International Journal of Systems Engineering
2021; 5(1): 25-33
http://www.sciencepublishinggroup.com/j/ijse
doi: 10.11648/j.ijse.20210501.14
ISSN: 2640-4222 (Print); ISSN: 2640-4230 (Online)

LDOCad – A Behavioral Model Generation Tool for an LDO Linear Regulator

Yong Gao[1], Chuanfeng Wei[2]

[1]Office of Science and Technology Administration, Macro Net Communication Co., Ltd., Beijing, China

[2]Space Segment System Design Department, Macro Net Communication Co., Ltd., Beijing, China

Email address:
yong_csic@163.com (Yong Gao), chfwei@163.com (Chuanfeng Wei)

To cite this article:
Yong Gao, Chuanfeng Wei. LDOCad – A Behavioral Model Generation Tool for an LDO Linear Regulator. *International Journal of Systems Engineering*. Vol. 5, No. 1, 2021, pp. 25-33. doi: 10.11648/j.ijse.20210501.14

Received: April 1, 2021; **Accepted**: May 10, 2021; **Published**: May 15, 2021

Abstract: Low drop-out (LDO) voltage regulators are widely used in many portable applications, such as cell phones, laptops, and earplugs. However, the design and simulation of the circuit takes a long time. Hence, the behavioral model for an LDO regulator is greatly needed. In this paper, the hybrid modeling method has been illustrated for generating a behavioral model for an LDO regulator circuit. The hybrid modeling method integrates a table-based modeling method and a circuit simplification method for generating an LDO linear regulator behavioral model. Based on research of the automation topologies and algorithms a behavioral model generation tool for an LDO linear regulator, LDOCad, is also described, including the software architecture and different modules of LDOCad. This tool provides the modeler, or circuit designer, with an approach to quickly and automatically generate a behavioral model for an LDO linear regulator. When the netlist processing algorithm, the table generation algorithm, and the modeled node and topology device extraction algorithm are executed, the hybrid model can be generated through the model topology formulator. The algorithms are generic and robust. The models tested demonstrate an accurate match with the performance of the original circuits and achieve from a 5 to 19 times speed improvement in simulations.

Keywords: LDO Regulator, Modeling, Automation

1. Introduction

With the advancement in silicon technology, it is possible to design a circuit with significantly large gates and high clock frequencies. Hence, it takes a lot longer to verify the large and complex design if the transistor level simulation is performed. One of the most common methods for solving his problem is to replace the large and/or complex blocks with behavioral models that describe the input and output relationship of the functional block [1]. Then the behavioral model of the system is simulated for verification and evaluation purpose. Often the simulation of the behavioral model of the circuit is much faster than the original design, which in fact decreases the design cycle of a mixed-signal system significantly [2].

There are two difficulties with the above methodology. The first one is the identification and division of the actual mixed-signal system. The research on analog circuit identification has been done for decade, but even so, not much progress has been made. Also, it is very challenging to model the wide variety of mixed-signal blocks that are available in today's circuits. One of the most popular method for creating behavioral models is manual abstraction and synthesis with many modeling techniques that are available currently, such as symbolic modeling techniques [3], numeric model order reduction techniques [4], root localization [5], table lookup method [6], circuit simplification and template modeling [7].

The hybrid modeling method joins the table lookup method and the feedback topology of an LDO linear regulator. Several algorithms are developed for generating the hybrid model automatically starting from the LDO regulator circuit netlist. Based on the research of the automation topologies and algorithms, the LDOCad is designed. This tool now provides the modeler or circuit designer with an approach to quickly and automatically

generate a behavioral model for an LDO linear regulator.

2. LDO Linear Regulator

"The voltage regulator is an electronic device that provides a specified and well-regulated voltage for powering other electronic devices" [8]. LDO regulators are composed of four basic components: a voltage reference, an error amplifier, a pass element, and a feedback network. Figure 1 shows the basic block diagram of an LDO regulator [9, 10].

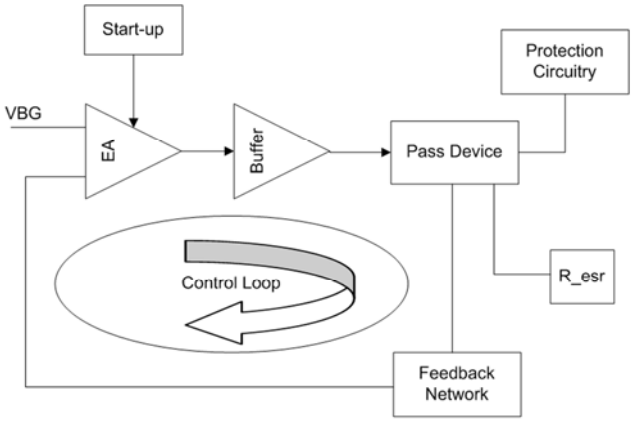

Figure 1. *Basic Block Diagram of an LDO Regulator.*

3. Model of the LDO Linear Voltage Regulator

For the behavioral modeling of an LDO linear regulator, a new behavioral modeling method is created, the hybrid behavioral modeling. The hybrid behavioral model joins the table based modeling method and the circuit simplification method. Both tables and the core part of the circuit are used in the model, which is the reason why it is named the hybrid modeling method. The table data is generated through Spectre [11] DC simulation. The topology devices in the model are extracted by the algorithm developed for this research, which will be illustrated later in this paper.

