Advantages of Heterogeneous Logic Block Architectures for FPGAs*

Jianshe He and Jonathan Rose
Department of Electrical and Computer Engineering,
University of Toronto, Toronto, M5S 1A4, Canada

Abstract

Most FPGAs use an array of identical logic blocks, largely because such devices are easy to design. Previous studies on logic block architecture, however, have concluded that while 4-input lookup tables (4-LUTs) make efficient use of area [Rose90] [Koul92], significantly more coarse-grain blocks such as 5-LUTs, 6-LUTs and 7-LUTs are superior in terms of system speed [Koul91] [Sing91] [Sing92]. These results suggest that a mixture of different size LUTs (for example 4-LUTs and 6-LUTs) may provide a better tradeoff between speed and density. This paper presents an architectural investigation into such heterogeneous FPGAs. We consider FPGAs that use two different sizes of lookup table logic blocks, and investigate the area-efficiency of different mixtures of different sizes of LUTs. Experimental results on a set of benchmark circuits indicate that several heterogeneous architectures achieve significant reduction in the number of programming bits and logic block pins compared to the industry standard 4-input lookup tables [Hsie90] [Hill92]. Furthermore, a 6-LUT/4-LUT combination will likely exhibit better performance with nearly equivalent area than a homogeneous 4-LUT FPGA.

1 Introduction

Commercial FPGAs usually consist of an array of identical logic blocks [Cart86] [Hsie90] [ElGa89] [Ahre90] [Wong89] [Wils92] [Algo89], or logic blocks that have very similar levels of functionality. It is possible that a heterogeneous mixture of logic blocks may provide superior area (which relates to logic density) because some portions of logic may simply be more efficient with one particular type of logic block than another. For example, consider the boolean network pictured in Figure 1a. Figure 1b is a mapping of that network using 4-input lookup tables (4-LUTs) and Figure 1c is a mapping of that network using 3-LUTs. As shown in Table 1, the 4-LUT solution uses one-third more lookup table bits (64 vs 48) but 20% fewer pins than the 3-LUT solution (because a single 3-LUT has 8 bits and 4 pins and a single 4-LUT uses 16 bits and 5 pins). Suppose that the network is mapped into a heterogeneous FPGA that contains 3-LUTs and 4-LUTs in equal numbers, as illustrated in Figure 1d. This circuit uses exactly two 3-LUTs and two 4-LUTs and hence requires only 48 bits and 18 pins. This heterogeneous FPGA thus requires 25% fewer bits and 10% fewer pins than the 4-LUT homogenous FPGA. It has the same number of bits and 25% fewer pins than the 3-LUT homogenous FPGA to implement this example. This example demonstrates that a heterogeneous mixture of logic blocks may exhibit superior area-efficiency.

	LUT types	#LUTs	#Bits	#Pins
Homo-geneous	only 3-LUT	6	48	24
	only 4-LUT	4	64	20
Hetero. ratio = 1	3-LUT and 4-LUT	2 3-LUT 2 4-LUT	48	18

Table 1: Comparison Measures of Heterogeneous vs. Homogenous FPGAs for Example Circuit in Fig.1

In this paper we focus our attention on heterogeneous mixtures of *two* sizes of lookup table. The larger lookup table will be referred to as the p-LUT, and the smaller as the s-LUT. An important architectural parameter is the ratio of the number of the two types of block, $r = \frac{\#_{s-LUTs}}{\#_{p-LUTs}}$ that are present in the FPGA.

We will assume that the two types of blocks will be grouped together in a "supertile" consisting of r s-LUTs and 1 p-LUT if $r \geq 1$, and 1 s-LUT with $\frac{1}{r}$ p-LUTs if $0 < r < 1$. Thus the FPGA would consist of an array of supertiles. Figure 2a gives an example of a supertile with p = 3, s = 2, and r = 2, and Figure 2b illustrates the array. Note that the notion of a *supertile* is an abstraction intended to represent only the ratio of different types of LUTs and is not meant to imply anything about the actual physical placement and routing structure.

In this framework, we are concerned with the questions: What are the best values of p, s, and r in terms of the area-efficiency of an FPGA?

The following section describes our experimental method of answering these questions and Section 3 presents experimental results.

*This work was supported by a grant from ITRC & NSERC Operating Grant #URF0043928.

