Estimating the Impact of Methodology on Analog Integrated Circuit Design Time

Panu Jalas and Timo Rahkonen
University of Oulu

Editor's note:
The paper discusses analog design practices and proposes a project management model for studying which analog design methodology will achieve the fastest time-to-market given the probability of design errors.
—Haralampos-G. Stratigopoulos, Sorbonne Universités

FOR ANY INTEGRATED circuit (IC) product design project, the purpose of a design methodology is to optimize the time-to-market (TTM) from a specification to a flawless design so that the product will fulfill the business requirements regarding performance, manufacturing cost, and testability. Yet, in the previous literature for the case of analog IC products, there has been little quantitative analysis of whether and when the application of different methodological principles really starts to improve the TTM performance of a design project. Most of the discussion has been qualitative in nature [1]–[3] and is often led by the makers of electronic design automation (EDA) tools [4]–[6] or is based on collecting experimental data from different design projects [7]. 1:14 ratios have been reported depending on the chosen design methodology [8]. However, detailed data of the observed design times and the details of the methodologies are usually not published. Additionally, the different types of product designs and device technologies of different companies are difficult to compare with each other.

Kundert [4], [6] notes that a design methodology requires investment. A top-down approach means investing extra time upfront, before the actual transistor design begins. Applying any design methodological principles means investment in more rigorous documentation and management of the design flow. Such investments are assumed to pay off if the designs have a reasonable amount of system-level complexity. Yet, for simple designs with a low transistor count and not much circuit complexity, investing a designer's time in tasks required by the principles of a methodology can be considered unnecessary. We discuss also other practical reasons, why analog design projects are sometimes executed even without emphasis on top-down approach.

We will present an Excel-based numerical model that estimates the effects of design errors in various phases of the design. Three different examples of varying complexity are compared.

Characteristics of analog IC design methodologies

The benefits of hierarchical and top-down design methodologies are generally acknowledged for the design of any digital circuit products because, with

Digital Object Identifier 10.1109/MDAT.2015.2497331
Date of publication: XX XXXX XXXX; date of current version: XX XXXX XXXX

Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC

digital design, the main challenge is usually to manage the complexity of the system rather than to manage the characteristics of individual devices or analog behavior of the elementary design cells [8]. However, despite the fact that the benefits of top-down approach for analog or analog-heavy mixed signal design projects are well known, in practice, the choice of the optimal methodology is not so obvious. The system complexity of an analog product may be perceived to be simple, and the design challenges are considered to be with the electrical characteristics of the lowest level functional blocks. In addition, the arguments for favoring bottom-up approach might include the following.

Low system complexity

The product consists of only a handful of functional blocks that cannot be reduced to smaller parts for verification. The interfaces of the blocks are assumed to be well known, and the design management estimates that there is little chance for system-level errors, especially if there are no feedback loops between the functional blocks. Putting extra effort into creating a vertical hierarchy just "for the sake of design methodology" may be perceived to be unnecessary and time consuming.

Tight sampling schedule

In the industry, the project timelines are usually very tight, especially if the customer expects to get "reasonably well functioning" engineering samples very quickly but can wait to get flawless production-ready circuits later. However, rushing to the tape-out by omitting sufficient amount of system level verifications of the initial version may backfire as nondetected system level errors.

Analog designs are always unique

There are many principles of digital design methodologies that cannot be directly applied to analog design, such as regularity of the design cells, reusable standard cell libraries, or rigorously implemented design flows that would allow the automation of physical implementation [8]. Unlike with digital design, the performance and electrical characteristics of analog circuit cells are often affected by the environment in which they are embedded and the available analog design blocks are seldom directly reusable as such [9]. Existing analog design blocks could even be interpreted as reference designs, usually needing some modification when used in a new environment. At minimum, all previously designed analog circuit cells need to be fully verified in the new environment.

However, there are also other "softer" hypothetical arguments that could cause a real-life design project to be executed in a bottom-up or nonhierarchical manner.

Analog design and system design are separated from each other

Behavioral modeling as a system-level verification tool is still a relatively new technique in analog design and has been in wider use for only ten years [10]. The system modeling is often performed by a separate system engineer who may work outside of the actual design team. This separation could lead to disconnections in the design flow so that the model level specifications are not properly propagated down to block and transistor levels.

Bottom-up problem solving and innovation

The challenges and innovations in analog design are often in the details of the individual bottom-level (transistor-level) circuit design cells. This makes analog design teams often attempt to solve the lowest level problems first, which in turn drives the execution of the entire design project to become bottom-up driven.

Differences in design cultures

The application of a specific design methodology usually requires the designers to put extra effort into communication, review practices, documentation, and reporting. However, in many cases, the individual designers and design teams come from different companies and backgrounds. Some analog designers may have experience using a heavily digital design-flavored VLSI methodology, whereas some analog designers may have a strictly analog circuit cell background. Even within the same company, designers may become reorganized to work on different types of products where different aspects of methodology are emphasized. Managing the communication and documentation cultures requires active guidance from the design managers so that the appropriate design methodology gets implemented in practice.

The design methodology will typically consist of guidelines on how the design work should be organized. The most obvious methodological principles concerning analog IC products would be as follows.

