

FPGA Global Routing Architecture Optimization Using a Multicommodity Flow Approach

Yuanfang Hu, Yi Zhu, Michael B. Taylor, Chung-Kuan Cheng
Department of Computer Science and Engineering, University of California, San Diego
9500 Gilman Dr., La Jolla, CA 92093-0114
{yhu,y2zhu,mbtaylor,kuan}@cs.ucsd.edu

Abstract

Low energy and small switch area usage are two of the important design objectives in FPGA global routing architecture design. This paper presents an improved MCF model based CAD flow that performs aggressive optimizations, such as topology and wire style optimizations, to reduce the energy and switch area of FPGA global routing architectures. The experiments show that when compared to traditional mesh architecture, the optimized FPGA routing architectures achieve up to 10% to 15% energy savings and up to 20% switch area savings in average for a set of seven benchmark circuits.

1 Introduction

Low energy and small switch area usage are two of the important design objectives in FPGA global routing architecture design. In contrast to ASIC design, which connect logic using fixed metal wires, FPGA connect by employing programmable switches. Although these programmable switches bring flexibility to FPGA architectures, they lead to greater energy and on-chip area usage, making FPGAs less favorable in energy-critical applications such as portable devices [3]. In this paper, we study how to effectively reduce energy and switch area usage of FPGA routing architectures through a multicommodity flow (MCF) model based CAD flow.

Topology optimization can effectively reduce energy and switch area of FPGA routing architecture. Traditionally, people adopt a mesh topology for FPGA global routing architectures due to its simplicity. However, as feature sizes shrink and die sizes grow, interconnect energy consumption may become a serious issue for these traditional mesh architectures [9]. Also, as pointed out in [6], mesh routing schemes suffer from unscalable switching area requirements. Therefore, exploring more complex topologies is a

promising approach to effectively optimize the energy and switch area of FPGA routing architectures.

Compared with topology optimization which has been widely studied for many years, wire style optimization emerges only in recent years as a result of rapid advances in signaling interconnect technologies. A few works explored the introduction of multiple signaling technologies to low power network-on-chip (NoC) design [7], as well as communication latency constrained low power NoC design [8].

We integrate both topology and wire style optimizations in our optimization framework to reduce energy and switch area of FPGA routing architecture. Our methodology is based on two MCF models, which synthesize the optimized FPGA global routing architecture with topology and wire style optimizations, and evaluate the optimized FPGA architecture over a set of benchmark circuits, respectively.

The rest of the paper is organized as follows: Section 2 briefly explains the ideas of topology and wire style optimizations. Section 3 first describes our improved CAD flow to generate optimized FPGA global routing architectures, and then explains in details the core components in the design flow, i.e., representative netlist generation and two MCF models. Section 4 presents the experimental results. We summarize our study in section 5.

2 Topology and Wire Style Optimizations

In our work, we perform two types of optimizations to reduce the energy and switch area of FPGA routing architectures. They are topology optimization and wire style optimization.

Wire style optimization studies how to assign various wiring technologies to wire segments in FPGA routing architectures. Recent advances in signaling interconnect technologies, such as wave-pipelined RC wires with repeated buffers, low-swing differential pairs, and on-chip transmission lines, provide us various wiring schemes to optimize

aggressively. These technologies, along with traditional minimal separated RC wires, display different tradeoffs between wire resources and power consumption. For example, on-chip transmission lines usually consume less energy, but with larger routing area. In comparison, traditional minimal separated RC wires occupy less space, but have worse energy efficiency.

Topology optimization is a generalization of the segmentation distribution technique [4], which introduces wire segments of various lengths to FPGA routing architecture. Although it was intensively studied to improve the routability, topology optimization brings more profound impact when combined together with wire style optimization. For example, for on-chip transmission lines, due to the overhead of its transmitter and receiver circuits, it brings energy and speed benefits only for long wires, hence a topology with long links can make better use of such advanced wiring technologies.

To perform topology optimization, we first generate a set of candidate topologies and put them in the topology library. The topologies of our optimized FPGA routing architectures are selected from the topology library. The library can be easily expanded by importing valuable candidate topologies.

