

# 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

## Abstract

Recent generation of FPGAs take advantage of speed and density benefits resulted from *heterogeneous* FPGAs, which provide either an array of homogeneous programmable logic blocks (PLBs), each configured to implement circuits with lookup tables (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. For a heterogeneous FPGA consisting of  $K_1$ -LUTs,  $K_2$ -LUTs,  $\dots$ , 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. The HeteroMap algorithm generates favorable results for Xilinx XC4000 series FPGAs and Lucent ORCA2C series FPGAs. Furthermore, the optimality of the HeteroMap algorithm enables us to quantitatively evaluate various heterogeneous architectures without the bias of mapping heuristics.

## 1 Introduction

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, each PLB in XC4000 series FPGAs [10] can be configured to implement either two separate 4-LUTs or a single 5-LUT, while each PLB in ORCA2C series FPGAs can be configured to implement either two separate 5-LUTs or a single 6-LUT. 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.

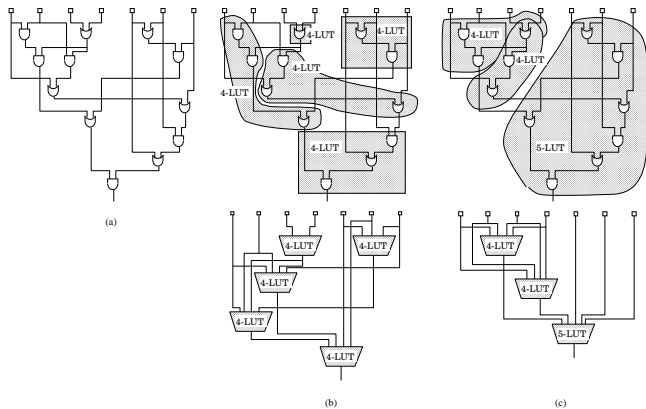


Figure 1: (a) The original circuit; (b) Mapping to 4-LUTs by Flowmap; (c) Mapping to 4-LUTs and 5-LUTs by HeteroMap.

Reference [4] provides a comprehensive survey of the existing technology mapping algorithms for homogeneous LUT-based FPGAs. However, none of these algorithms are able to deal with the delay optimization problem for heterogeneous FPGAs, as they assume the identical capacity and delay for every LUT. In [7], 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 their algorithm can not be easily extended for the general heterogeneous FPGAs employing three or more types of logic blocks.

In this paper, we present a general technology mapping problem formulation for heterogeneous FPGAs and a polynomial-time optimal technology mapping algorithm, named HeteroMap, for delay minimization in heterogeneous FPGA designs. The remainder of this paper is organized as follows. Section 2 presents the problem formulation and preliminaries. Section 3 describes the HeteroMap algorithm. Experimental results and comparative study are presented in Section 4. Section 5 concludes the paper. Due to the page limitation, the detailed proofs of theorems are left out, which are available in [6].

## 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, 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  $v$ . 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,  $K_1$ -LUT,  $K_2$ -LUT,  $\dots$ , and  $K_c$ -LUT ( $K_1 < K_2 < \dots < K_c$ ) with the delay of  $d_1, d_2, \dots$ , and  $d_c$  ( $d_1 < d_2 < \dots < d_c$ ). For a circuit mapped into a heterogeneous FPGA, we assume different access delays for heterogeneous LUTs but a constant delay for the interconnection<sup>1</sup>, which we call *heterogeneous LUT-delay model*. The unit-delay model used in [2] 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 follows: *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,  $\dots$ , and  $K_c$ -LUTs such that 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.

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  $(X, \bar{X})$ , 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.,  $n(X, \bar{X}) \leq K$ . 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.,  $vol(X, \bar{X}) = |\bar{X}|$ .

### 3 Delay Optimal Mapping for Heterogeneous FPGAs

In this section, we present a delay optimal technology mapping algorithm for heterogeneous FPGAs under the heterogeneous LUT-delay model. Applicable to any  $K$ -bounded ( $K \leq K_1$ ) Boolean network, HeteroMap runs in two phases. In the first phase, according to the topological order from PI to PO, HeteroMap uses the dynamic programming 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, 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 heterogeneous 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 sub-network 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* heterogeneous 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

<sup>1</sup>The constant interconnection delay can be counted into the LUT delays so that the interconnection delay can be set to zero.

