J. of Mult.-Valued Logic & Soft Computing, Vol. 20, pp. 595–623
Reprints available directly from the publisher
Photocopying permitted by license only

Flexible Ferroelectric-Capacitor Element for Low Power and Compact Logic-in-Memory Architectures

SHOTA ISHIHARA, NORIAKI IDOBATA, MASANORI HARIYAMA AND MICHITAKA KAMEYAMA

Graduate School of Information Sciences, Tohoku University,
Aramaki-aza-Aoba 6-6-05, Aoba, Sendai, Miyagi, 980-8579, Japan
E-mail: ishihara@ecei.tohoku.ac.jp, hariyama@ecei.tohoku.ac.jp,
kameyama@ecei.tohoku.ac.jp

Received: November 9, 2011. Accepted: March 8, 2012.

The “Von Neumann bottleneck” and large standby power become serious problems in recent deep-sub-micron technology. To solve these problems, this paper presents ferroelectric-based logic circuits called Flexible Ferroelectric-Capacitor (FFC) elements for logic-in-memory architectures. In an FFC element, storage and a logic function are integrated on non-volatile ferro-electric-capacitors to achieve low power and area-efficiency. Moreover, the FFC elements are designed to flexibly change the access transistor network to achieve high functionality and programmability to change the function. In this paper, FFC elements for binary logic and for multiple-valued logic are proposed. The FFC elements are evaluated using HSPICE simulations and compared to the equivalent CMOS circuits. Both of the FFC elements for binary logic and for multiple-valued logic consume no power in the standby state, and reduce the transistor count and the dynamic energy consumption by respectively more than 94% and more than 65%.

Keywords: Non-volatile storage, non-volatile logic, multiple-valued logic, non-destructive operation, capacity-based logic, programmable logic, power gating, content-addressable memory (CAM), FeRAM.

This paper is an extension of conference paper [2].

1 INTRODUCTION

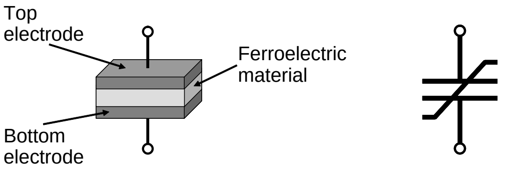
As technology scaling proceeds and the scales of VLSI systems become large, the “Von Neumann bottleneck” and large standby power become serious problems. The “Von Neumann bottleneck” is the communication bottleneck between memories and logic modules, which limits the throughput. Logic-in-memory architecture is proposed to solve the “Von Neumann bottleneck”. In logic-in-memory architectures, storage functions are distributed over a logic-circuit plane, and highly effective use of internal memory bandwidth is achieved. However, conventional logic-in-memory VLSIs generally become complicated because of the hardware overhead involved in distributing storage elements [5]. Another problem is the standby power caused by the leakage current. To reduce the standby power, low-power embedded applications tend to require frequent power ON and OFF cycles. However, SRAM-based VLSIs lose their stored data when the power is OFF. In order to retain the stored data, the data is rolled out to an external non-volatile memory such as the EEPROM or FLASH memory before power down, and reloaded the data into the VLSI after power up. This approach creates performance and power overhead. In VLSI, it is difficult to frequently power ON and OFF to reduce the standby power. To solve this problem, the use of on-chip non-volatile memory is necessary [1–4].

A ferroelectric-based logic circuit called the Complementary Ferroelectric-Capacitor (CFC) element has been proposed [6]. It is designed for logic-in-memory architectures to solve the “Von Neumann bottleneck” and reduce the standby power. The CFC element is based on binary logic, and in the CFC element, non-volatile storage and a logic function are integrated on Ferroelectric-Capacitors (FCs) by the capacitive coupling effect under the control of the external input and stored bit. The disadvantage of the CFC element is that a single CFC element only can execute simple functions such as two-input-AND or two-input-OR. In order to execute more complex functions, multiple CFC elements are combined. For example, an XOR function is implemented by two CFC elements for Content Addressable Memory (CAM), which is a typical logic-in-memory architecture. Such complex functions increase the number of transistors and FCs, and results in a large area and a high dynamic power consumption.

To solve these problems, this paper proposes low-power and high-functional ferroelectric-based logic circuits called Flexible Ferroelectric-Capacitor (FFC) elements for logic-in-memory architectures. Like the CFC element, non-volatile storage and logic functions are also integrated on FCs. The difference between the CFC element and the FFC element is that the FFC element can flexibly change the access transistor network to achieve high functionality with a small number of transistors and FCs. In this paper, two types of FFC elements are proposed, and both of them consume no power in the standby state. One is the FFC element for binary logic. A single FFC element for binary logic can perform as a binary non-volatile memory cell, and can execute five kinds of two-input binary logic functions, two kinds of three-input binary logic functions and one kind of binary arithmetic function. Compared to the equivalent CMOS circuit, the transistor count and the dynamic energy consumption are reduced by 95% and 66%, respectively. Compared to the equivalent CFC-based circuit, the transistor count, the FC count and the dynamic energy consumption are reduced by 43%, 50% and 27%, respectively. The other is the FFC element for multiple-valued logic. It introduces the multiple-valued storage and logic techniques into the FFC element for binary logic, and has a higher functionality with the same structure. A single FFC element for multiple-valued logic can perform as a three-valued non-volatile memory cell, and can execute six kinds of two-input multiple-valued logic functions, two kinds of three-input multiple-valued logic functions and one kind of binary arithmetic function. Compared to the equivalent CMOS circuit, the transistor count and the dynamic energy consumption are reduced by 96% and 65%, respectively. Compared to the equivalent CFC-based circuit, the transistor count, the FC count and the dynamic energy consumption are reduced by 60%, 67% and 45%, respectively.

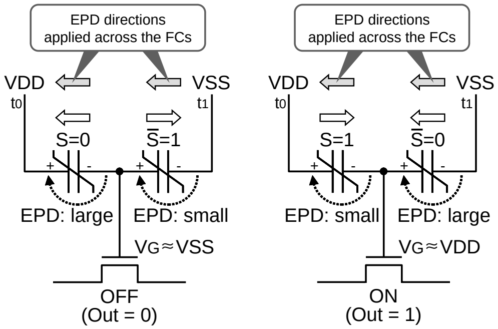
2 PRINCIPLE OF FERROELECTRIC CAPACITORS

An FC is obtained from a regular capacitor by replacing the dielectric with a ferroelectric material, as shown in Figure 1(a). Figure 1(b) is the symbol of an FC. An FC has two directions of the remnant-polarization, and is used as a variable capacitor. The capacitance of the FC is determined by the direction of the remnant-polarization and the direction of Electric Potential Difference (EPD) applied across the FC. An FC has a feature that its capacitance is large when the direction of the remnant-polarization and that of the



(a) Structure. (b) Symbol.

FIGURE 1
Ferroelectric capacitor.



(a) Stored value: “0”. (b) Stored value: “1”.

FIGURE 2
FC-based memory cell.

EPD applied across the FC are the same. On the contrary, the capacitance of the FC is small when the direction of the remnant-polarization and that of the EPD applied across the FC are opposite. The direction of the remnant-polarization is the stored data of the FC, and the direction of the EPD applied across the FC is an external input. An FC has a coercive voltage. If the direction of the EPD applied across the FC is opposite to that of the remnant-polarization and the amount of the EPD is larger than the coercive voltage, the remnant-polarization of the FC changes to the opposite direction. This is called destructive operation. Otherwise, the remnant-polarization of the FC does not change. This is called non-destructive operation.

In order to explain the behavior of the FC, consider the FC-based memory cell shown in Figure 2. The memory cell executes a non-destructive operation [6]. The memory cell has two FCs, and the remnant-polarization directions of the FCs are set to be complementary. The data representation of the stored data of the left FC is S, and that of the right FC is $\overline{S}$. The gate voltage of the pass transistor V_G is generated by the capacitive coupling effect, and V_G determines the state of the pass transistor. The states “OFF” and “ON” of the pass transistor correspond to the stored data “0” and “1”, respectively.

