EFFICIENCY OF STANDARD CELL LIBRARIES COMPOSITION

[1]Pedro E. M. Paganela, [1]Eduardo A. A. Canabarro, [2]André I. Reis, [1]Renato P. Ribas

[1]Instituto de Informática – UFRGS, Porto Alegre, Brazil
[2]Nangate Inc., Sunnyvale, CA, USA
{pempaganela, eaacanabarro, rpribas}@inf.ufrgs.br, are@nangate.com

ABSTRACT

This paper presents a methodology to analyze the composition of optimal standard cell libraries, for a given synthesis tool in a given mapping focus, in this case the circuit area optimization. Some methods are presented to find this ideal library and their results are shown to validate the purpose.

1. INTRODUCTION

In VLSI digital circuits, the synthesis from a logical description to the physical implementation is usually done by software, through synthesizing tools. These tools can make the mapping process using a library, where the cells are entirely described, to get the cells from (called library-based), or creating by itself the appropriated functions (called library-free) [8]. However, there are some questions related that should be taken into account, like which cells (logic functions and drive strengths) should be described in a specific library, how many cells should be enough to provide a good performance and the existence of a perfect library for a given synthesis tool.

2. MOTIVATION AND PURPOSE

When developing circuits to be used in mobile equipments, like a cell phone, the goal is low power consumption. In other applications, the main goal could be the highest operation frequency or the smallest area occupied. The cells that might be used are going to be described in a library if a library-based tool is being used. Thus, a specific library is considered efficient if it allows satisfying results for the mapping processes of a given set of circuits, with a common purpose. There are numerous methods to develop a library with high quality. For instance, through the usage of big cells [1], through the load of the transistors used for PMOS and NMOS [2], through the averiguation over the width ratio of the transistors [3], through some algorithms to perform the library development [4], and so on. Nevertheless, the quality of a given library cannot only be measured by the total amount of cells described by this methods and analysis.

The equilibrium between its efficiency for the users and its development cost is highly required. It is worthless to have a huge number of cells in a library if the synthesis tool does not use, at least, a high percentage of the set. The main purpose of this work is the investigation over the methods that could be used to develop an ideal library composition for two commercial CAD environments, denominated here CAD1 and CAD2.

3. METHODOLOGY

To make the experiments required for the investigation, the library used was GenLib 44-6 [9] with all possible unate functions up to six inputs, entirely characterized by Nangate Library Creator [10], a commercial software developed for this purpose. Extra widely used cells, not included in the GenLib description, such as adders, exclusive-OR and multiplexers were also added to the set. In this library, all the functions were characterized for three different sizes or drive strengths, each one doubling the load capacity of the previous one (X1, X2, X4, where X4 has four times the load capacity of X1). The circuits used were benchmarks provided by IWLS [5].

Three different exercises were made, focusing on the minimum circuit area.

→ Initially, a free mapping process (without restrictions) was performed for the entire set, in order to analyze the cell usage distribution (i.e. the number of instances).
→ After this, trying to check the importance of the least used cells, mapping processes with cell restrictions were made (the restriction was set to avoid the usage of the least used cells).
→ Eventually, the same concept of cell restriction was applied, but restricting the most used cells, to verify the impact of their utilization.

Thus, the general purpose of the work was to investigate the influence of a given library over the mapping processes realized with it.

4. INTRODUCING THE ANALYSIS

4.1. Cells used to built the library

When mapping a circuit in a given synthesis tool, it might be concluded that the best library is the one that contains the biggest number of available cells. To measure the quality of the library, the relation between development costs and efficiency must be analyzed.

Fig.1 shows the chart of the first hundred functions most used in the circuits mapped with CAD1. Y-axis represents the number of references for each function. The circuits used were provided by IWLS [5], five of them from a group called OPENCORES and three of them from another group called ITC99.

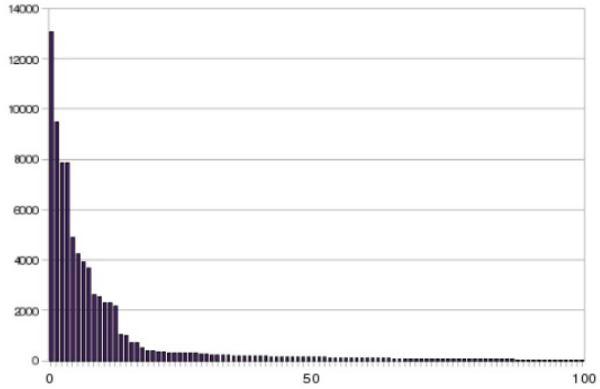


Figure 1 – Distribution of cells without restriction: different cells (X-axis) versus number of instances (Y-axis).

