CELL LIBRARY VERIFICATION METHODOLOGY

Simone Bavaresco, Caio G. P. Alegretti, Renato P. Ribas, André I. Reis

Nangate Research Lab – Instituto de Informática, UFRGS
Av. Bento Gonçalves, 9500 – CEP 91501-970, Porto Alegre, Brazil
{simoneb, cgpalegretti, rpribas, andreis}@inf.ufrgs.br

ABSTRACT

This paper introduces a new methodology for verifying standard cell libraries. The method creates a circuit with two main goals: first, it forces the use of at least one instance of each cell; and second, it guarantees that all input combinations will be exercised in at least one instance of a library cell. The resulting circuit is automatically constructed and mapped to ensure the verification of individual cell instances.

1. INTRODUCTION

The project of current integrated circuits (ICs) is so complex that the usage of CAD tools is mandatory. Logic synthesis – circuit description via logic models of components and functional blocks – is one of the stages of such project [1].

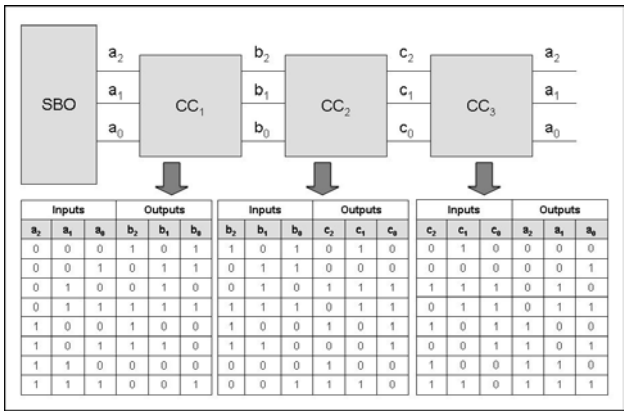
Within logic synthesis, there is the concept of technology mapping, i.e., transforming the logic description of a circuit (Boolean network) into a representation composed of interconnected instantiations of elements from a standard cell library, which is a set of logic functions previously implemented, from which any logic function can be mapped [2]. In order to assure the correct functioning of an IC made out of standard cells, the corresponding library should be verified. This verification should ensure that the library design meets all functional and other specifications [3].

The central idea of this work is to create a methodology to verify a given standard cell library in such a way that each cell is instantiated at least once and it is evaluated with all possible input combinations.

2. METHODOLOGY

The method creates a circuit with two main goals: first, it forces the use of at least one instance of each cell; and second, it guarantees that all input combinations will be exercised in at least one instance of a library cell. The resulting circuit is automatically constructed and mapped to ensure the test of individual cell instances. The method is divided into two steps: the creation of truth tables to be implemented and the technology mapping of the resulting circuits.

Truth tables are generated in such a way to do a one-to-one correspondence between a complete set of n-element vectors in the domain and another set in the image. The association among domain elements and image elements is random, changing the order in which the vectors are associated. The input of the first truth table corresponds to a vector in sequential binary order (SBO). The output of a given truth table is the input of the next one and so forth. The mapping is done individually for each output of the truth table. To perform the mapping we used the SIS software tool, taking the cell area as the only one out of the available parameters to estimate the cell cost [4].



a_2	a_1	a_0	b_2	b_1	b_0
0	0	0	1	0	1
0	0	1	0	1	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	1	0	0	1

b_2	b_1	b_0	c_2	c_1	c_0
1	0	1	0	1	0
0	1	1	0	0	0
0	1	0	1	1	1
1	1	1	0	1	1
1	0	0	1	0	1
1	1	0	0	0	1
0	0	0	1	0	0
0	0	1	1	1	0

c_2	c_1	c_0	a_2	a_1	a_0
0	1	0	0	0	0
0	0	0	0	0	1
1	1	1	0	1	0
0	1	1	0	1	1
1	0	1	1	0	0
0	0	1	1	0	1
1	0	0	1	1	0
1	1	0	1	1	1

Fig. 1. Methodology applied to a three-input cell library.

For instance, let a three-input standard cell library, as depicted in Fig. 1. Each CC_i (i=1,2,…,n) is a combinational circuit (CC), composed of functions that will be implemented by subsets of cells. In the first iteration, CC_1 takes as input an SBO (a_2 a_1 a_0) and generates b_2, the most signifying bit of the first random association, using the standard cell library with its original cost parameters. Standard cells already used to map the function b_2 will have their costs, represented in the library, maximized. This will be the new library used in the second iteration to map the next function b_1. After that, the new standard cells used to map the function b_1 will have their costs – represented in the library used to generate b_1 – maximized, and this will be the newer library used in the third iteration to map the function b_0. Next, the new standard cells used to map the function b_0 will have their costs – represented in the library used to generate b_0 – maximized.

The output of CC_1 – equivalent to the first random association (b_2 b_1 b_0) – will be the input for CC_2 which will generate a second random association (c_2 c_1 c_0), starting with c_2, then c_1, and ending up with c_0, maximizing the costs of new coming cells on each step, just like it was done to generate (b_2 b_1 b_0). This process is repeated until all the standard cells in the library have been used. In this example, it is supposed to happen with CC_2.