Topology library generation is one of the key issues to the success of our FPGA routing architecture optimization. Even after clustering the look-up tables (LUTs) into larger logic blocks, there are still a huge number of possible topologies. For example, for an FPGA with 10×10 logic blocks, each row or column has $2^{C(10,2)} = 2^{45}$ different connections, and the whole FPGA chip has $(2^{45})^{20} = 2^{900}$ different connections. It is impossible to explore them exhaustively with the current computation technology.

To reduce the size of topology library and only keep the most valuable and promising topologies, we make a few assumptions without loss of generality. First, we assume all wire segments have lengths of power of two, i.e. there are only wires in lengths of 1, 2, 4, 8, etc. Second, on each row or column, wire segments should repeat themselves consecutively along the whole routing channel. Third, all rows and columns should have identical connections. The reasoning of these assumptions is that: at the stage of FPGA global routing architecture design, the target applications are still unknown, therefore it is reasonable to design relatively regular and symmetric topologies to fit potential applications. Based on the above assumption, we exhaustively generate all qualified candidate topologies.

Figure 1 shows an example of topology and wire style optimizations in FPGA routing architectures. The blocks are logic blocks (LBs). There are wire segments of various lengths, and different wire segments can be implemented with different wire styles.

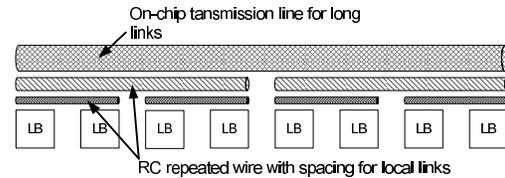


Figure 1. Topology and Wire Style Optimizations in FPGA Routing Architectures

3 Design Methodology

In our work, an MCF model based optimization framework, which integrates both topology and wire style optimizations, is introduced to an existing CAD flow. The optimization framework takes each candidate topology and available wire styles as inputs, and produces optimized capacities and wire style assignments for wire segments in FPGA routing architecture. By repeating this process for all candidate topologies in topology library, we can obtain the best topology with wire style optimization as our optimized FPGA routing architecture.

3.1 An Improved CAD Flow

Figure 2 shows our improved CAD flow. The inputs are a set of benchmark circuits. First, we use SIS [10] to perform technology independent logic optimization on each of the circuit. Next, these circuits are technology-mapped by FlowMap [5] into four-input LUTs (4-LUTs). We then use VPack [2] to pack these 4-LUTs into larger logic blocks. The resulting netlists are then fed into VPR [2], and are placed on the FPGA chip. So far, the steps are the same as in the existing design flow.

The shadowed steps in Figure 2 are our improved parts. We have a netlist generator to generate the representative netlist by extracting the characteristics of the input benchmark circuits. The representative netlist reflects the traffic distribution of the benchmark circuits, hence effectively guide the design of FPGA routing architectures to fit for the largest class of the benchmarks. Meanwhile, we use a topology generator to generate a set of candidate FPGA topologies. The representative netlist and the candidate FPGA topologies are then fed into the MCF interconnection synthesis tool, which models the FPGA routing optimization problems with specific objectives, such as energy or switch area usage. The output of the MCF formulations will be the optimized FPGA global routing architectures with topology and wire style optimization. In the last step, the benchmark circuits are fed to the optimized routing architectures, and a MCF routing evaluation model is used to evaluate the actual improvement.