Looking to the chart, the high usage concentration in a small set of cells is clearly observed. It allows the conclusion that there is a huge number of "not necessary" cells.

The behavior of the distribution for CAD2 was really close to the one for CAD1 and the functions used were very similar too, as observed in Fig. 2. Thus, from now, some results will be showed only considering data from CAD1, once their behavior is similar. Interpretations over the data from CAD1 will be considered the same for CAD2.

Also, more than the similarity of the distribution in a small set of cells, all the circuits had a similar distribution of usage, as shown in Fig. 3.

For a better visualization of the chart, the least used cells and the inverter cell were omitted, because they had a different scale and the same behavior. The Y-axis represents the percentage of a given cell in a given circuit. As shown, the behavior is almost the same for the entire group of circuits.

This analysis allows the conclusion that there will be a concentration of usage in a small set of cells contained in a library, not only for the general behavior of the circuits group, but for each circuit itself, if the synthesis tool is CAD1 or CAD2.

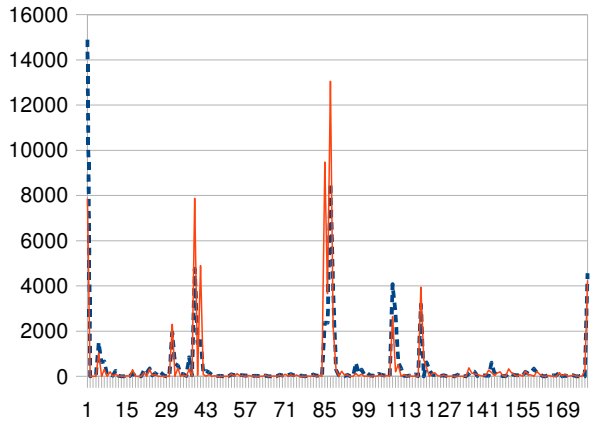


Figure 2 – Comparison between CAD1 and CAD2: different cells (X-axis) versus number of instances (Y-axis).

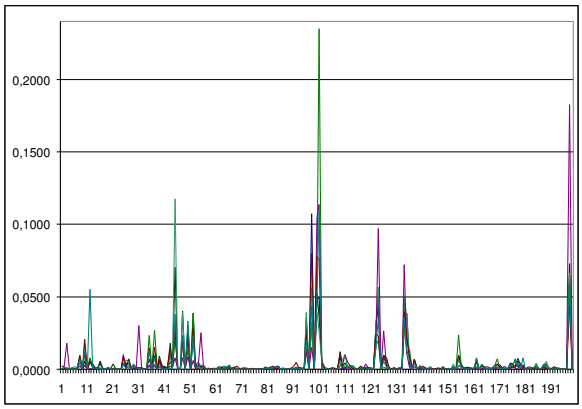
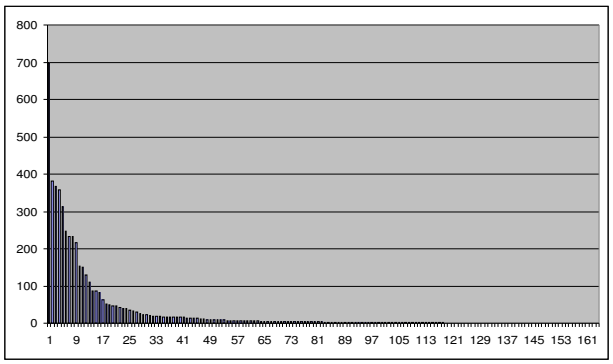


Figure 3 – Distribution of used cells for each circuit: different cells (X-axis) versus percentage of instances (Y-axis).