The hybrid modeling method joins the advantages of both the circuit simplification method and the table based method, with the good static and dynamic simulation of the simplified circuit and the simplicity of the table. At the same time, this method avoids the disadvantages of both the circuit simplification method and the table based method. This method does not need much circuit identification and also has the dynamic performance which is hard to get using the table based method only. Figure 2 shows the hybrid model topology of one of the LDO regulator circuits under research.

Compared with the LDO regulator circuit, the error amplifier is removed for the hybrid model. The more complex the structure is, the more difficult it is to identify. Hence, it is relatively easy to identify the devices that are in the hybrid model. Also, it reduces the simulation time. At the same time, the simulation results of the model without the error amplifier are just as good as the error amplifier available. The simulation speed of the hybrid model is up to 5 to 19 times, compared with that of the original circuit with the same test benches. The hybrid model shows very good line regulation, load regulation, transient line regulation, and transient load regulation. All the simulations are within 5% error, which meets the requirement for a behavioral model of an analog circuit. The hybrid model is composed of table based biases and the feedback stage. The voltage bias in the model is obtained by running DC simulation of the original circuit. Therefore, it is relatively easy to establish the table.

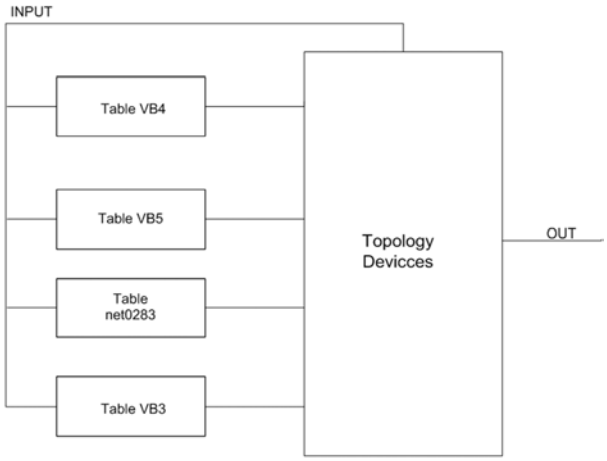

Figure 2. *Hybrid Model Topology of an LDO Regulator Circuit.*

4. Model Automation Methodologies and Algorithms

In order to automate the modeling of the LDO linear regulator, some model automation methodologies have been created and the corresponding algorithms are implemented with Python [12] programming language. These algorithms include Signal Path Tracing (SPT) [13] and Feedback Path Tracing (FPT), Modeled Node and Topology Device Extraction, Table Generation and Model Topology Formulation, as shown in Figure 3. Figure 3 is the flow chart for LDO circuit hybrid model automation.

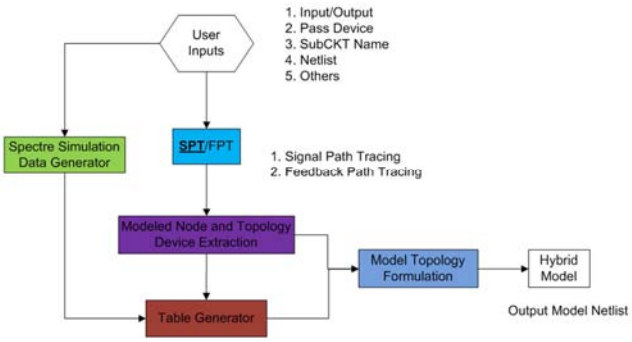

Figure 3. *Flow Chart for LDO Hybrid Model Automation.*

The function description of each algorithm is listed below:

1) Spectre Simulation Data Generator: this module is used to control simulation for extracting tables.

2) Modeled Node Extraction and Topology Device Extraction: this module is used to extract topology devices and modeled nodes. The topology devices are the devices that are extracted from the original netlist to be left in the model, which make up the primary feedback loop in the model. The modeled nodes are the break points between the tables and the primary feedback loop. Tables are used to model the characteristics of the secondary nodes.
3) Table Generator: this module is used to encapsulate tables into a sub-model.
4) Model Topology Formulation: this module is used to combine the information extracted above, form the model topology and generate model code.

The most critical module is the Modeled Node and Topology Device Extraction algorithm, which is illustrated in Figure 4. The flow chart shows the procedure for identifying the modeled nodes and the devices. For resistors, BJTs, capacitors, diodes and MOSFETs, a similar way is used for extracting the modeled nodes and topology devices. Only the MOSFET extraction procedures, therefore, are shown in the Figure 4. Signal path segments (signal transmission direction: start -> end) of a MOSFET device could be: G->D, G->S, S->D, D->S.

The primary nodes are the nodes in the primary feedback loop specified by the user through the GUI of LDOCad. The devices connected directly to the primary nodes are considered as the key devices from the aspect of modeling of the circuit. If a MOSFET device among them satisfies conditions below, it should be kept as a topology device.

