Delay-Optimal Technology Mapping for FPGAs with Heterogeneous LUTs

Jason Cong and Songjie Xu
Department of Computer Science
University of California, Los Angeles, CA 90095
cong@cs.ucla.edu sxu@cs.ucla.edu

Abstract

In order to maximize performance and device utilization, recent generation of FPGAs take advantage of speed and density benefits resulted from *heterogeneous* FPGAs, which provide either an array of homogeneous PLBs, each configured to implement circuits with LUTs of different sizes, or an array of physically heterogeneous LUTs. LUTs with different sizes usually have different delays. This paper presents the *first* polynomial-time optimal technology mapping algorithm, named HeteroMap, for delay minimization in heterogeneous FPGA designs. Taking different delays of heterogeneous LUTs into consideration, the HeteroMap algorithm computes the minimum mapping delay of a circuit based on a series of minimum height K-feasible cut computations at each node in the circuit. For a heterogeneous FPGA consisting of K_1-LUTs, K_2-LUTs, $\ldots$, and K_c-LUTs, HeteroMap computes the minimum delay mapping solution in $O(\sum_{i=1}^{c} K_i \cdot n \cdot m \cdot \log n)$ time for a circuit netlist with n gates and m edges. HeteroMap also effectively minimizes the area of the mapping solution by maximizing the volume of each cut and by the post-mapping packing operations. We have tested HeteroMap on MCNC benchmarks on Xilinx XC4000 series FPGAs [Xi97] which can implement 4-LUTs and 5-LUTs, and Lucent ORCA2C series FPGAs [Or96] which can implement 5-LUTs and 6-LUTs. The experimental results show that for XC4000 series FPGAs, HeteroMap can reduce 19% of the mapping delays and 7% of the post-layout delays with only 2% increase on the PLB numbers, compared with the results of FlowMap [CoDi94a]. For ORCA2C series FPGAs, HeteroMap reduces 9% of the mapping delays and 8% of the PLB number over FlowMap. Furthermore, the optimality of the HeteroMap algorithm enables us to quantitatively evaluate various heterogeneous architectures without the bias of mapping heuristics. A set of architecture study results of heterogeneous FPGAs are presented in this paper.

1 Introduction

The short design cycle and low non-recurring engineering (NRE) cost have made the field programmable gate arrays (FPGAs) an important technology in VLSI designs. In a traditional LUT-based FPGA device, the basic programmable logic block is a K-input lookup table (K-LUT) which can implement any Boolean function of up to K variables. In order to maximize performance and device utilization, recent generation of FPGAs take advantage of speed and density benefits resulted from *heterogeneous* FPGAs, which provide either an array of homogeneous PLBs, each configured to implement circuits with LUTs of different sizes, or an array of physically heterogeneous LUTs. For example, Xilinx XC4000 series FPGAs [Xi97] incorporate two 4-LUTs wired to a single 3-LUT in one programmable logic block (PLB)[1] such that each PLB can be configured to implement either two seperate 4-LUTs or a single 5-LUT, in addition to other configuration possibilities. In ORCA2C series FPGAs from Lucent Technologies [Or96], each PLB[2] can be configured to implement either two seperate 5-LUTs or a single 6-LUT, besides other configuration possibilities. Note that both XC4000 and ORCA2C devices provide heterogeneous LUTs through reconfiguration instead of presenting a physical array of LUTs of different sizes directly.

In a heterogeneous FPGA chip, larger LUTs can certainly cover more gates, but usually have longer delay. Therefore, an important problem is how to utilize the heterogeneous LUTs to minimize the overall circuit delay and/or area. For example, the circuit depicted in Figure 1(a) can be mapped into five homogeneous 4-LUTs, each of delay 1, with the total mapping delay of 3 (see Figure 1(b)). In Figure 1(c), the same circuit can be mapped into two 4-LUTs assuming each of delay 1 and one 5-LUT assuming of delay 1.5, with a total mapping delay of 2.5. In general, given a heterogeneous FPGA with LUTs of different sizes, we want to find an optimal mapping solution with the minimum delay or area.

In the past few years, extensive studies have been done on technology mapping for homogeneous LUT-based FPGAs. The previous LUT-based FPGA mapping algorithms can be roughly divided into four classes, area minimization (such as [MuNS90] [MuSB91b] [FrRC90] [FrRV91a]), delay minimization (such as [MuSB91a] [FrRV91b] [CoDi94a] [CoDG93] [YaWo94]), simultaneous area and delay minimization ([CoDi93a] [CoHw95a]) and routability optimization (such as [BhHi92] [ScKC92]). More comprehensive survey of existing FPGA mapping algorithms can be found in [CoDi96]. However, none of these algorithms are able to deal with the delay optimization problem for heterogeneous FPGAs, as they assume the same capacity and delay for logic blocks. In [HeRo94], an approach for technology mapping into heterogeneous LUT-based FPGAs was presented for area minimization, but their architecture assumes a mixture of only two types of LUTs with a fixed ratio in one FPGA chip, and

[1] in Xilinx XC4000 series FPGAs, PLB is called *configurable logic block (CLB).*

[2] in Lucent ORCA2C series FPGAs, PLB is called *programmable function unit (PFU).*

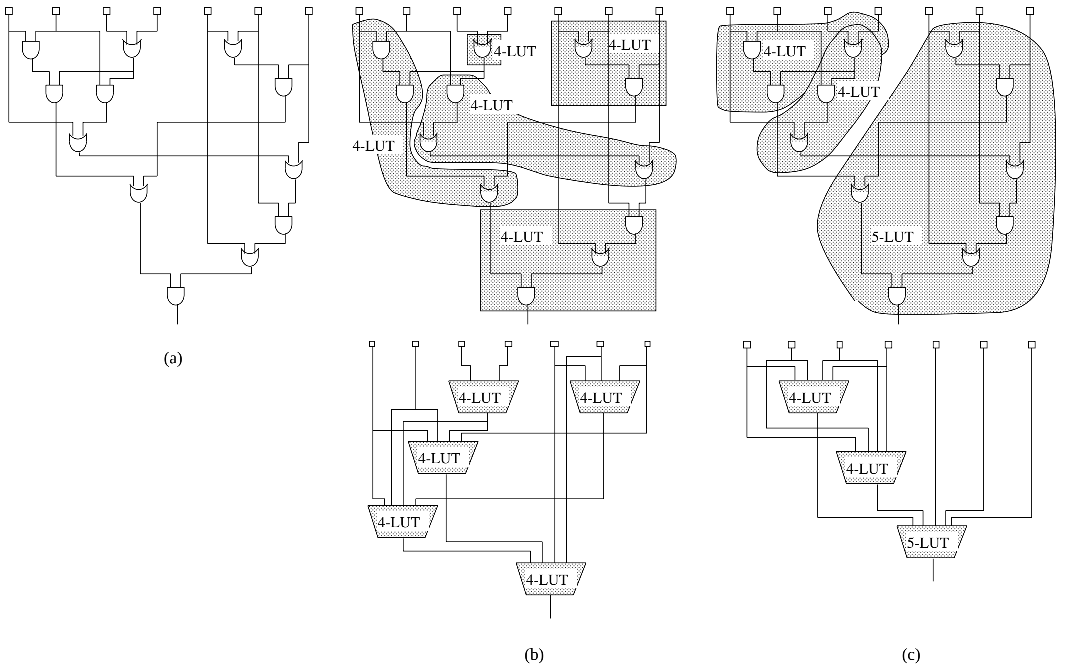


Figure 1: (a) The orginal circuit; (b) Mapping to uniform 4-LUTs by Flowmap; (c) Mapping to 4-LUTs and 5-LUTs by HeteroMap.

their algorithm can not be easily extended to deal with the area minimization problem for the general heterogeneous FPGAs employing three or more types of logic blocks.

