A Novel Hybrid Storage Architecture for Nonvolatile FPGA

Zewei Li * Yongpan Liu* Huazhong Yang*
Dept. of Electronic Engineering, Tsinghua University, Beijing, 100084, China*
{ypliu,yanghz}@tsinghua.edu.cn*

***Abstract*—High leakage power becomes an important factor hindering the deployment of FPGA (Field Programmable Gate Array) in portable devices. Emerging nonvolatile memory technologies can keep data with zero standby power and are promising for configurable memory in low power FPGAs. However, simple replacement of SRAM (Static Random Access Memory) with nonvolatile memory may increase tile area and delay significantly. This paper proposes an area efficient hybrid storage architecture for FeRAM(Ferroelectric Random Access Memory) based nonvolatile FPGA. We also discuss the tradeoff between tile area and configuration time in the hybrid storage system. The proposed design reduces the silicon area by 7.2 times compared with the previous FeRAM based FPGA and makes the nonvolatile tile area similar to the value in volatile SRAM based FPGAs.**

I. INTRODUCTION

FPGA is widely used in digital signal processing domain as important programmable devices. With the continuous scaling down of semiconductor technology, leakage power has become a critical factor hindering the deployment of FPGAs in low power applications. Most of mainstream FPGAs adopt SRAM for data and configuration storage, which leads to high leakage power. What's more, the configuration data stored in SRAM will be lost when FPGA is power off. An offchip nonvolatile memory is needed to reconfigure the FPGA after each power on. However, the reconfiguration time is usually high due to the limited bandwidth between offchip memory and SRAM. On the contrary, nonvolatile memory(NVM) can keep the data when power off. We can replace SRAM in FPGA with NVM to keep the configuration data. What's more, the computation data can also be backup in NVMs for future usage. As NVM can provide a zero-standby-power solution for FPGA, it is promising to investigate how to design nonvolatile FPGA with NVMs.

There are two different ways to implement nonvolatile FPGAs: 1) NVM cells are distributed in each tile to replace SRAM cells([1], [2], [3], [4]). In this architecture, the FPGA is configured in parallel and the wake-up time is very short. However the tile area may greatly increase due to the large area of NVM cells. Koga observed that the area of FeRAM based nonvolatile FPGA is 7.6 times larger than that of conventional FPGA [5]. 2) one NVM block is embedded in the FPGA to store configuration data and initializes FPGA during power up(e.g., LatticeXP FPGA). In this way, the NVM block has little effects on the tile area and signal delay, but the configuration time is long because configuration data is serially loaded into each tile. Pan analyzed the tile area and signal delay under those two architectures [6]. However, none of previous work discusses the design tradeoff between area and configuration time under a hybrid storage architecture.

This paper proposes a novel hybrid storage architecture for nonvolatile FPGAs. The hybrid architecture adopts partial distributed embedded FeRAMs to make a tradeoff between tile area and configuration time. We modified the CACTI tool to accurately model the area and time delay of FeRAM. Our evaluation demonstrates that the hybrid architecture reduces the tile area by 7.2 time compared with Koga's work and the configuration time is within several microseconds, which implies a good tradeoff between tile area and configuration time.

This work was supported in part by the NSFC under grant 61271269 and High-Tech Research and Development (863) Program under contract 2013AA01320, in part by the National Science and Technology Major Project under Contract 2010ZX03006-003-01, and in part by the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions under contract YETP0102.

