A Collaborative Platform for Facilitating Standard Cell Characterization

Azam Beg  
College of Information Technology  
United Arab Emirates University  
Al Ain, UAE  
abeg@uaeu.ac.ae

Amr Elchouemi  
Walden University  
Minneapolis, MN 55401, USA  
amr.elchouemi@waldnenu.edu

Raahim Beg  
Liwa International School  
Al Ain, UAE  
raahim.beg@gmail.com

Abstract—Circuit designers need to characterize their designs based on different requirements of area, power, delay, and transition times. To do so, they may need to create a large number of circuits (defined as Spice netlists) in different configurations. The netlists not only contain the unit-under-test but also the mechanisms for providing the input stimuli and measuring variables (such as voltage, current, and time). The characterization is usually carried out by a team of engineers who may be located on the same site or in different geographical locations. A web-based collaborative platform proposed in this paper facilitates and expedites the characterization process by producing the required netlists in a consistent and error-free fashion.

Keywords—collaborative platform, digital circuit, standard cell, characterization, Spice, testbench

I. INTRODUCTION

Collaboration is an iterative process that involves a number of persons or organizations to realize a common goal [2].

The design of today’s complicated integrated circuits (ICs) requires management of multi-discipline teams that may be stationed in different locations around the globe. The complexity of design process mandates a collaborative work setup.

Thanks to the Internet and intranet-based resources, the circuit designers can make use of shared computer-aided design/engineering (CAD/E) tools. Although in their infancy phase, chip design and simulation (EDA) tools that run inside web-browsers offer the advantages of collaboration and accessibility. Other benefits of web-based tools are their availability on different platforms without customized installation. An application of such collaborative tools is the automatic creation of descriptions and testbenches for characterizing standard cells.

A. Standard Cells

A generic circuit design flow is shown in Figure 1. Standard cells are a set of basic circuits used to build larger circuits. Some examples of basic cells included in a cell library are INV, NAND2, and NOR2 gates. Gates with more than two inputs, flip-flops and input-output pads are also needed. Considerable time and effort are required to create the library that fulfills the desired quality criteria. Only the cells that meet the preset timing and power constraints are selected for synthesis [3].

Usually a set of design rules (DRC rules) is required for the layout of the cells, for example, the rules for the routing grid structure and for the cell sizing. The chip foundries provide the layout guidelines. The layout is followed by the extraction of Spice models that allow accurate verification of cell functionality. Sometimes schematics are also used because they may allow the designers a better understanding of the circuit, and hence the ease of debugging [4].

B. Cell Characterization

Spice modeling/simulation for verifying a circuit’s function is a key component of the design flow. The simulations allow extraction of power and delay, as well as detection of timing constraints (e.g., setup time). Depending on the type and the size of a circuit, and the number of design iterations, the simulations can take a long time. So the designers resort to characterization, a process which produces simple models for functionality, delay/timing constraints, and power at the gate or at the system level.
the cell level [5].

Cell characterization involves definition of some global parameters such as PVT (process, voltage, and temperature) corners, wire loads, and limits for transitions. Power extraction and IR drop analysis involves measurement of static and dynamic power, and the propagation delay. More simulations are done for optimization of area, power, and delay/timing. Simple models obtained from characterization process include functionality, dimensions, and capacitance. Delay and transition models are lumped together in one model, while power and constraints form two other models [5].

Depending on the type of characterization, different test circuits, called the testbenches need to be created. A testbench must include an instantiation of the circuit-under-test (unit-under-test/UUT). Unlike a hardware description language (HDL) based UUT which needs just the binary input patterns, a Spice-based UUT needs analog stimuli (e.g., ramp input, periodic pulses, etc.) Appropriate input stimuli and carefully-crafted commands for measuring voltage/current output and timing are also included in the testbench. Such work of a team of practitioners or researchers performing design, characterization and verification of circuits (cells) is highly collaborative in nature.

In this paper, we introduce SpiceGen, an online platform for automatically generating the Spice testbenches for standard cell characterization. The platform offers the users a set of 19 standard (combinational logic) cells and creates the requisite testbenches in an error-free and time-efficient manner. The platform has its applications in the fields of industry, research, and academia.

