Practical Aspects of Script-Based Analog Design Using Precomputed Lookup Tables

Boris Murmann
Unievrsity of Hawaii
Department of Electrical and Computer Engineering
Honolulu, USA
bmurmann@hawaii.edu

2024 IEEE International Symposium on Circuits and Systems (ISCAS) | 979-8-3503-3099-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCAS58744.2024.10558027

***Abstract*— Ratio- and lookup-table based sizing methods were created to eliminate the need for SPICE-based iterative tweaking and to achieve a good match between theoretical analysis and simulated performance. Following such approaches can be gratifying for seasoned designers who have experienced the large mismatch between textbook hand calculations and simulation results firsthand. However, the circuit design novice may fall into the trap of tweaking sizing scripts iteratively and abandoning the connection to the analytical underpinnings of the target design. In this paper, we summarize the best practices for script-based analog circuit design toward the development of analog generators. These methods are illustrated using a two-stage amplifier as an example.**

Keywords*— *analog integrated circuits, design methodology, design automation, precomputed lookup tables

I. Introduction

Analog integrated circuit (IC) design is a highly sophisticated field that builds on decades of innovative work. However, due to its intricate tradeoffs and vast design space, it remains difficult to codify and automate it in the way this has been done for digital circuits [1]. This issue severely limits design productivity in an era of increasingly specialized, "long tail" IC developments that would strongly benefit from agile and re-use centric methodologies.

A promising direction for addressing this need is the ongoing work toward analog generators [2]–[4], which in their most basic form aim to capture the designer's intent and sizing steps through scripts that can be re-run for specification changes or technology porting. The transition toward generators is not only important for addressing immediate needs, but also lays the groundwork for future innovations. For instance, with a programmatic flow in place, it becomes easier to inject machine-learning-based methods, which may ultimately outperform humans [5]. Additionally, generator-based design improves reuse and reproducibility, which resonates with the new movement toward open-source chip design [6], [7].

This paper reviews the best practices for script-based analog circuit design using precomputed lookup tables [8]. The target audience includes newcomers who struggle to connect standard textbook material to such new methodologies, as well as researchers working on the development of analog circuit generators. The remaining sections are structured as follows. Section II reviews the motivation for lookup table-based design and explains which problems it solves (and which it does not). Section III introduces a design example (a two-stage amplifier) for further consideration and illustration of the recommended initial design steps. Section IV reviews some approaches for circuit optimization, while Section V covers ancillary considerations dealing with frequently asked questions. Finally, Section V ends this paper with concluding remarks.

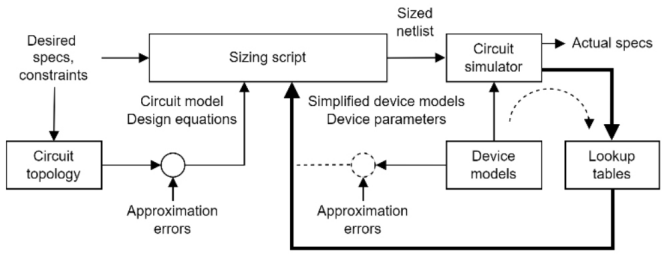

Fig. 1. Framework for script-based sizing using precomputed lookup tables.

II. Preliminaries

Fig. 1 illustrates the framework for systematic analog circuit design using a sizing script. At its core is a piece of software code (written, e.g., in Python or MATLAB), which links target specs, constraints and circuit topology to device sizes in a particular technology (represented by device models). An alternative flow may place the circuit simulator within an optimization loop. We exclude this option explicitly, as it is usually too time-consuming (even for small circuits).

