International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 13
October 2017

Design and Implementation of Hybrid Lut/Multiplexer Fpga Logic Architectures

Kola Shailaja
M.Tech(VLSI)
Dept of E.C.E
Chaitanya Institute of Technology and Science, Warangal, Telangana,India
Email: kola.shailaja@gmail.com

M.Swapna
Assistant Professor
Dept of E.C.E
Chaitanya Institute of Technology and Science, Warangal, Telangana,India
Email: sumedha_452@yahoo.com

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 CH Stone) 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 to2%. 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 anti fuses 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 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,

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 13
October 2017

then with asymmetric LUT LEs, show that the LUT elements present in commercial FPGAs provide unnecessary flexibility. Toward improved delay and area, the macro cell-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.

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≤k+3), the k/m-macro cell based FPGAs can achieve the same or similar mapping depth compared with the traditional k input 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.

III. PROPOSED ARCHITECTURES

A. MUX4: 4-to-1 Multiplexer Logic Element

The MUX4 LE shown in Fig. 3 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 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 intra cluster routing. 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-0or logic-1) can be routed to the data inputs accordingly. For three-input functions; consider that Shannon decomposition about one variable produces cofactors with at most two variables.

A second decomposition of the cofactors about one of their two remaining variables produces cofactors with at most one variable. Such single-variable cofactors can be fed to the data inputs (the optional inversion may be needed), with the decomposition variables feeding the select inputs. Likewise, functions of more than four inputs can be implemented in the MUX4 as long as Shannon decomposition with respect to any two inputs produces cofactors with at most one input.

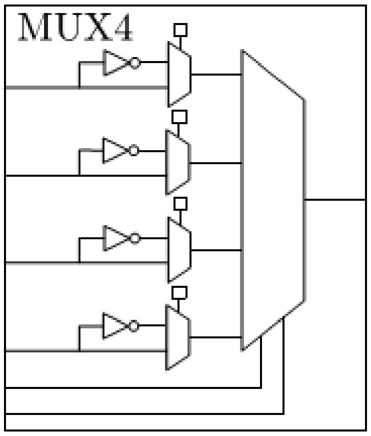


Fig.1. MUX4 LE depicting optional data input Inversions

B. Logic Elements, Fracturability, and MUX4-Based Variants

Two families of architectures were created:
1) Without fracturable LEs
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. For the fracturable architecture, we consider an eight-input LE, closely matched with the adaptive logic module in recent Altera Stratix FPGA families.For the MUX4 variant, Dual MUX4, we use two MUX4s within a single eight-input LE. In the configuration, shown in Fig. 2, the two MUX4s are wired to have dedicated select inputs and shared data inputs. This configuration allows this structure to map two independent (no shared inputs) three-input functions, while larger functions may be mapped dependent on the shared inputs between both functions. An architecture in which a 4-to-1 MUX (MUX4) is fractured into two smaller 2-to-1 MUXs was considered.

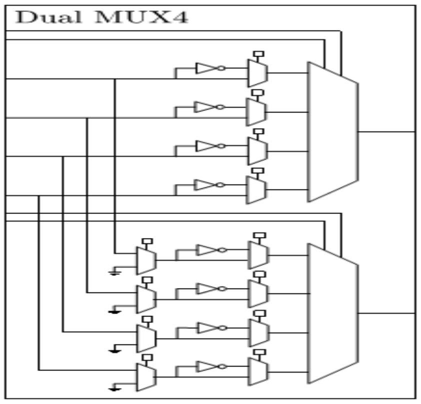


Fig.2. Dual MUX4 LE that utilizes dedicated select inputs and shared data Inputs

C. 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. Fig.3 shows this nonfracturable CLB architecture with BLEs that contain an optional register. We vary the ratio of MUX4s to LUTs within the ten elements 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. 4 shows the organization of our CLB and internal BLEs. 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. 2 shows the fracturable architecture with eight inputs to each BLE that contains two optional registers.

