

Hybrid LUT/MUX FPGA Logic Architecture for Size Reduction and Performance Improvement

S.Hemanath¹, K.Rajesh Kumar²

UG Scholar [VLSI Design], Department of ECE, Adhiyamaan College of Engineering, Hosur, TN, India¹

Associate Professor, Department of ECE, Adhiyamaan College of Engineering, Hosur, TN, India²

ABSTRACT Amending the performance and reducing the area in the racing circuits, an amalgamation of look up tables with multiplexer methodology is together applied. Implementing this variety of architecture an incipient MUX: LUT anatomical structure is designed, which works predicated on the figure of comparators and logical racing circuits .more congruous for this implementation both accounting for involutes logic block and routing area while maintaining mapping depth. The interconnections increasingly the ascendant allele subscriber to delay, area and energy wasting disease in Complementary Metal-Oxide Semiconductor (CMOS) digital circuits. Future implementation surmounts several circumscriptions found in antecedent little Joe implementations published so far, such as the desideratum for special features in the CMOS outgrowth or power-hungry current-mode cells. 512bit four lookup board we have to utilize, for the high caliber of operations in the FPGA. Proposed architecture of this paper will be orchestrated to implemented and additionally analytic thinking the output current, output voltage, area utilizing Xilinx 14.3.

KEYWORDS: Field-Programmable Gate Array (FPGA), Hybrid Complex Logic Block, Multiplexer (MUX).

I. INTRODUCTION

Field-programmable gate arrays (FPGAs) and lookup tables (LUTs) have been the primary winding system of logic element (LE) utilized to realize combinational logic. K-input LUT is generic and very flexible—able to implement any K-input Boolean function. Utilization of LUTs simplifies engineering 2 senses of map as the quandary is reduced to a graph covering quandary. The exponential domain price is paid as more sizably voluminous LUTs are considered. A value of K between 4 and 6 is typically optically discerned in industry and academia, and this mountain range has been demonstrated to offer a good area/carrying into action compromise. Currently, the number of other works have explored alternative FPGA LE architectures for performance melioration to close the large gap between FPGAs and application-specific integrated circuits (ASICs). We propose incorporating (some) hardened multiplexers (MUXs) in the FPGA logic blockage as a substance of increasing silicon area efficiency and logic density.

MUX-predicated logic blocks for the FPGAs have optically discerned prosperity in early commercial architectures, such as the Actel ACT-1/2/3 architectures, and efficient mapping to these complex body part has been field in the early 1990s. We can utilize in commercial Saratoga chip has waned, perhaps partly due to the facileness with which logic functions can be mapped into LUTs, simplifying the entire electronic computer aided design (CAD) flow. Notwithstanding, it is widely understood that the LUTs are inefficient at implementing MUXs, and that MUXs are frequently utilized in logic racing circuits. We underscore the inefficiency of LUTs implementing MUXs, consider that a six input LUT (6-LUT) is essentially a 64-to-1 MUX (to cull 1 of 64 truth-table rows) and 64-SRAMconfiguration cadre, yet it can only realize a 4-to-1 MUX (4 data + 2cull = 6 inputs).

We present a six-input LE predicated on a quarter-to-one MUX, MUX4, that can realize a subset of six-input Boolean logical system social function, and an incipient hybrid involutes logic block (CLB) that contains a coalescence of MUX4s and 6-LUTs. A Proposed MUX4s are minute compared with a 6-LUT (15% of 6-LUT area), and can efficiently map all {2, 3}-input functions and some {4, 5, 6}-input functions. This additament, we explore fracturability of LEs—the facility to split the LEs into multiple more minuscule elements—in both LUTs and MUX4s to increment logic concentration. A ratio of LEs that should be LUTs versus MUX4s is additionally explored toward optimizing logic denseness for both nonfracturable and fracturable FPGA architectures. Facilitate the architecture geographic expedition, we developed a CAD menstruation for mapping into the proposed hybrid CLBs, engendered utilizing ABC and VPR, and describe technology mapping techniques that embolden the cull of logic functions that can be embedded into the MUX4 elements.

