Standard Cell Library Validation Methodology

M. de Carvalho, C. Nunes, B. Canal[1], L. Puricelli, L. Reinicke, G. Webber,
A. Neutzling[2], M. Altieri, E. Conto, T. Nagel, [3]P. Butzen, [1,2]R. P. Ribas, [1]E. Fabris

Institute of Informatics / [1]PGMicro / [2]PPGC, UFRGS, Av. Bento Gonçalves, 9500, Porto Alegre – Brazil.
[3]Federal University of Rio Grande (FURG), Av. Itália, Km 8, Rio Grande – Brazil.
.

Abstract*— In digital IC design, the standard cell-based design is the most used in the industry. It accounts on a mature validated cell library to quickly design a reliable commercial IC. However, cell libraries are constantly modified and a series of validation tasks are employed to guarantee correct IC design and fabrication. This work presents a methodology for validating a standard cell library at different levels, from cell design verification to silicon validation. We use the methodology to validate from scratch two 0.6µm CMOS cell libraries designed for the Brazilian industry.

Keywords*— Validation methodology, tests, measurements.

I. Introduction

In the context of the recent investment in Brazilian semiconductor industry, an agreement between *CEITEC S.A* and the *Federal University of Rio Grande do Sul* (UFRGS) was signed with the goal of developing a standard cell library in the X-Fab 0.6µm technology. The project is financed by government agencies and also involves *NSCAD Microeletrônica* and the federal *IC-Brazil* professional training program [1]. The project started in December 2013 and is scheduled to end in June 2016.

Digital ICs are easily developed by using *Electronic Design Automation* (EDA) tools that exploits a trusted standard cell library to synthesize a high-level hardware description into silicon layout for IC manufacturing. Standard cell libraries contain an ensemble of logic functions and their corresponding layout scheme for silicon implementation. The cell library can be acquired by specific vendors but is commonly provided by the foundry in a *Process Design Kit* (PDK) to ensure the correct IC design and manufacture. Moreover, cell libraries may be frequently redesigned targeting particular requirements on timing, power consumption, supply voltage, or other. Thus, once a new cell library version is available, a validation process is necessary in order to guarantee correct commercial design manufacturing.

Usually, a standard cell library is validated at different design levels: from cell design to silicon. The last level accounts on manufactured a IC containing effective benchmarks for evaluating the cells functions and electrical characteristics. If this test IC includes configurable operating modes, it could also bring up reasonable test information to validate functionally and electrically the library.

In this paper, we describe the main efforts and procedures needed to validate a standard cell library. In Section II, we define the validation levels at different design phases and, from Section III to Section VI, we detail the techniques adopted at each level. Then, in Section VII, we show the results of the proposed validation methodology. Finally, in the last section, we draw some conclusions and future works.

II. Cell library validation levels

The digital IC design exploiting the standard cell-based method accounts on a mature validated cell library usually developed in a full custom manner by designers. The validation process comprises of a series of validation tasks occurring at different levels, as follows:

a) Gate (cell) design
b) Standard cell design flow
c) Benchmarks for silicon sign-off
d) Silicon validation

At the cell design step, once the electrical characteristics have been established through the transistors sizing, the two validation tasks are performed: design rule check (DRC) and the layout vs schematic (LVS). The DRC process identifies whether the silicon and metal layers respect the limits of dimensions and distances according to the manufacturing process restrictions, whereas the LVS task verifies whether the circuit functionality extracted from the physical layout corresponds to the desired schematic circuit behavior, as well as verifying corresponding gate, ports and net names. Lastly, the cells are electrically stimulated to determine if output delays and voltages match according to specifications.

At the design flow phase, designers usually chose several commonly used benchmarks to validate the library in the flow and check whether they can be correctly synthesized into their respective physical layouts. These benchmarks can be microprocessor cores (µP), encryption cores, communication cores, and etc. Although DRC and LVS tasks are executed during the cell design, it is necessary to run them again on the target design, because the ensemble of cells assembled on the same layout may produce wrong connections, shape sizes and distances. Also, it is important to highlight that the physical layouts produced in this phase are not sent for manufacturing, they must undergo the sign-off check phase.