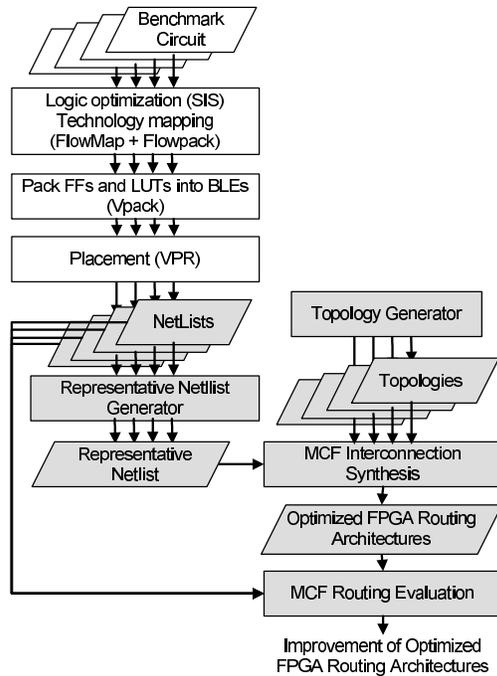


Figure 2. An Improved CAD Flow for FPGA Routing Architecture Optimization

Compared to the traditional CAD flow in FPGA design, our improved design flow is able to automatically generate candidate global routing architectures. This largely increases the flexibility of global routing architectures and explores a much larger design space.

In our improved CAD flow, there are two major components. One is a netlist generator to generate representative netlist, the other core component includes two MCF models, MCF interconnection synthesis and MCF routing evaluation. We describe each of them in the following sections.

3.2 Representative Netlist Generation

To achieve the best benefits of our FPGA routing optimizations, we need to have a good understanding of the nature of communications of our applications. The performance of both topology and wire style optimizations largely depends on the underlying communication pattern. Therefore, we generate a representative netlist from a set of FPGA benchmark circuits. The generated representative netlist should catch the characteristics of the benchmark circuits. The representative netlist generation is based on the statistical analysis of the candidate application circuits. Three sets of key parameters are to be determined. First, how many nets should the netlist have? Second, what is the size, e.g.

the number of pins, of each net? At last, what are the pin locations of each net?

In our work, we set the size of the representative netlist to be the maximum netlist size among all the benchmark circuits. Because a netlist with larger size usually requires more routing capacities to route, it is intuitive to assign the representative netlist with the maximum routing capacity requirements among the benchmarks, so that the optimized FPGA routing architectures can be reconfigurable to accommodate all benchmark circuits.

To determine the size of each net, we first count the net size of all the benchmark circuits, and calculate their distribution. Then we design the size of each net in the representative netlist to match this distribution pattern. For example, if 5% of nets in benchmark circuits have number of pins in the range from 30 to 35, and if the size of our representative netlist is 1000, we should evenly distribute 50 nets with pins in that range.

Finally, we need to determine the pin locations of each net. Random generation of pin locations may lose the intrinsic communication patterns of the benchmark circuits. Therefore, we analyze the distribution of the frequency of each pin in the candidate circuits, and generate a corresponding “pin pool” with frequency distribution for each pin in the pool. Then for each net, we pick pins from “pin pool” according to their frequency function. We determine the distance among pins by a *geometry distribution function*. The function is defined as the probability of the distance between two pins decreases exponentially with increasing distance, i.e., $P(k) = p(1-p)^k$, $k = 1, 2, \dots$ where k is the distance between two pins, p is the probability of links with distance 1, and $P(k)$ is the probability of links with distance k .

3.3 MCF Interconnection Synthesis and Routing Evaluation

As shown in our improved CAD flow, at the core of our optimization framework are two MCF models. One is the MCF interconnection synthesis model, which generates the optimized FPGA routing architectures with topology and wire style optimizations for representative netlist. The other model is the MCF routing evaluation, which evaluates the actual performance of these optimized FPGA routing architectures for the target set of benchmark circuits.

The major difference between these two MCF models lies in their constraints. For MCF interconnection synthesis, the capacity of each routing channel is unknown, hence it regards on-chip area resources of the routing channel as its constraints, and generates optimized capacities for each routing channel. The constraints for MCF routing evaluation are these output routing capacities.

The following subsections describe these two models in

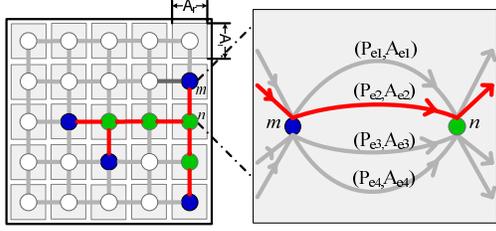


Figure 3. An Improved CAD Flow for FPGA Routing Architecture Optimization

detail. First, we show how to integrate the wire style optimization into the MCF models. Then, we present the formulations for each MCF model with various design objectives in mind. Finally, we briefly describe the algorithms that efficiently solve these MCF models.

