

# Structural Gate Decomposition for Depth-Optimal Technology Mapping in LUT-based FPGA Design

Jason Cong and Yean-Yow Hwang

Department of Computer Science  
University of California, Los Angeles

## Abstract

In this paper, we study the problem of decomposing gates in fanin-unbounded or  $K$ -bounded networks such that the  $K$ -input LUT mapping solutions computed by a depth-optimal mapper have minimum depth. We show (1) any decomposition leads to a smaller or equal mapping depth regardless the decomposition algorithm used, and (2) the problem is NP-hard for unbounded networks when  $K \geq 3$  and remains NP-hard for  $K$ -bounded networks when  $K \geq 5$ . We propose a gate decomposition algorithm, named DOGMA, which combines level-driven node packing technique (Chortle-d) and the network flow based optimal labeling technique (FlowMap). Experimental results show that networks decomposed by DOGMA allow depth-optimal technology mappers to improve the mapping solutions by up to 11% in depth and up to 35% in area comparing to the mapping results of networks decomposed by other existing decomposition algorithms.

## 1. Introduction

The lookup-table (LUT) based FPGAs have been a popular technology for VLSI ASIC design and system prototyping. A  $K$ -input LUT ( $K$ -LUT) can implement any function of up to  $K$  variables. The goal of LUT-based FPGA technology mapping is to cover a given network using LUTs such that either area or delay is minimized or routability is maximized in the final LUT network. The delay of a network can be estimated by the number of levels (i.e. depth). Two factors affect the mapping solution depth: the gate decomposition before mapping and the mapping algorithm. Several LUT mapping algorithms have been proposed for depth minimization [6, 9, 2]. In particular, the FlowMap algorithm [2] guarantees a depth-optimal mapping solution for any  $K$ -bounded network. However, FlowMap can not be applied directly to unbounded networks. Gate decomposition can be classified into structural decomposition and Boolean decomposition. The structural decomposition replaces multi-fanin (simple) gates by fanin trees while the Boolean decomposition decomposes the functionality of gates. This paper focuses on structural decomposition for depth minimization in LUT mapping.

Gate decomposition affects the mapping solution depth significantly. For example, assume  $K=3$ . The network  $N$  in Figure 1(a) is not  $K$ -bounded. If node  $v$  is decomposed in the way shown in Figure 1(b), there is no way to obtain a mapping solution of depth less than 3. However, if the decomposition shown in Figure 1(c) is carried out for node  $v$ , a mapping solution of depth equal to 2 can be obtained. Even for  $K$ -bounded networks, the depth of mapping solutions computed by FlowMap may decrease if gates are *further* decomposed before mapping [4].

Several gate decomposition routines have been used for LUT-mapping. The *tech\_decomp* and the *speed\_up* in SIS [10] and the *dmig* in [1] focus on minimizing the number of levels in the decomposed network. They do not directly minimize the depth of the *mapping solution*. Chortle-d [6] computes depth-optimal gate decomposition and mapping solutions for tree networks (may be unbounded) but produces suboptimal results for general networks. In this paper, we study the structural gate decomposition problem to decompose gates in a general network such that a mapping solution of minimum depth can be obtained.

The remainder of this paper is organized as follows. Section 2 defines basic terminology, presents properties of structural gate decomposition and formulates the problem. Section 3 gives the complexity results. A novel gate decomposition algorithm for depth-optimal mapping is presented in Section 4. Experimental results are presented in Section 5 and Section 6 concludes the paper.

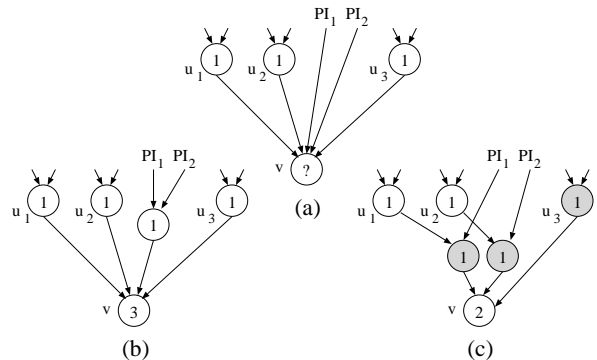


Figure 1 Decomposition of node  $v$  ( $K=3$ ). (a) initial network, (b) decomposition yielding mapping depth = 3, (c) decomposition yielding mapping depth = 2.

