IJMTARC – VOLUME – V – ISSUE - 18 – JUNE , 2017

ISSN: 2320-1363

A NOVEL FPGA DESIGN WITH HYBRID LUT/MULTIPLEXER ARCHITECTURE

A SHANKAR NAYAK*,P V VARAPRASAD RAO**

*Pg Scholar, Department of ECE, SLC's Institute of Engineering and Technology,Hyderabad.

**AssosciateProfessor& HOD, Department Of ECE, SLC's Institute of Engineering and Technology, Hyderabad

ABSTRACT

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. Multiple hybrid configurable logic block architectures, both nonfracturable 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, ABC logic synthesis and technology mapping, and VPR for packing, placement, routing, and architecture exploration. VPR is used to model the new hybrid configurable logic block and verify post place and route implementation.. In this paper experimentally, we show that for nonfracturable architectures, without any mapper optimizations, we naturally save up to~8% area post place and route. For fracturable architectures, experiments show that only marginal gains are seen after place-and-route up to~2%. For both nonfracturable and fracturable architectures, we see minimal impact on timing performance for the architectures with best area-efficiency. Keywords— FPGA, Multiplexer logic element, Complex logic block, mapping technologies

I INTRODUCTION

A field-programmable gate array (FPGA) is a block of programmable logic that can implement multi-level logic functions. FPGAs are most commonly used as separate commodity chips that can be programmed to implement large functions. However, small blocks of FPGA logic can be useful components on-chip to allow the user of the chip to customize part of the chip's logical function. An FPGA block must implement both combinational logic functions and interconnect to be able to construct multi-level logic functions. There are several different technologies for programming FPGAs, but most logic processes are unlikely to implement antifuses or similar hard programming technologies. Throughout the history of field-programmable gate arrays (FPGAs), lookup tables (LUTs) have been the primary logic element (LE) used to realize combinational logic. A K-input LUT is generic and very flexible able to implement any K-input Boolean function. The use of LUTs simplifies technology mapping as the problem is reduced to a graph covering problem. However, an exponential area price is paid as larger LUTs are considered. The value of K between 4 and 6 is typically seen in industry and academia, and this INTERNATIONAL JOURNAL OF PROFESSIONAL ENGINEERING STUDIES Volume VIII /Issue 1 / DEC 2016 IJPRES range has been demonstrated to offer a good area/performance compromise. Recently, a number of other works have explored alternative FPGA LE architectures for performance improvement to close the large gap between FPGAs and application-specific integrated circuits (ASICs)

II LITERATURE REVIEW

Recent works have shown that the heterogeneous architectures and synthesis methods can have a significant impact on improving logic density and delay, narrowing the ASIC–FPGA gap. Works by Anderson and Wang with "gated" LUTs, then with asymmetric LUT LEs, show that the LUT elements present in commercial FPGAs provide unnecessary flexibility. Toward improved delay and area, the macrocell-based FPGA architectures have been proposed. These studies describe significant changes to the traditional FPGA architectures, whereas the changes proposed here build on architectures used in industry and academia. Similarly, and-inverter cones have been proposed as replacements for the LUTs, inspired by and-inverter graphs (AIGs). Purnaprajna and Ienne explored the possibility of repurposing the existing MUXs contained within the Xilinx Logic Slices. to this work, they use the ABC priority cut mapper as well as VPR for packing, place, and route. However, their work is primarily delaybased showing an average speed up of 16% using only ten of 19 VTR7 benchmarks. In this article, we study the technology mapping problem for a novel fieldprogrammable gate array (FPGA) architecture that is based onk-input single-output programmable logic array- (PLA-) like cells, or, k/m-macrocells. Each cell in this architecture can implement a single output function of up to k inputs and up to m product terms. We develop a very efficient technology mapping algorithm, km flow, for this new type of architecture. The experimental results show that our algorithm can achieve depth-optimality on almost all the test cases in a set of 16 Microelectronics Center of North Carolina (MCNC) benchmarks. Furthermore it is shown that on this set of benchmarks, with only a relatively small number of product terms ($m \leq k+3$), the k/m-macro cellbased FPGAs can achieve the same or similar mapping depth compared with the traditional kinput single-output lookup table- (k-LUT-) based FPGAs. We also investigate the total area and delay of k/m-macro cell-based FPGAs and compare them with those of the commonly used 4-LUT-based FPGAs. The experimental results show that k/m-macro cell-based FPGAs can outperform 4-LUT-based FPGAs in terms of both delay and area after placement and routing by VPR on this set of benchmarks. This paper presents experimental measurements of the differences between a 90- nm

