This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2016.2551555, IEEE Transactions on Circuits and Systems II: Express Briefs

A Hybrid Logic Block Architecture in FPGA for Holistic Efficiency

Tao Luo, *Student Member, IEEE,* Hao Liang, *Member, IEEE,* Wei Zhang, *Member, IEEE,* Bingsheng He, *Member, IEEE,* and Douglas Maskell, *Member, IEEE*

***Abstract*—This brief presents a hybrid design of configurable logic block (CLB) composed of look-up tables (LUTs) and universal logic gates (ULGs). An ULG is designed to realize holistic efficiency compared with the corresponding LUT. Previous designs with ULGs are either based on pure ULG or LUT-ULG complementary architecture, which incur longer delay or double the area compared to LUT based design. In contrast, we propose a hybrid CLB that contains a mixture of LUTs and ULGs to address the generality problem as well as achieving the holistic benefits including the area, performance, and power. To exploit the advantage of ULGs thoroughly while not causing negative side-effects, the ratio of LUTs and ULGs in one CLB is explored by experiments. Experiment results show that compared to pure LUT design our proposed architecture design can save up to 17.1% logic power as well as 11.2% delay improvement and 10.4% logic area reduction. Compared to the state-of-the-art design, our proposed design has 3.8% improvement in Power Delay Product (PDP) and 17.1% improvement in area cost.**

***Index Terms*—Universal logic gate (ULG), hybrid logic block, field-programmable gate array (FPGA).**

I. Introduction

FPGA design has attracted many research efforts [1]. In order to improve the FPGA design, previous studies usually made trade-offs between power, performance, and area. For example, a previous study increases the area to reduce the power consumption in dark silicon [2]. However, in industry, the area which relates to the cost and performance is often first-class factor to be considered. A design to address the overall benefits would be best favored.

An N-input LUT is the basic unit in FPGA which can implement 2^{2^n} functions by configuring its 2^n configuration bits. However, among the total 2^{2^n} functions, only a small number of functions are frequently used in real designs [3]. Hence, the rarely used functions lead to unnecessary cost. An ULG is defined in [4] as a logic gate that can implement several functions by configuring its several configuration bits. By implementing only key functions, the area of an ULG can be much smaller than LUT, which in turn leads to lower power consumption. The key challenge to design an efficient ULG is to best cover the NPN classes detailed in Section II while maintaining small area and short critical path.

In this brief, we design an ULG with only four 2-input NAND gates, three configurable inverters and a 2-to-2 multiplexer. It has 61.5% and 65.5% improvements in speed and power respectively compared to its LUT counterpart. Since ULGs cannot implement all Boolean functions in applications, it faces the generality problem that it may takes much more ULGs compared to LUTs to cover some functions. In order to solve the problem, we propose a new logic block architecture whose configurable logic blocks consist of both ULGs and LUTs. With this hybrid architecture, a logic cell can be implemented by a physical block of LUT if it is not covered by ULG to achieve the best efficiency. In this hybrid design, a critical problem is to determine the suitable ratio between LUTs and ULGs. We explored the ratio by performing detailed and comprehensive experiments on the 20 largest circuits in MCNC benchmark suite [5]. We implement our architecture using $40nm$ CMOS technology and model it through VTR 7.0 [6]. Experiment results show that compared to pure LUT design our proposed architecture design can save up to 17.1% logic power together with 11.2% delay improvement and 10.4% logic area reduction. Compared to a recently proposed complementary Mega Cells (MCs) design [7], our hybrid architecture has 3.8% improvement in PDP and 17.1% improvement in area cost.

II. Preliminaries and Related Work

A. Functions Classified by NPN Equivalence

In digital circuit, each N-input single-output circuit corresponds to an N-input Boolean function. In the process of mapping Boolean functions to actual physical circuits, one efficient way is to classify the Boolean functions by NPN equivalence classes [8].

*D*efinition (NPN-Equivalence): Let f and g be the Boolean functions of two circuits. f and g are NPN equivalent if they have this property:

$$f \equiv g \Longleftrightarrow f = (g \circ X_\pi \circ X^\phi)$$

where π denotes permutation to the input while ϕ denotes phase inversion to the input and output:

$$X_\pi(x_1, ..., x_n) := (x_{\pi(1)}, ..., x_{\pi(n)})$$

$$X^\phi(x_1, ..., x_n, y) := (x_1^\phi, ..., x_n^\phi, y^\phi)$$

By definition, if we can get function g by permuting the input and inverting the input and output of function f, then the two functions are NPN equivalent. According to its definition, one NPN class can cover $2^{N+1} \cdot N!$ distinct functions. Hence, NPN classification is a very efficient and concise way to classify Boolean functions.

