HYBRID FPGA ARCHITECTURE

Alireza Kaviani and Stephen Brown
Department of Electrical and Computer Engineering
University of Toronto, Canada
Email: kaviani|brown@eecg.toronto.edu

Abstract

This paper[1] proposes a new field-programmable architecture that is a combination of two existing technologies: Field Programmable Gate Arrays (FPGAs) based on LookUp Tables (LUTs), and Complex Programmable Logic Devices based on PALs/PLAs. The methodology used for development of the new architecture, called *Hybrid* FPGA, is based on analysis of a large set of benchmark circuits, in which we determine what types of logic resources best match the needs of the circuits. The proposed Hybrid FPGA is evaluated by manually technology mapping a set of circuits into the new architecture and estimating the total chip area needed for each circuit, compared to the area that would be required if only LUTs were available. Preliminary results indicate that compared to LUT-based FPGAs the Hybrid offers savings of more than a factor of two in terms of chip area.

1 Introduction

Over the past several years, high capacity Field Programmable Devices (FPDs) have enjoyed a rapidly expanding market, and have become widely accepted for implementation of small to moderately large digital circuits. The two main types of FPDs, Field Programmable Gate Arrays (FPGAs) and Complex Programmable Logic Devices (CPLDs) are both widely used, and each offers specific strengths. FPGAs that are programmed with SRAM technology are usually based on LookUp Tables (LUTs); their main strengths are very high total logic capacity, in the range of tens of thousands of equivalent logic gates, and good speed-performance of 10 to 50 MHz system clock rates. On the other hand CPLDs consist of multiple PLA-based blocks, in which the OR planes are partly fixed. Their characteristics include medium capacity, in the range of a few thousand gates, and ultra high speed-performance, sometimes in excess of 100 MHz system clock rate.

In this paper, we suggest a new type of FPD that represents a marriage of FPGAs and CPLDs. The basis for this idea is that digital circuits are structured in such a way that parts of the circuit are well-suited for implementation using LUTs, while other parts can benefit more from the Product term-based (Pterm-based) structures found in CPLDs. Comparison with an architecture that has only LUTs indicates that the *Hybrid FPGA Architecture (HFA)* offers significant savings in terms of the total area. Also, the HFA creates the potential to reduce the depth of the circuit implemented in the FPGA, which may provide improvements in speed-performance.

This paper is organized as follows: Section 2 discusses related research on architecture of FPDs, Section 3 describes our research motivation, which is based on the analysis of BenchMark (BM) circuits, Section 4 presents the invented architecture, Section 5 gives an estimate of area gain provided by the HFA, and the last section contains final remarks.

2 Related Work

FPDs suffer from lower speed-performance and less logic capacity in comparison to custom-manufactured technologies, such as mask-programmed gate arrays. However much recent research has been devoted to improving FPD architecture. New applications continue to emerge as research in industry and academia results in more sophisticated products with higher total logic capacity and better speed-performance. Highlights of some recent research efforts on FPGA logic blocks is presented below.

The earliest research study that was reported on FPGA architecture focuses on complexity of the logic blocks [RFLC90]. The paper assumes an FPGA architecture based on LUTs, and varies the number of inputs to a LUT to measure the effects on implementation of a set of benchmark circuits. The basic conclusion reached is that LUTs with four or five inputs yield the best results in terms of chip area. We apply this result to our Hybrid FPGA, by using 4-input LUTs (4-LUTs). 4-LUTs are also found in commercial FPGAs, such as the Altera FLEX 8000 and Xilinx XC5000.

Most research on FPDs has focused on FPGAs, and little work has been published on CPLDs. However, the study in [KE92] investigated FPDs built using PLA-based logic blocks. According to [KE92], an FPD based on PLAs with 10 inputs, 12 Pterms, and 3 outputs achieves about the same level of logic density as FPGAs based on 4-LUTs. However, we are not aware of any commercial product that is based on such PLAs. [KE92] also introduced a model for estimating the chip area needed for a PLA-based logic block, and we utilize this later in our paper when discussing chip area needed for the HFA.

