TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Entwurfsautomatisierung

Hierarchical Optimization of Large-Scale Analog/Mixed-Signal Circuits Based-on Pareto-Optimal Fronts

Jun Zou

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. Jörg Eberspächer
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. Ulf Schlichtmann
	2.	Univ.-Prof. Dr. rer. nat. Doris Schmitt-Landsiedel

Die Dissertation wurde am 06.02.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 07.06.2009 angenommen.

路漫漫其修远兮,
吾将上下而求索.

屈原

My way ahead is long long one oh ,
I will seek my Beauty high and low .

qu yuan

Acknowledgements

The thesis is the result from my work as a research assistant at the institute for Electric Design Automation, Technische Universität München.

I would like to take this opportunity to express my sincere gratitude to many individuals who have given me a lot of supports during my three-year PhD study.

With utmost respect and gratitude, I wish to thank my advisor Dr. Helmut Gräb for his patience, valuable guidance and encouragement throughout the entire research. He never got tired of discussing my ideas and patiently proofread my publications.

I would also like to thank Professor Ulf Schlichtmann for giving me the chance to work at this institute. He fostered a creative atmosphere and a stimulating work environment at the institute that were essential for the successful completion of my research work.

A grateful word of thanks also to the committee member Professor Schmitt-Landsiedel for her interest in my work.

I am very grateful to my "analog" partner Daniel Müller. His collaboration and support contributed significantly to the successful completion of my research. And also thanks to Tobias Massier, who is generous with his time to help me.

Thanks to Dr. Michael Pronath, Dr. Volker Glöckel, Dr. Bernd Obermeier and Dr. Frank Schenkel for technical supporting on WiCkeD.

Thanks to Infineon Technologies AG and Qimonda AG for their financial support.

Finally, thanks to my parents Qinjuan Cao and Heqing Zou for their love and continuous support. In deepest appreciation, I dedicate my work and this dissertation to my wife Ying Zhang.

Munich, Jan. 2009

Jun Zou

Contents

Chapter 1

Introduction

Integrated circuits (ICs) nowadays exist almost everywhere in our life, we are in constant interaction with various kinds of IC products, from a cell phone in our pockets, or a digital TV at home to a GPS satellite roaming in outer space. In the last few decades, IC industry has grown with astounding speed. According to the report from the global semiconductor alliance (GSA), semiconductor industry revenue totaled $267.5 Billion in 2007, and the torrid growth is expected to continue in the future. IC industry is extremely dynamic with the rapid development in technology - today moving towards sub-45nm geometries and beyond until physical limits. The evaluation of the change on technology is well characterized by **Moore's law**, which states ”that the number of transistors per chip will double every 18 months” [Moo]. This dynamic development has also prompted many incremental challenges, e.g. increased circuit complexity, high design cost and short time to market (TTM).

Combined with advances in deep sub-micron technology, it becomes feasible to integrate hundred million transistors operating concurrently on a single monolithic substrate. Furthermore, various functionalities are tend to be monolithically integrated on one chip, which is usually called **systems on a chip** (SoC). A particular circuit is categorized as either digital or analog, depending on its intended application. Some examples of digital circuits are digital signal processing (DSP) units, memories circuits and micro-controllers with embedded software, while some examples of analog circuits are low noise amplifiers, low/band/high-pass filters, phase-locked loops. Today, most SoCs consist of digital and analog circuits, where they are integrated together in a mixed-signal chip [KHC+01].

The design on digital and analog circuits are two different arts. Digital circuits are comparatively insensitive to processes variation and operating conditions. They consequently offer a more robust behavior than their analog counterparts, although often costing more power, more area, low speed or other drawbacks. Digital circuit design is abstract from the physical details of the actual circuit implementation. A digital design is a top-down process, starting from circuit logic function definition, by means of behavioral description based on hardware description languages (HDLs), then automatic synthesis into gate level, finally to physical layout. Many mature **computer-aided design** (CAD) tools are provided by electrical design automation (EDA) vendors [Cada, Syn, Men] for digital circuit design. Compared to digital circuit with discrete-time and discrete-quantity signals, analog circuit deals with continuous-time and value-continuous signals. It makes analog circuits more difficult to abstract the structural characteristics from the physical realization, and hence increases analog design complexity. Moreover, the performances of analog circuits are more sensitive to variations during fabrication and operation than the performances of digital circuits. Except the well-established *spice*-like simulators (e.g. Eldo [Eld], Saber [Sab] and Spectre [Spe]), very few CAD tools are available for analog design. Up to now, automatic analog synthesis and layout tools are still absent in today market. In consequence, analog design is still a full-custom, multi-iterative-phase, intensive-knowledge and large portfolio of skills required task [AN96].

According to data from leading semiconductor manufacturers, the analog circuits in SoC are estimated to account for just 2% of the total transistors, yet these circuits are 20% of the area, 40% of the design effort and 50% of the re-spins. Hence, any critical analog circuit tends to be a bottleneck for design, implementation, verification, and migration to manufacturing for the overall SoC design, as seen in Fig. 1.1 [Cad02].

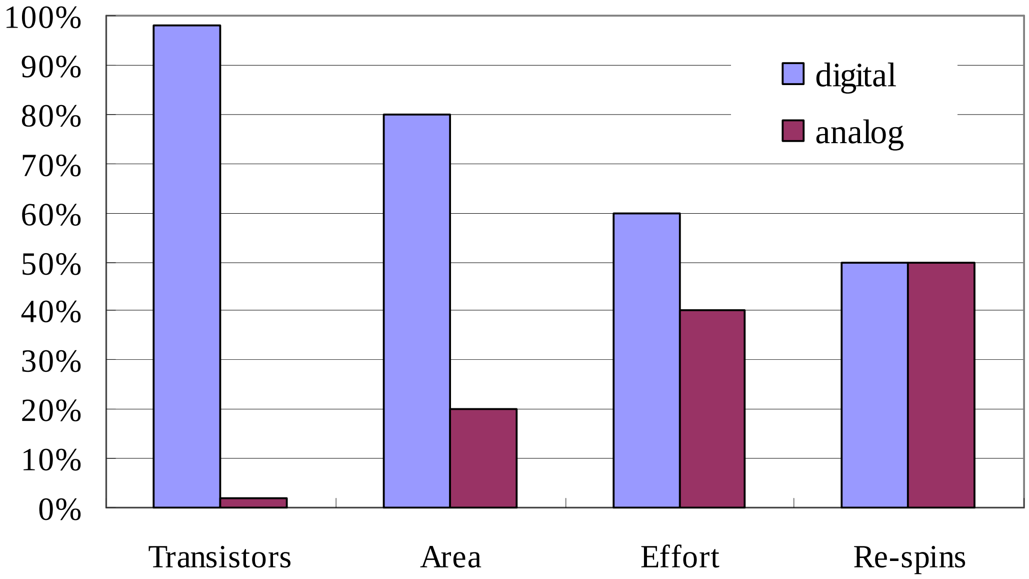


Figure 1.1: Digital versus analog design in SoC [Cad02]

The IC capacity has grown 58% per year, while the rate of productivity increase is only 21% annually, which results in an ever-widening design productivity gap [Ass]. An efficient way to close the gap is to use more advanced CAD tools not only for digital design but also for analog design. Recently, some automatic sizing tools for analog circuit have been introduced in industrial branch, such as WiCkeD [Wic] or Neolinear [Neo]. However these tools can handle only small analog circuits, e.g. operation amplifier (OP AMP). The scale of analog circuits becomes larger, moreover digital circuits are often mixed into the analog environment. This kind of circuit is called *large-scale analog/mixed-signal circuit*. An efficient and fast design flow usually becomes the key idea for commercial CAD tools. This thesis will address already well-established and upcoming design methods for analog circuits. An efficient design flow is proposed in order to realize a hierarchical optimization process of large-scale analog/mixed-signal circuits.

1.1 Motivation

1.1.1 Indispensable Analog Integrated Circuits

Although analog functionality will be replaced gradually by digital computation, e.g. DSP in place of analog filtering, there are still some typical functions that will always remain analog implementation [GR00]. Let's take a transmitter and a receiver in wireless communication system as the example here.

The wireless communication is principally based on the propagation of analog signals in our real world. Digital-to-analog converter (DAC) and analog-to-digital converter (ADC) are the bridges between the real world and the digital domain, as shown in Fig. 1.2 [Raz97]. In the transmitter path, the digital signal from baseband processor is firstly converted to analog signal through the DAC block. Subsequently, the analog signal is carried to a predefined high frequency*, which is generated by a phase-locked loop (PLL). Then the analog signal will be amplified by a power amplifier (PA) so that the signal can drive an outside antenna without too much distortion. In the receiver path, a low-noise amplifier (LNA) is in charge of filtering out the noise of the received analog signal. Another PLL provides a low-frequency (i.e. baseband-frequency) carrier which mixes together with the filtered signal to the baseband frequency. This down-converted analog signal is proceeded into the baseband processor through the subsequent ADC block. Moreover, either analog or digital circuits require stable biases (supply voltages/currents) for their operation, which are provided by analog circuits, e.g. generators and charge pumps.

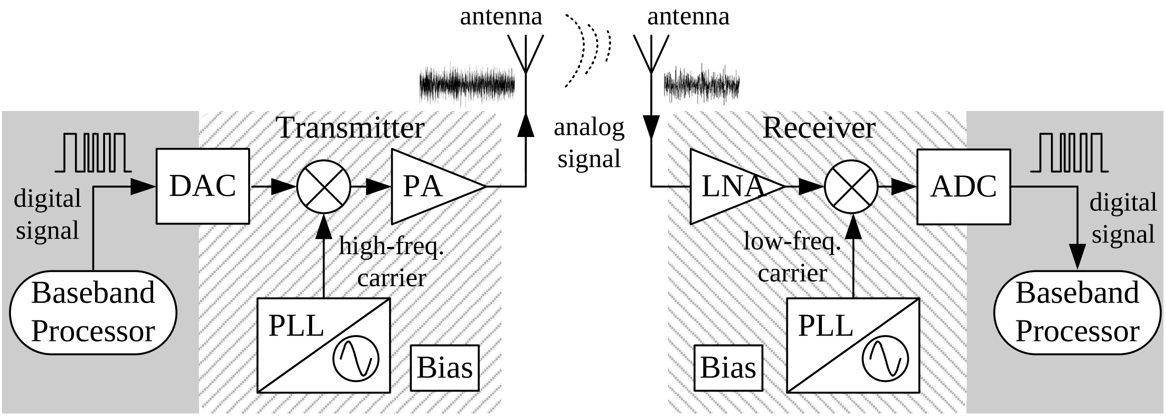


Figure 1.2: Analog circuits in wireless communication system

It is obvious that analog circuits are indispensable today in modern electronic systems. The mentioned above analog circuits, e.g. DAC/ADC and PLL, are usually the vital organ in telecommunication, automotive and many others applications. Actually, more sophisticated analog circuits facilitate the subsequent baseband process and make circuits more efficiently communicate with the real world.

* For example in UMTS wireless system, uplink frequency band is 1920-1980MHz and downlink frequency is 2110-2170MHz, while in GMS system, uplink frequency band is 890-915MHz and downlink frequency is 935-960MHz [Mis04].

1.1.2 Challenges in Design & Optimization of Analog Circuits

"Analog design is a process of choosing the correct subset of parameters to optimize, a choice that's highly dependent on the sophisticated knowledge and long years of experience of the analog circuit designer." [Wil]. Compared to the digital counterpart, analog design has always been a more involved process. In principle, the distinct challenges in analog domain stem from the following unique aspects of analog circuits.

The progress of IC technology is mainly presented by the shrinking of device sizes and the lessened supply voltages. On the one hand, analog circuits benefit from the reduction of device sizes like digital circuits. The circuits become smaller, faster and more power efficient. On the other hand, analog circuits suffer from scaled-down devices, reduced supply voltages, electronic noises and other factors. The smaller the devices are, the large their mismatch is. As the supply voltage goes down, analog designers face more difficulties due to less voltage headroom. For instance in a standard cascode current mirror structure, the current mirror has to operate in a certain voltage range to provide the desirable properties. Under the condition of a low supply voltage, the precise current mirror might be degraded due to an insufficient voltage headroom. Moreover, parasitic effects (e.g. gate/wire capacitance, cross talk, etc.) are more significant with the shrinking device dimension. Analog designers have to take into account these effects during schematic design phase in advance, whereas some effects can be quantified only after their physical layout. At the worst case, some unknown parasitic effects could result in undesired effects, e.g. latch-up phenomenon or large leakage currents.

As the transistor length decreases from 10μm in the 1970s to 45nm today, the impact of process variation on analog performance becomes more significant. Today, analog designers have to evaluate circuit performances at all process corners instead of at one normal corner. The process variations involve not only global/local process parameters but also operation conditions (supply voltages, temperature). *Process corner analysis* and *Monte-Carlo analysis* are used to verify the validity of circuit performance. Hence, many more simulations are needed for analog circuit verification than digital circuit verification.

Regardless of analog circuit or digital circuit, it always exists the conflicting relationship among performances. Power, speed and area are the typical performances of digital circuit. Besides the three performances, analog performances have many more forms: e.g. DC gain, gain bandwidth, phase margin and supply/substrate noise rejection in frequency domain, slew rate, locking time, propagation delay and jitter in time domain. Analog designers face more complex and elusive trade-off optimization problems in analog design. Moreover, the total design freedom in analog circuit is much bigger than that in digital circuit, although the design parameters of analog circuit are often interdependent. In case that analog circuits are designed manually, experienced designers usually size circuit with the help of "thumbs table". Fig. 1.3(b) [TMG02] shows an example of a design "thumbs table" for a two-stage CMOS OP AMP shown in Fig. 1.3(a). Four OP AMP's performances, i.e. slew rate (SR), voltage gain (DC gain), phase margin (PM) and gain-bandwidth product (GBW) are listed from left to right in the table. Four design parameters, i.e. differential-pair bias current (I), compensation capacitance (C_c) and input differential-pair transistor's width (W) are the dominant design parameters, which are listed from top to bottom in the table. For instance, when DC gain and PM are less than their respective specifications, there is only one way to increase their values simultaneously, i.e. by decreasing the value of I. At the same time, the values of SR and of GBW have to be observed so as to keep meeting their specifications too. It is obvious that the optimization task becomes too difficult for designers to comprehend with the increasing number of design parameters and of circuit performances taken into account.

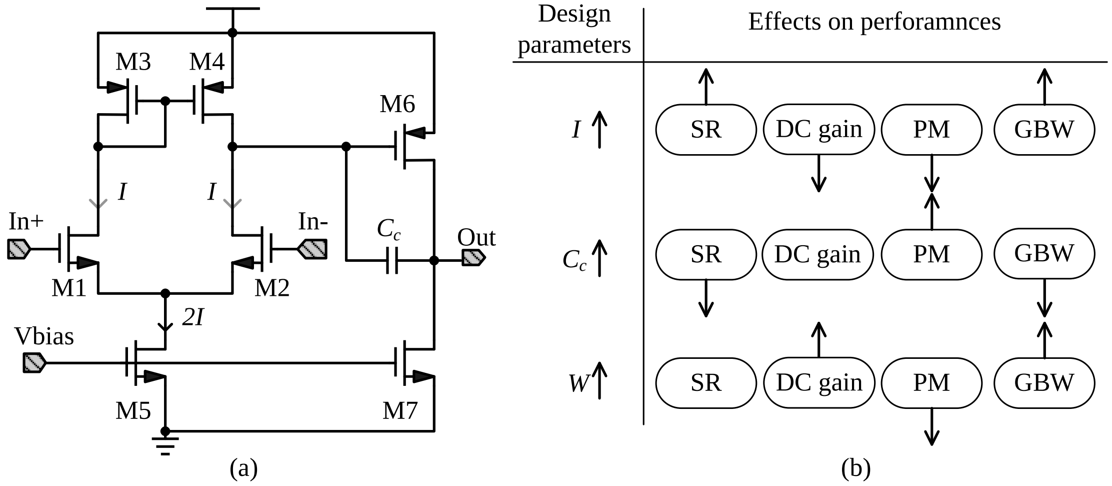


Figure 1.3: (a) Two-stage OP AMP (b) its “thumbs models” [TMG02]

1.1.3 Analog Bottleneck

"While analog and digital system performance increase exponentially over time, microprocessor performance increased more than a thousandfold compared with an increasing of only 10 times for ADCs" [BM04]. Fig. 1.4 shows the ever-widening gap between the relative performance of microprocessors and that of ADC over the last decades. The SoC’s performances are increasingly mainly limited by their analog circuits, not by their digital part.

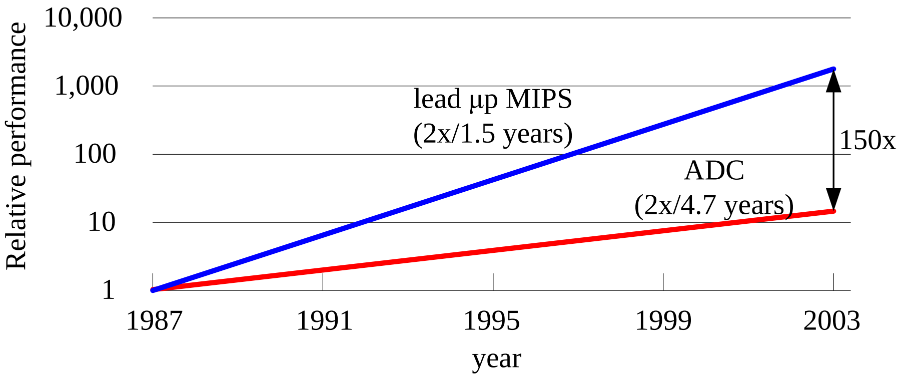


Figure 1.4: Relative performance of analog and digital circuits over time [BM04]

The analog bottleneck is caused not only by the difficulties and challenges of analog circuits themselves, but also by the lack of CAD tools on analog circuits. The design automation degree is much more developed on digital circuits than on analog circuits, which presents on various aspects, e.g. optimization algorithms on circuit design and layout, standard function models, etc. A comprehensive standard libraries are available for designers and these standard digital cells can be easily incorporated into each design, while most analog circuits are often essentially full-custom design every time. The automatic digital synthesis tools are utilized throughout the whole digital design flow, whereas only few specific analog design tools can be applied. Up to now, a general analog synthesis tool doesn't exist "due to the tremendous variability in analog circuits, devices and processes" [Wil]. Furthermore, the analog/mixed-signal circuits bring a new challenge to the traditional analog CAD tools. The *spice*-like numerical simulators are still applied to simulate the large analog/mixed-signal circuits, but it takes too much computer time. As the scale of circuits becomes larger and larger, one single simulation could last over hours or days, which designer cannot endure. Recently, some advanced circuit simulators, e.g. NanoSim [Syn], are developed for analog, digital and mixed-signal circuit simulation. Such simulators can provide much faster simulations than traditional analog simulators with acceptable decrease of simulation accuracy.

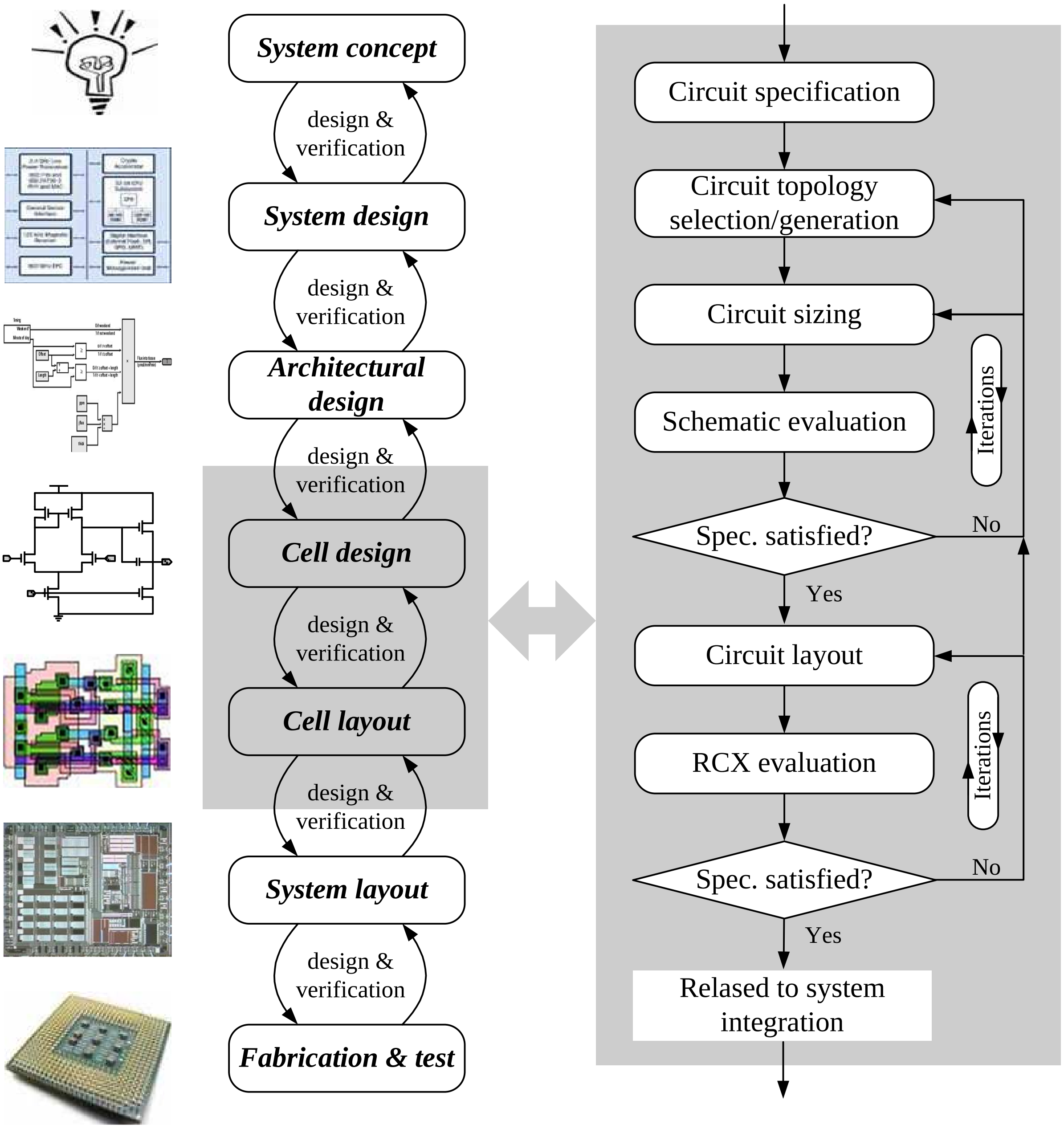
1.2 State of the Art

Though fully automatic synthesis on analog circuit is not yet available today, research on analog synthesis has developed in many directions over the past decades. In this section, a top-down design flow on analog design is briefly described at first. Then, a hierarchical design methodology is introduced for large and complex analog designs. After that, various kinds of automatic sizing methods for analog design are classified. Finally, two main optimization strategies for large-scale analog/mixed-signal circuits are discussed, i.e. flat and hierarchical optimization methodologies. Moreover, performance space exploration methods, which are capable of computation on the performance capability of circuits, are also summarized here.

1.2.1 Analog/Mixed Signal Design Flow

A top-down analog/mixed-signal design process is addressed in [GR00], as shown in Fig. 1.5(a). It mainly consists of seven design steps, which are listed as follows.

1. **Conceptual Design**, where product concept is developed regards to marketing requirements. Overall information on specifications and functionalities are gathered.
2. **System Design**, where the product concept transfers to an actual design plan. System architecture is designed here. Functionalities are defined to implement by software or hardware.
3. **Architectural Design**, where the whole system is partitioned into analog and digital subblocks. System functionality can be firstly verified at this stage by using behavioral function models. The models can be described in C, MATLAB or HDLs.
4. **Cell Design**, where the analog circuits are detailed implemented according to the specific requirements. The tasks include proper circuit topology selection, device sizing and circuit verification. More details are discussed in Sec. 1.2.2.



(a) (b)

Figure 1.5: (a) Hierarchical design steps of analog/mixed-signal integrated circuit design (b) detailed design processes on an analog circuit

5. **Cell Layout**, where each device symbol in circuit schematic are translated into their geometric shapes in circuit layout. The layout is a multi-layer placement of metal, oxide and other semiconduct materials.
6. **System Layout**, where all subblocks (including analog and digital circuits) are well placed and routed together. Chip area, less IR drop on supply nets, isolation of sensitive circuits from noise sources and other issues have to be considered here.
7. **Fabrication and Testing**, where IC chips are eventually produced through certain photolithography processes on silicon substrate. After fabrication, testing is performed to prove product functionality. Products are sorted to sell according to their qualities respectively.

The above seven steps can be classified into two categories: items 1-4 are referred to as the *frontend* design process, while items 5-7 are referred to as the *backend* design process. From item 1 to item 7, it is an ideal *forward progress*. However there exists rarely a pure forward progress in analog design. In fact, extensive simulations and validation steps are required to detect potential problems. If the design fails to meet the target requirement, *backtracking or redesign* processes are needed to revise the failure design steps. In this thesis, the sizing process of the cell design is the mostly focused topic.

1.2.2 Design Process on Analog Circuits

As analog circuits become larger and more complex, prevalent hierarchical design methodology has been introduced in many of the emerging experimental analog CAD systems [HRC89, DGS+96, dPDL+01, CSVM03]. For the design of a large-scale analog/mixed-signal circuit such as phase-locked loops or data converters, the whole circuit is typically decomposed into smaller building blocks, and the hierarchical decomposition goes forward until a level is reached to a physical implementation, i.e. circuit (transistor) level. For design on each analog block, the design steps can be described as the following steps, which are shown in Fig. 1.5(b):

- **Circuit Specification:** The specifications of each building block are derived from the initial system specifications. Examples of circuit specifications are the minimum DC gain, the minimum slew rate, the minimum bandwidth of an OP AMP.
- **Topology Selection/Generation:** Based on the specification requirements, designers choose a suitable circuit topology based on a set of already known alternative topologies. As the requirements become more demanding, new circuit topologies may need to be created.
- **Circuit Sizing:** Actual values are assigned to the design parameters of the circuit elements, such as transistor dimensions, resistance, capacitance, inductance and bias voltage and current. The goal of circuit sizing is to find a set of design parameters so that the circuit can provide the circuit performances which fulfill the predefined specification values.
- **Schematic/RCX Evaluation:** Performances of the sized circuit are evaluated by numerical simulation. A schematic simulation is a pre-layout evaluation, while a RCX simulation is a post-layout evaluation. Compared to a schematic netlist, a RCX netlist includes more parasitic data, e.g. resistance and capacitance on wires, decoupled capacitance between wires, etc. Hence the RCX evaluation validates circuit performances more accurate than the schematic evaluation but at the cost of simulation time.