Hierarchy

Hierarchical design means dividing the system into subsystems and dividing the subsystems further to lower level sub-blocks so many times that the complexity of the blocks is comprehensible in every detail. In digital or software design, this means that every detail can be comprehensively verified on a virtual test bench [1], [8]. For analog design circuits, we propose that instead of comprehensibility, it is more appropriate to discuss observability or verifiability. The lowest level of hierarchical partitioning for analog circuit cells would be the level where all the electrical characteristics of the analog circuit block can be reliably verified by simulation so that any change in schematics or layout design can be traced from the verification results.

Top-down approach

A top-down approach means that once the design is partitioned to subsystems and blocks hierarchically, the project starts from an architecture design of the top level using behavioral models of the subsystems and then advances to the lower levels of subsystems, blocks and, eventually, transistor-level design. A top-down-based approach to analog and mixed signal design has been the leading principle for platform-based design philosophy [1], which emphasizes starting the design process with system-level analysis.

Design flow, documentation, review practices, reusability, quality improvement, and organization

The design methodology consists of practical guidelines. The design flow describes how to input the circuit schematics, verify the operation by conducting an overall simulation of the functional and process corner conditions, perform the layout design, and finally, verify the physical implementation including the parasitics, so that the circuit block will eventually work as it is intended to.

Other important aspects of a design methodology could include things like guidelines for design review practices, documentation, identifying and estimating risks, and collecting the learnings for continuous development and steps for promoting innovation and managing the new intellectual property (IP) created during the design project.

A model to estimate the design time for bottom-up and top-down methodologies

We performed a theoretical comparison between top-down versus bottom-up and hierarchical versus nonhierarchical approaches for the "first time right silicon" design times. The design process starts from approved specifications, and it is completed when every aspect of the design is simulated and the outcome is verified to sufficiently match the expected performance. We studied three concrete examples of generic analog circuits by creating project schedules for each of them.

a) A low dropout voltage regulator (LDO), with only six functional blocks and one feedback loop inside the system (Figure 1a).
b) A step-down switching power supply (Switcher), with an analog control system for the regulator (Figure 1b).
c) A power management unit (PMU), an analog-heavy mixed signal system-on-chip (SoC) circuit, consisting of both the LDOs and the Switcher of a) and b) controlled by a digital state machine that takes input from both digital signals and an external sensor. The sensor input is converted to a digital form by an AD converter (Figure 1c). The digital design is assumed to be simple enough so that its design and verification is not in the critical path of the design project execution.

Figure 1a–1c shows the main design tasks, functional blocks, and hierarchical partitioning of the example circuits.

We assume that all the examples are realized with the same technology and using the same tools such that the comparison between design times can focus on only the choice of methodology. Although the examples are fully imaginary products, on a general level, they represent classes of real-life power management IC products that are widely used in electronics today. In our experience, the selection between hierarchical design, bottom-up and