In order to be able to validate the library in silicon, specific benchmarks must be selected [2]-[6]. They should be capable of maximizing distinct cell instantiation and configuring several test modes. They should have auto-validation function and easy access to inner logic for detecting badly designed cells. Finally, they must allow the observation of the library maximum frequencies and voltages, usually limited by the technology and manufacturing process. Clearly, these specific benchmark differs from the ones used in the design flow validation level. On the other hand, not having an actual real case design makes the cell library validation task incomplete. Therefore we have also devised a real case 16-bit CPU used for didactic activities based on the architecture presented in [7]. In addition, we developed it by using an industry-like digital ASIC flow accounting on the newly designed cell library under evaluation.

As soon as the chips containing the benchmarks are manufactured and packaged, a silicon validation task is performed, encompassing manufacturing and characterization tests [8][9]. The former identifies possible manufacturing errors while the latter is capable of determining electrical and timing information of a device under test (DUT), determining whether the designs meet functional correctness according to specifications.

III. Gate design validation

Each cell in the library is carefully drawn by a group of layout engineers that take into consideration the technology constraints. Figure 1(a) shows a typical NAND gate layout with double-headed arrows delimiting dimensions that need to be compliant with design rules that are determined by the manufacturing process limitations. Figure 1(b) shows a 3x3 array of NAND gates and is commonly set up (as a possible design scenario) by designers to check DRC between different cells. The basic idea behind this cell validation technique is to tightly place adjacent to the target cell (C) the same cell design in several different reflected configurations: normal, upside down, and left-side right. The target cell is placed in the center of the middle row (M). Also, the top (T) and bottom (B) cells are reflected with respect to M such that the T and B cells can share, respectively, the same V_{DD} and GND as the M cells.

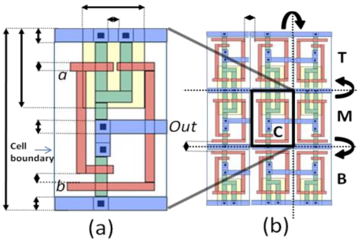


Figure 1. DRC in the cell (a) and between cells (b)

Another typical cell validation task is the LVS. Figure 2 shows a possible situation where DRC passes but LVS fails. From the designed cell (Fig. 2a), LVS extracts the electrical schematic (Fig. 2b) and compares it with the desired schematic (Fig. 2c). In the example shown in the figure, a connection (pointed out by the dark double-headed arrow) between ***b*** and ***out*** was mistakenly inserted in the polysilicon routing, so differing from the desired schematic. LVS also compares all gates, ports and net names between electrical and schematic levels, as highlighted in the picture by the light red double-headed arrow.

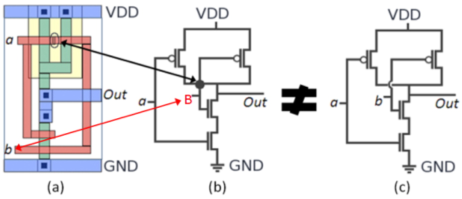


Figure 2. LVS procedure: (a) cell layout, (b) extracted schematic, and (c) targeted schematic

Finally, once DRC and LVS pass, the cell is thoroughly characterized, *i.e.*, it is stimulated in a simulation environment to obtain electrical and timing information (*e.g.* rising and falling transition times, output voltages, output currents and noise margins). This information has to comply with the desired specifications, otherwise the cell must be redesigned.

IV. Standard Cell design flow validation

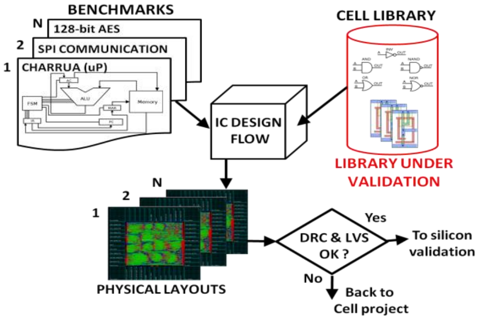


Figure 3. Cell library validation at IC design flow phase.