In this paper, we propose a general technology mapping problem for heterogeneous FPGAs and present a polynomial-time optimal technology mapping algorithm, named HeteroMap, for delay minimization in heterogeneous FPGA designs. Taking different delays of heterogeneous LUTs into consideration, the HeteroMap algorithm computes the minimum mapping delay of a circuit based on a series of minimum height K-feasible cut computations at each node in the circuit. For a heterogeneous FPGA consisting of K_1-LUTs, K_2-LUTs, ..., and K_c-LUTs, HeteroMap computes the minimum delay mapping solution in $O(\sum_{i=1}^{c} K_i \cdot n \cdot m \cdot \log n)$ time for a circuit netlist with n gates and m edges. HeteroMap also effectively minimizes the area of the mapping solution by maximizing the volume of each cut and by the post-mapping packing operations. We have tested HeteroMap on MCNC benchmarks on Xilinx XC4000 series FPGAs [Xi97] which can implement 4-LUTs and 5-LUTs, and Lucent ORCA2C series FPGAs [Or96] which can implement 5-LUTs and 6-LUTs. The experimental results show that for XC4000 series FPGAs, HeteroMap can reduce 19% of the mapping delays and 7% of the post-layout delays with only 2% increase on the PLB numbers, compared with the results of FlowMap [CoDi94a]. For ORCA2C series FPGAs, HeteroMap reduces 9% of the mapping delays and 8% of the PLB number over FlowMap. Furthermore, the optimality of the HeteroMap algorithm enables us to quantitatively evaluate various heterogeneous architectures without the bias of mapping heuristics. A set of architecture study results of heterogeneous FPGAs are presented in this paper.

The remainder of this paper is organized as follows. Section 2 presents the problem formulation and preliminaries. Section 3 describes our delay optimal technology mapping algorithm for heterogeneous FPGAs. Experimental results and comparative study are presented in Section 4. Section 5 concludes the paper.

2 Problem Formulation and Preliminaries

A Boolean network N can be represented as a directed acyclic graph (DAG) where each node represents a logic gate[3], and a directed edge (i, j) exists if the output of gate i is an input of gate j. Primary input (PI) nodes have no incoming edge and primary output (PO) nodes have no outgoing edge. We use $input(v)$ to denote the set of fanins of gate v. Given a subgraph H of the Boolean network, let $input(H)$ denote the set of *distinct* nodes outside H which supply inputs to the gates in H. For a node v in the network, a *K-feasible cone at v*, denoted C_v, is a subgraph consisting of v and its predecessors[4] such that $|input(C_v)| \leq K$ and any path connecting a node in C_v and v lies entirely in C_v. The *level* of a node v is the length of the longest path from any PI node to v. The level of a PI node is zero. The *depth* of a network is the largest node level in the network. A boolean network is *K-bounded* if $|input(v)| \leq K$ for each node v in the network.

A general LUT-based heterogeneous FPGA consists of c types of LUTs of K_1-LUT, K_2-LUT, $\ldots$, and K_c-LUT ($K_1 < K_2 < \ldots < K_c$) with the delays d_1, d_2, $\ldots$, and d_c ($d_1 < d_2 < \ldots d_c$, and d_1, d_2, $\ldots$, d_c may not be integer). Usually, d_1 is scaled to 1 during technology mapping. Homogeneous FPGAs can be viewed as the special heterogeneous FPGA with only one type of LUTs.

For a circuit mapped into a heterogeneous FPGA, we assume different access delays for heterogeneous LUTs but a constant delay for the interconnection[5], which we call *heterogeneous LUT-delay model*. The unit-delay model used in [CoDi94a] is a special case of heterogeneous LUT-delay model in homogeneous FPGAs.

Given these definitions, the technology mapping problem for heterogeneous FPGAs can be formulated as Problem 1.

Problem 1 *Given a K_1-bounded Boolean network N and the heterogeneous FPGA, transform N to an equivalent LUT network N' which consists of K_1-LUTs, K_2-LUTs, $\ldots$, and K_c-LUTs such that*

[3]In the rest of the paper, *gate* and *node* are used interchangeably for Boolean networks.

[4]u is a predecessor of v if there is a directed path from u to v.

[5]The constant interconnection delay can be counted into the LUT delays such that the interconnection delay can be set to zero.

the circuit delay and/or area are minimized.

In this paper, our primary objective is to minimize the circuit mapping delay under the heterogeneous LUT-delay model through technology mapping. Therefore, a mapping solution is said to be *optimal* if the mapping delay is minimized. The secondary objective is to reduce the area used in the technology mapping solution. Figure 1(b) shows an example of mapping a combinational circuit into uniform 4-LUTs, which is done by FlowMap [CoDi94a], while Figure 1(c) illustrates how the circuit is to be mapped into a heterogeneous FPGA with both 4-LUTs and 5-LUTs, which can be achieved by the HeteroMap algorithm presented in Section 3.

Given a network $N = (V(N), E(N))$ with a source s and a sink t, a $cut(X, \bar{X})$ is a partition of the nodes in $V(N)$ such that $s \in X$ and $t \in \bar{X}$. The *node cut set of* of $(X, \bar{X})$, denoted $C(X, \bar{X})$, is the set of nodes in X that are adjacent to some node in $\bar{X}$, *i.e.*,

$$C(X, \bar{X}) = \{x : (x, y) \in E(N), x \in X \text{ and } y \in \bar{X}\}$$

The node cut-size of $(X, \bar{X})$, denoted $n(X, \bar{X})$, is the number of nodes in $C(X, \bar{X})$. A cut $(X, \bar{X})$ is K*-feasible* if its node cut-size is no more than K, *i.e.*, $n(X, \bar{X}) \leq K$. Assuming that each edge (u, v) has a nonnegative capacity $c(u, v)$. Then the *edge cut-size* of $(X, \bar{X})$, denoted $e(X, \bar{X})$, is the sum of the capacities of the edges that go from X to $\bar{X}$, *i.e.*,

$$e(X, \bar{X}) = \sum_{u \in X, v \in \bar{X}} c(u, v)$$

Moreover, assuming that there is a given label $l(v)$ associated with each node v ($l(v)$ may not be an interger), the *height* of a cut $(X, \bar{X})$, denoted $h(X, \bar{X})$, is defined to be the maximum label of the nodes in $C(X, \bar{X})$. The *volume* of a cut $(X, \bar{X})$, denoted $vol(X, \bar{X})$, is the number of nodes in $\bar{X}$, *i.e.*, $vol(X, \bar{X}) = |\bar{X}|$. Figure 2 shows a cut $(X, \bar{X})$ in a network with given node labels, where $n(X, \bar{X}) = 3$, $e(X, \bar{X}) = 9$, $h(X, \bar{X}) = 2.5$, and $vol(X, \bar{X}) = 9$. The darkened nodes are in the node cut set.

3 Delay Optimal Mapping for Heterogeneous FPGAs

In this section, we present a delay optimal technology mapping algorithm, named HeteroMap, for heterogeneous FPGAs under the heterogeneous LUT-delay model. It is applicable to any K*-bounded* ($K \leq K_1$) Boolean network. Given a general Boolean network as input, if it is not K*-bounded* ($K \leq K_1$), we first transform it into a 2-input simple gate network by using any of the decomposition algorithms, such as tech_decomp in SIS [SeSL92], DMIG [ChCD92], and DOGMA [CoHw96]. The optimality of our algorithm holds not only for 2-input simple gate networks, but for any K*-bounded* ($K \leq K_1$) general Boolean network as well.

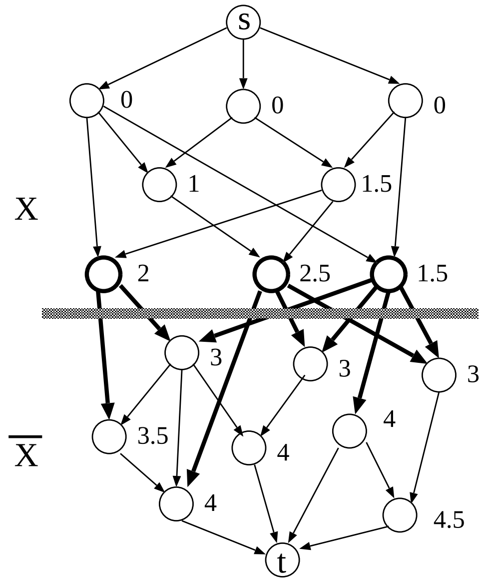


Figure 2: A 3-feasible cut of node cut-size 3, edge cut-size 9, height 2.5 and volume 9.