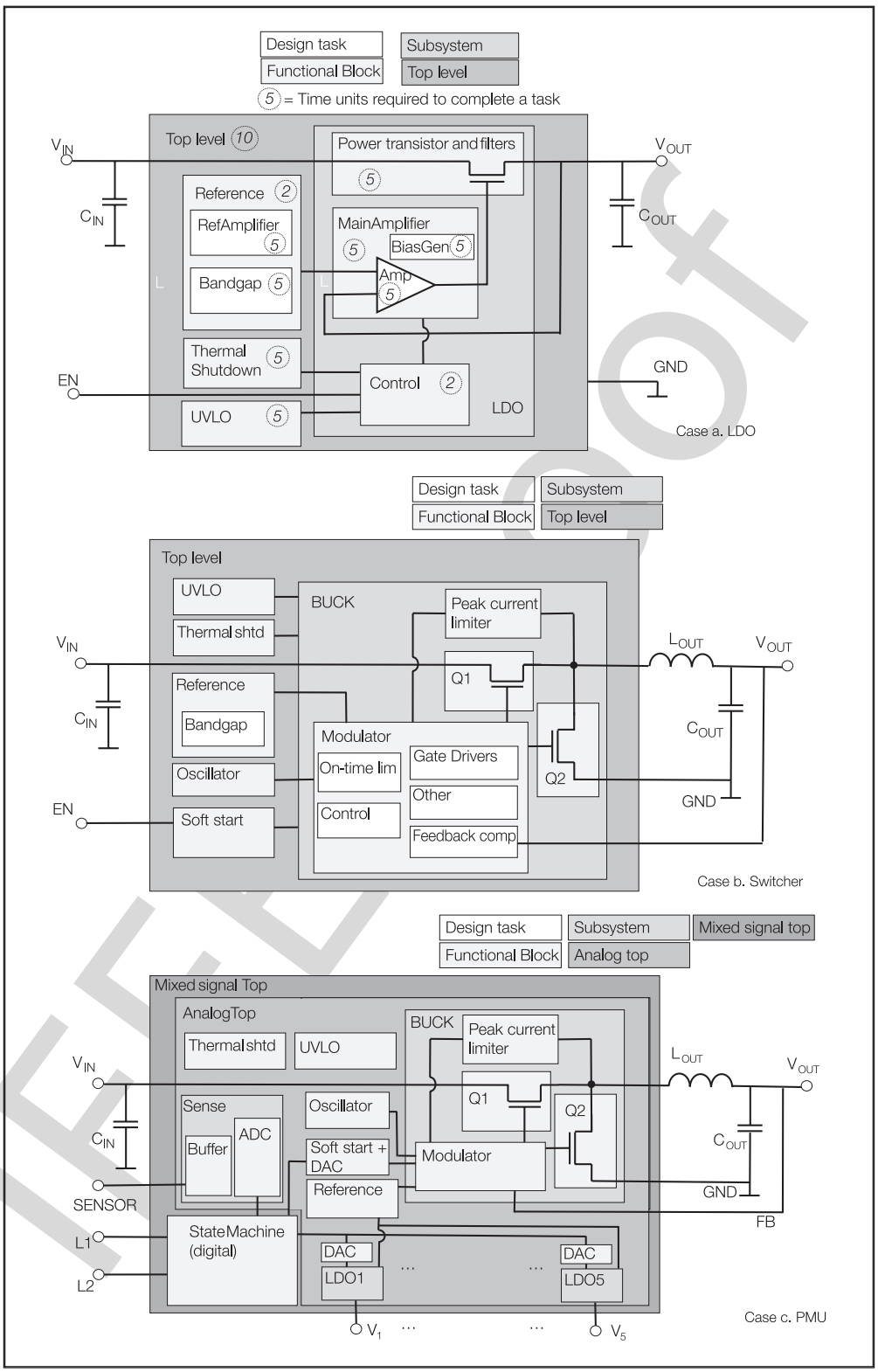

Figure 1. (a) Generic LDO regulator with enable. (b) Generic synchronous step-down switching mode power supply. (c) PMU with SMPS, 5x linear regulators, sensor input, and digital control.

top-down approaches are often done for the levels of system complexity represented by these types of designs. It is of particular interest to gain insight into when putting effort into hierarchical design and architectural system design really starts to improve the design time.

Functional block design flow and work tasks

In all of the above examples, the designs were partitioned into functional blocks, which are considered to be the lowest level of verifiability for analog testing. Designing the block consists of four work tasks.

- Schematics design: Circuit schematics capture, check of functionality and electrical design rules for typical cases and project meetings during the design work.
- Schematics verification: Corner and/or Monte Carlo simulations to verify the operation in all conditions, circuit design reviews, fixes and resimulations, documentation.
- Layout design: Layout design capture, DRC, and layout versus schematics (LVS) check.
- Layout verification: Postlayout verification with extracted parasitics, fixes and resimulations, layout design reviews.

For larger functional blocks, there can be several schematics design tasks, which need to be completed before the verification task starts.

Every project has a given number of analog designers, who are responsible for schematic design and verification of the circuit, and layout designers, who are responsible for layout design and layout verification. In addition, in the PMU project, there is also a digital designer for the digital design tasks. The design projects also contain separate tasks for floor planning, the design of ESD structures and pad ring or I/O pads.

We assume a simple design flow, where the dependencies between functional blocks and work tasks are arranged such that the beginning of every schematic design task depends on the completion of the schematics verification of the previous functional block, which is assigned to the same designer. The schematics verification starts once the schematics design of the same block is completed. The start of the layout design task depends both on the completion of the layout

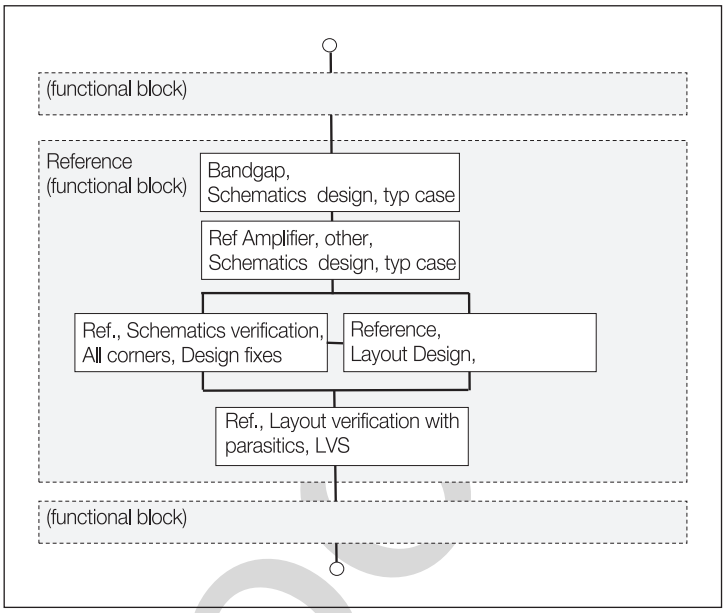

Figure 2. Design flow.

verification of the previous functional block and the completion of the schematics design of the same block (see Figure 2). We propose that this flow resembles the practices in real life and that the layout design is started before every corner of the schematics is fully verified. If the schematics need to be reworked, the layout is usually modified at the same time. This is also compliant with the philosophy of Electrically Aware Design Methodology [10]. However, we assume that the layout verification starts only after the schematics verification is complete.

