System and Framework for QA of Process Design Kits

M. C. Scott
Texas Instruments
Design Automation Group
6730 S. Tucson Blvd
Tucson, AZ 85716
scott_matthew@ti.com

M. O. Peralta
Texas Instruments
Device Modeling Lab
6730 S. Tucson Blvd
Tucson, AZ 85716
peralta_mike@ti.com

J. D. Carothers
University of Arizona
ECE Dept
P.O. Box 210104
Tucson, Arizona
carothers@ece.arizona.edu

Abstract

In this paper, we evaluate the dependencies between tools, data and environment in process design kits, and present a framework for systematically analyzing the quality of the design tools and libraries through the design flow. The framework consists of a regression engine which executes sets of tests in a distributed computing environment. These tests vary from simulations to validate models and simulators, to tests on layout versus schematics, parasitics extraction accuracy, and ultimately, tests to validate the extracted circuit integrity against the ideal. In particular, it is shown that test-chaining is required to obtain confidence in the simulation-to-silicon equivalence. A secondary objective is to identify and quantify the peak-error injection points. Finally, future work is outlined to extend the framework to automate entire design flows and provide capability for inter-tool constraint satisfaction and design optimization.

1. Introduction

The process of electronic design depends critically on the quality of electronic design automation tools and the integrity of their underlying libraries and environments. Conceivably, an error in any stage of the design process may propagate and expand further down the design flow leading to a critical fault. The existence and nature of such errors is hidden to the designer, and thus the designer must work blindly on a basis of confidence in the tools and libraries. Idealistically, the design team should need only consider the process-temperature-voltage corners and signal noise in defining the operating envelop of their design.

In this age of increasing complexity, shrinking geometries, higher frequencies, lower power and shortening market entry opportunities, it is already prohibitive to obtain design closure on signal integrity issues, not to mention contending with design kit tool and library errors. But, these two concepts: Design Kit Integrity and Design Complexity go hand-in-hand. Increasing demands on design performance and increasing susceptibility to signal integrity issues drive the design kit to increasing complexity. Likewise, the design kit complexity leads to exponential complexity in design susceptibility to kit errors.

In this paper, we present a systematic means of qualifying various stages and components of a design kit. Various cross-stage, cross-tool tests-chains propagate confidence. We are also looking to quantify the peak-errors of various stages and devices, and to identify opportunities for accuracy and efficiency improvement in both the design kit development process and in the design process itself. The paper is organized as follows.

In section 2 we evaluate the dependencies between tools, data and environment in a process design kit. The concerns of the design process and its error injection points must be considered when developing simulation and extraction tools in order to neither under or over-design the tools. Eventually, this knowledge will improve corners definitions and Monte-Carlo simulations.

In section 3, a framework for systematically validating the quality of the design tools and libraries through the design flow is presented. A tool "RegMan" (Regression Manager), is introduced which encapsulates the regression systems for validation of verification tools through the management and evaluation of jobs over LSF [3].

In section 4, Front-end device models and simulators, the methods for cross-checking models, schematics and simulation are presented. The RegMan tool is extended to run regression sets of the defined simulations and evaluations.

The Fifth section delves into the complexities of validation of Physical Verification tools such as DRC and LVS. The regression system was originally designed for this role as there are numerous tests to perform, and the evolutionary nature of the kits demands frequent re-runs of the test suites.

Section 6 discusses the use of the RegMan tool to execute various evaluations of a set of parastics extractions of layout structures. Comparisons of the resulting extracted netlists of a large suite of layout structures is made with respect to the industry standard 3D simulator Raphael [6] from Synopsys.

Section 7 wraps up the Physical Verification validation with a system for comparing simulations of extracted netlists to the ideal schematic simulation. The stage is motivated by the necessity to bind the layout to the original simulation - a more aggressive LVS.

In conclusion, sections 8 and 9 present a review of the merits of this system, and is possible extensions. As the tool has been expanded to exercise all stages of the analog design flow, it presents itself as a useful tool to designer and layout folk alike to run and manage the execution of large sets of jobs. For example, the tool may be extended to run a design through multiple Monte-Carlo simulations and corners as generated by extraction of minimum, typical and maximum parasitics.

2. Design Flows, Design Kits and Design Complexity

Entering the new millennium the semiconductor industry has found itself the victim of its own phenomenal success. The fierce competition that has driven down costs while simultaneously increasing performance metrics across the board has fostered an expectation and demand for continued progress. This progress has been found mostly in shrinking feature sizes and improved fabrication technologies. But, this path of improvement brings with it costs, complexities and side effects that must be addressed by advances in EDA tools and their underlying algorithms and usage models.