The HeteroMap algorithm has two phases. In the first phase, according to the topological order from PI to PO, HeteroMap uses the dynamic programing technique to compute the label for each node, which is the delay of the node if implemented by an LUT in a delay-optimal mapping solution. In the second phase, according to the reverse topological order starting from POs, HeteroMap generates the heterogeneous LUT mapping solution based on the node labels and cuts computed in the first phase. Our algorithm also minimizes the circuit area by maximizing the volume of each cut and by the post-mapping packing operations. The details are discussed in the following subsections.

3.1 Labeling Phase

Given a K-bounded Boolean network N, let N_t denote the subnetwork consisting of node t and all the predecessors of t. The *label* of t, denoted $l(t)$, is the delay of t in the *optimal* heteroheneous LUT mapping solution of N_t. The first phase of our algorithm computes the labels for all the nodes in N, according to the topological order starting from the PIs. The topological order guarantees that every node is processed after all of its predecessors have been processed. For each PI node u, we assign $l(u) = 0$. Suppose that t is the current node being processed. Then, for each node $v \neq t$ in N_t, the label $l(v)$ must have been computed. By including in N_t an auxiliary node s and connecting s to all the PI nodes in N_t, we obtain a network with s as the source and t as the sink. For simplicity, we still denote it as N_t. Figure 3(a) shows a Boolean network in which gate t is to be labeled. Figure 3(b) shows the construction of the network N_t. As there are c types of LUTs in a heterogeneous FPGA device, assuming that $LUT(t)$ is the K-LUT that implements node t in an optimal mapping solution of N_t under the heteroheneous LUT-delay model, K must be K_1, K_2, ..., or K_c. Therefore, we compute $l_i(t)$ for each i = 1, 2, ..., and c, which is the minimum delay of node t in the optimal mapping solution of N_t if t is implemented by a K_i-LUT. Label $l(t)$ is the minimum one among all $l_i(t)$ $(i = 1, 2, \ldots, c)$. That is,

$$l(t) = \min_{1 \leq i \leq c} l_i(t) \tag{1}$$

In order to compute $l_i(t)$ for each $i = 1$, 2, ..., and c, suppose $LUT_i(t)$ be the K_i-LUT that implements node t. Let $\bar{X}_i$ denote the set of nodes in $LUT_i(t)$ and X_i denote the remaining nodes in N_t. Then, $(X_i, \bar{X}_i)$ forms a K_i-feasible cut between s and t in N_t because the number of inputs of $LUT_i(t)$ is no more than K_i. Moreover, let u_i be the node with the maximum label in $C(X_i, \bar{X}_i)$, then, $l_i(t)$, the delay of $LUT_i(t)$, is $l(u_i) + d_i$. According to the definition of the height of a cut in Section 2, $l(u_i) = h(X_i, \bar{X}_i)$. Therefore, in order to minimize $l_i(t)$, we want to find a minimum height K_i-feasible cut $(X_i, \bar{X}_i)$ in N_t. Since for each i $(i = 1, 2, \ldots, c)$, we compute a minimum height K_i-feasible cut $(X_i, \bar{X}_i)$ in N_t and $l_i(t) = h(X_i, \bar{X}_i) + d_i$, Eq. 1 can be transformed to

$$l(t) = \min_{1 \leq i \leq c} \quad \min_{(X_i, \bar{X}_i) \text{ is } K_i\text{-feasible}} (h(X_i, \bar{X}_i) + d_i) \tag{2}$$

Based on the above discussion, we have

Lemma 1 *The label $l(t)$ computed by Eq. 1 gives the minimum delay of any mapping solution of N_t under the heterogeneous LUT-delay model.*

In Figure 3, we assume that there are two types of LUTs, 4-LUTs and 5-LUTs, with delays of 1 and 1.5, in a heterogeneous FPGA device. Through the minimum height 4-feasible cut computation, Figure 3(b) computes $l_1(t)$ to be 3, assuming that t is implemented by a 4-LUT. Similarly, Figure 3(c) computes $l_2(t)$ to be 2.5, assuming that t is implemented by a 5-LUT. In Figure 3(d), the label $l(t)$ is then set to be 2.5. The LUT that implements t in the optimal mapping solution is determined to be a 5-LUT which includes all the nodes in $\bar{X}$, where $(X, \bar{X})$ is the minimum height 5-feasible cut in N_t computed in Figure 3(c).

Lemma 2 (Monotone Property) *Let $l(t)$ be the label of node t and $l(t')$ be the label of a predecessor t' of t, then $l(t) \geq l(t')$.*

Proof: Suppose that the type of LUT that node t uses in the optimal mapping solution of N_t is a K_i-LUT. Then the cut $(X, \bar{X})$ formed by this K_i-LUT in N_t is a minimum height K_i-feasible cut in N_t, and $l(t) = h(X, \bar{X}) + d_i$. Since node t' is a predecessor of node t, $(X, \bar{X})$ also determines a K_i-feasible cut $(X', \bar{X}')$ in $N_{t'}$ with $h(X', \bar{X}') \leq h(X, \bar{X})$, where $X' = X \cap V(N_{t'})$ and $\bar{X}' = \bar{X} \cap V(N_{t'})$. Moreover, according to Eq. 2, $l(t') \leq h(X', \bar{X}') + d_i$. Therefore, $l(t) = h(X, \bar{X}) + d_i \geq h(X', \bar{X}') + d_i \geq l(t')$. ■

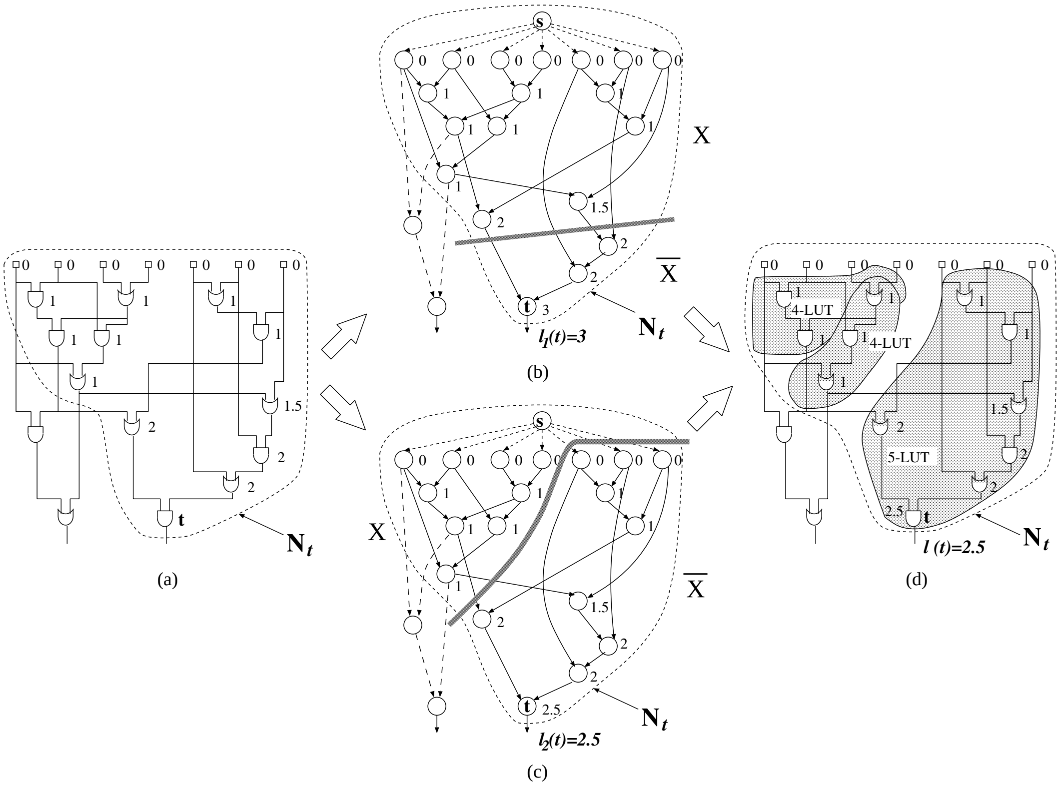


Figure 3: (a) The Boolean network in which gate t is to be labeled; (b) Construction of N_t and the computation of the minimum height 4-feasible cut in N_t such that $l_1(t) = 3$; (c) The computation of the minimum height 5-feasible cut in N_t such that $l_2(t) = 2.5$; (d) Determining $l(t) = min\{l_1(t), l_2(t)\} = 2.5$.

