Benchmark Circuits Improve the Quality of a Standard Cell Library

Rung-Bin Lin, Isaac Shuo-Hsiu Chou, Chi-Ming Tsai
Department of Computer Engineering and Science
Yuan-Ze University
Chung-Li, 320, Taiwan, R.O.C.

Abstract *-- The experience of designing and employing two benchmark circuits to improve the quality of a standard cell library is reported. It is found that most of the errors can be uncovered by making use of these two benchmark circuits to port the underlying cell library to the target environment. Two releases of a 0.25um standard cell library have been tested by these two benchmark circuits to ensure their quality.*

1. Introduction

A cell library is usually an ensemble of hundreds of each individual cell. If it is going to function as a whole to provide building blocks for larger designs, not only should each individual cell be correctly designed, but also the synthesized designs based on these library cells are absent from errors. However, designing of a cell library requires carrying out many complex tasks and involving design efforts from a number of engineers. Errors are easily made in this situation. Even in the case that each cell is designed correctly, errors may still exist. For example, layout rule violations may occur between two cells placed side by side even if the layouts themselves of these two cells are error free. In order to uncover the leftover errors of a cell library, benchmark circuits are usually designed to fulfill this mission. Thus, the quality of a cell library can be substantially improved.

In the past designing of a cell library is usually thought as a discipline in industrial community where the advancement in cell library design is largely made. Since a viable cell library will strengthen a company's competitiveness, the know-how with regard to designing a high quality cell library is not often seen in open literature [1-8]. However, with ever increasing availability of semiconductor foundry and commercial CAD tools to the university community and advancement in cell-based synthesis technology, more activities in cell library design within university community [9-11] have been carried out. Especially, a cell library is the most fundamental intellectual property (IP) for system-on-a-chip technology. The methodologies or problems of making a cell library into a viable IP can also be applied or found in building more complex IPs. Thus, the university community should be encouraged to do more research in designing of a high-quality cell library.

This work is supported by National Science Council, Republic of China under Grant NSC86-2622-E-009-009

There are certainly many interpretations for "high quality cell libraries". A cell library regarded as high quality by one company may not be considered as viable by another company. However, high quality cell libraries possess many common characteristics. Simply name a few here:

(a). the functionality of each individual cell should be correct in the models for logic synthesis and simulation,
(b). the timing performance figures of each individual cell claimed in the data sheet or models should be accurate enough,
(c). the layout of each cell should be free of design rule violations,
(d). the cells can be best utilized by a synthesis tool, and
(e). the cells can optimize placement and route of a large design.

In this paper we present our experience on designing and employing of two benchmark circuits to enforce (a), (b) and (c) which are essential to a high quality cell library. To justify (d) and (e) requires more benchmark circuits and elaborate works so that it will not be addressed further in this paper. The rest of this paper is organized as follows. Section 2 summarizes the types of errors which can be detected by the two benchmark circuits. Section 3 presents the benchmark circuit mainly used to uncover the errors belonging to error groups (1), (2), (3) and (4). Section 5 presents the benchmark circuit mainly used for certifying the timing performance figures (i.e. uncover the errors belonging to error group (5)).

2. Classification of Errors

The number of errors that could possibly occur in a cell library is enormous. It is impossible for us to list all of them. However, in order to facilitate our discussion, the errors that could possibly be detected by the two benchmark circuits are classified into the following five groups:

(1). *Incompleteness*: the type of errors resulting from missing the logical or physical entities for a particular cell. For example, the logical model for a particular cell is not present in the technology database for Synopsys Design Compiler. Or the functional module for a particular cell is not included in the database for Cadence Verilog-XL logic simulator.
(2). *Inconsistency*: the type of errors resulting from a cell's parameter not following a predefined value or standard. For example, the cell height of a particular cell is not equal to a preset value. Or the pins are not located on grids.