These complexities can roughly be partitioned through the design flow vertically into digital and analog centric. A major motivation of this project is addressing the various incarnations of signal integrity in many design types and attempting to evaluate, categorize and link these requirements in order to provide feasible solutions given the tools available.

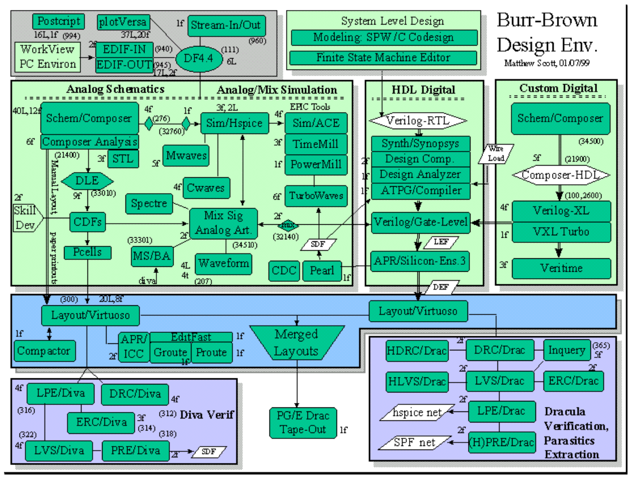


Fig 1. A 'simplified' design flow from Old days.

The design flow we evaluate principally consists of five stages: simulation, layout, layout verification, extraction, and back-annotated re-simulation. There are innumerable ways this flow may be constructed and extended. Figure 1 represents one implementation. There are multiple spice simulators, multiple layout tools, multiple DRC, LVS, and LPE tools. The device and cell libraries must work in all of them. Thus, the design kit development process is a bit involved. Many concerns must be taken into consideration and made to fit with the total kit objective. This leads to multiple re-works and gradual improvement. Given the interdependencies between libraries and tools, changes in any part may create faults propagated to others. A testing and sign-off quagmire results. The design kit complexity is directly driven by the design complexity and its need for multiple tools and flow, and their interdependencies.

DIGITAL COMPLEXITY

In the digital realm, design complexities are being dominated by interconnect parasitics and their signal-integrity degradation effects which introduce non-linearity's into the numerical optimization solutions provided by synthesis, placement and routing tools. The increasing dominance of variable interconnect effects over the static cell delays (gate delays) effectively diverges the synthesized timing estimate from the eventual physical. Numerical methods therefore become less viable and the designer is forced into a non-convergent state of iterations. The standard solution is to back-end layout parasitics extraction (LPE) and transistor level simulation. But, given the exponential increase in device counts, and its concomitant exponentially increasing state space, resorting to analog level analysis is often doomed to intractability. Solutions must be found that keep the complexity order of analysis linear with respect to device count, while complimentarily throwing more compute power behind the analysis to keep it tractable. This project addresses this complexity in providing a means for evaluating and validating various LPE and SI tools.

ANALOG COMPLEXITY

In the analog realm, the push towards higher frequencies and finer signal resolutions has demanded similar improvements in simulation and signal quality, such as signal matching. But, analog has a much more diverse bag of tricks than digital, which must be meticulously extracted to enable the analog formulae to work. Usually analog is not limited by the size complexity of digital, but rather the specification of constraints and the ability to optimize on those constraints in silicon. Critical to analog design is the accuracy of models and simulation, and the accounting for parasitic effects on sensitive lines in the layout. Knowledge of the accuracies of models and knowledge of the accuracy of parasitics extraction can help immensely in determining the degree of simulation coverage necessary to ensure performance.

DESIGN FLOW ERROR PROPAGATION

The presented expansion in complexity, speed, size, range and SI sensitivity of circuits is a problem that every engineer is aware of. The increases in all of these areas are plagued with inter-dependencies which are connected through the signal integrity realm. Each engineer is also acutely aware that there is a vast matrix of economic trade-offs in optimization of various parts, and likewise that there exists a chain of error introductions and variances within which they must exercise their design to ensure an envelop of operation during its expected lifetime. A short list of some of these concerns are tabulated in (Table 1). The typical engineer is not, however, empowered with the information or tools to globally account for all of these factors in their design planning and analysis. They can only set certain parametric goals and then hope that through use of various point tools and many iterations through the design flow they might converge to an acceptable solution with reasonable hope that all likely faults have been found. This work is also done with the assumption that the underlying design environment is correct and not changing, thereby allowing for a controlled-experiment environment. This is usually not the case. Importantly, the design engineer is not usually privy to the error levels of different tools, libraries and methods. A means is not available for management and satisfaction of design constraints across tools and flows. This must be provided by rigorous analysis of each stage, and development of an inter-tool constraint management utility. The system we present takes the first step in this direction.