According to Eq. 1 and Eq. 2, the key problem in computing $l(t)$ for node t is to compute the minimum height K_i-feasible cut $(X_i, \bar{X}_i)$ in N_t to get $l_i(t) = h(X_i, \bar{X}_i) + d_i$, where $i = 1, 2, \ldots,$ and c. The computation of a minimum height K-feasible cut in the case of heterogeneous LUT-delay model is more complicated than that under the unit-delay model, although $l(t)$ is monotone. First of all, we need to determine the range of the minimum height of a K_i-feasible cut in N_t, denoted $H_t(i)$.

Lemma 3 *Let t be a non-PI node in network N, then,*

$$\max\{l(u) : u \in input(t)\} - d_i \leq H_t(i) \leq \max\{l(u) : u \in input(t)\}. \tag{3}$$

Proof: Since $(N_t - \{t\}, \{t\})$ is a K-feasible cut in N_t, $l_i(t) \leq \max\{l(u) : u \in input(t)\} + d_i$. According to Lemma 2, $l_i(t) \geq \max\{l(u) : u \in input(t)\}$. Therefore, we can derive the range of $H_t(i)$ to be $\max\{l(u) : u \in input(t)\} - d_i \leq H_t(i) \leq \max\{l(u) : u \in input(t)\}$. ■

Based on Eq. 3, we get a height array, denoted $H_{t_{list}}(i)$[6], which includes all the distinct labels $l(u)$ ($u \in N_t$) that are in the range of Eq. 3. The maximum size of the height array is $|N_t|$. Then, we can perform a binary search over this height array to determin the minimum height $H_t(i)$ and hence $l_i(t)$.

To compute a K-feasible cut of height H in N_t, we apply the approach proposed in [CoDi94a], which converts the node cut problem on N_t into a standard edge cut computation problem on a flow network N_t' transformed from N_t. The edge cut problem can be solved by the augmenting path algorithm for max-flow computation. Assuming that the number of edges in N_t is m, we can determin in $O(K_i \cdot m)$ time whether N_t has a K_i-feasible cut of height H, and find one if such a cut exists. Therefore, we can find the K_i-feasible cut of *minimum height* $H_t(i)$ in N_t by binary search on the height array $H_{t_{list}}(i)$. Then, $l_i(t) = H_t(i) + d_i$. Based on the above discussion, we have

Lemma 4 *A minimum height K_i-feasible cut in N_t under heterogeneous LUT-delay model can be found in $O(K_i \cdot m \cdot \log |N_t|)$ time where m is the number of edges in N_t.*

For each $i = 1, 2, \ldots, c$, we compute $l_i(t)$, and the label of node t can be determined by Eq. 1 in $O(\sum_{i=1}^{c} K_i \cdot m \cdot \log |N_t|)$ time, where m is the number of edges in N_t. Applying the label computation for each node in N according to the topological order in the labeling phase, we have

Theorem 1 *Given a network N with n nodes and m edges, the labeling process can be done in $O(\sum_{i=1}^{c} K_i \cdot n \cdot m \cdot \log n)$ time.*

3.2 Mapping Phase

In the labeling phase, for each node u in N we have computed $l(u)$ and $LUT(u)$, the LUT implementing u in an optimal mapping solution of N_u. The second phase of the HeteroMap algorithm generates the K-LUTs in the optimal solution. This phase is similar to that in [CoDi94a], except that the LUTs generated in our mapping solution can be heterogeneous. We maintain a list L of nodes that have to be implemented by LUTs. Initially, L contains those nodes that have fanouts to the PO nodes. Then, we repeatedly remove a node u from L, create the LUT based on $LUT(u)$ computed in the labeling phase, and add into L all the nodes in $input(LUT(u))$ whose LUTs have not yet been created. The mapping phase ends when L only contains PI nodes. The entire second phase takes linear time. Moreover, it is not hard to see that the delay from any PI to the output of $LUT(u)$ is no more than $l(u)$. Therefore, the mapping solution is optimal. The nodes that are never added into L do not need to be implemented since they are completely covered by the LUTs implementing other nodes.

The HeteroMap algorithm is summaried in Figure 4.

[6]Our implementation can extract the sorted height array $H_{t_{list}}(i)$ in $\min\{O(n), O(H_{t_{list}}(i) \cdot \log H_{t_{list}}(i))\}$ time.

Algorithm HeteroMap
/* phase 1: labeling network */
for each PI node v **do**
$l(v) = 0$;
for each non-PI node t in N in topological order **do**
$l(t) = \infty$;
construct the network N_t;
$p = \max\{l(u) : u \in input(t)\}$;
for $i = 1$ to c **do**
get the height array $H_{t_{list}}(i)$ which includes all the distinct labels $l(u)$
($u \in N_t$ and $p - d_i \leq l(u) \leq p\}$);
compute the minimum height $(H_t(i))$ K_i-feasible cut $(X_i(t), \bar{X_i}(t))$ in N_t
by binary search on the height array $H_{t_{list}}(i)$;
$l_i(t) = H_t(i) + d_i$;
if $l(t) > l_i(t)$ **then**
$l(t) = l_i(t)$; $\bar{X}(t) = \bar{X_i}(t)$;
endif
endfor
/* phase 2: generating heterogeneous mapping solution */
L = list of all the PO nodes;
while L contains non-PI nodes **do**
take a non-PI node v from L;
generate an LUT v' to implement the function of v such that $input(v') = input(\bar{X}(v))$;
$L = (L - v) \cup input(v')$;
endwhile
end-algorithm;

Figure 4: Pseudocode of the HeteroMap algorithm.

3.3 Area Minimization in the HeteroMap Algorithm

The secondary objective of our technology mapping algorithm is area optimization, which is considered by maxmizing the volume of each cut during the mapping process and by the post-mapping heterogeneous predecessor packing operations. In the labeling phase, for each node t in the input network N, the HeteroMap algorithm computes a minimum-height K_i-feasible cut $(X_i(t), \bar{X_i}(t))$ for each $i = 1,\ 2,\ \ldots,\ c$ in N_t, and the final $(X(t), \bar{X}(t))$ is one of these $(X_i(t), \bar{X_i}(t))$. The nodes in $\bar{X}(t)$ will be packed into one LUT t' if an LUT is generated to implement t in the mapping phase. In general, the minimum-height K_i-feasible cut is not unique. Intuitively, the larger $vol(X_i(t), \bar{X_i}(t)) = |\bar{X_i}(t)|$, the more nodes we can pack into the K_i-LUT t', which leads to fewer LUTs in total. Therefore, our algorithm maximizes the volume of the cut during the minimum height K_i-fesible cut computation. HeteroMap uses the approach proposed in [CoDi94a], which finds the maximum volume minimum height K_i-feasible height in N_t with the same complexity as of computing any minimum height K_i-

feasible cut.

In [ChCD92], an operation called *predecessor packing* is proposed to minimize the number of K-LUTs in the homogeneous mapping solution. We extend this approach on the heterogeneous mapping solution, where one LUT capacitity can be different from another. If a K_a-LUT u ($a = 1,\ 2,\ \ldots$, or c) has a fanin K_b-LUT v ($b = 1,\ 2,\ \ldots$, or c), v has only one fanout and $|input(u) \cup input(v)| \leq K_a$, then v can be merged into u such that the LUT implementing v can be saved. This operation is carried out in the mapping solution of HeteroMap in the reverse topological oder from PO to PI, and ends when no such packing operations can be applied.

