A Switch Block Architecture for Multi-Context FPGAs Based on a Ferroelectric-Capacitor Functional Pass-Gate Using Multiple/Binary Valued Hybrid Signals

Shota ISHIHARA†a), Student Member, Noriaki IDOBATA†, Nonmember, Masanori HARIYAMA†, Member, and Michitaka KAMEYAMA†, Fellow

SUMMARY Dynamically Programmable Gate Arrays (DPGAs) provide more area-efficient implementations than conventional Field Programmable Gate Arrays (FPGAs). One of typical DPGA architectures is multi-context architecture. An DPGA based on multi-context architecture is Multi-Context FPGA (MC-FPGA) which achieves fast switching between contexts. The problem of the conventional SRAM-based MC-FPGA is its large area and standby power dissipation because of the large number of configuration memory bits. Moreover, since SRAM is volatile, the SRAM-based multi-context FPGA is difficult to implement power-gating for standby power reduction. This paper presents an area-efficient and nonvolatile multi-context switch block architecture for MC-FPGAs based on a ferroelectric-capacitor functional pass-gate which merges a multiple-valued threshold function and a nonvolatile multiple-valued storage. The test chip for four contexts is fabricated in a 0.35 μm-CMOS/0.60 μm-ferroelectric-capacitor process. The transistor count of the proposed multi-context switch block is reduced to 63% in comparison with that of the SRAM-based one.

key words: dynamically programmable gate array, multi-context switch, logic-in-memory circuit, multiple-valued threshold logic, nonvolatile storage, non-destructive operation

1. Introduction

Dynamically Programmable Gate Arrays (DPGAs) [1], [2] provide more area-efficient implementations than conventional Field Programmable Gate Arrays (FPGAs). Since most of the computations do not run simultaneously, large applications can be broken into several small sub-programs. These sub-programs can share the same hardware resources by scheduling them into different time slots.

One of typical DPGA architectures is multi-context architecture [3]. An DPGA based on multi-context architecture is Multi-Context FPGA (MC-FPGA). In MC-FPGAs, each program or application is assigned to a separate context. The MC-FPGAs have multiple memory bits per configuration bit for fast switching contexts. Figure 1 shows a typical MC-FPGA architecture which consists of a mesh-connected cellular array. Each cell consists of a logic block and a switch block. For a configuration data, a Multi-Context switch (MC-switch) is used. Figure 2 shows structure of the MC-switch with four contexts. In an MC-switch, each memory bit corresponds to a configuration memory bit for the relevant context. The configuration data is selected according to the signal of the context-ID (context signal). The context signal is broadcasted to all MC-switches. Figure 3 shows an example of a function of the MC-switch. The vertical axis denotes the state of the MC-switch (ON or OFF), and the horizontal axis denotes the context-ID. If the MC-switch is ON, terminals $T_L$ and $T_R$ are connected. Otherwise, the terminals are disconnected. In this example, the MC-switch is ON at contexts 0 and 2, and is OFF at contexts 1 and 3. To implement this function, the storage values of SRAM bits (M0, M1, M2, M3) in Fig. 2 are set to (1, 0, 1, 0).

The conventional MC-FPGAs using SRAM (Static Random Access Memory) as the configuration memory...
has two major problems [4]. The first problem is that the SRAM-based MC-FPGAs have large area and standby power for the configuration memory. The MC-FPGAs have a large number of memory bits to store the configuration data. Since the large area of an SRAM bit, the area of the configuration memory is dominant in area. Moreover, the large number of SRAM bits causes a problem in standby power because of their leakage current. The second problem is that the SRAM-based MC-FPGAs is difficult to apply for lower-power embedded applications such as mobile information applications since the power-up time and configuration power is large. The low-power embedded applications tend to require frequent power ON and OFF to save power consumption. However, the SRAM-based MC-FPGAs lose their configuration data when power is OFF. The SRAM-based MC-FPGAs must be reconfigured every time when power is ON. The configuration data is typically stored in an external nonvolatile memory such as the EEPROM or FLASH memory, and the data is loaded into the MC-FPGAs after power up.

In order to solve these problems, an MC-switch based on a Functional Pass-Gate (FPG) using multiple-valued signals was proposed [5]. The FPG performs as a pass transistor and a threshold logic, where the threshold value is stored in the FPG [6]. In Ref. [5], a Floating-Gate MOS (FGMOS) transistor is used as an FPG. To implement the efficient threshold function, multiple-valued signals are used. By using the FGMOS-FPG-based MC-switch, the MC-FPGA achieves small area, low standby power and instant power-ON. Although the MC-switch is more area-efficient than the SRAM-based one, it still has redundancy. In other words, there are multiple different threshold patterns to implement the same function.