## 2. Problem Formulation

Let  $K$  be the input size of an LUT. Let  $input(v)$  be the set of fanin nodes of node  $v$ . A primary input (PI) node has no fanins and a primary output (PO) node has no outgoing edges. A network  $N$  is  $K$ -bounded if every node  $v \in N$  satisfies  $|input(v)| \leq K$ . Otherwise, it is an unbounded network. Given a subnetwork  $H$ , we use  $input(H)$  to denote the set of distinct nodes outside  $H$  which supply inputs to nodes in  $H$ . Given a node  $v$  in network  $N$ , let  $N_v$  denote the subnetwork consisting of node  $v$  and all the predecessors of  $v$ . The *minimum mapping depth* of  $v$  in  $N$ , denoted  $MMD_N(v)$ , is defined as the minimum depth among all possible  $K$ -LUT mapping solutions of  $N_v$ . If  $N_v$  is unbounded, let  $MMD_N(v) = \infty$ . PI nodes have a mapping depth of 0. The minimum mapping depth of a network  $N$ , denoted  $MMD(N)$ , is the largest mapping depth among all PO nodes. Given a  $K$ -bounded network  $N$ , the FlowMap [2] algorithm computes  $MMD_N(v)$  for every node  $v \in N$  in polynomial time. A cut in  $N_v$  is a partition  $(X_v, \bar{X}_v)$  of nodes in  $N_v$  such that PI nodes are in  $X_v$  and  $v \in \bar{X}_v$ . The cutset of a cut, denoted  $n(X_v, \bar{X}_v)$ , is defined as  $input(\bar{X}_v)$ . A cut is  $K$ -feasible if  $|n(X_v, \bar{X}_v)| \leq K$ . The height of a cut, denoted  $height(X_v, \bar{X}_v)$ , is the maximum mapping depth for nodes in  $n(X_v, \bar{X}_v)$ . We have the following lemma based on results in [2].

**Lemma 1** A node  $v$  has  $MMD_N(v) = p$  if there is a  $K$ -feasible cut of height of  $p - 1$  in  $N_v$  but no  $K$ -feasible cut of height of  $p - 2$  or smaller exists.

Let node  $v \in N$  satisfies  $|input(v)| > 2$ . Given a decomposition algorithm  $D$ , we define a *decomposition step* at  $v$  by  $D$ , denoted  $D_v$ , as follows: Decomposition step  $D_v$  (i) chooses two nodes  $u_1$  and  $u_2$  from  $input(v)$ , (ii) removes edges  $(u_1, v)$  and  $(u_2, v)$ , (iii) introduces a new node  $w$  and new edges  $(u_1, w)$ ,  $(u_2, w)$ ,  $(w, v)$  and adds them to  $N$ . The resulting network is denoted as  $D_v(N)$ . For example, Figure 2(b) shows the result of one decomposition step at node  $v$  from Figure 2(a). Obviously, the introduced node  $w$  have the same gate type as  $v$ . We present two properties of the structural gate decomposition.

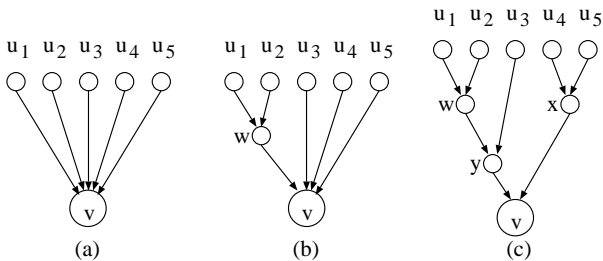


Figure 2 Decomposition of node  $v$ . (a) Before decomposition, (b)  $D_v(N)$  after one decomposition step of  $D_v$ , (c) after a sequence of decomposition steps.

**Lemma 2** Given a network  $N$ , any decomposition algorithm  $D$ , and any node  $v \in N$ , it must be true that  $MMD(D_v(N)) \leq MMD(N)$ .

**Lemma 3** If  $MMD_N(u) = MMD_N(v)$  for all  $u \in input(v)$  in a  $K$ -bounded network  $N$ , then  $MMD(N) = MMD(D_v(N))$  for any decomposition algorithm  $D$ .

According to Lemma 2, the further a network is decomposed, the smaller the mapping depth might be. Therefore, we decompose every gate into a binary fanin tree. Figure 2(c) is a complete decomposition of node  $v$ . Every decomposition step introduces one intermediate node and it requires  $|input(v)| - 2$  steps to decompose  $v$ . We formulate the following problems.

**Structural Gate Decomposition for K-LUT Mapping (SGD/K)** Given a simple-gate unbounded network  $N_\infty$ , decompose  $N_\infty$  into a 2-input network  $N_2$  such that for any other 2-input network decomposition  $N'_2$  of  $N$ ,  $MMD(N_2) \leq MMD(N'_2)$ .