4 Experimental Results and Comparative Study

4.1 Technology Mapping Comparison on Heterogeneous FPGAs

We have implemented the HeteroMap algorithm in C language on SUN SPARC workstation. We first compare FlowMap [CoDi94a] and HeteroMap on XC4000 series FPGAs, which can implement circuits with 4-LUTs and 5-LUTs of delay 1 and 1.5[7]. For FlowMap, we set $K = 5$ such that after mapping, the 5-input nodes will be implemented in 5-LUTs, and the remaining nodes will be implemented in 4-LUTs. The mapping comparison results between FlowMap and HeteroMap are summarized in Table 1. The "mapping delay" in columns 2 and 7 is the critical path delay in the mapped network under the heterogeneous LUT-delay model. The PLB number in columns 3 and 8 comes from packing every two 4-LUTs into one PLB and leaving all 5-LUTs as independent PLBs. Table 2 shows that HeteroMap reduces 19% of the mapping delays and 9% of the PLB numbers over FlowMap.

Match4K [CoPD96] is an intelligent post-mapping multi-step matching heuristic for XC4000 device family. One of its features is to implement a 5-LUT using a partial XC4000 PLB, which can significantly reduce the PLB number in the mapping solution. To take advantage of the Match4K algorithm in XC4000 mapping, we also compare FlowMap followed by Match4k with HeteroMap followed by Match4K, and perform layout using XACT5.2 FPGA development system on the mapping solutions. The comparison results are summarized in Table 2. All the benchmarks in Table 2 use the same XC4000 part (4013MQ208-5) during the physical design. The "mapping delay"s in columns 3 and 6 are the same as those in Table 1, which are the critical path delays in the mapped networks under the heterogeneous LUT-delay model. The post-layout delays are reported by XACT5.2 FPGA development system. After Match4K, HeteroMap can still reduce 7% of the post-layout delays with only 2% increase on the PLB numbers over FlowMap.

[7]The delay ratio of 4-LUT *v.s.* 5-LUT implementation in XC4000 PLB structure is derived from the timing characteristics specified in [Xi97]. To make it more realistic, this ratio also takes the interconnection delay into consideration.

XC4000 series FPGAs

Circuits	FlowMap mapping delay	FlowMap PLB	FlowMap 4-LUT	FlowMap 5-LUT	FlowMap CPU (s)	HeteroMap mapping delay	HeteroMap PLB	HeteroMap 4-LUT	HeteroMap 5-LUT	HeteroMap CPU (s)
5xp1	4.50	20	11	14	0.20	3.50	16	22	5	0.20
9sym	7.00	45	32	29	0.30	5.50	48	73	11	0.60
9symml	7.00	47	22	36	0.30	5.50	44	74	7	0.60
C499	6.50	118	72	82	3.50	5.50	107	147	33	67.30
C880	12.00	183	101	132	2.30	10.00	151	236	33	13.30
alu2	12.00	128	69	93	1.50	10.50	114	154	37	13.80
alu4	15.00	209	117	150	3.70	11.50	205	277	66	99.40
apex2	7.00	117	91	71	0.80	6.50	108	176	20	2.10
apex7	6.00	63	54	36	0.50	4.00	56	84	14	0.80
count	4.50	59	32	43	0.40	4.00	51	66	18	0.60
duke2	6.00	140	94	93	1.20	4.50	127	191	31	2.30
e64	4.50	124	85	81	1.10	3.50	107	165	24	1.40
misex1	3.00	13	5	10	0.10	2.50	12	18	3	0.10
rd84	6.00	34	18	25	0.40	4.50	31	29	16	0.90
vg2	5.00	31	28	17	0.20	4.00	27	45	4	0.30
z4ml	3.50	9	9	4	0.10	3.50	10	11	4	0.10
TOTAL	109.50	1340	840	916	16.60	89.00	1214	1768	326	203.80
Ari_Mean	6.84	83.75	52.50	57.25	1.04	5.56	75.87	110.50	20.38	12.74
Ari_Ratio	1	1	1	1	1	-18.72%	-9.40%	+110%	-64%	+1127%
Geo_Mean	6.22	59.20	37.11	39.20	0.57	5.05	54.09	76.57	14.27	1.52
Geo_Ratio	1	1	1	1	1	-18.70%	-8.63%	+106%	-63%	+164%

Table 1: The mapping comparison between FlowMap and HeteroMap on XC4000 series FPGAs

In order to compare HeteroMap with FlowMap on ORCA2C series FPGAs, HeteroMap takes the ORCA2C device as a heterogeneous FPGA which consists of 5-LUTs and 6-LUTs with the delays of 1 and 1.2[8]. For FlowMap, we set $K = 6$ such that after technology mapping each 6-input node is implemented by one 6-LUT which occupies one PLB, and all other nodes can be implemented in 5-LUTs every two of which are packed into one PLB. The comparison results are summarized in Table 3. Each "mapping delay" in columns 2 and 7 is the critical path delay in the mapped network under the heterogeneous LUT-delay model. Compared with FlowMap, HeteroMap reduces 9% of the mapping delays and 8% of the PLB number.

A heterogeneous FPGA could consist of three or more types of logic blocks. In order to show more clearly the benefits of exploring heterogeneousness in technology mapping, we test FlowMap and HeteroMap on a hypothetical heterogeneous FPGA architecture which consists of three types of LUTs with input sizes 4, 5, and 6, and the delays 1, 1.25, and 1.5, respectively. For FlowMap, we set

[8]The delay ratio of 5-LUT *v.s.* 6-LUT implementation in ORCA2C PLB structure is derived from the timing and switching characteristics specified in [Or96]. To make it more realistic, this ratio also takes the interconnection delay into consideration.

XC4000 series FPGAs

Circuits	FlowMap + Match4K			HeteroMap + Match4K		
	PLB	mapping delay	post-layout delay (ns)	PLB	mapping delay	post-layout delay (ns)
5xp1	16	4.50	41.00	14	3.50	36.70
9sym	35	7.00	54.10	39	5.50	53.10
9symml	37	7.00	60.60	38	5.50	53.00
C499	83	6.50	80.70	88	5.50	76.80
C880	132	12.00	140.20	132	10.00	106.80
alu2	94	12.00	110.10	94	10.50	107.50
alu4	149	15.00	128.40	163	11.50	160.20
apex2	87	7.00	84.00	91	6.50	70.70
apex7	46	6.00	55.00	49	4.00	52.70
count	44	4.50	52.90	39	4.00	46.90
duke2	99	6.00	75.90	103	4.50	69.90
e64	83	4.50	68.00	91	3.50	63.10
misex1	10	3.00	30.30	11	2.50	31.00
rd84	29	6.00	53.70	26	4.50	49.80
vg2	23	5.00	55.00	23	4.00	44.30
z4ml	7	3.50	34.70	8	3.50	34.50
TOTAL	974	109.50	1124.60	1009	89.00	1057.00
Ari_Mean	60.88	6.84	70.29	63.06	5.56	66.06
Ari_Ratio	1	1	1	+3.59%	-18.72%	-6.01%
Geo_Mean	44.23	6.22	64.24	45.32	5.05	59.77
Geo_ratio	1	1	1	+2.46%	-18.70%	-6.96%

Table 2: Post-layout comparison between FlowMap and HeteroMap on XC4000 series FPGAs

$K = 6$ such that after mapping each 6-input node is implemented by one 6-LUT, each 5-input node by one 5-LUT, and all remaining nodes by 4-LUTs. The results are shown in Table 4. Each "mapping delay" in columns 2 and 7 is the critical path delay under the heterogeneous LUT-delay model. In order to compare the area in the mapping solution, let $r = \frac{\text{area of } (K+1)-\text{LUT}}{\text{area of } K-\text{LUT}}$. In Table 4 we set $r = 2$ based on the SRAM cell count, which assumes that the area of each 4-LUT, 5-LUT and 6-LUT is 1, 2 and 4, respectively. Under this assumption, HeteroMap reduces 15% of the mapping delays and 20% of the area, compared with FlowMap. In general, the area ratio of r may not be 2 due to the additional logic and interconnects in a PLB. To make a more accurate area analysis, we perform a sequence of tests with $1 \leq r \leq 3$ and plot the results in Figure 5 to show the area comparison between FlowMap and HeteroMap. HeteroMap outperforms FlowMap on area when $r \geq 1.5$, and is always better than FlowMap in terms of the delay.

