closely related to one another, and these functions can be represented by a single graph with multiple roots (one root for each function), as shown in Fig. 2(c). This type of BDD is called a shared BDD. Another example of a shared BDD, a representation of 2-bit addition, which has three roots for three output bits (a 2-bit sum output and a 1-bit carry output), is shown in Fig. 3.

4. Constructing a BDD Node Device Using a WPG Dot Array

The function required for the nodes of BDDs is two-way switching controlled by an input variable. To implement this function using a WPG dot array, we propose the node device shown in Fig. 4(a). The device consists of a forked quantum wire with two gates (gate 1 and gate 0) wrapped around the wire. A binary voltage input (and its complement), specifying the value of a variable, is applied to gate 1 (and gate 0) to regulate the resistance of the tunnel junctions under the gates; *e.g.*, if variable x_i is logical 1, a low negative voltage is applied to gate 1 and a high negative voltage to gate 0. The device enables electron transport by tunneling from the 1-branch to the exit if $x_i = 1$ ($\overline{x_i} = 0$), and from the 0-branch to the exit if $x_i = 0$ ($\overline{x_i} = 1$). The device can be approximated by the equivalent circuit shown in Fig. 4(b).

5. Designing Quantum-Dot BDD Circuits

Any combinational logic can be implemented as a quantum-dot circuit by connecting the node devices in cascade formation to build the tree of a BDD graph (see Appendix A). In operating the circuit, we apply input voltages to the gates of the devices and inject electrons into the circuit from the 1-terminal, and then observe at each root whether the electrons flow out or not.

As an example, we show here a sample configuration of a dot circuit implementing the shared BDD shown in Fig. 2(c). We designed the circuit on the basis of the following principles.

(1) In composing the circuit, the 0-terminal and related branch connections in BDDs are unnecessary and can be omitted.

(2) A gradient potential from the 1-terminal to the roots must be appropriately established so that, in every node device, electrons will flow from the entry branches to the exit. This can be achieved by inserting additional node devices (dummy tunnel junctions) in electron paths so that every path from the 1-terminal to a root will have the same number of tunnel junctions.

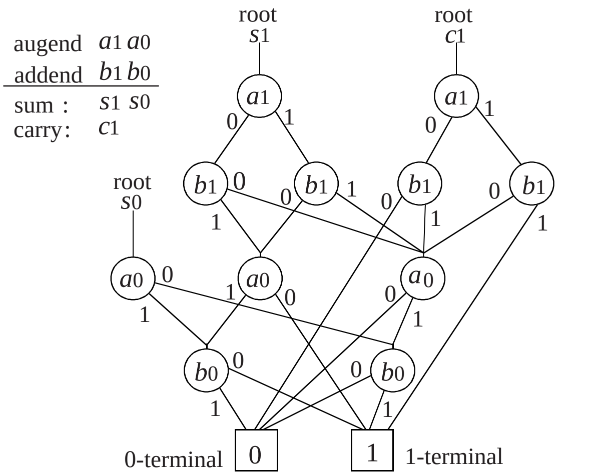
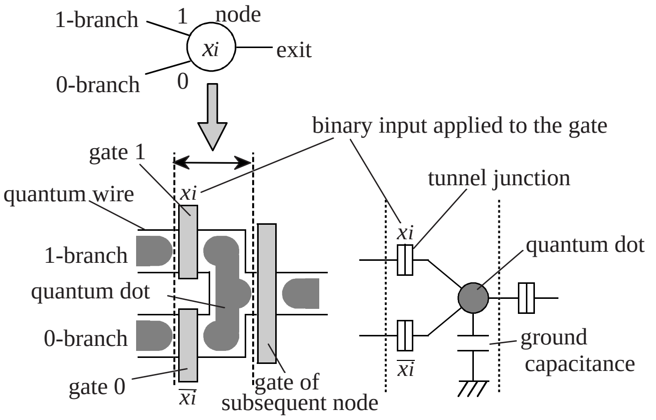


Fig. 3. Shared-BDD representation for 2-bit addition.

