

Chapter 2

Design Methodologies for Pipeline ADCs

As explained in Chap. 1, the design of ADCs in adverse digital technologies is a major challenge for designers. This challenge becomes more significant with the scaling of technology which brings about new obstacles (leakage currents, increment of the relative variability of technological parameters, low power supplies, etc). In order to overcome this challenge, designers need to develop robust design methodologies or have access to CAD tools which will allow them to simplify the design procedure. This chapter deals with this issue. Firstly, a brief overview of a conventional top-down design methodology will be given, describing the different hierarchical levels into which the design procedure is split. This will be followed by an explanation of the tools required and needs for supporting the top-down design methodology. Finally, we will present our proposed design methodology, emphasizing the improvements with respect to the conventional ones.

2.1 Top-Down Design Methodology

The design process of an ADC starts with the converter specifications ($ENOB$, f_s , FS , etc.) and ends with the physical layout. During this design procedure, the designers must explore numerous alternatives, study several trade-offs, identify design constraints and dependences and optimize the design in terms of power consumption and silicon area. Obviously, this is a major challenge which generally requires the development of robust and systematic design methodologies. One of the most common approaches is what is known as the top-down design methodology, where the converter design is split into several abstraction levels. Thus, initially the design process is tackled from a functional point of view (top abstraction levels) and refined downwards (low abstraction levels). Accordingly, the converter specifications are translated from the top to the bottom abstraction levels. There are several advantages to this hierarchical decomposition [93]: (1) the possibility to perform system architectural exploration and better overall system optimization at a high level before starting detailed circuit implementations; (2) the fact that it allows the proper identification and understanding of the trade-offs and dependences between the different

