Extended Abstracts of the 2014 International Conference on Solid State Devices and Materials, Tsukuba, 2014, pp110-111

Dynamically Reconfigurable Non-Volatile Multi-Context FPGA with CAAC-OS-based Programmable Routing Switches

Nora Bjorklund[1,2], Yuki Okamoto[1], Takeshi Aoki[1], Munehiro Kozuma[1],
Yoshiyuki Kurokawa[1], Takayuki Ikeda[1], Naoto Yamade[1], Yutaka Okazaki[1],
Hidekazu Miyairi[1], Masahiro Fujita[3], and Shunpei Yamazaki[1]

[1] Semiconductor Energy Laboratory Co., Ltd., 398 Hase, Atsugi-shi, Kanagawa, 243-0036, Japan
TEL: +81-46-248-1131, FAX: +81-46-270-3751, E-mail: nbi014@sel.co.jp
[2] Linkoping University, 581 83 Linkoping, Sweden
[3] VLSI Design and Education Center (VDEC), The University of Tokyo 2-11-16 Yayoi, Bunkyo-ku, Tokyo, 113-0032, Japan

Abstract

A dynamically reconfigurable non-volatile multi-context FPGA was realized using programmable routing switches constructed using a 1.0-µm crystalline OS FET/0.5-µm CMOS FET hybrid process. The programmable routing switch exhibits improved switching characteristics because of capacitive coupling at its floating node memory and enables the FPGA to safely implement dynamic reconfiguration without the interruption of its execution.

1. Introduction

A common implementation in field-programmable-gate array (FPGA) devices is the use of SRAM to store configurations. However, because SRAM-based designs are volatile, require a large area, and consume a large amount of power, techniques beyond the realm of CMOS have been emerging in recent years [1–4]. A hybrid process between the crystalline oxide semiconductor (OS) *c*-axis-aligned-crystalline In-Ga-Zn oxide (CAAC-IGZO) and CMOS can be used to decrease chip area with CAAC-IGZO FETs stacked on the top of the CMOS FETs. In previous studies, memories using this technology have been demonstrated to have several advantages, such as non-volatility, high-speed switching, and reduced retention power, compared to SRAM [2].

The CAAC-IGZO FET-based programmable routing switch (RS) in [2] stores configurations in a capacitance memory, thereby creating a floating node similar to that in FLASH devices. The memory node was boosted via capacitive coupling between the floating node and the source of a pass-gate transistor, a property that resulted in better switching capability. It is also a property that sets the CAAC-IGZO FET-based RS apart from other stackable non-volatile RSs such as MRAM [3] and RRAM [4].

In this paper, we propose a dynamically reconfigurable non-volatile FPGA that performs dynamic reconfiguration (DR) without interrupting the FPGA's execution (Fig. 1), by modifying the RS in [2]. The RS in [2] cannot correctly handle DR because its configuration memories are differently affected by boosting effects depending on the computations executed during DR. In contrast, the modified RS in this paper can eliminate such differences using a "pull-down" node on its input to create an even environment during DR, independent of the FPGA's active task.

2. The Modified Programmable Routing Switch

Reverse-boosting in the Reference Switch

The reference RS (Fig. 2) in [2] causes the FPGA behave unevenly when used for DR because the RSs in the FPGA routing can be affected by both boosting and reverse boosting effects caused by parasitic capacitances (C_p in Fig. 3). As visualized in Figs. 3a–b, boosting will occur only after DR has been executed on a low input on IN to *M1*. After *M0* has been turned off (thereby making *N0* a floating node), the voltage on *N0* will increase with ΔV for a high input (Fig. 3a) and remain unchanged for a low input (Fig. 3b). However, if DR is executed when the input is high, the voltage on *N0* will be unaffected for a high input on IN (Fig. 3c) and decrease with ΔV for a low input (Fig. 3d) (reverse boosting) after *M0* is turned off.

