

A Novel FPGA Design with Hybrid LUT /MUX Architecture

E.Ganesan, V.Sakthivel
PG Scholar, Assistant Prof
Paavai Engineering College, Namakkal

Abstract:

Field programmable gate arrays (FPGAs) are increasingly used as the computing platform for fast and energy efficient execution of recognition, mining, and search applications. Approximate computing is one promising method for achieving energy efficiency. Compared with most prior works on approximate computing, which target approximate processors and arithmetic blocks. Hybrid configurable logic block architectures for field programmable gate arrays that contain a mixture of lookup tables and hardened multiplexers are evaluated toward the goal of higher logic density and area reduction.

I. INTRODUCTION

Field Programmable Gate Arrays (FPGAs) are a convenient choice for low volume production as they are easy to design and program in a short time. However the reconfigurability of FPGAs renders them much larger, slower and more power consuming than their ASIC (Application Specific Integrated Circuits) counterparts [1]. ASICs, on the other hand, have a higher non-recurring engineering (NRE) cost and higher time to market. However, this issue is addressed by the introduction of Structured-ASICs which comprise of an array of optimized logic elements that can implement desired functionality by making changes to a few upper mask layers [2] [3]. FPGA vendors like Altera and Xilinx have also started making provision for migrating FPGA applications to Structured-ASIC. In this regard Altera has proposed a clean migration methodology from FPGA to Structured-ASIC while ensuring equivalence verification [4]. But, an FPGA completely loses its flexibility after its migration to Structured-ASIC. An ASIF, on the other hand, is reduced from FPGA like Structured ASIC but it retains enough flexibility to implement a set of predetermined applications. A mesh-based ASIF was initially presented in [5] where authors have shown that for a set of predetermined applications an ASIF is 81.5% smaller than a unidirectional mesh-based FPGA.

The MUX-based logic blocks for the FPGAs have seen success in early commercial architectures, such as the Actel ACT-1/2/3 architectures, and efficient mapping to these structures has been studied

[12] in the early 1990s. However, their use in commercial chips has waned, perhaps partly due to the ease with which logic functions can be mapped into LUTs, simplifying the entire computer aided design (CAD) flow. Nevertheless, it is widely understood that the LUTs are inefficient at implementing MUXs, and that MUXs are frequently used in logic circuits. To underscore the inefficiency of LUTs implementing MUXs, consider that a six input LUT (6-LUT) is essentially a 64 to-1 MUX (to select 1 of 64 truth-table rows) and 64-SRAM configuration cells, yet it can only realize a 4-to-1 MUX (4 data + 2 select = 6 inputs).

In this paper, we present a six-input LE based on a 4-to-1 MUX, MUX4, that can realize a subset of six-input Boolean logic functions, and a new hybrid complex logic block (CLB) that contains a mixture of MUX4s and 6-LUTs. A Hybrid configurable logic block architectures for field programmable gate arrays that contain a mixture of lookup tables and hardened multiplexers are evaluated toward the goal of higher logic density and area reduction.

The rest of this brief introduces the FPGAs in section I. Next the related works and an approximate computing methodology for FPGA-based design are considered in Section II. Then, in Section III, the proposed is presented. Section IV and V presents a experimental Results and performance analysis to illustrate the effectiveness of the hardware security approach. Finally, the conclusions are summarized in Section V.

II. RELATED WORKS AND AN APPROXIMATE COMPUTING METHODOLOGY

The memoization architecture generator contains a memorization wrapper generator that generates the architectural blocks needed for memoization. It takes similarity measure, threshold (Th), user inputs, and user selection, which are present in the configuration file, as the input. The similarity measure is defined by the user to compare two temporally spaced inputs; threshold (Th) is the threshold value within which the result of similarity measurement must exist.

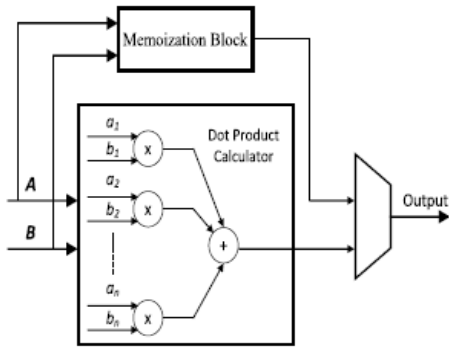


Fig.1

The memoization method that the user selects is taken as an input via user selection parameter. Depending on this selection, RTL wrapper is generated for either the static memorization or the dynamic memorization. The user supplied information, such as similarity measure, is best decided by the user as he/she is more aware of the domain of application to which the proposed memoization-based approximate computing is being applied. Here, the signal is used only in dynamic memoization wrapper, while the others are common signals in both the architectures.