Main share in this arrangement are as follows.

1. Two loanblend CLB architectures (nonfracturable and fracturable) that contain a coalescence of MUX4 LEs and the traditional LUTs conceding up to 8% area savings.
2. The Mapping techniques called NaturalMux and MuxMap targeted toward the hybrid CLB architecture that optimize for area, while preserving the pristine mapping depth.
3. Full phase of the moon post-place-and-route architecture evaluation with VTR7, and CHStone benchmark facilitated by LegUp-HLS, the Verilog-to-Routing project exhibiting impact on both area and hold.

Match with the preliminary issue, we have performed transistor level clay sculpture of the MUX4 LE, further report the fracturable architectures, and amalgamated the out-of-doors source implement-flow from C through LegUp-HLS to the VTR flow. A sparse crossbar has additionally been included in our CLBs, incrementing modeling precision. An incipient transistor-level modeling of the MUX4 additionally provides more precise answer as compared with the antecedent study. The Results have additionally been expanded with the inclusion of timing results as well as more sizably voluminous architectural ratio sweeps.

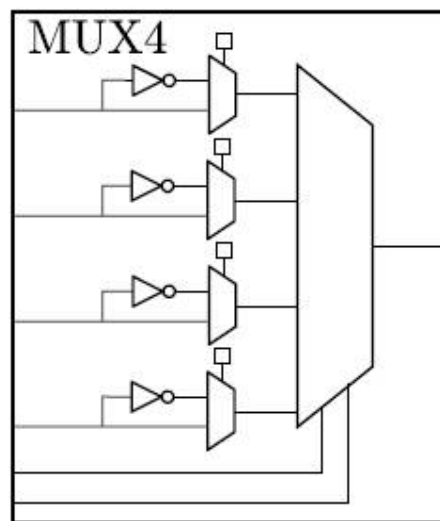


Fig 1 MUX4 LE depicting optional data input inversions.

II. MUX4: 4-TO-1 MULTIPLEXER LOGIC ELEMENT

MUX4 LE shown in Fig. 1 consists of a four-to-one MUX with optional anastrophe on its input that sanction the entelechy of any {2, 3}-input function, some {4, 5}-input functions, and one 6-input function—a 4-to-1 MUX itself with optional inversion on the data inputs. Four-to-one MUX matches the input pin count of a 6-LUT, sanctioning for fair comparisons with veneration to the connectivity and intra-cluster routing.

Any two-input Boolean function can be facilely implemented in the MUX4: Either the single valued function inputs can be tied to the cull lines and the truth table values (logic-0 or logic-1) can be routed to the data inputs accordingly. Or alternately, Shannon vector decomposition can be performed about one of the two variables—the variable can then victual a cull input. Shannon cofactors will contain at most one variable and can, consequently, be victual to the data inputs (the optional inversion may be needed).

Three inputs functions; consider that Shannon decomposition about one variable engenders cofactors with at most two variables. The second decomposition of the cofactor about one of their two remaining variables engenders cofactors with at most one variable. The single-variable cofactors can be victual to the data inputs (the optional inversion may be needed), with the decomposition variables victualing the cull inputs. Correspondingly, functions of more than four inputs can be implemented in the MUX4 as long as Shannon decomposition with deference to any two inputs engenders cofactors with at most one input. Regarding that input inversion on each cull input is omitted as this would only accommodate to permute the four MUX data inputs. Although this could avail routability within the CLB's internal crossbar, supplemental inversions on the cull inputs would not increment the issue of Boolean functions that are able to map to the MUX4 LE. Any two-input Boolean function can be facilely implemented in the MUX4: the two function inputs can be tied to the cull lines and the truth table values (logic-0 or logic-1) can be routed to the data inputs accordingly. Or alternately, Shannon decomposition can be performed about one of the two variables - the variable can then aliment a cull input. The Shannon cofactors will contain at most one variable and can, ergo, be alimented to the data inputs (the optional inversion may be needed).