CMOS field programmable gate array (FPGA) and 90-nm CMOS standard-cell application specific integrated circuits (ASICs) in terms of logic density, circuit speed, and power consumption for core logic. We are motivated to make these measurements to enable system designers to make better informed choices between these two media and to give insight to FPGA makers on the deficiencies to attack and, thereby, improve FPGAs. We describe the methodology by which the measurements were obtained and show that, for circuits containing only look-up table-based logic and flip-flops, the ratio of silicon area required to implement them in FPGAs and ASICs is on average 35. Modern FPGAs also contain "hard" blocks such as multiplier/accumulators and block memories. We find that these blocks reduce this average area gap significantly to as little as 18 for our benchmarks, and we estimate that extensive use of these hard blocks could potentially lower the gap to below five. The ratio of critical-path delay, from FPGA to ASIC, is roughly three to four with less influence from block memory and hard multipliers. The dynamic power consumption ratio is approximately 14 times and, with hard blocks, this gap generally becomes smaller. In this paper the new architectural proposals are routinely generated in both academia and industry. For FPGA's to continue to grow, it is important that these new architectural ideas are fairly and accurately evaluated, so that those worthy ideas can be included in future chips. Typically, this evaluation is done using experimentation. However, the use of experimentation is dangerous, since it requires making assumptions regarding the tools and architecture of the device in question. If these assumptions are not accurate, the conclusions from the experiments may not be meaningful. In this paper, we investigate the sensitivity of FPGA architectural conclusions to experimental variations. To make our study concrete, we evaluate the sensitivity of four previously published and well-known FPGA architectural results: lookup-table size, switch block topology, cluster size, and memory size. It is shown that these experiments are

III. PROPOSED ARCHITECTURES

MUX4: 4-to-1 Multiplexer Logic Element The MUX4 LE shown in

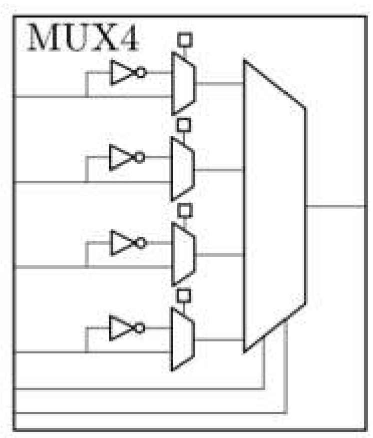


Fig. 1 consists of a 4-to-1 MUX

with optional inversion on its inputs that allow the realization of any {2, 3}-input function, some {4, 5}-input functions, and one 6- input function—a 4-to-1

IJMTARC – VOLUME – V – ISSUE - 18 – JUNE , 2017

ISSN: 2320-1363

MUX itself with optional inversion on the data inputs. A 4-to-1 MUX matches the input pin count of a 6-LUT, allowing for fair comparisons with respect to the connectivity and intracluster routing. Naturally, any two-input Boolean function can be easily implemented in the MUX4: the two function inputs can be tied to the select lines and the truth table values (logic-0 or logic-1) can be routed to the data inputs accordingly. Or alternately, a Shannon decomposition can be performed about one of the two variables—the variable can then feed a select input. The Shannon cofactors will contain at most one variable and can, therefore, be fed to the data inputs (the optional inversion may be needed). Fig. 1. MUX4 LE depicting optional data input inversions. Logic Elements, Fracturability, and MUX4-Based Variants Two families of architectures were created: 1) without fracturable LEs and 2) with fracturable LEs. In this paper, the fracturable LEs refer to an architectural element on which one or more logic functions can be optionally mapped. Nonfracturable LEs refer to an architectural element on which only one logic function is mapped. In the nonfracturable architectures, the MUX4 element shown in Fig. 1 is used together with nonfracturable 6-LUTs. This element shares the same number of inputs as a 6-LUT lending for fair comparison with respect to the input connectivity. Hybrid Complex Logic Block A variety of different architectures were considered—the first being a nonfracturable architecture. In the nonfracturable architecture, the CLB has 40 inputs and ten basic LEs (BLEs), with each BLE having six inputs and one output following empirical data in prior work. Fig. 2 m/ www.ieeemaster.com Mail : projects@lemenizinfotech.com shows this nonfracturable CLB architecture with BLEs that contain an optional register. We vary the ratio of MUX4s to LUTs within the ten element CLB from 1:9 to 5:5 MUX4s:6-LUTs. The MUX4 element is proposed to work in conjunction with 6-LUTs, creating a hybrid CLB with a mixture of 6-LUTs and MUX4s (or MUX4 variants).