ORCA2C series FPGAs

Circuits	FlowMap					HeteroMap				
	mapping delay	PLB	5-LUT	6-LUT	CPU (s)	mapping delay	PLB	5-LUT	6-LUT	CPU (s)
5xp1	2.20	12	10	7	0.10	2.20	12	12	6	0.20
9sym	4.60	30	12	24	0.20	4.20	30	35	13	0.60
9symml	4.60	28	15	21	0.20	4.20	30	39	11	0.60
C499	5.40	104	105	52	3.60	5.00	85	157	7	68.70
C880	8.20	141	91	96	2.00	8.00	138	220	28	18.00
alu2	8.40	95	77	57	1.30	8.00	97	153	21	18.10
alu4	9.40	181	134	114	3.20	8.60	168	237	50	128.50
apex2	5.60	98	83	57	0.60	5.00	97	137	29	2.30
apex4	6.00	506	316	348	6.10	5.20	406	765	24	30.10
apex6	4.80	143	170	58	1.20	4.00	132	227	19	1.90
apex7	3.40	53	56	25	0.40	3.20	49	82	8	0.80
count	3.60	42	37	24	0.30	3.00	36	61	6	0.60
des	3.60	457	199	358	91.10	3.40	458	368	274	139.80
duke2	4.40	110	87	67	0.80	4.00	106	164	24	1.90
e64	3.60	111	87	68	0.70	3.00	94	156	16	1.80
misex1	2.40	8	2	7	0.10	2.00	7	9	3	0.10
rd84	4.40	24	26	11	0.30	4.00	23	47	0	0.50
rot	6.00	165	139	96	1.90	5.60	153	222	42	6.20
vg2	3.40	24	35	7	0.20	3.20	25	46	2	0.30
z4ml	2.20	4	5	2	0.10	2.20	4	7	1	0.10
TOTAL	96.20	2336	1686	1499	114.40	88.00	2150	3144	584	421.10
Ari_Mean	4.81	116.80	84.30	74.95	5.72	4.40	107.50	157.20	29.20	21.05
Ari_Ratio	1	1	1	1	1	-8.52%	-7.96%	+86%	-61%	+268%
Geo_Mean	4.44	61.89	47.29	36	0.76	4.04	58.40	87.52	0	2.44
Geo_Ratio	1	1	1	1	1	-8.81%	-5.65%	+85%	N/A	+223%

Table 3: Comparison between FlowMap and HeteroMap on ORCA2C series FPGAs

4.2 Heterogeneous *v.s.* Homogeneous Architectures

The optimality of the HeteroMap algorithm enables us to quantitatively evaluate various heterogeneous architectures without the bias of mapping heuristics. A set of architecture study results of heterogeneous FPGAs are presented in this subsection. As an example, we use HeteroMap to map the circuits into a heterogeneous FPGA with four types of logic blocks, 3-LUTs, 4-LUTs, 5-LUTs, and 6-LUTs with delays of 1, 1.33, 1.67, and 2. FlowMap maps circuits into the homogeneous K-LUT FPGAs with $K = 3, 4, 5,$ and, 6 respectively. The comparision results on both area and mapping delay are summaried in Table 5. Each "mapping delay" in columns 2, 4, 6, 8 and 10 is the critical path delay in the mapped network under the heterogeneous LUT-delay model. Let $r = \frac{\text{area of } (K+1)-\text{LUT}}{\text{area of } K-\text{LUT}}$. In order to compare the area in the mapping solution, we set $r = 2$ in Table 5, which assumes that the area of one 3-LUT, 4-LUT, 5-LUT and 6-LUT is 1, 2, 4 and 8, respectively. With HeteroMap,

Circuits	FlowMap					HeteroMap				
	mapping delay	Area	4-LUT	5-LUT	6-LUT	mapping delay	Area	4-LUT	5-LUT	6-LUT
5xp1	2.75	40	8	2	7	2.50	37	9	2	6
9sym	5.50	117	3	9	24	5.25	103	29	9	14
9symml	5.75	110	4	11	21	5.00	108	18	11	17
C499	6.50	384	34	71	52	5.25	221	119	31	10
C880	10.25	514	52	39	96	9.00	397	197	42	29
alu2	10.50	340	42	35	57	9.25	271	119	38	19
alu4	11.75	654	70	64	114	10.50	541	173	74	55
apex2	7.00	347	47	36	57	5.75	269	143	29	17
apex4	7.50	1874	150	166	348	6.25	1279	853	113	50
apex6	6.00	532	40	130	58	4.50	445	147	103	23
apex7	4.25	174	38	18	25	3.75	150	74	18	10
count	4.50	144	26	11	24	3.50	108	60	14	5
des	4.50	1693	137	62	358	4.00	1583	373	57	274
duke2	5.25	396	46	41	67	4.25	307	177	37	14
e64	4.50	390	56	31	68	3.25	247	155	24	11
misex1	3.00	30	2	0	7	2.25	26	8	3	3
rd84	5.50	88	8	18	11	4.25	61	35	13	0
rot	7.50	574	88	51	96	6.50	450	182	74	30
vg2	4.00	75	23	12	7	3.50	68	38	7	4
z4ml	2.50	15	3	2	2	2.50	13	5	2	1
TOTAL	119.00	8491	877	809	1499	101.00	6684	2914	701	592
Ari_Mean	5.95	424.55	43.85	40.45	74.95	5.05	334.20	145.70	35.05	29.60
Ari_Ratio	1	1	1	1	1	-15.13%	-21.28%	+232%	-13%	-60%
Geo_Mean	5.47	222.36	24.21	0	36	4.62	177.78	72.88	20.19	0
Geo_Ratio	1	1	1	1	1	-15.57%	-20.05%	+201%	N/A	N/A

Table 4: Comparison between FlowMap and HeteroMap on a heterogeneous FPGA with three types of logic blocks

heterogeneous FPGAs always achieve better mapping delay than that of homogeneous FPGAs. On average, the mapping delays of the four homogeneous FPGAs are almost the same. As for the area, the heterogeneous FPGA outperforms the homogeneous FPGAs with $K = 4$, 5, and 6. When $K = 3$, the homogeneous FPGA gets 34% better area than the heterogeneous FPGA but has 25% longer delays on average. To better understand the impact of area ratio r and the mapping results on both area and delay, Figure 6(a) is plotted with $1 \leq r \leq 3$ as the X-axis and $area \times delay^2$ as the Y-axis [Ul83], and Figure 6(b) is plotted with $1 \leq r \leq 3$ as the X-axis and $area \times delay$ as the Y-axis [Ul83]. From Figure 6 we can see that the FPGA with heterogeneous LUTs, 3-LUTs or 4-LUTs is more efficient than the one with 5-LUTs or 6-LUTs. However, this architecture study conclusion relies on the delay ratio of heterogeneous LUTs. Table 6 shows how the heterogeneous LUTs with different capacities and delays are used in the delay optimal mapping solution of HeteroMap with different delay ratios, which could be very useful for the architecture design of future heterogeneous FPGAs. When the delays between different LUTs become closer, HeteroMap will make use of more larger LUTs in the

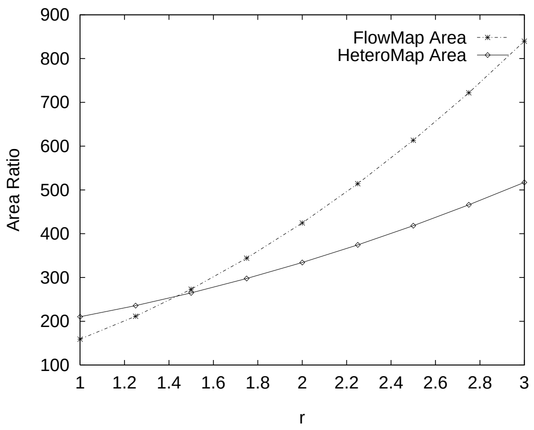


Figure 5: Area comparison between FlowMap and HeteroMap with respect to the area ratio of 4-LUT and 5-LUT.

delay optimal mapping solution.