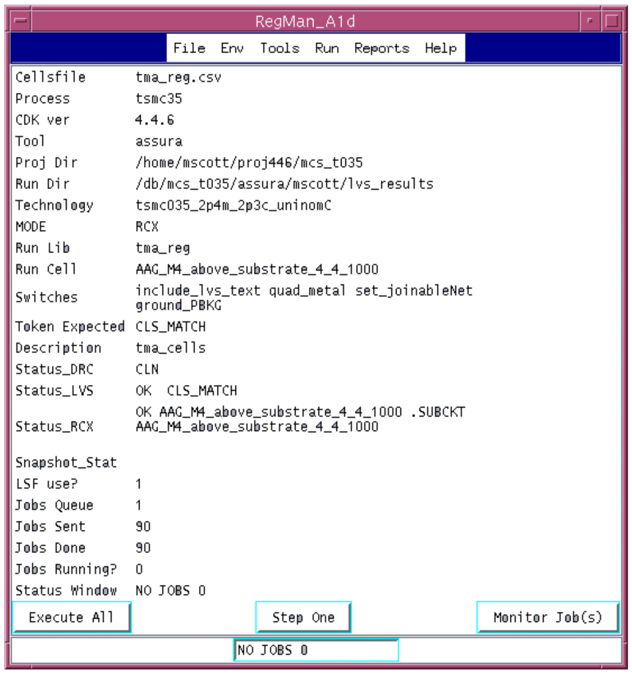
Table 1: Short list of Error Injection Points

Device Parameters Errors The choice of device parameters to isolate and characterize is usually limited. Most device modeling shops do not characterize for all operating regions recognized by the target simulation equations, instead opting to default to educated guesses. Probe noise notwithstanding.
Simulation Models, Equations Simulation accuracy can range from the quantum-field effects, through standard spice equations, on up to the table lookup methods of analog accelerators.
Simulation Program Accuracy Numerical Resolution, Time-Step Resolution Matrix Size
Monte-Carlo Coverage: Realistic bounds, Statistical Relevance
Process Corners Analysis May not be realistic, may not even be consistent
Statistical Simulations
Substrate Coupling Effects, ground-bounce
Schematic Entry Device Parameters Specification Simulation Parameters Specification
Pcell Device Layout Generation: Device permutations
Place and Route: Interconnect noise, delays
Design Rule Checks Percentage of Rules checked Types of Rules checked
Layout Parameter Extraction 2D, 2 ½ D, 3D, Field Solver, FEM RLC Extraction vs RC vs C Intra-Device Parasitics Geometries Extraction
LPE Re-Simulation High-Frequency IR Drop Node Compression, Reduction Methods
Mask Generation Optical Proximity Corrections, PSM
Fabrication Deposition over/under etch
Package Effects, Thermal Effects, Reliability Bond Pad RLC Substrate Noise

3. A Regression Manager

Regman (Regression Manager) is a graphical tool control framework designed to facilitate the distributed processing (LSF) automation of simulation, physical verification and parasitic extraction runs. It is generalized to work on any design kit and with any set of physical verification rules. It takes as input a comma separated vector formatted file describing test cells and their mode of pass/fail expectation. Outputs for physical verification validation (DRC, LVS) includes various reports on OK or NOK (not-ok) tests, matches, mismatches and failed runs. For the parasitics extraction, reports on accuracy of the extract tool versus an industry standard, TMA's Raphael, are reported]. On the simulation regression tests, results are compared to tables of pre-calculated expectation and summary reports are generated.