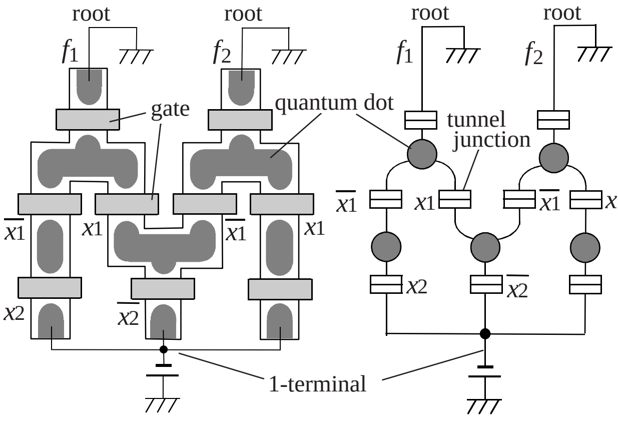


(a) device structure (b) equivalent circuit

Fig. 4. A node device consisting of a WPG dot array.

The resultant dot circuit is shown in Fig. 5(a). The circuit accepts two voltage inputs (specifying variables x_1 and x_2) and produces two corresponding outputs (f_1 and f_2). The inputs are applied to the gates labeled with variables x_1, $\overline{x_1}$, x_2, and $\overline{x_2}$ to modulate (turn on and off) electron transport through the tunnel junction under the gates. The nonlabeled gates are simple bias electrodes forming the dummy junctions. They are biased at a low negative voltage, so the tunnel junctions under the nonlabeled gates are always in the 'on' state. The equivalent circuit for this dot circuit is shown in Fig. 5(b). Each quantum dot has a ground capacitance, but for simplicity the capacitance is omitted in the figure.

To operate the dot circuit, we ground all the roots and apply a negative power voltage to the 1-terminal. Electrons are injected from the power source into the dot circuit and transported toward each root along the paths specified by the variables. The logical value for an output is 1 if electrons can reach the corresponding root, and 0 if they cannot. The dot circuit is analogous in operation to pass-transistor circuits composed of metal-oxide-semiconductor field-effect transis-



(a) Dot-array layout for the circuit (b) Equivalent circuit

Fig. 5. Dot circuit implementing the BDD graph in Fig. 2(c). The ground capacitance for each dot is omitted.

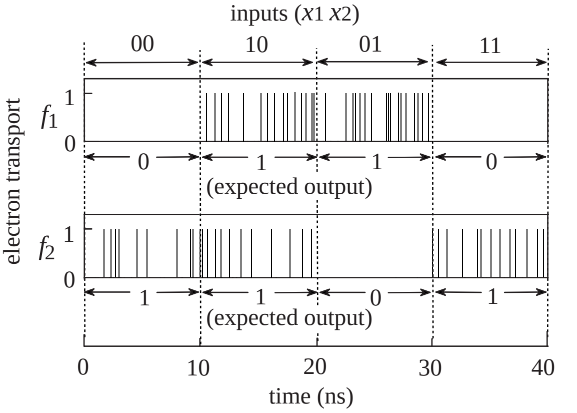


Fig. 6. Operation of the dot circuit in Fig. 5 (simulation). Each impulse signifies the arrival of an electron at a root. The expected logical values for each output bit are written in the figure. Temperature is assumed to be 0 K.

tors (MOSFETs), and it can be considered to be an ultralow-power version of a pass-transistor circuit. In the dot circuit, the electron flow (therefore the power consumption) is regulated by the *Coulomb blockade* in dot-array transmission lines, and it therefore can be set to a far smaller quantity than that in ordinary pass-transistor circuits (see Appendix B). The number of electrons flowing through the circuit during one logic operation can be reduced to ten or twenty by adjusting the tunnel junction and the ground capacitances [see Figs. 6 and 8(a)].