II. RELATE D WORK

One aspect of collaborative work focuses on enhancing the processes of conceptualizing a product and the related design tasks in a distributed, dynamic, and multi-disciplinary milieu. Such work is commonly known as collaborative design. Technological advances in communication in the past years have expanded the scope of engineering work beyond geographic limits and cultural barriers 0. At the same time, the academic and experimental learning is now possible and practical outside the classical, walled classrooms [6]–[8].

Cell characterization [5] entails the arduous task of running hundreds or even thousands of iterative Spice simulations [9]. Occasionally, use of models can expedite the characterization process [10] but the models cannot completely replace the use of actual simulations. Traditionally, the simulations are carried out on individual or in-house computers or servers, although Wilamowski et al. [11] proposed intranet/Internet-based versions of Spice. Recent examples of web-based Spice include [12] and [13]. Invariably, the job of creating the Spice netlists (i.e., UUTs and their testbenches) has still been an onus on the engineers.

A Spice-driven website for teaching power electronics covers experiments on just a single component (a power diode) [14]. Hashimoto et al.’s cell library generation tool has used simulated annealing for creating cell layouts for 130, 180, and 350 nm technology nodes [15]. An offline tool called AutoLibGen [16] has allowed characterization of cell library with 65 nm node (and not any state-of-the-art nodes).

III. INTRODUCTION OF SPICE GEN

Internet-based collaborative efforts cover a wide range of applications 0, but we have not come across any tools (offline or online) that generate Spice testbenches for characterization of standard cells.

One of the authors (of this paper) conceived and built an (offline) tool for automatically generating circuits (Spice netlists) and the corresponding testbenches. The Linux-based tool was built for individual use. The simulation results from the tool-generated netlists and testbenches have been utilized in multiple conference and journal papers [17]–[20]. Now for the first time, the tool has been enabled for online use by a large and diverse audience of academicians and practitioners.

The tool was originally developed as a set of Perl scripts. Perl is an interpreted language commonly used for handling text files, efficiently extracting information from them, and for generating reports. Perl is also a prevalent CGI (Common Gateway Interface) language for web development, and hence a logical choice for rapid transformation of our offline tool into SpiceGen – the online netlist-production platform.

IV. SPICE GEN’S STRUCTURE

A flowchart showing how SpiceGen works is shown in Figure 2. The netlists generated by SpiceGen depend on several input variables:

- Type of cell that needs to be characterized (19 CMOS gates): INV, buffer, NAND2, NAND3, NAND4, NAND5, NOR2, NOR3, NOR4, NOR5, AND2, AND3, AND4, AND5, OR2, OR3, OR4, OR5, or XOR2
- Technology node: 16 or 22 (nm)
- Supply voltage ($V_{DD}$) (volts)
- Transistor dimensions: $W_{pMOS}$, $L_{pMOS}$, $W_{nMOS}$, and $L_{nMOS}$ (nm)
- Type of testbench (further explained shortly): Voltage transfer characteristics; maximum frequency; or delay, current/power, power-delay product (PDP)
- Frequency of periodic input (in case of current, power, etc. measurement) (Hz)
- Inclusion of (HP PTM models [21]) MOS models in the testbench or not (Yes/No)

The netlists produced by SpiceGen depend on the type of the testbench:

- Voltage transfer characteristics: The netlist includes a single instance of the UUT. A ramp input is used for the INV-UUT; for all other gates a suitable combination of constant (either ‘high’ or ‘low’) input(s) and a ramp input are used.
• **Maximum frequency**: The netlist consists of a ring oscillator made of $n$-instances of the UUT. No external stimuli are needed.

• **Delay, current/power, PDP**: A single instance of the UUT is used but its inputs are driven by four series INVs; there is also a fanout-4 load for the UUT. Periodic pulses constitute the stimuli.

Appropriate ‘.measure’ commands are added to all netlists for measuring timing, current, and/or voltage.

Clearly, at this time SpiceGen only generates the netlists; the enhanced capability of online simulation of the circuits is in the works.

V. USING **SPICEGEN**

We present examples of two types of circuits/testbenches here. Figure 3 shows a circuit for generating voltage transfer curves for a NAND2 cell, and Figure 4 shows SpiceGen’s input screen for the curve-generation. Spice netlist is generated when ‘Submit’ button is pressed as shown in Figure 5. Similarly, Figures 6–8 depict an INV’s current measurement circuit, SpiceGen’s input screen and the corresponding netlist, respectively.