The tool is written in Perl/Tk and consists of 11K+ lines of code. The interface (Figure 2) enables the user to configure the environment, choose processes, choose various stages (LVS, DRC, RCX, SIM etc) to run on the cells list, choose subsets of cells to run, view the setup files for any cell, view log files and results for any cell. The list of basic options and actions available exceeds 160 and is beyond the scope of this report. Some of the side-advantages to RegMan include the creation of a snapshot of the environment for each cell run. This snapshot enables detection of changes in the environment which may render previous runs invalid. It has the side-advantage of enabling users to detect if the design-kit, schematic, layout, models or verification rules have changed.



```
Cellsfile       tma_reg.csv
Process         tsmc35
CDK ver         4.4.6
Tool            assura
Proj Dir        /home/mscott/proj446/mcs_t035
Run Dir         /db/mcs_t035/assura/mscott/lvs_results
Technology      tsmc035_2p4m_2p3c_uninomC
MODE            RCX
Run Lib         tma_reg
Run Cell        AAG_M4_above_substrate_4_4_1000
Switches        include_lvs_text quad_metal set_joinableNet
                ground_PBKG
Token Expected  CLS_MATCH
Description     tma_cells
Status_DRC      CLN
Status_LVS      OK  CLS_MATCH
                OK AAG_M4_above_substrate_4_4_1000 .SUBCKT
Status_RCX      AAG_M4_above_substrate_4_4_1000

Snapshot_Stat
LSF use?        1
Jobs Queue      1
Jobs Sent       90
Jobs Done       90
Jobs Running?   0
Status Window   NO JOBS 0

NO JOBS 0
```

Figure. 2 RegMan Interface

4. Front-End Device Models and Simulators

The accuracy of all tools and libraries are never any better than the accuracy of the device models. It has been found that rigorous measurement and characterization of the test chip devices, and cursory simulation tests are insufficient to guarantee accurate simulation. Tests in this project have revealed errors in the model parameters, in the netlists generated by the symbol and Pcell libraries, and in the simulators themselves. This section delves into the process of validating the models and simulators, and their consistency in more advanced techniques such as Monte-Carlo and Corners simulation. These tests are provided by the Modeling side, and are automated by RegMan through parameterized calls to a generalize Ocean script [4]. The Ocean script has built-in evaluators for each type of test, and thus is highly extensible to real design analysis.

The device models/simulation tests consist of the follow four classes.

I. Measurement vs. Simulation

Given a database of measured parameters, such as the models themselves, simulations are run on isolated devices and results compared to measured. The test fixtures include: DC I-V checks, capacitance over frequency, MOSFET capacitance checks, MOSFET transconductance checks, 1/f noise checks. Each test is repeated in each stage. Some tests are elaborated below. Figure 3 depicts the BJT I-V and MOSFET Cbg checks.

II. Schematic vs. Simulation

This type of quality assessment mostly applies to Resistors and Capacitors. For resistors a DC voltage source is applied across the resistor and the resulting simulated DC current is recorded. For capacitors, an AC voltage source of 1mv (rms), denoted v_{ac} , is applied across the capacitor and the resulting simulated AC current, denoted i_{ac} is recorded and used to calculated the Capacitance with the equation : $C = (i_{ac} / v_{ac}) / (2\pi f)$. In the current-voltage checks for Diodes, BJTs, and MOSFETs we merely bias the devices at typical design points and simulate what the currents are through the diode, the collector, or the drain terminals (or other terminals as desired). These values are then recorded and self-consistency checks are performed such as Simulator vs. Simulator, Measured vs. Simulated, etc.

III. Simulator vs. Simulator

In Figure 4, the results of a SimulatorX-Vs.-SimulatorY run of equivalent models is depicted. Some differences were expected, the others were quickly identified and remedied. The Sim-vs.-Sim also uses an Ocean script to compare signals. A simple rms(s1 – s2) is sufficient.

IV. Corners min, max vs. nominal, Monte-Carlo

Corners are checked against the nominal to make sure the low corner is less in value than the nominal and the high corner is greater in value than the nominal. Also low and high corners can be checked against the $\mu \pm 3\sigma$ values calculated from Monte-Carlo simulation runs. This can be done for DC current-voltage, AC capacitance-voltage, or any other type of device output characteristic including transient responses.