3.3.1 Integration of Wire Style Optimization

Assume an FPGA chip with $n \times n$ logic blocks. These logic blocks communicate with each other through $n \times n$ switch boxes at the intersection of the channels. A topology is defined as a bi-directed graph $G = (V, E)$, where, each node $v_i \in V$ represents a switch box, and each edge $e_{i,j} \in E$ represents routing tracks between switch boxes i and j . These wire tracks can be implemented with multiple wire styles. Assume there are k nets. For each net i , its communication demand is $d_i = 1$. Let t_i be the set of paths on Steiner trees to connect net i , and let $\mathbf{T} := \cup_i t_i$. Variable $f(t)$ denotes the amount of flow along Steiner tree t , for every $t \in \mathbf{T}$.

Figure 3 demonstrates an example on how to integrate multiple wire styles into MCF models. In a mesh architecture, we have a net of 4 pins (black nodes) to be routed. We connect these pins using a minimum Steiner tree (grey nodes are Steiner nodes), as shown in dark lines in left side of the figure. Then we use multiple edges to represent available wire styles, as shown in right side of the figure. For link (m, n) , there are 4 edges from node m to node n , which represents 4 types of candidate wire styles. A pair (P_e, A_e) is associated with each edge e . P_e is per bit energy on edge e . A_e is the wire pitch. If there is flow goes through edge 2 (as shown in the dark line), it means wire style 2 is selected for link (m, n) , and the capacity of edge 2 equals to the amount of the flow. Therefore, if we solve the MCF formulations and get the flow distribution, we can obtain the optimized global routing architecture with wire style optimization.

3.3.2 MCF Interconnection Synthesis

Different FPGA optimization problems correspond to different MCF interconnection synthesis formulations. MCF

model has the flexibility to adapt to various design objectives. In our work, we study three types of optimization problems, focusing on the energy optimization, the switch area optimization, and their co-optimization. These optimization problems are important and of the interests in modern FPGA routing architecture design.

For the first problem, we optimize the energy of the FPGA routing architecture. To estimate energy, for each edge e , we assume that P_e represents bit energy on link e and the corresponding switch box.

$$P_e = P_w + P_{sb}$$

where P_w and P_{sb} are bit energy on interconnects and switch box, respectively. When a flow of amount f goes through the edge and the corresponding switch box, the total energy is $P = P_e \cdot f$

P_{sb} can be estimated by

$$P_{sb} = P_s \cdot N_s$$

where P_s is energy for a single switch in a switch box, and N_s is the total number of switches in a switch box. Assume F_s is number of switches connected to each wire entering a switch box, and f is the amount of flow go through a switch box, we have:

$$N_s = 1/2 \cdot F_s \cdot f$$

The following is the formulation for MCF synthesis on energy optimization. The objective is to minimize the total energy of the routing architecture, which is the sum of per-bit energy on all routing tracks (as in Equation (1)). We have two constraints. The routability constraint (2) requires that all the nets in the representative netlist should be routable, while the routing area constraint (3) ensures that when we route the nets, the routing area usage cannot exceed the available on-chip area resources on the vertical or horizontal dimension A_r .

$$\text{Min} : \sum_{j=1}^k \sum_{t \in T_j} \sum_{e \in t} f(t) \cdot P_e \quad (1)$$

$$\text{s.t.} \quad \forall 1 \leq j \leq k : \sum_{t \in T_j} f(t) \geq 1 \quad (2)$$

$$\forall q : \sum_{e \in \text{Grid}(q)} A_e \cdot \sum_{t: e \in t} f(t) \leq A_r \quad (3)$$

$$\forall t : f(t) \geq 0 \quad (4)$$

The outputs of the MCF synthesis model are the optimized FPGA global routing architectures with expected design objectives, in this case, expected energy on routing architectures. Notice that the variables in the formulations

are $f(t)$. After we solve the MCF formulations, we can obtain the optimized capacity for each edge by calculating the accumulated $f(t)$ on that edge. In this way, we have the optimized FPGA routing architecture for a certain topology. Then we can repeat this process for each candidate topology and generate the optimized FPGA routing architectures with topology and wire style optimizations.