The simplest solution to this problem would be to set IN low during DR to constantly achieve boosting. However, without alteration of the RS design, the FPGA's execution will be interrupted because the contexts that store configuration data all use the same input.

Design of the Modified Programmable Routing Switch

Considering the abovementioned reasons, we propose a modified RS (simulated with 1.0-µm CAAC-IGZO FETs and 0.5-µm CMOS FETs), to preserve the low-leakage ability of the previous design, to utilize the enhanced switching resulting from the boosting, and to avoid reverse boosting effects and interruption of the FPGA's execution (Fig. 4). In the modified RS, a node *N1* is created on the input of the reference via the addition of transistors *M3* and *M4*. When WL0 is set high, *M4* and *M0* will be turned on, which will pull *N1* to ground and enable any content on the bit line (BL) to be written to the memory without interfering with the active context. Unnecessary leakage to ground is maintained extremely low because transistor *M4* is a CAAC-IGZO FET.

The gates of the NMOS pass-transistors *M2* and *M3* are connected to the context selector (context_0) and will be turned off as long as another context is used. When context_0 is in use, WL0 is always maintained low and connections between elements, such as programmable logic elements (PLE) and I/Os, will be decided depending on what has been written to the memory.

3. Performance and Area Usage for the Modified Programmable Routing Switch

Setup of Dynamic Reconfiguration Simulation

To show that DR is successful when using the modified RS in the routing of the MC-FPGA in [2] and to investigate how boosting effects influence its behavior, we simulated the MC-FPGA with two contexts (the PLE context switches were maintained as in [2]). The initial configuration of the FPGA is a shifter (Fig. 5a) held in context 0 that either shifts a single pulse (Fig. 5c) or a step signal (Fig. 5d). Context 1 is dynamically reconfigured to realize a ring oscillator (RO) (Fig. 5b) while the shifter is in use. The pulse shifter will generate a low input to the RSs during DR while the step shifter generates a high input.

Simulation Results

In Fig. 6, the voltage levels on *N0* are shown during the execution of the FPGA using either the reference or the modified RS. The results in Fig. 6a, clearly demonstrate how the voltage level on *N0* differs for the reference RS depending on whether the shifter used a pulse or a step signal during DR. The lowered voltage level on *N0* results in degraded switching properties for the RS, which causes a slower output frequency from the RO. However, for the modified RS (Fig. 6b), *N0* appears unchanged, irrespective of the shifter's input signal; this result indicates that the modified RS, in contrast to the reference, can handle DR without being influenced by the active context's execution.

Design Trade-offs and Layout

The addition of another pass gate to the RS increases the propagation delay and adds another voltage drop to the circuit. To overcome these drawbacks, we can overdrive the context signal to improve the strength of the output signal and decrease propagation delay. Fig. 7 shows how the oscillation period changes for the reference and the modified RS when the context signal is overdriven in a five-stage RO, in which multiplexers have been placed before the RSs to create a low/high signal during DR of context 1. As evident from the figure, the modified RS have a longer period than that of the reference when DR occurs during low input but a much shorter period than that of the reference when DR occurs during high input.

Furthermore, the addition of transistors to a design usually leads to larger area usage; however, the modified RS required no more area than that required for the reference RS because it could efficiently utilize the CAAC-IGZO layers in combination with the CMOS layers. Fig. 8 shows the layout for (*a*) all layers stacked, (*b*) the silicon gate and source layers, (*c*) the CAAC-IGZO layers, and (*d*) the supply lines. The modified RS uses the same signals as the reference and do not add new interconnects to the FPGA routing.

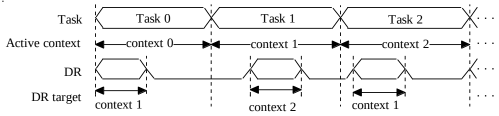

Fig. 1. Task execution and DR in an MC-FPGA.