From circuit specification to circuit sizing, it is a multi-iterations process. The device parameters have to be repeated to tune till the specifications are satisfied. If the selected circuit topology is not able to meet the given specifications, the circuit topology need to be reselected or regenerated. The layout of analog circuit has to be implemented correctly so that the circuit with layout parasitic effects can still fulfill its specification. All presented simulation and optimization results throughout this work are obtained by schematic evaluation.

1.2.3 Automatic Sizing Method on Analog Circuits

If the device values are sized, the circuit performances are uniquely determined. Hence, an optimization process of circuit performances can be referred to a circuit sizing process. In other words, a performance optimization process can be regarded as an automatic sizing process with the predefined circuit specification. Since the design parameters mostly outnumber the performances, which results in an underconstrained problem with many degrees of design freedom, the inverse mapping from circuit performances to design parameters is usually not unique and also unknown. Basically, there are two methods to solve that. One way is the *knowledge-based sizing* optimization approach by exploiting analog design knowledge and heuristics. The other way is the *optimization-based sizing* approach by interpreting the sizing process as an mathematical optimization problem.

1.2.3.1 Knowledge-Based Sizing Approaches

In case of manual design on analog circuit, designers doesn't need to find out the exact values of device parameters immediately, but rather search for circuit topology modifications, a set of pivotal device parameters and their right change directions that mostly determine the circuit performances. Then, designers have to modify design parameters and simulate the circuit several times until circuit provides the desirable properties. The “thumbs table” gives designers an initial idea of how to adjust the device parameters to approach the specification, but not precise. Compared to the qualitative “thumbs table” analysis, a more quantitative analysis is to use "design equation", in which circuit performances are formulated as a function of device parameters. For example, a well known design equation for the open-loop DC gain of the two-stage CMOS OP AMP shown in Fig. 1.3(a) can be expressed as

$$A = \frac{gm_{M1}gm_{M6}}{(gds_{M1}+gds_{M4})(gds_{M6}+gds_{M7})}. \tag{1.1}$$

where *gm* is the transconductance and *gds* is the output transconductance of MOSFET respectively. The expression gives a clear insight into which small-signal parameters of devices predominantly determine the DC gain in this OP AMP structure and how designers can tune devices to meet the certain specification. In the automatic sizing design flow, these design equations are reformulated in a reverse way so that the design parameters can be calculated for a set of given performance requirements. The reformulated equations are called *design plans*. The knowledge-based optimization approach is illustrated in Fig. 1.6. For a circuit topology under design, specific heuristic design knowledge (including design equations and design strategies) is acquired and programmed explicitly in some certain computer-executable forms. Through executing design plans during the analog synthesis, the design parameters can be automatically sized for a given set of input specifications.

Previously, device models are very simple which includes only few device parameters. With the low complexity, design equations can be created by the experienced analog designers. As the dimension of MOSFET scales down to sub-100nm today, a more comprehensive BSIM (Berkeley Short-channel IGFET Model) [†] is used to accurately reflect the transistor's behavior. Consequently, it is difficult to manually extract the equations between circuit performances and device parameters. Recently, *symbolic analysis* methods [GWS94, WFG$^+$95, FRV96] enable the automatic extraction of design equations on some analog circuits. "A symbolic simulator is a computer tool that takes as input an ordinary (*spice*-type) netlist and returns as output (simplified) analytic expressions for the requested circuit network functions in terms of the symbolic representations of the frequency variable and (some of) the circuit elements" [GWS94].

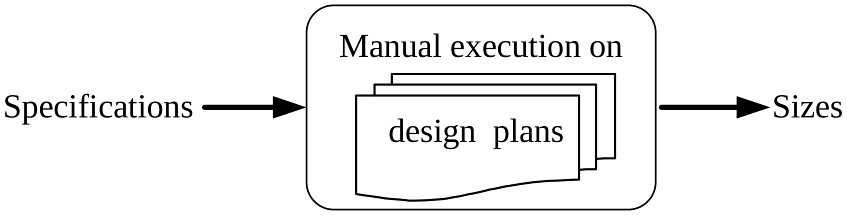


Figure 1.6: Knowledge-based optimization approaches

In the development process of automatic synthesis on analog circuit, the knowledge-based sizing approach is the first generation and some tools came into market in the mid to late 1980s, e.g. IDAC [DND87], OASYS [HRC89], BLADES [TP89], ISAID [TM95, MT95]. However, the knowledge-based approaches suffer from several disadvantages. First, it is very difficult to accurately formalize the circuit behavior. Even symbolic analysis can only handle with the limited kinds of performance on the special circuits. The application of this synthesis method is restricted basically on the circuits whose design plans are available. Second, the design plans have to be updated when the process technology develops from an old generation to the next new one. And it is also very distrustful whether the design equations in the old technology are still valid for the new technology. The updating of design equations costs many manual efforts and time consuming. Third, the optimization results are tightly dependent on the quality of the design equations. Its accuracy is normally lower than that of the simulation-based approach. Forth, procedural knowledge is also required to generate design plans, to handle failure and to backtrack, where many acquisition processes have to be manually conducted. The overall overhead costs much more than the cost by using direct design steps [Hja03]. In summary, the coverage range of the knowledge-based optimization approach was found "to be too small for the real-life industrial practice and therefore these approaches failed in the commercial marketplace" [GR00]. Moreover, the knowledge-based sizing approach is not a real optimization process in strict sense.

[†] "BSIM model is a physics-based, accurate, scalable, robust and predictive MOSFET SPICE model for circuit simulation and CMOS technology development." [BSI]

1.2.3.2 Optimization-Based Sizing Approaches

In order to make the sizing tools more flexible and extensible for various kinds of analog circuits, the optimization-based sizing strategy was developed. In this kind of approach the design result is determined by a numerical optimization algorithm instead of design plans. Some special numerical algorithms are used to implicitly solve the analog design freedom and to optimize the circuit performances under the given specification constraints. An optimization-based sizing approach consists of two main engines: *optimization engine* and *performance evaluation engine*, illustrated in Fig. 1.7. According to the method used for performance evaluation, two subcategories can be distinguished: *equation-based* approach and *simulation-based* approach. According to the numerical algorithm for optimization process, two subcategories can also be distinguished: *deterministic* and *stochastic*.

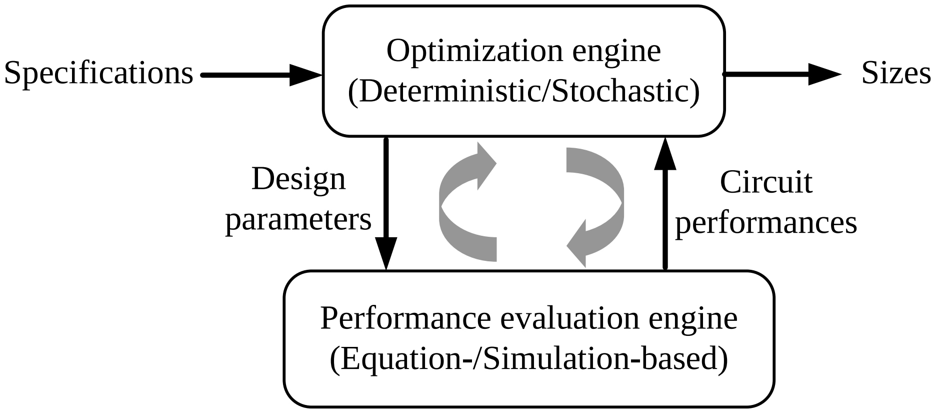

Specifications
Optimization engine
(Deterministic/Stochastic)
Sizes
Design
parameters
Circuit
performances
Performance evaluation engine
(Equation-/Simulation-based)

Figure 1.7: "Simulation-in-a-loop"-based optimization approaches

Equation-Based Approach means that the circuit performance is evaluated by a set of analytic design equations. The equations can be derived manually, e.g. OPASYN [KSpG90] and STAIC [HEL92], or by using symbolic analyzers, e.g. AMGIE [dPDL$^+$01, dPGS02]. In general, the big advantage of these analytic equations is their fast evaluation time. Recently, it has been shown that the designs of OP AMPs in [HBL98] and PLLs in [CPH$^+$03] "can be formulated as a posynominal convex optimization problem that can be solved by using geometric programming techniques, producing a close-by first-cut design in an extremely efficient way" [AH06]. The optimization time can be reduced to minutes or seconds. However, these analytic equations still have to be derived with big manual effort. The accuracy of performance prediction depends strongly on the quality of the analytic equations. Moreover, some circuit characteristics (e.g. transient responses) are difficult to accurately represent by analytic equations, and the current symbolic analysis methods cannot handle most circuit's large-signal properties yet.

Simulation-Based Approach means that the circuit performance is evaluated directly from a *spice*-like simulator. With improving computer power and advanced numerical algorithms in recent years, the idea of simulation-based approach [DR69], which comes from about

40 years before, becomes really practical and more popular today in analog synthesis, e.g. DELIGHT.SPICE [WRSVT88], FRIDGE [MPVARVH94], ANACONDA [PKR$^+$00], MAELSTROM [KPRC99]. These methods perform some forms of full numerical simulations to evaluate the circuit's performance in the optimization loop. Compared to the equation-based approach, a big advantage of the simulation-based approach is that the preparative effort is very low and there exists no issue on performance valuation. The work for designers is only to set up the proper testbenches, which define the real working-environment of circuits and the post-processing for the performance extraction. As long as the circuit performance can be extracted from the simulation, the setup for optimization can be accomplished in a short time usually. The performance prediction by using *spice*-like simulators is the most accurate, since precise device models, e.g. BSIM3 or BSIM4, are applied. The main drawback of the simulation-based approach is the long evaluation time, as performance values are extracted directly from circuit-level simulation and the simulation is executed in each optimization loop.

Deterministic/Stochastic Optimization differs on the applied mathematical algorithm for optimization process. The optimization engine determines the quality of the optimization results and the execution time of the optimization process. Numerical deterministic techniques are mostly based on gradient information and can find a solution in a short time. Sometimes, due to the nonlinear properties of analog circuit these optimization methods might stuck in a local optimum. To avoid a local optimum, stochastic approaches randomly sample on the objective function with a certain probability and can provide a global optimum at the price of a large number of performance evaluations.

According to the above-mentioned methods on the performance evaluation and on the optimization algorithm, the simulation-based approaches from the literatures can be categorized as follows:

Table 1.1: Classification of automatic sizing tools

	Equation-based	Simulation-based
Deterministic optimization	OPASYN [KSpG90]	DELIGHT.SPICE [WRSVT88]
	STAIC [HEL92]	MAELSTROM [KPRC99]
	GPCAD [HBL98]	WiCkeD [AEGP00]
	AMGIE_A [dPDL$^+$01]	ASF [KPH$^+$01]
	Speed [CHC$^+$05]	
Stochastic optimization	OPTIMAN [GWS90]	FRIDGE [MPVARVH94]
	AMGIE [dPDL$^+$01,dPGS02]	ASTRX/OBLX [ORC96]
	DONALD [DLG$^+$98]	ANACONDA [PKR$^+$00]

1.2.4 Optimization Methodology for Large-Scale Analog Circuits

1.2.4.1 Flat Optimization Methodology

In flat methodology, the whole design is attacked at once and all design parameters are treated at the same time. During the design of a large-scale analog/mixed-signal circuit, e.g. PLL or

A/D converter, sizing of all transistors at once will result in a design problem too complex to solve. Furthermore, it will take also too long runtime for the simulation-based sizing methods. Although, a fast performance estimator (equation-based evaluation) and a special algorithm which is capable of handling many design variables, can make the flat optimization realizable in the acceptable time cost, such as posynominal function with geometric programming for A/D converters in [Her02, LWT$^+$05] and PLLs in [CPH$^+$03]. However, the aforementioned disadvantages accompany with the flat optimization: big manual effort for the equation building and the preparation process, less accuracy of the optimization result and the limited application range.

1.2.4.2 Hierarchical Optimization Methodology

The idea of hierarchy is widely adopted nowadays in analog circuit design. For example in [VCD$^+$96], a complex video driver system has been divided into kinds of small analog function blocks, e.g. A/D converter, PLL, digital interface, which are relatively easier to design individually. Starting with the initial system specifications, an optimization process at the top level determines each target specification of the next lower-level design blocks. Through the same way, the hierarchical optimization processes proceed until all the devices at the lowest level of the hierarchy are sized. If any building block is not feasible or the specification cannot be fulfilled at the current hierarchical level, the optimization process at the next higher-level has to be re-conducted to get the new circuit specifications or architecture. The transfer from the initial system requirements to the block specifications is also known as *constraint transformation*. The key for a successful hierarchical design process is to strictly comply with the top-down constraint-driven (TDCD) rules [CCC$^+$97].

In order to avoid the design iterations, bottom-up characterization techniques are introduced into the hierarchical sizing approach in [HS96, KCJ$^+$00, BGV$^+$04, BNSV06]. Recently, a bottom-up characterization approach, i.e. *performance space exploration* (PSE) becomes a hot topic in academic region. PSE has been considered as a key to a true hierarchical design process based on the following two aspects. First, PSE makes it possible to realize an automatic selection of circuit topology, as PSE methods can compute the respective performance ranges of each circuit topology. It is easy to quantitatively compare them and to select the best one for the given requirements. Second, it provides the achievable performance space of lower-level and prevents the sizing on higher-level from producing requirements that cannot be achieved by lower-level realization. Many kinds of PSE approaches are developed and applied to a board range of design problems. Some are more customized to certain circuit types, for example [HMBL99] for LC oscillator and [Her02, BGH04] for A/D converter, and some are more general in [SG03, SGA03, SGA04]. According to the realization technique, three subcategories can be distinguished: *Intermediate performance modeling*, e.g support vector machines in [BJS03, BGV$^+$04], *stochastic optimization* techniques in [SG03, EMG05, SCP05], and *deterministic optimization* techniques in [SG03, SGA03, SGA04]. In the stochastic/deterministic optimization techniques, the performance values are fully evaluated by circuit simulation, while in the intermediate performance modeling, performance values are from simulation and estimation. Based on PSE method, a successful hierarchical top-down optimization process is realizable on various large-scale analog/mixed-signal circuits, for example [BNSV05, ESG$^+$06] for A/D converts and [TVRM04, ZMGS06] for PLLs.

1.3 Objectives of the Work

The goal of this thesis is to construct an effective and efficient optimization methodology for large-scale analog/mixed-signal circuits. The optimization methodology is intended to tackle today's analog bottleneck. There are two popular strategies, i.e. flat and hierarchical optimization methodology. Which of the two methodologies is better, certainly depends on the targeted application. This thesis follows the hierarchical strategy for the following reasons:

- Re-use of building blocks (also in system classes) may be easier.
- The clear distinction between system requirements and building-block requirements enables a deeper insight into the complex trade-offs in an interactive design process.
- Different building-block implementations can easily be investigated.

The top-down propagation of performance specification from a whole system to each building block is the key task in hierarchical design. How to define specification for each building block is the main challenge for manual design. A too stringent specification could overload the design of the building block, while a too loose specification could result in the whole circuit performance out of the original specification. An optimization-based automatic sizing method is applied at system level to find a good combination of the performances of each building block. Additionally, performance space exploration is used to guarantee that the optimized values of design parameters at the higher level can be achieved by the lower-level circuit realization.

In order to achieve more flexibility, more accuracy and more generality with low manual effort, the *simulation-based* performance evaluation method is adopted in automatic sizing process and performance space exploration process as well. According to [RSA99], the behavior of a circuit is usually well natured as long as it works in the correct region of operation. Hence, the corresponding *deterministic* methods are applied in both processes respectively, in order to keep the execution time in reasonable limits. In summary, a ***"simulation-in-a-loop"-based hierarchical optimization*** methodology will be proposed in this thesis. By applying the proposed methodology to some experimental circuits, e.g. PLLs and modulators, the following detailed objectives are achieved within this thesis and these works are published in papers below.

- A first-time-successful top-down sizing process is realizable without iteration redesign steps [ZMG$^+$05, ZMGS06, GZMS07, ZMGS07a].
- Hierarchical optimization of a large-scale analog/mixed-signal circuit is accomplished in reasonable time cost. To meet different performance specifications in various applications, the circuit resizing process can be quickly finished [ZMGS06, GZMS07, ZMGS07b].
- The detailed insight into the capability of the building blocks and the whole circuit system as well can be obtained by respective Pareto-optimal front computation [ZMGS07b].
- Based on the nominal Pareto-optimal front, the circuit performance can be maximized/minimized considering the capability of its building blocks. Based on the worst-case-aware Pareto-optimal front, the actual optimized performance value with a yield of the circuit after fabrication is obtained, where the impact of the inevitable fluctuations of statistical parameters and the variation of operation parameters are considered [ZMGS07a].

This thesis is organized as follows: Chapter 2 introduces two automatic design methods for analog design, i.e. automatic sizing method and performance space exploration. Chapter 3 proposes a comprehensive hierarchical optimization methodology on large-scale analog/mixed-signal circuits. And the practical applications of the proposed methodology are presented on charge-pump phase-locked loops in Chapter 4 and on switch-capacitor sigma-delta modulators in Chapter 5. Chapter 6 summarizes the main topics discussed in this thesis. Appendix A lists the sizing rules for CMOS design. Appendix B shows the system-level modeling of the ΣΔ Modulator in Simulink. Appendix C presents the relationship between phase noise and jitter and explains how to extract jitter performance from phase noise analysis. Appendix D lists the system-level modeling of the charge-pump phase-locked loop in Verilog-A.

Chapter 2

Automatic Design Methods on Analog Circuits

This chapter introduces two automatic design methods, i.e. automatic sizing method and performance space exploration, which are important components of the later proposed hierarchical optimization methodology. The two design processes will be formalized mathematically and the corresponding terminology and fundamental concepts are declared.

2.1 Automatic Circuit Sizing

Analog circuit sizing is usually referred to the determination on sizes of the circuit elements. Automatic sizing methods intend to automatically assign the device sizes according to the predefined circuit specification, and the sized circuit's performance can eventually achieve the target values. Let's take a standard current-mode-logic (CML) block in Fig. 2.1 to explain the corresponding basic knowledge of analog design.

2.1.1 Circuit Parameters

For a fixed topology and process technology, the circuit property is determined by its *circuit parameters* [SS88]. The circuit parameters are comprised of three types of parameters:

- *Design parameters*, vector $\mathbf{d}$[*], are sole designable circuit parameters, whose values can be chosen explicitly by designers. Typical design parameters of CMOS circuits are channel widths/lengths of transistors (W/L) such as $W_{0/1/2/3}/L_{0/1/2/3}$ for $M0$-$M3$ in Fig. 2.1, as well as the values of capacitors (C) and of resistors (R).
- *Statistical parameters*, vector $\mathbf{s}$, present the inevitable fluctuations in the manufacturing process. Typical statistical parameters are oxide thickness t_{ox} and threshold voltage V_{th} of transistors. These parameters are beyond the control of designers and are generally not shown in the circuit schematic.

[*] In this thesis, regular lower case letters denote scalars. Bold lower case letters denote vectors. Bold capitals letters are matrices.

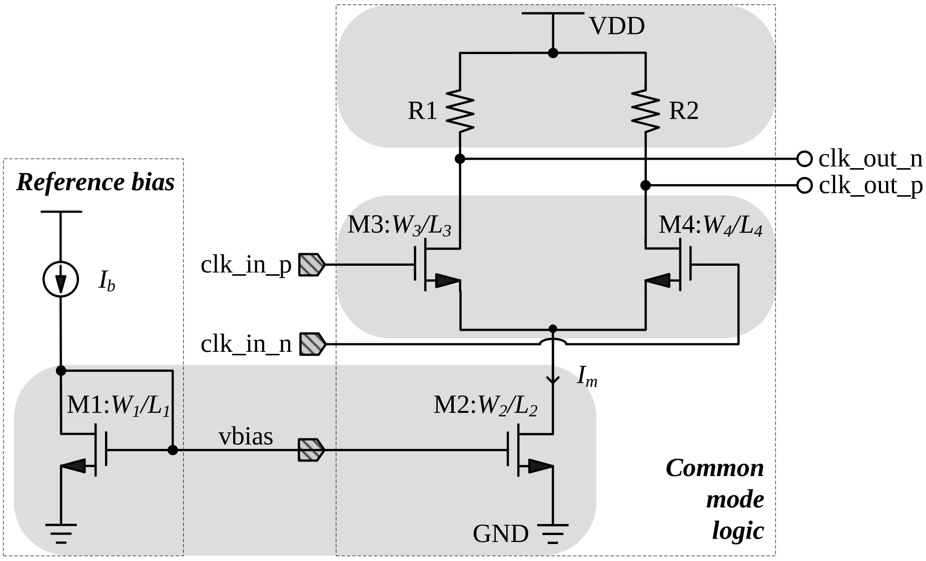


Figure 2.1: Schematic of a current-mode-logic(CML) cell

- *Operational parameters*, vector θ, take into account the variability of the operating conditions, such as ambient temperature (T), supply voltage (VDD) and bias current (I_b). The ranges of the operational parameters are given as part of the specifications and cannot be controlled by designers. For example, the outside temperature varies from -25°C to 115°C and the supply voltage varies from 1.0V to 1.2V.

The circuit parameters can be expressed as [Sch04]

$$\text{circuit parameters} = \begin{cases} \mathbf{d} & \in & \mathbb{R}^{n_\mathbf{d}} & \text{design parameters} \\ \mathbf{s} & \in & \mathbb{R}^{n_\mathbf{s}} & \text{statistical parameters} \\ \theta & \in & \mathbb{R}^{n_\theta} & \text{operational parameters.} \end{cases} \tag{2.1}$$

2.1.2 Circuit Performances and Evaluation

Circuit performances, vector **f**, characterize the behavior of a circuit. The performances of analog circuit **f** are dependent not only on its own circuit realization (i.e. circuit topology, device model and design parameters) but also its corresponding operation environment (e.g. stimuli, output loads). The flow of a simulation-based performance evaluation is briefly shown in Fig. 2.2. The start point is the testbench setup for the DUT (design under test) block. The testbench should represent the real operation environment, which characterizes the DUT's properties under the practical working conditions. For example, a CML cell acts as the DUT. Another CML cell is inserted between the outside stimuli and the DUT, so that the DUT can get a more real input signal (slew rate, input capacitance etc.). And another CML cell acts as the real load for the DUT. Then, the netlist of this testbench is the input of the numerical (*spice*-like) simulators. During the numerical operation process, node voltages and branch currents are calculated based on Kirchhoff's rules and with the help of iterative numerical integration methods. Their values construct a raw simulation data bank. Finally, the circuit behavior is

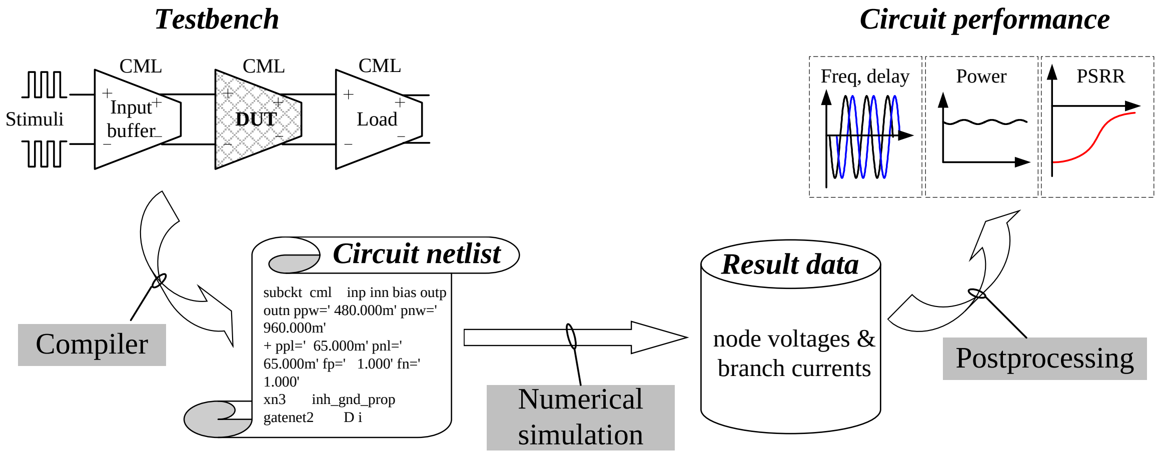


Figure 2.2: Simulation-based performance evaluation

determined by means of circuit simulation and the performance values can be extracted through postprocessing.

Analog design normally consists of two strategies: *Nominal Design* and *Robust Design*.

Nominal Design focuses only on how to adjust design parameters $\mathbf{d}$, while statistical parameters $\mathbf{s}$ and operational parameters θ are assigned to the fixed values (mean value normally). For a given circuit realization (i.e. topology and technology) and the corresponding testbench, the performance evaluation $\mathbf{m}_{\text{nom}}$ maps the circuit design parameters $\mathbf{d}$ to the circuit performances $\mathbf{f}$:

$$\mathbf{f} = \mathbf{m}_{\text{nom}}(\mathbf{d}), \quad \mathbf{f} \in \mathbb{R}^{n_{\mathbf{f}}}. \tag{2.2}$$

Robust Design intends to design circuits more robust against the inevitable variations on process and environment, while nominal design aims at optimizing the various performances at the same time under one certain process corner and environment condition. The performance evaluation $\mathbf{m}_{\text{rob}}$ maps the circuit design parameters $\mathbf{d}$, statistical parameters $\mathbf{s}$ and operational parameters θ to the circuit performances $\mathbf{f}$:

$$\mathbf{f} = \mathbf{m}_{\text{rob}}(\mathbf{d}, \mathbf{s}, \theta), \quad \mathbf{f} \in \mathbb{R}^{n_{\mathbf{f}}}. \tag{2.3}$$

Since the impact of the process variation during manufacturing process and operation environment is much more significant on analog circuits than on digital circuits, analog designers have to do many more simulations in order to acquire a comprehensive insight of the circuit performance. *Process corner analysis* is popular for verification circuit on various technical corners, which includes not only the variation of device process, e.g. five process corners for N/PMOS (TT, FF, SS, FS, SF [†]), high or low passive resistance and capacitance, but the variation of temperature and bias voltage/current as well. Fig. 2.3 shows that the delay of each CML cell in

[†] TT: Typical NMOS & Typical PMOS; FF:Fast NMOS & Fast PMOS; SS: Slow NMOS & Slow PMOS; FS: Fast NMOS & Slow PMOS; SF: Slow NMOS & Fast PMOS.
Additionally, the variation range is also defined for each process, e.g. 2.0σ TT or 3.0σ FF.

