NEMS Interconnect Relays for Reconfigurable Circuits

Lars Tatum
University of California
lpt@berkeley.edu

Erik Anderson
University of California
efa@eecs.berkeley.edu

March 31, 2020

Abstract

FPGAs, like all reconfigurable systems, exhibit significantly worse performance than their 1:1 ASIC counterparts. This disparity is primarily due to the extremely flexible yet power-hungry and physically large programmable routing network inside the FPGA. To alleviate this problem, several versions of CMOS compatible Back-End-Of-Line (BEOL) mechanical relays have been proposed to create denser and more efficient programmable interconnects. In this paper, we propose to analyze the performance and area of two such relay FPGA (rFPGA) architectures relative to the traditional CMOS-only FPGA. Our designs will use ASAP7 [1] a predictive 7nm PDK, to realistically gauge the benefits and drawbacks of rFPGAs in the modern semiconductor landscape.

I. Introduction

Field Programmable Gate Arrays (FPGAs) have long been used to enable rapid prototyping of digital systems pre-tapeout of Application Specific Integrated Circuits (ASICs)[2]. The framework they provide for high-performance reconfigurable digital systems is now proving useful for end-use applications including ML-acceleration, cryptocurrency mining, high-frequency stock trading, and software-defined radios. FPGA demand is predicted to continue rising as these protoyping and end-use markets increase in demand.

To enable this flexibility, FPGA-based systems are much more demanding than their ASIC counterparts. FPGAs come with larger die area requirements, delay, and dynamic power consumption. These factors increase the cost of FPGA-based systems and lower their potential performance over ASIC-based systems. Recently, CMOS compatible devices have been developed that could mitigate these area/power/performance drawbacks. Nano-Electro-Mechanical-Systems (NEMS) provide a promising, easily-integratable way to further FPGA-based systems. But first, a look at where all this overhead comes from in FPGAs.

A. The Island-Style FPGA architecture

FPGAs general consist of a general logic fabric as well as application-specific components, such as block RAM, DSPs, clocking, and I/O blocks[2]. This project will tackle the general logic fabric, since it makes up the majority of the FPGA.

Island-Style logic fabric can be divided into two major components: Configurable Logic Blocks and Reconfigurable Interconnects. Configurable Logic Blocks (CLBs) are used to implement arbitrary combinational and sequential logic circuits. They consist of Lookup Tables, Flip Flops, and Multiplexers. Each CLB can be programmed to perform an arithmetic or state-based portion of a larger logic system. CLBs are connected together using reconfigurable interconnects. Reconfigurable Interconnects, made up of Connection Blocks and Switch Boxes, are the "breadboard" of the FPGA. They allow the portions of the digital system to communicate. In an Island-Style Ar-

chitecture, Connection Blocks connect CLBs to routing wires that run throughout the fabric. Those wires are then connected to each other via Switch Boxes. These routing elements are primarily made up of muxes and SRAM, a setup that is readily implementable using NEM relays.

These Reconfigurable Interconnects dominate the area/power overhead incurred by FPGAs [3]. In an ASIC, interconnects are simple, passive elements; circuits are connected using fixed metal lines paths above the transistor devices. To enable design flexibility, FPGAs add SRAM configuration storage as well as the mux circuitry required to connect blocks and wires. Not only does this additional circuitry incur a large power/delay overhead, it also must be implemented completely in the Front-End-Of-Line alongside logic transistors, adding significantly to die area.

What if these power/performance/area drawbacks could be mitigated to enhance FPGA-based systems? ASIC circuits are connected solely via Back-End-Of-Line (BEOL) Metal lines. If FPGAs could emulate this structure, much of the overhead incurred from reconfigurable interconnects could be eliminated.

II. Proposed Techniques

In this work, the integration of nanoelectromechanical relays as reconfigurable interconnects is proposed as a way to deal with the overhead incurred by FPGA-based systems.

A. NEM Relays

Nanoelectromechanical Relays are electrically actuated mechanical switches. Notably, they experience zero leakage, have the potential for lower on-state resistance than NMOS pass gate, and can be implemented in the BEOL in a CMOS-compatible process flow[4]. The major disadvantages of NEM relays are large mechanical switching delay (> 1ns)[5] and lower reliability with respect to transistors (10^9 - 10^{10} cycles). The two relay device architectures examined in this work are lateral "crab style" relays and vertical interconnect relays. Due to their hysteretic properties, they can be implemented to replace both the interconnect pass-gates and their corresponding SRAM cells.