III. HYBRID LUT/MULTIPLEXER FPGA LOGIC ARCHITECTURES

Multiple hybrid configurable logic block architectures, both non-fracturable and fracturable with varying MUX:LUT logic element ratios are evaluated across two benchmark suites (VTR and CHStone) using a custom tool flow consisting of LegUp HLS, Odin-II front-end synthesis, AB logic synthesis and technology mapping, and VPR for packing, placement, routing, and architecture exploration. Technology mapping optimizations that target the proposed architectures are also implemented within ABC.

Fracturable 6-LUT

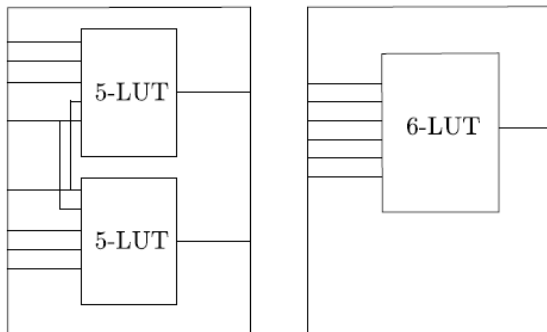


Fig.2 6-LUT that can be Fractured Into Two 5-Luts With Two Shared Inputs.

For the fracturable architecture, we consider an eight-input LE, closely matched with the adaptive logic module in recent Altera Stratix FPGA families. A 6-LUT that can be fractured into two 5-LUTs using eight inputs is shown in Fig.2. Two five-input functions can be mapped into this LE if two inputs are shared between the two functions. If no inputs are shared, two four-input functions can be mapped to each 5-LUT. For the MUX4 variant, Dual MUX4, we use two MUX4s within a single eight-input LE. In the configuration, the two MUX4s are wired to have dedicated select inputs and shared data inputs.

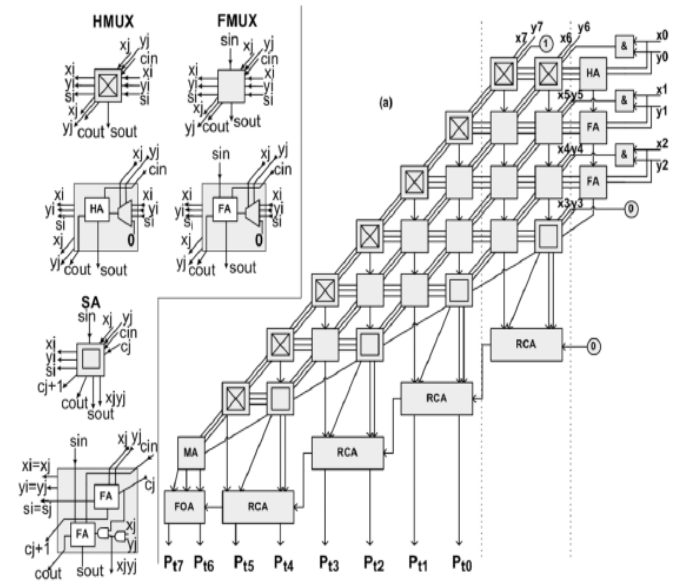


Fig.3 MUX Based Multiplier

We also propose a new error compensation circuit by using the dual group minor input correction vector to lower input correction vector compensation error. By modifying half adder addition like both sum and carry is one when input A and B equal to one. By constructing the error compensation circuit mainly from the “outer” partial products, the hardware complexity only increases slightly as the multiplier input bits increases.

IV. EXPERIMENTAL RESULTS

The proposed circuit are simulated and synthesized by using modelsim and xilinx12.1 which occurs low area than the existing. The experimental results are given in Table 1 and the simulation results of layout and the waveforms are shown in the fig.4 and fig.5. Then the RTL schematic of the proposed are shown in fig.5.