In order to find out the NPN classes distribution in general designs, we take the "MCNC Golden 20", i.e., the 20 largest MCNC benchmark circuits as our benchmark to explore their distribution. The benchmarks are shown in Table I. We synthesize the benchmark circuits and map them to 4-input LUTs using the widely accepted academic tool, ABC [9]. The NPN classes are calculated and the result of NPN classes distribution is shown in Table II. The Boolean functions listed in the Table are the representative functions of corresponding

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2016.2551555, IEEE Transactions on Circuits and Systems II: Express Briefs

TABLE I
20 LARGEST MCNC BENCHMARK CIRCUITS

alu4	des	ex5p	s38417
apex2	diffeq	frisc	s38584.1
apex4	dsip	misex3	seq
bigkey	elliptic	pdc	spla
clma	ex1010	s298	tseng

NPN classes. Note that the coverage number can vary because functions with the number of the input less than four can be classified into different NPN classes. The Table reveals a fact that most of the 4-input Boolean functions of the circuits are distributed in a small number of NPN classes. For example, only 5 NPN classes cover almost 80% Boolean functions.

TABLE II
NPN CLASSES DISTRIBUTION IN 20 LARGEST MCNC BENCHMARK CIRCUITS

Rank	Representative Function of NPN class	Appearance Ratio[%]
1	ABCD	22.618
2	A(B+C+D)	21.205
3	A(B+CD)	19.211
4	AB+CD	11.06
5	AB(C+D)	5.775
-	others	20.13

B. Related Work

There are a lot of previous studies that make efforts to optimize the logic block on FPGA in terms of performance, power, and area [3, 7, 10–23]. For area, shadow clusters were introduced to enhance area efficiency [10, 11]. Application specific hard logic was proposed to improve the performance [12]. Fine-grained power gating was used to reduce the static power [13]. Y. Hu et al. designed a logic block using a mix of LUTs and macrogates to reduce the number of configuration memory bits [19, 20]. However, placement and routing evaluation are not performed.

As a technique to minimize the logic, ULG was proposed to replace LUTs since 1994 [15, 17, 22, 23]. In 2010, the ULG was proposed again by Okamoto et al. to replace 4-input LUTs, 5-input LUTs and 6-input LUTs to reduce area overhead [3]. However, the 5-input LUTs and 6-input LUTs cannot be well replaced by ULGs because of the fact that for 6-input logic functions, a small number of NPN classes cannot cover most of the functions used in real designs. Besides, they used ULG alone, and their logic blocks cannot support arbitrary functions. Their synthesis flow is modified to only support functions covered by the ULG, which leads to overhead in terms of power, area, and performance.

Closely related to our study, Ahari et al. proposed an architecture, in which the traditional soft logic is replaced with Mega Cells (MCs) [7]. Each of them consists of a set of complementary Generic Reconfigurable Hard Logic (GRHL) and a conventional Look-Up Table (LUT). The GRHLs and LUT can be power gated exclusively. In fact, GRHLs in their design are actually ULGs. Their goal is to reduce the power at the cost of area increase. However, the area overhead is very large, because a whole LUT is used as a backup. Only certain portion of the logics in FPGA are able to run at one time, and the rest of logics are wasted. Besides, the use of fine-grained power-gating in their design also introduces extra circuitry with considerable cost and offsets the expected benefit. In contrast, we propose a new architecture to address the generality problem as well as achieving the holistic benefits including the area, performance, and power. Note in this brief, we only focus on 4-input functions instead of 5- or 6-input ones to achieve an overall improvement. The main reason is that 5- and 6- input functions cannot be classified under a small number of NPN classes.

III. PROPSED ARCHITECTURE

A. Design of ULG

Based on the observation above, we propose an ULG structure to cover all the five NPN classes listed in Table II. The logic circuit of proposed ULG is shown in Fig. 1. As shown in Fig. 1, the ULG consists of four 2-input NAND gates, three configurable inverters and a 2-to-2 multiplexer. The configurable inverter is actually a NXOR gate with an input connected to a configuration bit. By setting different values to the configuration bit of the configurable inverter, it can function as an inverter or a buffer.

