Architecture of a Multi-Context FPGA
Using a hybrid Multiple-Valued/Binary Context Switching Signal

Yoshihiro NAKATANI, Masanori HARIYAMA and Michitaka KAMEYAMA
Graduate School of Information Sciences, Tohoku University
Aoba 6-6-05, Aramaki, Aoba-ku, Sendai,
Miyagi, 980-8579, Japan
e-mail: {yoshi, hariyama, michi}@kameyama.ecei.tohoku.ac.jp

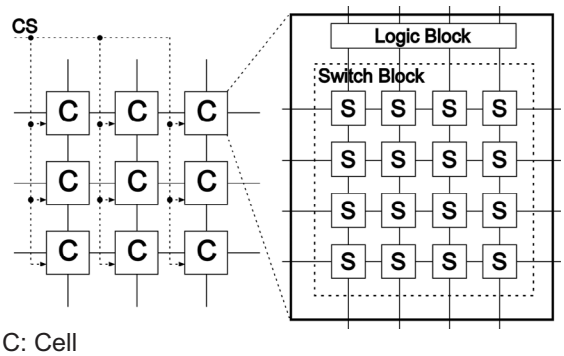
Abstract

Multi-context FPGAs have multiple memory bits per configuration bit forming configuration planes for fast switching between contexts. Large amount of memory causes significant overhead in area and power consumption. This paper presents two key technologies. The first is a floating-gate-MOS functional pass gate that merges storage and switching functions area-efficiently. The second is the use of a hybrid multiple-valued/binary context switching signal that eliminates redundancy of a conventional multi-context (MC) switch with high scalability. The transistor count of the proposed MC-switch is reduced to 7% in comparison with that of a SRAM-based one.

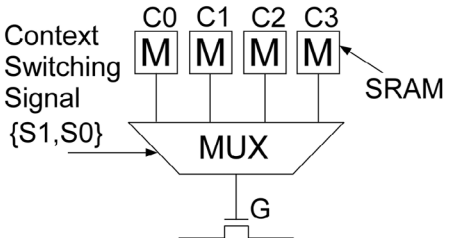
1. Introduction

Multi-context FPGAs (MC-FPGAs) have multiple memory bits per configuration bit forming configuration planes for fast switching contexts. However, the additional memory planes cause significant overhead in area and power consumption [1]. Figure 1 shows the overall structure of an MC-FPGA. Each cell consists of a programmable logic block and a programmable switch block. A switch block consists of multi-context switches (MC-switches). Figure 2 shows the structure of a conventional SRAM-based MC-switch for 4 contexts. The context switching signal (CSS) is broadcasted to all the MC-switches. Each MC-switch selects a configuration bit between pre-stored configuration bits according to CSS. The pass transistor is ON for G=1, and OFF for G=0. An MC-switch requires N SRAM bits for N contexts.

In order to reduce the overhead of configuration memory in MC-FPGAs, we employ a floating gate MOS functional pass gate (FGFP) that merges storage and switching function on a single floating gate MOS transistor (FGMOS)[2], [3]. The work [3] proposed an MC-switch with 4 FGMOSs that use a multiple-valued (MV) signal for context switching. In this MC-switch,



C: Cell
CSS: Context Switching Signal
S: Multi-context Switch
Fig. 1 Overall structure of an MC-FPGA.

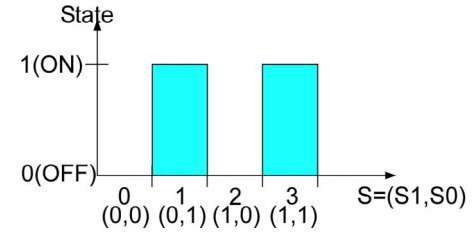


M: SRAM (Static Random Access Memory) bit
G: Configuration Bit
Fig. 2 Equivalent circuit of a conventional MC-switch (4 contexts).

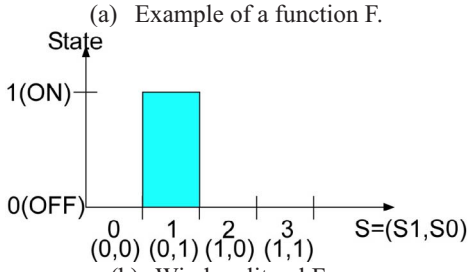
several pass transistors become ON redundantly for some configuration patterns. To eliminate this redundancy, a hybrid multiple-valued/binary signal is used for context switching. The proposed MC-switch has only 2 FGMOSs, each of which is exclusively ON. Although the proposed MC-switch requires more complex circuits for generating the context switching signal, they can be shared among several MC-switches, and its overhead is negligible.