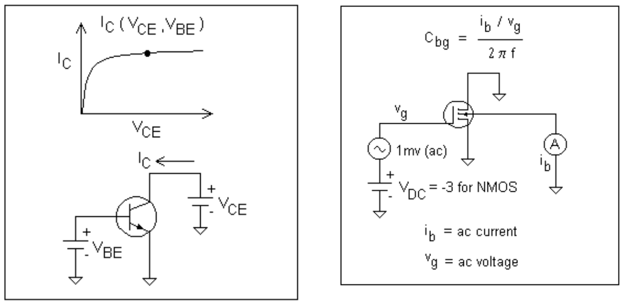


Figure 3. BJT I-V Check MOSFET Cbg Check

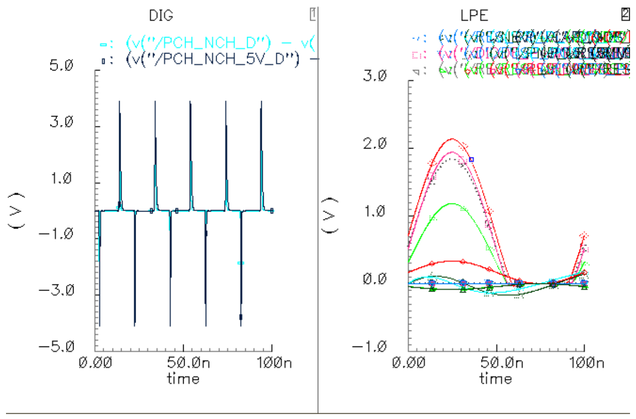


Figure 4. Two simulators results on equivalent circuits identified small differences in rise/fall time of MOSFETS, significant differences in diode simulation.

5. Physical Verification Tools and Flows Validation

Physical verification (PV) includes the Design Rules Checks (DRC) , Layout Vs. Schematics (LVS), and the follow on Layout Parasitics Extraction (LPE). The PV process is highly prone to error due to the almost superstitious ritual of transfer of a design from schematic to layout.

The DRC rules are generated from a long list of historical best practices and the subjective impressions of the Fabrication managers in terms of yield impact and reliability. A typical DRC rule deck may contain hundreds of rules, implemented in between 4000 and 6000 lines of code. According to the layout managers, these rules may implement only 2/3 of the actual constraints of the process. The rules are made to be as general as possible to avoid false errors and allow for layout creativity where warranted. Given that most rules apply to many permutations of geometry orientations and relations, the number of checks needed to test just the minimum and maximum boundary conditions of rules in intractable. In this respect, it becomes imperative to automate the process of rule-deck validation and accelerate through distributed processing. One can get by with the traditional pass/fail quilt, but the nature of the layers creating through Boolean operations within the rule decks leaves too much room for error whenever a rule is modified or added.

LVS rules are developed to ensure equivalence of the netlists generated from the schematics and layouts. Primarily, this entails checks of the isomorphic equivalence of the netlist graphs (or hookup), the device types and device sizes. Here there is quite a bit of leeway given to the actual construction of devices. For example, a MOSFET with W=x and L=y may be generated single-fingered, multi-fingered, inter-digitated with another device and surrounded by dummy poly. There are quite a few esoteric practices allowed in the translation from schematics to layout, including use of parasitic devices in the schematics, multiple-potential substrate regions for analog and digital sections, and smashing of parallel devices in either the schematic or layout. The lists of conceptual tests exceeds 200. The number of permutations of device constructions is in the hundreds. The list of Token/strings pairs by which RegMan evaluates the reports is about 100. Given that at least one relevant component in the kit is likely to change daily during kit development, this tool is seen as essential to the synchronization of parallel kit development.

6. Parasitics Extraction Tool Validation

To reduce the errors input from the physical design, Layout Parasitics Extraction (LPE) is employed to extract and back-annotate apparent final physical factors back into the design simulation – whereupon the design is re-simulated to determine if specs are still met. The effects induced by parasitics are lumped under the moniker 'Signal Integrity', or SI. The research, evaluation, development and implementation of LPE and signal integrity SI solutions usually falls into the domain of the Electronic Design Automation (EDA). As SI is the one facet of EDA which is coupled to all stages of design, it presents an added complexity of requiring high integration between tools and means of information sharing between various levels of tools to allow for best optimality at various levels. Thus, the solution that EDA provides for parasitics extraction and signal integrity must simultaneously be:

- Accurate: The analysis must be guarantee timing and SI simulations represent the real product
- Robust: The solution must be able to accommodate varying needs, resolutions
- Feasible: It must be simulatable within reasonable time and compute resources.
- Usable: The tools and methodologies must be integrated and accessible to the infrequent user.

The RegMan system facilitates the analysis of extraction tools to determine their accuracy. Likewise, it validates the 'rule-set' provided to the tool which defines the process technology and layers to be extracted. This EDA-team generated data is prone to error. Only through rigorous testing can we be certain that no error slipped through. To validate an extraction tool, comparisons are made against the industry standard TMA Raphael tool. The process consists of:

- Define the techfile for Raphael and LPE tool
- Raphael regresses over 11 primary topologies

  - Includes 35 permutations each on layers
  - Each layer permutes Width, Spacing, L
- Generate equivalent layout structures with Skill
- RegMan runs LVS and LPE over all structures
- RegMan parses the Raphael capacitance database
- RegMan parses the LPE extracted spice files
- RegMan compares and analyzes the results

The final report consists of the errors for each structure, averages for each primary layer and layer-pairs, and each class of W, S, L permutations. Said report facilitates identification of possible aberrations, and provides for determination of std. deviation and variance.

