An automated CAD tool for rapid technology characterization

Conference Paper · December 2017

DOI: 10.1109/ICM.2017.8268853

3 authors, including:

Omar Elgabry
Cairo University
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Ahmed Nader Mohieldin
Cairo University
78 PUBLICATIONS **667** CITATIONS
SEE PROFILE

An Automated CAD Tool for Rapid Technology Characterization

Omar Elgabry, Faisal Hussien, Ahmed N. Mohieldin
Electronics and Electrical Communications Engineering
Cairo University, Giza, Egypt
Email: omar.ahmed.elgabry@gmail.com, faisal@eng.cu.edu.eg, anader@eng.cu.edu.eg

***Abstract*—In this paper an automated CAD tool for rapid technology characterization is presented. The proposed methodology fills a gap found in old, as well as, new IC technologies. The tool allows doing a rapid characterization for different device models available in a technology. The tool is based on a pre-defined suite of test-benches in addition to SKILL code and OCEAN script to allow seamless integration in design environment. Finally, a design example is shown using 130 nm CMOS technology; results are shown for MOSFET devices characterization.**

I. Introduction

Recently, microelectronics technologies have been developing rapidly, resulting in an updated or new process development kits (PDKs) for designers to use. Each technology foundry is trying to optimize and enhance its technologies, and moves to smaller nodes to gain higher performance, area reduction, and lower power consumption maintaining the same functionality. This enhancement and optimization involve many aspects in the technologies, not only the spacing rules but also enhancements in the device modeling to get more realistic operation for the device. It also involves adding new devices for the technology. This is valid not only for the MOS transistors, but also for bipolar, resistors, capacitors...etc.

Nowadays with the huge activities from the technology foundries in the deep sub-micron technology i.e: 28nm, 14nm, or even the new 7nm, following new technologies updates is a challenging task for circuit designers or computer aided design (CAD) teams in companies, since they may be missing the experience to easily judge how this new technologies work or how this new enhancement will affect already existing designs. There is no fast and easy way to evaluate these new technologies, and understand how the devices in these technologies work.

There is also no systematic approach to know what the effects of these devices are, or to judge if the device models that will be used in the real designs are good enough and shows the expected results or not. The same gap exists with the new releases of an already existing technology.

As a result, an overhead time and cost is added to each PDK release for all circuit and CAD designers to explore and figure out the new features and challenges. Previous works to develop CAD tools for design automation in nanometer CMOS technologies have been reported [1][2]. This paper focuses more on one of the fundamental topics in the new technologies or enhancements of existing technologies, which is device modeling and characterization. There is no automated approach, as far as the authors know, to evaluate either a new device added to the PDK, nor to evaluate the new model updates for existing devices. Thus, circuit designers are not able to expect the effect of the new models on the existing circuits' functionality. This effect is even worse when a project is in a tape-out phase and an updated PDK is released. The designer has to judge which blocks or performance features need to be checked. Running the full simulation deck can possibly delay the tape-out date which is reflected in a cost hit and time-to-market for the product.

In this paper an initial implementation trying to fill this gap is provided. This will enable circuits designers or CAD teams in companies to evaluate the models of the new technologies rapidly, and understand how devices in these technologies work. In addition to that it will help them to fast understand the effect of the enhancements that are done to an existing technology. Thus they can compare models before and after modifications, and easily judge the effect of the updates on the performance of the designed circuits.

II. Tool Implementation

The proposed work concentrates on characterization of MOSFET devices. We understand the need of the designers or the CAD teams to run a tool and execute a lot of simulations automatically, without spending very long time and doing this step for each technology or for each PDK release. The user of the tool is not expected to be a technology expert or a circuit designer with a lot of experience to know how to create a specific test-bench to test a certain effect of the MOSFET device.

The tool is easily integrated to the design flow environment, so it is not required to have an expert to integrate this tool.

A. Design Methodology

The tool is based on a suite of pre-defined test-benches. The test-benches cover the intended checks/characterization that is required for the MOSFET device. User needs to select the device under test and all the remaining process is automated.

In Fig. 1 the proposed design flow methodology is presented. The methodology algorithm is shown in the following steps:

1) First step is to select the simulator that is needed to execute the simulation normally. Most of the design simulators such as Spectre or Eldo are supported by the tool. This only depends on the model files provided by the PDK and the supported simulator in the user design environment.