In order to eliminate this redundancy, an MC-switch based on FGMOS-FPG using multiple/binary valued hybrid signals was proposed [7], [8]. Like Ref. [5], multiple-valued signals are used to implement the threshold function. Moreover, binary-valued signals are also used to divide the function of an MC-switch into sub-functions such that each sub-function has only two contexts. Since the sub-functions are not overlapped in contexts, there is no redundancy in the MC-switch. However, FGMOSs require a dedicated process and a high programming voltage. As a result, FGMOSs are not suitable to integrate with CMOS circuits on a single chip.

This paper presents an area-efficient and nonvolatile MC-switch based on Ferroelectric-Capacitor FPG (FC-FPG). Like Refs. [7], [8], the proposed MC-switch uses the multiple/binary valued hybrid signals to eliminate the redundancy of the MC-switch. The process of the Ferroelectric-Capacitors (FCs) is compatible with CMOS transistors and the programming voltages of the FCs are the same as CMOS transistors. As a result, FCs are suitable to integrate with CMOS circuits on a single chip. Capacitive coupling of two FCs is exploited to realize the multiple thresholds using FCs that are binary storage devices. The test chip for four contexts is fabricated in a 0.35 µm-CMOS/0.60 µm-ferroelectric-capacitor process. The transistor count of the proposed multi-context switch block is reduced to 63% in comparison with that of the SRAM-based one.

2. Previous Work

An MC-switch based on an FPG using only multiple-valued signal was proposed [5]. The MC-switch is designed using threshold functions and multiple-valued signals. For the case of N contexts, an arbitrary function of an MC-switch is given by OR-ing N/2 window-literals at most. In other word, an N-context MC-switch is implemented by an circuit which OR-ing N/2 window-literals. An window-literal is implemented by AND-ing an up-literal and a down-literal. An up-literal is a monotone increasing function. Given the logical threshold value \( L_{th} \) and logical control value \( C_m \), an up-literal \( F_{UL}(C_m, L_{th}) \) is given by

\[
F_{UL}(C_m, L_{th}) = \begin{cases} 
1 & L_{th} \leq C_m \\
0 & \text{otherwise}
\end{cases}
\]  

(1)

An down-literal is a monotone decreasing function. Given the logical threshold value \( L_{th} \) and control value \( C_m \), a down-literal \( F_{DL}(C_m, L_{th}) \) is given by

\[
F_{DL}(C_m, L_{th}) = \begin{cases} 
1 & C_m \leq L_{th} \\
0 & \text{otherwise}
\end{cases}
\]  

(2)

In order to implement the MC-switch in an area-efficient manner, an FPG is employed [6]. Figure 4 shows the block diagram of the FPG and its function. The FPG performs as a pass transistor and a threshold logic, the logical value of the threshold is programmable and is stored in the FPG. The logical value of the threshold is called \( L_{th} \) in this paper. If the logical value of the control signal (Control) is larger than the logical threshold value, terminals \( T_L \) and \( T_R \) are connected, and the state of the FPG is ON. Otherwise, they are disconnected, and the state of the FPG is OFF. As a result, the FPG implements an up-literal.

Figure 5 shows the MC-switch based on FPG, The equivalent SRAM-based circuit of the MC-switch is shown in Fig. 2. Terminals \( T_L \) and \( T_R \) in Fig. 5 are respectively correspond to terminals \( T_L \) and \( T_R \) in Fig. 2. Signals \( C_m \) and \( C_m \) are logical multi-valued signals. For N contexts, the values of \( C_m \) 0, 1, 2 and 3 correspond to the context-ID 0, 1, 2 and 3. The logical value of \( C_m \) is defined as \( N - C_m - 1 \). Hence, \( C_m = N - 3 \) for \( N = 4 \). Note that the down-literal \( F_{DL} \) for \( C_m \) is implemented by the up-literal using \( C_m \). As a result, the FPG using \( C_m \) as the control signal implements a...
up-literal, and the FPG using $\overline{C_m}$ as the control signal implements a down-literal. Since a window-literal is generated by wire-ANDing an up-literal and a down-literal, the upper two FPGs implement a window-literal and the lower two FPGs implement another one. By wire-ORing the two window-literals, an arbitrary function of an $N$-context MC-switch is implemented. In Ref. [5], the FPG is implemented by a single four-valued FGMOS where the logical threshold value is programmed as $-0.5, 0.5, 1.5$ or $2.5$. As a result, a four-context MC-switch consists of four FGMOSs. The problem of the MC-switch using multiple-valued signals is its redundancy. The MC-switch consists four FPGs, each of which stores one of four threshold values. Therefore, the number of storage patterns of the MC-switch is $4^4 = 256$. However, the required number of storage patterns for a 4-context MC-switch is only $2^4 = 16$. Therefore, most of the storage patterns, $240 (= 256 - 16)$ patterns are redundant. Due to this redundancy, the MC-switch using multiple-valued signals is inefficient in area.