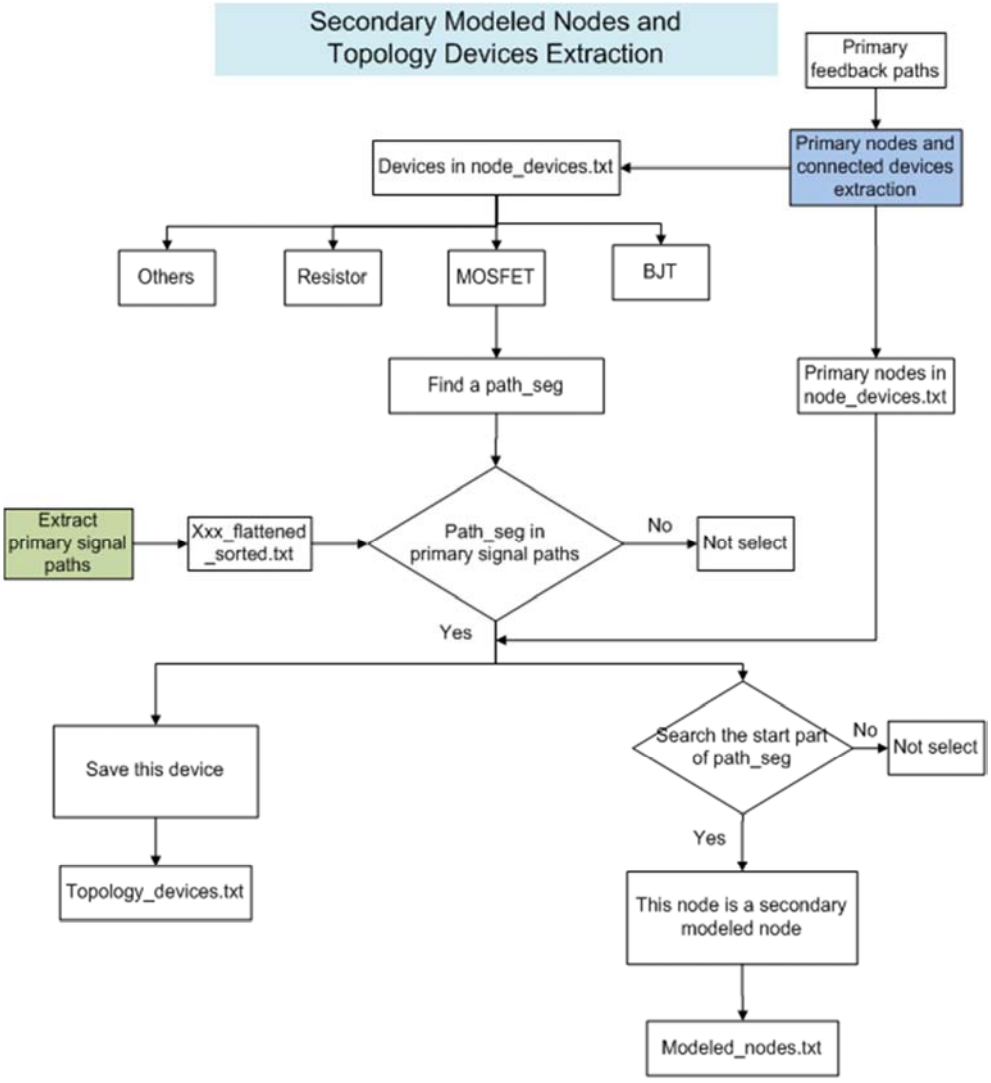

Figure 4. *Flow Chart for Topology Device and Modeled Node Extraction Algorithm.*

1) A signal path segment (end port of the device connected to a primary node) is available;
2) The signal path segment can be found in primary signal paths.

The following illustrates how the secondary modeled nodes are extracted:

1) Terminals connected to primary nodes are end ports;
2) From an end port search the start port according to the signal transmission direction [14]. The node corresponding to the start part of signal path segment is the secondary modeled node;
3) For example, if a device is MOSFET, its drain

connected to a primary node, then the drain is considered as the end port from the view of the signal transmission. Therefore the start ports of the signal transmission are the gate or the source. There two signal path segments (G->D, S->D) associated with this device. If the segments can be found in the primary signal paths, then the corresponding node of the start part of the segment is considered as a secondary modeled node.

5. LDO Software Design

A modular approach is used in the software design of the LDOCad tool. One algorithm is implemented with the Python programming language and tested, and then the next algorithm is designed, implemented and tested. Hence, each of the algorithms is able to function independently.

5.1. Software Architecture

The data flow diagram in Figure 5 shows the detailed software architecture of LDOCad by describing the communication of its components and the data files.

First, the user needs to input an experiment name and browse for the netlist, for which a model needs to be created. Once the netlist is browsed, the sub-circuit names are available for the user to choose. The node extraction module is used to generate input and output nodes of the circuit, and the gate, the source and the drain of the pass device. Next, the feedback extraction module uses the former user inputs to extract all the feedback paths for the user to specify the primary feedback path that determines the feedback topology of the hybrid model. Then the Spectre simulator is called to run the DC simulation for generating the dc.dc file, which is used by the Table Generation module to generate 1-D tables. Meanwhile, the topology formulator module is called to generate the hybrid behavioral model netlist.