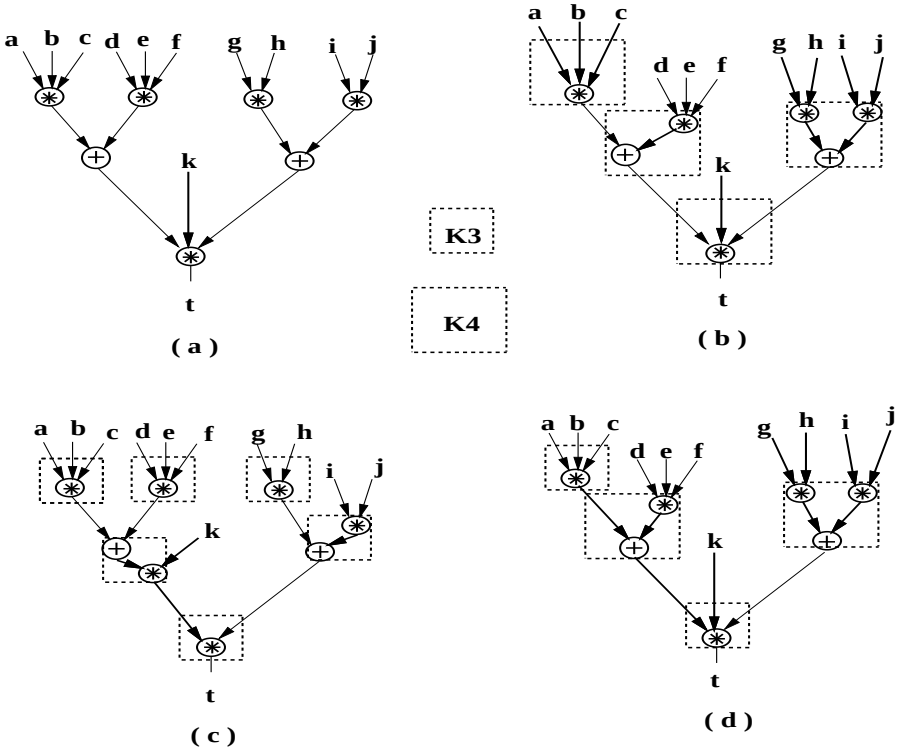

Figure 1: An illustration of homogeneous and heterogeneous mappings

2 Experimental Procedure and Area Measures

The above architectural questions will be answered using an experimental approach: a set of benchmark combinational circuits will be synthesized into a set of FPGAs. By synthesizing each circuit into FPGAs with different values of s, p and r, we can measure the area-efficiency of each such architecture for each circuit. We first discuss the synthesis procedure, and then describe the way in which the results are used to indicate area-efficiency.

2.1 Logic Synthesis for Heterogenous FPGAs

The synthesis procedure takes the combinational benchmark circuits and passes them through logic synthesis to determine a network of p-LUTs and s-LUTs that will implement the function of the input boolean network.

The key issue in synthesizing for heterogenous FPGAs comes in the technology mapping step. Combinational circuits are first optimized using technology-independent logic optimization [Bray87], which produces an optimized boolean network. Technology mapping selects which parts of the network are to be implemented using the available logic blocks. In a homogenous FPGA, (with an array of 4-LUTs, for example) the optimization goal is to minimize the total number of 4-LUTs. In a heterogenous FPGA, for example with 3-LUTs and 2-LUTs in ratio r=1, the mapping algorithm must minimize the number of *super-tiles*. This means that for every 3-LUT that is used, one 2-LUT *must* be used, and so the objective function is to minimize N_{sup}, where

$$N_{sup} = \mathbf{max}(N_3, N_2)$$

where N_3 is the number of 3-LUTs used, and N_2 is the number of 2-LUTs.

The difficulty we faced in performing this synthesis is that all logic synthesis tools for FPGAs are designed to solve the homogenous problem, in which all logic block are the same. Similarly, synthesis for standard ASIC gate arrays and standard cells does not apply because they are free to select *any* number of gate types available in the cell libraries.

To solve this problem we developed a new technology mapping algorithm which deals explicitly with non-homogeneity. That algorithm is described in [He93a] and [He93b].

The following section describes how the resulting netlist of p-LUTs and s-LUTs is used to estimate the area that would be required in an FPGA.

It should be pointed out that since our algorithm does not replicate logic at fanout nodes [He93a], the homogeneous mapping algorithm we use is also prevented from this replication. It is possible that this may affect our conclusions.