$N_t$ . Figure 2(a) shows a Boolean network in which gate  $t$  is to be labeled. Figure 2(b) shows the construction of the network  $N_t$ . Assuming that  $LUT(t)$  is the  $K$ -LUT that implements node  $t$  in an optimal mapping solution of  $N_t$ . Since  $K$  may be any of  $K_1, K_2, \dots$ , or  $K_c$ , we compute  $l_i(t)$  for each  $i = 1, 2, \dots$ , 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. We have

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

In order to compute  $l_i(t)$ , 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  $i$  ( $i = 1, 2, \dots, c$ ), 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

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

Based on the above discussion, we have

**Lemma 1** *The label  $l(t)$  computed by Eq. 2 gives the minimum delay of any mapping solution of  $N_t$  under the heterogeneous LUT-delay model.*

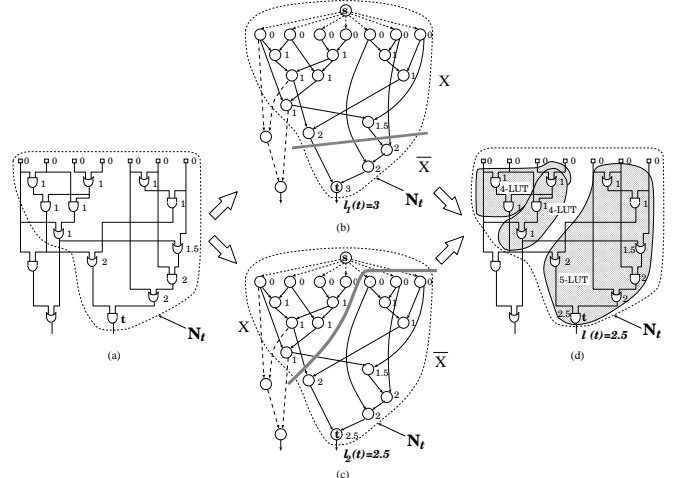


Figure 2: (a) Gate  $t$  to be labeled; (b) Construct  $N_t$  and compute the minimum height 4-feasible cut in  $N_t$  such that  $l_1(t) = 3$ ; (c) Compute the minimum height 5-feasible cut in  $N_t$  such that  $l_2(t) = 2.5$ ; (d) Determine  $l(t) = \min\{l_1(t), l_2(t)\} = 2.5$ .

In Figure 2, 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 2(b) computes  $l_1(t)$  to be 3, assuming that  $t$  is implemented by a 4-LUT. Similarly, Figure 2(c) computes  $l_2(t)$  to be 2.5, assuming that  $t$  is implemented by a 5-LUT. In Figure 2(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 2(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  $l(t) \geq l(t')$ .*

According to Eq. 2, the key problem in computing  $l(t)$  is to compute the minimum height  $K_i$ -feasible cut  $(X_i, \bar{X}_i)$  in  $N_t$  to get  $l_i(t)$  ( $i = 1, 2, \dots, c$ ). 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  $N$ , and  $I_t(i) = \max\{l(u) : u \in \text{input}(t)\}$ , then,  $I_t(i) - d_i \leq H_t(i) \leq I_t(i)$ .*

Based on Lemma 3, we get a height array, denoted  $H_{t_{i:st}}(i)$ , which includes all the distinct labels  $l(u)$  ( $u \in N_t$ ) in the specified range. Obviously, the maximum size of the height array is  $|N_t|$ . We can then perform a binary search over  $H_{t_{i:st}}(i)$  to determine  $H_t(i)$  and hence  $l_i(t) = H_t(i) + d_i$ . To compute a  $K$ -feasible cut of height  $H$  in  $N_t$ , we apply the approach proposed in [2].

**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  $N_t$ .*

By computing  $l_i(t)$  ( $i = 1, 2, \dots, c$ ),  $l(t)$  can be determined 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, 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

The mapping phase of the HeteroMap algorithm generates the LUTs in the optimal solution in a similar way as that in [2], except that the LUTs generated in our mapping solution can be heterogeneous. The entire second phase takes linear time.