The lateral "crab-style" relays are the result of many years of development at Stanford and UC Berkeley to create an efficient CMOS compatible NEM relay[4]. These relays are switched on/off using the gate to body voltage difference. When "off", no current flows across the air gap. When "on", current flows between the drain and source with low on-state resistance. With this design, the device can be body-biased to have an extremely small switching voltage, experimentally shown to be sub-50mV range.

The vertical relays are designed to be more compact and compatible with existing CMOS process flows. Rather than using deposited metal or polysilicon as the structural material, these vertical relays use the already-present metal vias/interconnects in a CMOS process flow[6]. These relays can be implemented as single-pole double-throw switches or simpler single-pole single-throw switches (like the lateral relays). Both have their own drawbacks and benefits. While the lateral relays involve a "custom" structure, these vertical ones are designed within the constraints of standard CMOS BEOL design rules; i.e, they use standard metal line/via structures, and dimensions are standard for the process node.

B. Relay Integration

In Connection Blocks and Switch Boxes, a combination of muxes and buffers are used to route Logic Blocks to wires and different routing wires together. Traditional CMOS muxes are implemented as two-stage serial pass transistor structures for minimal minimum area-delay product. For the relay based designs, these NMOS pass transistors (and corresponding configuration SRAM cells) are replaced with NEM relays. To reduce the number of series relays as well as the number of total relays, a one-stage NEM-based mux is preferable for reduced area and delay[7].

NEM relay based routing muxes can eliminate the need for configuration SRAM by implementing a CMOS half-select programing scheme[7]. Similar to the half-select scheme of

crossbar arrays, an array of row and column lines program the relays. The relay states are preserved by a constant holding voltage on the row lines and constant select voltage on the column lines. During configuration, an additional (positive or negative) select voltage is superimposed on on the row lines "to be programmed", and the column lines are grounded. This voltage is just enough to actuate the relay into the desired state, but not so much that catastrophic pull-in occurs. Since relays have no leakage current, the only power required to keep the relays in the "hold state" is the dynamic power used during configuration to charge the parasitic capacitance. Since state-of-the-art FPGAs consist of thousands of tiles, the half-select programming array is predicted to consume <5% of total die area.

III. Methods

SRAM-based island-style FPGAs represent the vast majority of commercial and academic FPGA implementations [8]. For this reason, this architecture will be used to construct three functionally equivalent FPGAs using the three different routing schemes discussed in Section II.

To create an island-style FPGA a basic *tile* must first be constructed. Each tile consists of one *configurable logic block* (CLB), two *connection boxes* (CBs), and one *switch box* (SB). The CLB contains N basic logic elements (BLEs) which each contain a K-input look-up-table (LUT) and a flip-flop. In addition, CLBs provide an intra-cluster routing network to connect the CLB I/O to the individual BLE I/O. The CBs connect the CLBs to their adjacent routing channels, while the SBs provide the connections between intersecting routing channels. It is common to fold the output portion of the CBs into the SB [7]. Thus, the same MUXes used for connecting routing channels can be enlarged to accomodate the additional CLB output connections.

To ensure that our results are consistent with previous works, we will create a very similar tile to the one described in [7]. This tile consists of 10 4-LUT BLEs w/ 22 input pins and 10 output pins. In order to simplify the design, the CBs in our tile will connect both the outputs and inputs of the CLB to the adjacent routing channel. The intra-CLB routing network will be fully populated, allowing for connection between any BLE input and CLB input as well as between any BLE output and CLB output. The SB flexibility will be set to 3 ($f_{SB} = 3$), allowing all incoming signals to the SB to be switched left, right, or straight through. The CB input and output flexibilities will be set to 0.2 ($f_{CB-in} = 0.2$) and 0.1 respectively ($f_{CB-out} = 0.1$). These flexibilities determine the proportion of signals within the adjacent routing channels that the CB can connect an input or output signal to. The channel width will be set to 104 wires ($W = 104$) with 80 of those wires representing the minimum channel width for CLBs with 10 outputs and length 4 routing segments ($W_{min} = 2 * 10 * 4 = 80$).

The CMOS-only FPGA will serve as a baseline for the NEMs designs. Because the programmable interconnect (PI) will be replaced with NEMs relays, the make up of our baseline's PI will significantly impact the perceived benefits. We will use simple NMOS pass-transistors to build the MUXes used for the PI and the LUTs. While transmission gates show promise for scaled nodes, commercial FPGAs continue to use pass-transistors with gate-boosting and internal buffering to achieve a slightly lower area-delay product [9].