A recent study, called Heterogenous FPGAs [HR93], investigated FPGA architectures with logic blocks of two different sizes. The paper reports the effects on area efficiency of LUT-based FPGAs, but with two sizes of LUTs in the same chip. A summary of the results is that on average a

[1] This work was funded by Hewlett Packard.

mixture of LUTs provides a savings of about 15% in area. The HFA presented in this paper is related to the Heterogenous FPGA, in the sense that two different logic blocks are available. However, the two approaches are quite different because the Heterogenous FPGA has two sizes of the same type of logic block (LUTs), while the HFA has two entirely different types of logic resources (LUTs and PLA-based blocks).

Another idea investigated in recent FPGA research examines memory modules with variable aspect-ratio [WRV95] that could be included as separate blocks in an FPGA. This idea is not orthogonal to the Hybrid FPGA, and so memory blocks could also be included in our architecture.

[SP95] suggests a logic block built from an array of Content Addressable Memory (CAM) cells, as opposed to LUT-based or PLA-based blocks. The CAM cells can be used in RAM-mode, in which case the logic block functions in the same way as a LUT. Also, multiple CAM cells can be combined to implement the equivalent functionality of a PLA. No comparison in terms of chip area or speed-performance is given in [SP95] between the CAM-based approach and a traditional FPGA or CPLD, and so we cannot comment on the relative merits of this idea in comparison to the HFA.

3 Combinational Nodes in Benchmark Circuits

Any digital circuit can be represented as a Directed Acyclic Graph (DAG) that consists of combinational and sequential nodes. Each combinational node of the circuit can be represented in sum of products form. As the first step toward defining the HFA, we wish to examine the combinational nodes present in real circuit, and produce a distribution of nodes with respect to size. We define the *size* of a node according to two parameters: 1) the number of inputs to the node, and 2) the number of Pterms in the sum of products representation of that node. The circuits used in this paper are from 1993 MCNC logic synthesis BM suite. Figure 1 shows the distribution of node sizes for all combinational nodes in 190 MCNC BMs after technology independent optimization[1]. In total, the BMs comprise 36304 combinational nodes. As shown in Figure 1 it is apparent that the majority of nodes are small. A closer examination reveals that more than 70% are 4-bounded and roughly 20% of the nodes have fanin equal or greater than 6; these latter nodes will be referred as *high fanin* nodes in this paper.

[1] The optimization procedure is discussed in Section 5.

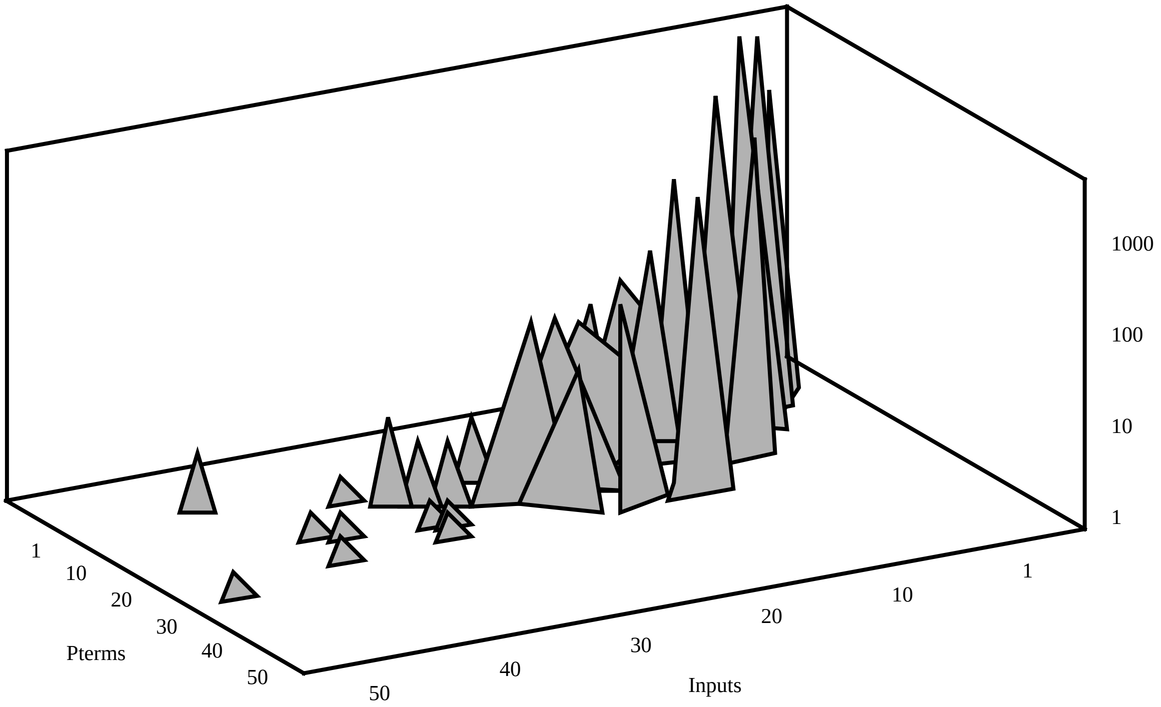


Figure 1 - *Node size distribution (approximate figurefor CDROM — see hardcopy for real data).*