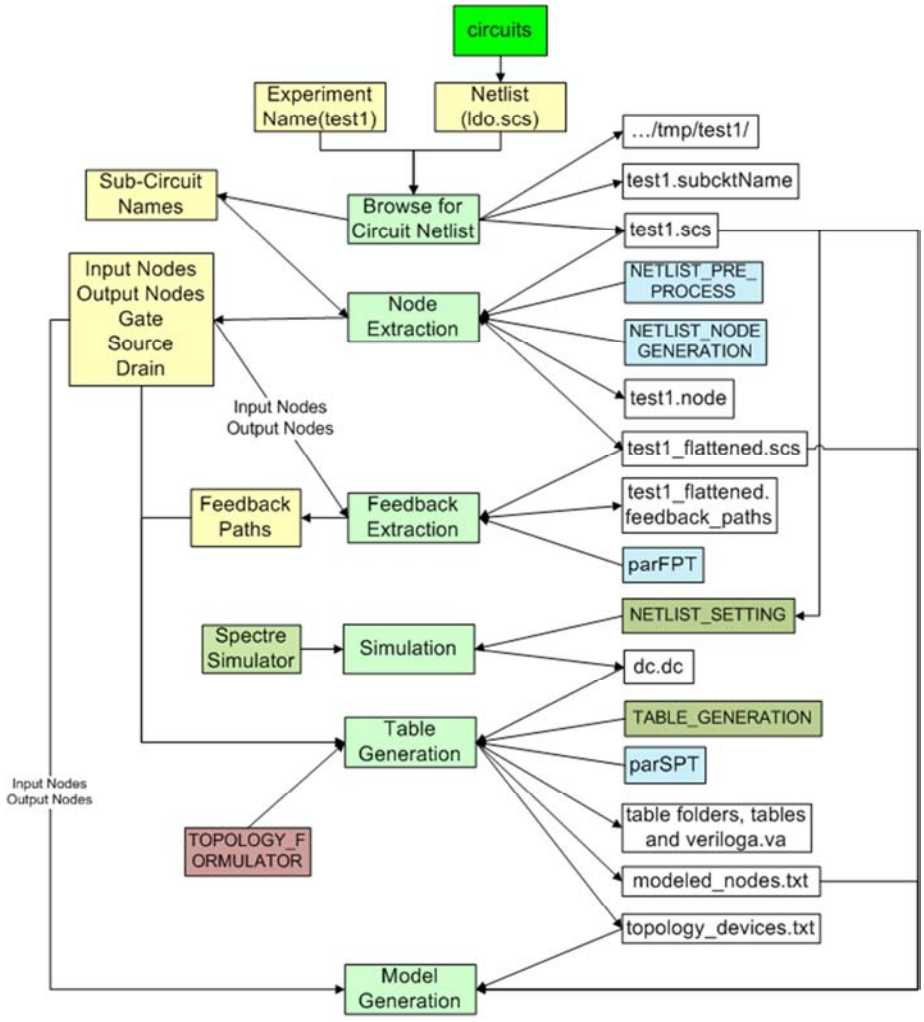

Figure 5. *Detailed Software Architecture of LDOCad.*

5.2. GUI Design

Figure 6 shows how each page of the GUI of LDOCad is connected. The data generated in the left big rectangle will be used when different buttons in the right big rectangle are clicked.

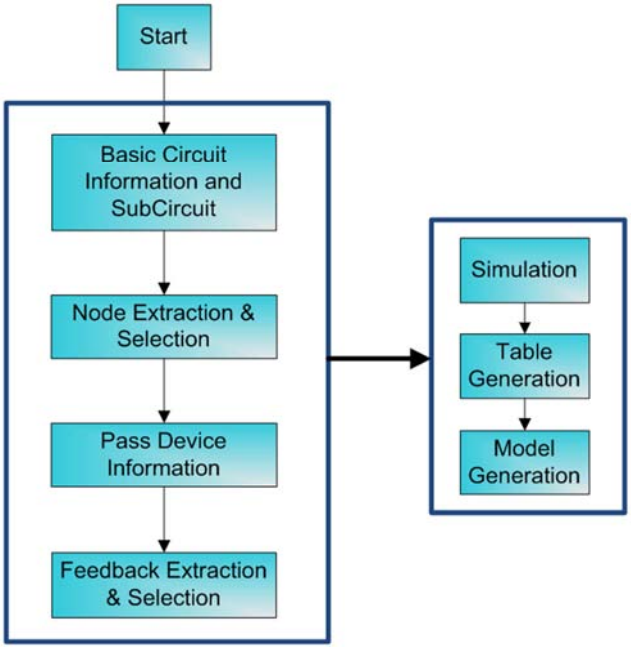

Figure 6. *LDOCad GUI Window Connection.*

The first window, shown in Figure 7, asks the user to give an experiment name. A folder with the experiment name will be created under the "tmp" folder, which stores all the data generated by LDOCad. Also, the user need to browser for an LDO netlist that he/she wants to make a behavioral model for. Once the netlist is selected, the sub-circuit names of the circuit will be displayed in the sub-circuit name list box. The user needs to select the sub-circuit name that is required to be flattened.

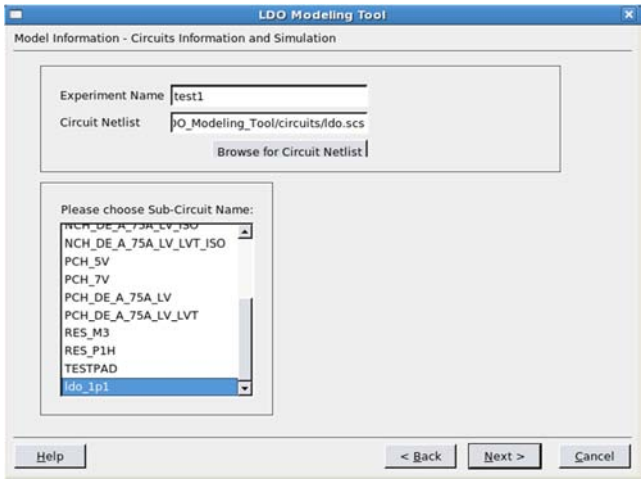

Figure 7. *First Window of LDOCad.*

In Figure 8, there is a Node Extraction button, which is used for extracting all the circuit nodes. After clicking it, all the nodes of the circuit will be displayed on this window and the next window shown in Figure 9. The user needs to select the input node (s) and the output node (s) based on the understanding of the circuit.