2.2 Relative Area Measurement

To determine the area of a netlist of logic blocks, one could perform the placement and global routing, and mea-

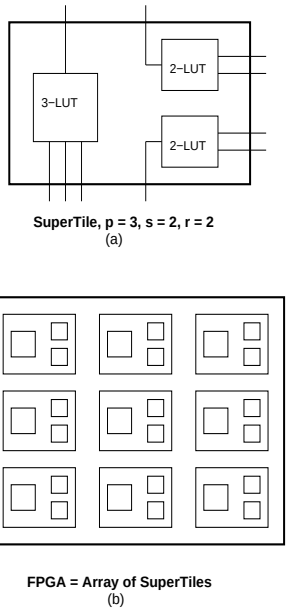

Figure 2: Example Supertile and Heterogeneous FPGA

sure the amount of wiring needed, as well as estimate the size of the logic blocks. While we took that approach in previous studies [Rose90] [Brow92], we have found that simple measures, such as counting the number of lookup table *bits* and logic block *pins* used in the design, leads to the same architectural conclusions. So, rather than going through full placement and routing, we calculate the number of pins and the number of bits to "measure" the area of a netlist in the following way:

First, calculate the number of supertiles. If the number of p-LUTs in the netlist is N_p and the number of s-LUTs is N_s, then the number of supertiles is given by:

$$N_{sup} = \mathbf{max}(N_p, \lceil \frac{N_s}{r} \rceil) \quad (1)$$

where r is the ratio of the number of s-LUTs to the number of p-LUTs in a supertile.

For a single K-LUT the number of bits is 2^K. In a supertile the number of bits is given by

$$N_{bit} = 2^p + r \times 2^s, \quad if \quad r \geq 1$$

or

$$N_{bit} = 2^s + \frac{1}{r} \times 2^p, \quad if \quad 0 < r < 1 \quad (2)$$

If the total number of supertiles used in a circuit is N_{sup}, then the total number of bits is given by

$$Total\ \#Bits = N_{sup} \times N_{bit} \quad (3)$$

Routing area is very important in the FPGA area determination, because it often requires from 50% to over 90% of the total area. For optimized placement and routing, the total number of pins of a circuit directly relates to the total amount of routing area. For this reason we count the total number of pins in evaluating the routing area.

Similar to the calculation of the number of bits, we can calculate the total number of pins of a circuit. For a supertile, the number of I/O pins, N_{pin}, is a function of p, s, and r and is given by

$$N_{pin} = (p+1) + r(s+1), \quad if \quad r \geq 1$$

or

$$N_{pin} = (s+1) + \frac{1}{r}(p+1), \quad if \quad 0 < r < 1 \quad (4)$$

where $(p+1)$ is for the p inputs and one output for a p-LUT and $(s+1)$ is for the s inputs and one output for an s-LUT.

From equations (1) and (4) we have

$$Total\ \#Pins = N_{sup} \times N_{pin} \quad (5)$$

3 Experimental Results

A total of 40 benchmark circuits from the MCNC logic synthesis benchmark suite were used as the basis for experimentation. We chose p to range from 3 to 7, and s to vary from 2 to p - 1. The ratio r was varied between 0.1 and 10.

Figure 3 plots the total number of bits of several heterogeneous combinations versus r, normalized with respect to the total number of bits of the same circuits implemented with homogenous 4-LUTs as the basic block. The total number of bits is a decreasing function of r. This figure is consistent with previous results for homogeneous FPGAs. As r increases, there are more s-LUTs which require significantly fewer bits to implement the same logic function of a circuit. The number of LUT bits, however, are not the dominant factor in total area (unless the lookup table size is larger than about 7). The number of pins to be connected is far more important [Rose90].

Figure 4 illustrates the normalized total number of pins with respect to homogeneous 4-LUTs for the same set of combinations of p and s as in Figure 3. These curves have similar shapes for all combinations of p and s, represented as (p, s) in the figure. They present several interesting results.

First, consider the combination of 4-LUT and 2-LUT with ratio $r = 0.5$. With this heterogeneous architecture, compared to a homogeneous 4-LUT FPGA, the number of bits is reduced by 22% and the number of pins is reduced by 10%. The combination of 5-LUT and 2-LUT has an 11% reduction in pins with a slight increase (11%) in bits. This data illustrate the conclusion that some heterogeneous architectures are more area-efficient than the best homogeneous architecture.

We believe that heterogeneous combinations such as (5, 2), (4, 2), (4, 3) are superior to homogeneous 4-LUT FPGAs because many logic circuits have a significant number of small fanin functions that have fanout greater than one. These can be efficiently implemented by 2 or 3-input LUTs.