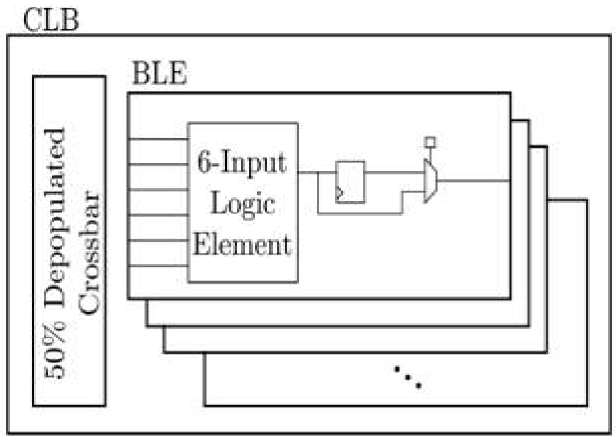


Fig. 2 shows the organization of our CLB and internal BLEs.

Fig. 2. Hybrid CLB with a 50% depopulated intra-CLB crossbar depicting BLE internals for a nonfracturable (one optional register and one output) architecture. m/ www.ieeemaster.com Mail : projects@lemenizinfotech.com Fig. 3. Hybrid CLB with a 50% depopulated intra-CLB crossbar depicting BLE internals for a fracturable (two optional registers and two outputs) architecture. For

fracturable architectures, the CLB has 80 inputs and ten BLEs, with each BLE having eight inputs and two outputs emulating an Altera Stratix Adaptive-LUT. The same sweep of MUX4 to LUT ratios was also performed.

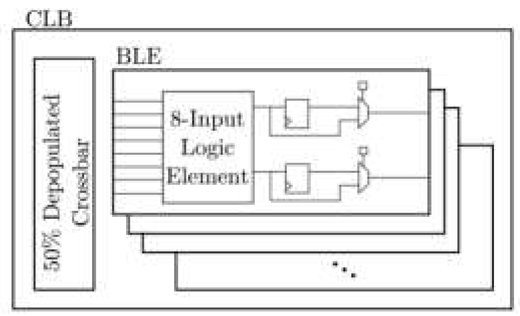
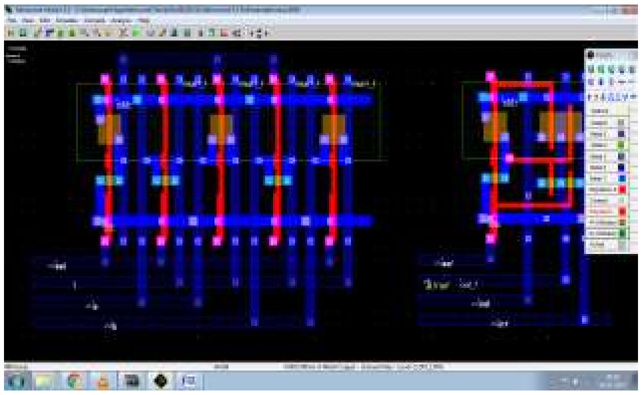
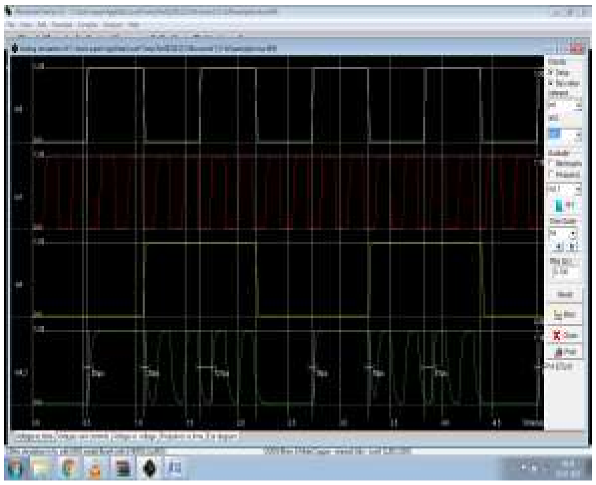