Figure 2(a) shows the memory cell which stores value “0”. Values $S0$ and $S1$ in the figure denote the remnant-polarization direction of the left FC and that of the right FC, respectively. The pair of the remnant-polarization directions is the stored data. To store value “0”, the remnant-polarization of the left FC and that of the right FC are respectively set to left and right in advance. To read the stored data, VDD and VSS are applied to terminals t_0 and t_1, respectively. The gate voltage of the pass transistor V_G is generated by the capacitive coupling effect of the two FCs, and V_G determines the state (ON/OFF) of the pass transistor. Since the relationship among the terminal voltages is $V_{t0} >= V_G >= V_{t1}$, the direction of the EPD applied across each FC is always left. In the left FC, the directions of the remnant-polarization direction and the EPD applied across the FC are the same. Therefore, the capacitance of the FC is small. In the right FC, the directions of the remnant-polarization direction and the EPD applied across the FC are opposite. Therefore, the capacitance of the FC is large. Since the gate voltage of the pass transistor V_G is generated by the capacitive coupling effect, the EPD between the electrodes of the left FC is large and that of the right FC is small. Therefore, voltage V_G is approximately the same as voltage *VSS*. As a result, the gate voltage of the pass transistor is lower than the threshold voltage, and the pass transistor is OFF. The output *Out* is “0”, and is the same as the stored bit *S*. In this way, the operation is non-destructive. The reason is as follows. In the left FC, the direction of the EPD applied across the FC is the same as that of the remnant-polarization. Therefore, the remnant-polarization of the left FC does not change. In the right FC, although the direction of the EPD applied across the FC is opposite to that of the remnant-polarization, the amount of the EPD is small and is not larger than the coercive voltage. Therefore, the remnant-polarization of the right FC does not change.

Figure 2(b) shows the memory cell which stores value “1”. In the left FC, the directions of the remnant-polarization direction and the EPD applied across the FC are opposite. Therefore, the capacitance of the FC is large. In the right FC, the directions of the remnant-polarization direction and the EPD applied across the FC are the same. Therefore, the capacitance of the FC is small. Since the gate voltage of the pass transistor V_G is generated by the capacitive coupling effect, the EPD between the electrodes of the left FC is small and that of the right FC is large. Therefore, voltage V_G is approximately the same as voltage *VDD*. As a result, the gate voltage of the pass transistor is higher than the threshold voltage, and the pass transistor is ON. The output *Out* is “1”, and is the same as the stored bit *S*. Similarly to the memory cell storing value “0”, the memory cell storing value “1” also executes a non-destructive operation.

3 FFC ELEMENT

3.1 Architecture

Figure 3 shows the function of a general logic-in-memory circuit such as an FFC element. It executes a logic function between the external input **In** and the stored input **S**. The obtained result is the output **Out**. Figures 4 and 5 shows the examples of the logic-in-memory circuits composed of FFC elements. The function of a single FFC element is shown in Figure 6. An FFC element performs as a logic element, a storage element and a pass switch.

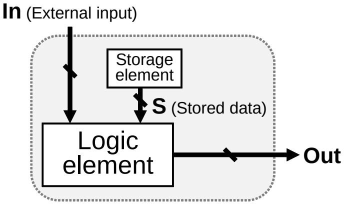


FIGURE 3
Function of a logic-in-memory circuit.

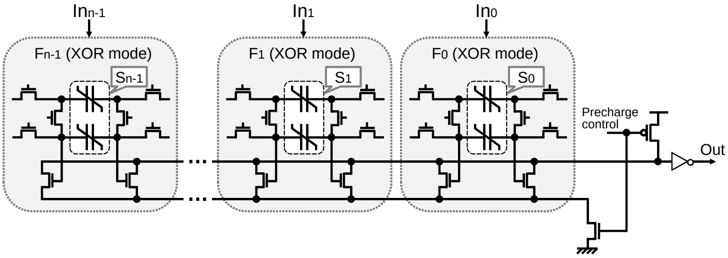


FIGURE 4
CAM word circuit composed of FFC elements.

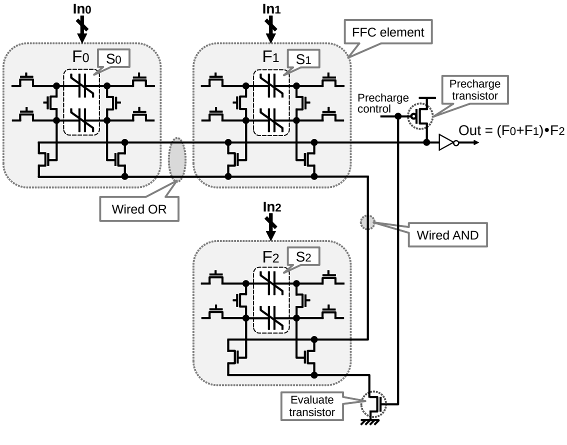


FIGURE 5
General structure of logic-in-memory circuit composed of FFC elements.

The logic element executes a switching function between the external input **In** and the internal input *S* which is stored in the storage element. If the result of the logic element is “1”, the pass switch turns ON. Otherwise, the pass switch turns OFF. Note that, the FFC element can execute various kinds of

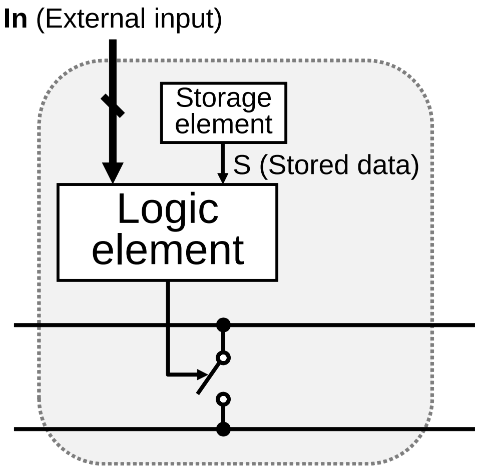


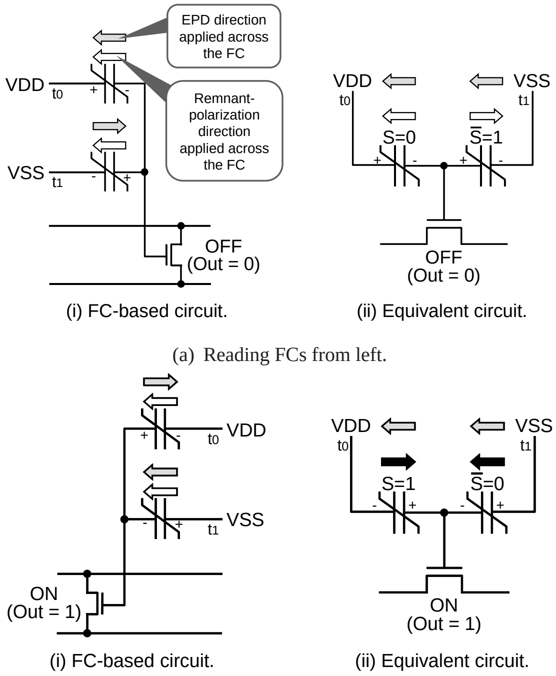
FIGURE 6
Function of an FFC element.

functions, and the executed function can be changed dynamically. The logical AND and OR operations between the results of the FFC elements can be implemented by the connections of the FFC elements. Additional precharge and evaluate transistors are used to control precharge and evaluate phases based on dynamic-logic style.

3.2 FFC Element for Binary Logic

An FC-based memory can also be implemented as Figure 7(a)(i), and Figure 7(a)(ii) is its equivalent circuit. Note that the white arrows and the black arrows on the FCs represent $S = 0$ and $S = 1$, respectively. The stored bit of the pair of FCs is $S = 0$, and it specifies that the remnant-polarization direction of each FC is left. In this case, the output *Out* is “0”, and is the same as the stored bit $S = 0$. Figure 7(b)(i) shows the case where the FCs are read from right, and Figure 7(b)(ii) is its equivalent circuit. In this case, the output *Out* is “1”, and is different from the stored bit $S = 0$. The stored values of Figures 7(a)(i) and 7(b)(i) are the same, but different outputs are obtained. This is because the different reading paths cause the different directions of the voltages applied to the FCs. Since the output *Out* depends on both the direction of the remnant-polarization of the FCs and the direction of the voltage applied across the FCs, *Out* depends not only on the stored data but also on the reading path. The proposed FFC exploits this feature to execute complex functions with a small number of FCs.

Figure 8 shows the structure of an FFC element. Like the CFC element, the two ferroelectric capacitors store a pair of complementary data and the FFC



(a) Reading FCs from left.

(b) Reading FCs from right.

FIGURE 7
Relationship between the reading path and the output *Out*.

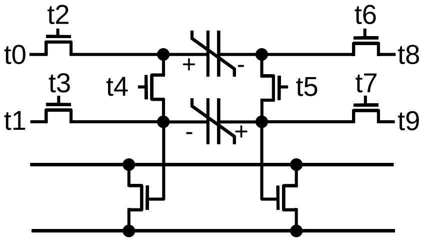


FIGURE 8
Structure of the FFC element.