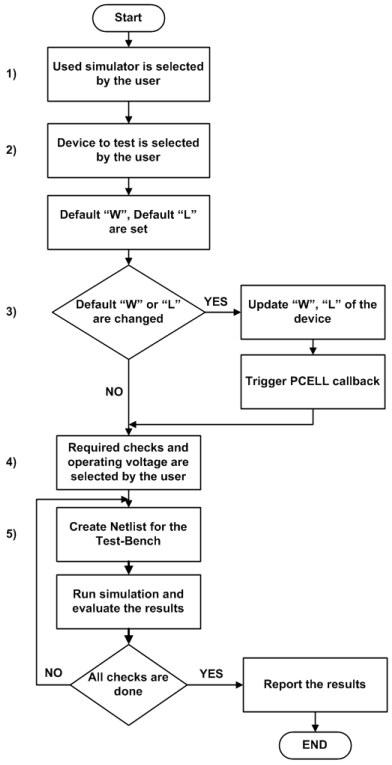

Fig. 1. Design Methodology Flow Chart

2) The user will start to select the device which is needed to be characterized using the browse GUI. As soon as the device is selected the default values for the length (L) and width (W) that are set by the PDK will be displayed, so that the user can know what the L and W values are.
3) In step 3 the user has the chance to decide either to characterize the device with the default L and W, or change their values to have a wide range of inspection. If the user decided to change length or width, the value of the changed parameters will be updated within the test-bench. Parameterized cell (P-cell) callbacks should be triggered to make sure of the consistency of the P-cell parameters with respect to the PDK settings, and also to make sure that the user did not enter an invalid value with respect to the PDK. Invalid L or W values may lead to a simulation errors or unexpected results in the simulation. If an invalid value is entered, the callback function will set this value to a valid one.
4) The values of the operation voltages should be set by the user. These values will be updated in the test-bench, moreover the user have to select the required characterization needed for the device, from a suite of pre-defined checks using check boxes provided in the GUI.
5) Then, the user will launch the tool. The tool will automatically select the test-bench related to a certain characterization/check, and will create the netlist for the required test-bench. The test-bench is run and the simulation results are evaluated and saved. These steps are done for each single characterization/check selected by the user and the tool will loop on all of required checks automatically.
6) At the end, the tool will report the results of the selected characterization either in the for of a graph or a comma-separated values (CSV) file that can be exported to an Excel sheet for later post processing or comparison.

Using the proposed flow, the user will have the possibility to characterize any number of transistors for a new technology or an existing one in an easy and fast way, without the need to build test cases for each device. This will also facilitate the comparison between transistors in different technologies or the same transistor after a PDK update.

B. Design Implementation

The tool implementation is based on two main parts:

1) Test-bench: it is a pre-designed circuit schematic, for a certain MOSFET characterization. Different schematics for different characterizations are technology and foundry independent making it easy to be used for any technology/foundry, during the execution.
   The flow automatically indicates the required test-bench for each characterization, and changes the generic instance in different test-benches with the selected device by the user. The instance parameters(for example L or W) are updated, before the flow starts generating the netlist of the test-bench.
2) Automation code: the code is built using SKILL and OCEAN script. The reason of using SKILL and ocean is to be able to integrate the tool easily within Cadence environment, where a big number of designers nowadays are using it as a design platform. SKILL is used to build up the user interface shown in Fig. 2, update the test-benches by the chosen device, update the device L or W and triggering the P-cells callback, and process data entered by the user that will be needed later for the simulation execution i.e.: chosen checks and operating voltages. OCEAN script is the main interface to preform the netlisting and execution of the simulations. It also enables us to support most of the available simulators.