Fig. 2.1 varies with a 72 selected technical corners (which is only a subset of the whole technical corners). The waveforms in the lower figure are the input and the output signals. The middle figure shows the delay values between input and output signals at the 72 corners. The upper figure shows the histogram of the delays.

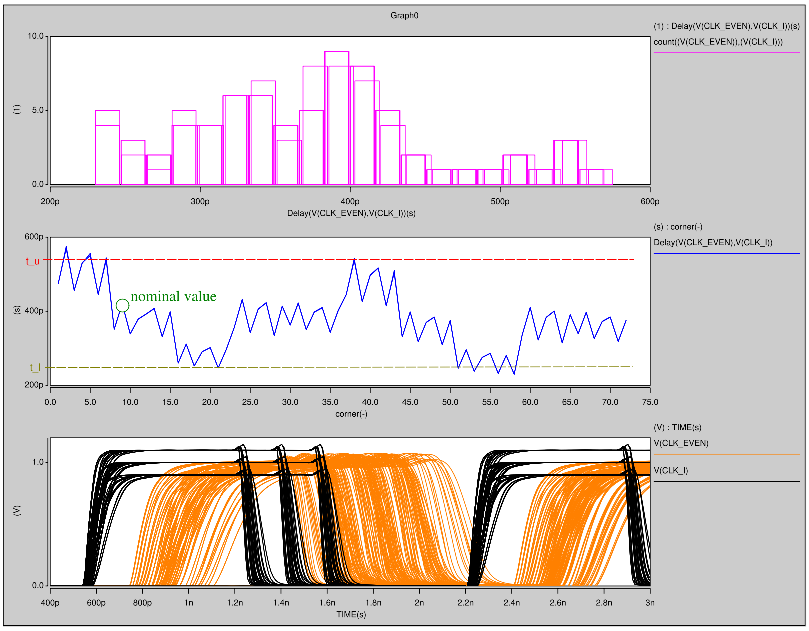


Figure 2.3: Delay variation of CML *vs.* process corners

Actually in simulation-based performance evaluation, Equations 2.2 and 2.3 are referred to the same mapping process, but the values of $\mathbf{s}, \theta$ are different between in nominal design and in robust design. Throughout this thesis, the mapping from circuit parameters to circuit performances is simplified to $\mathbf{f} = \mathbf{m}(*)$, where $*$ represents $\mathbf{d}$ in nominal design case and represents $\mathbf{d}, \mathbf{s}, \theta$ in robust design, respectively.

2.1.3 Circuit Specifications and Yield Estimation

Any design should have its targets or requirements. These requirements on circuit performances $\mathbf{f}$ are called circuit specifications. For example, lower specifications $\mathbf{f}_l$ or/and upper specifications $\mathbf{f}_u$ exist for performances $\mathbf{f}$, i.e.

$$\mathbf{f} \geq \mathbf{f}_l \quad \text{or/and} \quad \mathbf{f} \leq \mathbf{f}_u. \tag{2.4}$$

The circuit performances often suffer from the inevitable process fluctuation and the variation of operation condition. Although circuits are sized to meet their requirements in nominal design, some performances of the fabricated circuits lie out of the specification unfortunately. Such as the delay of the CML in Fig. 2.3, the delay of CML is designed at 400ps at the nominal case. However the delay value varies from case to case and some delays exceed the upper and the lower limits at some corners.

Monte Carlo (MC) analysis is the most popular way to estimate yield of ICs before final silicon tape-out. Device models include global process variation (from wafer to wafer) and local process variation (from die to die in a wafer). For each variation of parameter, a statistical model is used to describe its value distribution. According to the practical number of the statistical parameters in circuits, sufficient simulations (thousands or even many more) are run for the statistical collection. A random generator derives the actual values for these statistical parameters from their models, as shown in Fig. 2.4. The performances of all simulations are measured and then asserted whether they pass or fail their specifications. Finally, the yield can be estimated by

$$Y = \frac{\text{number of pass}}{\text{number of simulation}} = \frac{N_{\text{pass}}}{N_{\text{sum}}} \cdot 100\%. \tag{2.5}$$

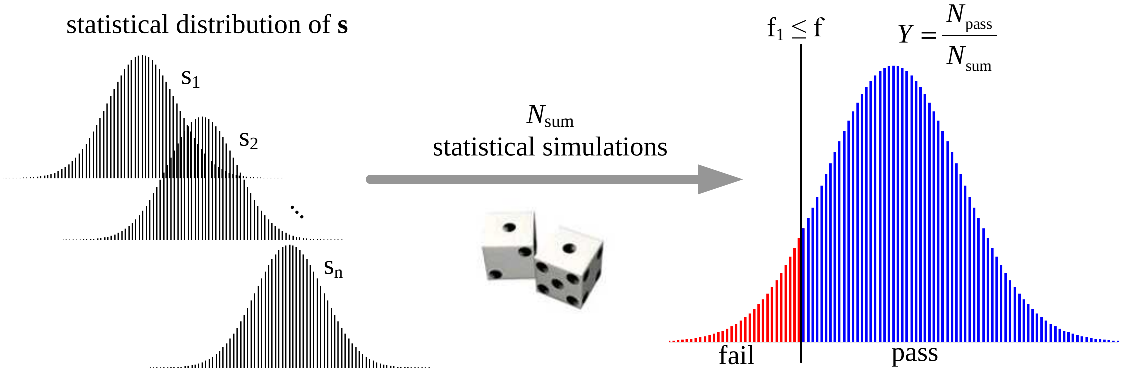

statistical distribution of **s**
s_1
s_2
s_n
N_{sum}
statistical simulations
$f_1 \leq f$
$Y = \frac{N_{\text{pass}}}{N_{\text{sum}}}$
fail
pass

Figure 2.4: Yield estimation by means of MC analysis

Since yield estimation by means of MC analysis is based on the huge simulation cost, academic branch tries to develop other efficient ways to estimate yield value. *Worst-Case Analysis* in [Gra93] is much quick method and with less simulation cost, which will be discussed in the Sec. 3.5.

The overall yield Y_{sum} for all performances **f** is defined as the cut set of all individual parametric yield Y_{f_i}:

$$Y_{\text{sum}} \leq \min(Y_1, Y_2, \ldots, Y_i, \ldots), \quad i = 1, \ldots, n_f. \tag{2.6}$$

For example, although the partial yield of Y_{f_1} is 99.9%, the maximal overall yield Y can be only 68%, because the smallest partial yield is 68% of performance f_2. It can also be known from this table, that we have to take care of each partial yield of every subblocks in the whole system design. It would be useless to maximize only partial yield at the cost of hurting other partial yield.

Table 2.1: Overall yield versus partial yield

Performance	f_1	f_2	f_3	f_4	f_4	Overall
Yield	99.9%	68%	90%	96%	76%	68%

2.1.4 Automatic Sizing Process

Since sole the design parameters **d** can be determined by designers, the sizing process mentioned in the remainder part of the thesis only refers to the sizing on the design parameters

d. An automatic sizing process can be interpreted as an optimization task, which intends to minimize the difference between the circuit performances **f**, i.e. $\mathbf{m}(\mathbf{d})$, and the predefined specifications $\mathbf{f}_{\text{spec}}$ by tuning the design parameters **d**, expressed as

$$\min_{\mathbf{d}} \|\mathbf{m}(\mathbf{d}) - \mathbf{f}_{\text{spec}}\| \quad \Longrightarrow \quad \mathbf{d}'_{\text{opt}}, \quad \mathbf{f}'_{\text{spec}} = \mathbf{m}(\mathbf{d}'_{\text{opt}}). \tag{2.7}$$

where $\mathbf{d}'_{\text{opt}}$ are the optimized results of the design parameters. The mapping process from circuit specifications to design parameters is not unique in mathematic since the design parameters usually outnumber the performances. In case of only considering performance specifications, the mathematical optimizer for automatic sizing process cannot guarantee physically meaningful circuit realization, e.g. transistor working out of saturation. This kind of malfunction doesn't influence the circuit performance at nominal case, but increases the sensitivity of the performance to process variation, environment variation and noise. To guarantee the automatic sizing results in the technically meaningful regions, sizing rules for CMOS technology are proposed in [GZEA01].

2.1.4.1 Sizing Rules

As each analog circuit builds on some elementary transistor-pair blocks, e.g. current mirror, level shifter or differential stage, the performance of the whole analog circuit is crucially dependent on the operation of these elementary blocks. In order to fulfill the desired analog function, most of these elementary structures have to follow some particular *sizing rules* (structural constraints), e.g. transistor's matching or saturation conditions. Hence, a successful and more reliable automatic sizing process can be reformulated as

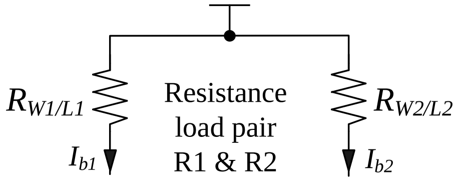
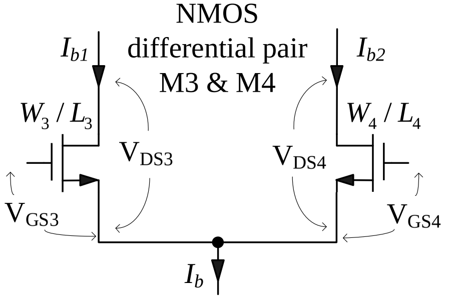
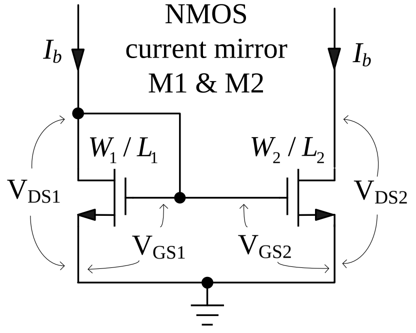
$$\min_{\mathbf{d}} \|\mathbf{m}(\mathbf{d}) - \mathbf{f}_{\text{spec}}\| \quad s.t. \quad \mathbf{c}(\mathbf{d}) \geq 0 \quad \Longrightarrow \quad \mathbf{d}_{\text{opt}}, \quad \mathbf{f}_{\text{spec}} = \mathbf{m}(\mathbf{d}_{\text{opt}}). \tag{2.8}$$

Without loss of generality, $\mathbf{c}(\mathbf{d}) \geq 0$ represents all fulfilled sizing rules, which are the additional constraints for the mathematic optimization.

For a given circuit topology, an *automatic* setup of the sizing rules are presented in [GZEA01, MSG03]. It consists of two main steps. First, circuit substructures are identified bottom-up in a hierarchical fashion as described in Tab. A.1 in Appendix A. Second, based on the recognized structures, the corresponding sizing rules are assigned to the individual transistors. Such as the CML circuit in Fig. 2.1, it is consisted of three structure elements: a resistance load pair (R1 & R2), an NMOS differential input pair (M3 & M4) and a current NMOS source (M2). M2 and M1 form a NMOS current mirror together. According to the design manual of [mun], all sizing rules for the three structure elements are listed in Tab. 2.2. As can be seen from this table, these sizing rules can be classified into three categories [Ste05], as shown in Tab. 2.3.

- **Geometric & Electrical**
  Geometrical sizing rules directly relate to the geometrical dimension of devices, e.g. width and length of transistors. Electrical sizing rules check whether devices work in the expected region. At the current development stage, electrical rules involve only circuits' DC simulation, which calculates the static state of the circuit. The DC voltages and currents are the initial state for nonlinear devices which work in AC (frequency) domain, and in large-signal (time) domain. The electrical sizing rules by means of DC simulation are not sufficient for circuits which work in transient operation.

Table 2.2: Sizing rules for common-mode-logic cell in Fig. 2.1

Structure element	No.	Constraint	Safety margin	Reason
 $R_{W1/L1}$ I_{b1} Resistance load pair R1 & R2 $R_{W2/L2}$ I_{b2}	1	$R_{W1,2} \geq m$	$m = 1\mu m$	limit relative variance of the resistance factor
	2	$R_{L1,2} \geq m$	$m = 1\mu m$	
	3	$R_{W1} = R_{W2}$		systematic load match
	4	$R_{L1} = R_{L2}$		
	5	$VDD - R_{1,2} \cdot I_b \leq m$	$m = 0.4V$	sufficient drain-source voltage headroom for M2 current source
 NMOS differential pair M3 & M4 I_{b1} I_{b2} W_3 / L_3 W_4 / L_4 V_{DS3} V_{DS4} V_{GS3} V_{GS4} I_b	6	$V_{GS3,4} - V_{th3,4} \geq m$	$m = 10mV$	inversion
	7	$V_{DS3,4} - (V_{GS3,4} - V_{th3,4}) \geq m$	$m = 10mV$	saturation
	8	$L_{3,4} \cdot W_{3,4} \geq m$	$m = 1\mu m^2$	limit V_{th} mismatch
	9	$L_{3,4} \geq m$	$m = 0.5\mu m$	limit relative variance of the transconductance factor
	10	$W_{3,4} \geq m$	$m = 0.5\mu m$	
	11	$V_{GS3,4} - V_{th3,4} \leq m$	$m = 1.0V$	reduce the influence of transconductance mismatch on the input offset
	12	$-m \leq V_{DS3} - V_{DS4} \leq m$	$m = 200mV$	reduce the influence of the channel length modulation factor on the current transmission coefficient
	13	$L_3 = L_4$		avoid transconductance mismatch and input offset voltage mismatch
	14	$W_3 = W_4$		
 NMOS current mirror M1 & M2 I_b I_b W_1 / L_1 W_2 / L_2 V_{DS1} V_{DS2} V_{GS1} V_{GS2}	15	$V_{GS1,2} - V_{th1,2} \geq m$	$m = 100mV$	inversion
	16	$V_{DS1,2} - (V_{GS1,2} - V_{th1,2}) \geq m$	$m = 100mV$	saturation
	17	$L_{1,2} \cdot W_{1,2} \geq m$	$m = 1\mu m^2$	limit V_{th} mismatch
	18	$L_{1,2} \geq m$	$m = 0.5\mu m$	limit relative variance of the transconductance factor
	19	$W_{1,2} \geq m$	$m = 0.5\mu m$	
	20	$-m \leq V_{DS1} - V_{DS2} \leq m$	$m = 200mV$	reduce the influence of the channel length modulation factor on the current transmission coefficient
	21	$L_1 = L_2$		limit systematic mismatches

- **Function & Robustness**
  Functional sizing rules guarantee the elementary structures to operate the desired analog functions, e.g. M1 & M2 working in saturation for current mirror operation. Robustness sizing rules define the design margin in order to decrease the sensitivity of analog performance due to the variation of process and of operation conditions, e.g. the minimal length-/width/area. The margin values in Tab. 2.2 are closely dependent on technology process.
- **Equality & Inequality**
  Equality sizing rules state that the design parameters have same values or differ only by a constant factor. In general, the equality relationship exists only for the geometric quantities, e.g. $L_1 = L_2$ in NMOS current mirror. Inequality sizing rules state the upper or lower bounds of the electrical or geometric circuit quantities, e.g. $V_{\mathrm{DS1}} \geq V_{\mathrm{GS1}} - V_{\mathrm{th1}}$ for M1 in saturation.

Table 2.3: Classification of the sizing rules on CML in Tab. 2.2

Rules No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Geometric	*	*	*	*		*	*	*	*	*			*	*			*	*	*		*
Electrical					*						*	*			*	*				*	
Function					*						*	*			*	*				*	
Robustness	*	*			*	*	*	*	*	*	*	*			*	*	*	*	*	*	
Equality	*	*	*	*			*	*	*	*	*	*	*		*	*	*	*	*	*	*
Inequality	*	*			*	*	*	*	*	*	*	*			*	*	*	*	*	*	

2.1.4.2 Automatic Sizing Flow

The design flow of a simulation-based automatic sizing process is briefly shown in Fig. 2.5. The starting point of the automatic sizing flow is the circuit topology in schematic including the corresponding testbenches. By using the graphical interface, e.g. Virtuoso schematic editor of Cadence [Cada], the circuit netlist can be generated and be forwarded to *spice*-like simulator. By means of circuit simulation, the circuit behavior can be determined, including the circuit DC operation points, i.e. node voltages and branch currents, AC (small-signal) performances, e.g. DC gain and phase margin, and transient (large-signal) performances, e.g. delay and slew rate. The geometric sizing rules are configured according to the circuit netlist, and the electrical sizing rules are evaluated by means of the DC simulations.

The performances to be optimized and their corresponding specifications are the inputs of the cost-function (objective-function) generator. With the help of mathematical optimization algorithm, circuit optimizer can find a set of design parameters after several optimization loops. The obtained circuit performances can fulfill their specifications. The cost function and the optimization method have big effect on the time cost of the sizing process and the quality of the final results. Although the cost functions appearing in analog circuit are nonlinear with potential local minimum, experimental results show that the deterministic method performs well, e.g. sequential quadratic programming (SQP) algorithm. An acceptable amount of the starting points result in good solutions while the optimization time is kept in a reasonable cost. The detailed realizations of cost functions and optimization algorithms are beyond the scope of this thesis.

The design parameters are the variables of the optimization. The upper and/or lower bounds of the design parameters should be predefined. For example, the minimal and the maximal limits of transistor dimensions are set to avoid any unrealistic physical implementation.

In the simulation-based automatic sizing method, circuit performances need to be newly evaluated in each optimization loop, so a mass of simulations are required. In order to accelerate the sizing process, the simulation tasks can be distributed onto a cluster of workstations and be executed in parallel. A master machine is in charge of collecting the simulation results and controls the optimization process.

The automatic sizing method provides an automatic mapping process from circuit specifications to design parameters. Currently, the simulation-based automatic sizing method is successfully applied to some analog circuits, e.g. OP AMP design, whose sizes are less than 20-30 devices and the self performance evaluations (simulations) are fast.

2.2 Performance Space Exploration

As we know, circuit simulation provides an automatic mapping process from design parameters to circuit performances. Compared to the mapping from one design point to one performance point, performance space exploration (PSE) intends to find the whole feasible performance space for a given circuit topology and process technology.

2.2.1 Feasible Parameter Space

There exists geometric limitations for each device, i.e. the lower or/and the upper bounds for length/width of devices. For example in 110nm CMOS process technology, the smallest physical channel length(L_{min}) of CMOS is 110nm. Generally in analog design, the minimal lengths of transistors are set at least $1.2 * L_{\text{min}}$, while the maximal bounds for length and width prevent the devices from excessively large. For *n*-number design parameters, the initial design parameter space is a $n_{\mathbf{d}}$-dimension parameter space. Each geometrical point in the initial whole space can be mathematically interpreted as a vector of design parameters $\mathbf{d} \in \mathbb{R}^{n_{\mathbf{d}}}$. In practice, the sizing rules in Sec. 2.1.4.1 separate the whole design space into two nonoverlapping subspaces: one parameter subspace where the associated sizing rules are violated ($\mathbf{c}(\mathbf{d}) < 0$) and the other parameter subspace where the sizing rules are satisfied ($\mathbf{c}(\mathbf{d}) \geq 0$). The subspace, where all sizing rules are fulfilled, is called *feasible parameter space* $\mathcal{D} \subset \mathbb{R}^{n_{\mathbf{d}}}$, i.e.

$$\mathcal{D} = \{\mathbf{d} \mid \mathbf{c}(\mathbf{d}) \geq 0\}, \quad \mathbf{c}(\mathbf{d}) \in \mathbb{R}^{q}, \quad \mathbf{d} \in \mathbb{R}^{n_{\mathbf{d}}}. \tag{2.9}$$

Fig. 2.6 illustrates the feasible parameter space $\mathcal{D}$ for $\mathbb{R}^2$.

To decrease the optimization complexity, less design variables are wanted. Through equality sizing rules, the explicitly algebraic relationships of the correlated design parameters are known. In consequence, the dimension of the design space can be decreased algebraically. It is worthy to mention that only the reduced design parameters are considered throughout this thesis. With

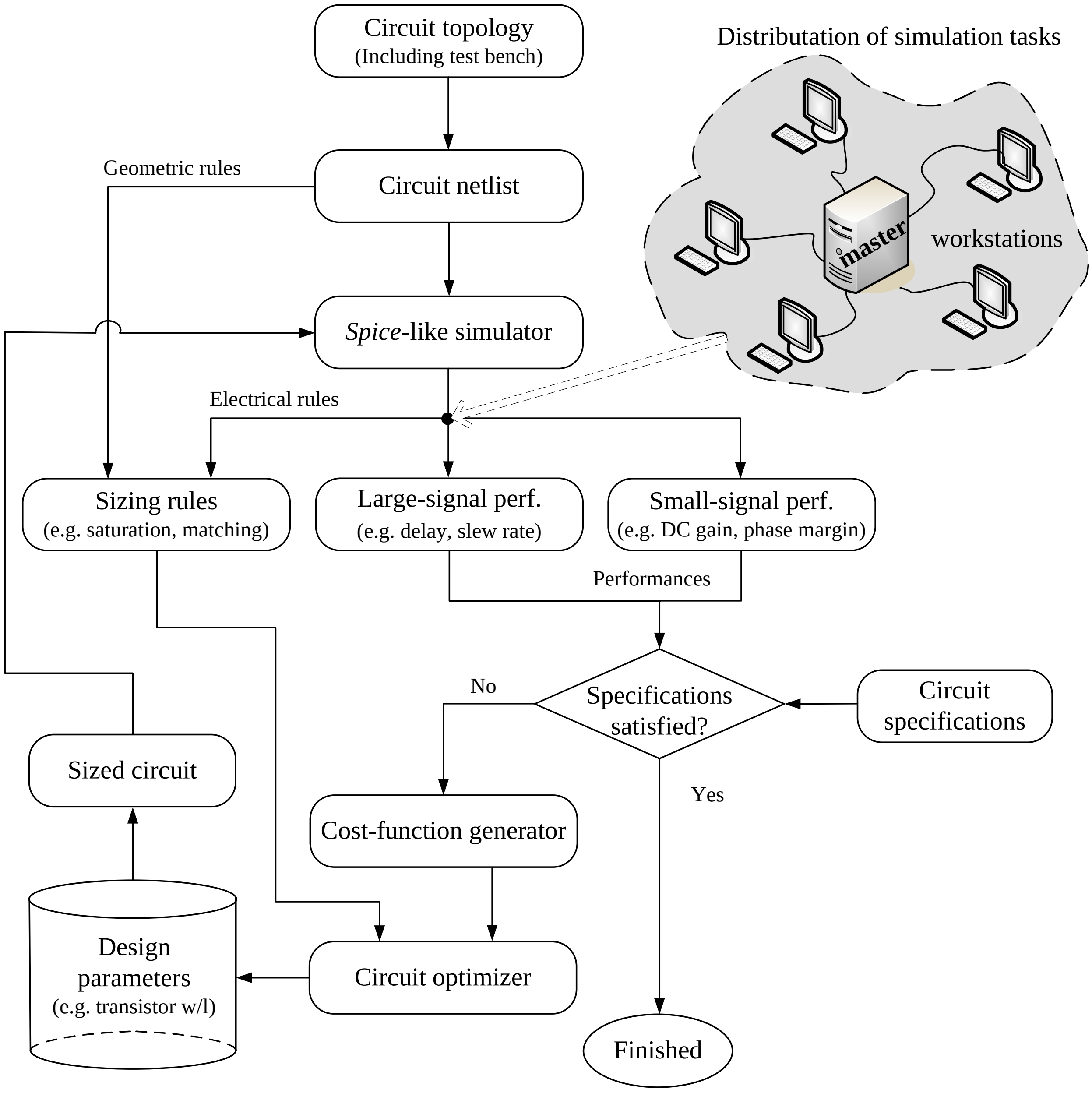


Figure 2.5: Design flow of the automatic sizing process

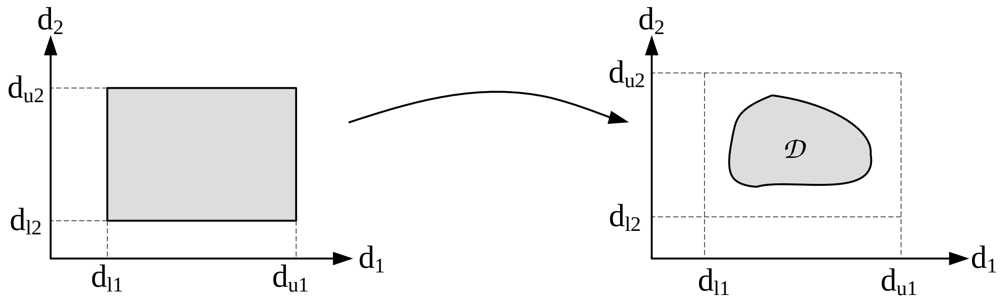


Figure 2.6: Feasible parameter space $\mathcal{D}$

the reduced design parameters, all inequality sizing rules can be reformulated as $\mathbf{c}(\mathbf{d}) \geq 0$, a single nonlinear vector inequality interprets the combination of all the sizing rules

$$\mathbf{c}(\mathbf{d}) \geq 0 \Longrightarrow \underset{i \in \{1,\dots,q\}}{\forall} c_i(\mathbf{d}) \geq 0, \tag{2.10}$$

where q is the total number of sizing rules and the index i denotes the i-th entry of the vector.

2.2.2 Feasible Performance Space

Based on the feasible parameter space $\mathcal{D}$, the feasible performance space $\mathcal{F} \in \mathbb{R}^{n_\mathbf{f}}$ can be obtained from the mapping $\mathbf{m}(\cdot)$, which expressed as

$$\mathcal{F} = \{\mathbf{f} \mid \mathbf{f} = \mathbf{m}(\mathbf{d}) \wedge \mathbf{d} \in \mathcal{D}\}, \quad \mathbf{d} \in \mathcal{D} \Longrightarrow \mathbf{c}(\mathbf{d}) \geq 0. \tag{2.11}$$

Each feasible parameter vector **d** can generate a performance value. The whole *feasible performance space* can be obtained by means of pointwise simulation. However, too many simulations are needed to find the whole range of $\mathcal{F}$. It would be more effective to search only for the boundary of $\mathcal{F}$, $\partial\mathcal{F}$, instead of the entire $\mathcal{F}$. It is worth noting that $\partial\mathcal{F}$ is not the mapping of the boundary of $\mathcal{D}$, $\partial\mathcal{D}$, generally.