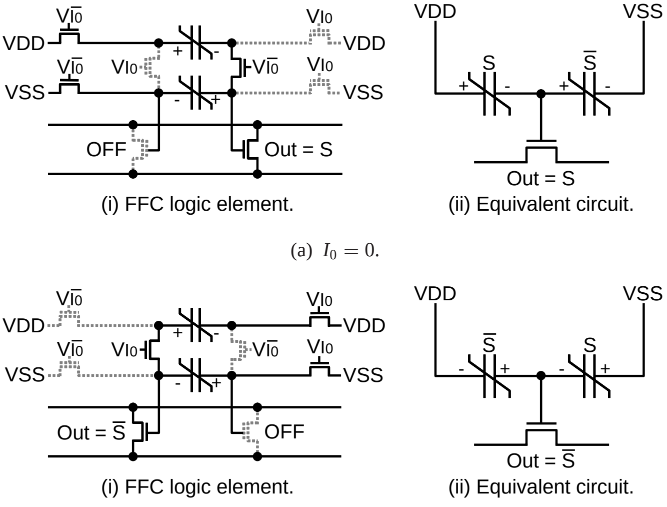
element also executes a non-destructive operation. The difference between CFC and FFC elements is that the FFC element can change the access transistor network flexibility to achieve a high flexibility using a small number of transistors and FCs.

The modes of the FFC element for binary logic are as follows, where I_0 and I_1 are binary external inputs and where S is a binary stored bit. Modes

BI_MEM, AND, OR and COMP are the same functions as those of the CFC element and the other modes are newly added functions.

BI_MEM:	Binary non-volatile memory cell for storing S.
BI_XOR:	$I_0 \oplus S$. Binary CAM cell.
BI_AND:	$I_0 \cdot S$.
BI_AND-INV:	$I_0 \cdot \overline{S}$.
BI_OR:	$I_0 + S$.
BI_OR-INV:	$I_0 + \overline{S}$.
BI_MUX-AND:	$\begin{cases} I_0 \cdot S & (if\ I_1 = 0), \\ I_0 \cdot \overline{S} & (if\ I_1 = 1). \end{cases}$
BI_MUX-OR:	$\begin{cases} I_0 + S & (if\ I_1 = 0), \\ I_0 + \overline{S} & (if\ I_1 = 1). \end{cases}$
BI_COMP:	Bit-serial comparator. Carry logic.

Figures 9(a) and (b) show the BI_XOR mode of the FFC element. In this mode, the FFC element executes $I_0 \oplus S$, where I_0 is an external input and S is the stored bit. The idea behind the BI_XOR mode of the FFC element is to read the FCs through different paths depending on the external input I_0. As mentioned above, different reading paths cause the different direction



(i) FFC logic element. (ii) Equivalent circuit.

(a) $I_0 = 0$.

(i) FFC logic element. (ii) Equivalent circuit.

(b) $I_0 = 1$.

FIGURE 9
BI_XOR mode of the FFC element.

I_0	S	*Out*
0	0	0(=S)
0	1	1(=S)
1	0	1(=$\overline{S}$)
1	1	0(=$\overline{S}$)

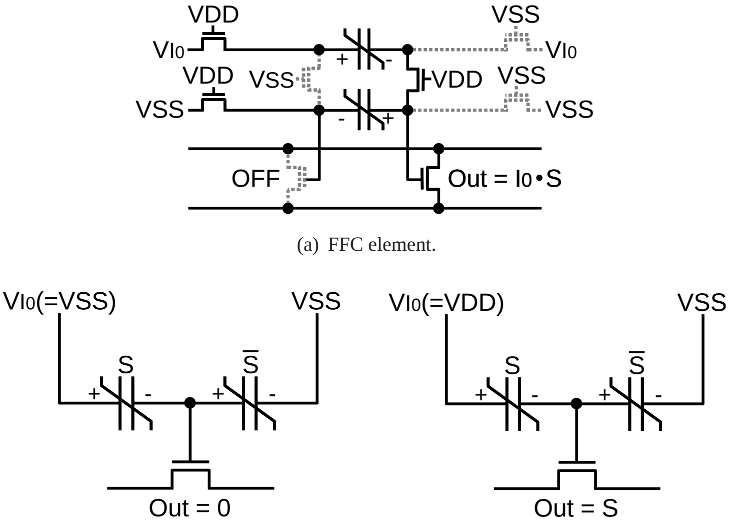
TABLE 1
Truth table of BI_XOR mode of the FFC element

of the voltage applied to the FCs. Since the output *Out* depends on both the direction of the voltage applied across the FCs and the direction of the remnant-polarization of the FCs, *Out* depends on both the reading path and the stored data. The different reading paths allow us to execute more complex functions with a smaller number of FCs. The external input I_0 controls the state (ON/OFF) of transistors in the reading paths. In other words, the turn-ON transistors in Figure 9(a)(i) are different from those in Figure 9(b)(ii). V_{I_0} is the voltage of logical value I_0. When I_0 is "0", V_{I_0} is *VSS*. When I_0 is "1", V_{I_0} is *VDD*. As shown in Figure 9(a), when $I_0 = 0$, the output *Out* is the same as S. As shown in Figure 9(b), when $I_0 = 1$, the output *Out* is the same as $\overline{S}$. As a result, as shown in Table 1, the BI_XOR mode of the FFC element executes $A \oplus S$. The BI_XOR mode of the FFC element also performs as a binary CAM cell. In a binary CAM cell storing either "0" or "1", when the value of the input and that of the store value are the same, the output is "0"; when the value of the input and that of the store value are different, the output is "1". The truth tables of the CAM cell and that of the BI_XOR mode are the same.

Figure 10 shows the BI_AND mode of the FFC element. In this mode, the FFC element executes $I_0 \cdot S$. Figure 10(b) shows the equivalent circuit when $I_0 = 0$. In this case, voltage V_{I_0} is *VSS*. Therefore, the gate voltage of the pass transistor is *VSS*, and then the pass transistor is OFF. As a result, the output *Out* is "0" regardless of the stored value. Figure 10(c) shows the equivalent circuit when $I_0 = 1$. In this case, voltage V_{I_0} is *VDD*, and then the output *Out* is the same as the stored bit S. Thus, only if I_0 and S are "1" is the output *Out* "1". In other words, the FFC element executes $I_0 \cdot S$.

Figure 11 shows the BI_AND-INV mode of the FFC element. In this mode, the FFC element executes $I_0 \cdot \overline{S}$. The difference between the BI_AND-INV and in BI_AND modes of the FFC element is that they use different reading paths.

Figure 12 shows the BI_OR mode of the FFC element. In this mode, the FFC element executes $I_0 + S$. Figure 12(b) shows the equivalent circuit when $I_0 = 0$. In this case, voltage V_{I_0} is *VSS*, and then the output *Out* is the same as the stored bit S. Figure 12(c) shows the equivalent circuit when $I_0 = 1$. In this

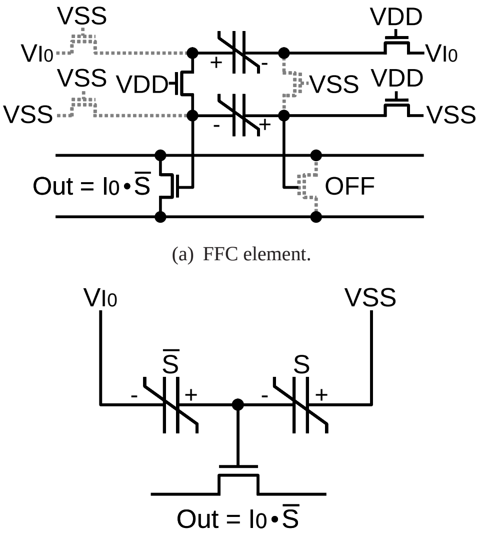


(a) FFC element.

(b) Equivalent circuit when $I_0 = 0$.

(c) Equivalent circuit when $I_0 = 1$.

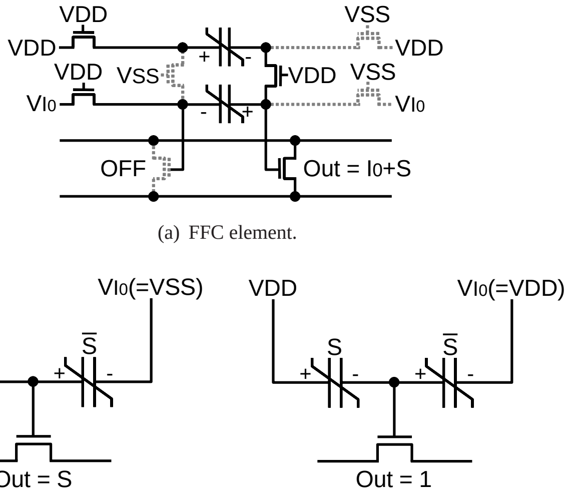
FIGURE 10
BI_AND mode of the FFC element.



(a) FFC element.

(b) Equivalent circuit.

FIGURE 11
BI_AND-INV mode of the FFC element.



(a) FFC element.

(b) Equivalent circuit when $I_0 = 0$.