The transistor count of the MC-switch is reduced to 7% and 50% in comparison with that of the SRAM-based MC-switch and the MC-switch using only MV-FGFPs, respectively.

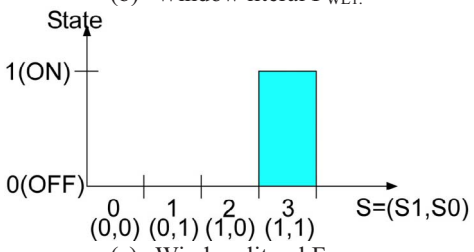
1-4244-0054-6/06/$20.00



(a) Example of a function F.

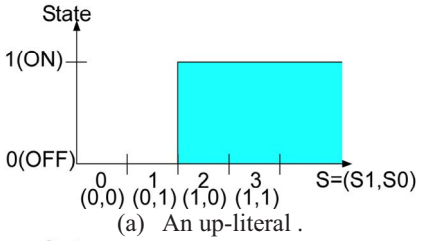


(b) Window literal $F_{WL1.}$

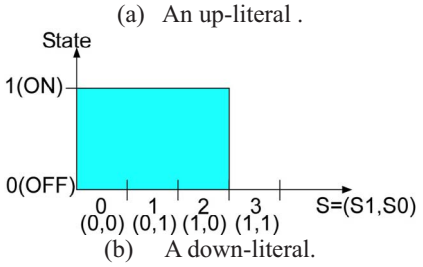


(c) Window literal $F_{WL2.}$

Fig. 3 Function of an MC-switch (4 contexts).



(a) An up-literal .



(b) A down-literal.

Fig. 4 Up-literal and a down-literal (4 contexts).

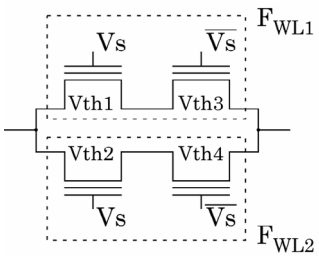


Fig. 5 Circuit of an MC-switch (4 contexts).

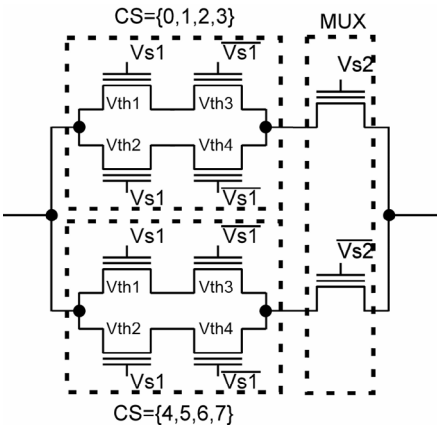


Fig. 6 Circuit of an MC-switch (8 contexts).

2. Previous work

This section describes the MC-switch using only a multiple-valued context switching signal (MV-CSS).

Figure 3(a) shows an example of the function F of the MC-switch that is only for MV-CSS = 1 and 3. The function F can be given by "OR-ing" the functions F_{WL1} (Fig.3(b)) and F_{WL2} (Fig.3(c)), each of which is called a window literal. Given S1 and S2 (S1 < S2), a window literal F_{WL} is defined as follows.

$$F_{WL}(S,S1,S2) = \begin{cases} 1 & S1 \le S \le S2 \\ 0 & otherwise \end{cases}$$

A window literal can be "AND-ing" the function F_{UL} and F_{DL} called "up-literal" and "down-literal" respectively. An up-literal is a monotone increasing function as shown in Fig. 4(a). Given the threshold value T, an up-literal $F_{UL}(S,T)$ is given by

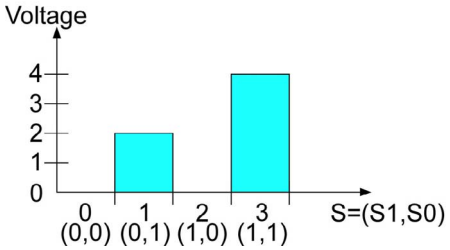
$$F_{UL}(S,T) = \begin{cases} 1 & T \le S \\ 0 & otherwise \end{cases}$$

A down-literal is a monotone decreasing function as shown in Fig. 4(b). Given the threshold value T, a down-literal $F_{DL}(S,T)$ is given by

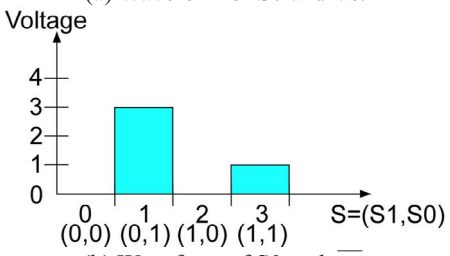
$$F_{DL}(S,T) = \begin{cases} 1 & S \le T \\ 0 & otherwise \end{cases}$$

The threshold value of an up-literal or a down-literal is programmed by injecting a controlled amount of electrons into the floating gate.