Once the cell library design has been completed and fully characterized, the IC design flow is explored to check that any sample benchmark can be sinthesized into a manufacturable silicon layout. The cell library validation at the IC design flow phase is shown in Figure 3. Several different real benchmarks (1 to N) are carefully selected as test cases in the flow. They must include typical circuits of different applications to stress the synthesis and mapping flow steps. At physical implementation phase, DRC and LVS are applied to each layout looking for any layout violation. If a failure is detected in any part of the design flow, the library under evaluation is not ready for silicon validation and must undergo cell design refinement. Note that in this phase, the physical layouts are not sent for manufacturing, they are only test cases to validate the cell library under the design flow usage.

V. Benchmarks for silicon validation

In order to make the cell library validation task easier, we have selected three special benchmarks. Two of them includes auto-validation functions for stimulating distinct cells. One is effective for evaluating the combinational cells and the other for the sequential cells [6][10]. Then, a third benchmark embedding a real case circuit was selected to perform a more realistic evaluation. It embeds both combinational and sequential logic in a typical industry-like design using the typical ASIC flow. All benchmarks are described in the following subsections.

A) Combinational cells

The most direct way of creating a chip specifically for cell library validation is by placing distinct cells parallel to each other sharing the same inputs and multiplexing their outputs [5]. This method can obviously validate each cells function, but electrical and timing information may be quite difficult to obtain through a circuit with only one logic depth, especially when attached to high capacitive input and output pins. Ideally, the cells should be placed in a circuit where they drive other cells. In [6], the authors proposed a methodology capable of automatically generating a combinational cell library benchmark for overcoming the above-mentioned issue, by placing all distinct cells in a circuit with several logic fan-outs and depths.

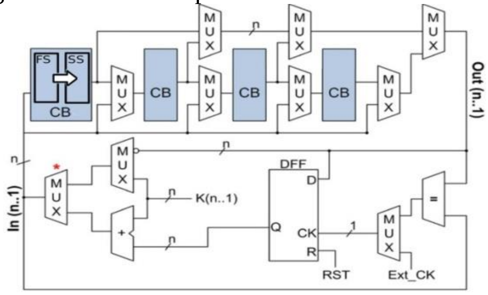


Fig 4. Benchmark for on silicon validation of combinational cells [6].

Figure 4 depicts the benchmark architecture which is composed of two main parts:

1) Combinational Blocks (CBs) (in blue) have two stages:

a. *First Stage* (FS): Distinct cells are arranged parallel to each other implementing a function denoted as: $\boldsymbol{f}$ **(*In*)**.

b. *Second Stage* (SS): Composed of a group of cells inverting the FS logic function, represented as: $\boldsymbol{f^{-1}}$**(*f* (*In*)).**

2) Operation mode Blocks (in white): The combined function of these modules allows selecting three test modes:

a. *Synchronous:* Tests the selected CB by analyzing the DFF register contents which are incremented at every clock cycle controlled by an external source

b. *Asynchronous:* Automatically tests the selected CBs performance by analyzing the DFF register contents, which is clocked when *In* and *Out* values are the same

c. *Closed-loop oscillation Built-In Self-Test (OBIST)*: Helps determining timing information of specific cells by oscillating a chosen path that propagates through the CBs.

B) SEQUENTIAL CELLS

An effective benchmark to validate sequential cells (FFs and Latches) must include testing all possible inputs by stimulating one at a time. One possible solution to achieve this goal is setting these cells in a ring oscillator configuration [8]. Figure 5 shows the proposed benchmark (3 inverting FFs and 1 buffer) which is capable of stopping ring oscillation whenever a cell is faulty, or not compliant with functional specifications. Also, it is effective on determining maximum oscillation frequency for a specific sequential cell library.

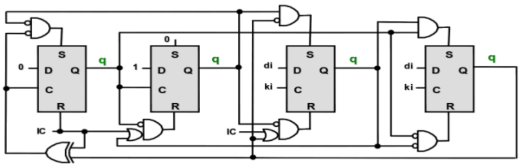


Figure 5. Benchmark for sequential cells.