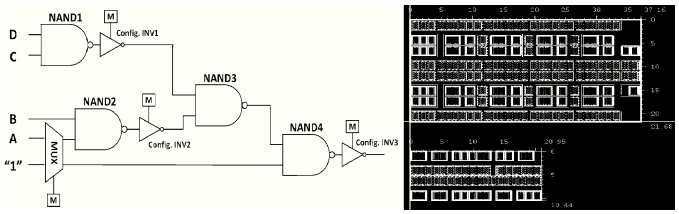


Fig. 1. Proposed logic circuit of ULG and layouts of the ULG and 4-LUT

According to Fig. 1, the ULG can implement the five NPN classes by configuring corresponding configurable inverters to different modes. Configurations for the five functions are summarized in Table III.

TABLE III
CONFIGURATION FOR THE FIVE FUNCTIONS

Logic function	Config. INV1	Config. INV2	Config. INV3	Gate with "1" as input	input
ABCD	inverter	inverter	buffer	NAND4	A,B,C,D
AB(C+D)	inverter	buffer	buffer	NAND4	A,B,$\overline{C}$,$\overline{D}$
AB+CD	buffer	buffer	inverter	NAND4	A,B,C,D
A(B+CD)	buffer	buffer	inverter	NAND2	A,B,C,D
A(B+C+D)	inverter	buffer	inverter	NAND2	A,B,$\overline{C}$,$\overline{D}$

With the permutation of the inputs and negation of the input and output, the proposed ULG can realize all functions covered by the 5 NPN classes. Permutation of inputs can be realized through configuration of routing network in FPGAs in an efficient manner. In order to reduce the overhead, we adopt an optimization method put forward and verified by Ahari et al. [7]. Instead of employing configurable inverters at the inputs of the ULG, the negation is fast forwarded to the next stage, when there are multiple fan-outs where both true and inverted forms are required. Note that the fast forwarding is not realized by simply pushing the negation from the input of the logic element to the output. It is realized in the technology mapping step, where a negation can be implemented by either the current stage or the next stage.

As shown in Fig. 1, the ULG only has four configuration bits. Compared with sixteen configuration bits of a conventional LUT, it can lead to dramatic area and power reduction.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2016.2551555, IEEE Transactions on Circuits and Systems II: Express Briefs

TABLE IV
CHARACTERISTICS OF PROPOSED ULG

Logic cell	delay(ps)	power(μw)	Number of SRAM bits	Number of transistors
ULG	75	0.59	4	62
LUT	195	1.70	16	134

From the layout of LUT and ULG, we can see that the ULG has a much smaller area than the conventional LUT. Compared with LUT, the ULG has 72.9% improvement in the area cost. With the shorter critical path, the performance of ULG is also better than that of conventional LUT. Table IV shows the characteristics of an ULG, extracted by HSPICE simulations using $40nm$ library. Note that the power number shows the total power of each logic cell under the common assumption of FPGA frequency at 200 MHz and input switching activity at 0.2 [24]. The characteristics of the LUT is measured and scaled by a commercial $40nm$ FPGA [24]. Since the LUT has different delay for different inputs, we take the average delay as the delay of the LUT to get stable results.

B. Proposed Complex Logic Block

Based on Table II, we know that the ULG can cover almost 80% of functions in designs. However, the rest of functions are also important for efficient implementation of intact designs. Hence, we propose a new hybrid CLB which contains both ULGs and LUTs. Fig. 2 shows the overall structure of the proposed CLB. A CLB contains two kinds of logic element,