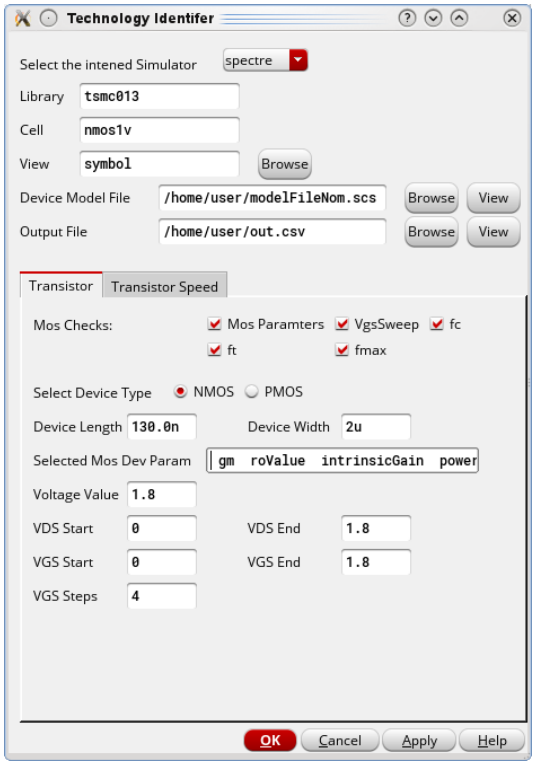

Fig. 2. Tool User Interface

C. Supported MOSFET Characterization

The tool currently supports the following checks for a MOSFET characterization:

1) MOS DC parameters where the user decides which parameters are relevant. When the user selects 'MOS Parameters' in the GUI shown in Fig. 2, user can then choose the parameters to be displayed. Some of these parameters are available from the DC operating point such as: g_m, v_{th}, g_{ds}, MOS operating region, all device currents i.e: i_{ds}, i_{gs}..etc and all device capacitance i.e: c_{gs}..etc.
   Other parameters are pre-defined in the tool such as:
   a) Drain output resistance "r_o" which is calculated at a given voltage using the equation:

$$r_o = \frac{1}{g_{ds}} \tag{1}$$

   b) Intrinsic gain which is calculated at a given voltage using the equation:

$$Intrinsic\,gain = g_m r_o \tag{2}$$

   c) Power efficiency which is calculated at a given voltage using the equation:

$$Power\,efficiency = \frac{g_m}{I_{DC}} \tag{3}$$

2) I-V Characterization which can be selected from 'VgsSweep' in the GUI, shown in Fig. 2, and it is reporting the I-V characteristics for the Transistor I_{DC} with respect to V_{DS} .
3) Corner frequency 'f_c' which is the frequency where flicker noise is equal to thermal noise. There is no direct way to evaluate f_c using noise simulations. Using simulator's noise analysis simulation, we obtain the value of total noise at a certain low frequency $f_1$that will be dominated by the flicker noise as shown:

$$V_{flicker} = \frac{K}{C_{ox} W L f_1} \tag{4}$$

where K is a process-dependent constant, C_{ox} is the oxide capacitance per unit area in MOSFET devices. In eq. (5), we evaluate a value for parameter M that is independent of the frequency

$$V_{flicker} * f_1 = M = \frac{K}{C_{ox} W L} \tag{5}$$

The value of total noise is then calculated at high frequency, that will be dominated by thermal noise which is independent of frequency as show in eq. (6).

$$V_{thermal} = 4kT\gamma g_m \tag{6}$$

where γ is a parameter widely used [3] to demonstrate the enhanced channel noise in short-channel transistors, k is the Boltzmann's constant, and T is is the absolute temperature. Thus using eq. (5) and (6), we obtain:

$$V_{flicker} = V_{thermal} \tag{7}$$

$$\frac{M}{f_c} = V_{thermal} \tag{8}$$

$$f_c = \frac{M}{V_{thermal}} \tag{9}$$

4) Unity current gain cutoff frequency 'f_T' is the frequency at which the current gain becomes equal to 1. It can be also activated by selecting 'ft' from the GUI, f_T is calculated by S-parameters using eq. (10) and (11) [4]

$$\mid H_{21}(f = f_T) \mid = 1 \tag{10}$$

where $\mid H_{21}(f) \mid$ is the magnitude of the current gain as a function of frequency, and can be obtained from the simulated Y-parameters of the transistor

$$\mid H_{21}(f) \mid = \frac{\mid Y_{21}(f) \mid}{\mid Y_{11}(f) \mid} \tag{11}$$

5) Unity power gain cutoff frequency 'f_{max}' is the frequency at which power gain becomes equal to 1, and it can

be activated by selecting 'fmax' from the GUI, f_{max} is calculated by using S-parameters as follows [4]

$$\mid MAG(f = f_{max}) \mid = 1 \tag{12}$$

where MAG is the maximum available gain defined as:

$$MAG(f_{max}) = \frac{1}{4} \mid H_{21}(f = f_T) \mid^2 \frac{r_{oeff}}{R_g + R_s} \tag{13}$$

$$MAG(f_{max}) = \frac{f_T^2 * r_{oeff}}{4 f_{max}^2 (R_g + R_s)} = 1 \tag{14}$$

where r_{oeff} is the effective drain output resistance, $R_g$is the input gate resistance and R_s is the source resistance.