Second decomposition of the cofactors about one of their two remaining variables engenders cofactors with at most one variable. While one-variable cofactors can be alimented to the data inputs (the optional eversion may be needed), with the decomposition variables alimenting the cull inputs. Moreover, functions of more than four inputs can be implemented in the MUX4 as long as Shannon decomposition with reverence to any two inputs engenders cofactors with at most one input. Regarding that input inversion on each cull input is omitted as this would only accommodate to permute the four MUX data inputs. While this could avail routability within the CLB's internal crossbar, supplemental inversions on the cull inputs would not increment the number of Boolean functions that are able to map to the MUX4 LE.

III. LOGIC ELEMENTS, FRACTURABILITY, AND MUX4-BASED VARIANTS

Two families of architectures were engendered: A. Without fracturable LEs and B. With fracturable LEs. Latest arrangement, the fracturable LEs refer to an architectural element on which one or more system of logic functions can be optionally mapped. The Nonfracturable LEs refer to an architectural element on which only one logic function is mapped. Favored nonfracturable architectures, the MUX4 element shown in Fig. A is utilized together with nonfracturable 6-LUTs. The element percentage the same number of inputs as a 6-LUT lending for fair comparability with reverence to the input connectivity. Being the fracturable architecture, we consider an eight-input LE, proximately matched with the adaptive logic module in recent Altera Stratix FPGA families. The 6-LUT that can be fractured into two 5-LUTs utilizing eight inputs is shown in Fig. B.

Fracturable 6-LUT

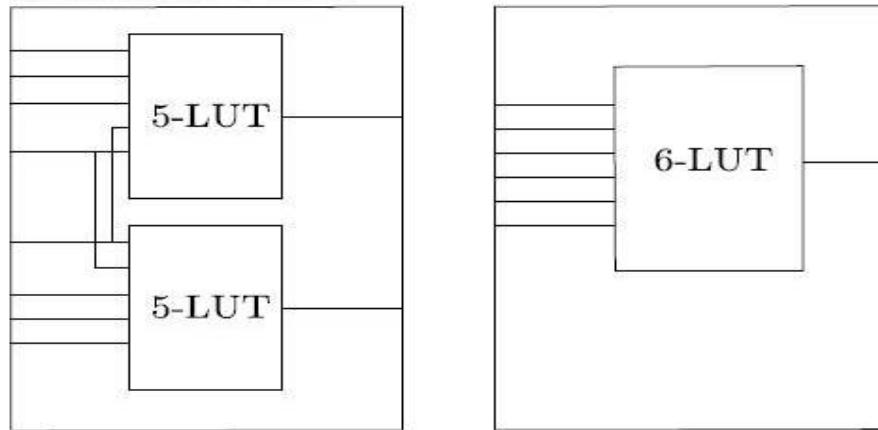


Fig 2 Fracturable 6-LUT that can be fractured into two 5-LUT with two shared inputs.

Then the Two five-input functions can be mapped into this LE if two inputs are shared between the two functions. In case that no inputs are shared, two four-input functions can be mapped to each 5-LUT. Since the MUX4 variant, Dual MUX4, we utilize two MUX4s within a one eight-input LE. Latest Contour, shown in Fig. C, the two MUX4s are wired to have dedicated cull inputs and shared data inputs.