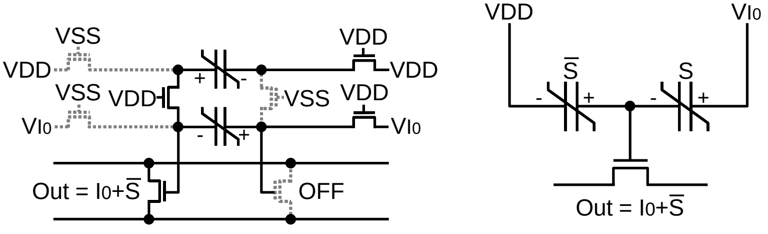
(c) Equivalent circuit when $I_0 = 1$.

FIGURE 12
BI_OR mode of the FFC element.

case, voltage V_{I_0} is *VDD*. Therefore, the gate voltage of the pass transistor is *VDD*, and then the pass transistor is OFF. As a result, the output *Out* is "0" regardless of the stored value. If either I_0 or $\overline{S}$ is "1", the output *Out* is "1". In other words, the FFC element executes $I_0 + S$.

Figure 13 shows the BI_OR-INV mode of the FFC element. In this mode, the FFC element executes $I_0 \cdot \overline{S}$. The difference between the BI_OR-INV and in BI_OR modes of the FFC element is that they use different reading paths.

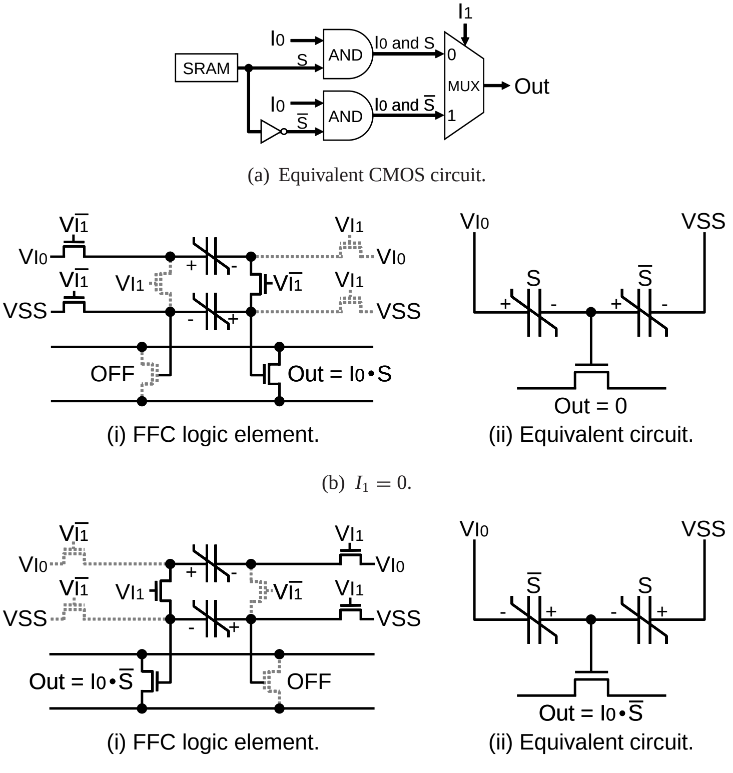
Figure 14 shows the BI_MUX-AND mode of the FFC element. In this mode, the FFC element executes $(\overline{I_1} \cdot (I_0 \cdot S)) + (I_1 \cdot (I_0 \cdot \overline{S}))$. Figure 14(a)



(a) FFC element.

(b) Equivalent circuit.

FIGURE 13
BI_OR-INV mode of the FFC element.



(a) Equivalent CMOS circuit.

(b) $I_1 = 0$.

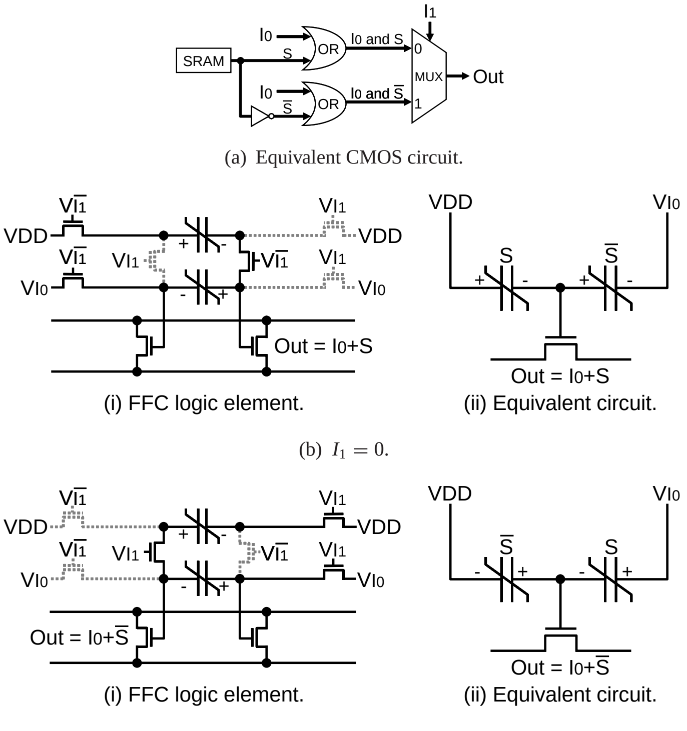
(c) $I_1 = 1$.

FIGURE 14
BI_MUX-AND mode of the FFC element.

shows the equivalent CMOS circuit. The external input I_1 selects the reading path. As shown in Figure 14(b), when $I_1 = 0$, the FFC elements is in the BI_AND mode which executes $I_0 \cdot S$. As shown in Figure 14(c), when $I_1 = 1$, the FFC elements is in the BI_AND-INV mode which executes $I_0 \cdot \overline{S}$.

Figure 15 shows the BI_MUX-OR mode of the FFC element. In this mode, the FFC element executes $(\overline{I_1} \cdot (I_0 + S)) + (I_1 \cdot (I_0 + \overline{S}))$. Figure 15(a) shows the equivalent CMOS circuit. The external input I_1 selects the reading path. As shown in Figure 15(b), when $I_1 = 0$, the FFC elements is in the BI_OR mode which executes $I_0 + S$. As shown in Figure 15(c), when $I_1 = 1$, the FFC elements is in the BI_OR-INV mode which executes $I_0 + \overline{S}$.

Figure 16 shows the BI_COMP mode of the FFC element. In this mode, the FFC element compares two k-bit binary numbers: $\mathbf{I_0} = (\mathbf{I_0^{k-1}}, \mathbf{I_0^{k-2}}, \cdots, \mathbf{I_0^0})$ and $\mathbf{I_1} = (\mathbf{I_1^{k-1}}, \mathbf{I_1^{k-2}}, \cdots, \mathbf{I_1^0})$ in a bit-serial manner, and

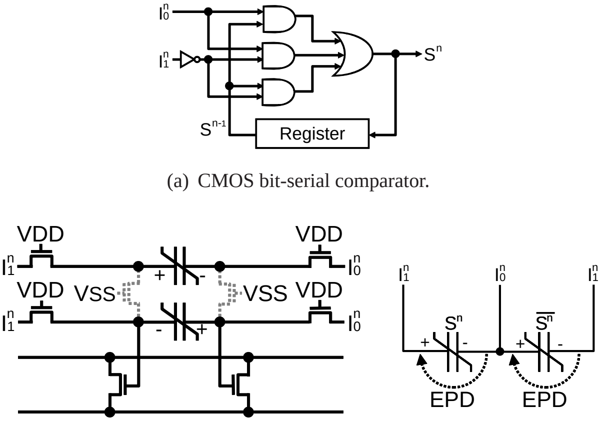


(a) Equivalent CMOS circuit.

(b) $I_1 = 0$.

(c) $I_1 = 1$.

FIGURE 15
BI_MUX-OR mode of the FFC element.



(a) CMOS bit-serial comparator.

(b) FFC element.

(c) Equivalent circuit.

FIGURE 16
BI_COMP mode of the FFC element.

I_0^n	I_1^n	S^n
0	0	S^{n-1}(unchange)
0	1	0
1	0	1
1	1	S^{n-1}(unchange)

TABLE 2
Truth table of a bit-serial comparator

generates the output $\mathbf{S} = (\mathbf{S^{k-1}}, \mathbf{S^{k-2}}, \cdots, \mathbf{S^0})$, where I_0^n, I_1^n and S^n represents the $(n+1)$th bits of $\mathbf{I_0}$, $\mathbf{I_1}$ and $\mathbf{S}$, respectively. Table 2 shows the truth table of a bit-serial comparator. Note that the register in the equivalent CMOS circuit is removed in the FFC element by exploiting the following write scheme of the FCs. When the EPD applied across an FC is $+VDD$, the direction of the remnant-polarization of the FC is set to left. When the EPD applied across an FC is $-VDD$, the direction of the remnant-polarization of the FC is set to right. When the EPD applied across an FC is 0, the direction of the remnant-polarization of the FC is unchanged. By exploiting this feature of an FC, the FFC in BI_COMP mode performs as a bit-serial comparator as follows. When $I_0^n < I_1^n$, the stored bit S^n is set to "0". When $I_0^n > I_1^n$, S^n is set to "1". When $I_0^n = I_1^n$, S^n is unchanged. As a result, the function of the BI_COMP mode of the FFC element is the same as that of the bit-serial comparator.