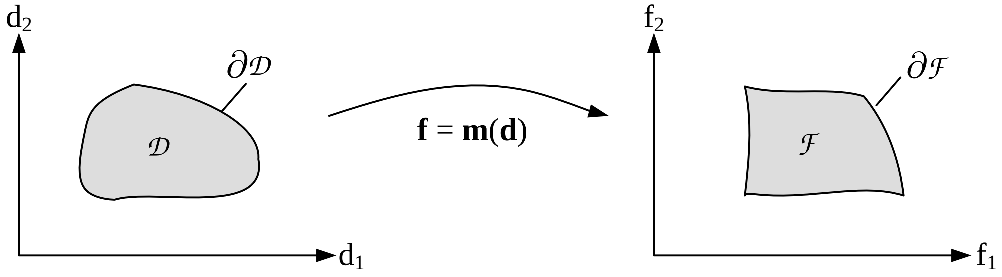


Figure 2.7: Feasible performance space $\mathcal{F}$

The knowledge of the feasible performance space is extremely useful for designers [Ste05]:

- For a given technology and circuit topology, the feasible performance space presents the circuit ultimate capabilities without violating any sizing rules. Various circuit topologies can be easily compared with each other. The advantages and disadvantages of each topology can be accurately evaluated by the comparison on their feasible performance spaces.
- Feasible performance space offers a whole insight into the circuit performance, while traditional optimization method generates only one optimized result. Hence, analog designers have more overview on the circuit capability, and can deliberately select an compromised optimal result among the conflicting performances.
- In a hierarchical design process on a large-scale analog/mixed signal circuit, feasible performance space can be taken as the additional design constraints for the higher-level design. Considering the performance capability of the lower-level circuit can avoid any iteration steps, e.g. redefinition on specifications of subblock.

2.2.3 Performance Space Exploration

The goal of performance space exploration (PSE) is to find a description of $\mathcal{F}$ in the performance space. In Equation 2.11, the feasible performance space $\mathcal{F}$ is coupled to the feasible parameter space $\mathcal{D}$. Since the performances $f_1, \dots f_n$ to be optimized are usually competing with each others, circuit sizing becomes a multi-objective optimization problem [LTZ79] over the design parameter $\mathbf{d}$ under salification with certain sizing rules $\mathbf{c}(\mathbf{d}) \geq 0$, mathematically expressed as [‡]

$$\max_{\mathbf{d}} \mathbf{f} \triangleq \mathbf{m}(\mathbf{d}) \triangleq \begin{bmatrix} m_1(\mathbf{d}) \\ \vdots \\ m_n(\mathbf{d}) \end{bmatrix} \quad s.t. \quad \mathbf{c}(\mathbf{d}) \geq 0 \quad \Longrightarrow \quad \mathbf{d}^*, \mathbf{f}^* = \mathbf{m}(\mathbf{d}^*). \tag{2.12}$$

In practice, it is rarely possible to find a set of design parameters that can improve all performances simultaneously. A trade-off situation mostly occurs among performances, where one performance is improved at the cost of other performances. This phenomenon introduces the concept of *Pareto Optimality* [HM79, LTZ79]. A set of performances $\mathbf{f}_a$ is considered more optimal than a set of $\mathbf{f}_b$, if $\mathbf{f}_a$ dominates $\mathbf{f}_b$, i.e.

$$\mathbf{f}_a \succ \mathbf{f}_b \Leftrightarrow \underset{i \in \{1,\dots,n\}}{\forall} (\mathbf{f}_{a,i} \geqslant \mathbf{f}_{b,i}) \wedge \underset{i \in \{1,\dots,n\}}{\exists} (\mathbf{f}_{a,i} > \mathbf{f}_{b,i}). \tag{2.13}$$

A Pareto optimal point is a set of performances $\mathbf{f}^*$ which is not dominated by any other set $\mathbf{f}$. All of the Pareto optimal points compose of a *Pareto-optimal front* of the performance. Generally, the points on the Pareto-optimal front are also called *efficient points*. A 2D Pareto-optimal front for the two performances f_1 and f_2 is shown in Fig. 2.8. The shaded area shows the feasible performance space $\mathcal{F}$.

Performance space exploration is a computationally expensive process. Under the compromise between accuracy and time cost, a deterministic PSE method, i.e. Normal-Boundary Intersection (NBI) method is adopted in this thesis. NBI method is fit for high accuracy and low-dimensional performance exploration.

2.2.3.1 Normal-Boundary Intersection

The Normal-Boundary Intersection in [SGA03] is a two-step process. In the first step, the individual maxima (IM) $\mathbf{f}_i^*$, where the individual performance f_i shows its global maxima f_i^*, are determined by maximizing the following objective function $o_{\mathrm{IM},i}$:

$$\begin{aligned} &\max_{\mathbf{d}} [o_{\mathrm{IM},i} \triangleq m_i(\mathbf{d})] \quad s.t. \quad \mathbf{c}(\mathbf{d}) \geq 0, \quad i = 1, \dots, n_f \\ &\Longrightarrow \quad \mathbf{d}^*, \quad f_i^* = m_i(\mathbf{d}^*), \quad \mathbf{f}_i^* = \mathbf{m}(\mathbf{d}^*). \end{aligned} \tag{2.14}$$

The IM build up the matrix $\mathbf{F}$:

$$\mathbf{F} = [\mathbf{f}_1^* \quad \cdots \quad \mathbf{f}_{n_f}^*]. \tag{2.15}$$

[‡] Optimization is formulated as maximization. Minimization can be included by maximization of the negative values.

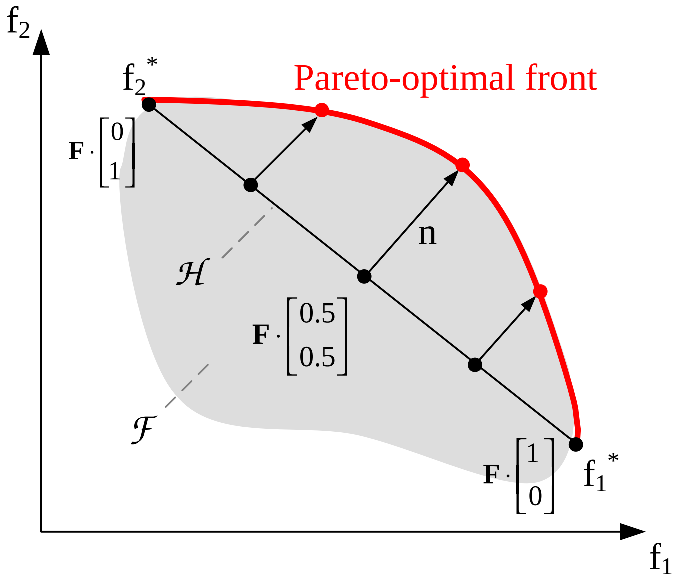

f_2
f_2^*
Pareto-optimal front
$\mathbf{F}\cdot\begin{bmatrix}0\\1\end{bmatrix}$
n
$\mathcal{H}$
$\mathbf{F}\cdot\begin{bmatrix}0.5\\0.5\end{bmatrix}$
$\mathcal{F}$
$\mathbf{F}\cdot\begin{bmatrix}1\\0\end{bmatrix}$
f_1^*
f_1

Figure 2.8: Normal boundary intersection [SGA03]

It is easy to know that these global maxima are the boundary points of the Pareto-optimal front. And the efficient points should locate between these IMs. Assume a polyhedron $\mathcal{H}$ connects with the individual IMs as its corner points. The key idea of NBI is to search for efficient points by starting from the points on $\mathcal{H}$ and going along the lines **n**, which are perpendicular to $\mathcal{H}$ and direct away from origin. If the respective portion of $\partial\mathcal{F}$ is convex, the search results in Pareto-optimal points. Moreover, when the starting points are evenly distributed over $\mathcal{H}$, the found efficient points are also well-placed on the Pareto-optimal front. In fact, the searching direction is not necessary to be exactly perpendicular to $\mathcal{H}$. The quasi-normal vector **n** is calculated by

$$\mathbf{n} = \mathbf{f}_1^* + \cdots + \mathbf{f}_{n_f}^*. \tag{2.16}$$

The efficient points are the best performance set of the circuit under satisfying the sizing rules:

$$\max_{\mathbf{d},\lambda} \lambda \quad s.t. \quad \mathbf{F}\cdot\mathbf{w} + \lambda\cdot\mathbf{n} = \mathbf{f}(\mathbf{d}) \wedge \mathbf{c}(\mathbf{d}) \geq 0, \tag{2.17}$$

where **w** is the set of weights $\mathbf{w} = [w_1 w_2 \cdots w_J]^T$ and

$$\sum_{j=1}^{J} w_j = 1 \quad \text{and} \quad w_j > 0 \quad \text{for} \quad j \in 1, \ldots, J. \tag{2.18}$$

Fig. 2.8 illustrates the method for $n_f = 2$ and $J = 3$. The points $\mathbf{F}\cdot\mathbf{w}$ lie on the connecting line of the individual minima in the 2D case.

2.3 Summary

In this chapter, two automatic design methods for analog design have been described, i.e. the top-down automatic sizing process and the bottom-up performance space exploration.

For automatic sizing process, the sizing rules are used to prevent the automatically sized circuits from senseless results. The design flow of the simulation-based automatic sizing method are discussed. Since performance evaluation is by means of circuit-level simulation, this method is feasible only for small analog circuit design currently.

For performance space exploration, [SGA03, SG03, Her03, SGA04, MSGS05] focus on how to efficiently and accurately perform PSE by means of statistical or deterministic, simulation-based or equation-based methods. A normal-boundary intersection method is applied in this thesis to generate Pareto-optimal fronts.

In the next chapter, how to embed both analog design methods into the proposed hierarchical optimization process will be introduced.

Chapter 3

Proposed Hierarchical Optimization Methodology

For a large-scale analog/mixed-signal circuit, a comprehensive hierarchical optimization methodology is proposed in this thesis. Both analog design methods discussed in Chapter 2, i.e. automatic sizing process and performance space exploration, are well integrated in this methodology. The details of the design flow are described in this chapter.

First of all, let we discuss why hierarchical optimization process is needed and a flat simulation-based optimization method is not feasible for a large-scale analog/mixed signal circuit. The main reasons are the following two factors:

- *Time cost:* A circuit-level simulation on large-scale analog/mixed signal circuit by using numerical simulator is a huge time-consuming process. It is well known that higher simulation accuracy is achieved by setting smaller step size in numerical simulators. Although the accuracy requirements (i.e. step size in the simulation setup) could be different for each block, so long as all blocks are simulated together, the step size of the numerical simulator has to be set small enough to provide an accurate evaluation on the whole circuit. Hence, the common step size is configured as the cut set of all individual step sizes. Why does a single transient simulation (large-signal analysis) of a phase-locked loop last hours or days? Let's have a look at one example in Fig 3.1, the states (voltages/currents) of the blocks *CP* and *PFD* need to be updated every 20ps and 30ps respectively. However, the common step size of the simulation has to be smaller than 8ps because the update rate of the block *VCO* is 8ps. Consequently, many computations on the blocks *PFD* and *CP* are redundant, which results in more computing cost and longer simulation time.
- *Optimization complexity:* The design space and the optimization complexity increases exponentially with the number of design parameters. If all blocks are simultaneously optimized at once, the optimization complexity will become extremely high due to the large number of design parameters.

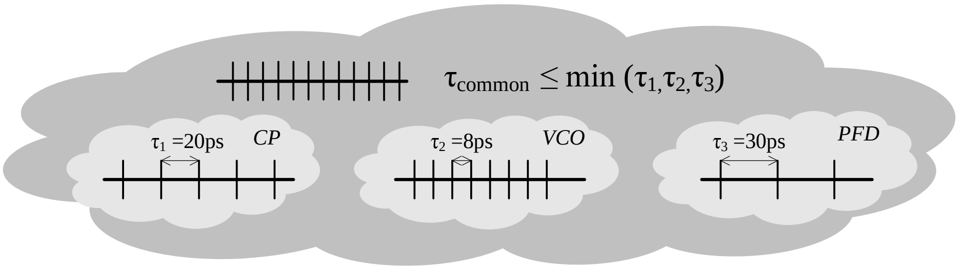


Figure 3.1: Step sizes in numerical simulation

3.1 Hierarchical Top-Down Circuit Sizing

Traditional bottom-up sizing method is not efficient for the design of a complex analog/mixed-signal circuit. One of the most common problem is failure to meet the initial system requirement, which is usually caused by the loose definition of specifications for building blocks. A hierarchical top-down sizing method is adopted to address this issue. The basic idea is that the whole system is partitioned into subsystems, which are further decomposed into smaller function blocks. Based on the initial requirements on the whole system, the respective specifications of each subsystem are first derived by the high-level design. Then, each function block is built in order to fulfill the subsystem requirements. Finally, all parts of system are integrated together and are verified with respect to the original requirements.

The definition of the each hierarchical level is closed dependent on the design system and its complexity. The number of hierarchical level could be three or even more. In this thesis, only two design-levels are considered in hierarchy. Fig 3.2 shows a two-level hierarchical decomposition of an analog/mixed-signal system, i.e. system level and circuit level. At the top of the hierarchy, the entire system performance can be expressed as

$$\mathbf{f} = [f_1, \quad f_2, \quad f_3, \quad \ldots]^T. \tag{3.1}$$

System-Level Parameters The entire system is partitioned into $\alpha = A, B, \ldots$ building blocks. The performances of α building block, which do have effect on the system performances, are noted as $\mathbf{p}_\alpha^T$. The whole system-level parameters can be expressed as

$$\mathbf{p} = [\mathbf{p}_A^T, \mathbf{p}_B^T, \ldots \quad]^T. \tag{3.2}$$

System-Level Simulation The system-level simulation provides a mapping of the system-level parameters onto the system performances: $\mathbf{p} \mapsto \mathbf{f}$. The simulation on system level runs on behavioral models, which represent circuit function in special languages instead of the detailed device models. Normally, the simulation based on behavioral models is very fast, so the simulation-based automatic sizing method is feasible on system-level design.

Automatic Sizing on System Level System performances are optimized subject to system-level constraints by sizing of the system-level parameters $\mathbf{p}$:

$$\begin{aligned} &\max_{\mathbf{p}} \mathbf{f} \quad s.t. \quad \mathbf{c}(\mathbf{p}) \geq 0 \\ &\Longrightarrow \quad \mathbf{p}_{\text{opt}}, \quad \mathbf{f}_{\text{opt}} = \mathbf{m}(\mathbf{p}_{\text{opt}}), \end{aligned} \tag{3.3}$$ with

$$\mathbf{c}(\mathbf{p}) = \{\mathbf{c}_s(\mathbf{p}) \geq 0 \quad \wedge \quad \mathbf{c}_{\mathbf{p}_\alpha}(\mathbf{p}_\alpha) \geq 0, \alpha = A, B, \ldots\}. \tag{3.4}$$

The system-level constraint $\mathbf{c}(\mathbf{p})$ includes two aspects: 1) the necessary design constraints for system itself: $\mathbf{c}_s(\mathbf{p}) \geq 0$, e.g. system stability criterion; 2) a pure top-down refinement of specifications may easily produce overambitious block specifications if the performance capabilities of the underlying analog circuit implementations are not taken into account. Consequently, low-level physical effects have to be propagated bottom-up. At system-level sizing, it is necessary to have a description of the performance capabilities of the each functional block: $\mathbf{c}_{\mathbf{p}_\alpha}(\mathbf{p}_\alpha) \geq 0$.

Circuit-Level Parameters At the circuit level, parameters such as transistor dimensions, resistance, capacitance, inductance and bias voltage/current are assigned to the design parameters of α building block at circuit level, expressed as

$$\mathbf{x}_\alpha = [W_1, L_1, W_2, L_2, \ldots, C, R, I_{\text{bias}}, \ldots]^T. \tag{3.5}$$

Circuit-Level Simulation Circuit simulation provides a mapping of the circuit-level parameters onto the circuit-level performances, i.e. the system-level parameters: $\mathbf{x}_\alpha \mapsto \mathbf{p}_\alpha, \alpha = A, B, \ldots, Z$. Normally, the simulation of each building block is very fast, so the simulation-based automatic sizing method is feasible on circuit-level design of each building block.

Automatic Sizing on Circuit Level The optimized values of system-level parameters, i.e. $\mathbf{p}_{\text{opt}}$, are propagated to the performance specifications of circuit-level. These specifications are achieved by the final circuit-level results, $\mathbf{x}_{\text{opt}}$. The automatic sizing process on α building block can be formulated as

$$\begin{aligned} &\min_{\mathbf{x}_\alpha} \|\mathbf{m}(\mathbf{x}_\alpha) - \mathbf{p}_{\text{opt}}\| \quad s.t. \quad \mathbf{c}_{\mathbf{x}_\alpha}(\mathbf{x}_\alpha) \geq 0, \quad \alpha = A, B, \ldots \\ &\Longrightarrow \quad \mathbf{x}_{\text{opt}}, \quad \mathbf{p}_{\text{opt}} = \mathbf{m}(\mathbf{x}_{\text{opt}}). \end{aligned} \tag{3.6}$$

$\mathbf{c}_{\mathbf{x}_\alpha}(\mathbf{x}_\alpha) \geq 0$ is the circuit-level sizing constraints. The automatic sizing processes for all building blocks can be executed in parallel.

In the hierarchical sizing process, we start with system specifications, follow the top-down propagation of the specifications, and end up with a fully sized circuit implementation on circuit level. Since the performance capabilities of the underlying analog physic implementations are already taken into account, a first-time-successful top-down sizing process can be realized consequently.

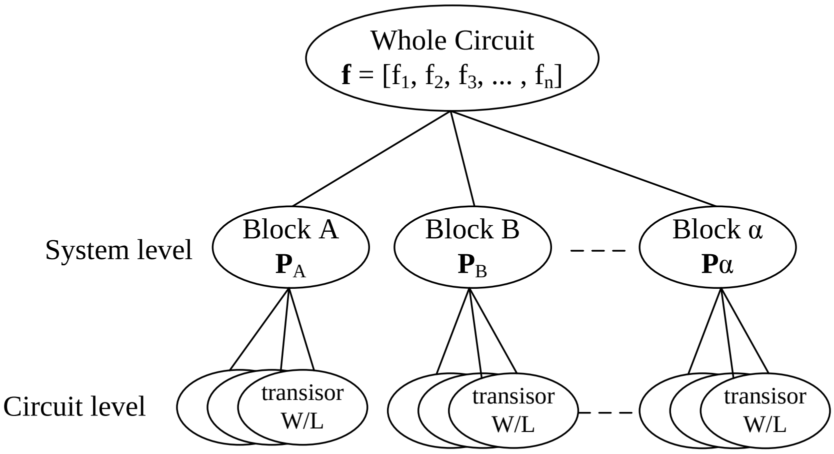


Figure 3.2: Hierarchical sizing of a large-scale analog/mixed-signal circuit

3.2 Pareto-Optimal Front in Hierarchical Optimization

Performance space exploration (PSE) transforms technological constraints bottom-up into feasible performance spaces. Therefore, the feasible region of system-level parameters $\mathbf{p}_\alpha$, i.e. $\mathbf{c}_{\mathbf{p}_\alpha}(\mathbf{p}_\alpha) \geq 0$ can be defined by PSE method on α block. Since PSE is a computationally expensive process, is it really necessary to know the entire feasible region of circuit performances for the optimization at system level? Most of current PSE methods are proposed to find the Pareto-optimal front instead of the entire feasible region. In practice, Pareto-optimal front is sufficient for the system-level optimization.

In analog circuits, a circuit performance has normally the monotone relationship with one design parameter. For example in Fig. 1.3(a), DC gain becomes larger with a larger width of the input NMOS, slew rate increases with the increasing bias current or with the decreasing compensation capacitance. The performances of PLLs in Chapter 4 and of A/D converters in Chapter 5 satisfy the monotone relationship with their design parameters. To simplify theory analysis, a two-dimensional design space $\mathbf{x} = (x_1, x_2)$ and a maximization problem with a two-dimension objective space $\mathbf{f} = (f_1(x_1,x_2), f_2(x_1,x_2))$ are taken here as an example. Suppose there exists a dominated point $(x_1^\diamond, x_2^\diamond)$ in design parameter space, which contributes to an efficient point on the Pareto-optimal front in the objective space. However, this assumption conflicts with the essential definition of the Pareto-optimal front. For the point $(x_1^\diamond, x_2^\diamond)$, there exists a point $(x_1^*, x_2^\diamond)$, whose x_1^* dominates $x_1^\diamond$. It results in

$$\begin{aligned} x_1^* \succ x_1^\diamond \quad &\Longrightarrow \quad f_1(x_1^*,x_2^\diamond) \succ f_1(x_1^\diamond,x_2^\diamond) \quad \wedge \quad f_2(x_1^*,x_2^\diamond) \succ f_2(x_1^\diamond,x_2^\diamond) \\ &\Longrightarrow \quad \mathbf{f}(x_1^*,x_2^\diamond) \succ \mathbf{f}(x_1^\diamond,x_2^\diamond). \end{aligned} \tag{3.7}$$

Or there exists a point $(x_1^\diamond, x_2^*)$, whose x_2^* dominates $x_2^\diamond$, it results in

$$\begin{aligned} x_2^* \succ x_2^\diamond \quad &\Longrightarrow \quad f_1(x_1^\diamond,x_2^*) \succ f_1(x_1^\diamond,x_2^\diamond) \quad \wedge \quad f_2(x_1^\diamond,x_2^*) \succ f_2(x_1^\diamond,x_2^\diamond) \\ &\Longrightarrow \quad \mathbf{f}(x_1^\diamond,x_2^*) \succ \mathbf{f}(x_1^\diamond,x_2^\diamond). \end{aligned} \tag{3.8}$$

Therefore, for a dominated design point $\mathbf{x}^\diamond$ and the corresponding performance $\mathbf{f}^\diamond$, there exists definitely a dominating point $\mathbf{x}^*$ and its corresponding dominating performances $\mathbf{f}^*$. In the hierarchical design, the circuit-level performances are used as design variables at the system level. Hence, an optimal system-level performance should be generated by the design points lying on the Pareto-optimal front of the circuit level. This is coincident with the conclusion in [EMG05]. A lower-level solution moving towards its Pareto-optimal front should result in a movement of its corresponding higher-level solution towards the higher-level Pareto-optimal front, as visualized in Fig. 3.3. According to the conclusion, the system-level parameters $\mathbf{p}_\alpha$ can be restricted on the Pareto-optimal front of α block during the system-level optimization.

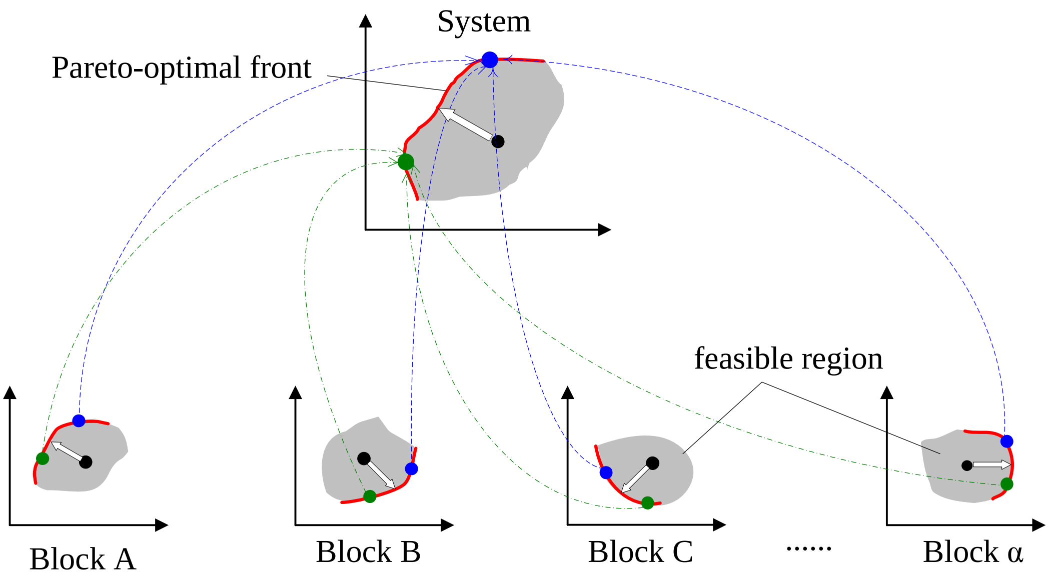


Figure 3.3: Pareto-optimal fronts in hierarchical optimization

3.3 Behavioral Modeling

By using a numerical simulator, circuit-level simulation based on the BSIM device model is accurate but requires huge computational time cost, which is not practical for the optimization process. As the number of design parameters and the complexity of analog/mixed signal circuits continue to increase, circuit-level simulation is and will continue to be a critical issue for designers. An alternative to the circuit-level simulation, more efficient *behavior-level simulation* is becoming popular and gradually accepted by analog designers. Behavior-level simulation uses *behavioral models*, "that reflect the terminal characteristics of functions realized by circuits rather than the circuit level (transistor-level) details of the circuit " [CPH94].

Behavioral models can be applied not only to bottom-up verification process but also to top-down design process. At the beginning design phase of a large-scale analog/mixed signal circuit, it is more efficient and economical to use behavior-level simulations to check design concepts and to explore several system architectures, instead of undertaking the detailed final circuit realization. At the final design phase, behavioral models can obviously reduce the verification time. This thesis focuses on the application of behavioral models in the top-down design process.

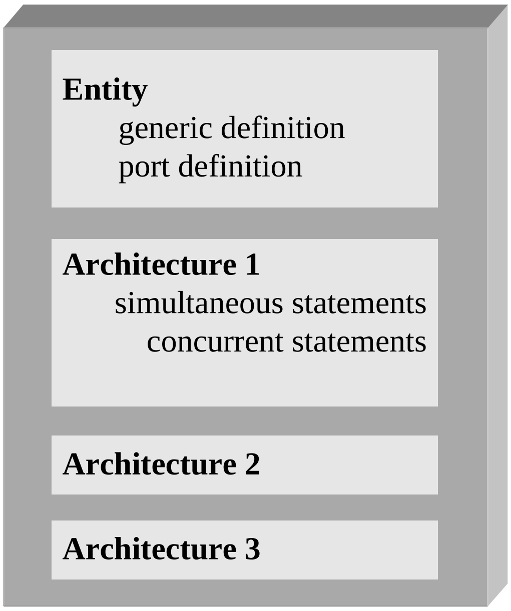
Should the behavioral model exactly or approximately represent the electronic functions of the circuit? A compromised decision has to be made between simulation cost and model accuracy. On the one hand, behavioral model describes a circuit functionality with relatively few components, making simulation much faster. On the other hand, a behavioral model includes more secondary circuit effects, presenting more accurate circuit behavior but making simulation slower. To achieve a good balance between computational efficiency and performance accuracy, the proper CAD tools are required which well cooperate with the behavioral models. Two most popular modeling methods in analog/mixed-signal design are briefly described in the following.