### 3.3 Area Minimization in the HeteroMap Algorithm

The secondary objective of our technology mapping algorithm is area optimization, which is considered by maximizing the volume of each cut [2] and by the post-mapping heterogeneous predecessor packing operations, extended from the predecessor packing technique proposed in [1] to the heterogeneous mapping solution.

## 4 Experimental Results and Comparative Study

### 4.1 Mapping Comparison on Heterogeneous FPGAs

We first compare FlowMap [2] and HeteroMap on XC4000 series FPGAs, which can implement circuits with 4-LUTs and 5-LUTs of delay 1 and  $1.5^2$ . Match4K [3] is an intelligent post-mapping multi-step matching heuristic for XC4000 devices to reduce the PLB number. We compare FlowMap with HeteroMap both followed by Match4K on MCNC benchmarks, and perform layout using the same XC4000 part (4013MQ208-5) on XACT5.2 FPGA development system. For FlowMap, we set  $K = 5$ . Table 1 shows that HeteroMap reduces 19% of the mapping delays (M-Dly) and 7% of the post-layout delays (PL-D) with only 2% increase on the PLB numbers over FlowMap.

<sup>2</sup>This delay ratio is derived from [10].

Circuits	FlowMap + Match4K			HeteroMap + Match4K		
	M-Dly	PLB	PL-D (ns)	M-Dly	PLB	PL-D (ns)
5xp1	4.50	16	41.00	3.50	14	36.70
9sym	7.00	35	54.10	5.50	39	53.10
9symml	7.00	37	60.60	5.50	38	53.00
C499	6.50	83	80.70	5.50	88	76.80
C880	12.00	132	140.20	10.00	132	106.80
alu2	12.00	94	110.10	10.50	94	107.50
alu4	15.00	149	128.40	11.50	163	160.20
apex2	7.00	87	84.00	6.50	91	70.70
apex7	6.00	46	55.00	4.00	49	52.70
count	4.50	44	52.90	4.00	39	46.90
duke2	6.00	99	75.90	4.50	103	69.90
e64	4.50	83	68.00	3.50	91	63.10
misex1	3.00	10	30.30	2.50	11	31.00
rd84	6.00	29	53.70	4.50	26	49.80
vg2	5.00	23	55.00	4.00	23	44.30
z4ml	3.50	7	34.70	3.50	8	34.50
TOTAL	109.50	974	1124.60	89.00	1009	1057
A_Mean	6.84	60.88	70.29	5.56	63.06	66.06
A_Ratio	1	1	1	-19%	+3%	-6%
G_Mean	6.22	44.23	64.24	5.05	45.32	59.77
G_Ratio	1	1	1	-19%	+2%	-7%

Table 1: 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  $1.2^3$ . 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 2. Compared with FlowMap, HeteroMap reduces 9% of the mapping delays (M-Dly) and 8% of the area in terms of the PLB number.

### 4.2 Heterogeneous v.s. Homogeneous Architectures

The optimality of the HeteroMap algorithm enables us to quantitatively evaluate various heterogeneous architectures using MCNC benchmarks without the bias of mapping heuristics. 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 delay 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 comparison results on both area and mapping delay are summarized in Table 3. 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 3, 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, heterogeneous FPGAs always achieve much better mapping delay than that of homogeneous FPGAs. 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 3 is plotted with  $1 \leq r \leq 3$  as the X-axis and  $\text{area} \times \text{delay}^2$  as the Y-axis [9]. From Figure 3 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.

<sup>3</sup>This delay ratio is derived from [8].

Circuits	FlowMap		HeteroMap	
	M-Dly	PLB	M-Dly	PLB
5xp1	2.20	12	2.20	12
9sym	4.60	30	4.20	30
9symml	4.60	28	4.20	30
C499	5.40	104	5.00	85
C880	8.20	141	8.00	138
alu2	8.40	95	8.00	97
alu4	9.40	191	8.60	168
apex2	5.60	98	5.00	97
apex4	6.00	506	5.20	406
apex6	4.80	143	4.00	132
apex7	3.40	53	3.20	49
count	3.60	42	3.00	36
des	3.60	457	3.40	458
duke2	4.40	110	4.00	106
e64	3.60	111	3.00	94
misex1	2.40	8	2.00	7
rd84	4.40	24	4.00	23
rot	6.00	165	5.60	153
vg2	3.40	24	3.20	25
z4ml	2.20	4	2.20	4
TOTAL	96.20	2336	88.00	2150
A_Mean	4.81	116.80	4.40	107.50
A_Ratio	1	1	-9%	-8%
G_Mean	4.44	61.89	4.04	58.40
G_Ratio	1	1	-9%	-6%