We wish to consider implementation of the nodes in Figure 1 in two types of logic resources: PLA-based cells, and LUTs. For a LUT with K inputs, the area of the cell is proportional to 2^K. For a PLA-based cell, the cell area is approximately proportional to K^2. Note that for $K=4$, $2^K = K^2$, but for $K < 4$ LUTs are more efficient than PLAs. Therefore 4-bounded nodes can be efficiently implemented using LUTs. This accounts for the majority of the nodes in circuits, but there is still a significant number of nodes with high fanin. These nodes could also be implemented using 4-LUTs, but the area required would be large. From Figure 1, we can observe that most high fanin nodes do not require a large number of Pterms. This implies that these nodes are well suited for implementation in PLAs.

The concept of suitability of nodes of different sizes in either LUTs or PLAs is illustrated in Figure 2, which shows the distribution of all combinational nodes in the logic plane. Each dot in the figure represents combinational nodes of a specific size, but the number of nodes of each size is not shown. In Figure 2, the nodes that lie in the lightly shaded rectangle efficiently fit into 4-LUTs. Similarly, PLAs are more attractive for implementing the nodes that lie in the heavily shaded box. Nodes that do not lie in either of these areas could be implemented using either LUTs or PLAs, but the cost would be relatively high. If both PLAs and LUTs are available in an FPGA architecture, a boot-like area of the logic plane is naturally supported, as indicated by the bold curve in Figure 2. By covering a wider area of the logic plane we can decrease the overall cost of the implementation of most of the circuits. Therefore the new HFA contains both PLA-based blocks, which we call *Programmable Array Logic Blocks (PALBs)* and 4-LUTs.

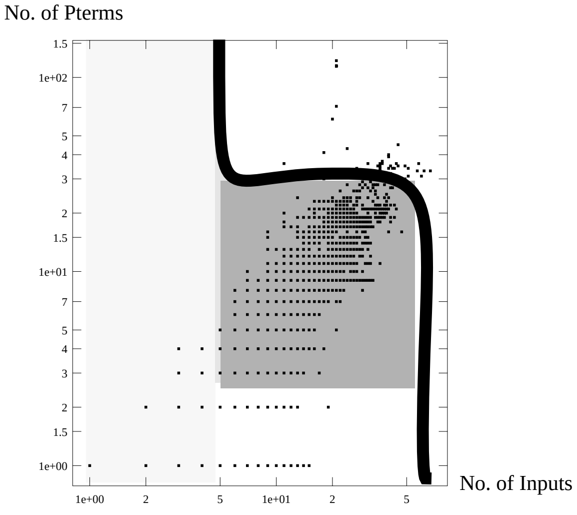


Figure 2 - *Logic plane.*

4 Hybrid 4-LUT+ PALB Architecture

Previous research [RCLF90, KE91] has studied the effects of the size of LUTs on the area efficiency in FPGAs and concluded that 4-LUTs provide good results. In this paper, we assume that the HFA uses 4-LUTs. This section explains the analysis that led to determination of the number of inputs, the number of Pterms and the number of outputs for the PALB.