3. MC-Switch Based on an FPG Using Multiple/Binary Valued Hybrid Signals

In order to eliminate the redundancy of the MC-switch using multiple-valued signals, binary-valued signals are combined with multiple-valued signals in the MC-switch [7], [8].

Figure 6(a) shows the function $F$ of an MC-switch for four contexts. As shown in Figs. 6(b) and (c), function $F$ is divided into two sub-functions $F_1$ and $F_2$ in such that each sub-function has only two contexts. Sub-function $F_1$ is for context 0 and context 1, and sub-function $F_2$ is for context 2 and context 3. Since the two sub-functions are not overlapping in contexts, there is no redundancy in the MC-switch using multiple/binary valued hybrid signals. A sub-function with two contexts is an up-literal or a down-literal. As a result, the MC-switch only requires two literals for sub-function $F_1$ and sub-function $F_2$, respectively.

Figure 7 shows the MC-switch using multiple/binary valued hybrid signals. The function of the MC-switch is generated by wire-ORing two FPGs. Two multiplexers are used for selecting the context signals for the two FPGs. As mentioned in Sect. 5, in a switch block, such multiplexers are shared among a lot of MC-switches. As a result, the hardware overhead of the multiplexers is negligible. The upper FPG operates at context 0 and context 1, and the lower one at context 2 and context 3. Each control signal of the FPGs is a product of a logical binary-valued context signal ($C_b$ or $\overline{C_b}$) and a logical multiple-valued context signal ($C_m$ or $\overline{C_m}$), where the product of $C_b$ and $C_m$ is given by

$$C_b \cdot C_m = \begin{cases} 0 & (C_b = 0) \\ C_m & (C_b = 1). \end{cases}$$

Table 1 summarizes the relation between the context-ID and the logical control signals ($C_b$, $\overline{C_b}$, $C_m$, $\overline{C_m}$). Table 2 shows
all functions of the MC-switch with four contexts. The left part shows the target functions. Columns “C0”, “C1”, “C2” and “C3” denote the states of the target functions at contexts 0, 1, 2 and 3, respectively. The right part shows the logical control signals and the logical threshold values of the upper FPG and the lower FPG to implement the target functions. This table shows that the MC-switch implements an arbitrary function for four contexts. Figure 8 shows the multiple/binary valued hybrid control signals used in the MC-switch. Signal $C_b$ determines which context each FPG operates at. The value “0” of $C_b$ corresponds to contexts 0 and 1, and value “1” to contexts 2 and 3. When $C_b = 0$ (contexts 0 and 1), only the upper FPG operates, while the FPG “operates” means that the state of the FPG is determined by the threshold operation. The lower FPG does not operate (is OFF) since the logical control signals $C_b \cdot C_m$ and $C_b \cdot \overline{C_m}$ are 0 for $C_b = 0$. On the other hand, when $C_b = 1$ (contexts 2 and 3), only the lower FPG operates, and the upper FPG is OFF. Signal $C_m$ and $\overline{C_m}$ are used to generate the control signals for an up-literal and a down-literal, respectively. The logical threshold value of the FPG corresponds to the threshold of the literal. For example, if the control signal $C_b \cdot C_m$ shown in Fig. 8(a) is selected, threshold operation is done at the FPG between $C_b \cdot C_m$ and the pre-determined logical threshold value $L_{th}$ as shown in Fig. 9. As a result, an up-literal for contexts 0 and 1 is obtained. Similarly, if the control signals $C_b \cdot \overline{C_m}$, $C_b \cdot C_m$, and $C_b \cdot \overline{C_m}$ shown in Fig. 8(b), (c) and (d) are selected, we obtain respectively a down-literal for contexts 0 and 1, an up-literal for contexts 2 and 3, and a down-literal for contexts 0 and 1.

| Target function | Upper FPG | Lower FPG |
|-----------------|-----------|-----------|
| C0 C1 C2 C3     | Lb Cb     | Lh Cm     |
| 0 0 0 0         | 2.5       | 0.5       |
| 0 0 0 1         | 2.5       | 1.5       |
| 0 0 1 0         | 2.5       | 1.5       |
| 0 0 1 1         | 2.5       | 0.5       |
| 0 1 0 0         | 1.5       | 1.5       |
| 0 1 0 1         | 1.5       | 1.5       |
| 0 1 1 0         | 1.5       | 1.5       |
| 0 1 1 1         | 1.5       | 1.5       |
| 1 0 0 0         | 1.5       | 1.5       |
| 1 0 0 1         | 1.5       | 1.5       |
| 1 0 1 0         | 1.5       | 1.5       |
| 1 0 1 1         | 1.5       | 1.5       |
| 1 1 0 0         | 0.5       | 1.5       |
| 1 1 0 1         | 0.5       | 1.5       |
| 1 1 1 0         | 0.5       | 1.5       |
| 1 1 1 1         | 0.5       | 0.5       |

Fig. 8  Hybrid signals used in the MC-switch using multiple/binary valued hybrid signals.