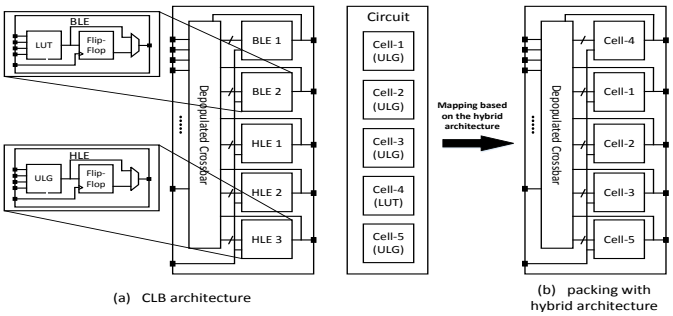


Fig. 2. Structure of proposed CLB and a mapping example

namely basic logic element (BLE) and hybrid logic element (HLE). BLE is made up with a pair of connected LUT and flip-flop, while HLE is made up with a pair of connected ULG and flip-flop. The connection in CLB is realized by a depopulated crossbar, which is widely used in industry [25, 26]. An example is shown in Fig. 2 on how the CLB accommodates the circuit cells. A circuit with five 4-inputs cells needs to be mapped into the hybrid CLB. Among the five cells, cell-1, cell-2, cell-3, and cell-5 can be implemented using ULG or LUT, which means they can be mapped into either BLE or HLE. Cell-4 can only be implemented using LUT and mapped into BLE. Assume that there are two BLEs and three HLEs in the example hybrid architecture as shown in Fig. 2(a). Fig.2 (b) shows the mapping result based on the hybrid architecture where cell-4 is packed into one of the two BLEs while the other four cells are packed into the rest blocks in the CLB.

The structure of depopulated crossbar is shown in Fig. 3. The depopulated crossbar is realized by removing some switch points from the fully populated crossbar at the expense of

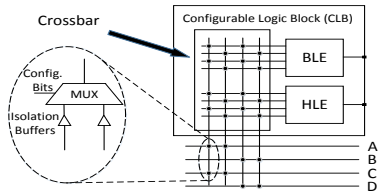


Fig. 3. Structure of depopulated crossbar

flexibility. Fig. 3 shows how to realize permutation through the configurable routing resource. Hence, the BLE and HLE are symmetric in terms of routing. The ratio of LUT:ULG in one CLB is a crucial parameter. We will explore the effect of this important parameter in details in our experiments in Section V.

IV. IMPLEMENTATION FLOW

Because the proposed architecture is significantly different to traditional architectures, we need to develop the corresponding mapping flow for it. Fig. 4 illustrates our mapping flow. First, we use ABC to do technology-independent optimization for input functions. Then, we perform technology mapping with the priority cuts mapper, targeting at 4-input LUT to get netlists composed of LUT cells. Note that we do not do any modification to the ABC in this step to avoid affecting the area and delay optimization goal of the ABC. Next, the LUT function cells are mapped to a netlist of mixed LUTs and ULGs. At last, the netlist is implemented on FPGA using VPR through packing, placement and routing.

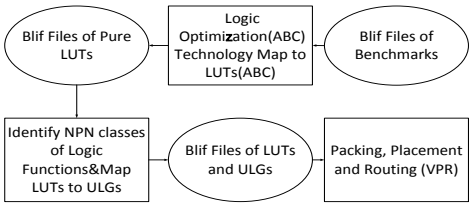


Fig. 4. CAD flow

A. Mapping to ULG

A parser-like tool is developed to map corresponding LUT cells to ULG cells. This parser-like tool takes as input the mapped netlist generated by ABC and calculates the NPN classes of logic functions of each LUT cell. If the logic function of a LUT cell can be implemented using ULG, then the tool maps the LUT cell to an ULG cell. Otherwise, it remains as a LUT cell. Hence, all the LUT cells in netlist generated by ABC that can be implemented by ULG would be mapped to ULG cells.

B. Physical Mapping Using VPR

Because ULG logic cells in netlist can be implemented by both physical blocks of LUT and ULG, while LUT logic cells can only be implemented by physical blocks of LUT, we cannot use traditional specification to specify the architecture. We use the "multiple mode" feature introduced in VTR 7.0 to specify two kinds of physical blocks in our proposed architecture. Physical block of LUT has two modes: one for LUT cells and the other one for ULG cells, while physical

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2016.2551555, IEEE Transactions on Circuits and Systems II: Express Briefs

block ULG only has one mode for ULG cells. The *mode* concept allows the VPR packer to pack both LUT cells and ULG cells into physical blocks of LUT.

Since the packer chooses the candidate cells to be packed in CLB with preference, while randomly placing the cells in the available physical blocks in CLB without preference, there can be a case that in one CLB, the ULG cells with high priority are packed into physical blocks of LUT, while the LUT cells with low priority have no physical blocks of LUT to be packed into. We have made modifications to the parker, which aim to reserve physical blocks of LUT, the scarce and more flexible resource in CLB, to LUT cells which can only be implemented using physical blocks of LUT. After packing the mixed netlist into CLBs, we use the tool chain of placer and router in VPR to implement the CLBs on the hybrid FPGA.