4.1 Analysis of Node Sizes in Benchmark Circuits

Since 4-bounded nodes will be implemented in 4-LUTs, the PALB should be designed in such a way that it is well-matched for implementation of nodes with more than four inputs. Also, we observed that many 5-input nodes are simple ORs or ANDs; these nodes can easily be decomposed and realized in 4-LUTs. It is not desirable that the nodes that are to be implemented in LUTs affect the PALB architecture. Figure 3 shows the node size distribution, in terms of # of Pterms and # of inputs, for all MCNC BMs, excluding 5-bounded nodes. With reference to the figure, there is a peak at nine Pterms, with some larger nodes and many smaller ones. To observe the effects of 5-input nodes, Figure 4 shows the same information, except that only 4-bounded nodes are excluded. Now, there is a new peak at 3 Pterms; it is clear that the number of 5-input nodes is significant and may strongly affect our analysis. Because the effects of 5-input nodes is pronounced and since many of these nodes are suitable for LUTs, it is important to exclude them when designing the PALB architecture. Similar statements can also be made for nodes with higher fanin, but the effects would be less important because there are fewer of these high fanin nodes. We decided to exclude only obviously decomposable 5-input and high fanin nodes (which are

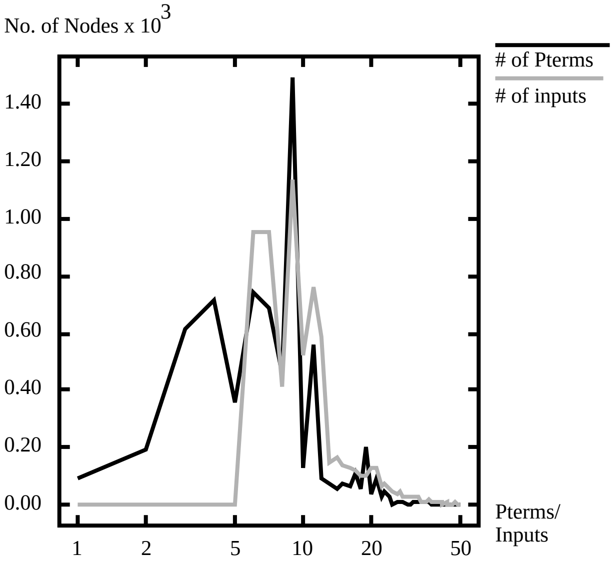


Figure 3 - *Node size distribution (excluding 5-bounded nodes).*

Excluding K-bounded nodes	Inputs (μ, σ)	Pterms (μ, σ)	Inputs /Pterms (μ, σ)	Pterms (filtered) (μ, σ)	Inputs / Pterms (filtered) (μ, σ)
K=6	(12.5, 6.56)	(9.32, 7.29)	(1.55, 1.02)	(6.64, 7.48)	(1.81, 1.32)
K=5	(11.57,6.48)	(8.47, 7.08)	(1.61, 1.01)	(6.25, 7)	(1.77, 1.21)
K=4	(10.04,6.32)	(7.23, 6.62)	(1.68, 1.03)	(5.53, 6.3)	(1.7, 1.07)
K=3	(8.59, 6.09)	(6.02, 6.17)	(1.78, 1.01)	(4.89, 5.62)	(1.62, 0.94)
K=2	(6.35, 5.45)	(4.36, 5.21)	(1.83, 0.94)	(3.77, 4.58)	(1.5, 0.76)
K=1	(4.54, 4.69)	(3.12, 4.26)	(1.74, 0.79)	(2.96, 3.65)	(1.29, 0.66)
K=0	(4.46, 4.66)	(3.07, 4.22)	(1.72, 0.79)	(2.91, 3.62)	(1.29, 0.66)

Table 1 - Architectural statistics.

nodes with only one Pterm), as explained shortly. A summary of statistics of node sizes excluding nodes with various numbers of inputs is presented in Table 1.

There are three statistical parameters that affect the PALB: 1) the average number of inputs, 2) the average number of Pterms and 3) the ratio of the number of inputs to the number of Pterms for each combinational node. Table 1 gives the mean and variance, (μ, σ) , of these parameters. Each row of the table corresponds to a specific value of *K* and shows the statistical data excluding *K*-bounded nodes. Also, under the columns denoted “filtered” additional nodes are excluded, according to the following assumptions: 1) all single-input Pterms in a node are merged into one multi-input Pterm. This is based on our observation that in the sum of products form of high fanin nodes there are many Pterms with only one input. As will be explained in the next subsection, these single-input Pterms can be merged into one, with almost no extra cost for the PALB architecture, 2) Nodes that are single Pterms (i.e. ANDs, ORs) are excluded. There are many combinational nodes with only one Pterm. Since these nodes are decomposable, LUTs are as good as a PALB for their implementation, so they should be excluded from the data that affect the PALB architecture.