We confirmed the logic operation of the sample circuit by computer simulation, using the equivalent circuit shown in Fig. 5(b), with a set of device parameters typical of a Schottky-WPG dot array, the tunnel junction capacitance was 10 aF, the ground capacitance of a dot was 20 aF, and the tunnel junction resistance was 1 MΩ in the 'on' state and 10 GΩ in the 'off' state. The voltage of the 1-terminal was set to −3.5 mV as an example and the temperature was assumed to be absolute zero. The results are shown in Fig. 6. Four input combinations ($x_1x_2 = 00, 10, 01, 11$) were applied in sequence to the gates of the circuit at 10 ns intervals. Each impulse in the figure signifies the arrival of an electron at a root. Thus, a total of fifteen electrons flowed out of root f_1 during the period from 10 ns to 20 ns. The circuit produces the expected correct outputs. (In this simulation, we used a modified Monte Carlo method. For details of this method, see ref. 4.) The number of output electrons, therefore the power consumption of the circuit, depends on the 1-terminal voltage, the device parameters, and the size and the configuration of the circuit. It can be controlled by adjusting the 1-terminal voltage.

6. Simulating the Logic Operation in a 2-Bit Adder Subsystem

As an example of a larger system, we designed a quantum-dot circuit implementing the shared BDD for 2-bit addition shown in Fig. 3. The designed circuit is shown in Fig. 7 in the form of an equivalent circuit. The circuit accepts two 2-bit binary inputs (augend a_1a_0 and addend b_1b_0) and produces the corresponding 2-bit sum output (s_1s_0) and a one-bit carry output (c_1).

We confirmed, by computer simulation, that the designed

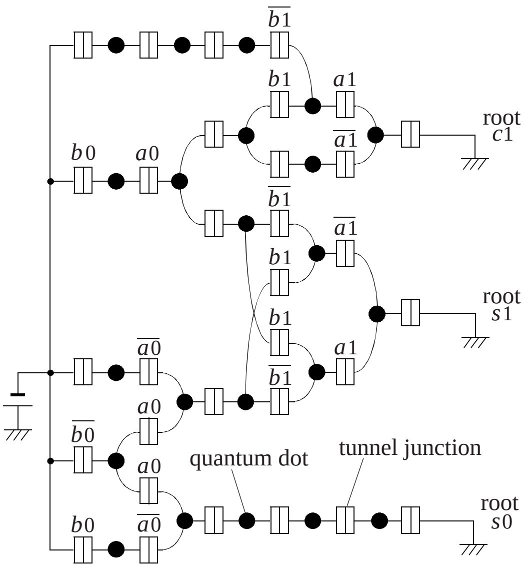


Fig. 7. Quantum-dot circuit for a 2-bit adder. The ground capacitance for each dot is omitted.

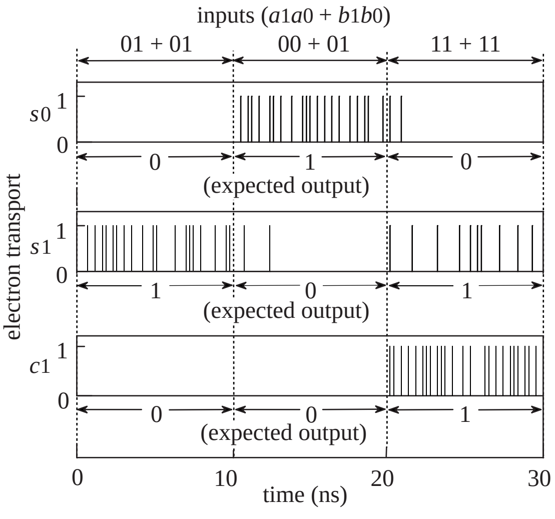
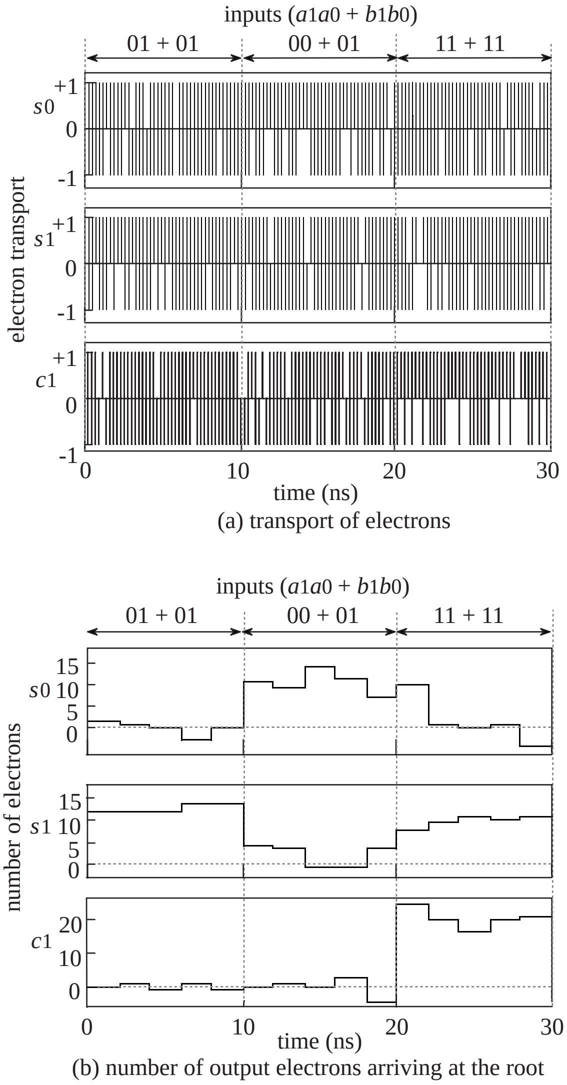