7. Validation of Extraction Circuits

The LPE generated netlist needs to be validated against the original (or ideal) schematic netlist to guarantee the equivalence of the layed-out intentional devices to their originators in the schematic. A schematic is created which contains instances of each of the model-symbol combinations, each being driven independently by some appropriate stimuli. A layout is then created equivalent to the schematic and then extracted with the LPE tool. The extracted layout is paired down to include only the intentional devices represented by the schematic. This is done by forcing the extractor to ignore device internal parasitics such as MOSFET AD, AD, PD, PS, and by stripping out the interconnect parasitics. The original schematic and layout are then simulated with the same stimuli driving each device. The resultant signals are compared through a simple difference v(s1) - v(s2).

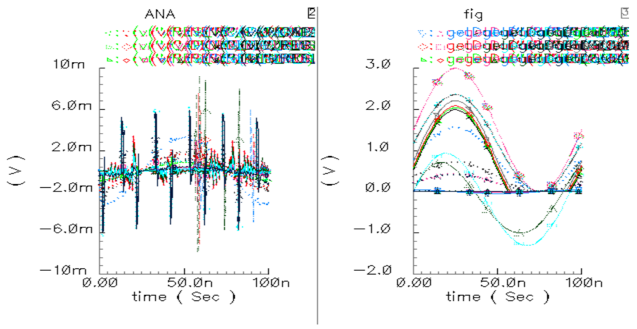


Figure 5. Extracted netlist vs. Ideal Schematic.

Since the signals should be identical, we should see only flat-lines on the waveform tool. Any fluctuation represents some difference in the device netlists between the original and the extracted. The script takes ∀ device pairs : v=rms(v(s1) - v(s2)), and reports and signal with rms > .00001 in a file and waveform such that the RegMan tool can automate and evaluate the process. In (Figure 5) the seft side represents MOSFETS, right side indicates Bipolar devices. The layout side still includes parasitics here.

8. Conclusion and Analysis

The discussed systems and regression manager framework have already proven invaluable to the development and validation of design kits. Many, heretofore undetectable, errors have been discovered and remedied. Accordingly, the tool has accelerated the development process by making comprehensive tests feasible with each step of tool or library modification. This in turn allows more aggressive kit development to implement more 'bells and whistles' with less trepidation of the likely error introduction. Better designs will ensue. Table 2 provides estimates on the benefits of this system.

Table 2. Estimated QA time: Manual vs. RegMan.

Test Stage	Manual Time	RegMan Time	Estimate # Tests	Repeats Estimate
Mod / Sim	6 wks	30 min	1000+	12+
DRC	4 wks	12 hrs	1000+	20+
LVS	2 wks	2 hrs	200+	20+
LPE	4 mo.	24 hrs	4000+	5+
LPE / Sim	1 day	1 hrs	20+	5+

Finally, A key factor in this system is the chaining, or overlap, of tests between tools and libraries. This propagation of 'confidence' can also enable visualization of error propagation and determination of peak-error injection points. Knowledge of peak and average errors improves design of tests for corners, MC and sensitivity.

References

[1] P. Chen, D. A. Kirkpatrick, and K. Keutzer, "Scripting for EDA Tools: A Case Study", ISQED, IEEE, 2001, pp. 87 - 93.

[2] M. A. Kraznicki, R Phelps, J R. Hellums, M McClung, R. A. Rutenbar, L R. Carley, "ASF: a Practical Simulation-Based Methodology for the Synthesis of Custom Analog Circuits", ICCAD, 2001, pp. 350-357

[3] Load Sharing Facility Users Guide, Platform Computing, http://www.platform.com/

[4] Ocean Users Guide, Cadence Design Systems, San Jose CA.

[5] Penberthy, J. S. and Weld, D., ``UCPOP: A Sound, Complete, Partial-Order Planner for ADL," Third International Conference on Knowledge Representation and Reasoning (KR-92), Cambridge, MA, October 1992.

[6] Raphael Interconnect Analysis Program, Reference Manaul, Synopsys, July 2000