V. Experimental Evaluation

We first use 20 benchmarks in the "MCNC Golden 20" to evaluate our hybrid architecture. Then we use 10 third-party benchmarks to show the effectiveness of our proposed hybrid architecture. The 10 benchmarks are a mixture benchmark set from IWLS 2005 benchmarks and VTR7.0 benchmarks. IWLS 2005 benchmarks [27] was published by International Workshop on Logic and Synthesis (IWLS), which contains diverse circuit designs derived from past conference benchmarks. All the benchmarks are pre-processed with the "ABC" and our self-developed parser.

We have evaluated proposed architecture in details from three aspects, including critical path delay, power and area, and compared the results with baselines, namely pure LUT based architecture and the complementary MCs architecture [7]. The hybrid CLB architecture and the pure LUT based architecture are both modeled using the XML-based VPR architectural language and $40nm$ process technology [28]. Note that the improvement over the complementary MCs design is calculated based on the results in the previous paper, our experiments have the same baseline architecture.

A. Critical Path Delay

Architectures with CLBs having LUT:ULG ratios from 1:9 to 9:1 are created and simulated, which are denoted as Arch. 1:9 to Arch. 9:1. Fig. 5 shows the average critical path delay of the 20 benchmarks with different architectures.

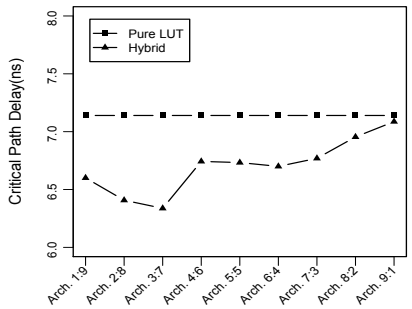
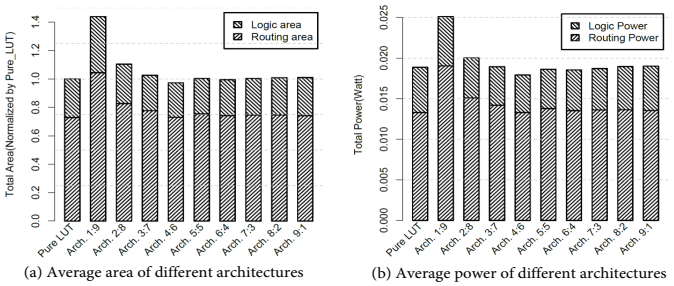


Fig. 5. Average critical path delay of different architectures

Average critical path delay of pure LUT architecture is $7.14ns$. According to Fig. 5, all the average critical path delay of hybrid architectures is shorter than that of pure LUT. The shortest average critical path delay of hybrid architectures is the one with Arch. 3:7, which is $6.34ns$. Compared to pure LUT architecture, it improves the delay by 11.2%. Arch. 3:7 improves the delay by 2.5%, compared with the complementary MCs design. The average critical path delay varies with different LUT:ULG ratios as our expected. For Arch. 1:9 and Arch. 2:8, extra CLBs are used to accommodate LUTs because of lack of LUT resources in CLBs. Therefore critical paths of them are longer than those of architectures with more LUT resources in CLBs. For architectures with ratio larger than 3:7, with increase of the ratio, the percentage of used physical blocks of ULG decreases, and the critical path delay increases correspondingly.

B. Area



(a) Average area of different architectures (b) Average power of different architectures

Fig. 6. Average area and power of different architectures

We adopt the built-in area estimator to measure the area cost [6]. The estimator uses an area model based on the transistor model from Predictive Technology Model (PTM) and parameters extracted from commercial FPGA devices. The average area of the 20 benchmarks with different architectures are shown in Fig. 6(a). We normalize the area numbers by total area of pure LUT architecture. For the logic area, according to Fig. 6(a), the lowest one is achieved with Arch. 4:6. It saves 10.4% logic area compared with pure LUT architecture. For architectures of CLBs with too small percentage of LUTs, such as Arch. 1:9, there are extra CLBs consumed to accommodate LUTs, which incurs more area. For architectures with CLBs having large percentage of LUTs, the percentage of used physical blocks of ULG decreases, which also leads to decrease of area saving. For the routing area, it decreases from Arch. 1:9 to Arch. 4:6, and almost remains unchanged from Arch. 4:6 to Arch 9:1. This is reasonable because the number of CLBs is almost not altered when the ratio of LUTs in CLB is large. For the total area, the lowest one is achieved with Arch. 4:6 too, which saves 2.7% total area compared to pure LUT architecture. As compared to the complementary MCs architecture, Arch. 4:6 saves 17.1% total area.