In the second case, we optimize the total switch area of switch boxes. Since the switch area is proportional to the number of switches, we try to minimize the total number of switches in switch boxes as our design objective. The constraints of this problem are exactly the same as those of the first case, i.e., the routability and routing area constraints. Therefore, we omit the constraints part of the formulations, and give only the objective function as follows, in which energy parameters P_e are simply replaced by switch quantity parameters N_s .

$$Min : \sum_{j=1}^k \sum_{t \in T_j} \sum_{e \in t} f(t) \cdot N_s \quad (5)$$

Furthermore, such MCF synthesis model can be easily applied to study the tradeoffs between multiple design factors. In our third case, we study the switch area constrained energy optimization problem, which means we optimize the energy of the FPGA routing architecture, while at the same time satisfying all requirements on total switch area usage. Compared with the first energy optimization problem, this problem has one more switch area constraint (6), where A_s is the given switch area budget. The objective function and the other constraints are exactly the same as formulation (1), (2), and (3).

$$\sum_{j=1}^k \sum_{t \in T_j} \sum_{e \in t} f(t) \cdot N_s \leq A_s \quad (6)$$

3.3.3 MCF Routing Evaluation

As we explained earlier, MCF routing evaluation model differs from MCF interconnection synthesis model only on that one of its constraints is routing channel capacity instead of on-chip routing area resources. Take energy optimization problem as an example, its MCF routing evaluation formulations are as follows, where $c(e)$ represents the capacity of edge e .

$$Min : \sum_{j=1}^k \sum_{t \in T_j} \sum_e f(t) \cdot P_e \quad (7)$$

$$s.t. \quad \forall 1 \leq j \leq k : \sum_{t \in T_j} f(t) \geq 1 \quad (8)$$

$$\forall e : \sum_{t: e \in t} f(t) \leq c(e) \quad (9)$$

$$\forall t : f(t) \geq 0 \quad (10)$$

The outputs of MCF routing evaluation model are the actual design results, such as total energy, for each of benchmark circuits. This evaluation process verifies the effectiveness of MCF interconnection synthesis model.

3.3.4 Algorithms to Solve MCF models

We use the polynomial time approximation algorithms to quickly solve the two MCF models. For MCF interconnection synthesis model, the algorithm is similar to that in [1]. For MCF routing evaluation model, the algorithm is slightly different, and was presented in [8]. Both algorithms are based on LP primal-dual theory and can obtain $(1 + \epsilon)$ optimal solution in polynomial time. The core idea in the algorithms is to iteratively perform minimum Steiner tree algorithm to update dual and primal solutions, so that the gap between them can finally reduce to below the error bound. Due to the limit of space, We do not discuss details in this paper.

4 Experimental Results

In our experiments, we use seven MCNC benchmark circuits [11] with moderate sizes. We first perform technology mapping to map these benchmark circuits to 4-LUTs. Then, we pack every 16 4-LUTs into a larger logic blocks, and finally place these logic blocks on island-style FPGA chip. Table 1 shows the size of resulting representative netlists of these seven benchmark circuits. Since the size of switch box array ranges from 10×10 to 11×11 , the representative netlist is of size 11×11 . We set p to be 0.1 in the geometry distribution function $f(k) = p(1 - p)^k$ in representative netlist generation, because we observe it best match the connection nature of our benchmark circuits.

Table 1. Size of Representative Netlist of MCNC Benchmark circuits

	alu4	apex4	diffeq	dsip	ex5p	misex3	tseng
size	11x11	10x10	11x11	11x11	10x10	11x11	10x10
# of nets	621	798	945	593	745	771	788