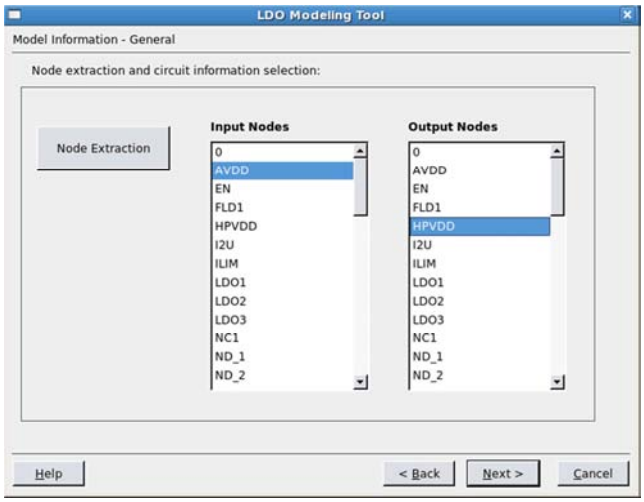

Figure 8. *Input/Output Selection Window.*

The window in Figure 9 is used for users to select the three terminals of the pass device of the LDO regulator. The selected information is used for extracting feedback paths.

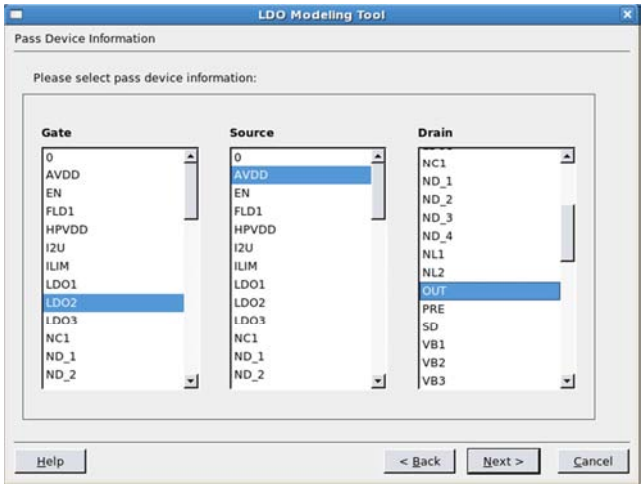

Figure 9. *Pass Device Terminal Selection.*

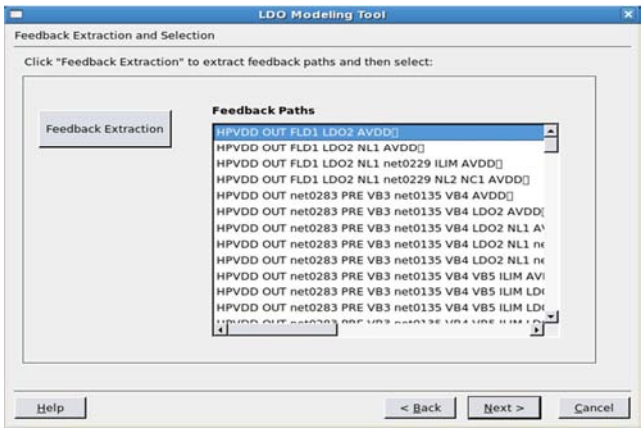

Figure 10. *Feedback Extraction and Selection.*

The Feedback Extraction button on this window, shown in Figure 10, is for extracting all the feedback paths of the circuit. Once it is clicked, all the feedback paths will appear

in the list box on the right for users to choose. The user needs to specify one feedback as the primary feedback path based on the study of the original circuit.

Figure 11 shows the last GUI window of LDOCad. The user needs to click the three buttons one after another because the data generated by clicking the "Simulation" button will be used when the later two buttons are clicked and the data generated by clicking the "Table Generation" button are used by running "Model Generation".

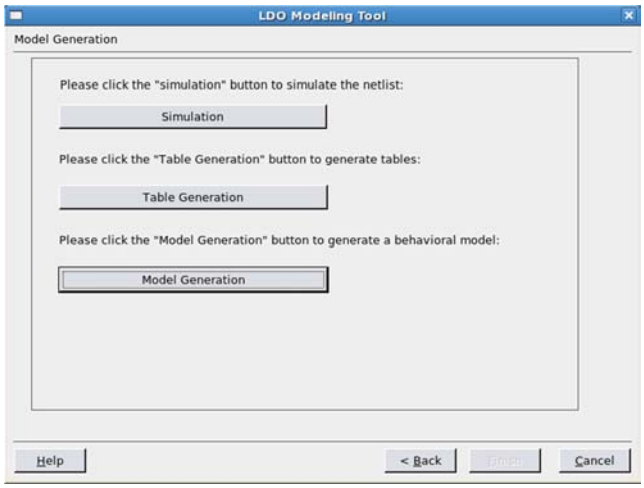

Figure 11. *Model Generation Window.*

6. Model Test and Verification

The test and verification of the behavioral model generated by LDOCad are illustrated in this section. An LDO linear regulator circuit from Texas Instruments (TI) has been used as an example.

6.1. Model Code

The model file is composed of four parts, which are device models, table instances, topology devices, and the Verilog-A table inclusion, as shown in Figure 12. This model is packaged as a symbol with three terminals, which are "AVDD", "HPVDD", and "EN". The user can use this symbol directly when a test bench is created for simulation.

6.2. Model Verification

To verify the behavioral model of the LDO regulator, simulations for both the original circuit and the model should be compared.