In addition to the bit-serial comparator, the BI_COMP mode of the FFC element also performs as a carry logic of a bit-serial adder since the truth table of the bit-serial comparator and that of the carry logic are similar. As shown in Figure 17, the function of the bit-serial adder is $\boldsymbol{I_0} + \boldsymbol{I_1}$ in a bit-serial manner. Table 2(a) shows the truth table of the carry logic, and the truth table can be transform into Table 2(b) which is similar to the truth table of

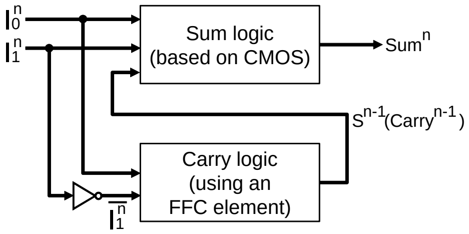


FIGURE 17
Bit-serial adder.

(a) Relationship between I_0^n, I_1^n and S^n.

I_0^n	I_1^n	S^n (Carry^n)
0	0	0
0	1	S^{n-1}(unchange)
1	0	S^{n-1}(unchange)
1	1	1

(b) Relationship between I_0^n, $\overline{I_1^n}$ and S^n.

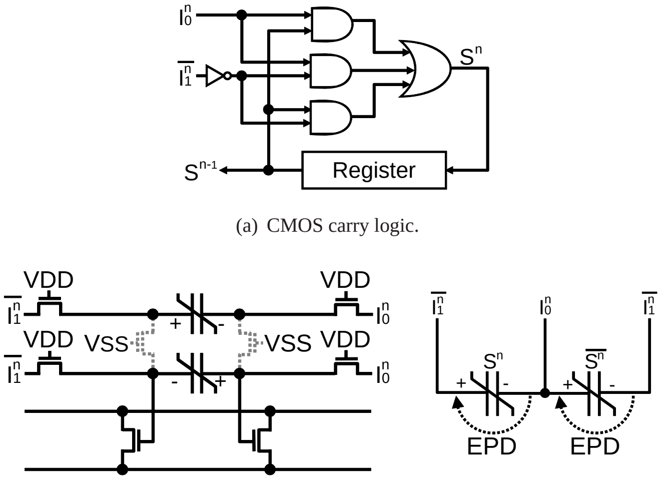
I_0^n	$\overline{I_1^n}$	S^n (Carry^n)
0	0	S^{n-1}(unchange)
0	1	0
1	0	1
1	1	S^{n-1}(unchange)

TABLE 3
Truth table of the carry logic

the BL_COMP mode of the FFC element. Exploiting this similarity, the carry logic can be implemented using BI_COMP mode of the FFC element. Note that S^{n-1} is generated by reading the store data of the FCs. Figure 18 shows the carry logic implemented using BI_COMP mode of the FFC element.

3.3 FFC element for Multiple-Valued Logic

The circuit of the FFC element for multiple-valued logic is the same as that for binary logic. The feature of the multiple-valued FFC element is that an FFC element stores a three-valued data. Figure 19 shows the fundamental principle of a multiple-valued FFC element. Values $S1$ and $S2$ denote the polarization directions of the left FC and the right FC, respectively. Depend-

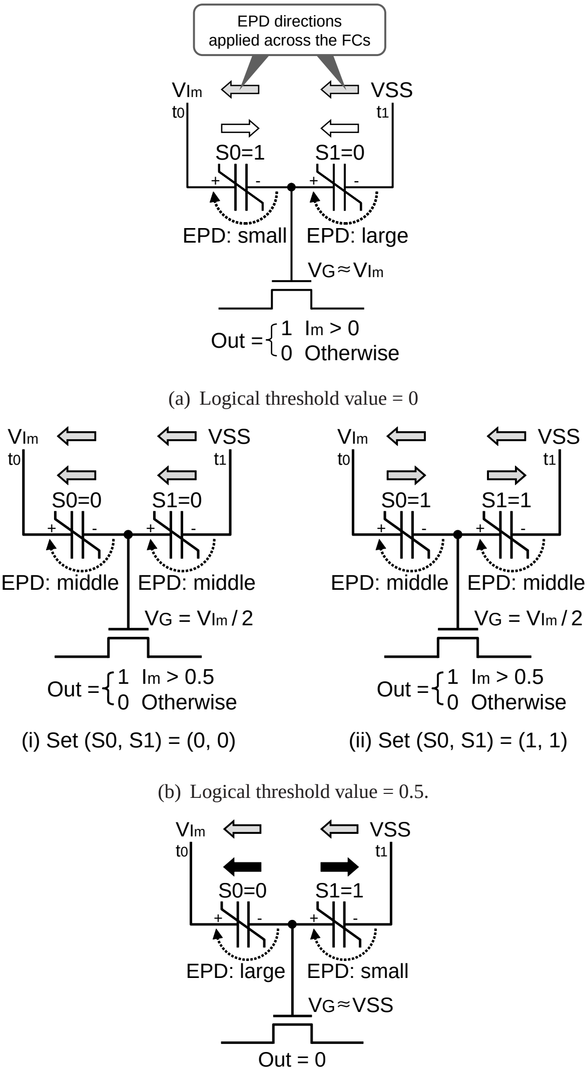


(a) CMOS carry logic.

(b) FFC element.

(c) Equivalent circuit.

FIGURE 18
Carry logic implemented using BI_COMP mode of the FFC element.



(a) Logical threshold value = 0

(b) Logical threshold value = 0.5.

(c) Logical threshold value = 1.

FIGURE 19
Fundamental principle of a multiple-valued FFC element.

ing on $S1$ and $S2$, there exists three threshold functions whose threshold voltages are different from each other. Note that the same value is stored in Figure 19(b) by two different sets $(S1, S2) = (0, 0)$ and $(1, 1)$. The logical value of the stored bit is $S \in 0, 0.5, 1$; the logical value of the external

Logical threshold value	I_m	***Out***
0	0	0
0	0.5	1
0	1	1
0.5	0	0
0.5	0.5	0
0.5	1	1
1	0	0
1	0.5	0
1	1	0

TABLE 4
Truth table of the MUL_THR mode of the FFC element

input is $I_m \in 0, 0.5, 1$; the logical value of the output is $Out \in 0, 1$. The circuit executes the following threshold function, and Table 4 shows the truth table.

$$Out = \begin{cases} 1 & I_m > S \\ 0 & otherwise. \end{cases} \tag{1}$$

To execute the threshold function, voltage V_{I_m} is applied to terminal t_0, and voltage *VSS* is applied to terminal t_1. Voltage V_{I_m} is the voltage of I_m. When the logical value of a signal is “0”, the voltage of the signal is *VSS*. When the logical value of a signal is “0.5”, the voltage of the signal is *VDDL* which is lower than *VDD*. When the logical value of a signal is “1”, the voltage of the signal is *VDD*. Figures 19(a), (b) and (c) are arranged in an increasing order of the logical threshold value. Note that the white arrows, the gray arrows and the black arrows on the FCs represent $S = 0$, $S = 0.5$ and $S = 1$, respectively.

Figure 19(a) shows the FC-based circuit in which the threshold voltage is smallest, and the logical value of the stored bit S is “0”. In Figure 19(a), the direction of the remnant-polarization of the left FC is right, and that of the right FC is left. Since the direction of the EPD applied across each FC is left, the capacitance of the left FC is much larger than that of the right FC. Since the gate voltage of the pass transistor V_G is generated by the capacitive coupling effect, the EPD applied across the left FC is much smaller than that of the right FC. Therefore, voltage V_G is approximately the same as voltage V_{I_m}. As a result, if $I_m > 0$, the pass transistor is ON and then *Out* is “1”. Otherwise, the pass transistor is OFF and then *Out* is “0”.