Secondly, observe the shape of the curves in Figure 4. They exhibit (as do all combinations) a minimum in between the homogeneous extremes, indicating that heterogeneous architectures are always superior to homogeneous

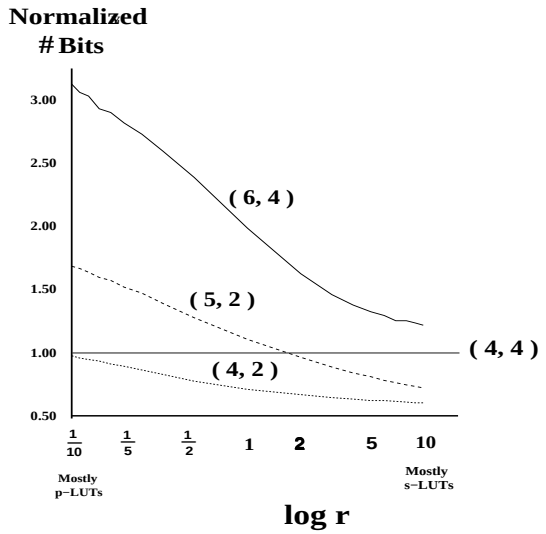

Figure 3: Normalized bit count vs r for different combinations of (p, s)

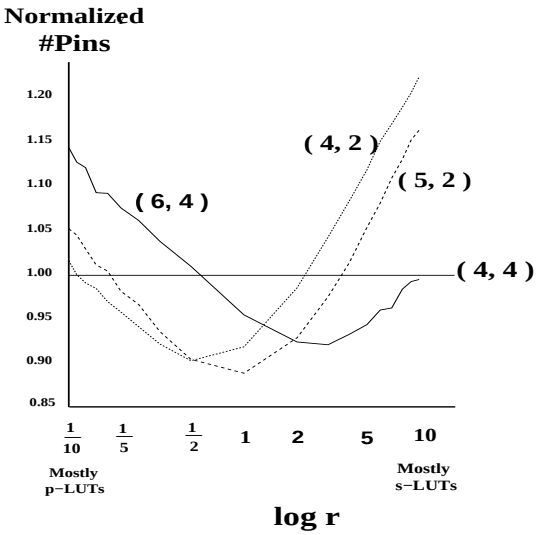

Figure 4: Normalized pin count vs r for the same set of combinations of (p, s) as in Fig.3

s \ p	7	6	5	4	3
2	1 – 2	1	1	0.5	0.5
3	2 – 3	2	1	1 – 2	
4	2 – 4	2 – 4	2 – 4		
5	2 – 6	2 – 5			
6	2 – 6				

Table 2: Value or range of ratio r that achieves minimal total pin count (within 1% difference from the minimum) for each combination of (p, s)

s \ p		7	6	5	4	3
2	pins	0.92	0.90	0.89	0.90	1.00
	bits	2.78	1.91	1.11	0.78	0.57
3	pins	0.90	0.89	0.91	0.91	
	bits	2.54	1.49	1.13	0.76	
4	pins	0.94	0.92	0.94		
	bits	2.25	1.47	1.12		
5	pins	1.00	1.01			
	bits	2.49	2.01			
6	pins	1.09				
	bits	3.76				

Table 3: Normalized minimum value of pin count and its corresponding normalized bits for each combination of p and s with respect to homogeneous 4-LUT

architectures. This shows that, as in the case of the example in section 1, most circuits benefit from having the choice of two different kinds of blocks.

Thirdly, The minimum values of r are different for different values of p and s, as shown in Table 2. Table 2 gives the value or range (within 1% difference of pin count from the minimum) of r that achieved the minimum total pin count for each combination (p, s). The ratio r at minimal pin count in all cases favors the LUT size that is closest to 4, which makes sense since that is the best homogeneous block. Table 3 gives the minimum value of pin count (normalized to the pin count of homogeneous 4-LUT) and its corresponding bit count (normalized to the homogeneous bit count) for each (p, s) corresponding to the ratio in Table 2.

The combination of (6, 4) is also worth noting. In this combination with ratios 3 and 4, the total numbers of pins are reduced by 8% and 7% with bit number increasing by about 50% and 40%, respectively. We believe that the increase in number of bits is nearly offset by the decrease in number of pins because pins dominate the area. Thus this architecture has nearly equivalent area as homogeneous 4-LUT FPGA. From previous research [Sing92], however, we expect a 6-LUT architecture to have roughly 25% less delay than a 4-LUT and so we can expect this combination to exhibit superior speed to the homogeneous 4-LUT FPGA.

4 Conclusions and Future Work

This paper has investigated the area-efficiency advantages of heterogeneous logic block FPGA architectures. We conclude that certain heterogeneous FPGAs exhibit better area than the most area-efficient homogeneous FPGA. The best ratio r corresponding to these minimum differs depending on the size of the lookup tables. We also demonstrated that some heterogeneous mixtures may deliver superior speed with equivalent area to the best homogeneous FPGA.