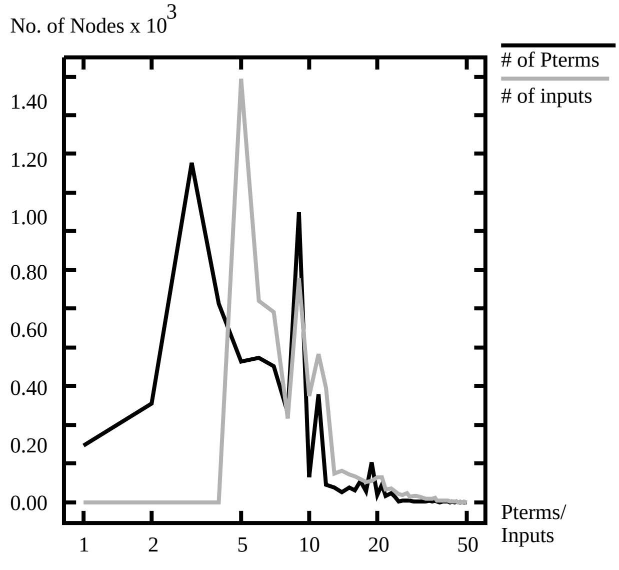


Figure 4 - *Node size distribution (excluding 4-bounded. nodes).*

Table 1 serves as a guide for designing the PALB architecture. Since we are using 4-LUTs it is reasonable to base the PALB on the *K = 4* row of the table, but for reasons discussed earlier, it is more appropriate to use the filtered columns. Thus the PALB should be designed to suit the parameters shown in the shaded boxes. We decided that 5 for the number of Pterms and 1.6 for the ratio of the number of inputs to the number of Pterms are the closest practical values to the averages shown in the shaded cells. In the next subsection we introduce an appropriate PALB architecture using these calculated values.

4.2 PALB Architecture

Figure 5 shows the PALB that we developed for use in the HFA. It has 16 inputs, 10 Pterms and 3 outputs. On average, 2 combinational nodes with 5 Pterms in each can be implemented in a PALB. There is one extra output to accommodate the implementation of small nodes. There are 2 flip-flops and one of them can accept asynchronous clock as well as the global clock. Real gates can be used to implement the NAND/AND/OR plane of the PALB as opposed to wired gates. This allows us to have many PALBs in one chip without concern for static power consumption. Therefore we designed the PALB in a NAND-NAND fashion. The meaning of the schematic in Figure 5 should be readily apparent, except for the connection shown for the inputs of the second NAND plane. Five Pterms are shown hardwired to produce the lowest of the three outputs (drawn as “•”). Also, another 5 Pterms can be programmably connected to this output as well (drawn as “**x**”). This architecture represents a combination of a classic fixed versus programmable OR-plane and is designed to be optimized for five Pterms and yet be configurable for up to 10 Pterms. Similar comments apply to the other PALB outputs.

A typical combinational node and different forms of its implementation are depicted in Figure 6. It is easy to see that all variants *a), b)* and *c)* are functionally equivalent. Figure 6c shows how Pterms with single inputs can be merged into one. Note that the extra NAND gate shown and inverters at its inputs are already available in our PALB, with no additional cost. In this example the number of Pterms is reduced