3.3.1 Modeling in Hardware Description Languages

Hardware description languages (HDLs) are programming languages designed to describe the behavior of physical devices and processes. Behavioral models written in various kinds of HDLs have their corresponding suitable simulators. Two mixed-signal hardware description languages, VHDL-AMS [VHD] and Verilog-AMS [Vera] are developed for high-level verification* on large-scale analog/mixed-signal systems. The commercial tools used are the Advance-MS simulator from Mentor Graphics [Men] for VHDL-AMS and the AMS Simulator from Cadence Design Systems [Cada] for Verilog-AMS. As indicated by their names, both mixed-signal simulators and HDLs can applied not only on digital circuits but analog circuits as well. The simulators are capable of simulation on pure circuit-level simulations, or on pure behavior-level simulations, or on mixed-level simulation (some blocks are circuit realization, while others are behavioral modules). Both digital and analog design share the same design environment and flow, which provides a straight-forward method to use behavioral models and improves design efficiencies. VHDL-AMS and Verilog-AMS describe complex continuous analog systems in form of differential algebraic equations (DAEs), which enables numerical simulators to evaluate the behavioral models like the usual device models. Additionally, they inherit the event-driven capability from digital simulator engines and allow analog event-driven models for analog simulation [CB99].

A VHDL-AMS model consists of one entity and one or more architectures, as shown in Fig 3.4(a). The *entity* specifies the interface of the model to the outside circuits. It includes the description of the model ports (the input/output pins that connect to other models) and the definition of its generic parameters. Fig 3.4(b) shows a mode of an ideal OP AMP with slew-rate limiting. There are two inputs (plus_in, minus_in) and one output (vout). VHDL-AMS introduces *quantity* to represent the unknown variables in the DAEs. The *architecture* contains the detailed implementation of the model, which can have three different styles: 1) Structure style is a description on the netlist connections of the elementary models. 2) Behavioral style is a transformation description from inputs to outputs by means of *concurrent statements* or/and *simultaneous statements*. 3) A style combines structural and behavioral elements. Simultaneous statements describe the continuous behavior by differential algebraic equations. Concurrent statements describe event-driven behaviors, e.g. concurrent signal assignment and process assignments like digital modeling. More details on VHDL-AMS modeling can be found in [CB99, APT02].

* "It is important to recognize that the AMS languages are primarily for verification. Unlike the digital languages, the AMS languages will not be used for synthesis in the foreseeable future because the only synthesis that is available for analog circuit is very narrowly focused" [KHC$^+$01].



```
Library ieee_proposed;
use ieee_poposed.electrical_systems_all;
entity opamp is;
   generic (gain: real : = 50;
   port {terminal plus_in, minus_in,
           output : electrical};
end entity amplifier;

architecture slew_limited of opamp is
   quantity v_in across plus_in to minus_in;
   quantity v_out across i_out through output;
   quantity v_amplified : voltage;
begin
   v_amplified == gain * v_in
   v_output == v_amplifed'slew(1.0e6, -1.0e6);
end architecture slew_limited;
```

(a) (b)

Figure 3.4: (a) Basic structure of VHDL-AMS Model (b) A VHDL-AMS Model of an ideal OP AMP with slew-rate limiting [APT02]

3.3.2 Modeling in Simulink

Simulink is a companion of MATLAB, developed by the MathWorks [Mat]. A user friendly graphical interface is available for building models, which makes it easier to maintain an overview of the system and subblocks. Simulink includes a comprehensive library of standard models, e.g. sinks, sources, linear and nonlinear components. Designers can create and customize their own blocks by assembling models directly using these standard blocks or coding in MATLAB m-files. Moreover, simulink has built-in ordinary differential equation (ODE) solvers, which are automatically configured multi-rate at run-time of each model, i.e. that different parts are sampled or updated at different rates.

A behavioral model example of an integrator in Simulink, which includes its non-idealities, is shown in Fig. 3.5 [MBF+03]. In the model, the limited slew rate and the limited bandwidth performances are modeled in “GBW & SR” MATLAB function. The limited gain is modeled through that only fraction α of the previous integrator output is added to each new input sample. The limited output range can be simply modeled by using a saturation block inside the feedback loop of the integrator. As we can seen from this example, all behavioral modules for sub-building block are already available in Simulink's library, the work for designers is to construct more complex models based on these fundamental cells.

Because of its strong capability and convenient use interface, Simulink is widely adopted in modeling tasks. However, Simulink is difficult to be integrated into the current popular analog/mixed-signal circuit design environments, e.g. Advance-MS [Men] and Virtuoso AMS [Cada]. While the behavioral modeling in HDLs is in the same design environment as circuit design, it is more convenient for designers to build up model, design circuit and simu-

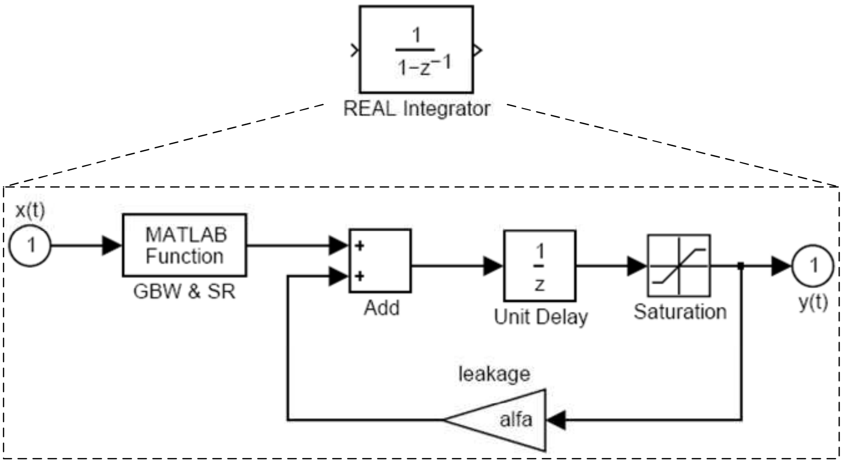


Figure 3.5: A simulink model for a real integrator [MBF+03]

late them in a single design environment. With the respective advantages and disadvantages of HDLs and Simulink, the optimal choice can always be made with respect to the task and the designers.

3.4 Proposed Hierarchical Optimization Flow

Based on the previous introduced hierarchical sizing flow, automatic sizing method, performance space exploration and behavioral modeling, a comprehensive hierarchical optimization methodology for large-scale analog/mixed signal circuits is proposed here and shown in Fig. 3.6. It consists mainly of four steps:

1. Starting with circuit-level realization of each building block, their proper testbenches and the corresponding extraction processes of performances are built firstly. These testbenches and the performance extractions are used not only in the performance space exploration process (bottom-up phase) but also in the automatic sizing process on circuit level (top-down phase).
2. Through applying PSE method to each block respectively, their individual Pareto-optimal front can be obtained. At system level, the whole circuit is modeled in HDLs or Simulink. In addition to the description of circuit function in these behavioral models, Pareto-optimal fronts are embedded so that the variations of the system-level parameters are restricted on their own Pareto-optimal fronts during the optimization process at system level.
3. Since the system performance can be quickly evaluated through simulation based on the behavioral models, a simulation-based automatic sizing process is feasible to generate optimized system-level parameters in acceptable time cost. These obtained values of the system-level parameters are directly propagated to the specifications of the circuit-level performances.
4. Afterwards, each building block can be designed individually and in parallel. The simulation-based automatic sizing process on each block can be accomplished with an acceptable time cost.

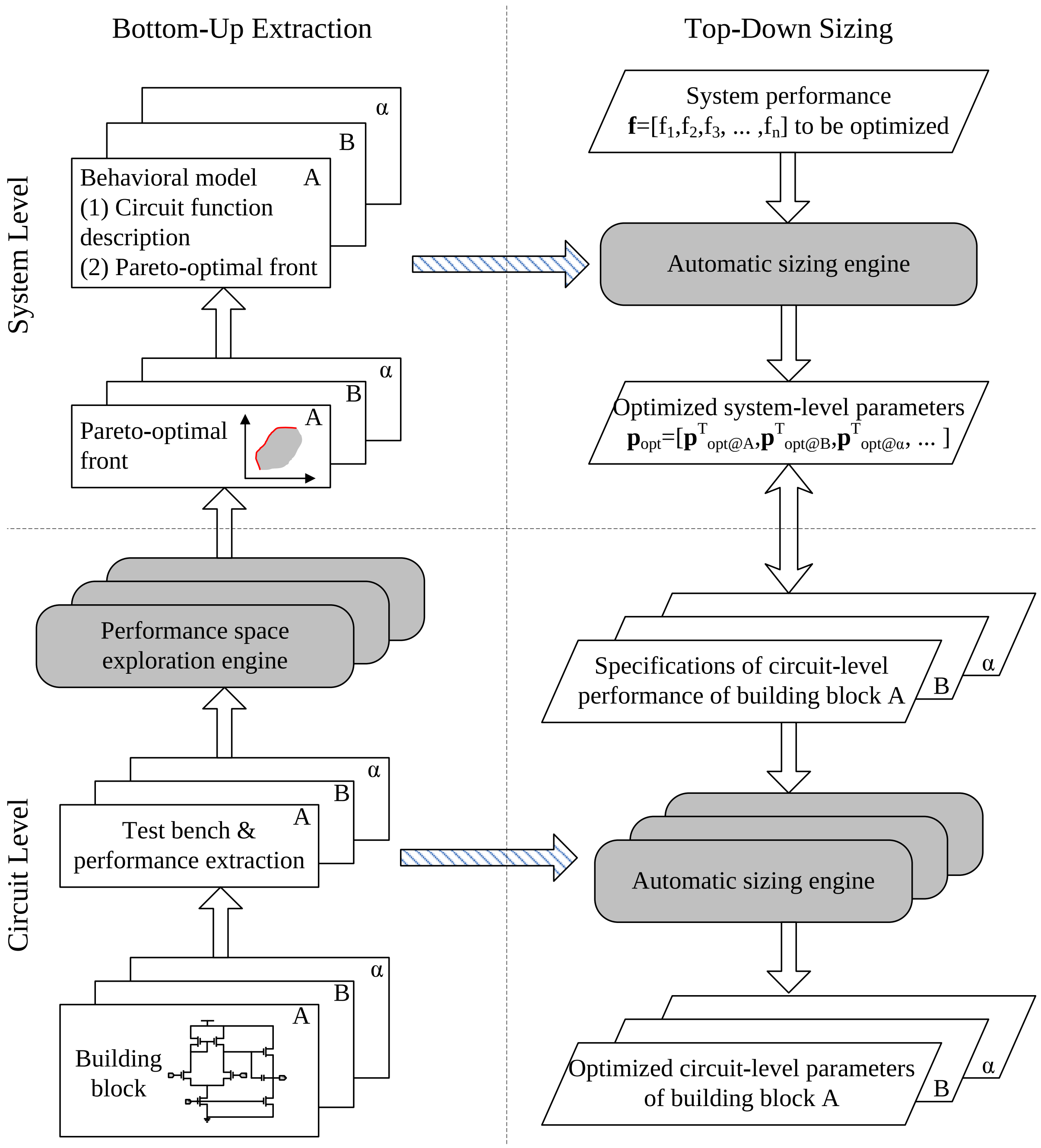

Bottom-Up Extraction
Top-Down Sizing
System Level
Circuit Level
α
B
A
Behavioral model
(1) Circuit function description
(2) Pareto-optimal front
Pareto-optimal front
Performance space exploration engine
Test bench & performance extraction
Building block
System performance
f=[f1,f2,f3, ... ,fn] to be optimized
Automatic sizing engine
Optimized system-level parameters
popt=[pTopt@A,pTopt@B,pTopt@α, ...]
Specifications of circuit-level performance of building block A
Automatic sizing engine
Optimized circuit-level parameters of building block A

Figure 3.6: Proposed hierarchical optimization design flow

In summary, the proposed hierarchical optimization methodology is characterized by a bottom-up extraction process of circuit capability and a top-down hierarchical automatic sizing process.

3.5 Hierarchical Optimization based on Worst-Case-Aware Pareto-Optimal Front

The Pareto-optimal fronts extracted from the most current PSE methods [SGA03, SGA04, EMG05, MSGS05] are nominal Pareto-optimal fronts, which consider only design parameters. As device sizes shrink, the effect due to the variations of statistical parameters and of operation parameters becomes more and more pronounced in analog design. The nominal Pareto-optimal fronts cannot represent the actual capability of the circuit after fabrication. Generally, it will result in a poor yield if a nominal design point lies in the tail of the performance distribution. Therefore, a *yield-aware* Pareto-optimal front (points on the front guaranteeing a fixed yield number) would be very useful for improvement on production yield.

In [TTR06], a solution generates firstly the nominal Pareto-optimal fronts, and then combines with Monte Carlo approximation to compute *yield-aware* Pareto-optimal fronts. Compared to the high computational cost Monte Carlo analysis, *worst-case analysis* is applied to extract a *worse-case-aware* Pareto-optimal front (design point on the front guaranteeing a target yield number even under the worst-case operation condition) in this thesis.

Worst-Case Analysis The following discussion of worst-case analysis is based on one performance. Worst-case analysis takes into account fluctuations in the fabrication process and changes in the operating condition. Therefore, statistical parameters and operational parameters are considered through the worst-case analysis, whereas design parameters remain constant. In consequence,

$$\mathbf{d}^* = \text{const} \quad \text{and} \quad \mathbf{c}(\mathbf{d}^*) = \text{const} \tag{3.9}$$

holds throughout the worst-case analysis.

Worst-case analysis (WCA) [AGC94] intends to find the lowest performance value that is obtained for a given design parameter set $\mathbf{p}^*$, a given tolerance region T_θ of operation parameters and a given tolerance region T_s of statistical parameters:

$$\min_{\mathbf{s},\theta} \mathbf{f}(\mathbf{d}^*, \mathbf{s}, \theta) \quad s.t. \quad \mathbf{s} \in T_s,\, \theta \in T_\theta, \tag{3.10}$$

$$\text{where} \qquad T_\theta = \{\theta |\, \theta_L \leq \theta \leq \theta_U\} \tag{3.11}$$

$$T_s = \{\mathbf{s}|\, \|\mathbf{s}\|^2_{\mathbf{C}} = (\mathbf{s}-\mathbf{s}_o)^T \mathbf{C}^{-1}(\mathbf{s}-\mathbf{s}_0) \leq \beta_w^2\}. \tag{3.12}$$

θ_L, θ_U define the lower and upper boundaries of the operation parameters. The circuit robustness β_w describes a weighted distance of the nominal design point $\mathbf{s}_0$ in the space of the statistical parameters $\mathbf{s}$. $\|\mathbf{s}\|^2_{\mathbf{C}}$ represents a tolerance ellipsoid according to the statistical parameter distribution, where $\mathbf{C}$ is the covariance matrix of statistical parameters. As we know, there always exists a matrix $\mathbf{G}$, such that the transformed statistical parameter $\mathbf{G}(\mathbf{s}-\mathbf{s}_0)$ follows the distribution of $N(0,1)$ [Pap91], as depicted in Fig. 3.7 and given by

$$\|\mathbf{s}-\mathbf{s}_0\|_{\mathbf{G}} = \|\mathbf{G}(\mathbf{s}-\mathbf{s}_0)\| = \sqrt{(\mathbf{s}-\mathbf{s}_0)^{\mathrm{T}}\mathbf{G}^{\mathrm{T}}\mathbf{G}(\mathbf{s}-\mathbf{s}_0)} \tag{3.13}$$

$$\mathbf{C}^{-1} = \mathbf{G}^{\mathrm{T}}\mathbf{G}. \tag{3.14}$$

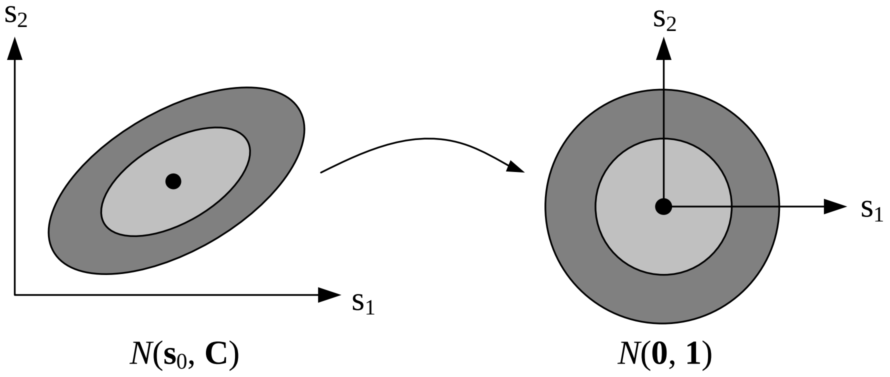


Figure 3.7: Transformation from $\mathbf{s} \sim N(\mathbf{s}_0, \mathbf{C})$ into $\mathbf{s} \sim N(0,1)$

Without loss of generality, the statistical parameter **s** is assumed as

$$\mathbf{s} \sim N(0,1), \tag{3.15}$$

whose statistical distribution is depicted in a unit cycle instead of ellipsoid throughout this thesis.

Using $\mathbf{s}_w$ to present the unique worst-case parameter set and f_w to denote the corresponding performance value at $\mathbf{s}_w$:

$$f_w = m(\mathbf{s}_w). \tag{3.16}$$

Then f_w is the smallest performance value that can be achieved for each **s** in the volume $\|\mathbf{s}\|^2 \leq \beta_w^2$. The definition of worst-case analysis can be additionally interpreted as following, visualized in Fig. 3.8:

- "finding the greatest lower bound of the performance that violates the specified circuit robustness β_w for all operation conditions,
- choosing a lower specification that results in at least the given robustness β_w under worst-case conditions,
- picking the smallest performance value in the given sphere $\|\mathbf{s}\|^2 = \beta_w^2$" [Sch04].

The robust measure β_w is also called the *worst-case distance*, which is the positive distance between the nominal parameter set $\mathbf{s}_0$ (origin in Fig. 3.8) and the worst-case parameter set $\mathbf{s}_w$, if the specification is fulfilled for all operation conditions at the nominal parameter set. Otherwise, it is the negative distance between the nominal parameter set and the worst-case parameter set $\mathbf{s}_w$. The worst-case distance is related to the robustness of a specification concerning disturbances in fabrication and operation. According to [Gra93], the yield of production can be derived from the worst-case distance. The larger the distance between the worst-case point and the nominal point is, the fewer produced circuits will lie behind the worst-case point and violate the specification. Therefore, the yield increases with the worst-case distance, whose relationship is listed in Tab. 3.1.

Table 3.1: Yield estimation by worst-case distance

β_w	-4	-3	-2	-1	0	1	2	3	4
Yield [%]	0.01	0.13	2.28	15.87	50.00	84.13	97.73	99.87	99.99

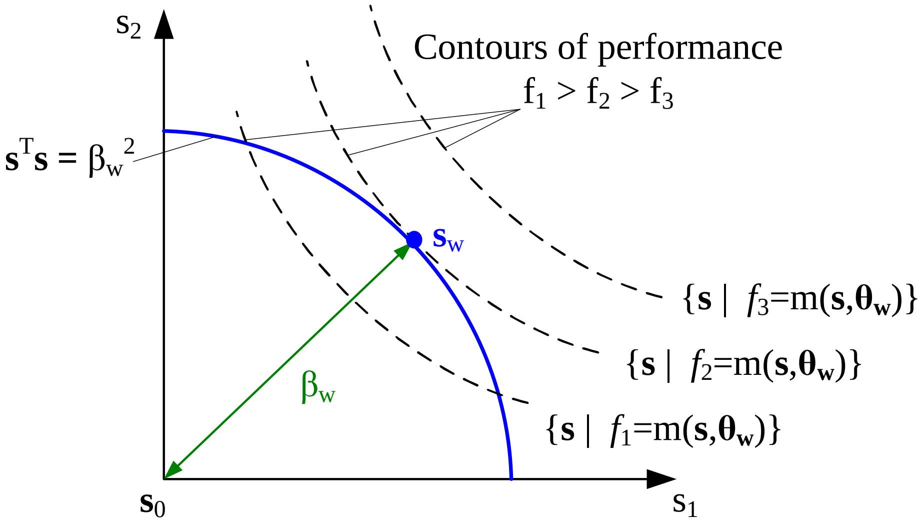


Figure 3.8: Visualization of the worst-case analysis definition for a given robustness β_w

According to a target yield value, the worst-case distance can be selected from Tab. 3.1. For instance, $\beta_w = 3$ corresponds to a yield of 99.87% that can be guaranteed for each performance. Then, the worst-case parameter set $\mathbf{s}_w$ can be calculated through Equation 3.10, and the corresponding worst-case performance value f_w is evaluated. Through applying the worst-case analysis (WCA) to each efficient point of the nominal Pareto-optimal front, their corresponding efficient points of the worst-case-aware Pareto-optimal front can be obtained. Fig. 3.9 illustrates the extraction process for a *worst-case-aware* Pareto-optimal front. There are two individual WCAs for f_1 and f_2 on each efficient nominal point. The respective worst-case values of f_1 and of f_2 build together an efficient point of the worst-case-aware Pareto-optimal front, although both values are not generated at the same time in practice. Instead of the nominal Pareto-optimal front, the worst-case-aware Pareto-optimal front is embedded into the behavioral modeling during the proposed hierarchical optimization methodology, the optimization results represent the actual circuit performances with a target yield after fabrication.

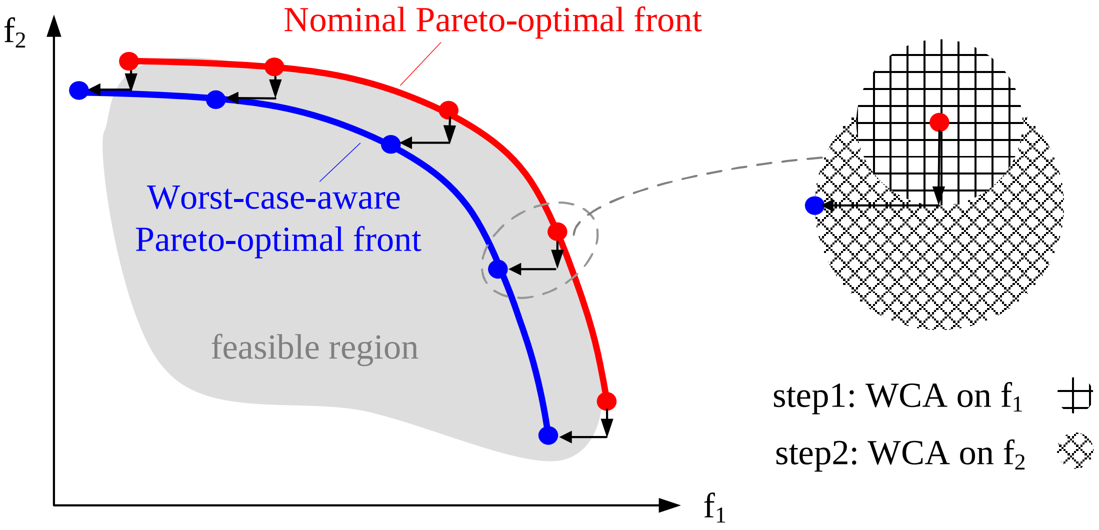


Figure 3.9: Worst-case analysis on nominal Pareto-optimal front

Chapter 4

Hierarchical Optimization of Charge-Pump Phase-Locked Loops

Phase-locked loops (PLLs) provide well-timed on-chip clock signals for various applications in areas such as communications, wireless systems, dynamic random access memory and disk drive electronics. Compared to the design concept of phase locking has almost keep same since it was invented in the 1930s, the design and implementation of PLLs have always developed with the progress of the ICs. The requirements on PLLs becomes more and more stringent, such as accurate clock timing, less power consumption and small area, robust noise rejection. Due to the PLL's circuit complexity, the optimization task of PLL is big challenge even for experienced analog designers. Currently, there are two strategies for optimization to design high performance PLLs. One way is through heuristic circuit topologies [Man96, Man03, WTHN04, CA05, KLK05]. The other way is through tuning design parameters for a given circuit topology [CPH$^+$03, TVRM04, ZMG$^+$05, TTR06, ZMGS06]. In this thesis, the second way is adopted. By using the proposed hierarchical optimization methodology in Chapter 3, an efficient design process for PLLs can be realized in an acceptable time cost.

In this chapter, Sec. 4.1 introduces the fundamentals of PLLs. Sec. 4.2 reviews the analysis on PLL systems and Sec. 4.3 lists out some performances of PLL and summarizes the major design trade-offs. Sec. 4.4 shows the details of the proposed hierarchical optimization method on a charge-pump phase-locked loop (CPPLL). Sec. 4.5 gives a comprehensive performance space exploration (PSE) on a whole CPPLL system. Finally, Sec. 4.6 concludes.

4.1 CPPLL Fundamentals

4.1.1 PLL Introduction

The primary function of phase-locked loops is to generate an output clock whose phase is locked to the phase of the input reference clock. In contrast to conventional feedback circuits operate on voltage/current amplitudes or their changing rate, PLL feedback systems work on signal's phases. The operation of PLLs is briefly described as follows. More details can be found in [Raz96, Raz01, Bes03].

The basic building blocks of a typical PLL consists of a phase detector (PD), a low-pass loop filter (LPF), a voltage-controlled oscillator (VCO) and a frequency divider (D) as shown in Fig. 4.1. In the forward path, the PD produces an error output signal $\Delta\phi$ based on the phase difference between the phase of the feedback clock ϕ_{fb} and the phase of reference signal ϕ_{ref}. A small frequency difference accumulates over time, which results in an increasing phase error. The error signal $\Delta\phi$ is filtered by the LPF and also transferred to a control voltage/current (V/I) to drive the subsequent VCO. According to the amplitude of the voltage/current, the VCO generates an output clock signal with the corresponding frequency. In the feedback path, the output frequency of the oscillation signal is scaled down by a factor N through the frequency divider. The scaled-down signal will compare to the reference input signal at the PD block. Through the control loop, the output frequency of VCO is adjusted till both phases ϕ_{ref} and ϕ_{out} align with each other. PLLs experience two work states during the lock process. When f_{out} approaches the desired output frequency asymptotically, the process is called as *nonlinear acquisition process*. When the frequency and the phase of the output clock are synchronized with the reference clock, i.e. a multiplies output frequency f_{out}: $f_{\text{out}} = N f_{\text{ref}}$, the output phase is locked with the input phase. This process is called as *linear locked-in state*.