The design teams of the example projects consists of five to eight block designers, as in Table 1. The structure of the project is optimized manually, in respect with the critical path, such that every designer has design tasks during the entire project and the design tasks are conducted in parallel as much as possible. All of the projects have

Table 1 Resources for the example projects.

Designers	a. LDO	b. Switcher	c. PMU
Analog designers	3	3	5
Layout designers	2	2	3
Digital designers			1

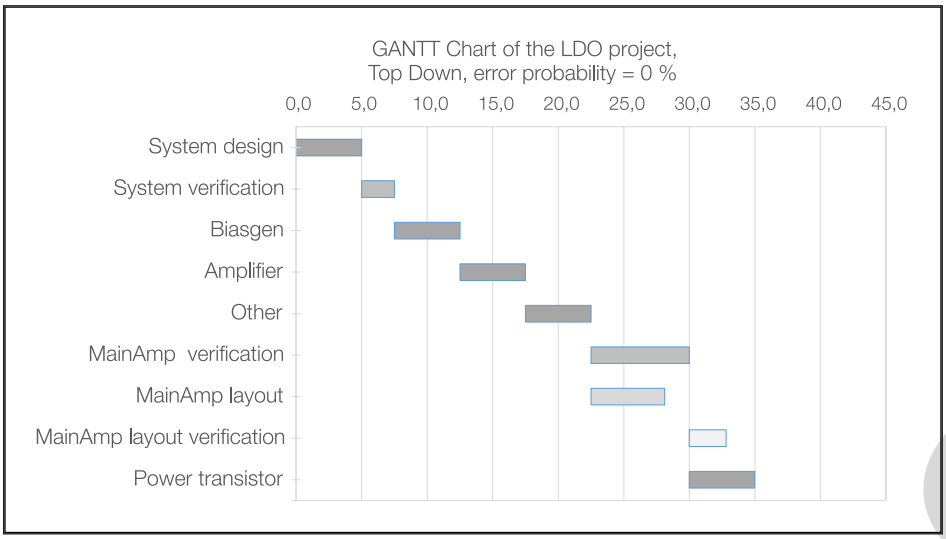

Figure 3. GANTT chart showing the beginning of the LDO project with the top-down approach in a case where p = 0%.

a relatively simple structure, and we expect that a manual check provides a satisfactorily optimal resource usage, such that the differences between the methodological approaches become dominant factors influencing the design time.

Figure 3 shows a sample of the GANTT chart for the top-down case of the example project A (LDO) constructed from an Excel worksheet.

The design times for each block are calculated so that the time t_{sd} consumed by schematics design, and in case of top-down approach also by system design, is estimated manually. In our experience, this is usually possible to do with good accuracy, especially if the designers have worked with similar types of circuits previously.

To estimate the verification time, it is proposed that the verification time of a circuit cell depends on the complexity of the circuit raised to a power of 0.5–1.5 [11]. In the model, we assume that the design time t_{sd} is also representative of the complexity of the circuit and assume a linear dependency between verification time and complexity. Thus, in an error-free case

$$t_{sv} = a \cdot t_{sd}$$

where a is an experimental multiplication factor that is highly dependent on the computing speed of performing simulations. Layout design t_{ld} and layout verification t_{lv} times are calculated as

$$t_{ld} = b \cdot (t_{sv} + t_{sd})$$
$$t_{lv} = c \cdot t_{ld}$$

where b and c are multiplication factors that can be estimated from previous experience. The factor of b is in practice dependent on the methodology and tools used in doing the layout and c is related to the speed of doing the postlayout verification. The layout design time t_{ld} is assumed to be dependent on the sum of schematics design and verification times because we assumed that the layout design of the circuit will continue as long as the verification of the schematics causes further rounds of iterations to the design. Layout verification time is, similar to schematics verification, assumed to be dependent on the complexity of the layout.

Error models

The impact of design methodology manifests in how well it helps to observe, locate, and facilitate fixing the errors that occur during schematics and layout design phases. Assuming that there were no design or modeling errors or specification violations at all, the simplest design methodology, with no time spent on anything but inputting the design and simulating it, should be the fastest way to complete the project. Yet, in real life, errors do occur, and locating the errors and fixing them increases verification times. The optimal methodology should capture all the mistakes and allow for fixing them in a way such that the overall design time is optimized.

In the model, we classify errors as system design errors, schematics design errors, and layout design errors. We assume that errors can be internal, when they are created during the design tasks, or external, when they are caused by factors outside the design team's own activity. Internal errors are often design mistakes, and external errors could be caused by inaccurate device or external component models, or sometimes surprising design specification changes initiated by changes in business case assumptions. Possible root causes for different types of errors are reviewed in Table 2. Fixing the errors increases the schematic and layout verification times t_{sv} and

t_{lv}, depending on their likelihood and severity. Thus, if the relative impact of a design error of type $d\,i$ increases the verification time with a relative factor of $i\,di$ (%), e.g., because of fixing the schematics and resimulating the design, and the probability for the occurrence of an error is p_{di} (%), the design verification time becomes increased by the term $t_{dv}\,i_{di}p_{di} = a \cdot t_{sd}\,i_{di}p_{di}$

Table 2 Error types.