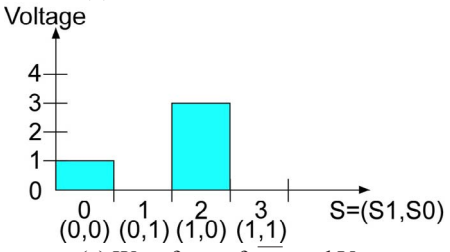
Each of an up-literal and a down-literal can be implemented only by a single FGFP [2]. Therefore, a window literal is implemented by series-connected FGMOSs for wired-AND. An MC-switch is implemented by connecting the circuit for window literal parallel for wired-OR as shown in Fig.5. Four FGMOSs are sufficient for 4 contexts since the function of an MC-switch includes 2 window literals at most.



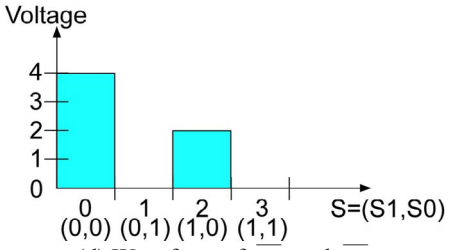
(a) Waveform of S0 and Vs.



(b) Waveform of S0 and $\overline{Vs}$.



(c) Waveform of $\overline{S0}$ and Vs.



(d) Waveform of $\overline{S0}$ and $\overline{Vs}$.

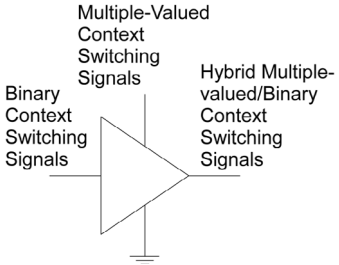
Fig. 7 Waveforms of MV/B-CSSs (4 contexts).

Let us consider implementing more than 4 contexts. Figure 6 shows the MC-switch for 8 contexts. It consists of 2 MC-switches for 4 contexts and an additional multiplexer (MUX).

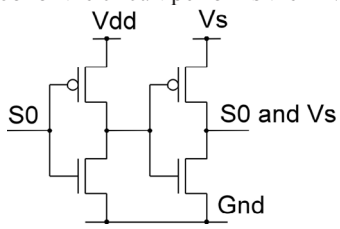
The window-literal-based implementation of the MC-switch has redundancy. For example, it requires 4 FGMOSs even when the function of the MC-switch is a single window literal.

3. Architecture and Evaluation

Figure 7 shows an example of a hybrid multiple-valued/ binary context switching signal (MV/B-CSS). Figure 8 shows the circuit to generate the MV/B-CSS. The circuit of generating MV/B-CSSs is small and that is broadcasted all MC-switches. Overhead of using MV/B-



(a) Symbol of the circuit performs the MV/B-CSS.



(b) Circuit performs the MV/B-CSS.

Fig. 8 Circuit which makes a MV/B-CSS.

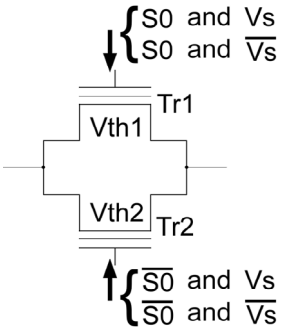


Fig. 9 Structure of MC-switch using MV/B-CSSs.

CSSs is small. The output is same as the MV-CSS when the binary CSS is 1. Otherwise, the output is 0. The context ID CSS = {0,1,2,3} is represented by a voltage Vs = {1,2,3,4}. The reason why CSS = 0 corresponds to Vs = 1 is that (Vs and S0) and (Vs and $\overline{S0}$) make difference when CSS = 0. The inversion of Vs is given by $\overline{Vs}$= 5 – Vs. Five-valued signals are required to make a clear distinction between the 0-level of binary and that of multiple-valued.

Figure 9 shows the structure of an MC-switch using a MV/B-CSS. The FGMOS Tr1 works when S0 = 1 (Vs = 2, 4), and the gate input of Tr1 is selected from two signals. "S0 and Vs" and "S0 and $\overline{Vs}$". Similarly, the FGMOS Tr2 works when S0 = 0 (Vs = 1, 3), and the gate input of be Tr2 is selected from two signals: "$\overline{S0}$ and Vs" and "$\overline{S0}$ and $\overline{Vs}$".