A practical application of SpiceGen is illustrated with the sizing of an INV. The goal was to find the lengths of nMOS and pMOS transistors that fulfill the performance (delay) criterion of a 22 nm INV. To do so we swept nMOS and pMOS lengths from 22 nm to 33 nm (an arbitrary range) with increments of 1 nm, for the purpose of measuring the gate delay. 121 different netlists generated by SpiceGen were simulated with NGSpice [22], and the simulation log files were parsed with a Unix-shell script. Finally, the data was plotted using Matlab. The results are shown as a 3-D/mesh plot in Figure 9. An alternative view using the contour plots (Figure 10) facilitates a designer’s task of finding the optimum transistor lengths that meet the timing target.

It is worth mentioning that the above example covers the investigation of a single design criterion (the delay) of a simple, one-input gate. Complete characterization of a gate is an iterative process and may require creation (and simulation) of thousands of different netlists, which is quite error-prone and time-consuming process, if not done automatically.

As mentioned earlier, the design or research team members may reside in one site or scattered around in different geographic locations. Regardless of how a team is positioned, SpiceGen enables the engineers to design and verify their designs in a consistent way.

SpiceGen also has many applications in academia. As an example, teams of students in a CMOS circuit design class may be asked to work on different cells. Use of SpiceGen allows the teams to create the circuits and their testbenches in a coherent manner. Yet another type of team assignment is studying the effects of variation in the cells operating at different supply voltages.

VI. CONCLUSIONS

We have presented SpiceGen, an online platform for automatically creating testbenches for standard cells. Currently, the cell library contains 19 basic gate variations. In a future version of SpiceGen, the users would be allowed to create characterization testbenches for arbitrary circuits. The additional capability of performing online simulations is also being looked into [23].
Figure 3. Circuit for acquiring voltage transfer curves of a NAND2 gate.

Figure 4. SpiceGen’s input screen for generating the Spice netlist for voltage transfer curve for a NAND2 gate.

-- nand2_1 VTC --
.include 22nm_HP.pm
vdd vdd 0 0.8V
.param vsupp = 0.8V
+wpmos = 88nm
+lpmos = 22nm
+wmos = 44nm
+lmos = 22nm
vin0 in0 0 pulse(0 0 1e-3 0.8)
vin1 in1 0 pulse(0 0.8 1e-3 0.8)
xuut0 in0 in1 out0 vdd 0 nand2_1
.tran 1e-06 1e-03
.print tran v(out0)
.measure tran vout find v(out0) at=0.5e-3
.subckt nand2_1 (input1 input2 output VDD VSS)
M1 output input1 VDD VDD pmos w=wpmos l=lpmos
M2 output input2 VDD VDD pmos w=wpmos l=lpmos
M3 output input2 wire VSS nmos w=wmos l=lmos
M4 wire input1 VSS VSS nmos w=wmos l=lmos
.ends nand2_1

Figure 5. SpiceGen’s output: netlist for generating voltage transfer curve for a NAND2 gate.

* -- inv current, etc. --
.include 16nm_HP.pm
vdd vdd 0 0.7V
vddUUT vddUUT 0 0.7V
.param vsupp = 0.7V
+wpmos = 48nm
+lpmos = 16nm
+wmos = 32nm
+lmos = 16nm
vin0 in0 0 pulse(0 0.7 1e-13 1e-13 5e-11 1e-10)
xinv00 in0 in01 vdd 0 inv
xinv01 in01 in02 vdd 0 inv
xinv02 in02 in03 vdd 0 inv
xinv03 in03 ins0 vdd 0 inv
xuut0 ins0 out0 vddUUT 0 inv
xld00 out0 out00 vdd 0 inv
xld01 out0 out01 vdd 0 inv
xld02 out0 out02 vdd 0 inv
xld03 out0 out03 vdd 0 inv
.tran 2e-15 2e-10
.measure tran ivdd AVG i(vddUUT) from=0 to=2e-10
.subckt inv (input output VDD VSS)
M1 output input VDD VDD pmos w=wpmos l=lpmos
M2 output input VSS VSS nmos w=wmos l=lmos
.ends inv_1

Figure 6. Circuit for measuring current, delay, etc. of an INV.

Figure 7. SpiceGen’s input screen for generating the Spice netlist for current measurement of an INV gate.

Figure 8. SpiceGen’s output: netlist for measuring the current of an INV gate.
REFERENCES