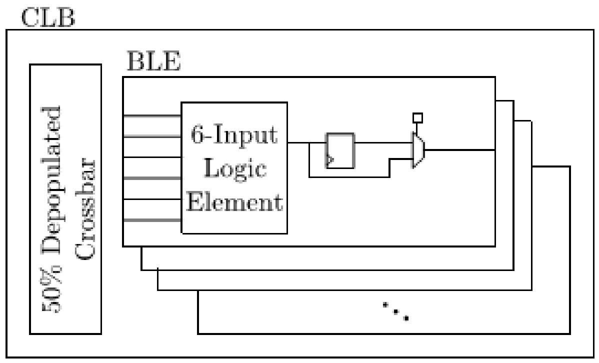


Fig. 3. Hybrid CLB with a 50% depopulated intra-CLB crossbar depicting BLE internals for nonfracturable (one optional register and one output) architecture.

We evaluate fracturability of LEs versus nonfracturable LEs in the context of MUX4 elements since fracturable LUTs are common in commercial architectures. For example, Altera Adaptive 6- LUTs in Stratix IV and Xilinx Virtex 5 6-LUTs can be fractured into two smaller LUTs with some limitations on inputs.

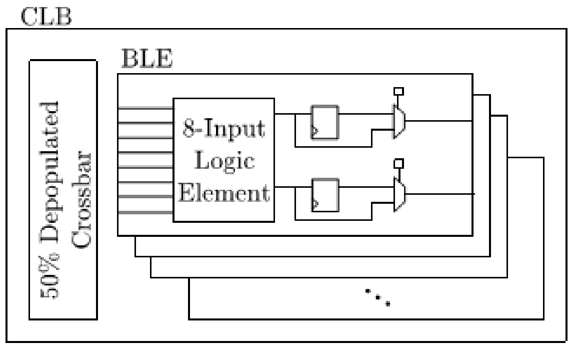


Fig.4. Hybrid CLB with a 50% depopulated intra-CLB crossbar depicting BLE internals for a fracturable (two optional registers and two outputs) architecture.

D. Area Modeling

1) MUX4 Logic Element: Initial estimates of the MUX4 element showed that the MUX4 is 10% the area of a 6-LUT overall. A 4-to-1 MUX can be realized with three 2-to-1 MUXs. Hence, the MUX4 element contains seven 2-to-1 MUXs, four SRAM cells, and four inverters in total. The optional inversion uses the four SRAM cells, whereas the rest of the LE configuration is performed through routing. In addition, the depth of the MUX tree is halved compared with the 6-LUT, which has six 2-to-1 MUXs on its longest paths. Conservatively, assuming constant pass transistor sizing and that the area of a 2-to-1 MUX and six transistor SRAM cell are roughly equivalent, the MUX4 element has (1/16)th the SRAM area and(1/8)th the MUX area of a 6-LUT.

These estimates were revised using transistor level modeling of the circuit blocks. Transistor level optimization of the constituent circuit blocks of an FPGA requires an understanding of the optimal area-delay tradeoffs for each individual circuit block. This requires extracting a representative critical path, which is a path whose composition of blocks and topology will be similar to the critical path of a specific design. Extracting the representative critical path allows us to judge to what extent each individual block is timing critical, which thus establishes an area-delay tradeoff goals for each block. This is in line with the transistor-level optimization tool developed previously.

We use the results of prior work to establish the optimal area-delay tradeoff for 6-LUTs in conventional island-style FPGA architecture with typical architectural parameters. However, we chose to use the minimum delay design for both the MUX4 and Dual MUX4 elements for the rest of the study as there is not a significant increase in area over the minimum area design.