(3). *Functional error*: the logical function for a particular cell is not correctly formulated in the models for logic synthesis and simulation.
(4). *Design rule violation*: the type of errors resulting from violating the layout design rule when cells are placed side by side.
(5). *Inaccuracy*: the real timing performance figures are inconsistent with those claimed in the data sheets or in the models for logic synthesis and simulation.

Note that we simply list the possible types of errors being able to be detected by the two benchmark circuits when they are employed to port a cell library into a working platform. Although there may still exist some errors which can not be found by our approach, we experience that the released standard cell library [11] consists of very few leftover errors.

3. Benchmark Circuit I

This benchmark circuit is designed to detect errors belonging to error groups (1), (2), (3), and (4). In order to achieve this objective, all the cells in the library are modularized according to their logic functions [3]. If the cells are of like logic function, they will form a circuit module (or subnetlist). For example, all AND gates are put together in the subnetlist for AND gates, all OR gates should be in the subnetlist for OR gates, etc. Figure 1 shows the block diagram of benchmark circuit I. Figure 2 shows a schematic of the subnetlist for NAND gates. All the sequential cells (not shown in Figure 1) are also modularized in the same way.

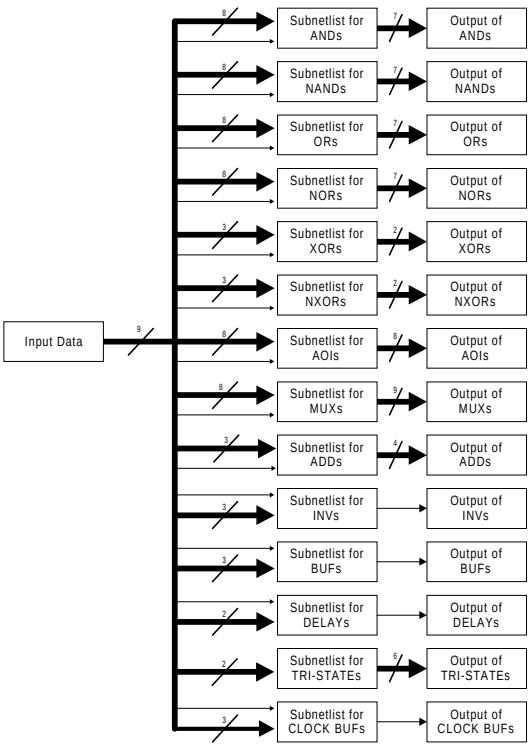

Figure 1. Benchmark circuit I

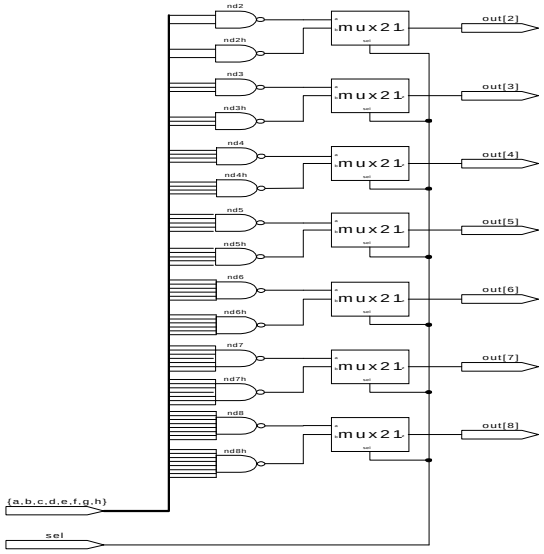

Figure 2. A subnetlist for NAND gates

A structural Verilog HDL module based on the underlying standard cell library is written to implement this benchmark circuit. The Verilog module is first imported to Cadence platform. Since all the cells are included in this circuit, if there is an incompleteness error (i.e., the logical module for a particular cell is not present in database), it will be caught here. This can also be done for the Synopsys synthesis technology database. The placement and routing of this circuit is then carried out. The inconsistency error will be spotted over the course of placement and route, while the layout rule violations are exposed by design rule checking. This approach has quite effectively spotted almost all the errors belonging to error groups (1), (2) and (4).