Fig. 9  Implementation for an up-literal.

4. MC-Switch Based on a Ferroelectric-Capacitor FPG Using Multiple/Binary Valued Hybrid Signals

4.1 Fundamental Principle of Ferroelectric-Capacitors

A Ferroelectric-Capacitor (FC) is one of nonvolatile storage. The FC is physically distinguished from a regular capacitor by substituting the dielectric with a ferroelectric material as shown in Fig. 10(a). Figure 10(b) is the symbol of an FC. An FC has two directions of the remnant-polarization, and an FC is used as a variable capacitor. The capacitance of the FC is determined by the direction of the remnant-polarization and the direction of the Electric Potential Difference (EPD) applied across the FC. If the direction of the remnant-polarization and the direction of the EPD are the same, the capacitance of the FC is small. If the direction of the remnant-polarization and the direction of the EPD are the opposite, the capacitance of the FC is large. 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. An FC-FPG (Ferroelectric-Capacitor Functional Pass-Gate) which executes non-destructive operation was proposed [9], [10]. The FC-FPG executes only binary-valued functions and cannot execute multiple-valued functions. Therefore, the FC-FPG cannot be applied to the MC-switch using multiple/binary valued signals.

4.2 Multiple-Valued Ferroelectric-Capacitor FPG

A multiple-valued FC-FPG for three-valued threshold functions was proposed [11]. Figure 11 shows the basic structure of the FC-FPG and its threshold functions, where values \( S_1 \) and \( S_2 \) denote the polarization directions of the left FC and the right FC, respectively. Depending on \( S_1 \) and \( S_2 \), there exists three threshold functions whose threshold voltages are different from each other. Note that the threshold function in Fig. 11 (b) is implemented by two different sets \((S_1, S_2) = (0, 0)\) and \((1, 1)\). In the following, the set \((S_1, S_2) = (0, 0)\) is used for the threshold function in Fig. 11 (b) since it executes non-destructive operation and the set \((S_1, S_2) = (1, 1)\) does not. To execute the threshold function, voltage \( V_{mul} \) is applied to terminal \( t_1 \), and voltage \( 0V \) is applied to terminal \( t_2 \). \( V_{mul} \) is the voltage of a multiple-valued signal corresponding to the control signal of the FPG shown in Fig. 4 (a). 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. Let \( V_{t1} \) and \( V_{t2} \) be the voltages of the terminals \( t_1 \) and \( t_2 \), respectively. Since the relationship among the terminal voltages is \( V_{t1} \geq V_G \geq V_{t2} \), the directions of the EPD applied across each FC is always left.

Figures 11 (a), (b) and (c) are arranged in an increasing order of the logical threshold value.

Figure 11 (a) shows the FC-FPG which the threshold voltage is smallest, and the logical threshold value is defined as 0.5. In Fig. 11 (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 of the two FCs, \( V_G \) determines the state (ON/OFF) of the pass transistor. Let \( V_{t1} \) and \( V_{t2} \) be the voltages of the terminals \( t_1 \) and \( t_2 \), respectively. Since the relationship among the terminal voltages is \( V_{t1} \geq V_G \geq V_{t2} \), the directions of the EPD applied across each FC is always left.

Figure 11 (a) shows the FC-FPG which the threshold voltage is smallest, and the logical threshold value is defined as 0.5. In Fig. 11 (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. As a result, voltage \( V_G \) is approximately the same as voltage \( V_{mul} \). In this way, the operation is non-destructive, and the reason is as follows. In the left 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 left FC does not change. In the right 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 right FC does not change.
it executes non-destructive operation and the set \((S_1, S_2) = (1, 1)\) does not. In the set \((S_1, S_2) = (0, 0)\), the direction of the remnant-polarization of the left FC and that of the right FC are left. Since the direction of the EPD applied across each FC is left, the capacitance of the left FC and 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 of the right FC. As a result, voltage \(V_G\) is approximately the same as voltage \(V_{mul}/2\). If the voltage of input signal \(V_{mul}\) is the same, \(V_G\) of Fig. 11 (a) is smaller than that of 11 (b). Therefore, to turn ON the pass transistor, Fig. 11 (b) requires a higher voltage of \(V_{mul}\). As a result, the threshold voltage of Fig. 11 (b) is larger than that of Fig. 11 (a), and the logical threshold value of Fig. 11 (b) is defined as 1.5. In this way, the operation is non-destructive, and the reason is as follows. In the left FC and the right FC, the direction of the EPD applied across each FC is the same as the direction of the remnant-polarization of the FC. Therefore, the remnant-polarization of the FCs do not change. The set \((S_1, S_2) = (1, 1)\) has the same logical threshold value as the set \((S_1, S_2) = (0, 0)\) but it executes destructive operation. The reason that the set \((S_1, S_2) = (1, 1)\) executes destructive operation is as follows. In the left FC and the right FC, the direction of the EPD applied across each FC is opposite to that of the remnant-polarization, and moreover, the amount of EPD is larger than the coercive voltage. Therefore, the remnant-polarization of the FCs change.