Table 2: Comparison between FlowMap and HeteroMap on ORCA2C series FPGAs

More detailed results are available in [6]. Table 4 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 delay optimal mapping solution.

Comparison	FlowMap on homogeneous FPGAs of $K$ -LUTs			
	$K = 3$		$K = 4$	
	M-Dly	Area	M-Dly	Area
A_Mean	8.10	352.70	8.06	537.20
A_Ratio	+25%	-34%	+25%	+0.2%

FlowMap on homogeneous FPGAs of $K$ -LUTs				HeteroMap on Hetero-FPGAs	
$K = 5$		$K = 6$		M-Dly	Area
M-Dly	Area	M-Dly	Area		
8.08	870.40	8.30	1273.60	6.45	536.15
+25%	+62%	+28%	+137%	1	1

Table 3: Homogeneous FPGAs *v.s.* heterogeneous FPGAs

Delay ratio = 1 : 1.33 : 1.67 : 2				Delay Ratio = 1 : 1.2 : 1.4 : 1.6			
3-lut	4-lut	5-lut	6-lut	3-lut	4-lut	5-lut	6-lut
1	1	1	1	-36%	-29%	+90%	+49%

Table 4: HeteroMap results with different delay ratios

## 5 Conclusions and Future Work

In this paper, we presented a delay optimal technology mapping algorithm, named HeteroMap, for heterogeneous FPGA designs.

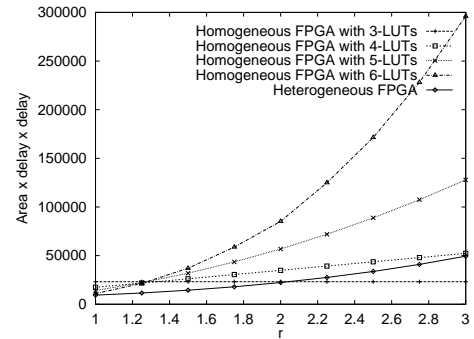


Figure 3:  $Area \times delay^2$  Comparison between homogeneous and heterogeneous FPGAs with respect to the area ratio of  $r$ .

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. Some heterogeneous FPGAs may have resource limitations on the LUTs of specific sizes, which we call *heterogeneous FPGAs with bounded resources*. The delay-optimal technology mapping for this type of heterogeneous FPGAs is still an open problem, and we are in the process of developing optimal or near-optimal algorithms for this 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

- [1] K. C. Chen, J. Cong, Y. Ding, A. B. Kahng, P. Trajmar, "DAG-Map: Graph-based FPGA Technology Mapping for Delay Optimization", IEEE Design and Test of Computers, Sep. 1992, pp. 7-20.
- [2] 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.
- [3] J. Cong, J. Peck, Y. Ding, "RASP: A General Logic Synthesis System for SRAM-based FPGAs", Proc. ACM International Symposium on FPGA, Feb. 1996, pp. 137-143.
- [4] 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.
- [5] 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.
- [6] J. Cong, S. Xu, "Delay-Optimal Technology Mapping for FPGAs with Heterogeneous LUTs", UCLA Computer Science Department Technical Report CSD-980015.
- [7] J. He, J. Rose, "Technology Mapping for Heterogeneous FPGAs", Proc. ACM/SIGDA International Symposium on FPGA, Feb. 1994.
- [8] Lucent Technologies, "ORCA OR2C-A/OR2T-A Series FPGAs Data Sheet", Lucent Technologies, Inc., Allentown, PA, 1996.
- [9] J. D. Ullman, "Computational Aspects of VLSI", Computer Science Press, Inc., Rockville, MD, 1983.
- [10] Xilinx, "The Programmable Logic Data Book", Xilinx Inc., San Jose, CA, 1997.