Figure 19(b) shows the FC-based circuits in which the threshold voltages are larger than Figure 19(a), and the logical values of the stored bit S are “0.5”. In the following, the set $(S0, S1) = (0, 0)$ is used for the threshold function since it executes a non-destructive operation, and the set $(S0, S1) = (1, 1)$ does not. In Figure 19(b)(i), the direction of the remnant-polarization of the left FC and that of the right FC are the same. Since the direction of the EPD applied across each FC is left, the capacitance of the left FC and that of the right FC are the same. Since the gate voltage of the pass transistor V_G is generated by the capacitive coupling effect, the EPD applied across the left FC is the same as that applied across the right FC. Therefore, voltage V_G is approximately the same as voltage $V_{I_m}/2$. As a result, if $I_m > 0.5$, the pass transistor is ON and then *Out* is “1”. Otherwise, the pass transistor is OFF and then *Out* is “0”. Similarly, in Figure 19(b)(ii), voltage V_G is approximately the same as voltage $V_{I_m}/2$. As a result, if $I_m > 0.5$, the pass transistor is ON and then *Out* is “1”. Otherwise, the pass transistor is OFF and then *Out* is “0”. In a typical manner, Figure 19(b)(ii) executes a destructive operation when V_{I_m} is *VDD* for the following reason. In the left FC and the right FC, the direction of the EPD applied across the each FC is opposite to that of the remnant-polarization of the FC. When V_{I_m} is *VDD*, the amount of EPD is larger than the coercive voltage. Therefore, the remnant-polarization directions of the FCs are changed. Note that, if V_{I_m} is *VSS* or *VDDL*, the amount of EPD is smaller than the coercive voltage; therefore, 19(b)(ii) executes a non-destructive operation.

Figure 19(c) shows the FC-based circuit in which the threshold voltage is larger than Figure 19(b), and the logical value of the stored bit *S* is “1”. In Figure 19(c), the direction of the remnant-polarization of the left FC is left, and that of the right FC is right. Since the direction of the EPD applied across each FC is left, the capacitance of the left FC is much smaller than that of the right FC. Since the gate voltage of the pass transistor V_G is generated by the capacitive coupling effect, the EPD applied across the left FC is much larger than that of the right FC. Therefore, voltage V_G is approximately the same as voltage *VSS*. As a result, *Out* is always “0” regardless of the value of I_m.

The modes of the FFC element for multiple-valued logic are as follows, where I_b is an binary external input, I_m is an three-valued external input and S_m is a three-valued stored bit.

MUL_MEM: Three-valued non-volatile memory for storing *S*.

MUL_THR: Three-valued threshold logic.

MUL_XOR: Extension of the BI_AND mode with addition of a stored value “don’t care”.
Ternary CAM cell.

MUL_AND: Extension of the BI_AND mode with addition of a stored value “don’t care”.

MUL_AND-INV: Extension of the BI_AND-INV mode with addition of a stored value “don’t care”.

MUL_OR: Extension of the BI_OR mode with addition of a stored value “don’t care”.

MUL_OR-INV: Extension of the BI_OR-INV mode with addition of a stored value “don’t care”.

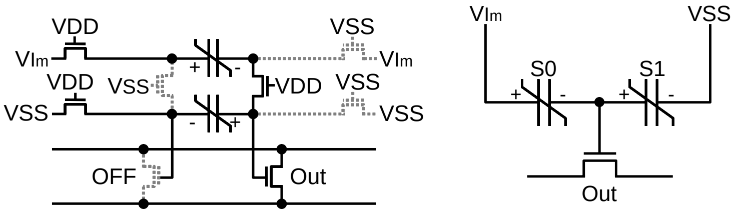
MUL_MUX-AND: Extension of the BI_MUL-AND mode with addition of a stored value “don’t care”.

MUL_MUX-OR: Extension of the BI_MUX-OR mode with addition of a stored value “don’t care”.

MUL_COMP: Same function of the BI_COMP mode.

Figure 20 shows the MUL_THR mode of the FFC element. In this mode, the FFC element performs as a three-valued threshold logic gate. Its equivalent circuit is the same as Figure 19, and the function is the same as Eq. 1. For the stored value “0.5”, the remnant-polarization directions of the FCs are set to be the same as that of Figure 19(b)(i). This is because Figure 19(b)(i) executes a non-destructive operation even when V_{I_m} is *VDD*.

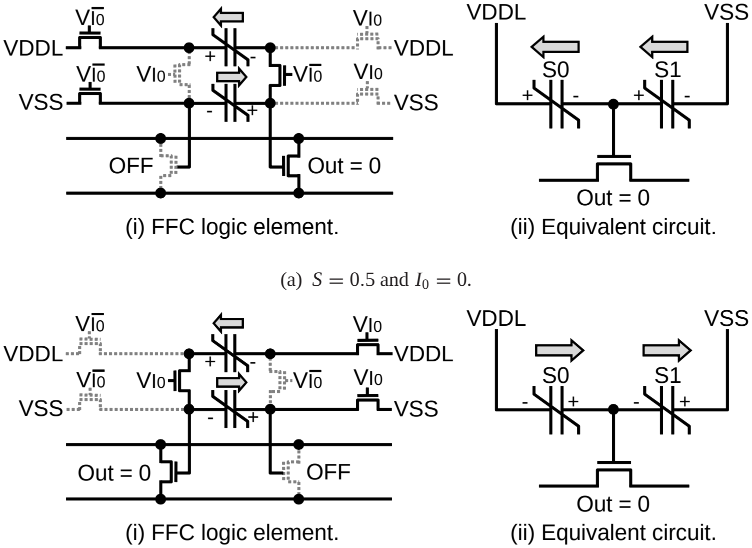
Figure 21 shows the MUL_XOR mode of the FFC element. The external input is a binary signal, and the stored bit a three-valued data. This mode is an extension of the BI_XOR mode with the addition of a stored value “0.5”. Table 5 shows the truth table of this mode. In addition to values “0” and “1”, the FCs store value “0.5”. Value “0.5” value performs as a don’t care, and value “0.5” stored in the FCs causes the output *Out* to be “0” regardless of the value of the external input I_b. When the stored value is either “0” or “1”, the behavior of the MUL_XOR mode is the same as the BI_XOR mode. Figures 21(a)(i) and 21(b)(i) show the behaviors of the MUL_XOR mode when the stored value is “0.5”. Note that the gray arrows on the FCs represent $S = 0.5$. Figure 21(a)(i) shows the behavior when I_b is “0”. The equivalent circuit of Figure 21(a)(i) is the same as Figure 19(b)(i). Since *VDDL* is corresponds to the voltage of logical value “0.5”, the value “0.5” is not larger than the stored value “0.5”. Therefore, the output *Out* is “0”. Figure 21(a)(ii) shows



(a) FFC element.

(b) Equivalent circuit.

FIGURE 20
MUL_THR mode of the FFC element.



(i) FFC logic element.

(ii) Equivalent circuit.

(a) $S = 0.5$ and $I_0 = 0$.

(i) FFC logic element.

(ii) Equivalent circuit.

(b) $S = 0.5$ and $I_0 = 1$.

FIGURE 21
MUL_XOR mode of the FFC element.

the behavior when I_b is “1”. The equivalent circuit of Figure 21(b)(i) is the same as Figure 19(b)(ii). Similarly to 21(a)(i), the output *Out* is “0”. As a result, the output *Out* to be “0” regardless of the value of the external input I_b. Note that the MUL_XOR mode executes a non-destructive operation even if the equivalent circuit of Figure 21(b)(i) is the same as Figure 19(b)(ii), since a low voltage *VDDL* is used as the apply voltage to the FCs. The MUL_XOR mode of the FFC logic also performs as a ternary CAM cell. A binary CAM cell stores either value “0” or value “1”. In addition to values “0” and “1”, a ternary CAM cell [8] stores value “0.5”. Value “0.5” is a don’t care, is used

S	$\boldsymbol{I_0}$	***Out***
0	0	0
0	1	1
1	0	1
1	1	0
0.5	0	0
0.5	1	0

TABLE 5
Truth table of the MUL_XOR mode of the FFC element

for a wild-card operation. The wild-card operation means that value “0.5” stored in a cell causes the output *Out* be “0” regardless of the value of the external input. As a result, the truth table of the ternary CAM cell is the same as that of the MUL_XOR mode.

Similarly to the MUL_XOR mode, modes MUL_AND, MUL_AND-INV, MUL_OR, MUL_OR-INV, MUL_MUX-AND and MUL_MUX-OR are respectively extensions of modes BI_AND, BI_AND-INV, BI_OR, BI_OR-INV, BI_MUX-AND and BI_MUX-OR with addition of a stored value “don’t care”.

4 SIMULATION

The proposed logic is designed using a 0.35μm-CMOS/0.6μm-ferroelectric-capacitor process. We compare an FFC element with the equivalent CMOS circuit and the equivalent CFC-based circuit by using an HSPICE simulation. The supply voltage and the temperature are set to 4.0V and 25℃, respectively.