**Structural Gate Decomposition for K-LUT Mapping of K-bounded Network (K-SGD/K)** Given a simple-gate  $K$ -bounded network  $N_K$ , decompose  $N_K$  into a 2-input network  $N_2$  such that for any other 2-input network decomposition  $N'_2$  of  $N$ ,  $MMD(N_2) \leq MMD(N'_2)$ .

There are two issues related to the problem of gate decomposition. (1) A smaller depth might be obtained when several gates are decomposed *simultaneously* instead of *independently* [4]. This is because the intermediate nodes could be shared during the decomposition of multiple gates of the same functional type. (2) Gate decomposition can be performed before the mapping phase in a *two-step* approach or embedded into the mapping process being part of an *integrated* approach. For example, Chortle-d [6] is an integrated approach (since it decomposes gates and maps LUTs in an interleaving manner) while *dmig* + FlowMap in [2] uses a two-step approach. We can show that the *best* two-step approach produces the same optimal mapping depth as that by the *best* integrated approach [4]. In this paper, we consider only independent gate decompositions in a two-step approach. Nevertheless, our gate decomposition algorithm takes into account the impact of gate decomposition on mapping to obtain a decomposed network which is most suitable for FlowMap to achieve a minimum mapping depth.

## 3. Complexity of SGD/K and K-SGD/K Problems

In this section, we only state our complexity theorems. Complete proofs can be found in [4].

**Theorem 1** The SGD/K problem is NP-hard for  $K \geq 3$ .

**Theorem 2** The K-SGD/K problem is NP-hard for  $K \geq 5$ .

## 4. Gate Decomposition Algorithm for Depth-Optimal LUT Mapping

Our decomposition algorithm, named DOGMA (Depth-Optimal Gate decomposition for MAPPING),

combines the level-driven node packing technique in Chortle-d and the network flow based labeling technique in FlowMap. Given a network  $N$ , DOGMA decomposes nodes from PIs to POs in a topological order. Let  $N(v)$  denote the network after decomposing the node  $v$ . DOGMA labels each node  $v$  by  $MMD_{N(v)}(v)$  as follows. Nodes in  $input(v)$  are grouped in such a way that each group consists of nodes with the same label (i.e. minimum mapping depth). Groups are processed in an ascending order according to their labels. For a group of nodes labeled  $p$ , nodes are packed into a minimum number of bins such that a  $K$ -feasible cut of height  $p - 1$  exists for the nodes in each bin (checked based on Lemma 1). Such a bin is called a *min-height  $K$ -feasible bin*. A node  $u_i$  is created for each bin  $B_i$  with fanins from nodes in  $B_i$  and a fanout to  $v$ . Node  $u_i$  will be given a label  $p$ . Note that according to Lemma 3, no matter  $u_i$  is further decomposed or not, the minimum mapping depth of the network is always the same. We then proceed to the group of a next higher label  $p + 1$ . For each node  $u_i$  created in the previous step, a buffer node  $w_i$  (with label  $p + 1$ ) is created with  $u_i$  as input. (All the buffer nodes will be removed after decomposition). These buffer nodes together with nodes in the group of label  $p + 1$  are again packed into a minimal number of min-height  $K$ -feasible bins. We continue this process until all nodes are packed into one bin which corresponds to the node  $v$ .

DOGMA is similar to Chortle-d in that decomposition is done by packing nodes into a *minimal* number of min-height  $K$ -feasible bins. However, Chortle-d integrates gate decomposition with technology mapping, and computes mapping depth based on the partially generated LUT network. Since the fanin constraint is not a *monotone* clustering constraint [2], Chortle-d may obtain inaccurate node mapping depth. Besides, Chortle-d enumerates all packing combinations for nodes on reconvergent paths, which is quite expensive. In contrast, DOGMA computes mapping depth as well as packs nodes into bins (to be discussed) using the network flow based computation. The mapping depth is always accurate and the reconvergent paths are taken into account naturally.

The problem remains to be solved is the min-height  $K$ -feasible bin packing problem defined as follows.

**Min-Height  $K$ -Feasible Bin Packing Problem** Given a set  $S_p \subseteq input(v)$  of nodes of minimum mapping depth  $p$  when decomposing node  $v$ , partition  $S_p$  into a minimal number of bins such that there is a  $K$ -feasible cut of height  $p - 1$  for nodes in each bin.