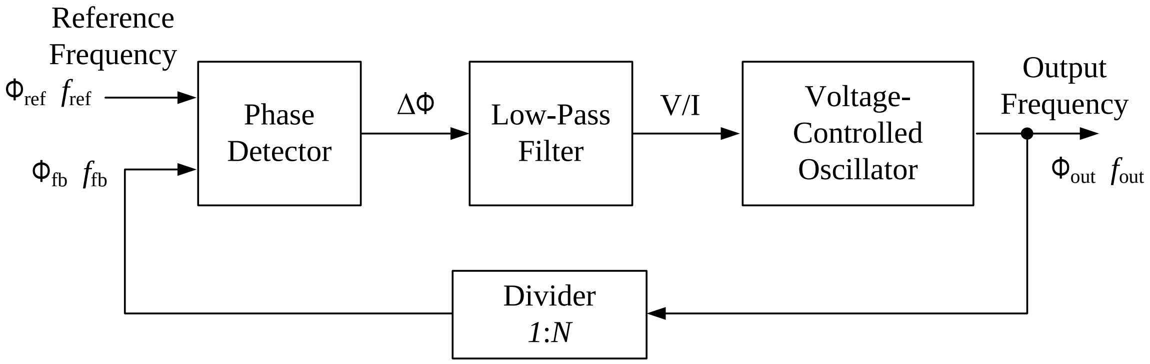


Figure 4.1: A block diagram for a typical PLL

4.1.2 CPPLL Building Blocks

Charge-pump phase-locked loops (CPPLLs) are widely used in most PLL systems, since they provide a theoretical zero static phase error (offset), an extended tracking range and a frequency-aided acquisition. A typical 3$^{\text{rd}}$-order CPPLL is one of the simplest and most effective circuit topology, which includes a phase frequency detector, a charge pump, a 2$^{\text{nd}}$-order passive RC filter, a voltage-controlled oscillator and a frequency divider, whose block diagram is shown in Fig. 4.2. Each building block is briefly described below.

4.1.2.1 Phase Frequency Detector (PFD)

PD can only detect the phase difference between two input signals and produce an error signal that is proportional to the phase difference. However the PD is insensitive to the frequency difference of the input signals. When the frequency of the feedback signal is far from the

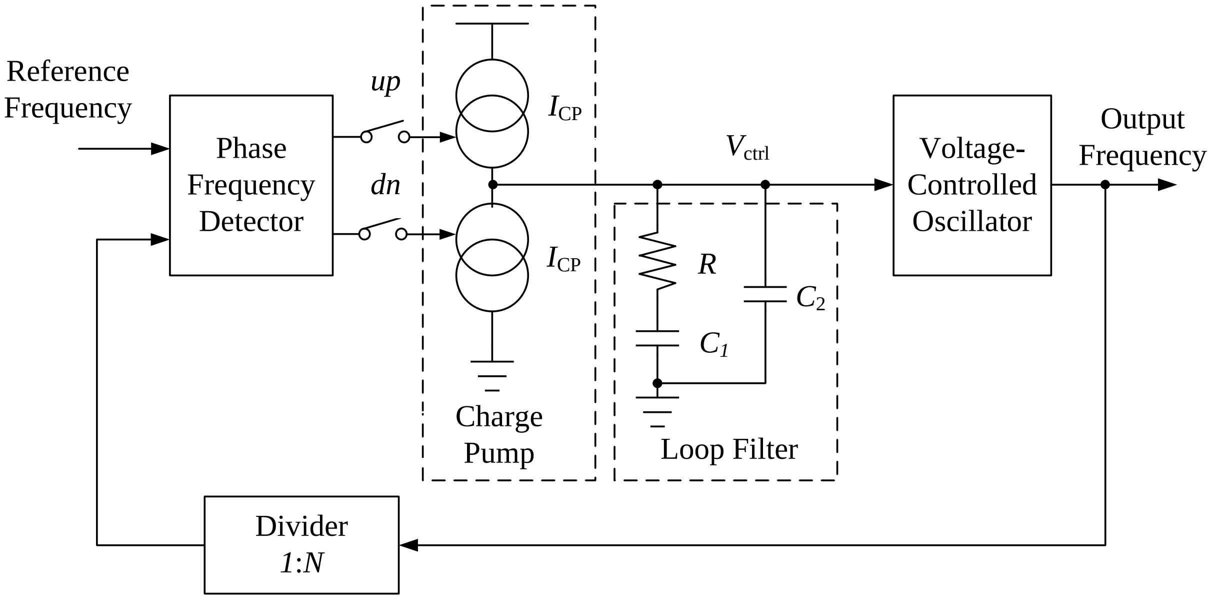
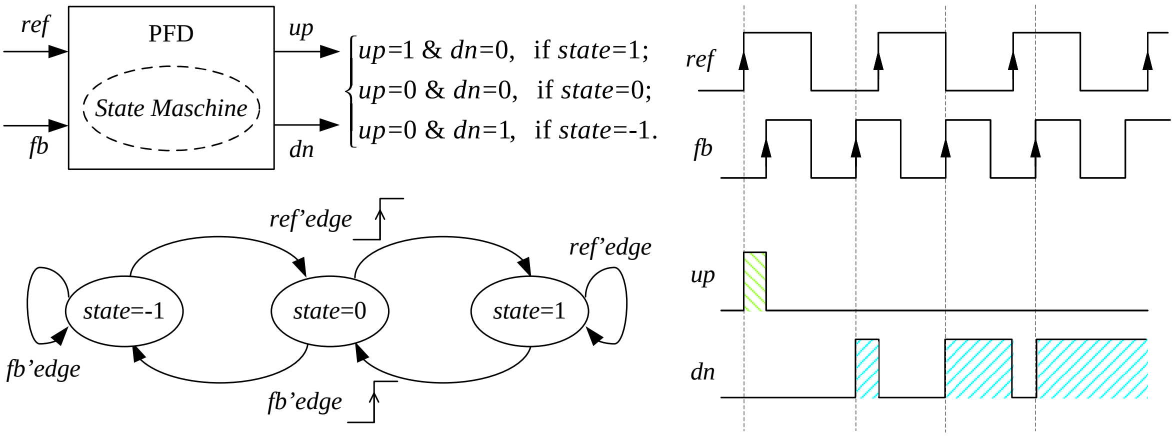


Figure 4.2: Building blocks of a 3^{rd}-order CPPLL

frequency of the reference signal, the PLL may lock at a wrong frequency, e.g. $f_{\text{out}} = 0.5 * N f_{\text{ref}}$. The problem is due to an inadequate acquisition range of the PD. A phase frequency detector (PFD) is used to tackle the problem. PFD can detect the difference not only of the phase but of the frequency as well, so that the PLL can lock under any condition if the PLL system itself is stable. Therefore, PFD is the preferred comparator type compared to other phase detectors as Multiplier PDs or JK-Flip-Flops.



(a) (b)

Figure 4.3: (a) State machine of a PFD (b) Clock diagram example

A typical PFD usually has three logical work-states, as shown in Fig. 4.3(a). Initially, PFD is in the “ground” state, i.e. $state = 0$, where signals $up = dn = 0$. If a rising transition of ref comes earlier than fb, then PFD becomes “charging" state, i.e. $state = 1$, where signal $up = 1$, signal $dn = 0$. The PFD remains in this state until a rising transition occurs on fb earlier than on ref, then the PFD returns to “ground”. If the next rising edge of fb is still ahead of ref, the PFD jumps from “ground" to “discharging", i.e. $state = -1$, where signal $up = 0$, signal $dn = 1$. Fig. 4.3(b) shows the switching processes of signals up and dn according to the rising transitions of both input signals ref and fb. Note that signals up and dn cannot be active at the same time.

4.1.2.2 Charge Pump and Loop Filter (CP & LF)

As shown in Fig. 4.2, a charge pump (CP) consists of two current sources: one source current and one sink current. In ideal case, the amplitude of both current sources are identical. The CP charges and discharges the subsequent loop filter according to the control signal from PFD. When signal up (dn) is active (=1), the current flows into (out of) the loop filter. In consequence, the LF’s output voltage will be pulled up (down).

The loop filter in PLL is usually a low-pass passive filter consisting of a resistor R in series with a capacitor C_1. Through the combination operation of the CP and of the LF, the discrete digital pulse signal from the PFD is firstly converted to a current pulse, and then transferred to a continuous voltage signal, which determines the output frequency of the subsequent VCO block. Since the voltage signal directly modulates the VCO frequency, any dithering of the voltage will introduce excessive jitters on the VCO output signal. As well known, the voltage across a capacitor cannot instantaneously change. Due to often switching on/off actions on the source current and the sink current in CP, a voltage ripple on the output node of the loop filter occurs with a value $I_{\mathrm{CP}} \cdot R$ at the rising edge of each PFD pulse, while another ripple occurs with same value but in the opposite direction at the falling edge of the PFD pulse. In order to suppress the undesired voltage ripples, another capacitor C_2 is placed in parallel with the R and C_1 network as shown in Fig. 4.2.

In the PLL phase-controlled feedback system, a 1$^{\mathrm{st}}$-order RC passive filter introduces one pole and one zero. A VCO block is a phase integrator which contributes one intrinsic pole. Hence the whole PLL feedback system has two poles and one zero together. A 2$^{\mathrm{nd}}$-order PLL system is unconditional stable. Since the additional capacitor C_2 introduces another pole, the PLL system increases from 2$^{\mathrm{nd}}$-order to 3$^{\mathrm{rd}}$-order. The degradation of system stability due to the additional pole has to be taken into account, which will be more discussed in Section 4.2.

4.1.2.3 Voltage-Controlled Oscillator (VCO)

How to adjust the phase of PLLs? The principle way is to tune the frequency of the PLL’s output signal, i.e. the output frequency of the voltage-controlled oscillators (VCOs). VCOs can be realized in various technologies, e.g. CMOS ring oscillators or LC oscillators. LC-VCO has better phase noise (jitter) performance for a given power drain [HL99] and can achieve very high frequencies. However, CMOS ring oscillators are capable of monolithic integration and cost less power compared to LC-VCOs.

A five-stage CMOS single-ended ring oscillator is shown in Fig. 4.4. In this kind of VCO, the input voltage V_{ctrl} controls the current through the delay elements, thus determines the delay time of each stage and ultimately determines the output oscillation frequency. An ideal VCO

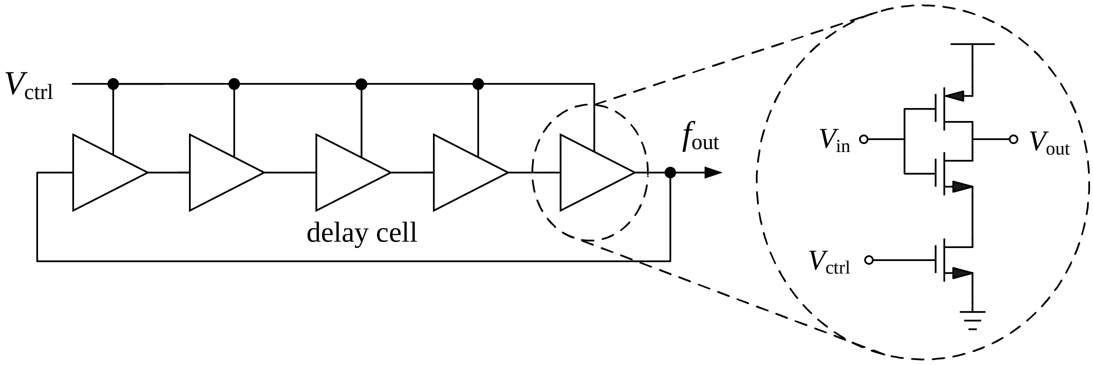


Figure 4.4: A five-stage ring oscillator

generates a periodic signal whose frequency is a linear and limited function of the controlling voltage, as shown in Fig. 4.5. The output frequency f_{out} can be expressed as

$$f_{\text{out}} = (V_{\text{ctrl}} - V_{\text{min}}) \cdot K_{\text{VCO}} + f_{\text{min}}, \tag{4.1}$$

where K_{VCO} is defined as the linear gain of VCO, i.e.

$$K_{\text{VCO}} = \frac{f_{\text{max}} - f_{\text{min}}}{V_{\text{max}} - V_{\text{min}}}. \tag{4.2}$$

Each VCO has its output frequency range, here f_{min} is the minimal limit and f_{max} is the maximum limit, V_{max} and V_{min} correspond to the minimal and maximum input voltages and V_{ctrl} is the input control voltage of VCO, i.e. the output voltage of the loop filter.

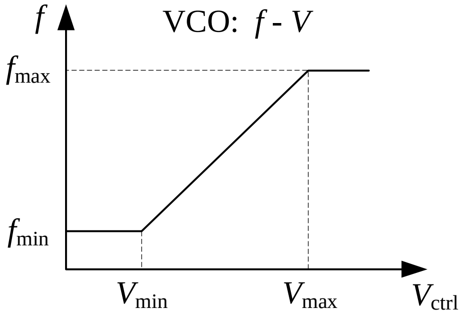


Figure 4.5: Frequency *vs.* input control voltage for an ideal VCO

4.1.2.4 Divider (D)

As a crystal is not capable to fulfill the increasing requirement on high speed circuits, PLLs now take the role to provide a reference clock signal for other circuits in a chip. By appropriate configuration of the divide ratio, PLLs can generate an output signal which has much higher frequency than the input frequency and inherits much more stability. The divider's value can be integer or fraction according to the reference frequency and the desired output frequency.

Fig 4.6 shows the circuit diagram of a $1/4$ divider. There are two D flip-flops (DFFs) connecting in serial. The output of the first DFF is the input of the second DFF, and the inversion of the second DFF feeds back to the first DFF. Both DFFs are driven by a same input clock signal and are asynchronously reset.

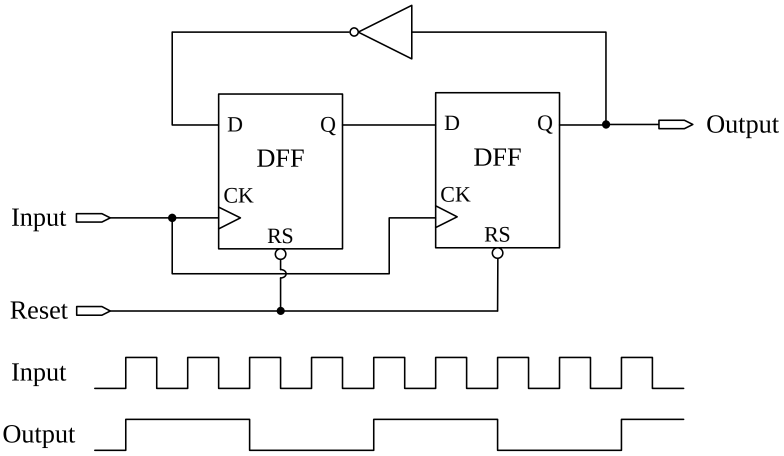


Figure 4.6: Circuit diagram of a 1/4 divider

4.2 Analysis Methods on PLL System

Same as analysis on conventional feedback control systems, *s-domain* analysis (linear approximation) in [M.G80] is usually used to gain the intuition when PLLs work in lock-in state. *Impulse invariance* method in [HS88] and *state space* analysis in [HBMM04] are used to estimate the properties when PLLs work in nonlinear acquisition process including the discrete sampling nature of PFD operation. The following equations in this section are excerpted from these references.

4.2.1 *s*-domain Analysis

When a CPPLL works in lock-in state, there is a small phase difference between the reference input signal and the scaled-down feedback signal. Meanwhile, the reference frequency is much higher than the loop bandwidth of the whole CPPLL. Hence, the sampling operation of PFD can be approximated as a continuous-time action. A linear model in *s*-domain is applied to analyze the CPPLL system. The linear model in Fig. 4.7 shows the transfer function of each building block. The model provides the overall transfer function for the phase $\phi_{\mathrm{out}}(s)/\phi_{\mathrm{ref}}(s)$.

A subtracter represents the comparison operation of the PFD block. The gain of PFD along with the CP can be expressed as $I_{\mathrm{CP}}/2\pi$; the transfer function of a 2^{nd}-order LF, $F_{\mathrm{LF}}(s)$, can be derived using linear analysis and is equal to

$$F_{\mathrm{LF}}(s) = \frac{s+\frac{1}{RC_1}}{C_2 s(s+\frac{C_1+C_2}{RC_1C_2})} = \frac{\frac{s}{\omega_z}+1}{(C_1+C_2)(\frac{s}{\omega_{p3}}+1)}, \tag{4.3}$$

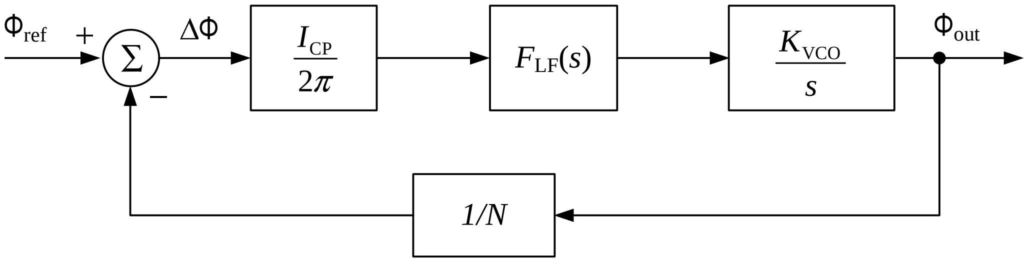


Figure 4.7: Linear model of a CPPLL

where the loop filter has one zero at $\omega_z = \frac{1}{RC_1}$ and one pole at $\omega_{p3} = \frac{C_1+C_2}{RC_1C_2}$. The VCO is usually assumed as an ideal integrator with the linear gain K_{VCO}. The open-loop transfer function $LG(s)$ of the 3rd-order CPPLL is therefore equal to

$$LG(s) = \frac{I_{\mathrm{CP}}}{2\pi}\frac{s+\frac{1}{RC_1}}{C_2 s(s+\frac{C_1+C_2}{RC_1C_2})}\frac{K_{\mathrm{VCO}}}{s}\frac{1}{N} = \frac{K_{\mathrm{VCO}}I_{\mathrm{CP}}}{2\pi N C_2}\frac{s+\frac{1}{RC_1}}{s^2(s+\frac{C_1+C_2}{RC_1C_2})} = K_{\mathrm{s}}\frac{s+\omega_z}{s^3+s^2\omega_{p3}}, \tag{4.4}$$

where $K_{\mathrm{s}} = \frac{K_{\mathrm{VCO}}I_{\mathrm{CP}}}{2\pi N C_2}$. The PLL open-loop transfer function has one zero and three poles in which two poles are at the origin. The corresponding closed-loop transfer function $H(s)$ of the 3rd-order CPPLL can be expressed as

$$\begin{aligned} H(s) &= \frac{\phi_{\mathrm{out}}(s)}{\phi_{\mathrm{ref}}(s)} = \frac{N\cdot LG(s)}{1+LG(s)} \\ &= \frac{K_{\mathrm{VCO}}I_{\mathrm{CP}}}{2\pi C_2}\frac{s+\frac{1}{RC_1}}{s^3+\frac{C_1+C_2}{RC_1C_2}s^2+\frac{K_{\mathrm{VCO}}I_{\mathrm{CP}}}{2\pi N C_2}s+\frac{K_{\mathrm{VCO}}I_{\mathrm{CP}}}{2\pi N R C_1 C_2}} \\ &= K_{\mathrm{s}}N\frac{s+\omega_z}{s^3+s^2\omega_{p3}+sK_{\mathrm{s}}+K_{\mathrm{s}}\omega_z}. \end{aligned} \tag{4.5}$$

s-domain analysis is a fundamental method to know the properties of PLL system in lock-in state, such as a step response to noise interference. However, *s*-domain analysis is initially applicable to continuous-time systems. Because the PFD samples and compares the input signals at discrete timing, *s*-domain analysis is not sufficient to predict the properties of the CPPLL system. Moreover, Equation 4.5 cannot represent the CPPLL's nonlinear acquisition behavior.

4.2.2 Impulse Invariance Analysis

To take into account the PFD's discrete-time sampling nature, *z*-domain analysis has been applied for a 2nd-order CPPLL in [HS88] and for a 3rd-order CPPLL in [JYM$^+$07]. *Impulse invariant transformation* is one common method to map the transfer function from *s*-domain to *z*-domain. For the blocks CP, LF and VCO, the *z*-domain descriptions can be directly transferred from their corresponding *s*-domain descriptions respectively, where $s = j\omega$ is substituted by $z = e^{j\omega T}$ and T is the period of the reference signals. According to the clock diagram shown in Fig. 4.3(b), the output signal of PFD, i.e. phase error $\Delta\phi$, is a discrete-time event sequence, whose pulse width is approximately equal to the time difference between the rising edges of ref and fb. Therefore, the $\Delta\phi$ can be accurately modeled as weighted impulse to drive the subsequent charge pump, whose z-domain description is $\frac{\alpha T}{2\pi}$, as shown in Fig. 4.8. The value of α can be calculated by the latter Equation 4.8, i.e. $\alpha = e^{-\omega_{p3}T}$.

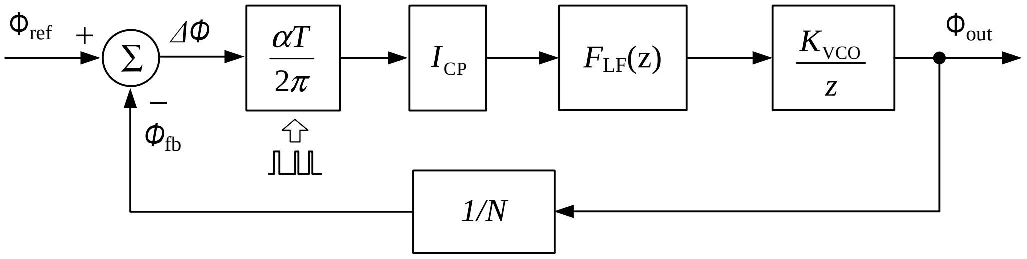


Figure 4.8: Discrete model of a CPPLL

The detailed transfer steps are described as follows. When input of the continues-time system $LG(s)$ in Equation 4.4 is a discrete-time event sequence, the corresponding impulse response $LG(t)$ can be calculated by the *Inverse Laplace Transform*:

$$LG(t) = \frac{K_S}{\omega_{p3}^2}[\omega_{p3}\omega_z t - (\omega_{p3} - \omega_z)e^{-\omega_{p3}t} + (\omega_{p3} - \omega_z)]. \tag{4.6}$$

By n-number sampling in T interval time, the sampled impulse response $LG(nT)$ is

$$LG(nT) = \frac{K_S}{\omega_{p3}^2}[\omega_{p3}\omega_z(nT) - (\omega_{p3} - \omega_z)e^{-\omega_{p3}nT} + (\omega_{p3} - \omega_z)]. \tag{4.7}$$

By applying z transformation on $LG(nT)$, the discrete z-domain transfer function for the open-loop equation of the 3$^{\text{rd}}$-order CPPLL can be derived as

$$\begin{aligned} LG(z) &= \frac{K_{VCO} I_{CP} R C_1}{N\omega_{ref}(C_1+C_2)} \cdot \frac{z^2[\frac{C_1}{C_1+C_2}(1-e^{-\omega_{p3}T})+\omega_z T] - z[\frac{C_1}{C_1+C_2}(1-e^{-\omega_{p3}T})+\omega_z T e^{-\omega_{p3}T}]}{z^3 - z^2(2+e^{-\omega_{p3}T}) + z(1+2e^{-\omega_{p3}T}) - e^{-\omega_{p3}T}} \\ &= K_z \frac{z^2[\frac{K_c}{K_c+1}(1-\alpha)+\omega_z T] - z[\frac{K_c}{K_c+1}(1-\alpha)+\omega_z T\alpha]}{z^3 - z^2(2+\alpha) + z(1+2\alpha) - \alpha}. \end{aligned} \tag{4.8}$$

where $\omega_{ref} = 2\pi/T$, $k_z = \frac{K_{VCO} I_{CP} R C_1}{N\omega_{ref}(C_1+C_2)}$ and $K_c = \frac{C_1}{C_2}$. The closed loop transfer function $H(z)$ is equal to

$$\begin{aligned} H(z) &= \frac{\phi_{out}(z)}{\phi_{ref}(z)} = \frac{N \cdot LG(z)}{1+LG(z)} \\ &= NK_z \frac{z^2[\frac{K_c}{K_c+1}(1-\alpha)+\omega_z T] - z[\frac{K_c}{K_c+1}(1-\alpha)+\omega_z T\alpha]}{z^3 + z^2[k_z(\frac{K_c(1-\alpha)}{K_c+1}+\omega_z T) - \alpha - 2] + z([2\alpha+1-k_z(\frac{K_c(1-\alpha)}{K_c+1}+\omega_z T)] - \alpha}. \end{aligned} \tag{4.9}$$

4.2.3 State Space Analysis

Compared to s-domain and z-domain analysis in the last two sections, state space analysis in [HBMM04] represents the PLL in time-domain by using the difference equations for signal phases and state-space variables for the node voltages. The difference equation describes the relationship between the input phase $\phi_{\text{ref}}(t)$ and the output phase $\phi_{\text{out}}(t)$, which is given by

$$\Delta\phi = \phi_{\text{ref}} - \phi_{\text{out}}, \tag{4.10}$$

where

$$\begin{aligned} \phi_{\text{ref}}(t) &= \phi_{\text{ref}}(0) + \omega_{\text{ref}} t & (4.11) \\ \phi_{\text{out}}(t) &= \phi_{\text{out}}(0) + \omega_{\text{fr}} t + K_{\text{VCO}} \int_0^t v_{\text{ctrl}}(\tau) d\tau. & (4.12) \end{aligned}$$

$\phi_{\text{ref}}(0)$ is the initial input phase condition and ω_{ref} is the input reference frequency. $\phi_{\text{ref}}(out)$ is the initial output phase condition and ω_{fr} is the free oscillation frequency of VCO. K_{VCO} is the gain of VCO, as expressed in Equation 4.2.