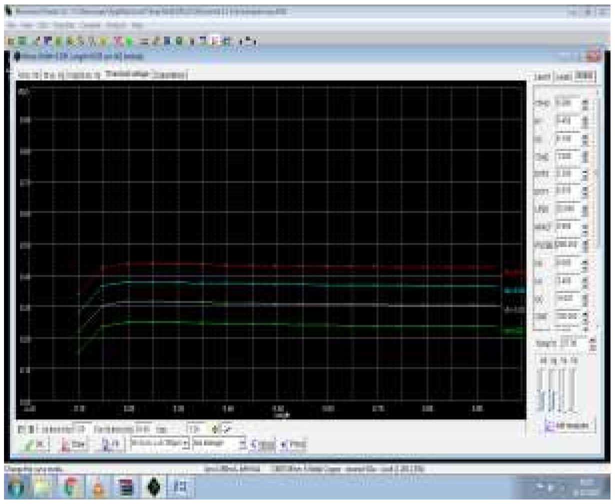
Fig. 3 shows the fracturable architecture with eight inputs to each BLE that contains two optional registers.

RESULTS:

Layout







IV CONCLUSION

We have proposed a new hybrid CLB architecture containingMUX4 hard MUX elements and shown techniquesfor efficiently mapping to these architectures. Weighting ofMUX4-embeddable functions with our MuxMap techniquecombined with a select mapping strategy provided aid tocircuits with low natural MUX4-embeddable ratios. We

IJMTARC – VOLUME – V – ISSUE - 18 – JUNE , 2017 ISSN: 2320-1363

alsoprovided analysis of the benchmark suites postmapping, discussingthe distribution of functions within each benchmarksuite. From our first set of experiments with nonfracturablearchitectures, area reductions of up to 8% were seen fora 4:6 MUX4:LUT architecture in the CHStone suite witha 2:8 architecture most viable for the VTR suiteswith ~5% area savings. Our second set of experimentswith fracturable architectures showed that the flexibility of afracturable LUT is very powerful, reducing the impact ofthe MUX4 LEs, yielding smaller ~2%–3% area savingsover the VTR7 and CHStone benchmark suites with lessaggressive 2:8 and 1:9 architectures, respectively. Interestingly,we again found that different architectural conclusions canbe made based on the benchmark circuits employed in an architecture study, since CHStone benchmarks generallypreferred more aggressive MUX4:LUT architecture ratios.The CHStone benchmarks being high-level synthesized withLegUp-HLS also showed marginally better performance andthis could be due to the way LegUp performs HLS onthe CHStone benchmarks themselves. Overall, the additionof MUX4s to FPGA architectures minimally impact FMax
and show potential for improving logic-density in nonfracturablearchitectures and modest potential for improving logicdensityin fracturable architectures.

REFERENCES

[1] J. Rose *et al.*, "The VTR project: Architecture and CAD for FPGAsfrom verilog to routing," in *Proc. ACM/SIGDA FPGA*, 2012, pp. 77–86.

[2] Y. Hara, H. Tomiyama, S. Honda, and H. Takada, "Proposal andquantitative analysis of the CHStone benchmark program suite forpractical C-based high-level synthesis," *J. Inf. Process.*, vol. 17,pp. 242–254, Oct. 2009.

[3] A. Caniset *al.*, "LegUp: High-level synthesis for FPGA-basedprocessor/accelerator systems," in *Proc. ACM/SIGDA FPGA*, 2011,pp. 33–36.