Once all standard cells of the library have been instantiated in the CCs the checking process can be

started: instead of generating a new random association of elements, the last CC in the IC will generate the identity association as output. This is the warranty that all standard cells are functioning properly. So, the output of CC_2 ($c_2\ c_1\ c_0$) will be the input for CC_3, which will be associated to the SBO ($a_2\,a_1\,a_0$). The set of all circuits CC_i is the logic specification of an IC capable of validating this standard cell library.

Changing the representation of cell costs in the library used to generate each logic function increases the probability of unused cells being used, although there is no warranty that all cells will ever be used. Indeed, there are some logic functions that can only be mapped into specific cells, which have already been instantiated. Therefore, some cells would be used repeatedly, while others would not be used.

3. EXPERIMENTAL RESULTS

The proposed methodology was implemented using the SIS software tool [4] to map the logic functions into standard cell libraries lib2.genlib and 33-4.genlib. The input of the first truth table corresponds to a nine-element SBO vector, since nine-input cells are focused in the experimental work.

The library lib2.genlib contains 29 cells, and 27 of them should be evaluated, since one cell corresponds to logic constant 1 and another one corresponds to logic constant 0. Table I and Fig. 2 show the results obtained. With three iterations, all cells were verified. So, only logic functions b_8, b_7, and b_6 were mapped into lib2 cells. However, the entire random association ($b_8 \ldots b_0$) is the input to the checking stage.

Table I: Experimental results for lib2.genlib.

Iteration	Input	Output	New cells
1st	SBO	b_8	23
2nd	SBO	b_7	3
3rd	SBO	b_6	1
checking	$b_8 \ldots b_0$	SBO	

The same library was verified, although without changing the representation of the cell costs in the library on each iteration. In the first iteration 23 cells were used; in the second iteration, two more cells were used; and from the third to the ninth iteration, the last two unused cells have not been evaluated.

The library 33-4.genlib is composed of 89 cells, out of which 87 should be verified. Table II and Fig. 3 summarize the results obtained. With four iterations, 65 cells have been evaluated. After that, no unused cell has ever been added to the list of verified ones. Therefore, the checking stage did not take place.

Table II: Experimental results for 33-4.genlib.

Iteration	Input	Output	New cells
1st	SBO	b_8	34
2nd	SBO	b_7	16
3rd	SBO	b_6	12
4th	SBO	b_5	3
5th…9th	SBO	$b_4 \ldots b_0$	0

4. CONCLUSION

Although, in the first experiment, it was enough to change the representation of the cell costs to verify all lib2.genlib cells, it can be seen on the second experiment – 33-4.genlib – that just by changing cell costs one cannot evaluate all standard cells in a library.

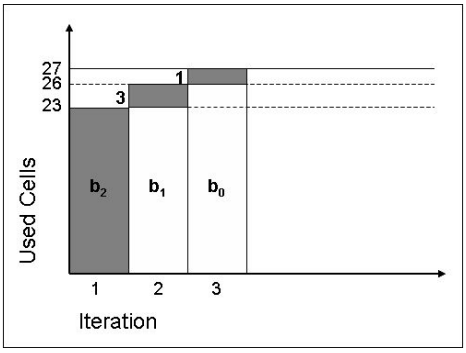


Fig. 2. Usage of standard cells in library lib2.

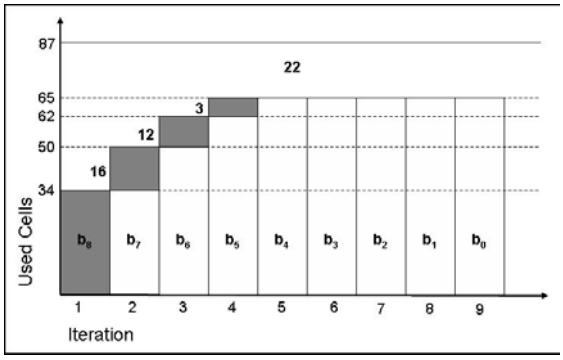


Fig. 3. Usage of standard cells in library 33-4.

A new approach must be used to force the instantiation of the remaining cells. In this case, binary decision diagrams (BDDs) – graph representation of Boolean functions [5] – shall be used to assure the evaluation of all cells in a library. Future work on this subject will focus on the feasibility of this BDD-based approach. So, the software will accomplish the verification of all cells, for all input combinations. As such, it can be concluded that this work in progress has achieved the expected results so far.

5. REFERENCES

[1] Weste N., Eshraghian K., *Principles of CMOS VLSI Design*, 2nd ed., Addison-Wesley, Reading, 1993.

[2] Correia V.P., *Mapeamento Tecnológico para Bibliotecas Virtuais Simétricas e Assimétricas com Minimização da Profundidade Lógica do Circuito* (master's thesis), Instituto de Informática da UFRGS, Porto Alegre, 2005.

[3] Bushnell M. L., Agrawal V. D., *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*, Kluwer, Norwell, 2000.

[4] Sentovich E. M. et al. *SIS: A System for Sequential Circuit Synthesis,* University of California, Berkeley, 1992.

[5] Minato S.I., *Binary Decision Diagrams and Applications for VLSI CAD*, Kluwer, Norwell, 1996.