5 Conclusions and Future Work

In this paper, we presented an optimal technology mapping algorithm, named HeteroMap, for delay minimization in heterogeneous FPGA designs. Taking different delays of heterogeneous LUTs into consideration, the HeteroMap algorithm computes the minimum mapping delay of a circuit in polynomial time based on a series of minimum height K-feasible cut computations on each node in the circuit. HeteroMap also effectively minimizes the area by maximizing the volume of each cut and by the post-mapping heterogeneous predecessor packing operation.

We are extending the HeteroMap algorithm to handle the general delay model in heterogeneous FPGA designs. The new delay model will take both edge delay and the heterogeneous LUT delay into consideration, which could be very benificial for both heterogeneous architecure study and technology mapping for the current FPGAs with configurable heterogeneousness.

Some heterogeneous FPGAs may have resource limitations on the LUTs of specific sizes, which we call *heterogeneous FPGAs with bounded resources*. For example, Altera FLEX10K device family [Al96] provides in a single device both a logic array of 4-LUTs and an embedded memory array with a series of embedded memory blocks (EMBs) whose number varies from 3 to 12 according to the device size. Each EMB in FLEX10K devices has much larger capacity than that of a 4-LUT, and can implement any single 11-input 1-output, 10-input 2-output, 9-input 4-output, or 8-input 8-output logic function. The delay-optimal technology mapping for heterogeneous FPGAs with bounded resources is still an open problem, and we are in the process of developing optimal or near-optimal algorithms for this

Circuits	FlowMap on homogeneous FPGAs of K-LUTs								HeteroMap on Hetero-FPGAs	
	$K=3$		$K=4$		$K=5$		$K=6$			
	mapping delay	Area	mapping delay	Area	mapping delay	Area	mapping delay	Area	mapping delay	Area
5xp1	6	59	5.33	74	5.00	100	4	136	3.33	77
9sym	7	128	8.00	182	8.34	244	8	288	6.67	161
9symml	8	126	8.00	182	8.34	232	8	288	6.33	171
C499	9	368	8.00	446	8.34	616	10	1256	6.67	432
C880	13	339	13.33	540	13.34	932	14	1496	11.33	530
alu2	15	265	14.66	412	13.34	648	14	1072	12.00	475
alu4	16	476	17.33	720	16.67	1068	16	1984	13.33	890
apex2	9	252	9.33	388	8.34	648	10	1120	7.33	393
apex4	9	1102	9.33	1814	10.00	3052	10	5304	7.67	1585
apex6	7	456	6.66	858	6.67	1028	8	1824	5.67	751
apex7	6	150	5.33	208	6.67	360	6	648	5.00	244
count	6	127	5.33	172	5.00	300	6	488	4.67	192
des	9	2091	8.00	3016	8.34	5232	6	4456	5.00	3048
duke2	6	266	6.66	416	6.67	748	8	1232	5.33	420
e64	5	248	5.33	388	5.00	664	6	1240	4.33	361
misex1	4	32	4.00	46	3.33	60	4	72	2.67	49
rd84	6	81	6.66	120	6.67	172	8	296	5.33	101
rot	11	384	9.33	616	10.00	1072	10	1880	8.67	718
vg2	5	76	5.33	102	6.67	180	6	336	4.67	105
z4ml	5	28	5.33	44	5.00	52	4	56	3.00	20
TOTAL	162	7054	161.27	10744	161.73	17408	166	25472	129.00	10723
Ari_Mean	8.10	352.70	8.06	537.20	8.08	870.40	8.30	1273.60	6.45	536.15
Ari_Ratio	+25%	-34%	+25%	+0.2%	+25%	+62%	+28%	+137%	1	1
Geo_Mean	7.53	200.06	7.49	296.78	7.51	453.08	7.68	699.77	5.89	288.09
Geo_Ratio	+27%	-30%	+27%	+3%	+27%	+57%	+30%	+142%	1	1

Table 5: Performance Comparison between homogeneous FPGAs and heterogeneous FPGAs

problem.

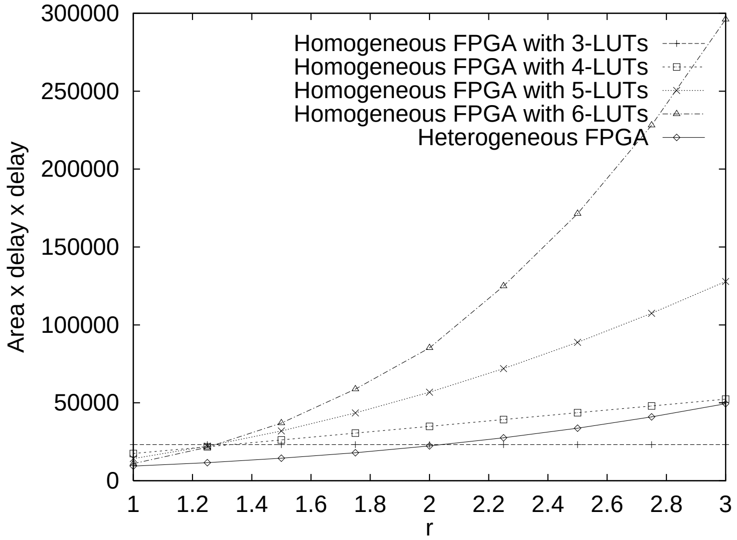
6 Acknowledgments

This work is partially supported by Lucent Technologies Inc. and Xilinx Inc. under the California MICRO Program and National Science Foundation Young Investigator Award MIP9357582.

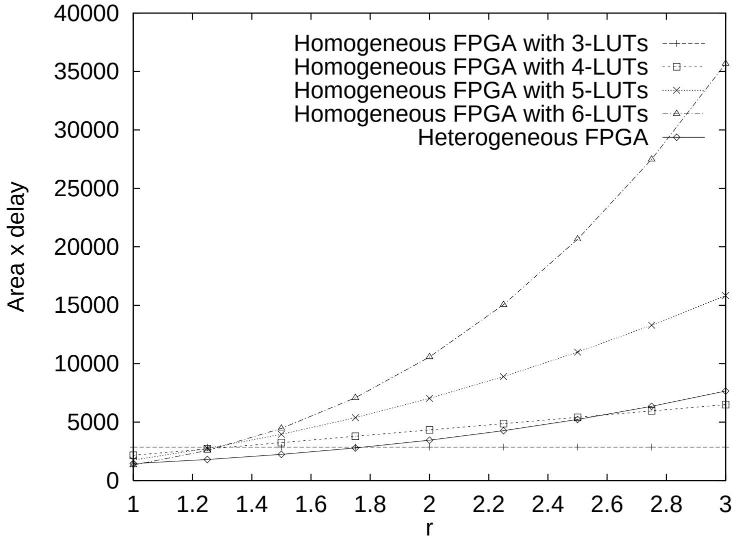
References

[Al96] Altera, *"Programmable Logic Devices Data Book"*, Altera Corp., San Jose, CA, 1996.

[BhHi92] N. Bhat, D. Hill, *"Routable Technology Mapping for FPGAs"*, First International ACM/SIGDA Workshop on Field Programmable Gate Arrays, Feb. 1992, pp. 143-148.



(a) $Area \times delay^2$ Comparison between homogeneous and heterogeneous FPGAs with respect to r



(b) $Area \times delay$ Comparison between homogeneous and heterogeneous FPGAs with respect to r

Figure 6: Comparison between homogeneous and heterogeneous FPGAs with respect to the area ratio of r.

[ChCD92] K. C. Chen, Jason Cong, Yuzheng Ding, Andrew B. Kahng, Peter Trajmar, *"DAG-Map: Graph-based FPGA Technology Mapping for Delay Optimization"*, IEEE Design and Test of Computers, Sep. 1992, pp. 7-20.