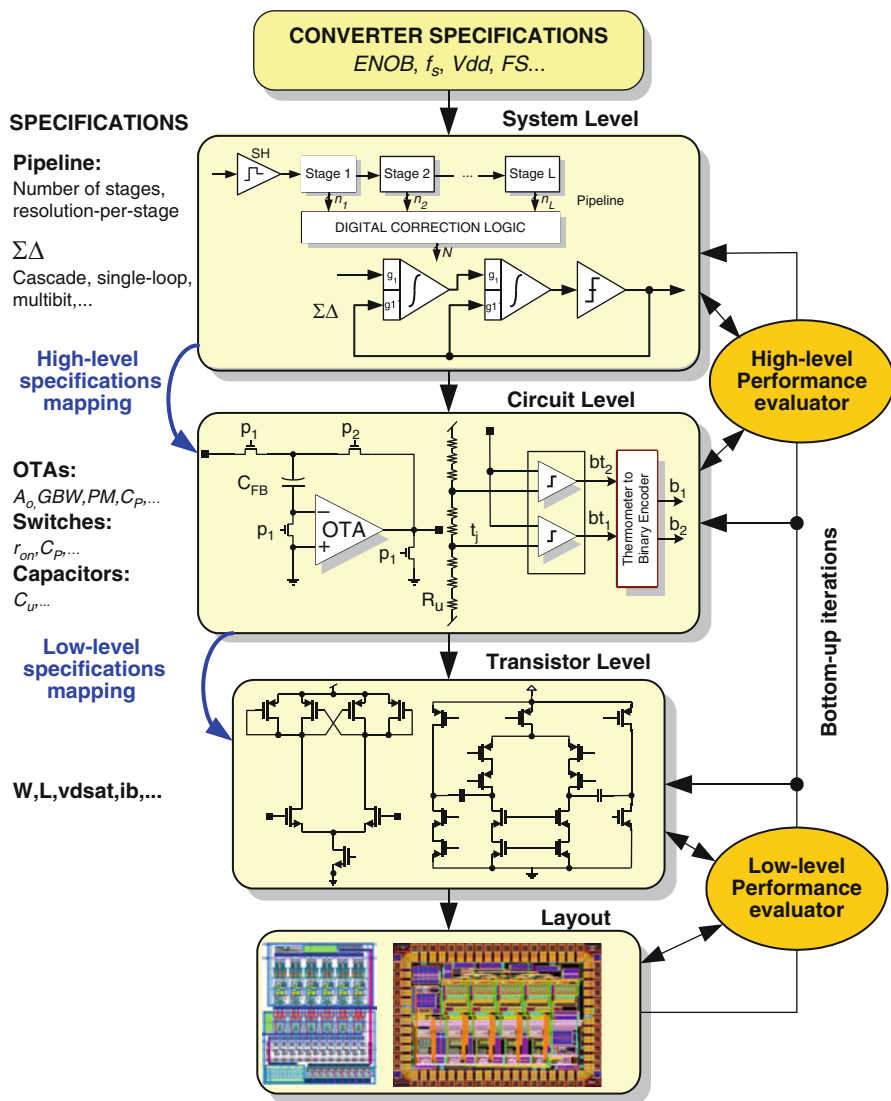


Fig. 2.1 Top-down design methodology for ADCs

hierarchical levels and (3) the fact that the converter design is reduced to find the specifications for each hierarchical level, allowing a much less time-consuming modular design.

Figure 2.1 shows a conventional top-down design methodology for ADCs. As can be inferred, the design procedure is split into four hierarchical levels^{†1}:

^{†1} Note that more hierarchical levels could be considered.

- A system level, where architectural aspects such as the type of converter, the topology, etc. are discussed. At this level, the basic building blocks of the converter can be considered a black box and described at functional level.
- A circuit level, where the basic building blocks are described in greater detail using simple and efficient models which take into account the major non-idealities of the actual circuit implementation.
- A transistor level, where the basic building blocks are described up until electrical level by using complex non-linear differential equations and very accurate models.
- A layout level, where the physical implementation of the transistors is considered.

Obviously, the converter performance evaluation will be much more difficult the closer we get to the bottom hierarchical levels. In fact, specific simulators will be required at the different abstraction levels: (1) a high-level performance evaluator for circuit- and system-level simulations, providing low CPU times and medium accuracy, and (2) low-level performance evaluator for transistor and layout levels, requiring high CPU times but providing high accuracy.

Let us explain the top-down design methodology in Fig. 2.1 in detail. The starting point is provided by the converter specifications, i.e., the *ENOB*, sampling rate, voltage supply, *FS* input range, etc. The objective is to satisfy these converter specifications with minimum power consumption and area. According to these specifications, the converter architecture (pipeline, $\Sigma\Delta$, algorithmic, flash, SAR, . . .) must be selected. At this system level, we must also determine the converter topology, that is to say, the number of stages and resolution-per-stage for a pipeline converter, the single-loop, cascade or multibit approach for $\Sigma\Delta$ converters, etc. Once the architecture has been selected, the converter specifications must be mapped onto high-level specifications for the basic building blocks of the ADC (*high-level mapping procedure*), such as the switch-on resistances (r_{on}), the values for the capacitors (C_u), the OTA DC gain (A_o), Gain-Bandwidth Product (*GBW*), phase margin (*PM*), etc. These circuit-level specifications must then be translated onto physical dimensions, or in other words, onto transistor-level specifications (*low-level mapping procedure*). To complete the top-down design methodology, the physical layout must be carried out.

Although a top-down design methodology attempts to implement a modular design in which every hierarchical level is in principle independent from the remainder, some bottom-up iterations are inevitable. This is due to the trade-offs and dependences between the specifications at different hierarchical levels. For instance, the specifications at circuit level depend on the parasitics from the layout. Hence, a top-down design methodology must be refined with several bottom-up iterations and verifications to satisfy converter specifications. This procedure can therefore take a long time. The more efficient and robust the design methodology, the less time required.

Note that in this design methodology, four key elements are required: (1) procedures to map the specifications from each abstraction level to the subsequent one; (2) verification tools or simulators which allow us to guarantee that the converter specifications are being satisfied; (3) a set of models to characterize converter performance in a specific way and (4) power and area estimation models. These key elements will be discussed in the next section.

2.2 Key Elements in a Top-Down Design Methodology

The main components required in a top-down design methodology have already been presented. This section will examine the most common approaches for the implementation of these components, emphasizing the practical aspects which limit the feasibility of the design methodology.