If the feedback signal is behind the reference signal, the PFD generates a positive phase error, i.e. $\Delta\phi > 0$, the capacitor in loop filter is consequently charged. When the feedback signal leads ahead of the reference signal, the PFD generates a negative phase error, i.e. $\Delta\phi < 0$, the capacitor is discharged. The charging/discharging current can be represented by

$$i_p = \begin{cases} \pm I_{\text{CP}}, & \text{if } 0 \leq \text{t} \leq t_p \\ 0, & \text{if } t_p \leq t \leq T_-, \end{cases} \tag{4.13}$$

where t_p means the duration of the charging or the discharging period, which is calculated by

$$t_p = \frac{|\Delta\phi|}{\omega} = \frac{|\Delta\phi| T_-}{2\pi}. \tag{4.14}$$

The PFD's output will update per $T\_$ time period. The value of $T\_$ varies with the PLL's locking process, principally equal to the time interval between two sequential rising edges either of the reference or of the feedback signal. Fig. 4.9(a) shows the definitions of the variables t_p and $T\_$.

By charging or discharging process, the output voltage of the loop filter goes up or down consequently. These voltages in the RC network can be accurately described by the following differential equations:

$$\begin{aligned} \frac{dV_{\text{c}}}{dt} &= \frac{V_{\text{ctrl}} - V_{\text{c}}}{RC_1} & (4.15) \\ \frac{dV_{\text{ctrl}}}{dt} &= \frac{V_{\text{c}} - V_{\text{ctrl}}}{RC_2} + \frac{i_{\text{p}}}{C_2}, & (4.16) \end{aligned}$$

where V_{ctrl} and V_c are the two state-space variables. V_{ctrl} is the control voltage for VCO, i.e. the output voltage of the loop filter. V_c is the voltage on the capacitor C_1, as shown in Fig. 4.9(b). Using from Equation 4.11 to 4.16, an accurate behavior of CPPLLs can be interpreted for the nonlinear acquisition process and the linear lock-in process as well. More details can be found

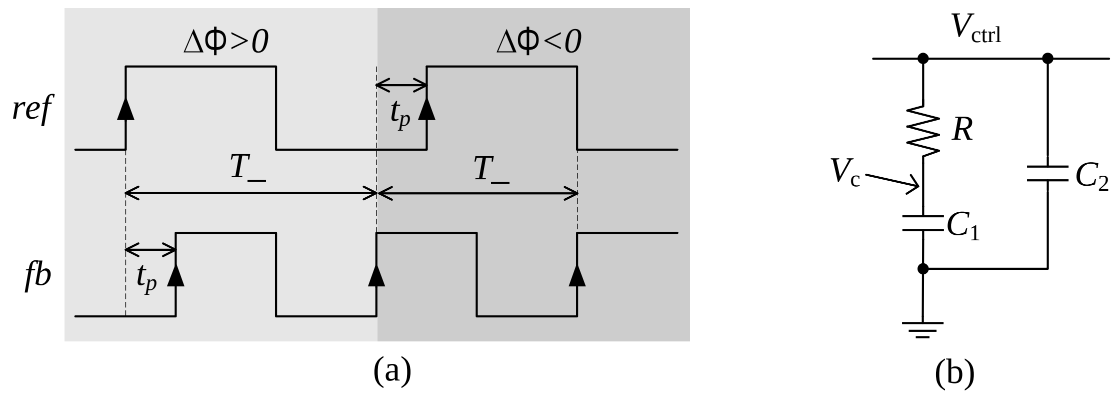


Figure 4.9: (a) Definition of t_p and T_- (b) Two state-space variables in RC network [HBMM04]

in [HBMM04]. Through approximation, linearization and calculation of these above equations, the closed-loop transfer function of the 3rd-order CPPLL in lock-in state can be derived as

$$H(z) = NK_z \frac{z^2[\frac{K_\text{c}}{K_\text{c}+1}(1-\alpha)+\omega_z T] - z[\frac{K_\text{c}}{K_\text{c}+1}(1-\alpha)+\omega_z T\alpha]}{z^3+z^2[k_z(\frac{K_\text{c}(1-\alpha)}{K_\text{c}+1}+\omega_z T)-\alpha-2]+z([2\alpha+1-k_z(\frac{K_\text{c}(1-\alpha)}{K_\text{c}+1}+\omega_z T)]-\alpha}. \quad (4.17)$$

It is worth noting that the closed-loop transfer function $H(z)$ in Equation 4.17 calculated by the state space analysis is exactly identical to the closed-loop transfer function $H(z)$ in Equation 4.9 calculated by the impulse invariant transformation for the 3rd-order CPPLL.

4.3 Performances of PLLs

The design of PLLs has to be tailor-made to apply for various kinds of scenarios. For example, PLLs in clock generation should be insensitive to noise sources and provide a clean clock signal, otherwise a clock signal with large jitter may result in bit errors. PLLs in frequency synthesizer should lock the desired frequency quickly, otherwise a slow locking process might make PLL fail to lock the frequency. In this thesis, two typical important performances of PLLs, i.e. locking time and phase noise (jitter), are the main objects to be optimized while considering the stabilities of PLLs in lock-in state and in nonlinear acquisition process as well.

4.3.1 Locking Time

When a PLL begins to oscillate at an unlocked frequency which is very different of the desired frequency, and then always approaches till a predefined deviation with the desired frequency, e.g. $\delta = 2.5\%$ in this thesis. The corresponding time-cost to lock the frequency is defined as the locking time T_s, as shown in Fig. 4.10.

To save power consumption in a microprocessor, some circuits are often switched to power down mode when they are idle. PLLs are usually in charge of synchronizing the external and

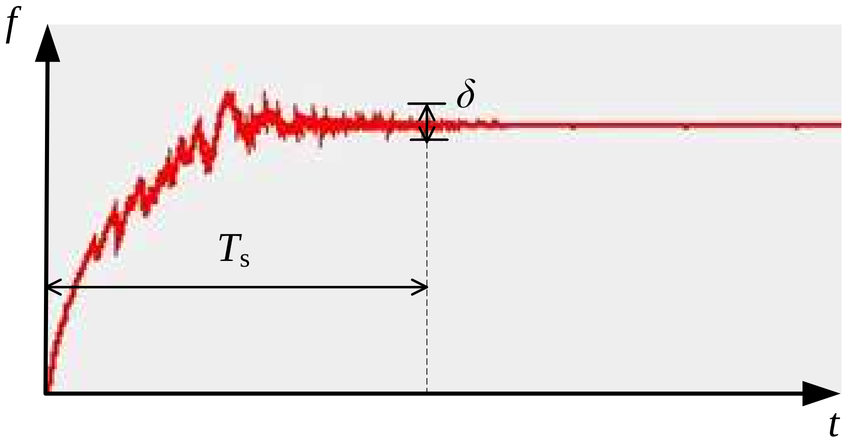


Figure 4.10: Locking time definition

the internal clock signals. The locking time performance of PLLs determines how often the circuit can be shut down, and how long the other part of the circuits in the system have to wait after the PLL is turned on again.

4.3.2 Phase Noise & Jitter

Phase noise and jitter are different ways to characterize a same nature phenomenon. When viewed in frequency domain, an ideal clock signal appears as a peak at one frequency, while a real clock signal with noise appears as “skirt” around the peak shown in Fig. 4.11, denoting the frequency fluctuations. This is referred to as phase noise. When viewed in time domain, phase noise appears as “fuzz” on rising and falling transient edges shown in Fig. 4.13, implying that the signal period is not kept constant. The period variation is referred to as jitter. Here, some types of phase noise and jitter and their correlation are presented in this thesis. More theory analysis and mathematic modeling can be found in [Raz96, DMR00, HL98, HLL99, LH00, Meh02].

4.3.2.1 Phase Noise

An ideal sine-wave signal can be described by

$$x(t) = A\cos(2\pi f_0 t + \phi), \tag{4.18}$$

where A is the nominal amplitude the signal, f_0 is the nominal frequency of the oscillation, and ϕ is an arbitrary fixed phase reference. The spectrum of the ideal sine-wave signal without any fluctuation concentrates at the frequency $f = f_0$, as shown in Fig. 4.11[*]. However such an ideal signal does not exist in the real world. A practical output signal is normally given by

$$x(t) = A(t)\cos(2\pi f_0 t + \phi(t)), \tag{4.19}$$

[*] Only one single side, i.e. $0 \leq f < \infty$, is presented here.

where $A(t)$ and $\phi(t)$ represent the amplitude and the phase fluctuations of the signal respectively. Phase noise describes phase fluctuations of the signal due to *random frequency fluctuations*, its spectrum exhibits noise "skirt" the nominal frequency (also called carrier frequency). Fig. 4.11 shows the spectrum of the ideal signal and the spectrum of the practical signal with phase fluctuations.

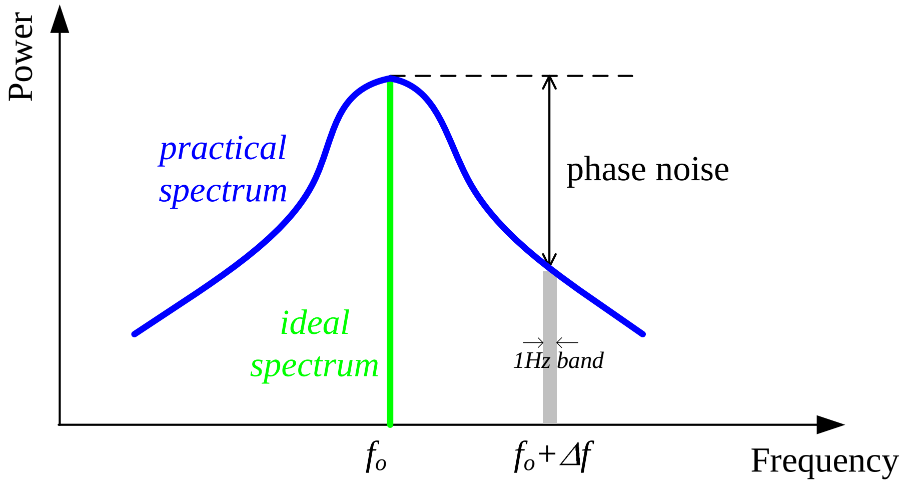


Figure 4.11: Oscillator power spectrum [Rob03]

Phase Noise Characterization Phase noise is used to represent the random phase fluctuations $\phi(t)$, which has power spectral density of the signal phase, i.e. $S_\phi(\Delta f)$ where Δf is the offset from the carrier frequency. In practice, S_ϕ is very difficult to directly measure by using a spectrum analyzer. It is common to measure the power spectral density of the signal itself, i.e. $S_v(\Delta f)$, rather than $S_\phi(\Delta f)$. The phase noise $L(\Delta f)$ at a frequency offset Δf from the carrier with a measure bandwidth of 1Hz can be calculated by

$$L(\Delta f) = 10log_{10}[\frac{P_{\text{sideband}}(f_0+\Delta f, 1\text{Hz})}{P_{\text{carrier}}}], \tag{4.20}$$

where $P_{\text{sideband}}(f_0+\Delta f, 1\text{Hz})$ and P_{carrier} represent the single side-band powers of the offset frequency and of the carrier frequency respectively. In Fig. 4.11, $P_{\text{sideband}}(f_0+\Delta f, 1\text{Hz})$ can be represented by the area of the rectangle with 1Hz bandwidth at offset Δf, and P_{carrier} can be represented by the total area under the power spectrum curve. Phase noise can be approximately calculated by the height difference of the spectrum power at f_0 and at $f_0+\Delta f$.

When power spectrum in Fig. 4.11 is specified in dBc/Hz[†] at a given offset, the phase spectrum can be depicted in Fig. 4.12. A single-sideband phase noise spectrum falls at different rates caused by different dominant noise sources. The spectrum can be divided into three regions: a $1/\Delta f^3$ region, a $1/\Delta f^2$ region and a flat region ($1/\Delta f^0$) [HL98, Abi97]. In the frequency range with very small offset frequency, the flicker noise of devices generally dominates and the spectrum in this region falls at $1/\Delta f^3$. In the "white frequency" variation region, white or uncorrelated fluctuations dominate, e.g. thermal noise of devices, and the spectrum falls at $1/\Delta f^2$. External noise sources dominate in the third region.

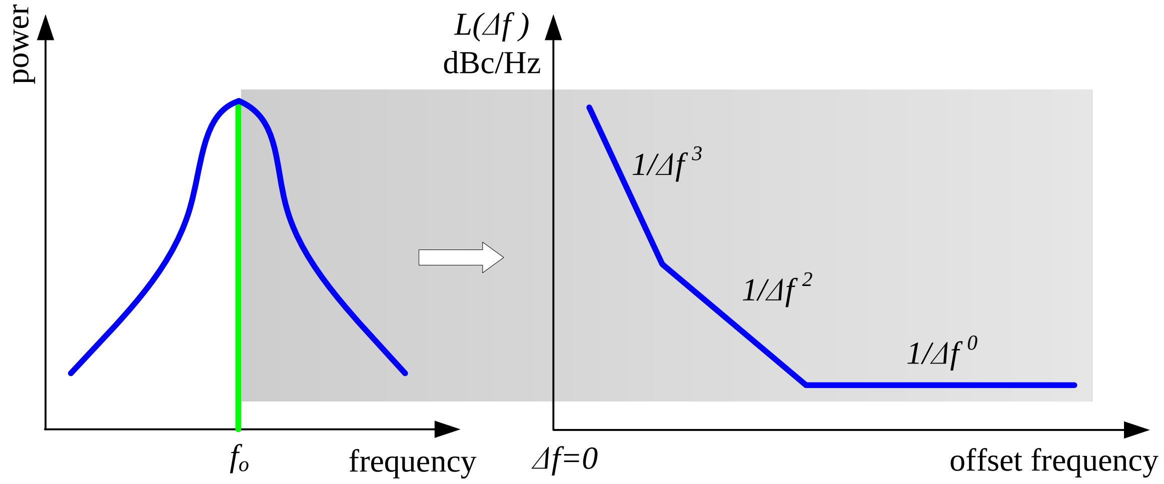


Figure 4.12: A typical spectrum for signal-sideband phase noise [Rob03]

4.3.2.2 Jitter

Designers prefer to characterizing the signal properties in time domain rather than in phase spectrum for applications on clock generation or recovery. As shown in Fig. 4.13(a), suppose $\{t_n\}$ is an ideal square-wave clock with nominal period T. The phase fluctuation results in the unstable zero-crossing or transition time on $\{t_n^*\}$. Jitter is defined as the time deviation between the transition events (rising/falling edges) in $\{t_n^*\}$ and the corresponding transition events in $\{t_n\}$. The discrete jitter values are shown in Fig. 4.13(b).

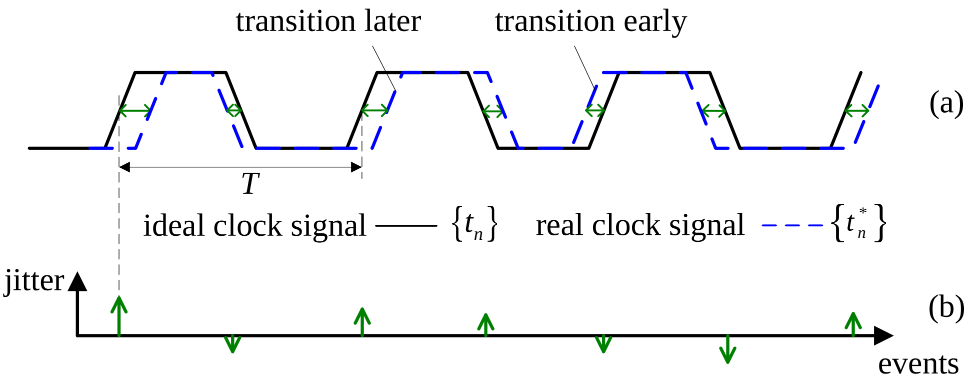


Figure 4.13: Jitter definition

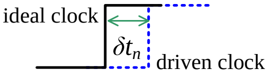
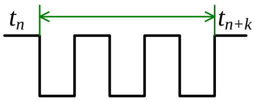
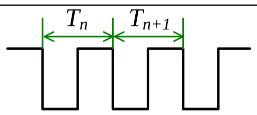
Jitter Types Based on the involved noise sources, jitter can be classified into two forms: *deterministic jitter* and *random jitter*. Deterministic jitter is caused by effects in a predictable manner, such as channel bandwidth limitation, cross talk, duty cycle distortion, supply noise and etc. Random jitter is caused by stochastic processes in an unpredictable manner, such as

[†] dBc is the decibels relative to the carrier.

thermal and flicker noises. Deterministic jitter is usually bounded and has own maximal value, while random jitter is usually assumed to be Gaussian (normal) distribution. The total jitter exhibits usually the convolution of the deterministic jitter and the random jitter together.

Based on the jitter behavior in PLLs, jitter can be classified into another two forms: *synchronous jitter* and *accumulating jitter* [Kun05]. Synchronous jitter is exhibited mainly in driven blocks such as PFD, CP and Divider. In these blocks, the frequency of the output signal is exactly same as the input frequency, and the phase of the output signal fluctuates directly with respect to any fluctuation of the input phase. Accumulating jitter is exhibited in autonomous blocks such as VCO. In these blocks, a fluctuation at the output is not a direct result of one fluctuation event at the input signal, but rather an accumulation result of all previous input fluctuations.

Table 4.1: Jitter metrics based on [Kun05]

Jitter Metrics	Time Diagram	Mathematic Calculation	Remark
edge-to-edge jitter		$J_{ee} = \sqrt{var(\delta t_n)}$	· a scalar jitter metric · no information of correlation
long-term jitter		$J_k = \sqrt{var(t_{n+k} - t_n)}$	· in unit of time · information of correlation on k adjacent/distant transitions
cycle-to-cycle jitter		$J_{cc} = \sqrt{var(T_{n+1} - T_n)}$	· a scalar jitter metric · only information of correlation on the adjacent transitions

Jitter Metrics Jitter can be measured and evaluated in various ways. Here the three metrics defined in [Kun05] are adopted here and illustrated in Tab. 4.1. In PLLs, edge-to-edge jitter is only defined for the driven blocks, i.e. PFD, CP and Divider, while the remaining jitter metrics are suitable for both driven and autonomous blocks.

4.3.2.3 Extracting Jitter from Phase Noise

As circuit speed increases, the requirements on jitter-measuring equipment become more critical. It is easier to characterize signal by measuring its phase noise in frequency domain rather than by measuring its jitter in time domain[‡]. As mentioned before, phase noise and jitter characterize the same phenomenon. Phase noise can be interpreted as a measurement of jitter at a specified offset frequency away from the carrier. In other words, the jitter value can be derived from a phase noise measurement.

[‡] "For example, most jitter measuring oscilloscopes are only capable of measuring jitters down to 1psRMS. Most modern real-time oscilloscopes only have bandwidths up to 7GHz. Phase-noise equipment, on the other hand, can measure noise levels of the best low-noise oscillators available (much less than 1ps in time domain) and offer bandwidths of up to 40GHz". [jit]

The translation between phase noise and timing jitter has been explored in [HLL99, Dra01]. As shown in Appendix C.1, the relationship between the time jitter $\sigma_{\Delta T}$ and noise power spectral density (psd), $S_{\phi}(f)$ is

$$\sigma_{\Delta T}^2 = \frac{8}{w_0^2}\int_0^{\infty} S_{\phi}(f)\sin^2(\pi f \Delta T)df. \tag{4.21}$$

In Appendix C.2, the jitter value of a PFD/CP can be extracted from the phase noise simulation on the PFD/CP blocks:

$$J_{\text{PFD/CP}} = \frac{T}{K_{\text{det}}}\sqrt{\frac{var(\delta n)}{2}}, \tag{4.22}$$

where K_{det} is the gain of the PFD/CP, normally equal to $I_{\text{CP}}/(2\pi)$, and $var(\delta n)$ is equal to

$$var(\delta t_n) = \int_0^{\infty} S_n(f)df, \tag{4.23}$$

where $S_n(f)$ is the power spectral density of the δt_n sequence. The jitter value of a VCO can be extracted from the phase noise simulation on the VCO block:

$$J_{\text{VCO}} = \frac{\Delta f}{f^{1.5}}10^{\frac{L(\Delta f)}{20}}. \tag{4.24}$$

Note the equation is only valid when the phase noise at Δf, which is in the $1/f^2$ region.

4.3.3 Stability of PLLs

In feedback control systems, the stability of system is always the top topic for designers. When PLLs work in lock-in state, they can be assumed as continuous-time system. Based on s-domain analysis, *phase margin* is normally taken as the stability criterion. Phase margin (PM) and unity gain bandwidth (ω_{UGB}) can be calculated according to Equation 4.4. The corresponding bode diagram for the open-loop transfer function is shown in Fig. 4.14. The phase margin of the loop is expressed as

$$\text{PM} = tan^{-1}(\frac{\omega_{\text{UGB}}}{\omega_{\text{z}}}) - tan^{-1}(\frac{\omega_{\text{UGB}}}{\omega_{\text{p3}}}), \tag{4.25}$$

where ω_{UGB} is the open-loop unity-gain bandwidth, the zero frequency is $\omega_{\text{z}} = \frac{1}{RC_1}$ and the third pole frequency $\omega_{\text{p3}} = \frac{1}{R(C_1 \| C_2)}$. The phase margin at ω_{UGB} benefits from the zero ω_z, but is degraded by the pole ω_{p3}. Generally, phase margin should be not less than 45° [GM92] to guarantee a system stability. An proper setup of the capacitor ratio C_1/C_2 can provide a sufficient phase margin. Moreover, the phase margin is immune to the variation of the absolute capacitance values. C_2 is usually chosen to be about $\frac{1}{10}C_1$.

When PLLs work in nonlinear acquisition process, it could happen in the unstable state although its phase margin is 52° ($> 45°$), as shown in Fig. 4.15(a). This instability could be explained by the fact that the phase margin in z-domain analysis is only 40.2°. Based on the s-domain transfer function $H(s)$ in Equation 4.5 and the z-domain transfer function $H(z)$ in Equation 4.9, the corresponding bode diagrams of the closed-loop transfer functions for the 3rd-order CPPLL are depicted respectively in Fig. 4.15(b).

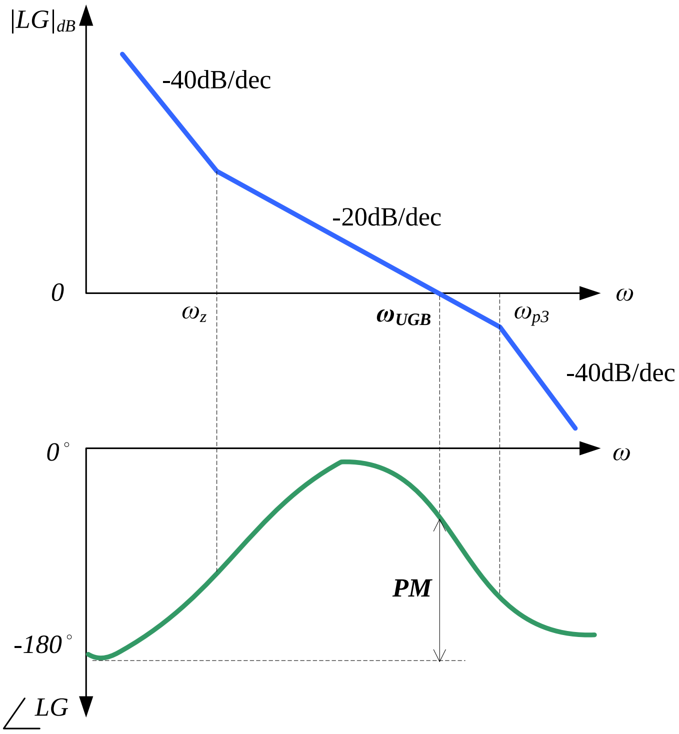


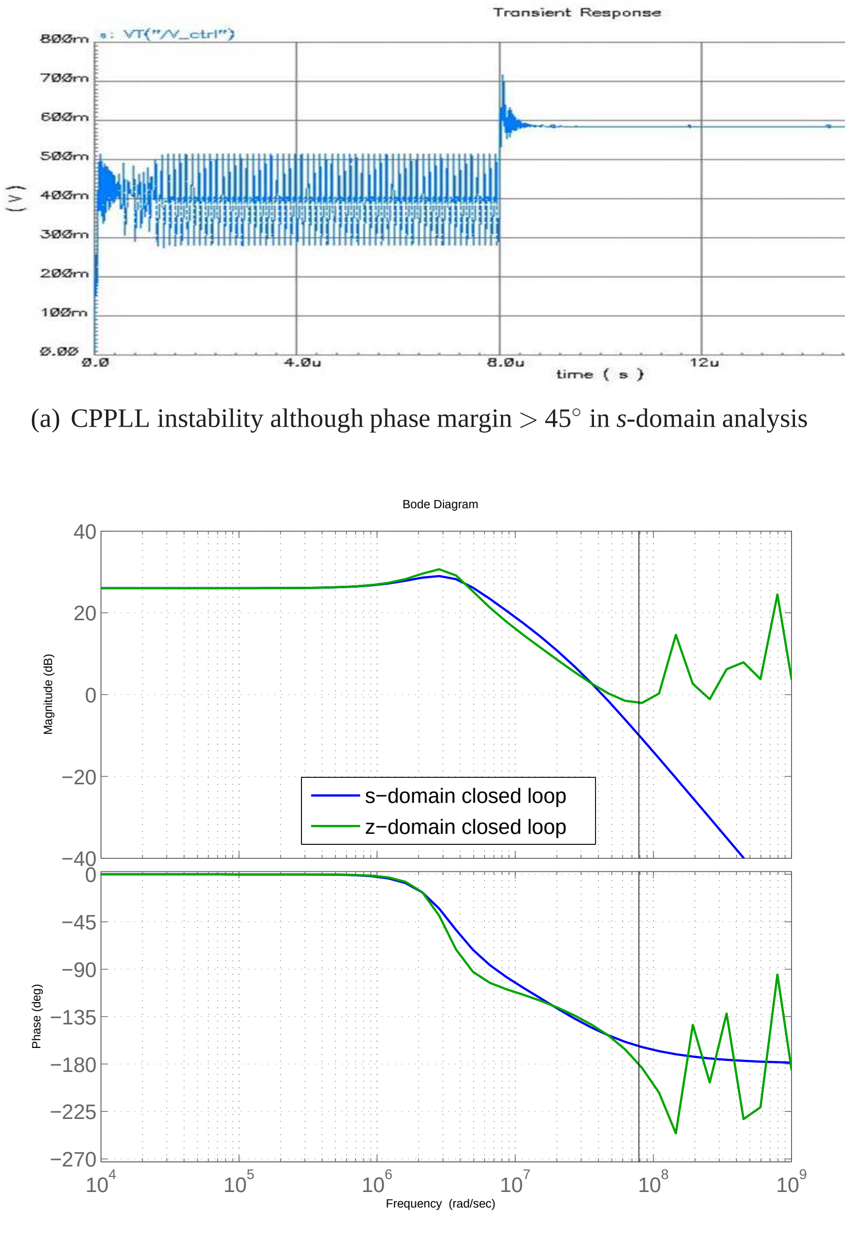
Figure 4.14: Bode diagram for open loop of a third-order CPPLL

When the PFD update frequency (input frequency) is comparable to the unity loop bandwidth ω_{UGB} of the PLL, the excessive phase shift introduced by the feedback delay has to be considered. Besides, the discrete sampling nature of PFD cannot be ignored either. *s*-domain analysis is not capable of accurate predicting the PLL stability any more, *z*-domain analysis and the state-space analysis are introduced to tackle such instability problem. This instability scenario can also be explained by the positions of zeros and poles in root locus plot. Although all zeros and poles of the PLL closed-loop transfer function are in the left part of the *s*-domain root locus, as long as one pole is out side of the unit circle in the *z*-domain root locus, the PLL cannot be stable. More details can be found in [HBMM04, JYM+07]. Therefore, a stability constraint for the nonlinear acquisition process is defined here: the ratio between the reference signal frequency to the unity-gain-bandwidth (RUR) expressed as

$$\mathrm{RUR} = \frac{\omega_{\mathrm{ref}}}{\omega_{\mathrm{UGB}}}, \qquad (4.26)$$

where ω_* is the corresponding angle frequency of f_*, where $\omega_* = 2\pi f_*$. To be simple, both are called frequency throughout this thesis. Based on the results of [JYM+07], the *s*-domain transfer function matches very well with their corresponding *z*-domain transfer function when the RUR is larger than 20.