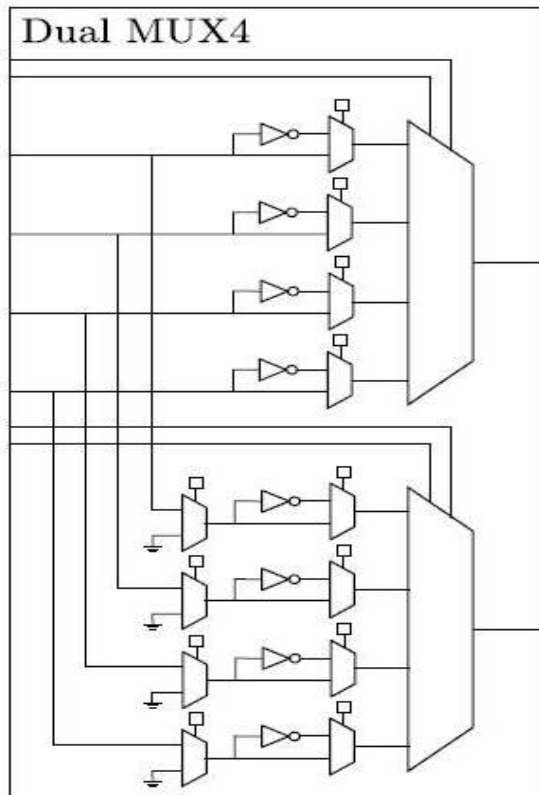


Fig 3 Dual MUX4 LE that utilizes dedicated select inputs shared data inputs

A contour sanctions this socialorganization to map two independent (no shared inputs) three-input functions, while more sizably voluminous functions may be map dependent on the shared inputs between both functions. The architecture in which a four-to-one MUX (MUX4) is fractured into two more minuscule two-to-one MUXs was first considered. Though, since a two-to-one MUX's mapping flexibility is quite constrained (can only map 2 input functions and the 3 input 2-to-1 MUX itself), little benefit was integrated compared with the overheads of qualification the MUX4 fracturable and poor area upshot were observed.

IV. SUBSISTING SYSTEM

- ✓ Current subsisting system, it has six input LUTs predicated on 4X1 MUX in Boolean logic functions.
- ✓ Inclination of minimal efficiency on timing area and performances and withal more delay occurs in the functions.
- ✓ Considering Practical implementations, the cost requisite is comparatively high, but the expected output will be achieved, so that we have to bear the cost for accessing.

DISADVANTAGES OF SUBSISTING SYSTEM

- Delay rate is too high.
- More doe intake.
- Cost Sumptuous.
- Less carrying into action.

V. PROPOSED METHODOLOGY

- The Two loanbend building Involute Logic Block (CLB) architectures (non-fracturable and fracturable) that contain a cumulation of MUX4 Logical Elements (LEs) and the traditional LUTs yielding more area deliverance.
- A Mapping technique called NaturalMux and MuxMap targeted toward the hybrid CLB architecture that optimize for area, while preserving the pristine mapping depth.
- The full of the moon post-place-and-route architecture evaluation with Verilog-to-Routing (VTR) benchmarks facilitated by LegUp-HLS, the Verilog-to-Routing project, exhibiting impact on both area and delay.

ADVANTAGES OF PROPOSED SYSTEM

- Delay is comparatively reduced.
- Doe intake is reduces.
- Low cost sumptuous in work.
- High carrying into action.

VI. LITERATURE SURVEY

Toward facilitate the growing of time to come FPGA architectures and CAD implements -- both embedded programmable framework and pristine-play FPGAs -- there is a desideratum for an astronomically immense scale, publicly available software suite that can synthesizes electronic circuit into facilely-described hypothetical

FPGA architectures. Those circuits should be captured at the HDL level, or higher, and pass through logical and physical synthesis. Comparable implement must provide detailed modelling of field, carrying into action and energy to enable architecture exploration. On the point of software catamenia they evolve to sanction design capture at ever higher levels of abstraction, this downstream full phase of the moon-implementation flow will always be required.

This paper outline shows the present status and incipient relinquishment of an perpetual close eavor to engender such a flow - the 'Verilog to Routing' (VTR) project, which is a broad collaboration of researchers. Capable of three core centre puppet: ODIN II for Verilog Elaboration and front-end hard-block synthesis, ABC for logic synthesis, and VPR for physical synthesis and analysis. The ODIN II now has a simulation capability to avail verify that its output is veridical, as well as specialized synthesis at the elaboration tone for multipliers and recollections. An ABC is utilized to optimize the 'soft' logic of the FPGA. A VPR-predicated packing material, locating and routing is now plenarily timing-driven (the anterior release was not) and includes incipient capability to quarry intricate logic blocks.

Trendy integration we have integrated a set of four astronomically immense benchmark circuits to a rooms of antecedent-relinquished Verilog HDL circuits. Determinately, we illustrate the utilization of the incipient flow by utilizing it to avail architect a floating-point building block in an FPGA, and contrast it with a prior, much longer endeavour that was required to do equipollent.