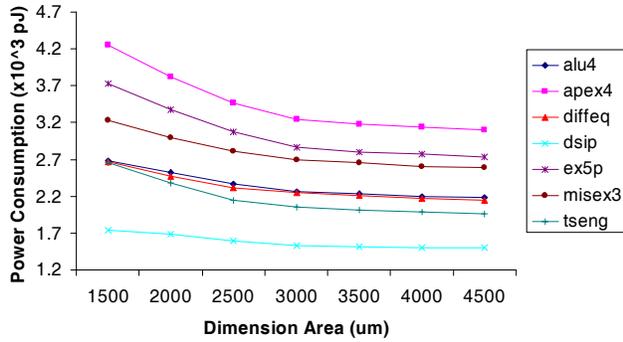


Figure 4. Energy of Benchmark Circuits under Various Routing Area Constraints

We generate the candidate topologies using the topology generator described in section 2. In our experiments, we assume the available segment lengths are 1, 2, 4, and 8. Segment of length 1 is mandatory, while other three types of segments are optional. Abiding such assumptions, for FPGA of size 11×11 , the total number of generated candidate topologies is 93.

We assume 4 types of candidate wires, RC wires with $1 \times$, $2 \times$ and $4 \times$ minimum global pitch and transmission line with $10 \times$ minimum pitch. These wire styles have decreasing energy consumption but occupy increasing on-chip routing area.

In our MCF approximation algorithms, we set error tolerance ϵ to 1%. All of the following experiments are based on $0.18 \mu m$ design technology. Since each grid has the same vertical and horizontal dimension, for convenience, we use only the vertical dimension to represent the area budget, therefore the unit of area in our experiments is um .

4.1 Energy Optimization

We first demonstrate the impact of the available on-chip routing resources on our energy optimization. Then we compare our optimized routing architectures with traditional mesh architecture to show the improvement from the energy optimization.

4.1.1 Optimized Energy under Various Routing Area Constraints

Figure 4 shows the energy of seven benchmark circuits on our optimized FPGA routing architectures. The x-axis is routing area budgets from $1500um$ to $4500um$, which represents the area constraint from tightest to loosest. The y-axis is energy in unit $\times 10^3 pJ$. As area constraints become looser, energy of all benchmark circuits keep decreasing,

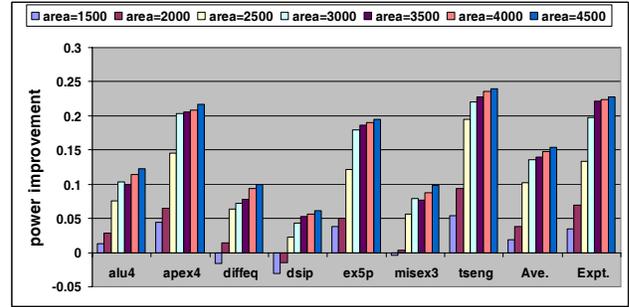


Figure 6. Improvement of Energy of Optimized Architectures Over Mesh Architecture

where circuit *apex4* gains the largest improvement of 27.1% (from 4.26 to $3.11 \times 10^3 pJ$). The improvements are from both topology and wire style optimizations, since as routing area budgets increase, more energy-efficient but area consuming wires can be adopted in corresponding topologies to reduce the overall energy of FPGA routing architectures.

Figure 5 shows the detailed topology and wire style assignments in optimized FPGA routing architectures under various routing area constraints. The black blocks represent the switch box arrays, and different wire styles are in different colors. Figure 5 (a) is when the routing area is $1500um$, and Figure 5 (b) is when the routing area is $4500um$. We observe that in (a) the $1 \times$ RC wires are used for most of the connections to save the area usage. Also at the outer regions of the chip, some energy-efficient transmission lines are used to reduce energy, because in those regions the communication flow is not as congested as in the center of the chip, consequently there is room for wire style optimization. In (b), since now we have abundant routing area for wire style optimization, transmission lines are adopted for all the long links, and RC wire with $4 \times$ minimum pitch are adopted for those short links. As a result, the energy of (b) architecture is 20% less than that of (a) architecture. The bottom of the figure shows the topologies of the corresponding FPGA architectures.

From Figure 5, we see a clear trend to adopt wider wires as routing area budget increases to reduce energy. Also wires at the center of the chip usually are more area-efficient than wires at the outer regions of the chip.