C) REAL CASE BENCHMARK

The third benchmark comprises of both combinational and sequential cells in a real case study such that we can evaluate whether the ASIC design flow embedding the target cell library actually works. We have selected a 16-bit CPU core based on [7] which has also been used to validated the ASIC flow in Section IV. In the past years, this 16-bit CPU named *Charrua* has been used for IC design training within the IC-Brazil professional training program [1] and was initially created for teaching Computer Architecture and programming at the *University of Vale do Itajaí* [7]. The Charrua core described in [7], is placed in the top-level design together with two IP RAMs and an SPI bootloder using the APB protocol bus, as shown in Figure 6. Moreover, it has 6 test pins, 5 operational pins, 4 bootloader pins and 16 I/O pins for external communication and test purposes.

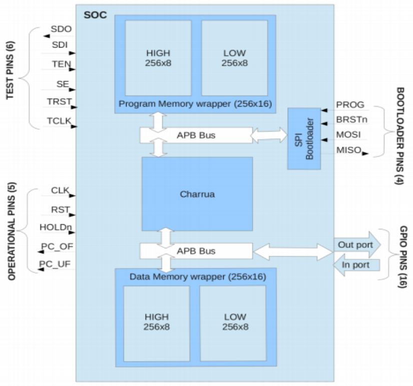


Figure 6. Proposed Charrua benchmark top-level view.

In addition, we have implemented a shadow scan-chain to evaluate and validate the benchmark. This means that, for each FF, we added in parallel a scan FF (SFF) such that their outputs are multiplexed. This strategy allows evaluating not only the combinational logic but also normal FFs, since Design-for-Test (DfT) synthesis usually replaces all normal FFs with SFF. This DfT strategy is depicted in Figure 7. If normal or scan FFs (but not both) have layout bugs, then the design can still be evaluated.

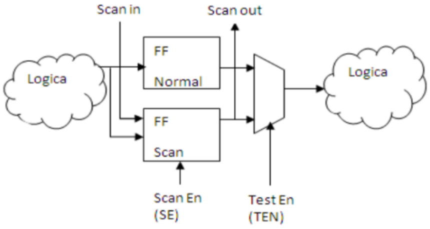


Figure 7. Design-for-test strategy.

VI. SILICON IC VALIDATION

Silicon validation tasks, usually called as *bring-up*, are the last step in the development of an IC. In contrast to pre-silicon verification that tests the IC model in a virtual environment, post-silicon validation occurs on the actual tangible IC prototype running slow and at-speed tests [8][9].

Silicon validation tasks basically falls into two categories: manufacturing and characterization tests. The first allows determining if the tested sample was correctly built by controlling and observing gate-by-gate net-by-net using structural test patterns. Once one good sample is found, characterization tests must be applied to this sample in order to gather important timing information about the manufactured device. In the latter test, test patterns are manually defined according to synthesis results and specifications. For both test types we employed a typical validation setup shown in Figure 8.

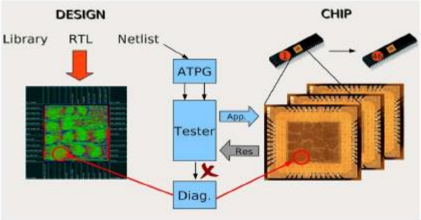


Figure 8. Proposed post-silicon validation setup.

First, we separately analyze the design model and physical chip perspectives. From the design perspective, we use the sign-off netlist of our benchmarks to generate test patterns. This task may or may not account on an Automatic Test Pattern Generator (ATPG). From the physical chip perspective, the chips will be stimulated by the tester according to the test stimuli previously generated and produce results that are collected by the tester. The measured results are then compared to expected golden results. For manufacturing tests, the expected results are computed during pre-silicon verification by using simulation tools. Whereas for characterization tests, the results are determined from the specifications and synthesis reports. Also, a diagnosis task allows analyzing faulty results and detect manufacturing errors or, for the purpose of our work; to determine cell and routing layout bugs, thus allowing cell library layout refinement.