As we shall discuss later, the sizing process via a programmatic script is generally non-trivial as it resembles an inverse problem: we need to find the design parameters that lead to the desired specs (the final output). There are also other, more elementary issues. The first is that the transistor model equations used for circuit simulation (e.g., BSIM, PSP) are too complex for evaluation within the sizing script. One possibility is to work with simplified analytical expressions that yield only small approximation errors across a wide range of gate biases. This is the philosophy behind the inversion coefficient approach by Enz et al. [9]. On the other hand, the method discussed here relies on precomputed lookup table data generated by the simulator to bypass the approximation issue altogether (bold line in Fig. 1). For a range of bias conditions and MOSFET channel lengths, small-and large-signal parameters (g_m, I_D, etc.) are tabulated once and subsequently used in all calculations. As discussed in [8], these data are often interpreted as ratios (such as g_m/I_D, g_m/C_{gs}, I_D/W, etc.) for normalized design, with g_m/I_D being the main knob that sets the transistor's inversion level.

A problem that is not solved by any existing script-based design methodology is that there are generally approximation

errors between the actual circuit behavior and the simplified equations used in the sizing script. This issue has become more noticeable with the introduction of methodologies that minimize errors due to device model approximations. Especially for inexperienced designers, it is sometimes difficult to tell why the simulation results do not match the design intent; is it due to circuit or device model simplifications? An oft-forgotten remedy for this problem is to decouple the two issues. In other words, the circuit should first be understood and validated using the simplest possible simulation model. The next section provides an example that illustrates the suggested approach.

III. Circuit Example

A relatively basic, yet non-trivial design problem is a two-stage amplifier as shown in Fig. 2. Especially for inexperienced designers, it is difficult (and not recommended) to immediately jump into transistor-level SPICE simulations and optimizations with this schematic. Dealing with the common-mode feedback, creating the bias potentials, and ensuring that all devices operate in saturation distracts from understanding the essential tradeoffs and developing the required intuition. A better approach is to begin with a high-level circuit model (shown in Fig. 3), establish the key design equations, and run initial SPICE simulations to assess their accuracy. For example, if we are interested in computing the phase margin (PM), we may be tempted to use the following first-order equations found in commonly used textbooks:

$$\omega_u \approx \beta \frac{g_{m1}}{C_c} \quad \omega_{p2} \approx \frac{g_{m2}}{C_2} \quad PM \approx 90° - \tan^{-1}\left(\frac{\omega_u}{\omega_{p2}}\right) \qquad (1)$$

Here, β is the circuit's feedback factor, ω_u is the feedback loop's unity gain frequency, ω_{p2} is the nondominant pole. Are these approximate expressions good enough for use in a sizing script? Especially in modern technologies with low intrinsic gain ($g_m r_o$) and large parasitic capacitances, these equations are often quite inaccurate. A few test simulations of the circuit in Fig. 3 with reasonable estimates can reveal this issue quickly and help us proceed with the following refinements (also available in common textbooks):

$$\omega_u \approx \frac{\beta g_{m1} R_1 g_{m2} R_2}{R_1(C_1 + C_c(1 + g_{m2}R_2)) + R_2(C_2 + C_c)}$$
$$\omega_{p2} \approx \frac{g_{m2}}{C_1} \frac{1}{1 + \frac{C_2}{C_c} + \frac{C_2}{C_1}} \qquad (2)$$

where R_1 and R_2 are the total stage 1 and 2 output resistances, and C_1 is the stage 1 load capacitance (see Fig. 3).

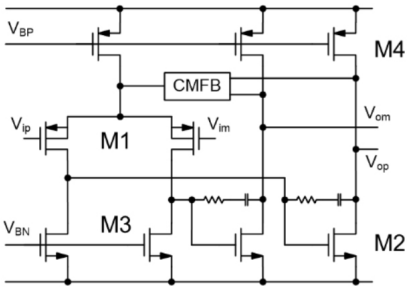

Fig. 2. Miller-compensated, fully differential two-stage amplifier.

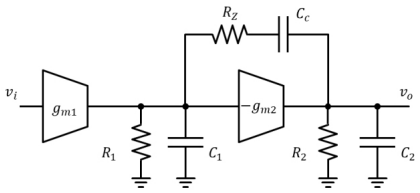

Fig. 3. High-level model of the circuit in Fig. 2. The g_m elements can be implemented using voltage-controlled current sources (VCCS).