6.2.1. DC Sweep Analysis

For the DC sweep analysis of the LDO linear regulator, the line regulation and the load regulation performances are often compared between the model and the circuit from the stand point of behavioral modeling for an LDO regulator. The line regulation is shown in Figure 13 and the load regulation is shown in Figure 14. The solid curve represents the circuit simulation result, and the dot curve represents the model simulation result. The line regulation of the model fits the circuit very well, with very little error before the output voltage is regulated. The load regulation curves of the model and those of the circuit overlap.

```
...
device models for the following: NCH_7V_ISO, NCH_5V_ISO, PCH_DE_A_
75A_LV, PCH_7V, PCH_5V, RES_M3
...
// Library name: LDO_Hybrid_Model
// Cell name: Hybrid_Model
// View name: schematic
subckt Hybrid_Model AVDD HPVDD EN
I1 (AVDD VB3 GND) VB3
I2 (AVDD net0283 GND) net0283
I3 (AVDD VB5 GND) VB5
I4 (AVDD VB4 GND) VB4
I5 (AVDD NL1 GND) NL1
MN_MIR5_ldo_1p1 (0 FLD1 VB3 0) NCH_7V_ISO m=1 _par0="NHVTA_...
MN_MIR6_ldo_1p1 (0 FLD1 VB3 0) NCH_7V_ISO m=1 _par0="NHVTA_...
MP5_ldo_1p1 (OUT FLD1 net0283 OUT 0) PCH_7V m=1 _par0="PHVTA_...
MN_CC3_ldo_1p1 (0 LDO2 VB5 FLD1) NCH_5V_ISO m=1 _par0="NHVTA_...
MP_MIR4_ldo_1p1 (AVDD LDO2 VB4 AVDD 0) PCH_DE_A_75A_LV m=1...
MP_SD3_ldo_1p1 (AVDD LDO2 EN AVDD 0) PCH_7V m=1 _par0="PHVTA_...
MP_SD4_ldo_1p1 (AVDD LDO2 EN AVDD 0) PCH_7V m=1 _par0="PHVTA_...
MP_OUT_ldo_1p1 (AVDD OUT LDO2 AVDD 0) PCH_5V m=1 _par0="PHVTA_...
MP_SEN_ldo_1p1 (AVDD OUT LDO2 NL1 0) PCH_5V m=1 _par0="PHVTA_...
Rs1_ldo_1p1 (0 HPVDD OUT) RES_M3 _par0="RES_M3_LBC7" bodyR=0.500...
ahdl_include "VB3/veriloga.va"
ahdl_include "net0283/veriloga.va"
ahdl_include "VB5/veriloga.va"
ahdl_include "VB4/veriloga.va"
ahdl_include "NL1/veriloga.va"
ends Hybrid_Model
// End of subcircuit definition.
```