Figure 11 (c) shows the FC-FPG which the threshold voltage is larger than Fig. 11 (b), and the logical threshold value is defined as 2.5. In the direction of each FCs in 11 (c) is the opposite of that in 11 (a). Therefore, in Fig. 11 (c), the EPD applied across the left FC is much larger than that of the right FC. As a result, \(V_G\) is approximately the same as 0V. The threshold voltage is larger than Fig. 11 (b), and the logical threshold value is defined as 2.5. Similarly to Fig. 11 (a), the operation is non-destructive.

When implementing the MC-switch using the basic FC-FPG, there are two problems. The first one is that the gate voltage of the pass transistor \(V_G\) is not complete “\(Vdd\)” or “\(0V\)”, and then the pass transistor is not complete “ON” or “OFF”. As a result, the delay and the power consumption of the data path are increased. The second one is the leakage current between the two electrode or in FC. The leakage current changes voltage \(V_G\) after executing the threshold function, and then causes a malfunction.

To solve these problems, we proposed an FC-FPG with a sense amplifier and a feed-back loop in Ref. [11]. Figure 12 shows the structure of the FC-FPG. The sense amplifier is used to binarize the gate voltage of the pass transistor \(V_G\). The feed-back loop is used to apply the same voltage across the two electrodes of each FC, and the remnant-polarizations of the FCs are retained steadily. However, the feed-back loop causes an area overhead because mainly of the large transistor count of the feed-back loop using multiplexers.

In order to overcome this area overhead, an area-efficient FC-FPG which does not require the multiplexers is proposed in this paper. In the proposed FC-FPG, two cross-coupled inverters perform not only as a sense amplifier to binarize the gate voltage of the pass transistor \(V_G\) but as a memory bit to store the binarized result. As a result, the multiplexers for the feed-back loop of Fig. 12 is eliminated.

Figure 13 shows the structure of the proposed FC-FPG. The two cross-coupled inverters perform as a sense amplifier to binarize the output voltage and as a memory bit to store the binarized output. The FC-FPG has three modes: INITIALIZE mode, OPERATE mode and STORE mode. To execute a function, the FC-FPG executes the modes in the order: INITIALIZE mode, OPERATE mode and STORE mode. Figure 14 shows the behavior of the INITIALIZE mode. Note that all the values of the signals shown in Fig. 14 are logical values. Figure 14 (a) shows the FC-FPG in the INITIALIZE mode, and Fig. 14 (b) shows the equi-
In the INITIALIZE mode, in order to initialize node OutG, terminal Initialize is set to logical value “1” and terminal t2 is set to logical value “0”. Moreover, in order not to change the polarization of the FCs, terminals t1 and t3 are also set to logical value “0” which is the same as terminal t2. Figure 15 shows the behavior of the OPERATE mode. Figure 15 (a) shows the FC-FPG in the OPERATE mode, and Fig. 15 (b) shows the equivalent circuit. In the OPERATE mode, the FC-FPG executes the threshold functions shown in Fig. 11, and the cross-coupled inverters which perform as a sense amplifier binarizes the output OutG to a binary signal OutSA. Figure 16 shows the behavior of the STORE mode. Figure 16 (a) shows the FC-FPG in the STORE mode, and Fig. 16 (b) shows the equivalent circuit. In the STORE mode, signal OutG is set to the binarized result OutSA, and is stored in the cross-coupled inverters. Moreover, in order not to change the polarization of the FCs, terminal Store is set to logical value “0” and terminal Store is set to logical value “1”. As a result, the EPD of two electrodes of each FC is “0”, and then the polarization of each FC is retained steadily.

5. Area-Efficient Switch Block Architecture and Evaluation

MC-FPGAs have a high degree of redundancy in configuration data between contexts [4], [12]. That is, in MC-FPGAs, less than 3% of configuration data are changed when contexts are switched. Therefore, most of configuration data are always “1” or always “0”, and most of MC-switches are “always ON” or “always OFF” as shown in Table 3, where columns “C0”, “C1”, “C2” and “C3” denote the states of the target functions at contexts 0, 1, 2 and 3, respectively. Based on this observation, an area-efficient switch block using the proposed MC-switch is proposed. Figure 17 shows the structure of the switch block. The switch block is based on typical crossbar network structure, where crosspoint switches are placed in a two dimensional array of N columns and N rows. In the switch block, a crosspoint switch is implemented by a single MC-switch. The switch block is divided into the upper part and the lower part, each of which consists of N/2 rows of MC-switches. The multiplexers on the top and on the bottom of each part are used to select the control signals for the MC-switches. Each of the top multiplexers corresponds to the top multiplexer of Fig. 7, which is used to control the upper FPG of the MC-switch of Fig. 7. Similarly, each of the bottom multiplexers corresponds to the bottom multiplexer of Fig. 7. Figure 18 shows the structure of the multiplexer. The multiplexer consists of a conventional multiplexer and an FC-based non-volatile SRAM [13]. In order to reduce the area overhead of the multiplexers, each of the multiplexers is shared among

| Table 3 Configuration patterns of always ON and always OFF. |
|----------------------|------|------|------|------|
| Always ON            | C0   | C1   | C2   | C3   |
| Always OFF           | 0    | 0    | 0    | 0    |