Now that we have established confidence in the design equations using a high-level circuit model, we can proceed to the transistor level, where the first test should purposely focus on a non-optimal implementation. For example, it could be structured as follows:

- Pick a reasonable channel length and g_m/I_D for all transistors (e.g., twice the minimum L and 15 S/A).
- Pick reasonable g_m for the two stages and size the widths of all transistors using the selected g_m/I_D and the corresponding current density I_D/W from table lookup.
- Run a DC operating point analysis and record the device's small-signal parameters (all g_m, r_o, C_{gs}, C_{db}, etc.).
- Calculate the expected unity gain frequency and phase margin and compare with the simulated values from a SPICE AC simulation.

We performed these steps for a circuit in 180-nm CMOS and observed the numbers summarized in Table I. The main point here is to show that the refined equations (2) match the SPICE result closely, while (1) predicts a significantly higher phase margin. Such calibrations between the design equations and the simulated performance of an actual circuit are essential before jumping into the creation of a sizing script. The same approach can be followed for other circuits that can be described by an equation set. Typically, we want to find a low-complexity description that yields a "good enough" agreement with the simulator. With further progress in open-source IC design, one could imagine an openly shared library that has such equations worked out for a broad range of building blocks. In some cases, the equations can be generated automatically using tools like SLiCAP [10].

TABLE I. Numerical Example

	Calculated using (1)	***Calculated using (2)***	***SPICE simulation***
f_u (MHz)	99.72	93.98	94.05
PM (deg)	83.6	77.0	76.4

IV. Optimization

Once we have a calibrated setup between the design equations and SPICE benchmarking of the circuit in question, we can engage in optimizations, i.e., determine the optimum sizing that meets our performance criteria and constraints. The analog circuit design and design automation communities have explored an uncountable number of ways in which this may be pursued [1]. By far the worst option is to tweak circuit parameters nearly randomly and iteratively by hand, until a satisfactory outcome is achieved. This makes the design non-

portable to another process technology without repeating the tweaking process. Furthermore, potential users may not trust the circuit, since there is usually no documentation on how the final design point was reached.

A step in the right direction is to follow well-established baseline heuristics. For example, for the two-stage amplifier of the previous section, we know that C_c should be larger than approximately $3C_1$ and smaller than approximately $C_2/3$ [11]. These heuristics can be codified and can lead to a reasonably well-documented design. The disadvantage is that we may not achieve the absolute best performance, or the proper heuristics may not be known for all circuits of interest.

A more ambitious direction is to untangle the design equations such that the optimal sizing parameters can be determined from the specs using a low-dimensional sweep across a small number of "main design knobs." For illustrative purposes, we summarize here such a "knowledge-based" approach for two-stage amplifier design with capacitive feedback, as presented in [8]:

1. Begin by neglecting self-loading due extrinsic capacitances (e.g., junction capacitances); these are not known until the circuit is sized.
2. Set up a two-dimensional sweep across the major design knobs β/β_{max} (where β_{max} is the unloaded feedback factor) and C_2/C_c.
3. Compute the following for each sweep value:
   a. C_c using the total integrated noise specification. This also lets us calculate most other capacitances.
   b. g_{m1} according to the desired ω_u.
   c. $(g_m/I_D)_1$ given g_{m1} and C_{gg1} (total gate capacitance of the input device) via table lookup, quantifying the link between g_m/I_D and $\omega_T = g_m/C_{gg}$. This also defines I_{D1}.
   d. g_{m2} based on the desired ω_{p2}.
   e. $(g_m/I_D)_2$ given g_{m2} and C_1 via table lookup. This also defines I_{D2}.
   f. The total current $I_{D1} + I_{D2}$.
4. Pick the design point in the two-dimensional space that minimizes the total current.
5. Complete the sizing and compute the self-loading capacitances. If self-loading is significant, repeat all calculations from the beginning to "anneal" the design until convergence.