Threshold operation for "AND-ing" the MV-CSS and the binary one implements the same function as "AND-ing" two window literals. Only 2 FGMOSs are sufficient for an MC-switch for 4 contexts if we do not consider transistors to select the control signal.

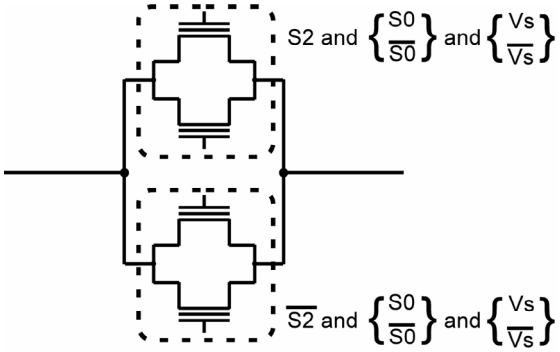


Fig. 10 Circuit of an MC-switch using MV/B-CSSs.

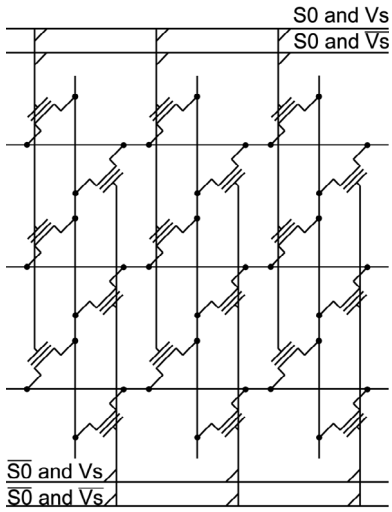


Fig. 11 Circuit of MC-SB using MV/B-CSSs.

Let us consider implementing more than 4 contexts. Figure 10 shows the MC-switch for 8 contexts using MV/B-CSS. It consists of 2 MC-switches for 4 contexts. It does not require any additional MUX unlike the MC-switch using only the MV-CSS. More context selection bits such as S2 are merged into the hybrid MV/B-CSS without any overhead.

We apply MC-switches to a Multi-context Switch Block (MC-SB). Figure 11 shows an MC-SB using MC-switches. The SB has the structure of cross-bar network. The MC-switch is used as a cross-point switch. For simplicity, the MC-SB has 3 columns and 3 rows. For a context, a single cross-point switch on each column and row is ON at most. We can map the possibly-ON cross-point switch on a column to the same MC-switch on the column for any context. As a result, N independent control signals are sufficient for an N×N MC-SB. In other words, a context switching signal is shared with MC-switches on the same column.

We compare the transistor counts of the proposed MC-switch and the proposed MC-SB with the SRAM-based MC-switch and MC-SB respectively. The transistor count of the proposed MC-switch is reduced to 7% of SRAM-based one (Table 1). Table 2 shows the comparison result when assuming that the switch block consists of 10×10 cross-point switches. The transistor count of the proposed MC-SB is reduced to 8% and 60% of that of the SRAM-based one and the FGFP-based one using only MV-CSS.

Table 1 The transistor count of an MC-switch.

	Transistor count
SRAM-based one	31
Only MV- FGFP-based one [2]	4
Proposed one	2

Table 2 Transistor count of MC-SB (10×10).

	Transistor count
SRAM-based one	3100
Only MV-FGFP-based one [2]	400
Proposed one	240

4. Conclusion

This paper presents the architecture of MC-switches using FGFPs and MV/B-CSS. The key technology is FGFPs and hybrid multiple-valued/binary context switching signals.

Using FGFPs for MC-switches, the area of an MC-switch is less than that of the SRAM-based one. Using the MV/B-CSS, the redundancy of a conventional MC-switch is eliminated with high scalability. The use of FGFPs will be efficient in static power consumption in comparison with the SRAM-based one because no supply voltage is required to keep the storage.

Acknowledgment

This work was supported by Industrial Technology Research Grant Program from New Energy and Industrial Technology Development Organization (NEDO) of Japan.

References

[1] S. Trimberger, D. Carberry, A. Johnson and J. Wong, “A Time-Multiplexed FPGA,” *FCCM*, pp.22-28 (1997)

[2] T. Hanyu, M. Kameyama, “Multiple-Valued Logic-in-Memory VLSI Architecture Based on Floating-Gate-MOS Pass-Transistor Logic,” *IEICE Trans. Electron.*, Vol.E82-C, No.9 (1999)

[3] M. Hariyama, W. Chong, S. Ogata and M. Kameyama, “Novel Switch Block Architecture Using Non-Volatile Functional Pass-gate for Multi-Context FPGAs,” *ISVLSI*, pp.46-50 (2005)