Fig. 8. Operation of the adder circuit at 0 K (simulation).



(a) transport of electrons

(b) number of output electrons arriving at the root

Fig. 9. Operation of the adder circuit at 20 K (simulation).

circuit performs add operations correctly for all possible input combinations. Some of the simulation results are shown in Figs. 8 and 9. (The same device parameters as for the preceding circuit were used. The power voltage of the 1-terminal was set to -10 mV for the results shown in Fig. 8 and to -20 mV for those in Fig. 9.) Figure 8 shows the results for absolute zero (0 K). Each impulse in the figure signifies the arrival of an electron at a root. The circuit produces the expected correct output.

Figure 9(a) shows the results for 20 K. The circuit produces noisy output due to thermal agitation; i.e., a countercurrent or countertransport of electrons from a root into the circuit is frequently observed, shown by negative impulses in the figure. Nevertheless, the correct output can be retrieved by counting the net number of output electrons. The results are shown in Fig. 9(b). In this example, the counting was repeated every two nanoseconds; *e.g.*, a total of nine electrons flowed out of root s_0 during the period from 12 ns to 14 ns. By using a longer period for the counting, we can operate the circuit at higher temperatures.

7. Summary

We proposed a method for constructing quantum-dot logic circuits that can be used to develop ultralow-power LSIs. These dot circuits consist of dot arrays fabricated using a WPG, and they perform logic operations on the basis of the shared binary-decision diagram. We designed sample dot circuits such as elemental logic gates and adder subsystems and confirmed their successful operation by computer simulation. We will be able to construct large logic systems using quantum-dot circuits.

1) S. Kasai, K. Jinushi, H. Tomozawa and H. Hasegawa: Jpn. J. Appl. Phys. **36** (1997) 1678.
2) Y. Satoh, S. Kasai, K. Jinushi and H. Hasegawa: Jpn. J. Appl. Phys. **37** (1998) 1584.
3) S. B. Akers: IEEE Trans. Comput. **27** (1978) 509.
4) N. Kuwamura, K. Taniguchi and C. Hamakawa: IEICE Trans. Electron. C-II **77** (1994) 221.

Appendix A

Any combinational logic can be implemented by a single quantum-dot circuit. However, it is desirable for constructing practical systems that a number of quantum-dot circuits can be combined into a cascade to build a large, complex system. To realize this cascadablity, we are developing a connection circuit (buffer) consisting of subquarter-micron MOSFETs, which accepts the logic output of a preceding quantum-dot circuit and operates the node devices of following circuits.

Appendix B

It is possible to achieve low-power pass-transistor circuits by using MOSFETs with high “on” resistance or by inserting high resistances into the electron paths. However, neither method is practical for the following reasons.

(1) It is difficult to set “on” resistance of MOSFETs to a large value with high accuracy.
(2) A MOSFET with high “on” resistance cannot work well as a switch because “off” resistance of the MOSFET is finite.
(3) A high resistance occupies a large area, so it cannot be used in LSIs.