Error types			Examples of error mechanisms
System design	Internal	s_i	Block specification inaccuracies leading to instability or extra noise at the system level, EMC issues, mixed signal issues due to pickup of digital signals.
	External	s_e	Changes in specifications or application environment
Schematics design	Internal	d_i	Mistakes in design assumptions, behavior in corner conditions
	External	d_e	Inaccuracies in device models or mismatched data
Layout design	Internal	l_i	Mistakes in layout design assumptions, wire thicknesses, noise pickup, cross-talk
	External	l_e	Inaccuracies in technology parameters, external component models or assumptions about external wiring parasitics

$$t_{dv} = a\,t_{sd}(1 + i_{di}p_{di}).$$

In the same way, layout design errors increase the layout verification time

$$t_{lv} = c \cdot t_{ld}(1 + i_{li}p_{li})$$

where i_{li} and p_{li} are expected values for the relative impact and probability of a layout design error l_i. If the model is applied to actual real-life cases, the values for the relative impact i and probability p for occurrence of the design errors can be collected from previous projects.

In this study, we chose the probability of different types of errors as a variable parameter because we propose that they could be interpreted to measure many things. Internal errors can be related to the following:

- the design skills of the team;
- the lack of design management to implement a satisfactory design flow, reviews, and documentation;
- a schedule that is too tight or has too few resources.

Density of external errors can be proportional to the following:

- the maturity of the technology;
- cooperation of design management and technology and business teams;
- understanding the application environment with respect to the technical details.

The model parameters a, b, and c can be interpreted to describe the computing power and the impact of the choice of the EDA tools in the design flow.

- The parameter a is proportional to the computing speed of performing verification simulations.
- The possibility to implement a layout synthesis tool, which helped to automatize steps in the layout design phase, could reduce parameters b and c and speed up the layout design and verification phases.
- The design flow of Figure 2 and equations (1)–(5) are expected to be able to model both custom and semicustom approaches to layout design [12]. However, for a semicustom approach, the assumption in (2), that the layout design time t_{ld} is linearly dependent on t_{sd} and thus on the complexity of the design, should be studied more in detail. In this article, we assume that the designs, such as the examples a)–c), are usually done with the custom approach, which is usually expected to provide better chances for area and power routing optimization for analog designs (studied, for example, in [12]).

For the design team, many of the external errors appear as changes in the circumstances, but if we consider the IC product development as a joint effort between the customer, marketing, technology team,

and designers, then also inadequate specifications or models can be considered as errors as well.

Model assumptions for top-down versus bottom-up and hierarchical versus flat designs

The model can be used to compare three cases of methodology: top-down (TD) hierarchical, bottom-up (BU), and a "flat" design with no vertical hierarchy. The flat design can also be described as a design project with no methodology other than the partitioning of the design into functional blocks—for small designs and incremental development this still happens.

For all cases of methodology, we assume that every design task (system, schematics, layout) can generate both internal and external design errors in the verification of each functional block. In the case of a flat hierarchy, the challenge is to locate and fix the system-level errors s_i and s_e because the only step after verifying the functional blocks will be the time-consuming top-level simulations. Hence, both internal and external system design errors contribute to the top-level verification time relative to the complexity and design time of the sum of all blocks together.

The difference between hierarchical BU and the flat design model is that with the BU approach, each level of hierarchy is verified separately such that the internal system design errors s_i can be caught and fixed during the verification of each level of abstraction. For BU, we assume that the system design errors contribute to the verification time of subsystems proportionally to the sum of the design times of the immediately lower level of subsystems.

Although it is possible to make internal design mistakes in creating intermediate subsystems, we assumed for both hierarchical approaches (BU and TD) that external design errors d_e, such as problems with device models, should already be detected at the lowest level of verifying functional blocks and that they no longer appear during the verification of higher level subsystems. The external layout design errors l_e, in contrast, occur only during the verification of the top-level layout.

In the TD approach, the design process starts with architectural system design, and the transistor-level design of functional blocks begins only after the system design is fully verified (Figure 3). Hence, we assume that system errors s_i and s_e (Table 2) occur only in the system design stage and not later. In real life, this assumption is partly too strict. Often, the specification of several of the supporting analog blocks can be deemed unlikely to change or sufficiently independent so that it is relatively safe to start their design before the system level is verified in every corner. In that sense, the model might favor the BU approach in some cases.

Modeling results for LDO, Switcher, and PMU

We applied the design project modeling tool to the example cases a)–c), with the estimated design times for functional blocks shown in Table 3. All of the example cases are assumed to be designed independently from a scratch. The design task times t_{sd} are expressed in general time units rather than concrete workdays or workweeks. The relative amount of design time reflects the writers' view of the estimated complexity of the design tasks such that the examples would be comparable to each other. The multiplication factors a, b, and c, used to calculate the layout design and verification times according to (1)–(3), were chosen to be $a = 0.5$, $b = 0.25$, and $c = 0.5$.

Table 3 Estimated design times for the functional blocks.