C. Power

We adopt the power estimator called "VersaPower" to measure the power consumption, which is integrated with VTR framework [6]. Fig. 6(b) shows the average power of the 20 benchmarks with different architectures. All the power is obtained where the period time equals to corresponding critical

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSII.2016.2551555, IEEE Transactions on Circuits and Systems II: Express Briefs

path delay shown above. From Fig. 6(b), we can observe logic power, routing power and total power. For logic power, benchmarks implemented with pure LUT architecture have average logic power as $5.57mw$. As to hybrid architectures, the average logic power decreases from Arch. 1:9 to Arch. 4:6 first, then increases from Arch. 4:6 to Arch. 9:1. And the lowest one is $4.62mw$ achieved with Arch. 4:6. Trend of logic power related to the change of LUT:ULG ratio is similar to that of logic area for the same reasons. Compared with pure LUT architecture, hybrid architecture saves 17.1% logic power. For the total power, benchmarks implemented with pure LUT architecture have average total power of $18.86mw$. The lowest total power of hybrid architectures is $17.93mw$ achieved with Arch. 4:6. Compared with pure LUT architecture, Arch. 4:6 saves 4.9% of the total power. However, the complementary MCs design improves power efficiency by 5.8% compared with Arch. 4:6.

According to the evaluation of proposed architecture from the three aspects, we can see that the proposed hybrid architecture is efficient in all three aspects, which means it can realize holistic efficiency. Among various hybrid architectures with different LUT:ULG ratios, Arch. 4:6 is the most efficient one. Therefore, Arch. 4:6 is a recommendable architecture for general applications. Compared to the complementary MCs architecture, although Arch. 4:6 has less power saving, it has shorter critical path delay. Compared to the complementary MCs architecture, Arch. 4:6 has 3.8% improvement in PDP and 17.1% improvement in area.

D. Third-Party Benchmarks Test

In addition to the "MCNC Golden 20" benchmark set, we also evaluate our proposed architecture with third-part circuits, including five different circuits from widely used IWLS 2005 benchmarks (s832, s1196, s1238, s1488 s1494) [27] and the VTR7.0 benchmarks (i10, sha, sin, stereovision0 and sterovision3). According to the analysis above, Arch. 1:9 to Arch. 5:5 are critical architectures. The results are shown in Fig. 7. For

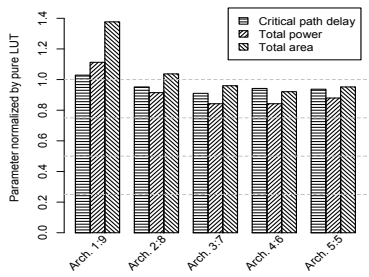


Fig. 7. Results of the third-party benchmarks

simplicity of comparison, the data in the figure is normalized by its corresponding data with the pure LUT architecture. The most efficient performance is achieved at Arch. 3:7, which has 9% improvement compared with the pure LUT architecture. The most efficient area is achieved at Arch. 4:6, which has 18% improvement in logic area compared with the pure LUT architecture. The improvement in total area is around 4%. The architecture with the most efficient power is Arch. 3:7 and Arch. 4:6. They both have 15.7% improvement in total power compared with the pure LUT architecture.

VI. Conclusion

In this brief, we propose a hybrid logic block architecture which realizes holistic efficiency in terms of power, performance, and area. In order to reduce the power consumption, critical path delay and area overhead, we design a holistic efficient ULG. Moreover, in order to keep the generality of logic resources on FPGAs, we propose a hybrid CLB architecture which can exploit the benefit of proposed ULG while keeping the ability to support arbitrary logic functions. The experimental evaluation shows that our proposed architecture design is efficient in power, delay and area.

Acknowledgment

This work is in part supported by a MoE AcRF Tier 2 grant (MOE2012-T2-1-126) in Singapore and the Grant R9336 of the Hong Kong SAR.