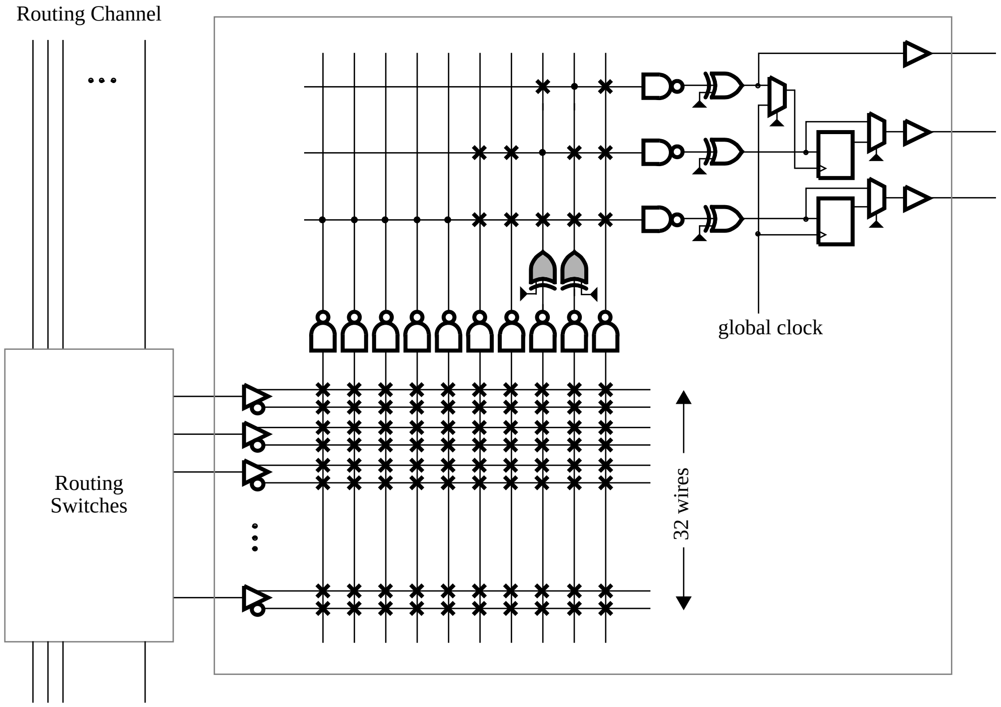
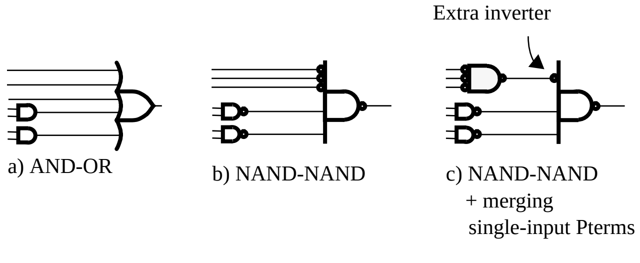


Figure 5 - *The PALB architecture.*

from 5 to 3 with the cost of one extra inverter. The shaded XORs in Figure 5 serve as programmable inverters for the purpose of merging single-input Pterms into one multi-variable Pterm. This feature of the PALB is motivated by the filtered columns in Table 1, and its effects on the distribution of node sizes is shown in Figure 7, which provides the distribution of the number of Pterms after merging. The Pterm distribution in Figure 4 is repeated in Figure 7 to facilitate a comparison between the two. The simple idea of merging decreases the number of Pterms in the observed 190 MCNC BMs approximately 25% on average. This is clear from Figure 8, which shows the average number of Pterms with and without merging. This feature of the PALB thus reduces the logic resources needed to implement circuits, leading to better area efficiency.



a) AND-OR

b) NAND-NAND

c) NAND-NAND
+ merging
single-input Pterms

Figure 6 - *Several forms of implementation of a node.*

5 Estimate of the Area Gain

An accurate measure of the gain provided by the new architecture requires CAD tools for performing technology mapping, routing and placement of the circuits. These tools are not yet available, and so we provide an approximation of the expected gain of the new architecture compared to a 4-LUT-based FPGA. The results of the comparison for 10