Design time t_{sd} (time units)	LDO	Switcher	PMU
1	Floorplanning	Floorplanning	
2	LDO control, ESD structures, Padring, Filters, LDO subsystem	ESD design, Padring,	Floorplanning (analog+digital), ESD structures, Padring, LDO control, Filters, Sense subsystem, Analog top
5	All other design tasks	All other design tasks	All other design tasks
10		Top level	System design (TD), ADC, digital design, mixed signal top

The first research question was "how does an increase in the probability of system design errors impact

the total design time for the three methodological approaches?" When we assumed that;

- the relative impact i for all types of errors was expected to be 100%, such that, if an error occurred, it doubles the verification time (including redesigns and reverifications);
- the probability p for system errors was swept while p for all other error types was held at 10 %.

Result of sweeping simultaneously the probability p for both errors s_i and s_e is shown in Figure 4a. This situation could describe a case, where many functional blocks are reused from previous designs. Then, the likelihood for design errors or external errors caused by inaccuracies of technology parameters is relatively small and could be assumed to be constant, and the new errors are caused mainly by system level issues.

As expected, if the probability of system errors is low, the flat design provides the fastest total design time, and top-down is slower roughly by the time spent on architectural system design and its verification. However, once the likelihood for system errors increases above 10 %, putting effort into system design seems to be the most viable alternative for all the example designs.

Figure 4b shows the results, when the probability of internal design errors d_i and l_i are swept from 0 % to 50 %. The error probability for internal design errors could be related, e.g., to too tight schedule and too little resources, the complexity of the IP or the design skills of the team.

Figure 4c shows the results for external errors d_e and l_e. The probability of external errors can be related to the maturity of the technology and design kits or possibly to other external factors influencing verification methodology.

Figure 4d shows a case in which the probability of all error types increases simultaneously. This represents a case, where very little IP has been reused, the technology is new, and there is a likelihood for all types of errors. In such a case, the top-down approach is clearly the way to go, even for smaller designs. If the probability of errors increases beyond 25%, the total design time starts to grow rapidly unless both hierarchical and TD methods are applied. If the error probability is less than 20%, the hierarchical BU approach may still be viable. The quadratic-like behavior of the TTM curve for the BU approach is caused by the fact that impacts of errors on lower level hierarchy are taken into account in calculating the verification time of the next level of hierarchy, and thus, there is a squared term in the calculation of the verification times of subsystems and the top level.

Figure 4e shows the results for the PMU of Figure 4d, when varying the verification time and layout design time multiplication factors a, b, and c, which are used as model parameters. If the parameters are doubled, the impact to the design times of bottom-up and flat designs are quite substantial. On the other hand, decreasing the model parameters cannot compensate for the lack of methodology.

Conclusion

The purpose of this article was to propose a model and a project management tool to estimate the TTM for the design of an analog IC product. We used the model to investigate the "first time right silicon" design times for three examples of analog power management circuits: a linear low dropout regulator (LDO), switch mode DCDC converter (Switcher), and mixed signal PMU, representing typical and increasing complexity. The model parameters chosen are believed to be realistic, but can be adjusted to be based on data of previous projects of the team under study.

When studying the three example cases using the parameters presented in the previous section, we found that a hierarchical top-down approach is a recommendable methodology for all cases if there is any chance for system design errors. Especially in the case of the PMU, the hierarchy of the design is critical to keep the total design time in control even if there were only block level design errors. Yet, for small designs consisting of six to seven functional blocks, and in cases where the likelihood for design iterations due to errors is very small, a flat design might still work and even be faster than putting significant effort into methodology.

The model did not take into account the other important benefits of a top-down approach, such as providing faster information to marketing, application engineering, and production test development. Including these factors could be a future topic for improving the model. Also the impact of different