Figure 12. *Model Code of Example LDO Regulator.*

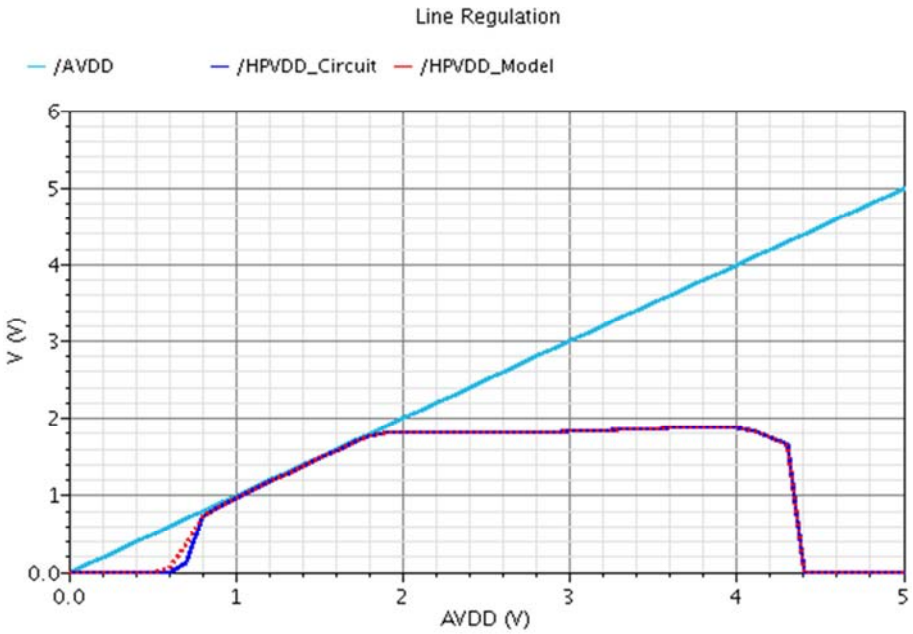

Figure 13. *Line Regulation.*

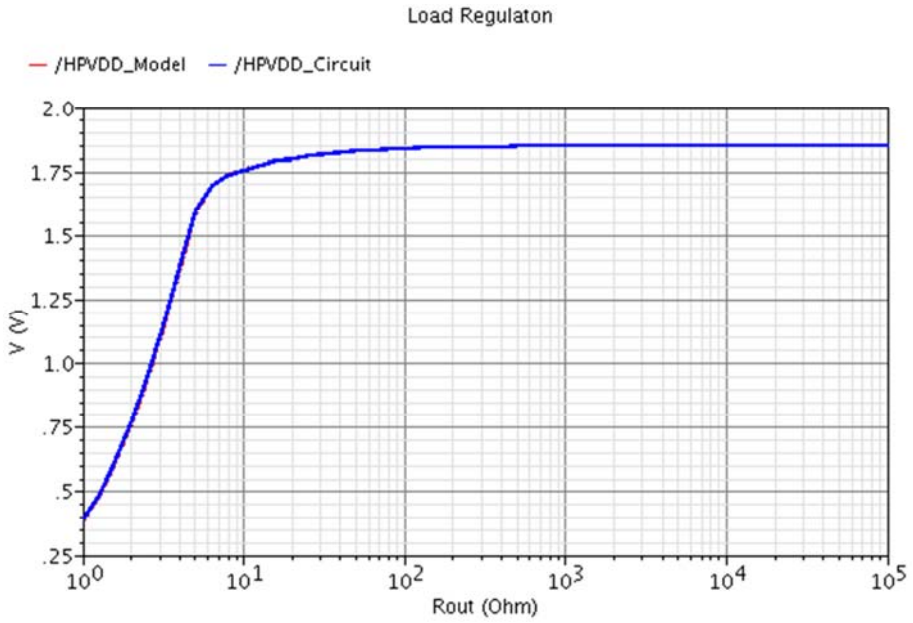

Figure 14. *Load Regulation.*

6.2.2. Transient Analysis

For the transient response of the LDO linear regulator, often the line transient regulation and the load transient regulation are tested for an LDO behavioral model. For the line transient response, as shown in Figure 15, the input changes from 2.8 V to 3.3 V. The output changes from about 1.81 V to about 1.84 V, correspondingly. Not only does the line transient response curve fit that of the circuit, but also the oscillation peaks fit very well.

For the load transient response of the LDO regulator, the output resistance changes from 500 Ohms to 250 Ohms. The output voltage of the model changes from about 1.848 V to about 1.846 V, which fits the performance of the circuit well, shown in Figure 16.

6.2.3. Simulation Speed Comparison

The simulation speed of a behavioral model is another important parameter for the evaluation of the model. For the system level verification of a circuit design, it is crucial to have a relatively fast behavioral model, since that's one of the most important applications of the behavioral model – to decrease simulation time. For some big and complicated designs, the simulation of the system may take days. Hence, to shorten the simulation time to a reasonable amount of time, such as a few hours, the circuit designer benefits a lot and it makes their work more efficient [15].

The hybrid model of the LDO linear regulator has a much

faster simulation speed compared with that of the original circuit. As shown in Table 1, the model speeds up at least 5 times that of the circuit among all the simulations.

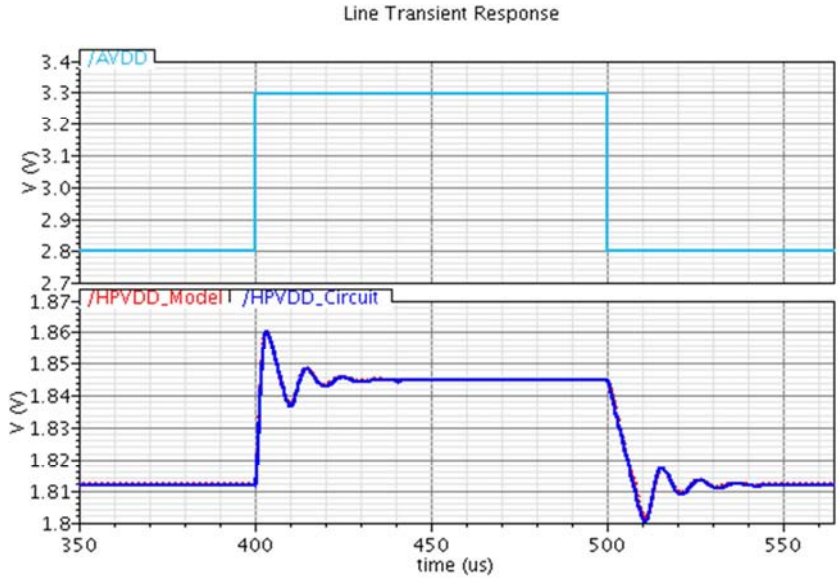

Figure 15. *Line Transient Regulation.*

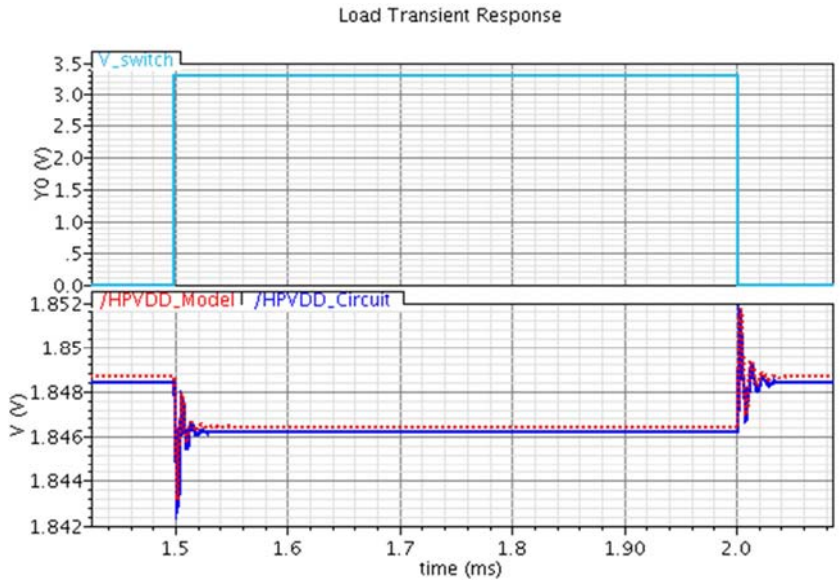

Figure 16. *Load Transient Regulation.*

Table 1. *Simulation Speed Comparison.*

Test Bench	Simulation Time for the Circuit	Simulation Time for the Hybrid Model	Speedup
Line Regulation	250 ms	30 ms	8.3 X
Load Regulation	280 ms	30 ms	9.3 X
Line Transient Regulation	1.17 s	220 ms	5.3 X
Load Transient Regulation	550 ms	100 ms	5.5 X
AC Analysis	170 ms	20 ms	8.5 X