To uncover the functional errors in the models for logic synthesis and simulation requires more works. From now on, this benchmark circuit will be called *benchmark circuit under test*. Then, a circuit based on the intrinsic operators provided by the Verilog language itself is designed to generate a set of golden patterns. This circuit will be called *golden benchmark circuit*. The golden benchmark circuit is designed by the approach similar to that used to design the benchmark circuit under test. That is, the structure of the golden benchmark circuit is the same as that shown in Figure 1. Each cell instance except the multiplexer used to select the drive class for output in a subnetlist of the benchmark circuit under test will have a counterpart statement in the golden benchmark circuit. The counterpart statement is composed of the Verilog intrinsic operators. For example, the Verilog statement *nd2 U01 (a, b, z)* represents an instance of a 2-input NAND gate in the target library, where *nd2* is the cell name, *U01* is the instance name, *a* and *b* are the input pins, and *z* is the output pin. Then there will be a counterpart statement *assign z=!(a&b)* in the golden benchmark circuit to perform the function of 2-input NAND in the circuit. This counterpart statement is made up of the intrinsic operators

"*!*" and "&". Figure 3 shows the subnetlist for NAND gates in the golden benchmark circuit. The function of this module is the same as the schematic shown in Figure 2.

```
module GNands (a, b, c, d, e, f, g, h, sel, out);
    input a, b, c, d, e, f, g, h, sel;
    output [8:2] out;
        assign out[2]=!(a&b);
        assign out[3]=!(a&b&c);
        assign out[4]=!(a&b&c&d);
        assign out[5]=!(a&b&c&d&e);
        assign out[6]=!(a&b&c&d&e&f);
        assign out[7]=!(a&b&c&d&e&f&g);
        assign out[8]=!(a&b&c&d&e&f&g&h);
endmodule
```

Figure 3. A subnetlist for NAND gates in the golden benchmark circuit.

The golden benchmark circuit is carefully verified and then used to generate a set of golden output patterns. The golden output patterns will be used to check the functionality of the cells in the benchmark circuit under test. Figure 4 shows the basic concept of this methodology. Since the majority of cells in the underlying library have eight or less than eight inputs and most of the subnetlists for the like gates have similar structures, the subnetlists with 8-bit inputs and 1-bit selector only require 512 test patterns. Thus, a binary counter is employed to generate a set of test patterns ranging from 0 to 511. Both of the circuits are simulated with the same test patterns. The golden pattern file and the output pattern file are compared by "*diff*", an UNIX command, to find out the difference between two files. If no difference is detected, it indicates the functions of all the cells in the library under test are correct. Otherwise, there may be errors in the truth tables of some cells. This methodology has been used to test a 0.25μm standard cell library. Several functional errors have been detected.

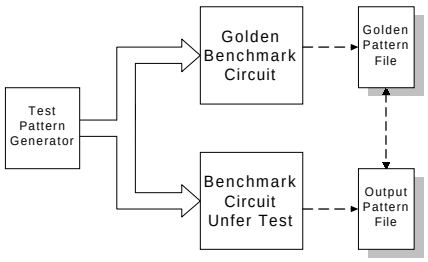

Figure 4. The concept of functionality verification

4. Benchmark Circuit II

This benchmark circuit is designed mainly to detect the errors belonging to error group (5). It can also be used to spot some errors belonging to the other groups. Although there are a variety of performance figures, they usually influence the timing performance figures. Thus, we will focus on the issues of timing performance verification.

The benchmark circuit for timing performance verification contains a variety of common logic circuits that allow us to evaluate the performance of a cell library. Figure 5 shows a block diagram of the benchmark circuit consisting of a *common block*, a *delay block* and a *special block*. The common block includes some common logic circuits performing some of the 74 series logic functions. The delay block includes unit cell delay circuit, fanout loading delay circuit and interconnect loading delay circuit. The unit delay circuit is employed to evaluate the delay of some basic gates. The fanout loading delay circuit is employed to evaluate the influence of fanouts on the cell delay. The interconnect loading delay circuit is used to evaluate the influence of interconnect on the cell delay. The special block includes bus circuit, tri-state buffer cells, ring oscillator, buffer cells and SSN (Simultaneous Switching Noise) test control circuit. Figure 6 shows a part of unit cell delay circuit.