[4] E. Ahmed and J. Rose, "The effect of LUT and cluster size on deepsubmicronFPGA performance and density," *IEEE Trans. Very LargeScale Integr. (VLSI)*, vol. 12, no. 3, pp. 288–298, Mar. 2004.

[5] J. Rose, R. Francis, D. Lewis, and P. Chow, "Architecture of fieldprogrammablegate arrays: The effect of logic block functionalityon area efficiency," *IEEE J. Solid-State Circuits*, vol. 25, no. 5,pp. 1217–1225, Oct. 1990.

[6] H. Parandeh-Afshar, H. Benbihi, D. Novo, and P. Ienne, "RethinkingFPGAs: Elude the flexibility excess of LUTs with and-inverter cones,"in *Proc. ACM/SIGDA FPGA*, 2012, pp. 119–128.

[7] J. Anderson and Q. Wang, "Improving logic density through synthesisinspiredarchitecture," in *Proc. IEEE FPL*, Aug./Sep. 2009, pp. 105–111.

[8] J. Anderson and Q. Wang, "Area-efficient FPGA logic elements:Architecture and synthesis," in *Proc. ASP DAC*, 2011, pp. 369–375.

[9] J. Cong, H. Huang, and X. Yuan, "Technology mapping and architectureevalution for k/m-macrocell-based FPGAs," *ACM Trans. Design Autom.Electron. Syst.*, vol. 10, no. 1, pp. 3–23, Jan. 2005.

IJMTARC – VOLUME – V – ISSUE - 18 – JUNE , 2017

ISSN: 2320-1363

[10] Y. Hu, S. Das, S. Trimberger, and L.He, "Design, synthesis andevaluation of heterogeneous FPGA with mixed LUTs and macro-gates,"in *Proc. IEEE ICCAD*, Nov. 2007, pp. 188–193.

[11] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs,"*IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 2,pp. 203–215, Feb. 2007.

[12] K. Karplus, "Amap: A technology mapper for selector-based fieldprogrammablegate arrays," in *Proc. 28th ACM/IEE DAC*, Jun. 1991,pp. 244–247.

[13] A. Mishchenko, S. Chatterjee, and R. Brayton, "DAG-aware AIGrewriting a fresh look at combinational logic synthesis," in *Proc. 43rdAnnu. DAC*, 2006, pp. 532–535.

[14] V. Betz and J. Rose, "VPR: A new packing, placement and routing toolfor FPGA research," in *Proc. 7th Int. Workshop FPL*, 1997, pp. 213–222.

[15] S. A. Chin and J. H. Anderson, "A case for hardened multiplexers inFPGAs," in *Proc. FPT*, Dec. 2013, pp. 42–49.

[16] M. Purnaprajna and P. Ienne, "A case for heterogeneous technologymapping:Soft versus hard multiplexers," in *Proc. IEEE 21st Annu. Int.Symp. FCCM*, Apr. 2013, pp. 53–56.

[17] (2011). *Virtex-6 FPGA User Guide*.[Online]. Available:http://www.xilinx.com

[18] (2011). *Stratix IV Device Handbook*.[Online]. Available:http://www.altera.com

[19] G. Lemieux and D. Lewis, "Using sparse crossbars within LUT,"in *Proc. 9th Int. Symp. ACM/SIGDA FPGA*, 2001, pp. 59–68.

[20] C. Chiasson and V. Betz, "COFFE: Fully-automated transistor sizing forFPGAs," in *Proc. Int. Conf. FPT*, Dec. 2013, pp. 34–41.

[21] *Predictive Technology Model*. [Online]. Available: http://ptm.asu.edu/,accessed 2015.

[22] Altera, private communication, Mar. 2014.

[23] A. Mishchenko, S. Cho, S. Chatterjee, and R. Brayton, "Combinationaland sequential mapping with priority cuts," in *Proc. IEEE/ACM Int.Conf. ICCAD*, Nov. 2007, pp. 354–361.

[24] A. Yan, R. Cheng, and S. J. E. Wilton, "On the sensitivity ofFPGA architectural conclusions to experimental assumptions, tools,and techniques," in *Proc. 10th Int. Symp. ACM/SIGDA FPGA*, 2002,pp. 147–156

© IJMTARC