[CoDi93a] J. Cong, Y. Ding, *"On Area/Depth Trade-off in LUT-Based FPGA Technology Mapping"*, Proc. 30th ACM/IEEE Design Automation Conference, pp. 213-218, June 1993, pp. 213-218.

[CoDG93] J. Cong, Y. Ding, T. Gao, K. Chen, *"An Optimal Performance-Driven Technology Mapping Algorithm for LUT based FPGAs under Arbitrary Net-Delay Models"*, Proc. 1993 International

Circuits	Delay ratio = 1 : 1.33 : 1.67 : 2					Delay Ratio = 1 : 1.2 : 1.4 : 1.6				
	mapping delay	3-LUT	4-LUT	5-LUT	6-LUT	mapping delay	3-LUT	4-LUT	5-LUT	6-LUT
5xp1	3.33	11	3	1	7	2.80	6	4	2	6
9sym	6.67	35	19	8	7	5.80	9	7	12	18
9symml	6.33	35	10	7	11	5.40	11	6	18	14
C499	6.67	70	67	27	15	5.80	59	57	31	17
C880	11.33	158	86	20	15	10.20	109	46	60	37
alu2	12.00	81	47	31	22	10.60	55	23	63	28
alu4	13.33	114	100	58	43	11.60	81	60	90	60
apex2	7.33	127	59	23	7	6.40	76	38	38	34
apex4	7.67	663	317	62	5	7.20	329	325	219	58
apex6	5.67	119	62	97	15	5.00	93	42	119	22
apex7	5.00	48	32	13	10	4.20	46	18	23	13
count	4.67	52	24	13	5	4.00	38	12	19	9
des	5.00	172	232	57	273	4.40	166	119	114	274
duke2	5.33	162	47	23	9	4.80	90	34	51	32
e64	4.33	145	68	16	2	3.80	99	51	36	21
misex1	2.67	3	5	3	3	2.40	4	2	4	3
rd84	5.33	27	7	13	1	4.80	23	9	14	1
rot	8.67	148	85	50	25	7.40	97	45	85	61
vg2	4.67	27	21	5	2	4.00	24	18	9	4
z4ml	3.00	4	0	2	1	2.60	4	0	2	1
TOTAL	129.00	2201	1291	529	478	113.20	1419	916	1009	713
Ari_Mean	6.45	110.05	64.55	26.45	23.90	5.66	70.95	45.80	50.45	35.65
Ari_Ratio	1	1	1	1	1	-12.24%	-35.53%	-29.05%	+90.74%	+49.16%
Geo_Mean	5.89	57.91	0	15.34	8.27	5.16	39.55	0	26.76	15.92
Geo_Ratio	1	1	1	1	1	-12.47%	-31.71%	N/A	+74.43%	+92.58%

Table 6: HeteroMap results with different delay ratios

Conference on CAD and Computer Graphics, Aug. 1993, pp. 599-603.

[CoDi94a] J. Cong, Y. Ding, *"FlowMap: An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs",* IEEE Transactions on Computer-Aided Design, Feb. 1994, Vol. 13, No. 1, pp. 1-12.

[CoHw95a] J. Cong, Y. Hwang, *"Simultaneous Depth and Area Minimization in LUT-Based FPGA Mapping",* UCLA Computer Science Dept. Tech. Report CSD-950001, January 1995.

[CoHw96] Jason Cong, Yean-Yow Hwang, *"Structural Gate Decomposition for Depth-Optimal Technology Mapping in LUT-based FPGA Design"* 33rd ACM/IEEE Design Automation Conference, June 1996.

[CoPD96] Jason Cong, John Peck, Yuzheng Ding, *"RASP: A General Logic Synthesis System for SRAM-based FPGAs",* Proc. ACM 4th International Symposium on FPGA, Feb. 1996, pp. 137-143.

[CoDi96] J. Cong, Y. Ding, *"Tutorial and Survey Paper — Combinational Logic Synthesis for LUT Based Field Programmable Gate Arrays",* ACM Transactions on Design Automation of Electronic Systems, Vol. 1, No. 2, April 1996, pp. 145-204.

[CoXu98] J. Cong, S. Xu, *"Technology Mapping for FPGAs with Embedded Memory Blocks",* Proc. ACM International Symposium on FPGA, Monterey, CA., Feb. 1998, pp. 179-188.

[FrRC90] R. J. Francis, J. Rose, K. Chung, *"Chortle: A Technology Mapping Program for Lookup Table-Based Field Programmable Gate Arrays"*, Proc. 27th ACM/IEEE Design Automation Conference, June 1990, pp. 613-619.

[FrRV91a] R. J. Francis, J. Rose, Z. Vranesic, *"Chortle-crf: Fast Technology Mapping for Lookup Table-Based FPGAs"*, Proc. 28th ACM/IEEE Design Automation Conference, June 1991, pp. 613-619.

[FrRV91b] R. J. Francis, J. Rose, Z. Vranesic, *"Technology Mapping for Delay Optimization of Lookup Table-Based FPGAs"*, MCNC Logic Synthesis Workshop, 1991.

[HeRo94] J. He, J. Rose, *"Technology Mapping for Heterogeneous FPGAs"*, Proc. ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Feb. 1994.

[Ka91a] K. Karplus, *"Xmap: A Technology Mapper for Table-lookup Field-Programmable Gate Arrays"*, Proc. 28th ACM/IEEE Design Automation Conference, June 1991, pp. 240-243.

[MaLi94] A. Mathur, C. L. Liu, *"Performance Driven Technology Mapping for Lookup-Table Based FPGAs Using the General Delay Model"*, Proc. ACM/SIGDA International Symposium on Field Programmable Gate Arrays, Feb. 1994.

[MuNS90] R. Murgai, Y. Nishizaki, N. Shenoy, R. Brayton, A. Sangiovanni-Vincentelli, *"Logic Synthesis Algorithms for Programmable Gate Arrays"*, Proc. 27th ACM/IEEE Design Automation Conference, 1990, pp. 620-625.

[MuSB91a] R. Murgai, N. Shenoy, R. K. Brayton, A. Sangiovanni-Vincentelli, *"Performance Directed Synthesis for Table Look Up Programmable Gate Arrays"*, Proc. IEEE International Conference on Computer-Aided Design, Nov. 1991, pp. 572-575.

[MuSB91b] R. Murgai, N. Shenoy, R. K. Brayton, A. Sangiovanni-Vincentelli, *"Improved Logic Synthesis Algorithms for Table Look Up Architectures"*, Proc. IEEE International Conference on Computer-Aided Design, Nov. 1991, pp. 564-567.

[Or96] Lucent Technologies, *"ORCA OR2C-A/OR2T-A Series FPGAs Data Sheet"*, Lucent Technologies, Inc., Allentown, PA, 1996.

[ScKC92] M. Schlag, J. Kong, P. K. Chan, *"Routability-Driven Technology Mapping for Lookup Table-Based FPGAs"*, Proc. 1992 IEEE International Conference on Computer Design, Oct. 1992, pp. 86-90.

[SeSL92] E. Sentovich, K. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. Stephen, R. Brayton, A. Sangiovanni-Vincentelli, *"SIS: A System for Sequential Circuit Synthesis"* U.C.Berkeley Technical Report UCB/ERL M92/41, May. 1992.

[Wo91] Nam-Sung Woo, *"A Heuristic Method for FPGA Technology Mapping Based on the Edge Visibility"*, Proc. 28th ACM/IEEE Design Automation Conference, June 1991, pp. 248-251.

[Ul83] J. D. Ullman, *"Computational Aspects of VLSI"*, Computer Science Press, Inc., Rockville, MD, 1983.

[Xi97] Xilinx, *"The Programmable Logic Data Book"*, Xilinx Inc., San Jose, CA, 1997.

[YaWo94] H. Yang, D. Wong, *"Edge-Map: Optimal Performance Driven Technology Mapping for Iterative LUT Based FPGA Designs"*, Proc. 1994 IEEE/ACM International Conference on Computer-Aided Design, Nov. 1994, pp. 150-155.