4.1.2 Energy Improvements Over Traditional Mesh Architecture

We compare the energy of our optimized FPGA routing architectures with that of traditional mesh routing architecture. Figure 6 shows the energy improvement in percentage. In x-axis, each group of bars present the energy under var-

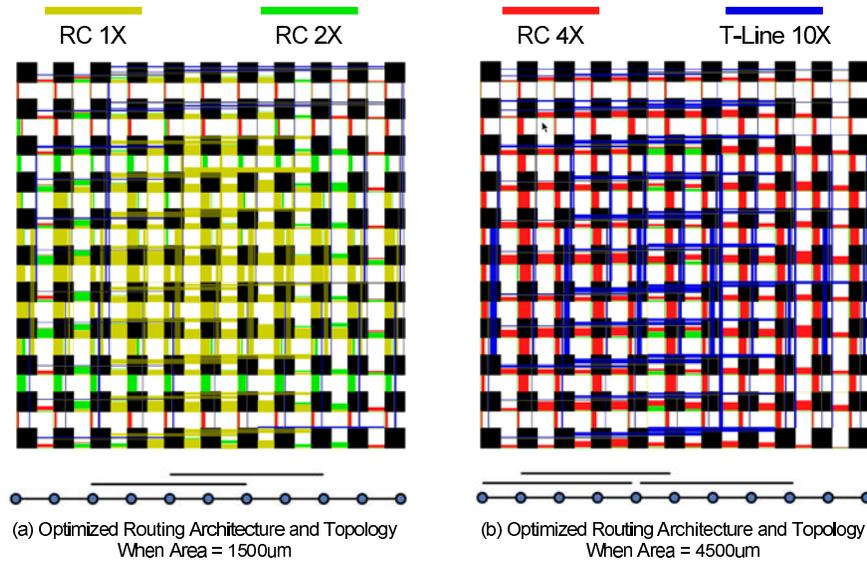


Figure 5. Optimized FPGA Routing Architectures and Corresponding Topologies (a)When routing area constraint is tight (b) When routing area constraint is loose

ious area constraints for a certain benchmark circuit. The last bar is energy for representative netlist from MCF interconnection synthesis model, which indicates the estimated improvement of our design.

Circuit *disp* has the smallest improvement, ranging from -3% to 6%; circuit *tseng* has the largest improvement from 5% to 24%. In average, our optimized routing architecture can achieve energy savings from 2% to 15% over mesh architecture. When area budget is small, such as 1500um, our optimized routing architecture has no obvious advantages over traditional mesh architecture, because we do not have enough routing area to adopt better wiring technologies. The major improvement occurs When area budget increase from 1500um to 2500um. Further increasing of routing area budget does not bring too much benefits.

4.2 Switch Area Optimization

Our methodology can be easily applied to various design objectives. In this experiment, we optimize switch area of FPGA routing architectures. We use number of total switches in switch box as objective. Since the total number of switches is not affected by wire styles, the routing area constraint is not a major issue in switch area optimization. Figure 7 shows the switch area improvement when compared to mesh architecture for the seven benchmark circuits. In average, 15% to 20% switch area improvement can be seen.

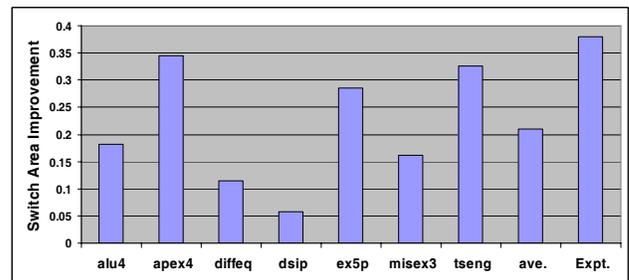


Figure 7. Improvement of Switch Area of Optimized Architectures Over Mesh Architecture

4.3 Switch Area Constrained Energy Optimization

In this experiment, we study the tradeoffs between energy and switch area optimizations in our unified optimization framework. Figure 8 depicts the optimized energy under various switch area constraints. The x-axis represents the number of switches in switch boxes. The y-axis is energy in unit $\times 10^3 pJ$. Each curve represents the energy of the representative netlist under given routing area budget.

As the number of switches increases, energy decreases because the communication flow can be routed to more energy-efficient paths, which may have high switch costs.