7. Conclusions

As a behavioral modeling tool, LDOCad can help modelers to generate a behavioral model in a very effective way. The whole process is highly automated and convenient to use. With basic knowledge of the LDO linear regulator, the user can use the tool to generate a hybrid model. The Spectre simulator has been integrated into the LDO Modeling Tool to support Verilog-A and generate 1-D tables.

Modeling approaches, modeling automation methodology and algorithms, software design and implementation, and examples for demonstrating LDOCad are included in the paper. The hybrid modeling method is adopted to model the

LDO linear regulator because of its superiority in terms of model simulation speed and feasibility of automation among all the modeling approaches. Several concepts and algorithms are used or developed to automate the modeling process. These algorithms include netlist processing, signal/feedback path tracing, modeled node and topology device extraction, table generation, and model topology formulation, etc. With all these algorithms executed, topology devices together with tables can be obtained and integrated automatically. This makes up the hybrid model for an LDO regulator.

The LDOCad tool for generating a hybrid model for an LDO regulator has been successfully designed and implemented. It provides the platform for the user to enter circuit information for generating a hybrid model. LDOCad development includes the software architecture design and the GUI design. With the advent of LDOCad, the behavioral model of an LDO linear regulation can be automated. Cost and time saving are enormous for generating an LDO regulator model with the help of this tool.

Example circuits have been used to demonstrate the LDOCad tool. The hybrid models generated by the tool are also tested and compared with the modeled circuits. The generated models meet the behavioral model requirements for an LDO regulator. The models represent the most important electrical performances of the LDO regulators. At the same time, they have fast simulation speed and good accuracy.

Acknowledgements

I would like to express my thanks to all of the people who supported me on the research. Foremost in this list is my advisor Dr. Alan Mantooth. I would like to express my gratitude for his comprehensive advice, immense support, and incredible patience throughout my Master's study. My thanks are also extended to all of my group members including Yongfeng Feng, Rui Mao, Naveed Hingora, Zihao Gong and Brett Shook, especially Yongfeng Feng, for their great help and valuable discussions which contributed to this thesis.

References

[1] Y. Feng and H. A. Mantooth, "Algorithms for automatic model topology formulation," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 28, pp. 502-515, 2009.

[2] Ashok Bindra; Alan Mantooth, "Modern Tool Limitations in Design Automation: Advancing Automation in Design Tools is Gathering Momentum," in IEEE Power Electronics Magazine Volume 6, Issue 1. 2019, pp. 28-33.

[3] C. Borchers, "Symbolic behavioral model generation of nonlinear analog circuits," Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, vol. 45, pp. 1362-1371, 2002.

[4] H. Mantooth, L. Ren, X. Huang, Y. Feng and W. Zheng, "A survey of bottom-up behavioral modeling methods for analog circuits," in Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on, 2003.

[5] X. Huang, C. S. Gathercole and H. A. Mantooth, "Modeling nonlinear dynamics in analog circuits via root localization," IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, vol. 22, pp. 895, 2003.

[6] G. Odasso and E. Macii, "A table-based macromodel for behavioral delay estimation," in Signals, Systems, and Computers, 1999. Conference Record of the Thirty-Third Asilomar Conference on, 2002, pp. 772-774.

[7] D. Enright and R. Mack, "A high-level approach to modeling nonlinear analog architectures," in Circuits and Systems, 1997. Proceedings of the 40th Midwest Symposium on, 2002, pp. 1306-1309.

[8] P. Y. Kuo, D. Zhou and Z. M. Lin, "A low-dropout regulator with low ESR, low line regulation and high currency efficiency using low output-resistance voltage buffer," in Electron Devices and Solid-State Circuits, 2007. EDSSC 2007. IEEE Conference on, 2008, pp. 473-476.

[9] K. N. Leung and P. K. T. Mok, "A capacitor-free CMOS low-dropout regulator with damping-factor-control frequency compensation," Solid-State Circuits, IEEE Journal of, vol. 38, pp. 1691-1702, 2003.

[10] Yang Hao, Huang Shengming and Duan Quanzhen, "Low-power LDO Design of High-efficiency Class AB OTA Based on Adaptive Biasing," in Journal of Physics: Conference Series Volume 1754, Issue 1. 2021.

[11] Cadence® Virtuoso® Spectre® Circuit Simulator.

[12] Fred L. Drake, "Python Tutorial".

[13] H. A. Mantooth and P. E. Allen, "A higher level modeling procedure for analog integrated circuits," Analog Integr. Cir. Signal Proc., vol. 3, pp. 181-195, 1993.

[14] XiaoLing Huang, "Automatic Behavioral Model Generation of Nonlinear Analog Circuits," pp. 20, 2005.

[15] Simon Schmidt, Max Richter, Jens Oberrath and Paolo Mercorelli, "Control oriented modeling of DCDC converters," in IFAC-PapersOnLineVolume 51, Issue 2. 2018, pp. 331-336.