The NEMs relays can be modeled as simple RC circuits when configured in the ON-state [7] [5] and can be ignored, because they are physically disconnected from the circuit, when in the OFF-state. Modeling the dynamics of the NEMs relays as they switch on and off is not as important as the steady-state behavior for this analysis. Switching time will only affect the FPGA configuration time, and thus as long as the relays are not prohibitively slow, which they have been shown not to be [5], then the dynamics of the switches can be ignored for our purposes.

Preliminary area analysis can be done by comparing each FPGA tile. It is important to account for the additional metal area used by

the relays as this will significantly impact the place-and-route quality due to higher routing congestion. Further analysis can be performed on the full FPGAs to determine the dissipated power and maximum operating frequency for each design for a given set of test circuits. This will require connecting an array of FPGA tiles and running placement, packing, and routing to extract realistic performance and utilization numbers. A popular open source tool known as the Verilog-to-Routing (VTR) project [10] will be used to perform this analysis.

IV. Hypothesis

The vertical 4T relays should show a significant area reduction compared to the lateral relays and CMOS-only FPGAs. However, due to the reduced routing resources caused by the use of additional metal layers in the vertical relays, we predict that the integrated performance results will counterbalance the benefits from the reduced lateral area. In both NEMS design cases, we expect the area and performance to improve compared to the CMOS-only designs.

V. Conclusion

We have reviewed several methods for creating BEOL NEMs relays in CMOS processes. Our proposed work will include the creation of 3 island-style FPGA tiles including NEMS supporting circuitry for direct comparison of area and performance between a traditional CMOS-only FPGA and two different styles of NEMS based FPGAs.

References

[1] L. T. Clark et al. "ASAP7: A 7-nm finFET predictive process design kit". en. In: *Microelectronics Journal* 53 (July 2016), pp. 105–115. ISSN: 0026-2692. DOI: 10.1016/j.mejo.2016.04.006. URL: http://www.sciencedirect.com/science/article/pii/S002626921630026X (visited on 04/01/2020).

[2] I. Kuon, R. Tessier, and J. Rose. *FPGA Architecture: Survey and Challenges*. 2008.

[3] A. Corporation. "White Paper 40-nm FPGA Power Management and Advantages". In: December 2008 (2008), pp. 1–15.

[4] A. Peschot, C. Qian, and T.-j. K. Liu. "Nanoelectromechanical Switches for Low-Power Digital Computing". In: (2015), pp. 1046–1065. DOI: 10.3390/mi6081046.

[5] F. Chen et al. "Integrated circuit design with NEM relays". en. In: *2008 IEEE/ACM International Conference on Computer-Aided Design*. San Jose, CA, USA: IEEE, Nov. 2008, pp. 750–757. ISBN: 978-1-4244-2819-9. DOI: 10.1109/ICCAD.2008.4681660. URL: http://ieeexplore.ieee.org/document/4681660/ (visited on 03/15/2020).

[6] U. Sikder et al. "Back-End-of-Line Nano-Electro-Mechanical Switches for Reconfigurable Interconnects". In: 41.4 (2020), pp. 625–628.

[7] C. Chen et al. "Efficient FPGAs using nanoelectromechanical relays". In: *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*. FPGA '10. Monterey, California, USA: Association for Computing Machinery, Feb. 2010, pp. 273–282. ISBN: 978-1-60558-911-4. DOI: 10.1145/1723112.1723158. URL: https://doi.org/10.1145/1723112.1723158 (visited on 03/10/2020).

[8] U. Farooq, Z. Marrakchi, and H. Mehrez. "FPGA Architectures: An Overview". en. In: *Tree-based Heterogeneous FPGA Architectures*. New York, NY: Springer New York, 2012, pp. 7–48. ISBN: 978-1-4614-3593-8 978-1-4614-3594-5. DOI: 10.1007/978-1-4614-3594-5_2. URL: http://link.springer.com/10.1007/978-1-4614-3594-5_2 (visited on 03/10/2020).

[9] C. Chiasson and V. Betz. "Should FPGAS abandon the pass-gate?" In: *2013 23rd International Conference on Field programmable Logic and Applications*. ISSN: 1946-1488. Sept. 2013, pp. 1–8. DOI: 10.1109/FPL.2013.6645511.

[10] K. E. Murray et al. "VTR 8: High Performance CAD and Customizable FPGA Architecture Modelling". In: *ACM Trans. Reconfigurable Technol. Syst.* (2020).