2.2.1 *Simulator and Circuit Modelling*

As explained in Sect. 2.1, performance evaluators are required to verify whether the converter specifications are satisfied, and also a set of models which allow us to characterize the operation of the converter. Depending on the abstraction level, two main categories can be distinguished: low- and high-level simulators.

2.2.1.1 Low-Level Performance Simulator

Traditionally, SPICE[94] and Spectre[95] simulators have been the most common solutions for evaluating the performance of electronic circuits at transistor level. These simulators numerically solve a system of non-linear differential-algebraic equations which characterize the electronic circuit by using conventional numerical analysis techniques such as numerical integration or Newton-Rapshon iterations. Although these are general-purpose circuit simulators, their numeric algorithms are very slow and their required CPU time quickly increases along with the size of circuit. Therefore, they are used to evaluate the performance of the ADC in the final steps of the design procedure (low abstraction levels) and not for exploration purposes or preliminary estimations.

2.2.1.2 High-Level Performance Simulator

In order to evaluate the performance of the ADC at higher abstraction levels, where fast simulations are required for exploration purposes, more specific and faster alternatives have been developed. In particular, two main types of high-level performance simulators can be distinguished:

- a) *Equation-based* simulators [13, 96], where the converter performance is evaluated by means of simple equations which refer to fundamental limits of different topologies and contemplate trade-offs between power consumption, resolution, speed, etc. Relatively short computation times are required, although the accuracy of the results depends on these equations. Furthermore, equations are obtained through exhaustive analysis of converter topologies. These therefore are closed simulators as equations must be changed every time the topology is changed.
- b) *Behavioural modelling-based* simulators [97–99], where the basic building blocks of the converter are characterized in a more abstract mathematical way by

their input-output behaviour. These higher-level models describe the behaviour of the circuit by means of relatively simple equations which allow a fast evaluation of the performance of the circuit. Obviously, this is achieved to the detriment of the accuracy. Nevertheless, a good speed-accuracy trade-off can be obtained if the major non-idealities of the real circuit implementation are modelled with sufficient accuracy. One of the main problems encountered when using this technique is the lack of systematic methods for the creation of behavioural models and the fact that it generally requires a huge research effort.

2.2.2 *Power and Area Estimators*

The objective of the design methodology is to satisfy the converter specifications with minimum power consumption and area. For this purpose, power and area estimators must be developed. Again, two categories of estimators can be distinguished according to the abstraction level: low- and high-level estimators.

2.2.2.1 **Low-Level Estimators**

At low abstraction levels, the circuits are described at transistor level where non-linear and complex models, such as BSIM3 models [100], are used. These models provide detailed information about the power consumption and area, so highly accurate estimations can be obtained.

2.2.2.2 **High-Level Estimators**

At higher abstraction levels, behavioural models or macromodels are used to describe the behaviour of the circuit at a functional level. Hence, the power and area are estimated by means of simple equations based on experienced designer knowledge. For instance, the power consumption of an amplifier is commonly estimated from the required transconductance (g_m), the transistor overdrive voltage (V_{dsat}) and the voltage supply (V_{dd}) as follows [1–5]:

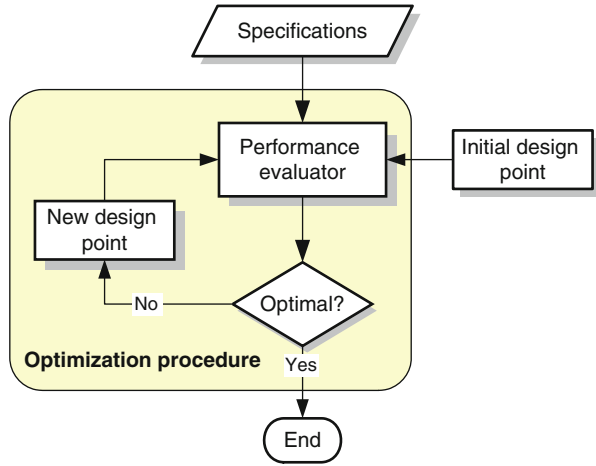
$$P = (g_m v_{dsat}) \cdot V_{dd} \cdot \eta_t \quad (2.1)$$

where η_t is a scaled factor which depends on the amplifier topology. Obviously, this kind of expression provides rough estimations but is enough for exploration purposes.