An example of a resulting contour plot is shown in Fig. 4. An advantage of this approach is that it points to an optimum design point with relatively low computational effort. A key challenge is to develop the above-shown design plan, which essentially tries to invert the system of design equations to a large extent, keeping sweeps to the lowest possible dimensionality. Such design plans may not exist for all circuits of interest. Also, anyone who re-uses the script may be steered again toward iterative tweaking, but this time with the script and its coding details, instead of the circuit itself.

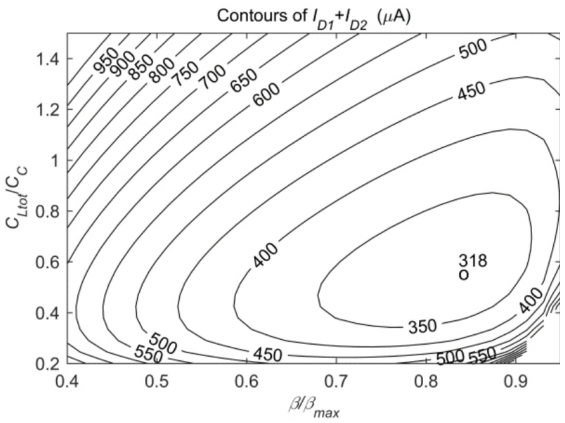

Fig. 4. Example of a contour plot generated by a sizing script that minimizes the current consumption of a two-stage amplifier with capacitive feedback.

The reason for blind script tweaking is often that it is relatively difficult to understand, much like the original circuit. However, no matter how the optimization is done, a savvy designer should always understand the circuit's main tradeoffs and be able to "hand-calculate" a design point that matches SPICE results. Whenever this connection breaks, simulation errors may be overlooked and innovations in circuit architecture based on intuitive understanding will no longer be possible.

Fig. 5 shows an alternative example from a folded-cascode amplifier design script [12] using "forward evaluation" instead of a knowledge-based design plan. The plot contains 10^6 design points that can be quickly computed using lookup table data. For every design point, the channel lengths and g_m/I_D of each matched transistor pair as well as the total bias current are randomly drawn from within reasonable intervals. Other parameters may be set as external constraints. The resulting plot gives a feel for the tradeoff space as well as the reachable performance metrics given external constraints and post-computation filters (for example, color coding the phase margin for each point). Overall, this method is similar to simulation-based approaches with "SPICE in the loop," except that it runs much faster since simulations are replaced by table lookups. Given the outcome shown in Fig. 5, some decision making (either manual or automatic) must still follow to select the final design point.

As of today, there is no widely established standard on how to implement systematic and broadly re-usable analog circuit design scripts, though the problem was believed to be solved in the 1980s [13]! In part this can be explained by the fact that analog designers work largely in isolated "silos" with little to no code sharing among them. There is hope that the emergence of the open-source circuit design ecosystem may improve this situation (see e.g., [14] for an innovative example). With open-source process design kits and open-source tools, design scripts and complete circuits can be shared and collaborated on via developer platforms like GitHub. In other fields, and most notably in machine learning and AI, this approach has led to explosive progress and innovation.

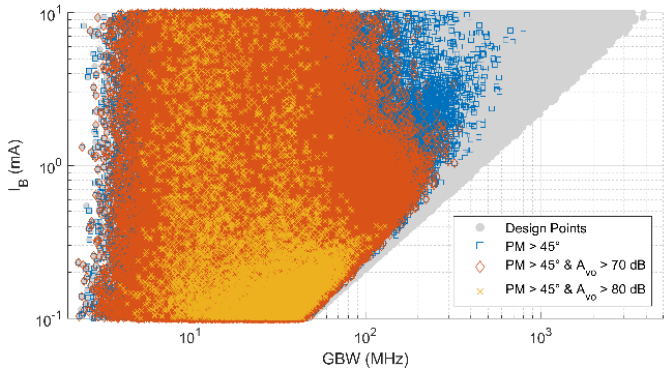

Fig. 5. Tradeoff space evaluation using forward evaluation (from [12]).