Finally, we highlight that our benchmarks have internal pattern generators, self-test mechanisms and easy access to test ports to help us accurately evaluate the cell library without having to use probe stations or expensive automatic test equipments.

VII. CASE STUDY AND RESULTS

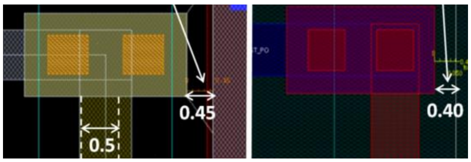
We designed two cell libraries versions in 0.6µm technology: one standard (CTC06ST) and one small area version (CTC06LA). Both libraries have 403 combinational cells and 26 sequential ones. We have started the validation flow herein described but there are still a long path to finish testing all samples in order to validate both cell libraries. In addition, our efforts account on a group of 16 people executing design and validation tasks using Cadence tools. Results of each validation step are described in the following subsections.

A) GATE DESIGN VALIDATION RESULTS

We had difficulties in this initial stage as re-design was often needed to meet DRC and LVS validation rules as well as sizing constraints. Common mistakes stemmed from simple last-minute modifications, which caused severe errors that were hard to detect. After iterating through cell design refinment and characterizing it each time, we were able to complete design and validation tasks for each cell. This initial phase required approximately 2000 hs on designing and 1000 hs on validation for both library versions.

B) STANDARD CELL DESIGN FLOW VALIDATION

We stressed the cell design flow by using different benchmarks: 128-bit AES encryption core, 128-bit AES fixed secret, different versions of a CPU core, the benchmarks in Section V, and several small peripherals, like SPI modules. Initially, summing all errors of all benchmarks, we obtained more than 20,000 errors. These problems were mainly notch errors that arouse from exporting a .DEF file from Encounter to Virtuoso.



(a) (b)

Figure 9. (a) Virtuoso and (b) Encounter .

Figure 9 shows an example of notch error in Virtuoso (Fig. 9a) occurring at this validation stage. As it can be seen, double vias had different definitions in the foundry technology .LEF file in both tools. In Virtuoso they were slightly dislocated to the left (50nm) as well as the connection (0.5μm to the right). Note that some layers cannot be seen by Encounter (Fig. 9b), so it does not report these DRC errors, whereas in Virtuoso, many of errors like in the picture were spotted. The reported DRC errors were eliminated at the instant the library .LEF file used by Encounter was modified to match Virtuoso technology file.

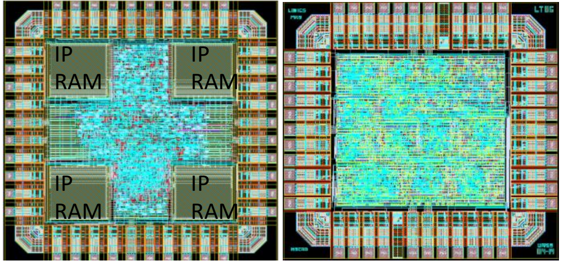
C) BENCHMARK VALIDATION

The combinational benchmark explained in Section V(A) was verified in logic and electrical simulations and validated on silicon using a mature 0.6μm technology provided by XFAB[11]. Simulation results and real measurements have shown to be very similar, especially the maximum frequency (~7MHz) when all combinational blocks were stimulated together. The sequential benchmark was also verified using logic and electrical simulations by combining several different FF ring oscillator configurations. Although results have not yet been published, they have shown to oscillate at 12MHz to 50MHz frequencies. Finally, the 16-bit CPU core was throughly verified and stimulated, achieving a performance of upto 20MHz clock frequency. Similarly to [12], an assembly code was evolved by an EA-based tool to maximize verification coverage metrics (~85%). Moreover, we exploited and ATPG and a fault simulator to evaluate functional and scan-chain test patterns coverages that provided more than 93% fault coverage.