Toward facilitate the development of future FPGA computer architectures and CAD implements both embedded programmable framework and pristine-sport FPGAs -- there is a desideratum for a sizably voluminous scale, publicly available software suite that can synthesize circuits into facilely-described hypothetical FPGA architectures. Those circuits should be captured at the HDL level, or higher, and pass through logical and physical synthesis. Corresponding an implement must provide detailed modelling of expanse, carrying into action and vigour to enable architecture exploration. On the point of software stream they evolve to sanction design capture at ever higher levels of abstraction, this downstream full-effectuation stream will always be required.

This paper describes the present condition and incipient relinquishment of a perpetual effort to engender such a stream - the 'Verilog to Routing' (VTR) projection, which is a broad collaboration of research worker. There are three core implements: ODIN II for Verilog Elaboration and front-end hard-closure deductive reasoning, ABC for logic synthesis, and VPR for physical synthesis and psychoanalysis. ODIN II now has a pretense capability to avail verify that its output is veridical, as well as specialized synthesis at the elaboration stair for multipliers and recollections. ABC is utilized to optimize the 'soft' logic of the FPGA.

The VPR-predicated packing material, emplacement and routing is now plenarily timing-ride (the anterior release was not) and includes incipient capability to target involute logic blocks. In additament we have integrated a set of four astronomically immense benchmark circuits to a suite of previously released Verilog HDL circuits. Determinately, we illustrate the utilization of the incipient sream by utilizing it to avail architect a floatingpoint unit in an FPGA, and contrast it with a prior, much longer effort that was required to do equipollent.

This arrangement, we introduce an incipient out of doors source high-level synthesis implement called LegUp that sanctions computer software technique to be utilized for computer hardware design. The LegUp accepts a standard C program as input and automatically compiles the program to a hybrid architecture containing an FPGA-predicated MIPS soft processor and custom hardware expeditors that communicate through a standard bus interface. The Results appearance that the implement engenders hardware solutions of commensurable quality to a commercial high-level synthesis implement.