![Fig. 17 Switch block structure using the proposed MC-switches.](image-url)
the MC-switches on the same column of each part. Note that all MC-switches on the same column of each part implement either only up-literals or only down-literals for each two contexts since the controls from the top and bottom multiplexers are shared among them. Figure 19 shows a single column of the switch block. First, let us focus on the functions of the MC-switches at contexts 0 and 1. The relation between the control signals selected by the top multiplexers and functions of the MC-switches is shown in Table 4. The left columns of Table 4 shows the control signals which are selected by the multiplexers. Columns “Top MUX of the upper part” and “Top MUX of the lower part” denote the control signals selected by the top multiplexer of the upper part and the top multiplexer of the lower part, respectively. The right columns of Table 4 shows the functions implemented by the MC-switches. Column “Upper part at C0 and C1” denotes the functions implemented by each MC-switch on the upper part at contexts 0 and 1. Similarly, column “Lower part at C0 and C1” denotes the functions implemented by each MC-switch on the lower part at contexts 0 and 1. For example, if the top multiplexer of the upper part selects \( \overline{C_k} \cdot C_m \) and the top multiplexers of the lower part selects \( C_k \cdot \overline{C_m} \), each MC-switch of the upper part implements an up-literal and each MC-switch of the lower part implements a down-literal at contexts 0 and 1. Similarly, the functions of the MC-switches at contexts 2 and 3 are determined by the control signals which are selected by the bottom multiplexers. As mentioned above, most of MC-switches are “always ON” or “always OFF”. Therefore, an efficient way of implementing “always ON” or “always OFF” for each MC-switch is important. In the proposed switch block, although the control signals are shared among the MC-switches on the same column, each MC-switch can be independently set to “always ON” or “always OFF”. This is because configuration patterns of “always ON” and “always OFF” can be implemented by either an up-literal or a down-literal. The mapping efficiency for more complex configuration patterns can be improved by mapping same-type configuration patterns onto the same part of the same column. For example, the MC-switches which implement up-literals are assigned to the upper part, and the MC-switches which implement up-literals are assigned to the lower part.

The test chip for four contexts is fabricated in 0.35 \( \mu \)m-CMOS/0.60 \( \mu \)m-ferroelectric-capacitor process. Figure 20 shows the micro-photograph of the test chip. The three-valued threshold operation of the FC-FPG on the test chip is confirmed. The logical values for the multiple-valued signals “0”, “1” and “2” are represented by voltages 0V, 2.0V and 3.3V, respectively.

Table 5 shows the comparison in terms of transistor counts between the FC-FPG-based MC-switch in Ref. [11] and the proposed MC-switch. Compared to the FC-FPG-
Table 5  Comparison in terms of transistor counts between the MC-switch in Ref. [11] and the proposed MC-switch (four contexts).

|                     | MC-switch in Ref. [11] | Proposed |
|---------------------|------------------------|----------|
| Transistor count    | 34                     | 22       |
| FC count            | 4                      | 4        |

Table 6  Comparison between the SRAM-based MC-switch and the proposed MC-switch.

|                     | SRAM-based | Proposed |
|---------------------|------------|----------|
| Transistor count    | 37         | 22       |
| Capacitor count     | 0          | 4        |
| Leakage current     |            |          |
| Active              | 153pA      | 62pA     |
| Standby             | 153pA      | 0A       |
| Dynamic power       | 445mW      | 179mW    |
| of the context      | (Total: 410mW) | (Total: 410mW) |
| switching           |            |          |
| Delay of the context switching | 427ps | 237ps |
|                     | (Total: 153ps) | (Total: 153ps) |

Fig. 21  Relations between switch block size and transistor count.

Fig. 22  Structure of the proposed MC-switch with eight contexts.

Based MC-switches in Ref. [11], the transistor count is reduced to 65%, because the feedback loop using multiplexers is eliminated in the proposed MC-switch.