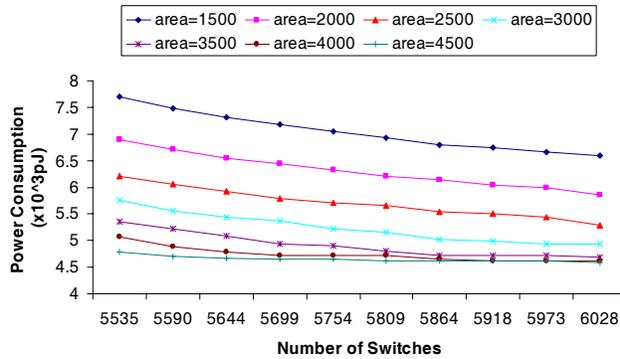


Figure 8. Switch Area Constrained Low Power Optimization for FPGA Routing Architectures

An interesting observation is that when the routing area budget is less, changing the switch area budget has larger impact on energy. For example, when changing number of switches from minimum to maximum, the energy changes by 16.7% for the curve area=1500um, which is only 4.6% for the curve area=4500um. This is because a tighter routing area budget necessitates the use of narrow but energy-costly wires, leaving a larger space for wire style optimization. When the routing area budget is abundant, the energy is already quite optimized no matter the switch area constraints.

5 Conclusions

In this paper, we present an improved MCF model based CAD flow to perform aggressive optimizations, such as topology and wire style optimizations, to reduce the energy and switch area of FPGA global routing architectures. The experiments show that when compared to traditional mesh architecture, our optimized architectures achieve up to 10% to 15% power savings and up to 20% switch area savings in average for a set of seven benchmark circuits. As future work, we can apply the methodology to other design objectives, such as interconnect delay in FPGA global routing architectures.

6 Acknowledgment

We thank Dr. Mike Hutton at Altera Corp. for his valuable discussion on the original idea of the work in this paper. We also thank California MICRO funding and the support of Altera.

References

- [1] C. Albrecht, "Provably Good Global Routing by A New Approximation Algorithm for Multicommodity Flow," *International Symposium on Physical Design*, pp.19-25, 2000.
- [2] V. Betz, and J. Rose, "VPR: A New Packing, Placement and Routing Tool for FPGA Research," *International Workshop on Field-Programmable Logic and Applications*, pp.213-222, 1997.
- [3] V. Betz, J. Rose, and A. Marquardt, "Architecture and CAD for Deep-Submicron FPGAs," *Kluwer Academic Publishers*, February, 1999.
- [4] S. Brown, M. Khellah, and G. Lemieux, "Segmented Routing for Speed-Performance and Routability in Field-Programmable Gate Arrays," *Journal of VLSI Design*, vol. 4, no. 4, pp.275-291, 1996.
- [5] J. Cong, and Y. Ding, "FlowMap: An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, pp.1-12, January, 1994.
- [6] A. DeHon and R. Rubin, "Design of FPGA Interconnect for Multilevel Metallization," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 12, No. 10, pp. 1038-1050, October, 2004.
- [7] Y. Hu, H. Chen, Y. Zhu, A. A. Chien, C.K. Cheng, "Physical Synthesis of Energy-Efficient NoCs Through Topology Exploration and Wire Style Optimization," *International Conference on Computer Design*, pp.111-118, October, 2005.
- [8] Y. Hu, Y. Zhu, H. Chen, R.L. Graham, and C.K. Cheng, "Communication latency aware low power NoC synthesis," *Design Automation Conference*, pp.574-579, June, 2006. 995.
- [9] K. Poon, S. Wilton and A. Yan, "A Detailed Power Model for Field Programmable Gate Arrays," *ACM Transactions on Design Automation of Electronic Systems*, Vol. 10, Issue 2, pp. 279-302, April, 2002.
- [10] E.M. Sentovich, et al, "SIS: A System for Sequential Circuit Analysis," *Technical Report No. UCB/ERL M92/41, University of California, Berkeley*, 1992.
- [11] S. Yang, "Logic Synthesis and Optimization Benchmarks, Version 3.0," *Technical Report, Microelectronics Center of North Carolina*, 1991.