Table 6 summarizes the comparison results among a binary FFC element, the equivalent CMOS circuit and the equivalent CFC-based circuit. The structures of the equivalent CMOS circuit and the equivalent CFC-based circuit are shown in Figures 22 and 23, respectively. The equivalent CFC-based circuit is modified from the original CFC element to achieve the same function as the binary FFC element. Compared to the equivalent CMOS circuit, the transistor count and the energy consumption are respectively reduced by 95% and 66% with almost the same delay. Because FCs are placed directly on top of the CMOS transistors, the area overhead of the FCs is very small and the area of the CFC element is approximately proportional to the transistor count. Therefore, the area is greatly reduced in the FFC element because of its small transistor count. Moreover, the FFC element greatly reduces the

	CMOS	**CFC**	**Binary FFC (Proposed)**
Transistor count	167	14	8
Ferroelectric capacitor count	0	4	2
Delay [ps]	599	545	601
Energy / operation [fJ]	1067	502	364
Standby power [fW]	92407	0	0
Volatility	Volatile	Non-volatile	Non-volatile

TABLE 6
Comparison result among a binary FFC element, the equivalent CMOS circuit and the equivalent CFC-based circuit

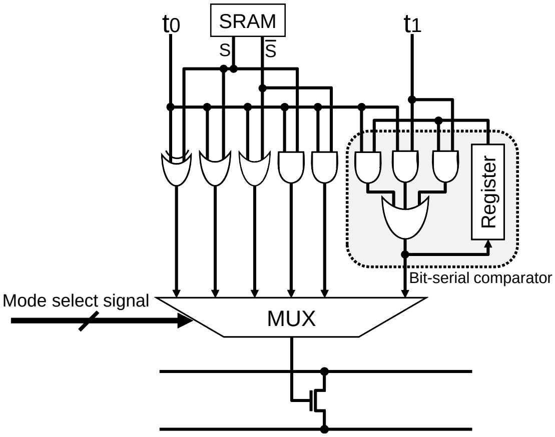


FIGURE 22
Equivalent CMOS circuit of a binary FFC element.

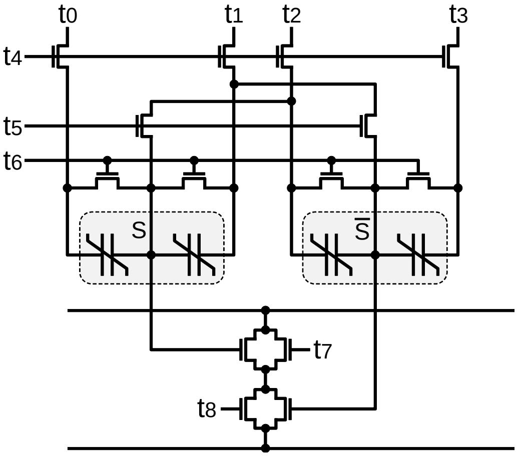


FIGURE 23
Equivalent CFC-based circuit of a binary FFC element.

standby power, because no permanent voltage supply is required for holding the stored data. Compared to the equivalent CFC-based circuit, the transistor count, the FC count and the energy consumption are respectively reduced by 43%, 50% and 27% with 11% delay overhead. The small energy consumption of the FFC element is because of its small transistor and FC counts.

Table 7 summarizes the comparison result among a multiple-valued FFC element, the equivalent CMOS-based circuit and the equivalent CFC-based

	CMOS	CFC	Multiple-valued FFC (Proposed)
Transistor count	217	20	8
Ferroelectric capacitor count	0	6	2
Delay [ps]	831	1005	1363
Energy / operation [fJ]	1330	838	463
Standby power [fW]	119142	0	0
Volatility	Volatile	Non-volatile	Non-volatile

TABLE 7
Comparison result among a multiple-valued FFC element, the equivalent CMOS circuit and the equivalent CFC-based circuit

circuit. The structures of the equivalent CMOS circuit and the equivalent CFC-based circuit are shown in Figures 24 and 25, respectively. The equivalent CFC-based circuit is based on binary logic and executes the same function as the multiple-valued FFC element. Compared to the equivalent CMOS circuit, the transistor count and the energy consumption are reduced by 96% and 65%, respectively. The delay overhead of the FFC element is 64%, and the reason is as follows. When the logic value of either the input of the FFC or the logic value of the stored data is “0.5”, even if the case that the pass transistor is set to “ON” and its gate voltage is larger than the threshold voltage, the gate voltage is much smaller than “VDD”. In this case, the pass transistor is not completely “ON”, and the delay increases. Compared to the equivalent

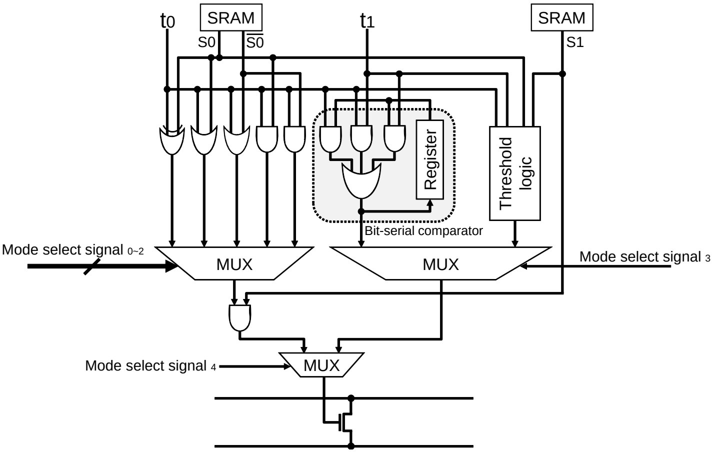


FIGURE 24
Equivalent CMOS circuit of a multiple-valued FFC element.

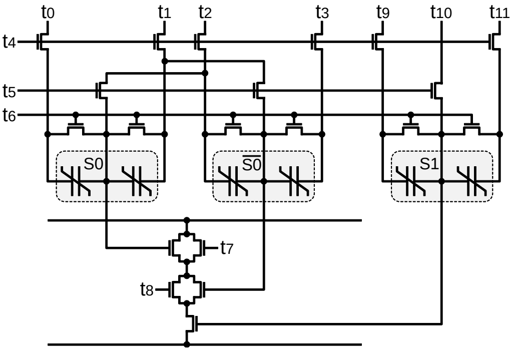


FIGURE 25
Equivalent CFC-based circuit of a multiple-valued FFC element.

CFC-based circuit, the transistor count, the FC count and the energy consumption are respectively reduced by 60%, 67% and 45% with 36% delay overhead. The small energy consumption of the FFC element is because of its small transistor and FC counts.

5 IMPLEMENTATION ISSUES

In this paper, the proposed circuit is designed using a 0.35um-CMOS/0.6um-ferroelectric-capacitor process. The supply voltage is 3.3V which is higher than the recently scaled CMOS processes. In the scaled CMOS processes, the supply voltages are difficult to make lower compared to the 90 nm process, since the threshold voltages of MOSFETs are set to high voltages for inhibiting the standby power increase. For example, the standard supply voltage in a 90nm process is 1.0V, and in a scaled process such as 65nm is increased to 1.2V. In recently FCs, the operating voltage is 1.3V [9] which is near the standard supply voltage of MOSFET in a 65nm technology node. Moreover, many studies of low voltage techniques for FCs are undergoing [9–11], and the operating voltage of FCs is expected to become lower in the future.

Typical FC-based circuits suffer from reliability issues and perform destructive operations. To solve these problems by circuit design, the CFC element has been proposed [6]. The reliable technique proposed in the CFC element can be used in the proposed FFC element. Reference [6] has proposed a restore scheme to achieve high durability for the repetitive execute cycles. The restore scheme is executed after the operation. In this scheme, the applied voltages to the FCs are inverted from that in the operation, in order to recover the remnant-polarization charge in the FCs. This scheme provides

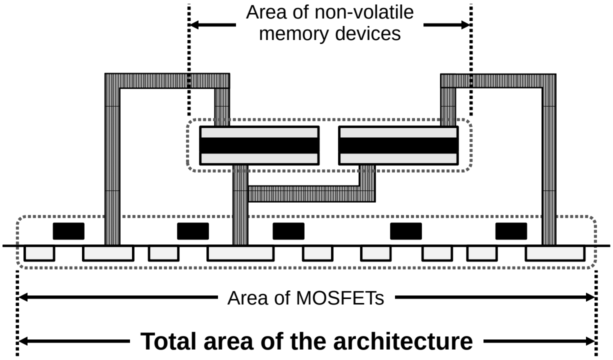

Area of non-volatile memory devices
Area of MOSFETs
Total area of the architecture

FIGURE 26
Total area of a logic-in-memory architecture.

a high durability for the repetitive execute cycles. This technique also can be used in the proposed circuit to achieve high reliability.

The benefits of FC-based architectures compared to the spin-based architectures are low-power and high-speed [12]. Different from spin-based devices, the FC is capacitance-based device, and then no direct current flows in operation. As a result, low power is achieved. Moreover, the merit of small switching charge of the FC enables the high-speed. The drawback of FC-based architectures compared to the spin-based architectures is assumed to be the area, since spin-based devices is expected to become smaller than the FC in the scaled processes. However, the total area of a logic-in-memory architecture based on the proposed circuit is determined by the larger one of the area of the non-volatile memory devices and the area of the MOSFETs. This is because these non-volatile memory devices are placed directly on the top of the MOSFETs, as shown in Figure 26. When the proposed circuit is designed using a 0.35μm-CMOS/0.6μm-ferroelectric-capacitor process, the area of the MOSFETs is about five times larger than that of the FCs, and determines the area of the architecture. In this case, the area of the FC-based architecture and that of the spin-based architecture are the same. Even in scaled processes, this trend will likely continue for a long time until the circuit area of MOSFETs is larger than that of FCs.