In the future, we will investigate the speed-area tradeoff further by optimizing both delay and area.

5 Acknowledgements

The authors wish to express their thanks to Bob Francis and Kevin Chung for many helpful discussions.

References

[Ahre90] M. Ahren, A. El Gamal, D. Galbraith, J. Greene, S. Kaptanoglu, K. Djarmarajan, L. Hutchings, S. Ku, P. McGibney, J. McGowan, A. Samie, K. Shaw, N. Stiawalt, T. Whitney, T. Wong, W. Wong, and B. Wu, "An FPGA Family Optimized for High Densities and Reduced Routing Delay," *Proc. 1990 CICC* , May 1990, pp.31.5.1 - 31.5.4

[Algo89] CAL 1024 Datasheet, Algotronix Ltd. Edinburgh, Scotland, 1989

[Bray90] R. Brayton, G. Hachtel, and A. Sangiovanni-Vincentelli, "Multilevel Logic Synthesis," *Proc. IEEE*, Vol.78, No.2, Feb. 1990, pp.264-300.

[Brow92] S. Brown, R. Francis, J. Rose, Z. Vranesic, "Field-Programmable Gate Arrays", *Kluwer Academic Publishers*, June, 1992.

[Cart86] W. Carter, K. Duong, R. Freeman, H. Hsieh, J. Ja, J. Mahoney, L. Ngo, and S. Sze, "A user Programmable Reconfigurable Gate Array," *Proc. 1986 CICC*, May 1986, pp.233-235.

[ElGa89] A. El Gamal, J. Greene, J. Reyneri, E. Rogoyski, K. El-Ayat and A. Mohsen, "An Architecture for Electrically Configurable Gate Arrays," *IEEE J. Solid State Circuits.* Vol. 24, No. 2, April 1989, pp.394-398.

[He93a] J. He and J. Rose, "Why Are an FPGA's Logic Blocks All The Same? - (A Technology Mapping Algorithm for Heterogeneous FPGAs), " Submitted to *DAC93*, 1993.

[He93b] J. He, *M.A.Sc. Thesis in Preparation,* University of Toronto.

[Hill92] D. Hill, B. Britton, B. Oswald, N. Woo, S. Singh, T. Poon, and B. Krambeck, "A New Architecture for High-Performance FPGAs," *2nd international Workshop on Field-Programmable Logic and Applications*, Aug., 1992.

[Hsie90] H. Hsieh, W. Carter, J. Ja, E. Cheung, S. Schreifels, C. Erickson, P. Freidin, L. TinKey, and R. Kanazawa, "Third-Generation Architecture Boosts Speed and Density of Field-Programmable Gate Arrays," *Proc. 1990 CICC*, May 1990, pp.31.2.1-31.2.7

[Koul91] J. Kouloheris and A. El Gamal "FPGA Performance vs. Cell Granularity," *Proc. 1991 CICC*, May 1991, pp. 6.2.1 - 6.2.4.

[Koul92] J. Kouloheris and A. El Gamal, "FPGA Area vesus Cell Granularity - lookup Tables and PLA Cells," *ACM/SIGDA Workshop on FPGAs* (FPGA'92), Feb. 1992, pp.9-14.

[Rose90] J. Rose, R. Francis, D. Lewis and P. Chow, "Architecture of Field-Programmable Gate Arrays: The Effect of Logic Block Functionality on Area Efficiency," *IEEE J. Solid-State Circuits*, Vol.25, No.5, Oct.1990, pp.1217 - 1225.

[Sing91] S. Singh, J. Rose, D. Lewis, K. Chung, and P. Chow, "Optimization of Field- Programmable Gate Array Logic Block Architecture for Speed," *Proc. 1991 CICC*, May 1991, pp.6.1.1- 6.1.6.

[Sing92] S. Singh, J. Rose, P. Chow, and D. Lewis, "The Effect of Logic Block Architecture on FPGA Performance," *IEEE J. Solid State Circuits.* Vol. 27, No. 3, March 1992, pp.281-287.

[Wils92] R. Wilson, "Altera Flexes Programmable Logic Muscles," *Electronic Engineering Times*, Oct.5, 1992.

[Wong89] S. Wong, H. So, J. Ou, and J. Costello, "A 5000-Gate CMOS EPLD with Multiple Logic and Interconnect Arrays," *Proc. 1989 CICC*, May 1989, pp.5.8.1 - 5.8.4.