Table 6 shows the comparison between the SRAM-based MC-switch and the proposed MC-switch. In terms of the transistor count, the proposed MC-switch reduces the transistor count to 59%, because the threshold function and the nonvolatile storage are merged into the proposed FC-FPG. Although the proposed MC-switch requires FCs and the SRAM-based one does not, the FCs of the proposed MC-switch are placed directly on the top of the CMOS transistors and there is no area overhead. In terms of the leakage current, the proposed MC-switch reduces the leakage current to 41% in the active state. Moreover, since the proposed MC-switch is nonvolatile, it is possible to turn OFF the power supply for saving the standby power in the standby state. In terms of the dynamic power of context switching, the proposed MC-switch is smaller than that of the SRAM-based one. In terms of the delay of context switching, the proposed MC-switch is larger than that of the SRAM-based one. The reason is as follows. The FC-FPG has three modes: INITIALIZE mode, OPERATE mode and STORE mode. To execute a function, the FC-FPG executes the modes in the order, INITIALIZE mode, OPERATE mode and STORE mode. The delay of the OPERATE mode occupies a large portion of the total delay. From Table 6, the delay of the context switching of the FC-FPG is larger than that of the SRAM-based MC-switch. This is mainly due to the large delay of the OPERATE mode. In the OPERATE mode, the data stored in the FCs is read, and then the data is binarized by the sense amplifier. The large delay of the OPERATE mode is due to the multiple level voltages. In other words, since the voltage of the input of the sense amplifier is not complete “Vdd” or “Gnd”, the delay of the sense amplifier is large. A possible solution is using a high-speed sense amplifier. However, such a high-speed sense amplifier requires a large area overhead. In the proposed MC-switch using FC-FPGs, the delay of the context switching is about 1.5 ns, which is fast enough for switching the context within a clock cycle because even the clock period of state-of-art dynamically reconfigurable VLSIs is larger than 3.3 ns (300 MHz).

Figure 21 shows the relations between switch block size and transistor count. In actual MC-FPGAs, a switch block size has more than $32 \times 32$ crosspoint switches. In a $32 \times 32$ switch block, the transistor count of the SRAM-based one is reduced to 63%.

The proposed MC-switch can be extended for more than four contexts by adding FPGs. Figure 22 shows an MC-switch with eight contexts and Table 7 shows the code table of the signals used in Fig. 22. The MC-switch consists of four FPGs connected in parallel. Like the proposed four-
context MC-switch, each FPG implements a sub-function which has only two contexts. In other words, the FPGs FG1, FG2, FG3 and FG4 operate at sets of contexts (contexts 0 and 1), (contexts 2 and 3), (contexts 4 and 5) and (contexts 6 and 7), respectively. By wire-ORing the FPGs, the MC-switch is implemented. Note that the multiplexers for selecting control signals can be shared with other MC-switches like the proposed four-context MC-switch. Generally, an N-context MC-switch is implemented by N/2 FPGs connected in parallel.

6. Conclusion

An area-efficient MC-switch based on an FC-FPG using multiple/binary valued hybrid signals was presented. To eliminate the multiple different threshold patterns to implement the same function, binary-valued signals are used to divide the function into sub-functions. Since the sub-functions are not overlapped in contexts, there is no redundancy in the MC-switch. As a result, the number of FPGs for the MC-switch is reduced to half in comparison with the FPG-based MC-switch. As a result, the number of FPGs for the MC-switch is reduced to half in comparison with the FPG-based MC-switch. Generally, the MC-switch is implemented by N/2 FPGs connected in parallel.

Table 7 Code table of the signals used in the proposed MC-switch with eight contexts.

| Context-ID | Binary-valued | Multiple-valued |
|------------|---------------|-----------------|
|            | C0  | C1  | C0  | C1  | C0  | C1  | C0  | C1  |
| 0          | 0   | 1   | 0   | 1   | 0   | 1   | 0   | 1   |
| 1          | 0   | 1   | 0   | 2   | 1   | 0   | 1   | 2   |
| 2          | 0   | 1   | 1   | 0   | 1   | 2   | 1   | 0   |
| 3          | 0   | 1   | 1   | 0   | 2   | 1   | 2   | 0   |
| 4          | 1   | 0   | 0   | 1   | 1   | 2   | 1   | 0   |
| 5          | 1   | 0   | 0   | 2   | 1   | 2   | 0   | 1   |
| 6          | 1   | 0   | 1   | 0   | 1   | 2   | 1   | 0   |
| 7          | 1   | 0   | 1   | 0   | 2   | 1   | 0   | 1   |

FPG using resistive division. As a result, we can use advanced storage devices with high device densities for the FPGs.