V. Ancillary Considerations

In this section, we summarize additional practical aspects that address frequently asked questions by newcomers in lookup table-based design.

A. Transistor model checks

In any circuit sizing methodology, the quality and correctness of the generated designs critically relies on accurate transistor models for circuit simulation. In the design approach discussed in this paper, these models are not only used for final performance verification, but also for the generation of precomputed lookup tables that drive the design optimization. Along with the initial "calibration" of the design equations (discussed in Section III), a designer should always check the transistor models for potential issues.

One common issue is the proper modeling of thermal noise. A MOSFET exhibits a thermal drain current noise power spectral density (PSD) of $4kT\gamma g_m$ where the parameter γ tends to be in the range of 1…2 for the latest short channel devices (depending on the gate bias). It is good practice to extract the value of γ for the models provided by the foundry. This can be done using a simple AC noise analysis and by dividing the observed PSD by $4kTg_m$. While we cannot know the exact value to expect, faulty models often show extreme outlier behavior, for example $\gamma = 0.2$ (which is physically impossible).

Another useful sanity check is to plot g_m/I_D and f_T (transit frequency) against V_{GS}. This can be easily done, for instance, from the precomputed lookup tables. The expected shape of both curves is well understood from device physics. g_m/I_D should show a weak inversion plateau and monotonically transition toward (but not exactly meet) a $\sim 2/(V_{GS}-V_t)$ asymptote. The device's f_T should show a maximum with subsequent roll-off in strong inversion. Despite the high level of complexity in today's device model templates, it is possible for foundry models to deviate from these basic features with poor parameter population.

B. Switches

The primary application of g_m/I_D-based design using lookup tables are class-A circuits that operate with a constant bias current. However, the small-signal data captured in lookup tables are also useful for sizing switches, as used for example in an analog-to-digital converter's track-and-hold circuits. The first order on-resistance can be found from the lookup tables as $1/g_{ds}$ at $V_{DS} = 0$ V (see Section 6.5 in [8]).

C. Dynamic circuits

An increasing fraction of modern circuit design has moved toward dynamic circuits that avoid constant bias currents and are often difficult to analyze. In many cases, however, it is possible to assess basic performance metrics from the bias and small-signal parameters captured in lookup tables. For example, the speed and noise of a dynamic comparator can be estimated using a small-signal model around its metastable point, in combination with a non-steady-state model for any integrating nodes in the circuit (see e.g., [15], [16]).

D. Distortion modeling

The transistor's inversion level and its proxy g_m/I_D are most frequently used in linear circuit analysis. However, it has been shown that both g_m/I_D [17] and the transistor's inversion coefficient [18] can also be used to predict harmonic and intermodulation distortion products (assuming that the small distortion approximation holds). It is therefore possible to incorporate distortion specs in sizing scripts for RF circuits and other applications that are constrained by nonlinear effects.

VI. Conclusion

This paper has provided a review of practical aspects for systematic circuit design with precomputed lookup tables. In light of the ongoing trend toward analog circuit generators, we must establish a solid bridge between the vast amount of existing knowledge in analog design and code-driven sizing scripts. Especially for the novice, it is important to begin the script development using simplified circuit models and test cases that align the underlying design equations with the SPICE benchmarking results. Additional work is needed to motivate today's designers to embrace reproducible script-based design flows. The trend toward open-source design with shared code will likely enable much-needed progress in this direction.

References

[1] A. F. Budak *et al.*, "CAD for analog/mixed-signal integrated circuits," *Advances in Semiconductor Technologies: Selected Topics Beyond Conventional CMOS*, pp. 43–60, Sep. 2022, doi: 10.1002/9781119869610.CH3.

[2] J. Crossley *et al.*, "BAG: A designer-oriented integrated framework for the development of AMS circuit generators," *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers*, pp. 74–81, 2013, doi: 10.1109/ICCAD.2013.6691100.