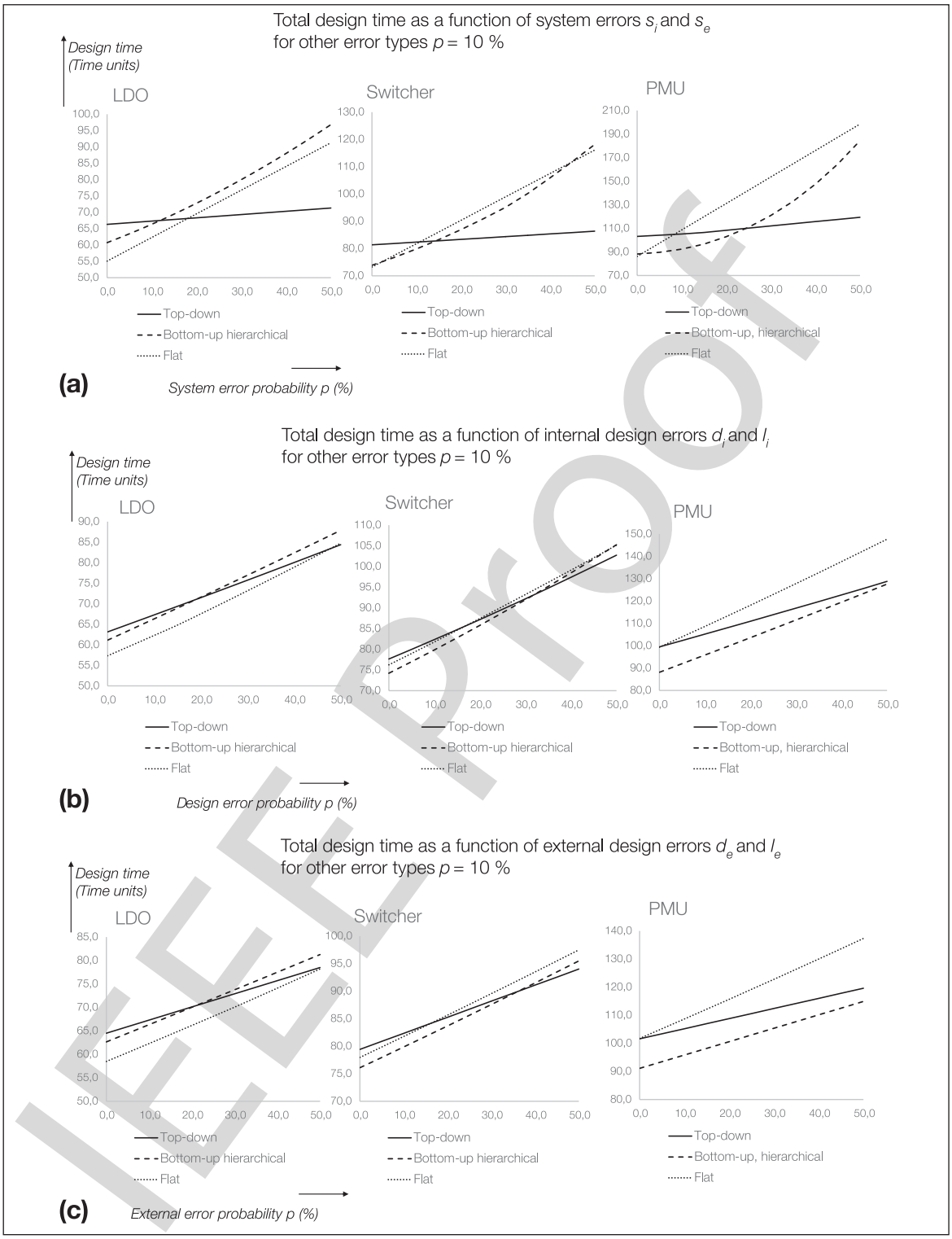

Figure 4. (a) Design time as a function of different types of errors for LDO, Switcher, and PMU. (b) Design time as a function of system errors. (c) Design time as a function of internal design errors.