VII. EXPERIMENTAL RESULT

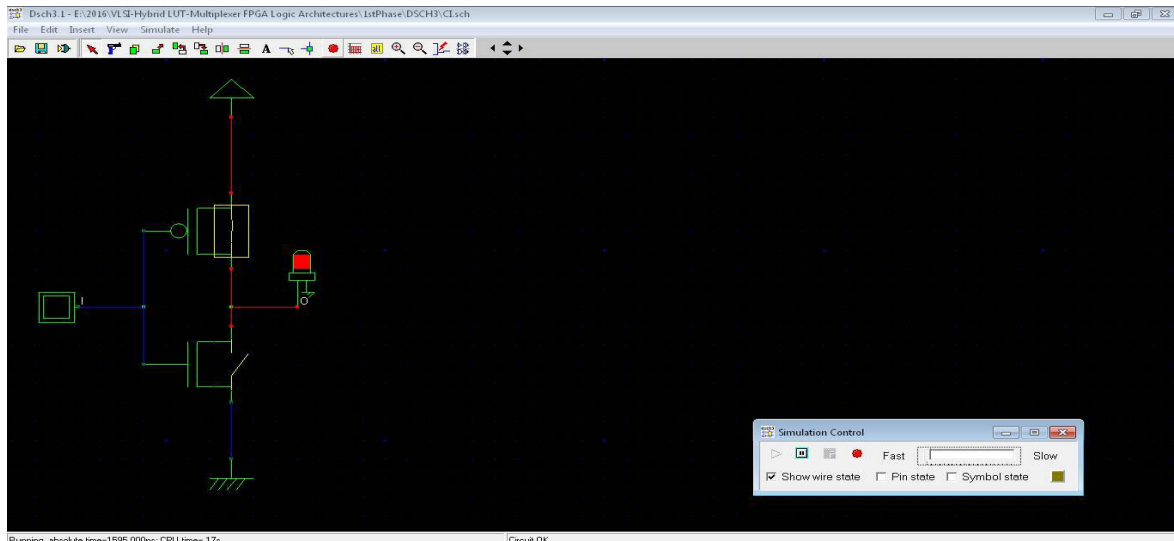


Fig 4 Schematic Design

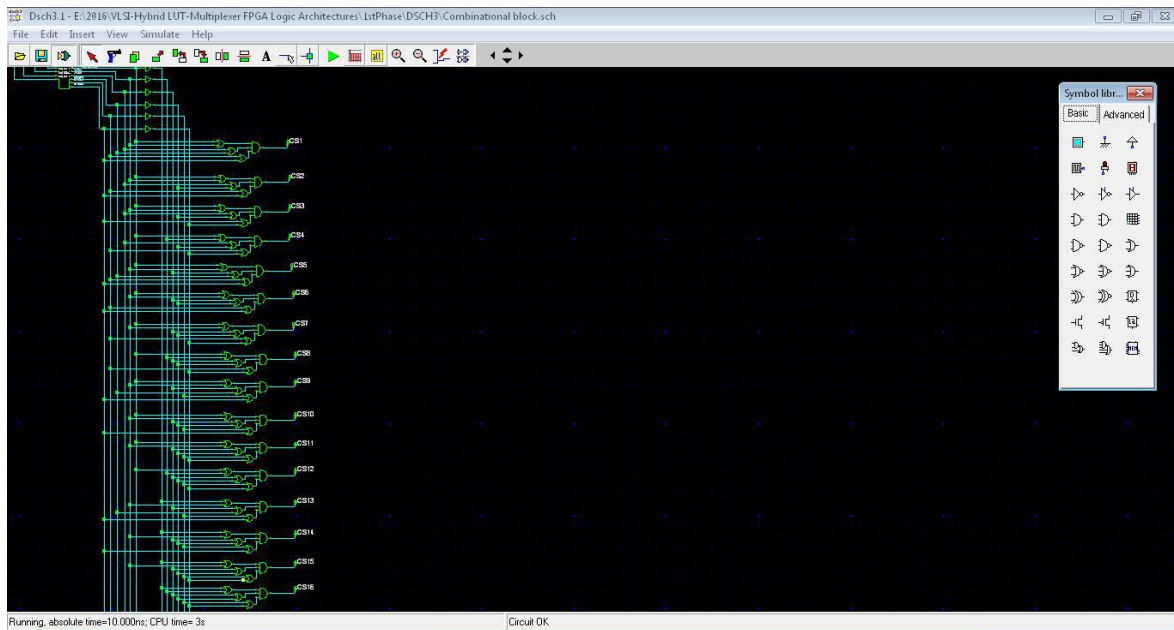


Fig 5 MUX Design

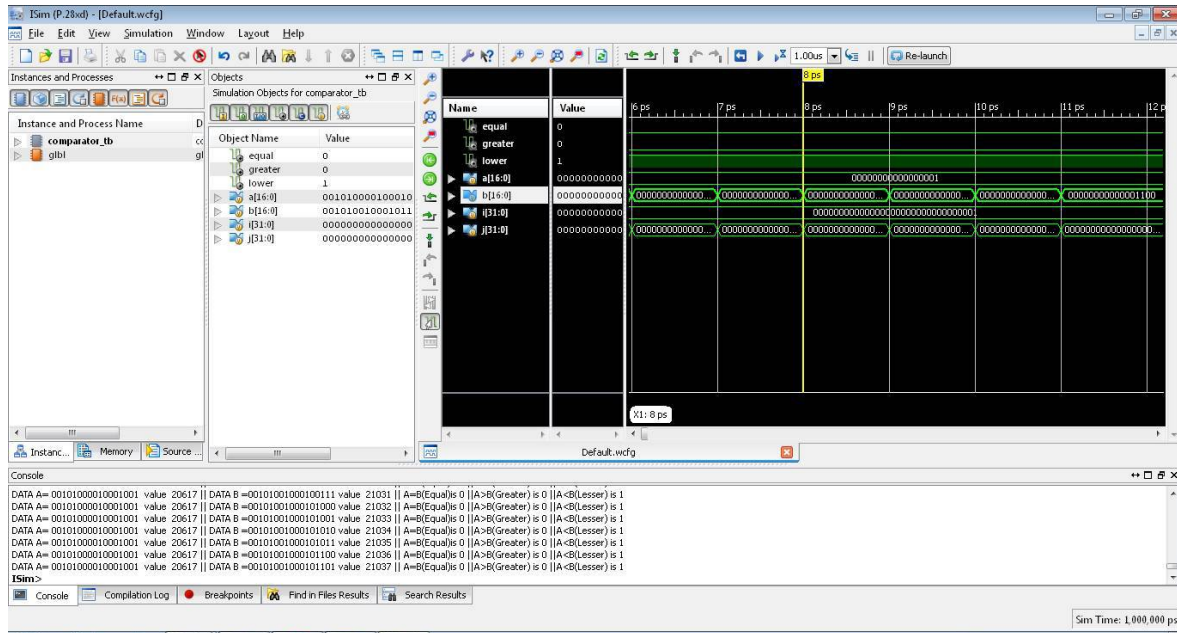


Fig 6 Xilinx Design

VIII. CONCLUSION AND FUTURE SCOPE

The Propose incipient hybrid CLB architecture containing MUX and logical elements and display the techniques for efficiently mapping to these architectures. Operation of LUT payoff more power preserving and area preserving nature with monetary value benefits. A coalescence of MUX: LUT engenders prosperous power consumption and size reductions schemas in single unit.

Proposed scheme guarantees the size and nature of the LUT MUX in more efficient manner, however in future this work is elongated to interconnect many features, which are increasingly the ascendant contributor to delay, area and energy consumption in CMOS digital circuits. The implementation surmounts several inhibitions found in precedent implementations published so far, such as the desideratum for special features in the CMOS process or magnate-hungry present mode cells.

REFERENCES

- [1] J. Rose et al., "The VTR project: Architecture and CAD for FPGAs from verilog to routing," in Proc. ACM/SIGDA FPGA, 2012, pp. 77–86.
- [2] Y. Hara, H. Tomiyama, S. Honda, and H. Takada, "Proposal and quantitative analysis of the CHStone benchmark program suite for practical Cbased High-level synthesis," J. Inf. Process., vol. 17, pp. 242–254, Oct. 2009.