We shall give three heuristics and one exact method to solve the problem. Our heuristics are based on the max-flow algorithm and bin-packing heuristics. We define the total cut size  $TC_p(S)$  of a set  $S$  of nodes with label  $p$  to be the size of the min-cut of height  $p - 1$  which separates  $S$  from all PIs. A set  $X$  of nodes of label  $p$  can be packed into one bin as long as  $TC_p(X) \leq K$ . Nodes are packed in a decreasing order of individual cut sizes. The first two heuristics, named MC-FFD and MC-BFD, to the min-height

$K$ -feasible bin packing problem are based on the first-fit-decreasing (FFD) and best-fit-decreasing (BFD), which are two heuristics for the bin packing problem [8]. The third method is the maximal-share-decreasing (MC-MSD) heuristic which packs nodes that can maximally share a cut together. The fourth method is inspired by the dynamic programming approach for the number partitioning problem [7]. Instead of partitioning numbers, we ask whether there is a way to partition the nodes in  $S_p$  into  $k$  subsets  $X_1, X_2, \dots, X_k$  such that  $TC_p(X_i) \leq K$  ( $1 \leq i \leq k$ ). We can solve the problem by dynamic programming. By searching the minimal  $k$  ( $k = 2, 3, \dots$ ), the min-height  $K$ -feasible bin packing problem can be solved optimally. We refer to this method as the MC-DP algorithm. The details of these algorithms can be found in [4].

## 5. Experimental Results

We have implemented the DOGMA algorithm with MC-FFD, MC-BFD, MC-MSD, and MC-DP packing methods using the C language and incorporated our implementation into the RASP FPGA synthesis system [5]. In our experiments, we optimize the MCNC benchmark circuits for area using standard SIS scripts, decompose them into simple gate networks, apply gate decomposition routines to obtain 2-input networks, and obtain the final LUT networks using a depth-optimal technology mapper. We choose  $K = 5$  in the experiments.

We compare the performance of our four methods for the min-height  $K$ -feasible bin packing in DOGMA and observe that the impact of the four methods on mapping results is almost the same. Since MC-FFD is faster than other three methods, DOGMA employs MC-FFD to solve the packing problem. We compare DOGMA with two decomposition routines: the *tech\_decomp* in SIS [10] and the *dmig* in [1]. The *tech\_decomp* routine is based on a balanced-tree heuristic which only minimizes the gate level locally. The *dmig* routine minimizes the gate level of the decomposed networks. The decomposed networks by the three algorithms are all mapped by CutMap [3], an enhancement of FlowMap. The results are shown in Table 1. We see CutMap produces the same or smaller depth for circuits decomposed by DOGMA. On average, DOGMA allows CutMap to achieve 10% and 4% depth reduction comparing to *tech\_decomp* and *dmig*, respectively.

We compare DOGMA + CutMap with existing gate decomposition and depth-oriented mapping algorithms. The tested circuits are area-optimized MCNC benchmarks. The mappers TechMap-D [9], FlowMap [2], and CutMap [3] are used for comparison. The *dmig* was used in [2] while the *speed\_up* was used in [9] and [3] to prepare 2-input networks for technology mapping. The results are in Table 2. Comparing results from [2, 3] with ours, we see gate decomposition routines *speed\_up*, *dmig*, and DOGMA decompose gates equally well in terms of mapping depth. However, networks decomposed by DOGMA allow CutMap to reduce 16% of LUTs in the mapping solutions.

Circuit	<i>tech_decomp</i>		<i>dmig</i>		DOGMA	
	LUT	<i>d</i>	LUT	<i>d</i>	LUT	<i>d</i>
5xp1	24	3	23	3	24	3
9sym	66	5	66	5	59	5
apex2	154	6	155	5	151	5
apex4	770	7	792	6	770	5
clip	37	4	37	4	38	3
con1	3	2	3	2	3	2
duke2	160	5	173	4	177	4
e64	108	9	108	9	108	9
misex1	18	2	18	2	16	2
misex2	32	3	31	3	36	2
misex3	179	17	174	16	176	16
rd73	25	3	27	3	23	3
rd84	52	4	52	4	54	4
sao2	47	4	44	4	42	4
vg2	23	4	29	3	29	3
Total	1698	78	1732	73	1706	70
	-0.5%	+11%	+2%	+4%	1	1

Table 1 Comparison of *tech\_decomp*, *dmig* and DOGMA.

In [9], TechMap-D obtained the smallest depth because it integrated logic synthesis into technology mapping. However, 35% more LUTs are generated comparing to DOGMA + CutMap.