[3] M. Videnovic-Misic, "Programmatic analog IC design: MOSAIC bootcamp [Society News]," *IEEE Solid-State Circuits Magazine*, vol. 14, no. 4, pp. 65–68, Dec. 2022, doi: 10.1109/MSSC.2022.3205462.

[4] A. Hammoud, V. Shankar, R. Mains, M. Saligane, J. Matres, and T. Ansell, "OpenFASOC: An Open Platform Towards Analog and Mixed-Signal Automation and Acceleration of Chip Design," *International Symposium on Devices, Circuits and Systems, Conference Proceedings*, pp. 1–4, 2023, doi: 10.1109/ISDCS58735.2023.10153547.

[5] A. F. Budak, M. Gandara, W. Shi, D. Z. Pan, N. Sun, and B. Liu, "An Efficient Analog Circuit Sizing Method Based on Machine Learning Assisted Global Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 5, pp. 1209–1221, May 2022, doi: 10.1109/TCAD.2021.3081405.

[6] T. Ansell and M. Saligane, "The Missing Pieces of Open Design Enablement: A Recent History of Google Efforts," in *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers*, 2020, pp. 1–8.

[7] B. Murmann and M. Kassem, "Democratizing IC Design: The Story of a New Movement and the Launch of the SSCS PICO Program [Society News]," *IEEE Solid-State Circuits Magazine*, vol. 13, no. 4, pp. 123–130, Nov. 2021, doi: 10.1109/MSSC.2021.3111376.

[8] P. G. A. Jespers and B. Murmann, *Systematic Design of Analog CMOS Circuits: Using Pre-Computed Lookup Tables*. Cambridge University Press, 2017. doi: 10.1017/9781108125840.

[9] C. Enz, M. A. Chalkiadaki, and A. Mangla, "Low-power analog/RF circuit design based on the inversion coefficient," *Proc. European Solid-State Circuits Conference*, pp. 202–208, Oct. 2015, doi: 10.1109/ESSCIRC.2015.7313863.

[10] A. Montagne, "SLiCAP: Symbolic Linear Circuit Analysis Program." Accessed: Oct. 30, 2023. [Online]. Available: https://analog-electronics.tudelft.nl/SLiCAP.html

[11] W. M. C. Sansen, *Analog Design Essentials*. Springer US, 2006. doi: 10.1007/B135984.

[12] A. A. Youssef, B. Murmann, and H. Omran, "Analog IC Design Using Precomputed Lookup Tables: Challenges and Solutions," *IEEE Access*, vol. 8, pp. 134640–134652, 2020, doi: 10.1109/ACCESS.2020.3010875.

[13] L. R. Carley and R. A. Rutenbar, "How to automate analog IC designs," *IEEE Spectr*, vol. 25, no. 8, pp. 26–30, 1988, doi: 10.1109/6.7160.

[14] Z. Li and A. Chan Carusone, "Design and Optimization of Low-Dropout Voltage Regulator Using Relational Graph Neural Network and Reinforcement Learning in Open-Source SKY130 Process," in *IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2023, pp. 1–9.

[15] B. Razavi, "The StrongARM latch [A Circuit for All Seasons]," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12–17, Mar. 2015, doi: 10.1109/MSSC.2015.2418155.

[16] D. Bankman, "Mixed-Signal Processing for Machine Learning | Stanford Digital Repository," Stanford University, 2019. Accessed: Jun. 30, 2020. [Online]. Available: https://purl.stanford.edu/tt451tg9318

[17] P. G. A. Jespers and B. Murmann, "Calculation of MOSFET distortion using the transconductance-to-current ratio (gm/ID)," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, May 2015, pp. 529–532. doi: 10.1109/ISCAS.2015.7168687.

[18] F. Chicco, A. Pezzotta, and C. C. Enz, "Charge-Based Distortion Analysis of Nanoscale MOSFETs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 2, pp. 453–462, Feb. 2019, doi: 10.1109/TCSI.2018.2868387.