II. HYBRID STORAGE ARCHITECTURE

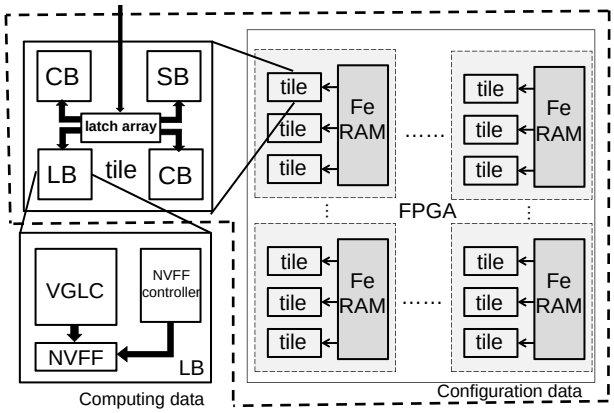

Fig. 1. Proposed storage architecture for FPGA

In FeRAM based FPGAs, the location of FeRAM has a significant effect on the tile area and configuration time. The fully distributed architecture causes large tile area and long interconnection delay, where each tile contains distributed FeRAMs. However, the centralized architecture includes a large embedded FeRAM block, which needs a long time to program. We propose the partial distributed architecture in Fig. 1. The architecture contains three levels: 1)FeRAM blocks for configurable data. Many FeRAM blocks are distributed in FPGA, which contain the configuration information of several adjacent tiles. 2)Latch array in each tile. The latch array controls the logic and interconnect functions in each tile. 3) NVFF in the logic blocks. NVFFs use ferroelectric capacitors to backup computing data in registers. The proposed FPGA adopts an island style architecture and VGLC(Variable Grain Logic Cell) logic blocks.

a) FeRAM block: The FeRAM cell has the 1T1C structure and sense amplifiers are used to fetch the data stored in ferroelectric capacitors. A column of FeRAM cells share one sense amplifier. To analyze the area and access time of FeRAM, we create a NVM model using CACTI, where the area and time delay parameters are extracted based on similar methods in NVSim [7].

b) Latch array: Latch array is made up of latches in each tile. It stores the data to configure function and interconnection in a tile. A latch has an enable signal and an output signal to control a switching transistor. During initialization, configuration data in the corresponding FeRAM is loaded into the latch array in sequence. By distributing configuration data in several FeRAMs, the FPGA configuration can be done in parallel and configuration time is short and predictable.

c) NVFF: In the logic block, all flip-flops are replaced by nonvolatile ones [5]. NVFFs consist of a D flip-flop, a pair of ferroelectric capacitors(FeCaps) and a control circuit. FeCaps are used to backup data in the D flip-flop. The control circuit detects the rise or drop of supply voltage, and backup data to or recovery data from FeCaps when power failure or recovery happens. The FeCaps will not be accessed when FPGA operates normally. Therefore, NVFFs will not bring performance penalty.

The three-level memory hierarchy plays different roles. When FPGA is powered off, the configuration data in the latch array is lost. However, the configuration data in FeRAM is remained. At the same time, the control circuit in NVFFs detects the voltage drop and saves the computing data into FeCaps. When FPGA is powered on, we perform both reconfiguration and computing data recovery. The configuration data in FeRAM blocks is loaded into the latch arrays in parallel. Compared with SRAM based FPGAs, where all configuration data is loaded serially into each tile, the configuration time is greatly reduced. During data recovery, the latest computing data is recovered from NVFFs instead of loading data from offchip NVM. Therefore, the proposed hybrid storage architecture can achieve fast sleep and wakeup features. However, there is a design tradeoff to decide the parameters of the hybrid architecture. As experimental results have shown, the number and capacity of FeRAM blocks affect both FPGA area and configuration time and we need perform the architecture exploration in the experiments.

III. EXPERIMENT VALUATION

In this section, we first demonstrate configuration and data recovery in the nonvolatile FPGA. Furthermore, we discuss the area-configuration tradeoff in the proposed design. All designs are evaluated based on a $0.13\mu m$ Ferroelectric-CMOS process technology. We use Synposys DC compiler and simulator to calculate area and delay of logic blocks.

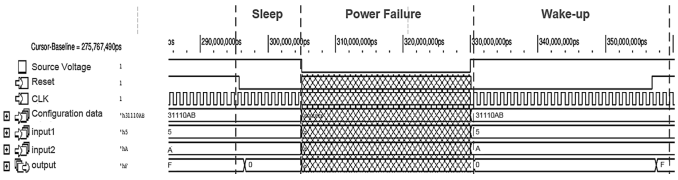

Fig. 2. Simulation of the non-volatility of logic block

Fig. 2 show the recovery progress of configuration data after power failure and the reconfiguration time is within several microseconds, because the propose hybrid storage architecture can load data in parallel.