AQ 1

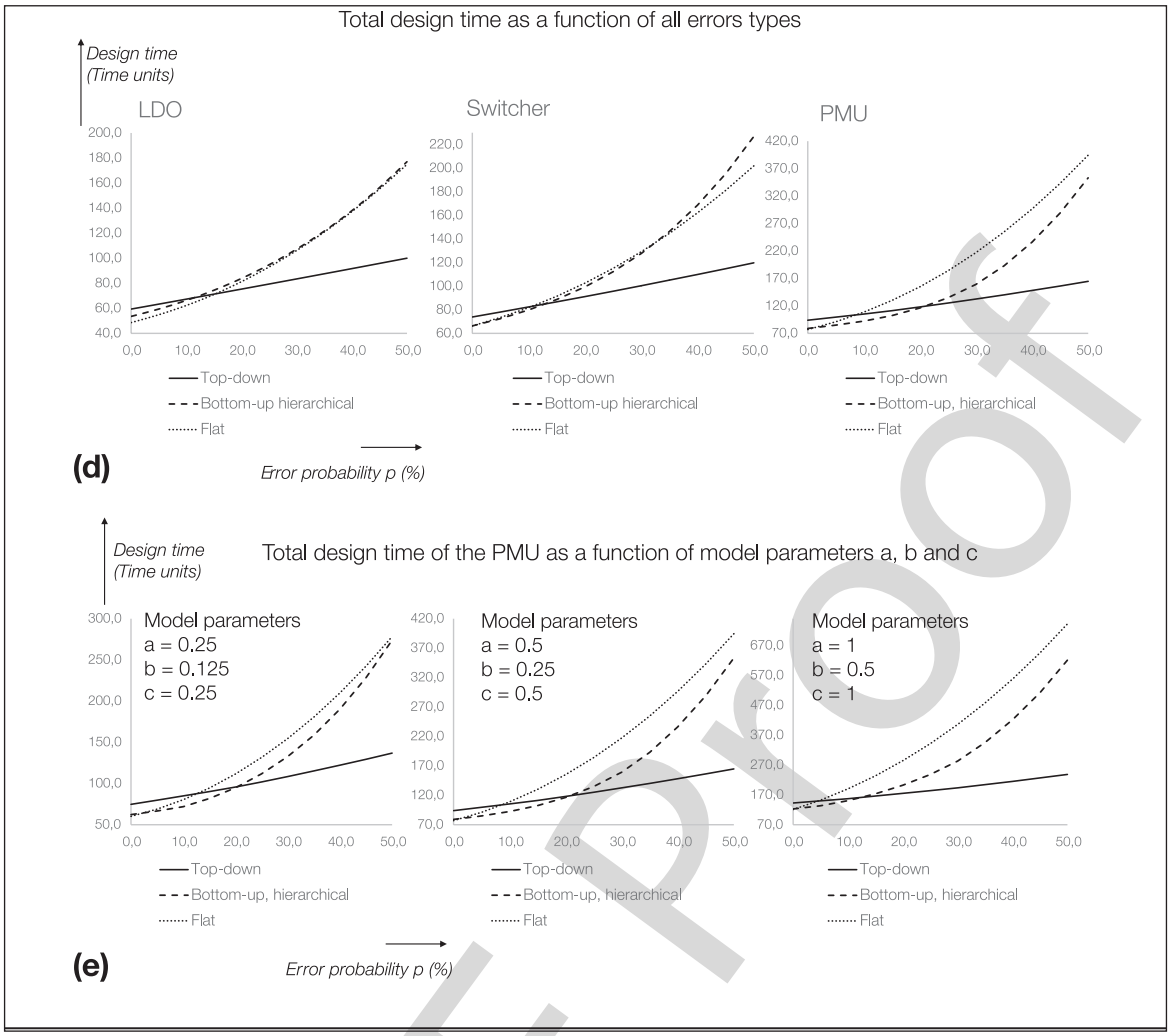

Figure 4 *(Continued.)* (d) Design time as a function of external technology, model and tool related errors. (e) Design time as a function of all error types simultaneously. Design time of the PMU as a function of all error types with different model parameters *a, b*, and *c*.

design flows and EDA tool methodologies to the model parameters *a*, *b*, and *c* would be an important topic to study further in order to experimentally validate the feasibility of the model and improve its accuracy and applicability.

An interesting interpretation of the findings is that the implementation of the appropriate design methodology may be more critical than the probability density of individual design errors in regard to complex designs. The results of the previous section suggest that the best approach from a TTM point of view would be to rigorously maintain the required level of the chosen design methodology and even accept a higher risk for individual errors rather than to compromise the methodology. ■

References

[1] E. Malavasi et al. *A Top-Down, Constraint-Driven Design Methodology for Analog Integrated Circuits*, New York, NY, USA: Springer-Verlag, 1996.

[2] P. Nuzzo and A. Sangiovanni-Vincentelli "Robustness in analog systems: design techniques, methodologies and tools," in *Proc. IEEE Int. Symp. Ind. Embedded Syst.*, Jun. 2011, pp. 194–203.

[3] D. Leenaerts, G. Gielen, and R.A. Rutenbar "Solutions and outstanding challenges for mixed-signal and RF IC design," in *Proc. Comput. Aided Design*, 2001, pp. 270–277.

[4] K. Kundert "Principles of top-down mixed-signal design," presented at the Int. Solid-State Circuit Conf., 2003.

[5] K. Kundert and H. Chang "Top-down design and verification of mixed-signal circuits," *Planet Analog*, Jun. 28, 2005.

[6] K. Kundert "A formal top-down design process for mixed-signal circuits," presented at the Electronic Design Processes Workshop, Monterey, CA, USA, 2001.

[7] F. Badstubner and A. Vorg "Quantitative productivity measurement in IC design," in *Proc. Design Autom. Test Eur.*, 2008, pp.934–935.

[8] N. H. E. Weste and D. M. Harris "Design methodology and tools," in *CMOS VLSI Design, A Circuits and Systems Perspective*, 4th ed., Reading, MA, USA: Addison-Wesley, 2011, Ch. 15.

[9] M. Hamour, R. Saleh, S. Mirabbasi, and A. Ivanov "Analog IP design flow for SoC applications," in *Proc. Int. Symp. Circuits Syst.*, 2003, vol. 4.

[10] D. White, A. Shah, P. Krishnan, M. McSherry, and A. Ginetti "Electrically aware design methodologies for advanced process nodes," *Mixed Signal Methodology Guide*, CDN Press, 2012, pp. 249–284.

[11] J. Vlach *Computer Methods for Circuit Analysis and Design*, New York, NY, USA: van Nostrand Reinhold, 1994.

[12] D. Clein "What is full custom layout design," *EETimes*, Jun. 8, 2001, http://www.eetimes.com/document.asp?doc_id=1277368

Panu Jalas is currently a Senior Researcher at the University of Oulu, Oulu, Finland. He has worked as Director of Engineering at Fairchild Semiconductor and National Semiconductor and as a Managing Director of Quantronics. His interests include the research of best practices in IC design management and product development. Jalas has a PhD in physics from the University of Helsinki, Helsinki, Finland.

Timo Rahkonen is a Full Professor in Circuit Design and Circuit Theory at the University of Oulu, Oulu, Finland. His research interests include analog/RF circuit design, efficient simulation algorithms, and distortion contribution analysis. Rahkonen has an MSc and a PhD in electrical engineering from the University of Oulu.

■ Direct questions and comments about this article to Panu Jalas, Oulu Southern Institute University of Oulu, FI-90014, Oulu, Finland; pjalas@gmail.com.

AUTHOR QUERY

AQ 1: Please note that Figure 4 is too large to fit on one page and has been divided over 2 pages.