2) FPGA Area Model:

Although determining the area of a MUX4 element relative to a 6-LUT is important, we need to also examine global FPGA area considering the number of CLB tiles, area overheads within the CLB and routing area per CLB. Throughout this paper, global FPGA area was estimated assuming that, per tile, 50% of the area is inter cluster and intra cluster routing, 30% of the area is used for LUTs, and 20% for registers and other miscellaneous logic, following Anderson and Wang and a private communication.

It is important to note that this 50%–30%–20% model is an estimate based on a traditional full FPGA design where-by the routing and internal CLB crossbars are optimized toward 6-LUTs. Production of an optimized FPGA utilizing our new MUX4 elements would surely change said model.

However, optimizing the entire routing architecture toward our MUX4 variants, measuring the routing architecture, and closing the loop by creating a more accurate model is out of the scope of this work. Using this model, we can make some observations about the hybrid CLB architecture. The 30% that normally would account for ten 6-LUT LEs within the tile is now split between the smaller MUX4 elements and 6-LUTs.

IV. RESULTS

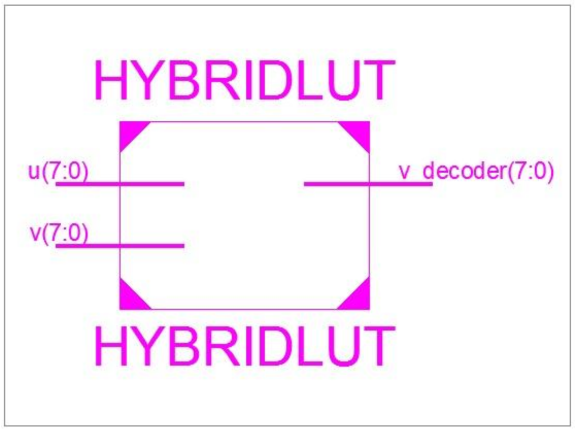
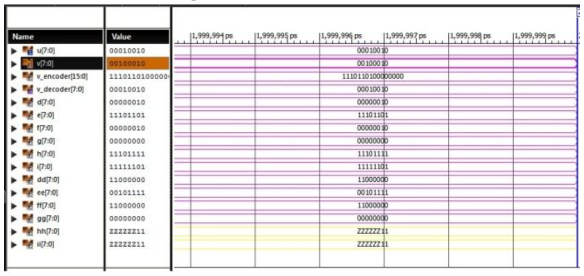


Fig.5. RTL Schematic



Name	Value
u[7:0]	00010010
v[7:0]	00100010
v_encoder[15:0]	1110110100000000
v_decoder[7:0]	00010010
d[7:0]	00000010
e[7:0]	11101101
f[7:0]	00000010
g[7:0]	00000000
h[7:0]	11101111
i[7:0]	11111101
dd[7:0]	11000000
ee[7:0]	00101111
ff[7:0]	11000000
gg[7:0]	00000000
hh[7:0]	ZZZZZZ11
ii[7:0]	ZZZZZZ11

Fig.6. output waveform

V. CONCLUSION

In this paper we proposed a new hybrid CLB architecture containing MUX4 hard MUX elements and shown techniques for efficiently mapping to these architectures. We also provided analysis of the benchmark suites post mapping, discussing the distribution of functions within each benchmark suite. The area reductions for nonfracturable architectures, is 8% and MUX4:LUT ratio is 4:6 and in the case of fracturable architecture the area reductions are 2%.

The CH Stone benchmarks being high level synthesized with Leg Up-HLS also showed marginally better performance and this could be due to the way LegUp performs HLS on the CHStone benchmarks themselves. Overall, 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 architecture.

VI.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 C-based 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 deep submicron 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 field programmable 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. Ienne, "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 synthesis inspired architecture," 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.

Authors:

Kola Shailaja studying M.Tech (VLSI) from Chaitanya Institute of Technology and Science, Warangal, Telangana,India.

M.Swapna working **as** Assistant Professor in Dept of E.C.E from Chaitanya Institute of Technology and Science, Warangal, Telangana, India.