s.no	Parameter	Existing	Proposed
1	SLICE	15	11
2	LUT	27	21

Table.1 Comparison Table

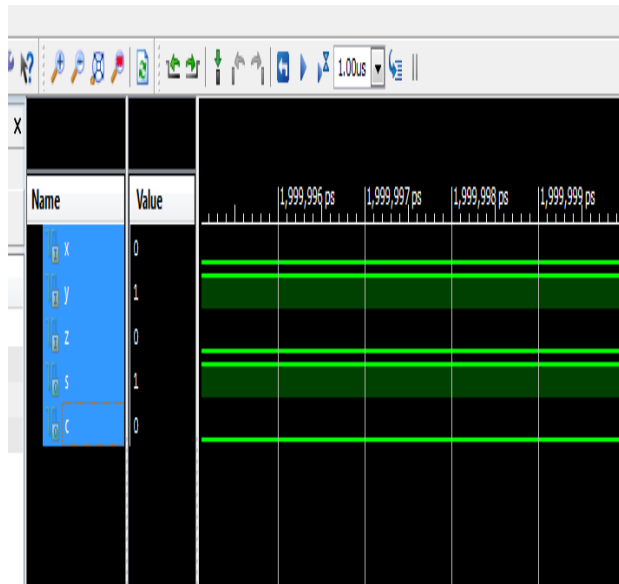


Fig.4 Simulation Results

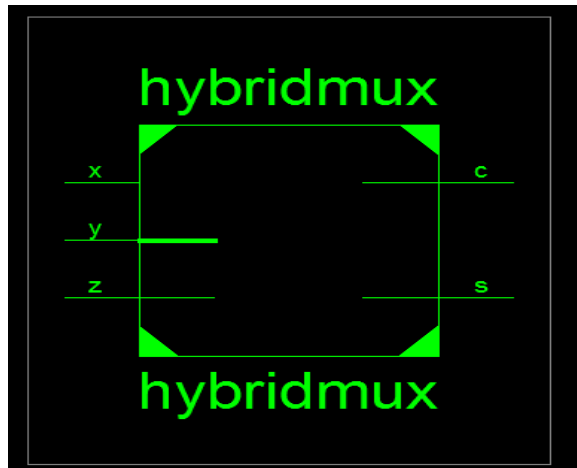


Fig.5 RTL Schematic

V. PERFORMANCE ANALYSIS

The performance of our proposed hybrid lut with the existing scheme based on the area and error compensation which was given in Fig. 5

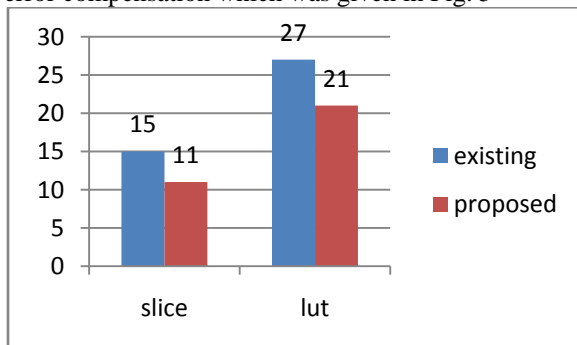


Fig.7 Performance Analysis of Existing and Proposed

VI. CONCLUSION

We have proposed a new hybrid CLB architecture containing MUX4 hard MUX elements and shown techniques for efficiently mapping to these architectures. new multiplexer based truncation scheme with lower average and mean square errors than existing methods. For both non fracturable and fracturable architectures, we see minimal impact on timing performance for the architectures with best area-efficiency. the addition of MUX4s to FPGA architectures minimally impact FMax and show potential for improving logic-density in nonfracturable architectures and modest potential for improving logic density in fracturable architectures.

REFERENCES

- [1] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," in Proceedings of the 2006 ACM/SIGDA 14th international symposium on Field programmable gate arrays. ACM New York, NY, USA, 2006, pp. 21–30.
- [2] K. Wu and Y. Tsai, "Structured ASIC, Evolution or Revolution," April 2004, pp. 103–106.
- [3] T. Okamoto, T. Kimoto, and N. Maeda, "Design Methodology and Tools for NEC Electronics Structured ASIC," April 2004, pp. 90–96.
- [4] J. Pistorius, M. Hutton, J. Schleicher, M. Iotov, E. Julias, and K. Tharmaligham, "Equivalence Verification of FPGA and Structured ASIC Implementations," August 2007, pp. 423–428.
- [5] H. Parvez, Z. Marrakchi, and H. Mehrez, "ASIF: Application Specific Inflexible FPGA," in International Conference on Field-Programmable Technology, 2009. FPT 2009, 2009, pp. 112–119.
- [6] V. Betz and J. Rose, "VPR: A New Packing Placement and Routing Tool for FPGA research," International Workshop on FPGA, pp. 213–22, 1997.
- [7] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and singledriver wires in fpga interconnect," in IEEE Conference on FPT, 2004, pp. 41–48.
- [8] Z. Marrakchi, H. Mrabet, E. Amouri, and H. Mehrez, "Efficient tree topology for fpga interconnect network," in ACM Great Lakes Symposium on VLSI, 2008, pp. 321–326.
- [9] L. Sterling and E. Shapiro, The Art of Prolog, 2nd ed. Cambridge, MA, USA: MIT Press, Mar. 1994.
- [10] A. K. Mishra, R. Barik, and S. Paul, "iACT: A software-hardware framework for understanding the scope of approximate computing," in Proc. Workshop Approx. Comput. Across Syst. Stack (WACAS), Salt Lake City, UT, USA, 2014.
- [11] C. Alvarez, J. Corbal, and M. Valero, "Fuzzy memoization for floatingpoint multimedia applications," IEEE Trans. Comput., vol. 54, no. 7, pp. 922–927, Jul. 2005.
- [12] C. Á. Martínez, J. C. S. Adrián, and M. V. Cortés, "Dynamic tolerance region computing for multimedia," IEEE Trans. Comput., vol. 61, no. 5, pp. 650–665, May 2012.