Circuit	[9]	[2]	[3]	Ours
	<i>speed_up</i>	<i>dmig</i>	<i>speed_up</i>	DOGMA
	TechMap-D	FlowMap	CutMap	CutMap
	LUT( <i>d</i> )	LUT( <i>d</i> )	LUT( <i>d</i> )	LUT( <i>d</i> )
5xp1	17 (2)	22 (3)	23 (3)	24 (3)
9sym	9 (3)	60 (5)	62 (5)	59 (5)
9symml	9 (3)	55 (5)	58 (5)	50 (4)
C499	148 (4)	68 (4)	143 (5)	68 (4)
C880	213 (7)	124 (8)	205 (8)	98 (8)
alu2	197 (8)	155 (9)	144 (8)	138 (9)
apex6	252 (5)	238 (5)	233 (4)	231 (5)
apex7	86 (4)	79 (4)	80 (4)	68 (4)
count	71 (4)	31 (5)	69 (3)	31 (5)
des	1395 (8)	1310 (5)	986 (5)	938 (5)
duke2	175 (4)	174 (4)	178 (4)	173 (4)
misex1	18 (2)	16 (2)	15 (2)	16 (2)
rd84	16 (3)	46 (4)	45 (4)	53 (4)
rot	315 (6)	234 (7)	239 (6)	210 (7)
vg2	36 (4)	29 (3)	39 (4)	27 (3)
z4ml	9 (2)	5 (2)	12 (3)	5 (2)
Total	2966(69)	2646(75)	2531(73)	2189(74)
	+35%(-7%)	+21%(+1%)	+16%(-1%)	1 (1)

Table 2 Comparison of DOGMA + CutMap with previous results.

## 6. Conclusion

In this paper, we study the structural gate decomposition for depth-optimal LUT mapping. We show gate decomposition leads to a smaller or equal mapping depth regardless the decomposition algorithm used, and the problem is NP-hard for unbounded networks when  $K \geq 3$  and remains NP-hard for  $K$ -bounded networks when  $K \geq 5$ . We propose a gate decomposition algorithm (DOGMA) for depth-optimal mapping. Experimental results show a reduction of up to 10% in mapping depth. Together with CutMap, we achieve comparable mapping depth with up to 35% reduction in area comparing to previous results.

## Acknowledgement

This work is partially supported by NSF Young Investigator (NYI) Award MIP-9357582, and grants from Xilinx, Xerox PARC, and AT&T Microelectronics under NSF NYI and California MICRO programs. The authors would like to thank Dr. Robert Francis for his helpful discussions.

## References

- [1] Chen, K. C., J. Cong, Y. Ding, A. B. Kahng, and P. Trajmar, "DAG-Map: Graph-based FPGA Technology Mapping for Delay Optimization," *IEEE Design and Test of Computers*, pp. 7-20, Sep. 1992.
- [2] Cong, J. and Y. Ding, "An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs," *IEEE Trans. on Computer-Aided Design*, Vol. 13, pp. 1-12, Jan. 1994.
- [3] Cong, J. and Y.-Y. Hwang, "Simultaneous Depth and Area Minimization in LUT-Based FPGA Mapping," *Proc. ACM 3rd Int'l Symp. on FPGA*, pp. 68-74, Feb. 1995.
- [4] Cong, J. and Y.-Y. Hwang, "Structural Gate Decomposition for Depth-Optimal Technology Mapping in LUT-based FPGA," in *UCLA Computer Science Dept. Tech. Report CSD-950045*, (December 1995).
- [5] Cong, J., J. Peck, and Y. Ding, "RASP: A General Logic Synthesis System for SRAM-based FPGAs," *Proc. ACM 4th Int'l Symp. on FPGA*, Feb. 1996.
- [6] Francis, R. J., J. Rose, and Z. Vranesic, "Technology Mapping of Lookup Table-Based FPGAs for Performance," *Proc. IEEE Int'l Conf. on CAD*, pp. 568-571, Nov. 1991.
- [7] Garey, M. and D. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco (1979).
- [8] Horowitz, E. and S. Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, Potomac, Maryland (1978).
- [9] Sawkar, P. and D. Thomas, "Performance Directed Technology Mapping for Look-Up Table Based FPGAs," *Proc. 30th ACM/IEEE Design Automation Conf.*, pp. 208-212, June 1993.
- [10] Sentovich, E., K. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. Stephen, R. Brayton, and A. Sangiovanni-Vincentelli, "SIS: A System for Sequential Circuit Synthesis," *U.C. Berkeley Technical Report UCB/ERL M92/41*, May, 1992.