2.2.3 *Synthesis Procedures*

Apart from a simulator, a set of models for the basic building blocks and power and area estimators, synthesis procedures for carrying out high- and low-level

Fig. 2.2 Basic block diagram of a simulation-based optimization procedure



specifications mapping are required in a top-down design methodology. The most flexible approach consists in using an optimization core combined with an evaluation engine as shown in Fig. 2.2 [93, 101, 102]. The aim is to satisfy the specifications required by using the evaluation engine as performance evaluator and the optimization core as a search engine for the selection of design parameters. Thus, at each iteration of the optimization procedure, the circuit performance is evaluated at a given point of the design parameter space. According to such an evaluation, a movement in the design parameter space is generated and the process is repeated again until the optimum power consumption and area values for the design parameters are found. Typical optimization algorithms are simulated annealing or genetic algorithms. They can also be combined with deterministic algorithms for fine tuning.

In general, simulation-based optimization procedures are used to map both the converter specifications onto circuit-level specifications (*high-level specifications mapping*) and those circuit-level specifications onto transistor-level specifications (*low-level specifications mapping*). The difference between the high-level and low-level synthesis procedures lies in the performance evaluator. In high-level synthesis procedures, high-level performance simulators such as behavioural [97, 101, 102] or equation-based simulators [13, 96] are used, whereas low-level performance simulators like SPICE or Spectre are used in low-level synthesis procedures [15, 103–106].

2.3 Proposed Synthesis Procedure

In this monograph, a top-down design methodology has been developed to synthesize pipeline ADCs at transistor-level. It is able to map the converter specifications directly onto transistor-level specifications by combining an accurate behavioural simulator, a simulated annealing optimization core and a set of Matlab routines for the reduction

of the design parameter space, accurate estimation of parasitics, transistor dimensions and power consumption. In order to show the advantages of the design methodology proposed, we will provide a detailed explanation of how a pipeline converter can be synthesized following a conventional top-down design methodology. The main problems encountered with this design methodology will then be discussed and our proposed solutions will be presented.

As shown in Fig. 2.1, the first step in a top-down design methodology is to select the most suitable converter topology in terms of power consumption and area. Specifically, in a pipeline converter we must select the number of stages and the resolution-per-stage. Typically, this task is reserved for experienced designers who decide which is the best solution in accordance with the knowledge acquired from previous designs. Once the topology has been selected, the converter specifications must be mapped onto high-level specifications for the basic building blocks (*high-level specifications mapping*). A simulation-based optimization procedure is normally used for this purpose. As performance evaluator, a behavioural simulator is the most common approach as it provides fast performance evaluations and an acceptable resolution. In order to achieve this speed, the basic building blocks are described using simple models and parameters. For instance, the switches are modelled by single switch-on resistances (r_{on}) and the OTAs are typically described by 1-pole macromodels as shown in Fig. 2.3. Thus, the OTA behaviour is characterized by a single transconductance (g_m) with a maximum output current (i_o); an output resistance ($1/g_o$) which defines the DC gain ($A_o = g_m/g_o$) and the input (C_p) and output parasitics (C_o). However, some design parameters, such as the parasitic capacitors or currents, cannot be arbitrarily sized since they are very much dependent on the OTA topology and technology. Therefore, they must be initially estimated or related together with other design parameters. Note that this can pose serious limitations since the feasibility of the design methodology can be extremely constrained by a suitable estimation of these parameters. In any case, and in accordance with these descriptions and constraints, the synthesis procedure tries to obtain the values for the design parameters which satisfy the converter specifications with the minimum power consumption and area. At this level, power consumption and area are estimated roughly since they are usually approximated by simple equations as in Eq. (2.1).