References

[1] I. Kuon and J. Rose, "Measuring the gap between fpgas and asics," *TCAD*, vol. 26, no. 2, 2007.
[2] M. B. Taylor, "Is dark silicon useful?: harnessing the four horsemen of the coming dark silicon apocalypse," in *DAC*, 2012.
[3] Y. Okamoto, Y. Ichinomiya, M. Amagasaki, M. Iida, and T. Sueyoshi, "Cogre: A configuration memory reduced reconfigurable logic cell architecture for area minimization," in *FPL*, 2010.
[4] C. Edwards, "A special class of universal logic gates (ulg) and their evaluation under the walsh transform," *International Journal of Electronics Theoretical and Experimental*, vol. 44, no. 1, 1978.
[5] K. McElvain, "Iwls93 benchmark set: Version 4.0," in *Distributed as part of the MCNC International Workshop on Logic Synthesis*, vol. 93, 1993.
[6] J. Luu and et al., "Vtr 7.0: Next generation architecture and cad system for fpgas," *TRETS*, vol. 7, no. 2, 2014.
[7] A. Ahari, B. Khaleghi, Z. Ebrahimi, H. Asadi, and M. B. Tahoori, "Towards dark silicon era in fpgas using complementary hard logic design," in *FPL*, 2014.
[8] D. Chai and A. Kuehlmann, "Building a better boolean matcher and symmetry detector," in *DATE*, 2006.
[9] Berkeley Logic Synthesis and Verification Group, ABC: A System for Sequential Synthesis and Verification, Release 70930. http://www.eecs.berkeley.edu/ alanmi/abc/
[10] P. Jamieson and J. Rose, "Enhancing the area-efficiency of fpgas with hard circuits using shadow clusters," in *FPT*, 2006.
[11] H. Parandeh-Afshar, G. Zgheib, D. Novo, M. Purnaprajna, and P. Ienne, "Shadow aics: Reaping the benefits of and-inverter cones with minimal architectural impact," in *FPGA*, 2013.
[12] Xilinx, "Virtex-6 fpga configurable logic block," http://www.xilinx.com/support/documentation, 2012.
[13] S. Ishihara, M. Hariyama, and M. Kameyama, "A low-power fpga based on autonomous fine-grain power gating," *TVLSI*, vol. 19, no. 8, 2011.
[14] J. H. Anderson and Q. Wang, "Improving logic density through synthesis-inspired architecture," in *FPL*, 2009.
[15] C.-c. Lin, M. Marek-Sadowska, and D. Gatlin, "Universal logic gate for fpga design," in *ICCAD*, 1994.
[16] J. H. Anderson and Q. Wang, "Area-efficient fpga logic elements: architecture and synthesis," in *ASPDAC*, 2011.
[17] C.-C. Lin and M. Marek-Sadowska, "On designing universal logic blocks and their application to fpga design," *TCAD*, vol. 16, no. 5, 1997.
[18] S. A. Chin and J. H. Anderson, "A case for hardened multiplexers in fpgas," in *FPT*, 2013.
[19] J. Cong, H. Huang, and X. Yuan, "Technology mapping and architecture evalution for k/m-macrocell-based fpgas," *TODAES*, vol. 10, no. 1, 2005.
[20] Y. Hu, S. Das, S. Trimberger, and L. He, "Design, synthesis and evaluation of heterogeneous fpga with mixed luts and macro-gates," in *ICCAD*, 2007.
[21] S. A. Chin, J. Luu, S. Huda, and J. H. Anderson, "Hybrid lut/multiplexer fpga logic architectures," *TVLSI*, vol. 24, no. 4, 2016.
[22] S. Thakur and D. Wong, "On designing ulm-based fpga logic modules," in *FPGA*, 1995.
[23] Z. Zilic and Z. G. Vranesic, "Using decision diagrams to design ulms for fpgas," *TC*, vol. 47, no. 9, 1998.
[24] Stratix IV GX, Device Handbook. "Volume 1, SIV5V1-4.1, Altera, 2010."
[25] Altera, "Altera corporation. stratix iv device family overview," http://www.altera.com/literature/hb/stratix-iv/stx4 siv51001.pdf, November 2009.
[26] G. Lemieux and D. Lewis, "Using sparse crossbars within lut," in *FPGA*, 2001.
[27] C. Albrecht, "Iwls 2005 benchmarks," Tech. Rep., 2005.
[28] J. Luu, J. H. Anderson, and J. S. Rose, "Architecture description and packing for logic blocks with hierarchy, modes and complex interconnect," in *FPGA*, 2011.