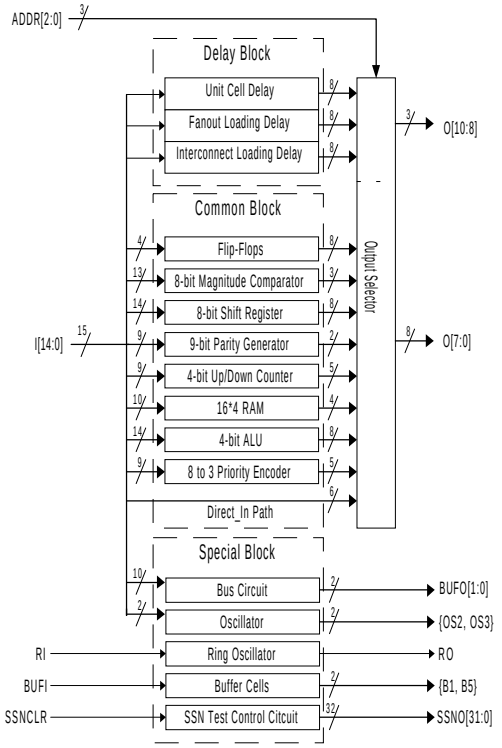

Figure 5. Block diagram of benchmark circuit II

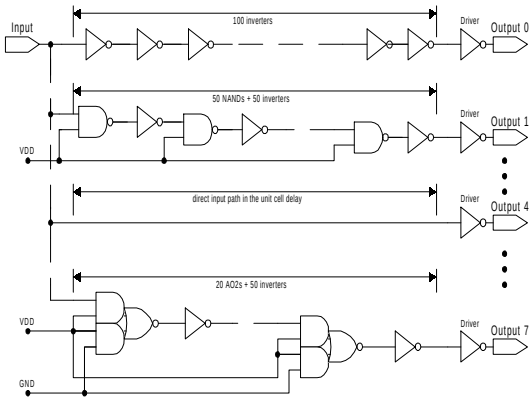

Figure 6. The unit cell delay circuit

The benchmark circuit is designed with Verilog HDL. After the function is verified by the Cadence Verilog-XL logic simulator, the benchmark circuit is placed and routed by Cadence Cell3 Ensemble. The layout is verified by LVS, ERC and DRC. One of the layout views of this benchmark circuit is illustrated in Figure 7.

Certifying the timing accuracy of a cell is often required to realize a real circuit through silicon implementation. After the real circuit is fabricated, a cell's timing performance figures are measured and compared to the timing data obtained by SPICE simulation. If they are matched, then the cell's timing performance figures are proved to be accurate. If they are mismatched, something must be wrong with timing characterization or SPICE models. Unfortunately, a university-made cell library usually has little chance to be qualified by fabricated test chips. So, we are trying hard to make our test chip fabricated in one way and to doubly check the timing performance figures of the circuits by running the benchmark circuit through TimeMill [12] in another way. We find that the timing delay figures obtained by TimeMill is not too far different from those obtained by the characterization system [13].