- [3] A. Canis et al., "LegUp: High-level synthesis for FPGA-based processor/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 deepsubmicron FPGA performance and density," IEEE Trans. Very Large Scale Integr. (VLSI), vol. 12, no. 3, pp. 288–298, Mar. 2004.
- [5] J. Rose, R. Francis, D. Lewis, and P. Chow, "Architecture of fieldprogrammable gate arrays: The effect of logic block functionality on 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. Jenne, "Rethinking FPGAs: 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 synthesinspired architecture," in Proc. IEEE FPL, Aug. /Sep. 2009, pp. 105–111.
- [8] J. Anderson and Q. Wang, "Area-efficient FPGA logic elements: rchitecture and synthesis," in Proc. ASP DAC, 2011, pp. 369–375.

- [9] J. Cong, H. Huang, and X. Yuan, "Technology mapping and architecture evaluation for k/m-macrocell-based FPGAs," *ACM Trans. Design Autom. Electron. Syst.*, vol. 10, no. 1, pp. 3–23, Jan. 2005.
- [10] Y. Hu, S. Das, S. Trimberger, and L. He, "Design, synthesis and evaluation 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 fieldprogrammable gate arrays," in *Proc. 28th ACM/IEEE DAC*, Jun. 1991, pp. 244–247.
- [13] A. Mishchenko, S. Chatterjee, and R. Brayton, "DAG-aware AIG rewriting a fresh look at combinational logic synthesis," in *Proc. 43rd Annu. DAC*, 2006, pp. 532–535.
- [14] V. Betz and J. Rose, "VPR: A new packing, placement and routing tool for 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 in FPGAs," in *Proc. FPT*, Dec. 2013, pp. 42–49.