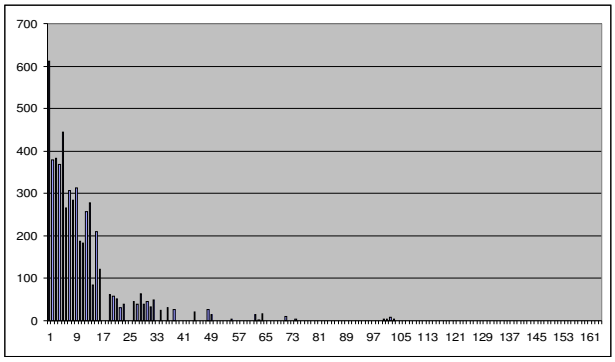
4.2. Restricting the least used cells

With a big amount of cells described in a library, it is concluded that both CAD1 and CAD2 used just a small set of the entire group, as proved above. So, restricting the least used functions would make possible the analysis of their relevance to the mapping process. Through fifteen rounds, the circuits were mapped with a decrease of the 10 least used functions, taken from a previously built list (the list had data from the exercise above and was built considering the average usage rate for the entire group of circuits). The circuits were mapped, for each round, using the same original circuit description (verilog file), not using the mapped verilog from the previous round.

Fig. 4 shows the distribution for the first and last rounds of the circuit "usb_func" (OPENCORES group) [5].



(a)



(b)

Figure 4 – Distribution of used cells to "wb_conmax": different cells (X-axis) versus number of instances (Y-axis).
(a) round 1 - without restriction;
(b) round 15 - last round with less used cells restriction.

It is important to mention that the functions removed for each round were the 10 least used, taken progressively from a previously built list.

Fig. 4 charts shows that, in order to replace the forbidden functions, both CAD1 and CAD2, instead of looking for functions that have not been used before, replaced them synthesizing the functions with others already used, creating a higher concentration level. The total amount of used cells stayed the same, as well as the circuit area (in square microns) . This is shown through Fig. 5 and Fig. 6.

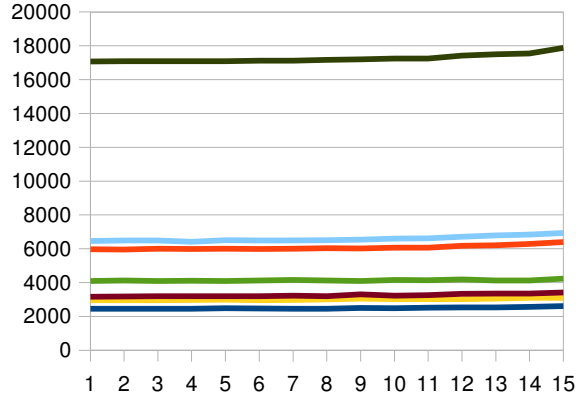


Figure 5 – Number of instances for each round: number of rounds (X-axis) versus number of instances (Y-axis).

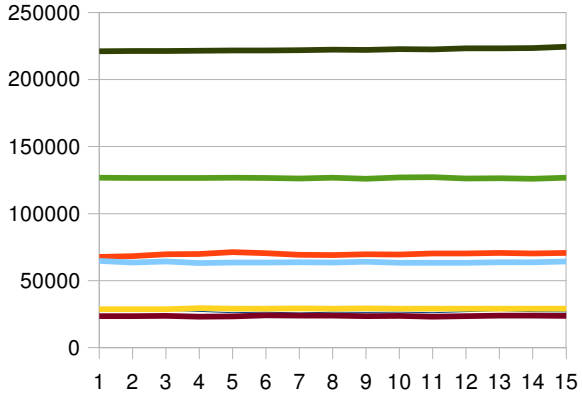


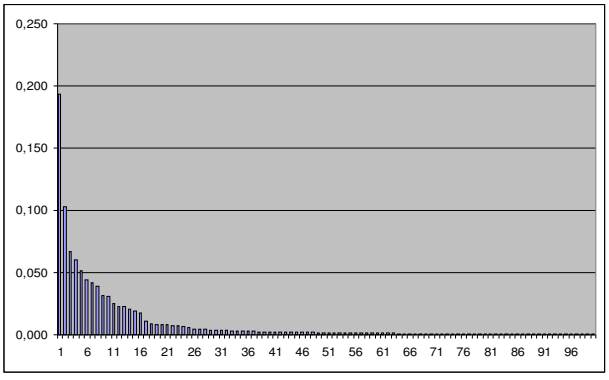
Figure 6 – Circuit area for each round: number of rounds (X-axis) versus circuit area (in square microns) (Y-axis).

Data from circuit "b18" was taken away, in order to preserve the scale of the chart.