Next, these high-level specifications must be translated onto transistor level (*low-level specifications mapping*). The capacitors and the switch-on resistance values can be easily mapped onto electrical dimensions. However, the transistor-level mapping of the OTA parameters is not trivial. The design parameters obtained in the

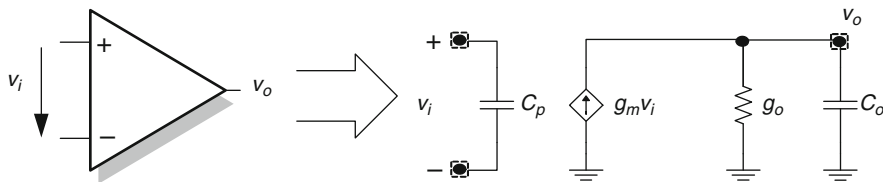


Fig. 2.3 OTA 1-pole macromodel

high-level mapping procedure, that is, transconductances (g_m), saturation currents (i_o), DC gains (A_o) or equivalent load capacitances (C_{eq}), are related to the OTA closed-loop operation. However, these design parameters are traditionally translated onto open-loop specifications[1–5], such as Gain Product Bandwidth (GBW), Margin Phase (PM), slew-rate (SR), etc. A simulation-based synthesis procedure can subsequently be used once more to map these open-loop specifications onto transistor-level specifications. In this case, a general purpose electrical simulator like SPICE or Spectre is employed as performance evaluator. Note that the simulation of these building blocks in open-loop configurations can be carried out quickly even if electrical simulators are used, since they are normally based on DC and AC analysis. Another possibility is to design the OTA by hand. However, these synthesis procedures present a serious problem: if the final transistor-level parameters differ from those estimated during the *high-level specifications mapping*, converter performance can be degraded and a new redesign (or bottom-up iteration) is required. In addition, the fulfilment of the open-loop specifications does not guarantee achieving the target performance when the feedback loop around the OTAs is closed. For illustration purposes, let us assume that an OTA with specific open-loop parameters A_o , GBW , PM , SR and C_{eq} has been synthesized. If the final transistor-level implementation of the OTA presents parasitic capacitances different to those estimated from the high-level mapping procedure, the closed-loop operation can deteriorate noticeably. This is due to the fact that the parasitic capacitances can define the feedback factor to a great extent, in turn modifying the equivalent load capacitance and, as a consequence, possibly slowing down the closed-loop operation of the OTA. Furthermore, the closed-loop operation of the OTA might even become unstable if two-stage topologies are considered. Therefore, a good estimation of the transistor-level specifications is essential when reducing bottom-up iterations and accurately predicting the converter performance. As the final step of the conventional design methodology, the layout of the basic building blocks is carried out.

Against top-down design methodologies which follow the procedure described above, we propose a novel design methodology which presents several advantages:

- 1) The optimum topology selection is guaranteed and does not require experienced designer knowledge since all converter topologies desired can be quickly explored and synthesized at transistor level.
- 2) The behavioural models developed for the basic building blocks provide high accuracy and efficiency since they take into account the major non-idealities of their actual circuit-level implementation, including both small- and large-signal effects which can cause noticeable degradation of the converter performance. Hence, our behavioural simulations show a worst-case deviation from the transistor-level simulations lower than 0.3-bit accuracy. In addition, accurate descriptions have been developed both for one- and two-stage MC OTA topologies.
- 3) The converter specifications are directly mapped onto transistor-level specifications by intrinsically considering the closed-loop operation of the SC circuits. Therefore, intermediate open-loop design parameters are not used. Furthermore, the mapping procedure proposed reduces the design parameter space to only three

design variables. From these design variables, the remaining parameters are optimally sized and several constraints are applied. Thus, the optimization results are found quickly.

- 4) Transistor-level parasitics, as well as sizes for transistors and power consumption are accurately estimated thanks to a set of Matlab routines. These routines include valuable information on the technology using look-up table techniques.
- 5) The bottom-up iterations are drastically reduced since the parasitics and transistor-level parameters are well estimated from the beginning. In fact, only fine tuning is required after the post-layout parasitics are established.
- 6) Both high- and low-level specifications mapping are combined in a single synthesis procedure, which basically allows us to consider the dependences between the different abstraction levels and improve the synthesis results. In addition, the converter design time is drastically reduced since intermediate specifications, mapping procedures or user iterations are not required.

Device-Level Modeling and Synthesis of
High-Performance Pipeline ADCs

Ruiz-Amaya, J.; Delgado-Restituto, M.;

Rodríguez-Vázquez, Á.

2011, XIII, 209 p., Hardcover

ISBN: 978-1-4419-8845-4