Acknowledgments

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] A. DeHon, “Dynamically programmable gate arrays: A step toward increased computational density,” Proc. Canadian Workshop of Field-Programmable Devices, pp.47–54, 1996.
[2] A. DeHon, “DPGA utilization and application,” Proc. ACM International Symposium on Field-Programmable Gate Arrays, pp.115–121, 1996.
[3] S. Trimberger, D. Carberry, A. Johnson, and J. Wong, “A time-multiplexed FPGA,” Proc. IEEE Symposium on FPGA-Based Custom Computing Machines, pp.22–28, 1997.
[4] H.M. Waidyasooriya, W. Chong, M. Hariyama, and M. Kameyama, “Multi-context FPGA using fine-grained interconnection blocks and its CAD environment,” IEICE Trans. Electron., vol.E91-C, no.4, pp.517–525, April 2008.
[5] M. Hariyama, S. Ogata, and M. Kameyama, “A multi-context FPGA using floating-gate-MOS functional pass-gates,” IEICE Trans. Electron., vol.E89-C, no.11, pp.1655–1661, Nov. 2006.
[6] T. Hanyu, K. Teranishi, and M. Kameyama, “Multiple-valued logic-in-memory VLSI based on a floating-gate-MOS pass-transistor network,” IEEE Solid-State Circuits Conference (ISSCC) Digest of Technical Papers, pp.194–195, 1998.
[7] Y. Nakatan, M. Hariyama, and M. Kameyama, “Switch block architecture for multi-context FPGAs using hybrid multiple-valued/binary context switching signals,” Proc. International Symposium on Multiple-Valued Logic (ISMVL), pp.651–654, 2006.
[8] Y. Nakatan, M. Hariyama, and M. Kameyama, “Architecture of a multi-context FPGA using a hybrid multiple-valued/binary context switching signal,” Proc. International Parallel and Distributed Processing Symposium (IPDPS), pp.25–29, 2006.
[9] H. Kimura, T. Hanyu, M. Kameyama, Y. Fujimori, T. Nakamura, and H. Takasu, “Complementary ferroelectric-capacitor logic for low-power logic-in-memory VLSI,” IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers, pp.160–161, 2003.
[10] H. Kimura, T. Hanyu, M. Kameyama, Y. Fujimori, T. Nakamura, and H. Takasu, “Complementary ferroelectric-capacitor logic for low-power logic-in-memory VLSI,” IEEE J. Solid-State Circuits, vol.39, no.6, pp.919–925, 2004.
[11] Y. Nakatan, M. Hariyama, and M. Kameyama, “Architecture for multi-context FPGAs using ferroelectric-based functional pass-gates,” IEICE Technical Report, ICD2006-143, 2006.
[12] I. Kennedy, “Exploiting redundancy to speedup reconfiguration of an FPGA,” Proc. International Conference on Field Programmable Logic and Application, pp.262–271, 2003.
[13] T. Miwa, J. Yamada, H. Koike, T. Nakura, S. Kobayashi, N. Kasai, and H. Toyoshima, “A 512 Kbit low-voltage NV-SRAM with the size of a conventional SRAM,” Proc. Symposium on VLSI Circuits Digest of Technical Papers, pp.129–132, 2001.
[14] S. Masui, S. Fueki, K. Masutani, A. Inoue, T. Teramoto, T. Suzuki, and S. Kawashima, “The application of FeRAM to future information technology world,” Topics in Applied Physics, pp.271–283, 2004.
Shota Ishihara received the B.E. degree in Information Engineering and M.S. degree in Information Sciences from Tohoku University, Sendai, Japan, in 2007 and 2009, respectively. He is currently working toward the Ph.D. degree in Graduate School of Information Sciences, Tohoku University. His primary research interest is in the area of reconfigurable computing.

Noriaki Idobata received the B.E. degree in Information Engineering and M.S. degree in Information Sciences from Tohoku University, Sendai, Japan, in 2008 and 2010, respectively. His primary research interest is in the area of reconfigurable computing.

Masanori Hariyama received the B.E. degree in electronic engineering, M.S. degree in Information Sciences, and Ph.D. in Information Sciences from Tohoku University, Sendai, Japan, in 1992, 1994, and 1997, respectively. He is currently an associate professor in Graduate School of Information Sciences, Tohoku University. His research interests include VLSI computing for real-world applications such as robots, high-level design methodology for VLSIs and reconfigurable computing.

Michitaka Kameyama received the B.E., M.E. and D.E. degrees in Electronic Engineering from Tohoku University, Sendai, Japan, in 1973, 1975, and 1978, respectively. He is currently Dean and Professor in the Graduate School of Information Sciences, Tohoku University. His general research interests are intelligent integrated systems for real-world applications and robotics, advanced VLSI architecture, and new-concept VLSI including multiple-valued VLSI computing. He received the Outstanding Paper Awards at the 1984, 1985, 1987 and 1989 IEEE International Symposiums on Multiple-Valued Logic, the Technically Excellent Award from the Society of Instrument and Control Engineers of Japan in 1986, the Outstanding Transactions Paper Award from the IEICE in 1989, the Technically Excellent Award from the Robotics Society of Japan in 1990, and the Special Award at the 9th LSI Design of the Year in 2002. Dr. Kameyama is an IEEE Fellow.