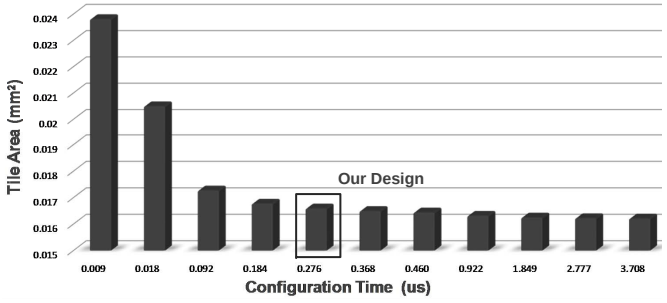

Fig. 3. The effect of FeRAM number on tile area and configuration time

What's more, we investigate the area-configuration tradeoff in the partial distributed architecture. We assume the embedded FeRAM blocks are uniformly distributed in FPGA. Given a total FeRAM capacity, the number of FeRAM blocks plays an important role in the tradeoff analysis. As the number of FeRAMs increases, the configuration time is reduced due to higher parallel degree. However, the tile area increases as well as the intra-tile delay. It is because the distributed FeRAM blocks will occupy some extra tile area and prolong the interconnect delay. The simulation results in Fig. 3 show the tradeoff between equivalent tile area and configuration time. The equivalent tile area consists of two parts: tile area and FeRAM area. The tile area is estimated by Synopsys DC compiler and the FeRAM area is calculated based on the CACTI area model. In Fig. 3, our design is marked with a black box. The fully distributed architecture (not marked) lies in the left of the graph and causes a large tile area. While the centralized architecture (not marked) lies in the right of the graph and causes long configuration delay. As we can see, the equivalent tile area is around 0.0167 mm^2. While the area of Koga's work is 0.12 mm^2 and the area of conventional tile with 4-clustered 4-LUT is $0.016mm^2$ [5]. Our design improves the circuit density by 7.2 times compared with Koga's work and is very close to that of conventional SRAM based FPGA. The proposed storage architecture also provides an area-configuration tradeoff.

IV. CONCLUSIONS

This paper proposes a hybrid storage architecture with partial distributed FeRAM blocks to tradeoff tile area with configuration time. Embedded FeRAM blocks store the configuration data of adjacent tiles. The latch array controls the logic blocks and the routing switches directly. The NVFFs in each tile are used to backup the computing data. We synthesize the tile and create a area-delay FeRAM model. After that, we validate the function of the nonvolatile FPGA and analyze tile area and configuration time under different FeRAM sizes. Compared with the previous work, the tile area in the proposed design decreases by 7.2 times and the configuration time is within several microseconds.

REFERENCES

[1] W. Zhao, E. Belhaire, C. Chappert, and P. Mazoyer, "Spin transfer torque (stt)-mram–based runtime reconfiguration fpga circuit," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 9, no. 2, p. 14, 2009.

[2] S. Paul, S. Mukhopadhyay, and S. Bhunia, "A circuit and architecture codesign approach for a hybrid cmos–sttram nonvolatile fpga," *Nanotechnology, IEEE Transactions on*, vol. 10, no. 3, pp. 385–394, 2011.

[3] Y. Y. Liauw, Z. Zhang, W. Kim, A. Gamal, and S. S. Wong, "Nonvolatile 3d-fpga with monolithically stacked rram-based configuration memory," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*. IEEE, 2012, pp. 406–408.

[4] Y. Chen, J. Zhao, and Y. Xie, "3d-nonfar: three-dimensional non-volatile fpga architecture using phase change memory," in *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design*. ACM, 2010, pp. 55–60.

[5] M. Koga, M. Iida, M. Amagasaki, Y. Ichida, M. Saji, J. Iida, and T. Sueyoshi, "A power-gatable reconfigurable logic chip with feram cells," in *TENCON 2010-2010 IEEE Region 10 Conference*. IEEE, 2010, pp. 317–322.

[6] Y. Pan, Y. Li, H. Sun, W. Xu, N. Zheng, and T. Zhang, "Exploring the use of emerging nonvolatile memory technologies in future fpgas," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 21, no. 4, pp. 771–775, 2013.

[7] X. Dong, C. Xu, Y. Xie, and N. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 31, no. 7, pp. 994–1007, July 2012.