In addition, the proposed circuit is used for the capacitance-based non-volatile devices, and not only for the FC. The FC is one type of capacitance-based non-volatile devices that we can use. In the future, more advanced capacitance-based memory devices than the FC will probably be proposed. In those devices, the proposed circuit will be more useful. Moreover, the proposed circuit structure can be also applied to other storage devices. For example, spin-based devices can be used instead of FCs, and the function of an FFC element can be implemented by using resistive voltage division instead of capacitive coupling effect.

6 CONCLUSION

This paper proposed low-power and high-functional ferroelectric-based logic circuits called FFC elements for logic-in-memory architectures, in order to solve the "Von Neumann bottleneck" and reduce the standby power. In an FFC element, storage and a logic function are integrated on the ferroelectric-capacitors, and the FFC element is suitable for logic-in-memory architectures which can solve the "Von Neumann bottleneck". The storage of the FFC element is non-volatile, and the standby power is greatly reduced because no permanent voltage supply is required to hold the stored data. Moreover, for area efficiency, the FFC element can flexibly change the access transistor network and introduces the multiple-valued storage and logic techniques, in order to achieve high functionality with a small number of transistors and FCs.

The proposed FFC element can be exploited in SIMD (Single Instruction Multiple Data) architectures such as for image processing as shown in Figures 27 and 28. In SIMD architectures, the area of the control circuit is small, and the large proportion of the area is occupied by the processing elements. Therefore, the efficient implementation for processing elements is important. Since each processing element consists of the circuits for the storage and logic functions, the FFC elements are suitable for its implementation. This is because the FFC elements integrate the storage and logic functions. In the FFC-based SIMD architecture shown in Figure 27, each processing element executes a word. Since the processing elements are controlled by

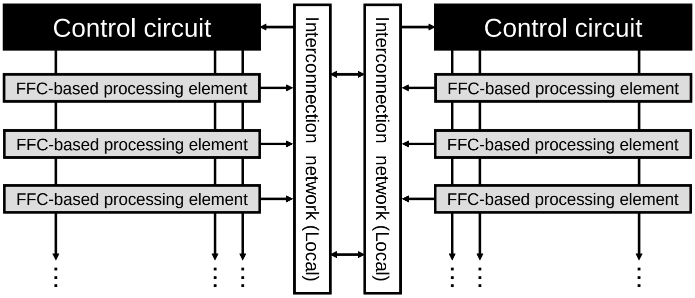

Control circuit
FFC-based processing element
FFC-based processing element
FFC-based processing element
Interconnection network (Local)
Interconnection network (Local)
Control circuit
FFC-based processing element
FFC-based processing element
FFC-based processing element

FIGURE 27
SIMD architecture exploiting FFC elements.

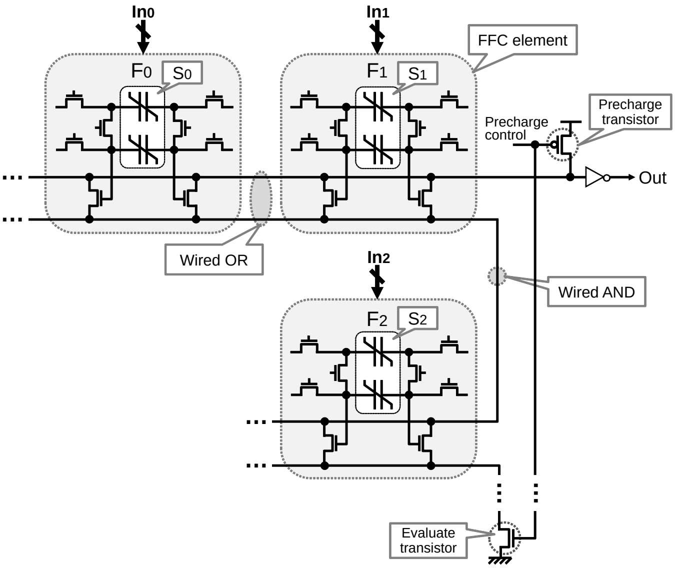


FIGURE 28
FFC-based processing element.

the same control circuit, the same operation is simultaneously performed on a large number of words. As a result, word-parallelism is achieved. Moreover, as shown in Figure 28, a processing element consists of FFC elements, and each FFC element has its own storage and logic function. Therefore, the same operation is simultaneously performed on a large number of bits in a word, and bit-parallelism is achieved, like Content-Addressable Memories (CAMs).

ACKNOWLEDGMENT

This work is supported by VLSI Design and Education Center (VDEC) the University of Tokyo in collaboration with Cadence Design Systems Inc. and Synopsys Inc. This work is partially supported by ROHM CO., LTD.

REFERENCES

[1] Hariyama, M., Ishihara, S., Idobata, N., and Kameyama, M. (2008). Non-volatile Multi-Context FPGAs Using Hybrid Multiple-Valued/Binary Context Switching Signals. In *Proc. International Conference on Engineering of Reconfigurable Systems & Algorithms (ERSA)*, pages 309–310.

[2] Ishihara, S., Idobata, N., Hariyama, M., and Kameyama, M. (2009). A Fine-Grain SIMD Architecture Based on Flexible Ferroelectric- Capacitor Logic. In *Proc. International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, pages 271–274.

[3] Ishihara, S., Idobata, N., Hariyama, M., and Kameyama, M. (2010). A Switch Block Architecture for Multi-Context FPGAs Based on Ferroelectric-Capacitor Functional Pass-Gate Using Multiple/Binary Valued Hybrid Signals. *IEICE Trans. Inf. & Syst.*, **E87-D(8)**, pages 2134–2144.

[4] Ishihara, S., Idobata, N., Nakatani, Y., Hariyama, M., and Kameyama, M. (2011). A Switch Block for Multi-Context FPGAs Based on Floating-Gate-MOS Functional Pass-Gates Using Multiple/Binary Valued Hybrid Signals. *Journal of Multiple-Valued Logic and Soft Computing*, **17(5-6)**, pages 553–580.

[5] Kimura, H., Hanyu, T., Kameyama, M., Fujimori, Y., Nakamura, T., and Takasu, H. (2003). Complementary Ferroelectric-Capacitor Logic for Low-Power Logic-in-Memory VLSI. In *IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, pages 160–161.

[6] Kimura, H., Hanyu, T., Kameyama, M., Fujimori, Y., Nakamura, T., and Takasu, H. (2004). Complementary Ferroelectric-Capacitor Logic for Low-Power Logic-in-Memory VLSI. *IEEE Journal of Solid-State Circuits*, **39(6)**, pages 919–925.

[7] Noda, H., Nakajima, M., Dosaka, K., Nakata, K., Higashida, M., Yamamoto, O., Mizumoto, K., Tanizaki, T., Gyohten, T., Okuno, Y., Kondo, H., Shimazu, Y., Arimoto, K., Saito, K., and Shimizu, T. (2007). The Design and Implementation of the Massively Parallel Processor Based on the Matrix Architecture. *IEEE Journal of Solid-State Circuits*, **42(1)**, pages 183–192.

[8] Pagiamtzis, K., and Sheikholeslami, A. (2004). Content-Addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey. *IEEE J. Solid-State Circuits*, **41(3)**, pages 712–727.

[9] Takashima, D., Shiga, H., Hashimoto, D., Miyakawa, T., Shiratake, S., Hoya, K., Ogiwara, R., Takizawa, R., Doumae, S., Fukuda, R., Watanabe, Y., Fujii, S., Ozaki, T., Kanaya, H., Shuto, S., Yamakawa, K., Kunishima, I., Hamamoto, T., Nitayama, A. (2010). A scalable shield-bitline-overdrive technique for 1.3V Chain FeRAM. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 262–263.

[10] Qazi, M., Clinton, M., Bartling, S., Chandrakasan, A.P. (2011). A low-voltage 1Mb FeRAM in 0.13m CMOS featuring time-to-digital sensing for expanded operating margin in scaled CMOS. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 208–210.

[11] Qazi, M., Clinton, M., Bartling, S., Chandrakasan, A.P. (2012). A Low-Voltage 1 Mb FRAM in 0.13 um CMOS featuring time-to-digital sensing for expanded operating margin. IEEE Journal of Solid-State Circuits, **47(1)**, pages 141–150.

[12] Takashima, D. (2011). Overview of FeRAMs: Trends and Perspectives. In *Proc. Non-Volatile Memory Technology Symposium (NVMTS)*, pages 36–41.