6) Transistor speed is calculated based on the speed of an inverter. We evaluate the value by calculating the average propagation delay t_p .
To do so we inject an input step signal to the inverter and evaluate the propagation delay when the output changes from high to low (90%) t_{pHL}, and the same is done with the propagation delay when the output changes from low to high (90%) t_{pLH}

$$t_p = \frac{t_{pLH} + t_{pHL}}{2} \tag{15}$$

III. Results

A design example using TSMC 130nm CMOS technology will be used to show the results from of the tool. The results show the characterization for NMOS1V device versus NMOS3V device. Fig. 2 shows an example of how to set up the user interface with all needed information to set for NMOS1V device. All the possible checks mentioned in section II.C are selected for the design example.The same can be repeated for the other device NMOS3V.

In Fig. 3 and Fig. 4 a snapshot from the CSV output file exported in Excel sheet is presented, with the following parameters: c_{gs}, g_{ds}, v_{th} ,i_{ds},r_o ,g_m, intrinsic gain, power efficiency, f_c, f_T, f_{max} and transistor speed at 1.2V for the NMOS1V, and 3V for the NMOS3V. Although more NMOS parameters can be reported but these results were selected for showing the concept only.

In Fig. 5 shows the I-V characteristics for the NMOS1V operating between 0V and 1.2V compared to NMOS3V operating between 0V and 3V.

device	cgs (f)	gds	vth (V)	ids (A)	ro (Ohm)	gm (S)
nmos1v	-1.46E-15	0.000117	0.698305	0.001442	8528.7	0.0012716
nmos3v	-1.18E-15	2.93E-05	0.781723	0.001075	34113.6	0.0005172

Fig. 3. Device parameters(c_{gs}, g_{ds}, v_{th},i_{ds},$r_o$and g_m)

device	Intr. Gain	Pwr Eff	fc (HZ)	ft (HZ)	fmax (HZ)	Tr Speed (s)
nmos1v	10.8454	0.881603	2.03E+07	9.82E+10	8.00E+10	5.00E-11
nmos3v	17.6423	0.481194	7.64E+06	3.17E+10	2.00E+10	6.00E-11

Fig. 4. Device parameters(intrinsic gain, power efficiency,f_c,f_T,f_{max} and speed)

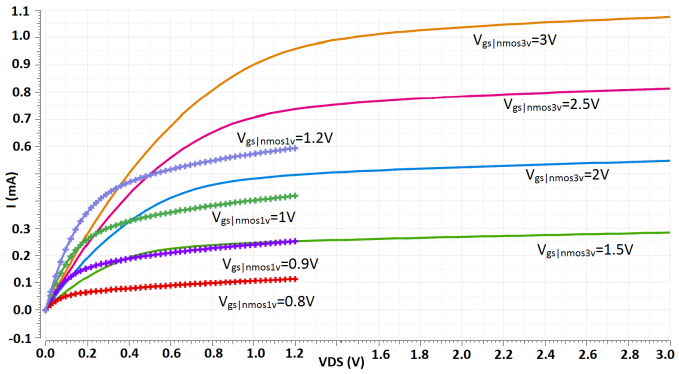

Fig. 5. I-V Characteristics

IV. Conclusion

An automated CAD tool for rapid technology characterization was presented. It is based on combination between pre-defined test-benches with SKILL and OCEAN scripts. The presented tool is technology independent which renders it suitable to be used by wide range of users. The current available features of the transistor characterization were shown with an example of the user's interface and the reported results and outputs.

The tool presented is still under further development and more features will be added and presented in the future.

References

[1] K. Siwiec,T. Borejko,W. A. Pleskacz "LC-VCO Design Automation Tool For Nanometer CMOS Technology", Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2012

[2] K. Siwiec,T. Borejko,W. A. Pleskacz "CAD Tool for PLL Design", Design and Diagnostics of Electronic Circuits and Systems (DDECS),2011

[3] S. R. Nassif "Technology modeling and characterization beyond the 45nm node", Design Automation Conference, Asia and South Pacific, 2008

[4] S. Voinigescu "High-frequency devices" in High-Freqency Integrated Circuits, February 2013