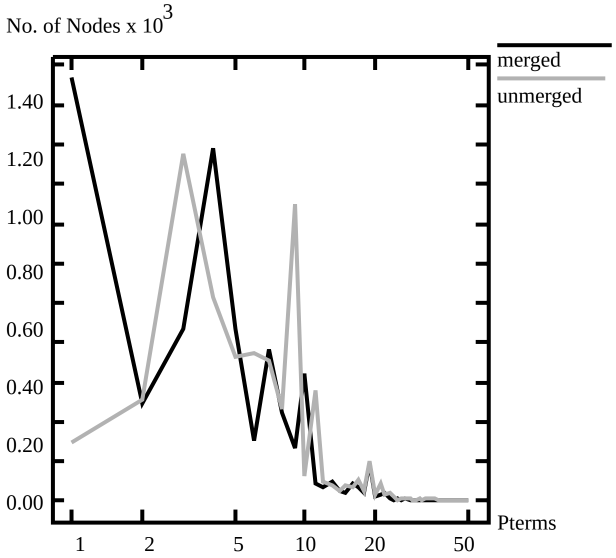


Figure 7 - *Distribution of the no. of Pterms (4-bounded excluded + merged).*

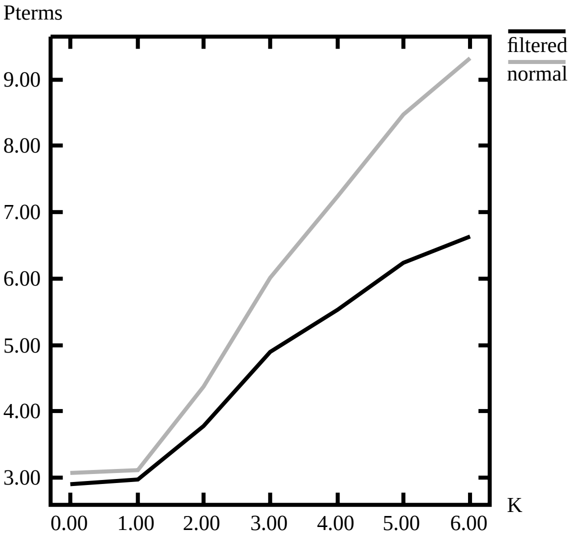


Figure 8 - *Average no. of Pterms.*

MCNC BMs are presented in Table 2. The MCNC BMs in EDIF format were converted to BLIF [S92] and after optimization were mapped to 4-LUTs using *Flowmap* [CD94]. All MCNC BMs were passed through one run of the standard technology independent optimization *script.rugged* before mapping. The script is provided by SIS [S92] and uses various methods of optimizing combinational circuits.

The numbers in the “4-LUT (area)” column of the table are the 4-LUT counts after technology mapping using Flowmap. To estimate the relative area efficiency of the HFA after technology independent optimization, the BMs were technology mapped manually to the new architecture. The resulting numbers of PALBs and 4-LUTs for each circuit are shown in the table. The total chip area in the HFA is estimated in the column labeled “HFA (area)”, in terms of equivalent 4-LUT count assuming each PALB takes area equal to four 4-LUTs. This assumption is supported by the previous research reported in [KE92] and [KE91], which provides estimates for area of both PLA-based cells and LUTs. According to the area models in these papers, if the area of an SRAM cell is about $100\ \mu\text{m}^2$, which is reasonable assuming a 0.5 micron technology, then the area of a PLA-based cell with 16 inputs, 10 Pterms, and 3 outputs is about the same as the area for four 4-LUTs (the four 4-LUTs are considered as one “block” with local interconnect). It is also important to mention that total chip area in an FPD is strongly influenced by the routing resources. We assume that a PALB requires about the same amount of routing area as four 4-LUTs, since both of these logic resources have 16 inputs and the number of outputs (3,4) is similar.

The area gain is calculated as the ratio of the area in the 4-LUT based architecture to the area of the HFA for the same circuit. Overall gain can be obtained either by taking the average of the gains for individual BMs, or by calculating the gain of the meta circuit that consists of all 10 BMs. Although the gain is strongly affected by the type of the circuit, for the BMs in Table 2 the gain increases with the size of the circuit on average. Therefore we conjecture that higher gains will be obtained for larger circuits, but this is difficult to verify without automatic CAD tools for mapping of large circuits.

Table 1 - Estimate of the area gain.

BM	4-LUT (area)	HFA (area)	PALBs	4-LUTs	Area gain
5xp1	78	28	4	12	2.8
9sym	172	69	11	25	2.5
count	55	39	8	7	1.4
C499	74	74	0	74	1
9symml	159	58	9	22	2.7
misex1	21	16	3	4	1.3
s298	1970	271	62	23	7.3
z4ml	18	10	1	6	1.8
vg2	69	32	4	16	2.2
alu2	213	105	14	49	2
Average					2.5
Total:	2829	702	116	238	4

Note that the results in Table 2 imply various mixtures of LUTs and PALBs. Our assumption is that enough resources of each type would be available in a real chip. The mixture of the PALBs and 4-LUTs is a function of the type and the size of the circuit. However, we believe that this issue is not critical for two reasons: 1) LUTs and PALBs are interchangeable resources; LUTs can be used for implementation of high fanin nodes if there are not enough PALBs, and PALBs can implement small nodes if too few LUTs are available, and 2) a commercial HFA family would comprise various chips with different numbers of PALBs and LUTs. It will be the responsibility of the HFA technology mapper to make the best usage of the available resources. As explained earlier in the paper, roughly about 10%-20% of the combinational nodes in MCNC BMs are 5-input or obviously decomposable high fanin nodes. These nodes can be implemented in either 4-LUTs or PALBs without any significant loss or gain in terms of area. Therefore the target for these nodes might be determined by a technology mapper toward the goal of maintaining the desired balance between the number of 4-LUTs and the PALBs. Preliminary investigations indicate that allocating 50% of the area of the chip to 4-LUTs and the rest to PALBs (which implies that the number of 4-LUTs would be 4 times the number of PALBs) is reasonable for large BMs.