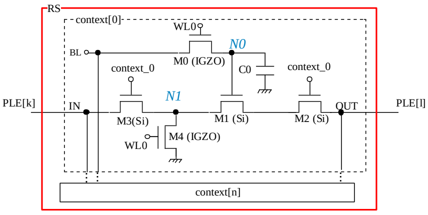

Fig. 4. Schematic of the modified RS.

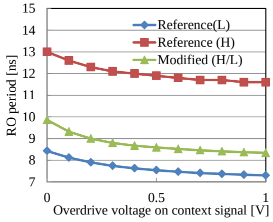

Fig. 7. Period of the RO for the modified and reference RS, depending on the overdrive voltage applied to the context signal.

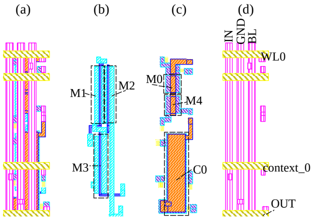

Fig. 8. Layout of the modified RS.

3. Conclusions

We designed a non-volatile MC-FPGA that performs DR. The FPGA uses a new RS based on a crystalline oxide semiconductor FET/CMOS FET hybrid technology to ensure uniform performance without interrupting the execution of the FPGA. The RS creates a low-input environment during DR to achieve boosting on its memory that improves its switching capability.

References

[1] D. Choi, K. Choi, and J. D. Villasenor: *IEEE Trans. VLSI Syst.* **16** (2008) 874.

[2] M. Kozuma *et al*., *Jpn. J. Appl. Phys.* **53** (2014) 04EE12

[3] Y.Y Liauw *et al*., *ISSCC Dig. Tech. Papers* (2012) 406.

[4] D. Suzuki *et al., Ext. Abstr. Solid State Devices and Materials* (2010) 1146.

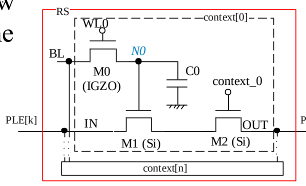

Fig. 2. Schematic of the reference RS's context.

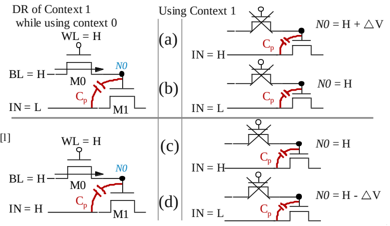

Fig. 3. Boosting effects on node *N0***.**

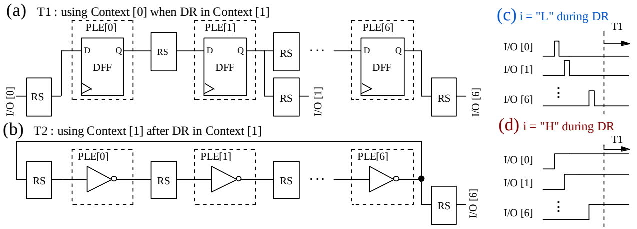
(a) T1 : using Context [0] when DR in Context [1]
PLE[0]
PLE[1]
PLE[6]
D
Q
DFF
RS
I/O [0]
I/O [1]
I/O [6]
(b) T2 : using Context [1] after DR in Context [1]
(c) i = "L" during DR
(d) i = "H" during DR
T1

Fig. 5. The shifter circuit (a) and the RO circuit (b). Subfigure (c) shows the signal for the pulse input to the shifter and (d) the step input.

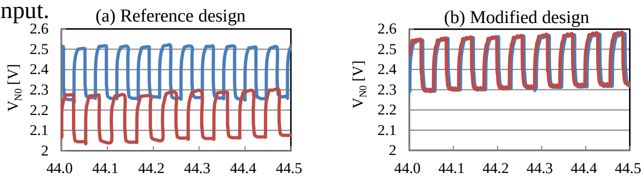

Fig. 6. Voltage level on *N0* for the reference (a) and the modified RS (b) after DR with either a low input (blue) or high input (red) on IN.