D) SILICON VALIDATION

In order to silicon-prove the combinational cells functions of the CTC06ST and CTC06LA libraries, we generated 2 combinational benchmark described in Section V, one for each library, namely LibTestST (25 samples) and LibTestLA (25 samples) respectively. The LibTestLA also embedded several FF ring-oscillator benchmarks, described in Section VB. Moreover, the validation setup accounted on an Altera FPGA Cyclone III Terasic board implementing the automated tester for manufacturing tests. We also resorted to a 250MHz Agilent Oscilloscope, a voltimeter and an ex-Agilent Logic Analyzer 16903A to perform characterization tests to acquire timing data.

We performed manufacturing tests in the LibTestLA samples and results showed some combinational cell layout bugs. All samples presented the same errors that were detected exactly on one CB block and on the comparator logic. Although the layout had bugs, most cells were working correctly and preliminary characterization test results show that this small area cell library is slower by around 20% in comparison to results shown in [11] and is capable of reducing die area by 44%. Moreover, 3 sequential benchmarks using this library worked correctly, providing 7.5 to 12.5 MHz frequencies with a jitter that could not be correctly measured. Recently, we have adjusted the LibTestLA and finished the 16-bit CPU using the CTC06ST cell library version. Both sign-offs have been sent for manufacturing thanks to the Brazilian multi-user project offered by CEITEC. Figure 10 shows their corresponding 3x3 mm^2 sign-off layouts.



(a) (b)

Figure 10. (a) Charrua CPU (b) LibTestLA.

VIII. CONCLUSIONS

We have presented a standard cell library validation approach which tackles validation at several different stages. The proposed approach has the goal of delivering a trusted cell library to a customer which will take for granted that using it he will be able to manufacture any design for the target foundry (as long a certain number of rules are respected). We explained these validation steps starting from bottom-up, that is beginning from basic standard cell design to a real manufactured digital ASIC prototype that accounts on the provided cell library. Also, the proposed benchmarks have shown to be very effective on detecting layout bugs that helped quickly correcting mistakes and arranging another tapeout. Results have demonstrated that the proposed cell library validation approach is on the right track.

ACKNOWLEDGEMENTS

Project partially founded by CNPQ (grant 305113/2014-3) in partnership with CEITEC S.A.

REFERENCES

[1] http://www.ci-brasil.gov.br/index.php/en/

[2] Keshava, J., et al. *Benchmark Circuits Improve the Quality of a StandardCell Library*, ASP-DAC Design Automation Conference, 1999,p.173-176.

[3] Bo Liu et al. *Sub-threshold Custom Standard Cell Library Validation*, IEEE International Symposium on Quality Electronic Design, 2014.

[4] Agatstein, W. et. al, *Validating an ASIC standard cell library*, IEEE ASIC Seminar and Exhibit, 1990, p12/6.1 - p12/6.5.

[5] A.P. Singh, N.S. Panwar, *On Silicon Timing Validation of Digital Logic Gates A Study of Two Generic Methods*, IEEE International Conference on Microelectronics, 2006, pp.424-427.

[6] R.P. Ribas, et al. *Contributions to the evaluation of ensembles of combinational logic gates*, ed. Elsevier: Microelectronics Journal, 2011,pp.371-381.

[7] http://bipide.com.br/

[8] Keshava, J., et al. *Post-silicon Validation Challenges:How EDA and Academia Can Help*, IEEE Design Automation Conference, 2010, pp.3-7.

[9] Mitra, S., et al. *Post-Silicon Validation Opportunities, Challenges and Recent Advances*, IEEE Design Automation Conference, 2010,pp.12-17.

[10] R Ribas et Al. *Performance and Functional Test of Flip-Flops using Ring Oscillator Structure*. IEEE,Design and Test Workshop (IDT), 2011, pp42 47.

[11] M. De Carvalho et al., *On-Silicon Validation of a Benchmark Generation Methodology for Effectively Evaluating a Combinational Cell Library Design*, IEEE Latin American Test Symposium. 2016, pp.135-140.

[12] M. De Carvalho et al. *An Enhanced Strategy for Functional Stress Pattern Generation for System-on-Chip Reliability Characterization*. IEEE Microprocessor Test and Verification Workshop(MTV), 2010, pp29-34.