The manual technology mapping that was done for the HFA was pessimistic. For example one 4-LUT was assigned to each 4-bounded node for the sake of simplicity, even though in some cases a few of the 4-bounded nodes might be merged into one 4-LUT. Therefore a good technology mapper that targets the HFA might slightly increase the gain. Also, for mapping to 4-LUTs, the comparisons provided here are based on Flowmap, because of its convenient availability as part of SIS. Since Flowmap is designed to minimize the depth of the circuit, other algorithms [FS94] whose primary objective is optimizing for area may produce slightly lower numbers of 4-LUTs. This would reduce the gain shown in Table 2 but we may still expect an improvement of more than a factor of 2 in terms of the area efficiency.

6 Final Remarks and Future Work

The area gain shown in the previous section is enough to justify the merit of the HFA; however, there are some additional benefits that are worth mentioning. These benefits have not been evaluated yet, but we believe that the invented architecture has the potential to be exploited toward them. Commercial FPDs are considerably slower than mask programmable gate arrays, mostly due to delays associated with programmable switches. Therefore it is desirable to reduce the depth of the critical path in the circuits when implemented in FPDs. Allowing high fanin nodes in the circuit reduces the depth and thus may increase overall speed-performance. Also, this decreases the total number of nodes in the circuit, which may simplify the tasks of placement and routing. The PALBs in the HFA provide these advantages by efficiently realizing high fanin nodes.

The next major step in this research is to implement an appropriate technology mapper for the new architecture. Also, the depth reduction mentioned above should be investigated. Although we believe that the HFA will not complicate routing issues, and determination of an appropriate balance for the number of PALBs versus 4-LUTs is not difficult, in-depth investigations of these issues are also part of our future plans. As a final comment, it is likely that the HFA can be enhanced in several ways, and we intend to continue improving upon our suggested architecture.

Acknowledgments

We are grateful to Hewlett Packard for funding our research, and would like to extend special thanks to Kenneth Rush for initiating contact with our group. We would also like to thank Michael D. Hutton for providing us with optimized MCNC benchmark circuits.

References

[CD94] J. Cong and Y. Ding, “FlowMap: An Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs,” IEEE trans. on CAD of integrated circuits and systems, January 1994.

[FS94] A.H. Farrahi and M. Sarrafzadeh, “Complexity of the Lookup-Table Minimization Problem for FPGA Technology Mapping,” IEEE trans. on computer-aided design of integrated circuits and systems, Nov. 1994.

[HR93] J. He and J. Rose, “Advantages of Heterogeneous Logic Block Architectures for FPGAs,” Proceedings of 1993 Custom Integrated Circuits Conference, San Diego, May 1993 pp. 7.4.1 - 7.4.5

[KE91] J. L. Kouloheris and A. El Gamal, “FPGA Performance vs. Cell Granularity,” Proceedings of the 1991 Custom Integrated Circuits Conference.

[KE92] J. L. Kouloheris and A. El Gamal, “PLA-based FPGA Area vs. Cell Granularity,” Proceedings of the 1992 Custom Integrated Circuits Conference.

[RFLC90] J. Rose, R.J. Francis, D. Lewis, and P. Chow, “Architecture of Field-Programmable Gate Arrays: The Effect of Logic Block Functionality on Area Efficiency,” IEEE JSSC, Vol. 25 No. 5, October 1990, pp. 1217-1225.

[S92] E. M. Sentovich et al., “SIS: A System for Sequential Circuit Synthesis,” Electronics Research Laboratory, Memorandum No. UCB/ERL M92/41.

[SP95] A. Stansfield and I. Page, “The Design of a New FPGA Architecture,” 5th International Workshop on Field Programmable Logic and Applications, FPL’95, University of Oxford, Aug 1995.

[WRV95] S. Wilton, J. Rose, Z. Vranesic, “Architecture of Centralized Field-Configurable Memory,” 3rd ACM International Symposium on Field-Programmable Gate Arrays, FPGA ‘95, Monterey Bay, CA, Feb 1995.