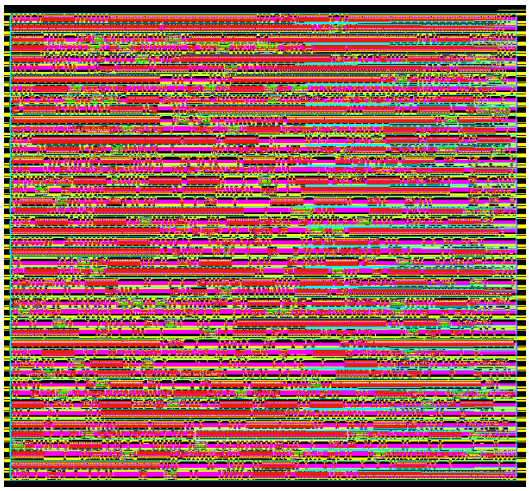

Figure 7. A layout of the benchmark circuit for performance verification

5. Conclusions

Two benchmark circuits have been designed and employed to improve the quality of a standard cell library. The types of errors, commonly occurring in a cell library, are first classified into groups. Each benchmark circuit is respectively designed to uncover different types of errors. We find that many errors could have been released together with the cell library if the benchmark circuits are not employed to uncover the hidden errors. Also, the benchmark circuits are designed with Verilog HDL such that they can be easily understood and highly adaptable to the needs of cell library development.

I/O cells form an essential part of a cell library. Although the two benchmark circuits can also be employed to find most of the errors occurring in I/O cells, more elaborate works must be done or special circuits must be designed to qualify the I/O cells. Realization of the I/O cells through silicon implementation is indispensable to qualifying their functionality. However, true silicon implementation to qualify a university-made cell library is still a hurdle yet to jump over for the coming years. If the hurdle can be removed, the academic IP research and design will thrive.

References

[1] W. Agastein, K. McFaul and P. Themins, "Validating an ASIC Standard Cell Library," Proceedings of the Third Annual IEEE ASIC Seminar and Exhibit, pp. P12-6.1-p12-6.4, September 1990.

[2] B.R. Blaes, M.G. Buehler, and Y-S Lin, "Propagation Delay Measurements from a Timing Sampler Intended for Use in Space," IEEE Transactions on Nuclear Science, Vol. NS-34, No. 6, pp. 1470-1473, December 1987.

[3] Bob Eisenstadt, "Improving CAD Software Quality," ASIC & EDA, pp. 38-41, December 1992.

[4] Lindsey Vereen and Mahendra Jain, "Benchmarks Define ASIC Performance," ASIC & EDA, pp. 18-25, September 1993.

[5] Ken Scott and Kurt Keutzer, "Improving Cell Libraries for Synthesis," Proceedings of the IEEE Custom Integrated Circuits Conference, pp.128-131, 1994.

[6] T. R. Bednar, R.A. Piro, D.W. Stout, L. Wissel, P.S. Zuchowski, "Technology-Migratable ASIC Library Design," IBM J. RES. DEVELOP. Vol. 40, No. 4, July 1996, pp. 377-386.

[7] S. Sunter "Designing a CMOS Standard Cell Library," Proceedings of the IEEE Custom Integrated Circuits Conference, pp.237-240, May 1987.

[8] D.V. Heinbuch, (Ed.) "CMOS3 Cell Library," (Addison Wesley, Reading Mass., 1988).

[9] J.W.W. Chong, R.G. Forbes, "Design of Basic CMOS Cell Library," IEE Proceedings G [Circuits, Devices and Systems], Vol: 139, Iss:2 , pp. 256-260, April 1992.

[10] http:// www.erc.msstate.edu.

[11] Chi-Ming Tsai, Jinq-Chang Chen, Issac Shuo-Hsiu Chou, Shu-Ren Ker, Jiunn-Ren Chen, Hui-Hsiang Tung, Tsai-Min Chiang, Chyi-Bin Lin, Rung-Bin Lin, and Yin-Kuan Lin, "Development of a 0.25um CMOS Standard Cell Library," Proceedings o f International Conference on Computer Systems Technology for Industrial Applications-Chip Technology, Hsinchu, Taiwan, pp. 73-78, April 1998.

[12] TimeMill User Manual, Release 3.3, EPIC Design Technology, Inc., 1995.

[13] Rung-Bin Lin and Shu-Ren Ker, "An Automatic Library Development System," The 8th VLSI Design/CAD Symposium, Taiwan, pp.237-240, August 1997.