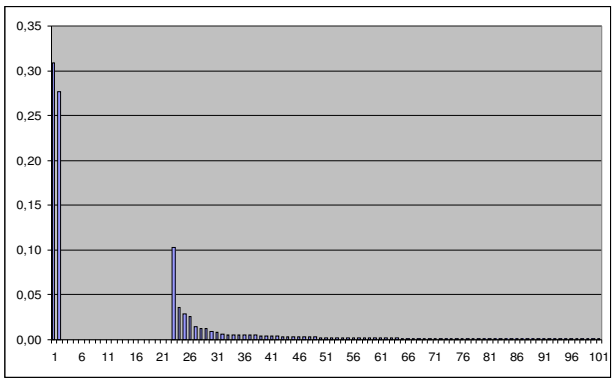
Through this exercise, the overall conclusion is that the least used cells do not affect the efficiency of the library. The final library, with a considerable smaller set of cells, showed almost the same efficiency of the complete library.

4.3. Restricting the most used cells

The data shows that there is a group of cells always chosen by the tools, which represents the group of the most used functions. However, if these "most used functions" disappear, how would the software perform the mapping process? Would it search for functions not used before? In order to analyze this situation, a restriction over the most used functions was applied. Through 5 rounds, restricting 4 functions per round, the mapping process was performed. In Fig. 7, the charts showing the usage distributions in the first and last rounds (Y-axis represents the percentage of each function for the entire group of circuits):



(a)



(b)

Figure 7 – Distribution of Used Cells in all circuits: different cells (X-axis) versus percentage of instances (Y-axis).
(a) round 1 - without restriction;
(b) round 5 - last round with most used cells restriction.

When restricting the most used functions, the inverter and NAND2 could not be restricted, because one of the tools need the NAND2 to make the mapping and the inverter should be preserverd to avoid a substantial effect over the distribution. Instead of using functions not used before, the tools began to concentrate more and more on the small group, creating an even bigger usage concentration, confirming that there will not be a harmonic usage distribution.

5. CONCLUSION

The main goal of this work was to demonstrate that an optimal library for a given synthesis tool does not require a huge amount of logic functions to optimize the circuit performance. A small set of well chosen cells is enough, as proven through the exercises described above, and already indicated by other authors [6][7]. However, to find the best set of cells, a huge number of cells must be previously characterized. Moreover, the technology mapping engine plays an important role in this evaluation. Novel mapping algorithms could provide a better distribution of different logic functions, if some improvement is demonstrated. Evaluation of timing and power constraints are on going

6. REFERENCE

[1] J. Seo, I. Markov, D. Sylvester and D. Blaauw, “On the decreasing Significance of Large Standard Cells in Technology Mapping”, *IWLS*, Lake Tahoe, CA, USA, 2008.

[2] C. Fisher, R. Blankenship, J. Jensen, T. Rossman and K. Svilich, "Optimization of standard cell libraries for low power, high speed, or minimal area designs," *Custom Integrated Circuits Conference, Proceedings of the IEEE 1996*, pp.493-496, 5-8 May 1996.

[3] D. S. Kung and R. Puri, "Optimal P/N Width Ratio Selection for Standard Cell Libraries," *iccad*, p. 178, 1999 International Conference on Computer-Aided Design (ICCAD '99), 1999.

[4] A. N. Lokanathan, J. B. Brockman and J. E. Renaud, "A Methodology for Concurrent Fabrication Process/Cell Library Optimization," *dac*, pp. 825-830, 33rd Annual Conference on Design Automation (DAC'96), 1996.

[5] Benchmarks ITC99 (b14, b18, b20) and OPENCORES (mem_ctr, systemcaes, tv80, usb_funct, wb_conmax) from IWLS, www.iwls.org, 2005.

[6] Carl Sechen, Miodrag Vujkovic, "Optimized power-delay curve generation for standard cell ICs," *iccad*, pp. 387-394, 2002 International Conference on Computer-Aided Design (ICCAD '02), 2002.

[7] N. M. Duc and T. Sakurai, "Compact yet high-performance (CyHP) library for short time-to-market with new technologies," *Design Automation Conference, 2000. Proceedings of the ASP-DAC 2000. Asia and South Pacific*, pp.475-480, 2000.

[8] F. Marques, L. Rosa, R. P. Ribas, S. Sapatnekar and A. Reis, “DAG Based Library-Free Technology Mapping”, *Great Lakes Symposium on VLSI 2007 (GLSVLSI’07)*, pp. 293 – 298, 2007.

[9] E. M. Sentovich, K. J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. R. Stephan, R. K. Brayton and A. Sangiovanni-Vincentelli, “SIS: A system for sequential circuit synthesis”, *Technical Report No. UCB/ERL M92/41*, EECS Department, University of California, Berkeley, 1992.

[10] Nangate Library Creator™, http://www.nangate.com, June 2008.