In summary, the stability of the CPPLL is dependent on the phase margin, the unity-gain-bandwidth and the reference frequency.



(a) CPPLL instability although phase margin $> 45^\circ$ in s-domain analysis

(b) Comparison of s-domain and z-domain analysis in terms of Bode diagrams

Figure 4.15: CPPLL instability observed in z-domain analysis

4.3.4 Design Trade-offs

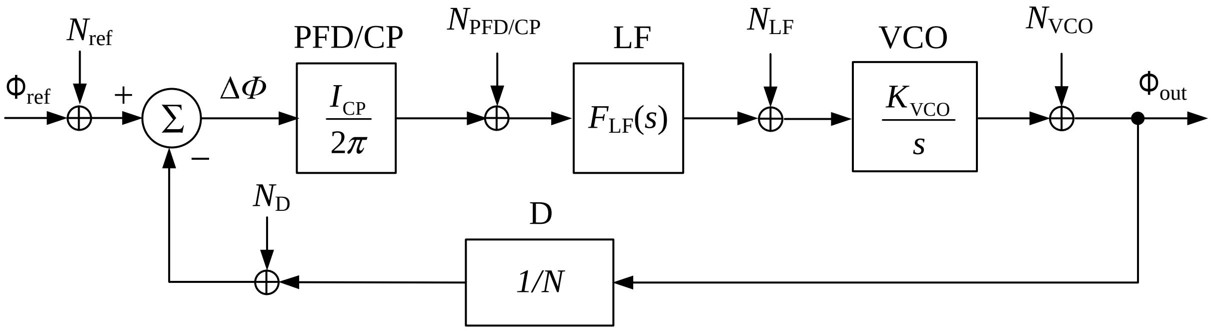
The loop bandwidth is a key design factor for PLL performances, which determines locking time, jitter and stability of PLLs. Essentially, the loop bandwidth ω_{BW}, i.e. the -3dB bandwidth for the PLL closed-loop transfer function is same as the unity-gain-bandwidth (0dB bandwidth) for the PLL open-loop transfer function, ω_{UGB}. Therefore, the loop bandwidth ω_{BW} can be computed by Equation 4.4.

Although the nonlinear acquisition process cannot be accurately predicted through *s*-domain analysis, the locking time is inversely proportional to the loop bandwidth of the PLL according to [Bes03, HBMM04]. With a larger loop bandwidth, the PLL can lock more quickly the expected output frequency and vice versa.

Noise in each building block of a PLL contributes to the total output noise, which is characterized in terms of phase noise in phase domain or jitter in time domain. Assuming that all the noise sources are not correlated and they are placed at the corresponding input node in Fig. 4.16 based on *s*-domain analysis, the overall noise power of the PLL can be computed by

$$N^2_{\text{total}} = N^2_{\text{ref@out}} + N^2_{\text{PFD/CP@out}} + N^2_{\text{LF@out}} + N^2_{\text{VCO@out}} + N^2_{\text{D@out}}, \qquad (4.27)$$

where N^2_{total} is the total output noise power and the $N^2_{***\text{@out}}$ presents the noise power due to noise source N_{***}. It is easy to know that the noise transfer function (NTF) of N_{ref} and N_{D} is same as the closed transfer function of the PLL in Equation 4.5. Hence both NTFs have also a low-pass filter characteristic. The transfer function is



N_{ref} : Noise from the reference signal;
$N_{\text{PFD/CP}}$: Noise from the PFD/CP;
N_{LF} : Noise from the LF;
N_{VCO} : Noise from the VCO;
N_{D} : Noise from the D;

Figure 4.16: Linear model of a CPPLL with noise sources

$$\frac{N_{\text{out}}}{N_{\text{ref\&D}}} = \frac{K_{\text{VCO}} I_{\text{CP}}}{2\pi C_2} \frac{s + \frac{1}{RC_1}}{s^3 + (\frac{C_1+C_2}{RC_1C_2})s^2 + \frac{K_{\text{VCO}} I_{\text{CP}}}{2\pi N C_2} s + \frac{K_{\text{VCO}} I_{\text{CP}}}{2\pi N R C_1 C_2}}. \qquad (4.28)$$

If we move the noise from PFD/CP ($N_{\text{PFD/CP}}$) to the input of the PFD, the transfer function of $N_{\text{PFD/CP}}$ to the output noise should be same as the transfer function of N_{ref} and N_{D}. So we can simply obtain the actual NTF for PFD/CP noise through Equation 4.28 divided by the gain of PFD/CP, i.e.

$$\frac{N_{\text{out}}}{N_{\text{PFD/CP}}} = \frac{K_{\text{VCO}}}{C_2} \frac{s + \frac{1}{RC_1}}{s^3 + (\frac{C_1+C_2}{RC_1C_2})s^2 + \frac{K_{\text{VCO}}I_{\text{CP}}}{2\pi NC_2}s + \frac{K_{\text{VCO}}I_{\text{CP}}}{2\pi NRC_1C_2}}. \tag{4.29}$$

Therefore, Equation 4.29 shows a low-pass characteristic as well. Similarly, the NTFs for LF and VCO are calculated respectively and expressed as

$$\frac{N_{\text{out}}}{N_{\text{LF}}} = K_{\text{VCO}} \frac{s^2 + \frac{c_1+c_2}{RC_1C_2}s}{s^3 + (\frac{C_1+C_2}{RC_1C_2})s^2 + \frac{K_{\text{VCO}}I_{\text{CP}}}{2\pi NC_2}s + \frac{K_{\text{VCO}}I_{\text{CP}}}{2\pi NRC_1C_2}} \tag{4.30}$$

$$\frac{N_{\text{out}}}{N_{\text{VCO}}} = \frac{s^3 + \frac{c_1+c_2}{RC_1C_2}s^2}{s^3 + (\frac{C_1+C_2}{RC_1C_2})s^2 + \frac{K_{\text{VCO}}I_{\text{CP}}}{2\pi NC_2}s + \frac{K_{\text{VCO}}I_{\text{CP}}}{2\pi NRC_1C_2}}. \tag{4.31}$$

The NTF for LF in Equation 4.30 shows a bandpass filter characteristic, while the NTF for VCO in Equation 4.31 has a high-pass characteristic. As seen from the above NTFs, the overall loop transfer function are different filter types for each noise source in PLL, as shown in Fig. 4.17. In order to reduce the output noise due to the input noise from reference signal, divider and PFD/CP, the loop bandwidth of the PLL should be as small as possible. When the noise from the VCO dominates, the loop bandwidth should be larger. Apparently, a compromising selection of the loop bandwidth has to be made to optimize the overall PLL noise performance.

The respective NTF in z-domain for each noise source can be calculated based on the $H(z)$ in Equation 4.9. Although z-domain analysis can provide more accurate description on PLL properties, NTFs in s-domain are usually used to predict the noise transfer behavior of PLLs, since noise performance in PLL's lock-in state is mostly considered.

Moreover, the loop bandwidth is also involved in the stability of PLLs, as discussed in Sec. 4.3.3. Since a PLL is actually a discrete-time circuit due to the sample nature of the PFD, a too large loop bandwidth may make the PLL unstable during the nonlinear acquisition process. In summary, the impacts of the loop bandwidth on locking time, noise performance and system stability of PLLs are listed in Tab. 4.2.

Table 4.2: Trade-offs in CPPLL

loop bandwidth	↗
locking time	+
suppression noise from input, divider & PFD/CP	-
suppression noise from VCO	+
loop stability	-

note: + improving, - deteriorating

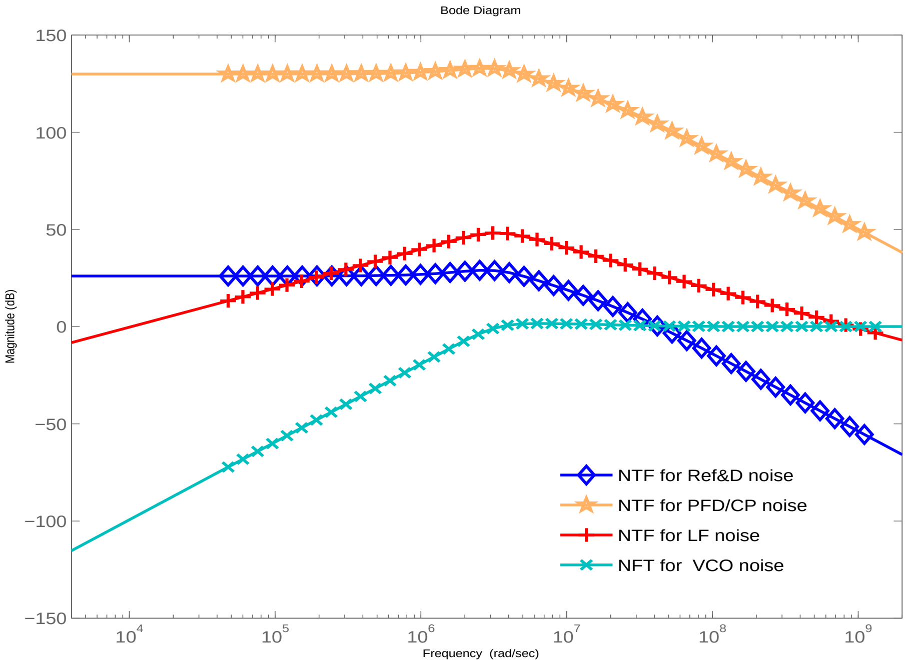


Figure 4.17: Loop transfer function from each noise source to PLL's output

4.4 Example: Hierarchical Optimization of a CPPLL

The trade-offs of PLLs design listed in Tab. 4.2 show us how difficult it is to optimize PLLs by hand-design. Moreover, the PLL performance evaluation by circuit-level simulation is very time consuming. For example, a transient simulation for locking time performance requires about 1.5 hours on this CPPLL experimental circuit. Suppose an optimization process needs 100 simulations, it would last 6.25 days only for optimization on the locking time. To optimize jitter performance, another long transient simulation needs to conduct in order to get sufficient samples of the output periods. Then jitter value can be estimated based on the standard deviation of these period-samples. Hence, the simulation for jitter lasts even longer. In consequence, the "simulation-in-a-loop" based optimization method cannot directly apply to the CPPLL. Here, the hierarchical optimization methodology proposed in Chapter 3 will be applied to the CPPLL, and the whole design process can be accomplished in a reasonable time cost.

The 3$^{\text{rd}}$-order CPPLL architecture is already shown in Fig. 4.2. Since the PFD and the D building blocks are pure digital circuits, they will not be optimized in this thesis. The values of the resistance and the capacitances in the LF will be calculated, while their circuit-level realization will not be discussed here. The focus is on two important analog building blocks: i.e. the CP and the VCO.

The topology of the CP is an outside-biased current-steering configuration, as shown in Fig. 4.18. The reference current I_{bias} is mirrored by the current mirror (P1&P2) into the charging path (P2&P4) or by the current mirror (N0&N2) into the discharging path (N2&N4). When signal up (dn) is active, the source current flows into (out of) the loop filter, so that the output voltage of the loop filter rises (drops), which forces a higher (lower) oscillation frequency consequently. Note that signals *up* and *dn* cannot be active at the same time, which means the LF cannot be charged and discharged at the same time. Signals $\overline{up}$ and $\overline{dn}$ are the complement of the signals *up* and *dn* respectively. The dummy switches (N5&P6) are intended to avoid any voltage peak at the output node.

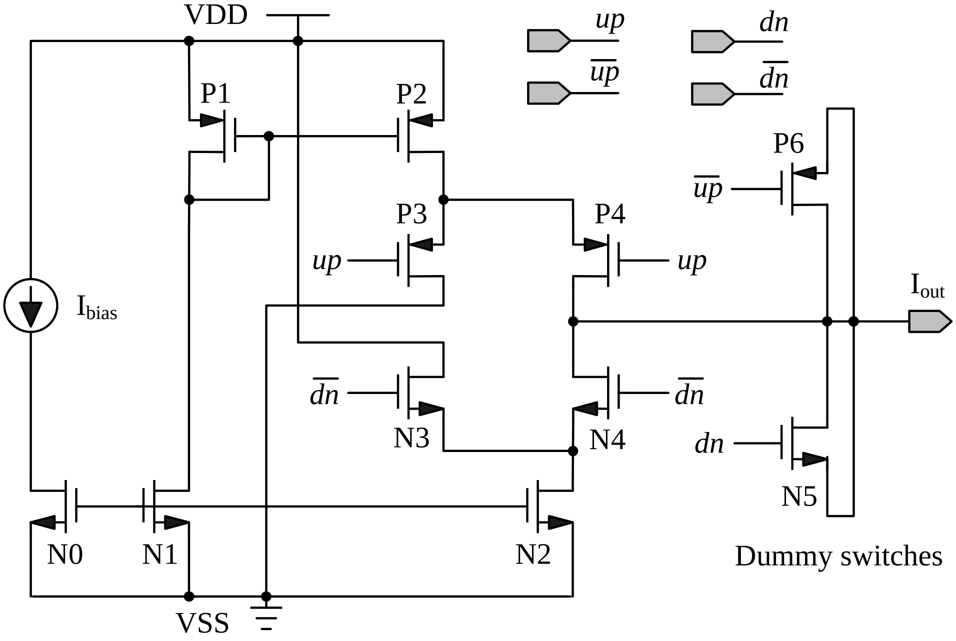


Figure 4.18: Schematic of CP Block

The VCO is a five-stage single-end ring oscillator, as shown in Fig. 4.19. The input voltage V_{ctrl} controls the current through the delay elements, thus determines the delay time of each stage and in turn determines the output oscillation frequency. The upper and lower transistors, e.g. P1&N1, act as the voltage control current sources (VCCS), and the middle transistors, e.g. P6&N6, act as the delay elements. The current flowing through these delay elements is generated by these current mirror structures, e.g. (N0&N1 and P0&P2).

4.4.1 CPPLL Hierarchical Modeling

In this experiment, the CPPLL is realized in a 180nm technology with a supply voltage VDD of 3V, using a reference signal f_{in} of 25MHz to generate the output frequency ranging from f_{min} (=150MHz) to f_{max} (=500 MHz). In the hierarchical modeling of the CPPLL, system performances are modeled in dependence of system-level parameters from each building block, while these system-level parameters can be determined by simulation of the blocks on circuit level. Fig. 4.20 gives an overview of the hierarchical performance modeling of the CPPLL.

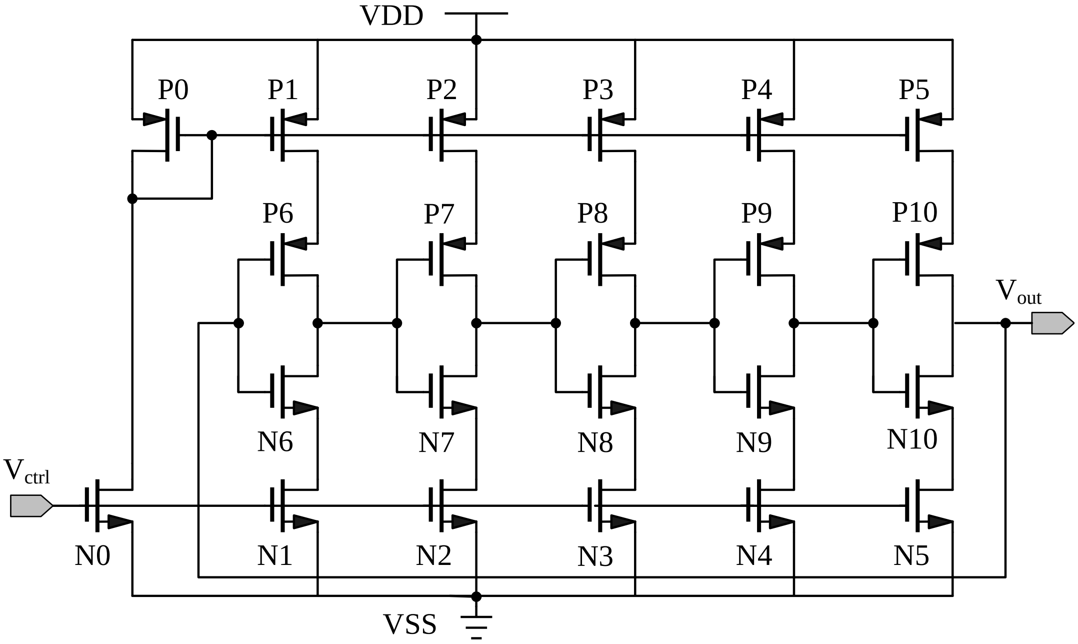


Figure 4.19: Schematic of VCO Block

4.4.1.1 CPPLL on System Level

System-level performances Six important performances of the CPPLL system are considered: *locking time, jitter, power consumption, unity-gain bandwidth, phase margin and output frequency*.

The locking time is defined as the time taken by the CPPLL to synchronize with or to lock onto a new frequency. The performance optimization shall be considered at the worst case. Therefore the locking time is defined as the time for the output frequency directly jumping from f_{min} (150MHz) to f_{max} (500MHz).

The total output noise of the CPPLL comes from the CP and the VCO in this experiment, while the input signal and other building blocks are considered as noise-free. The CPPLL exhibits various noise transfer characteristics with respect to the different noise sources, as discussed in Sec. 4.3.4. Hence, the design of the loop bandwidth has to be a trade-off for jitter of the VCO and jitter of the CP when minimizing the total output jitter. The jitter performance shall also be optimized at its worst case, i.e. when the CPPLL works at its minimal output frequency f_{min} [§].

With the predetermined digital building blocks PFD and divider, the power consumption is defined as the sum of the power consumption in the CP and VCO (we consider only analog blocks[¶]):

$$Power = VDD \cdot 4 \cdot I_{\text{CP}} + VDD \cdot I_{\text{VCO}}. \tag{4.32}$$

[§] At the minimal output frequency, the current consumption is minimal (VCO current consumption is minimal and other blocks keep constant) so that the jitter performance is worst, which can also be seen from the Pareto-optimal front of the VCO in Sec. 4.4.3

[¶] The current consumption of LF is not needed to calculate, since the charging or discharging currents in LF are from or to CP.

We consider the worst-case power consumption that occurs when the CPPLL works at its maximal output frequency $f_{\max}$.

For the optimization process, locking time, jitter and power consumption are taken as the optimization targets at the system level:

$$\mathbf{f} = [T_s, J_{\text{sum}}, Power]^T. \tag{4.33}$$

The output frequency range of the CPPLL is determined by the VCO, therefore this performance is directly forwarded as a circuit-level performance of the VCO block. Additionally, two constraints concerning stability are considered for system-level optimization:

- The stability criterion RUR for nonlinear acquisition process. We specify RUR ≥ 20.
- The stability criterion PM for linear lock-in state. We specify PM $\geq 45^\circ$

system-level parameters For each building block of the CPPLL, a set of system parameters $\mathbf{p}$ that are capable of capturing the influence of the blocks on the system performances & tradeoffs, are defined:

- CP: outside-biased reference current I_{CP}, jitter J_{CP}
- VCO: gain K_{VCO}, current consumption I_{VCO}, jitter J_{VCO}
- LF: loop filter elements R, C_1, C_2
- D: divider value N [‖]

4.4.1.2 CPPLL on Circuit Level

Circuit-level performances On circuit level, the system-level parameters become performances of the building blocks that can be simulated in dependence of circuit-level design parameters. For the CP for instance, the absolute values of the simulated charging and discharging currents I_{up} and I_{down} are equal to the value of the outside-biased current I_{CP} on behavioral level, and the system-level jitter J_{CP} becomes a circuit-level performance of the CP. For the VCO, gain, current consumption, jitter and output frequency range are the circuit-level performances corresponding to the system-level parameters.

Circuit-level parameters The transistor dimensions, i.e. widths/lengths, are the circuit-level parameters. In the CP in Fig. 4.18, there are two sets of current mirror structures, i.e. P1&P2 and N0&N1&N2. The transistor dimensions in a current mirror structure should be identical. In the VCO in Fig. 4.19, there are also two current mirror structures, i.e. P0&P1&P2&P3&P4&P5 and N0&N1&N2&N3&N4&N5. The dimensions of other transistors in both circuits are set as same as that of the digital circuit. Since the transistors in VCO block work in large-signal function, the sizing rules in Fig. 4.20 are only related to the geometric rules in Tab. A.2. For the current mirror transistors in CP block, the sizing rules includes not only geometric rules but electrical sizing rules as well.

[‖] The divider value is determined by the input and the expected output frequency.

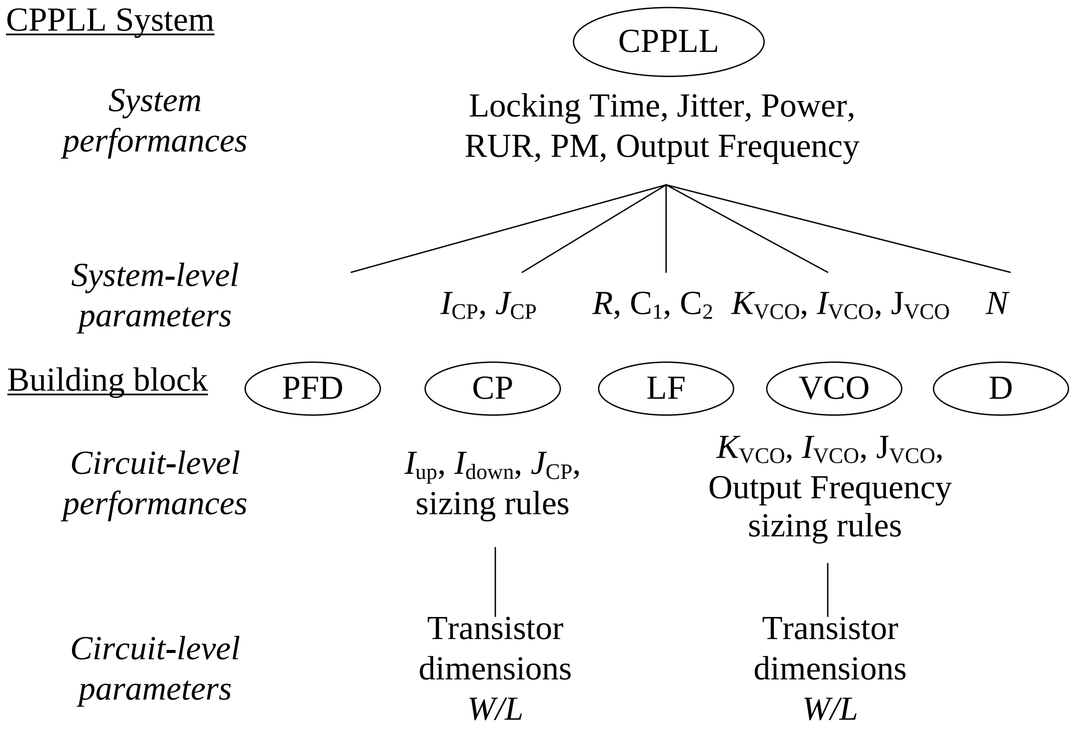

CPPLL System
CPPLL
System performances
Locking Time, Jitter, Power, RUR, PM, Output Frequency
System-level parameters
I_{CP}, J_{CP}
R, C_1, C_2
K_{VCO}, I_{VCO}, J_{VCO}
N
Building block
PFD
CP
LF
VCO
D
Circuit-level performances
I_{up}, I_{down}, J_{CP}, sizing rules
K_{VCO}, I_{VCO}, J_{VCO}, Output Frequency sizing rules
Circuit-level parameters
Transistor dimensions W/L
Transistor dimensions W/L

Figure 4.20: Hierarchical performance modeling of the CPPLL

4.4.2 Modeling CPPLL in Verilog-A

Recently, various behavioral models [AETL96, SG99, HKH00, OHSA04, Kun05] have been developed to accelerate the system-level simulation of PLLs. In this thesis, the system-level models are based on [Kun05], which are realized in Verilog-A [Verb]. To achieve efficient modeling and fast extraction, two sets of models are used for locking time and jitter respectively. The first set of models, list D.1-list D.4 in Appendix D, focuses on more accurately representing the nonlinear acquisition locking process, while the second set, list D.5-list D.8, focuses on more efficient modeling of the jitter in order to accelerate jitter simulation.

Models for locking time extraction In order to more accurately model the nonlinear acquisition process, some properties of building blocks are included and presented as follows. For example, the rising/falling time, delay time of the PFD block are firstly extracted from circuit level simulation and then are parameterized as *td_ul* and *td_dl* in the behavioral model.

```
module PFD (ref, feedback, u, ub ,db ,d);
  input ref, feedback;
  output u, ub, db,d;
  electrical ref, feedback, u, ub, db, d;
  ......
  parameter real tt=120p from (0:1000000);  //rise time and fall time
  // the following delay times are extracted from circuit level simulation
  // for "u" and "ub" signal:
  // ----------  rise delay  -------------  fall delay
  parameter real td_u1=37p;  parameter real td_u2=190p;
  // for "d" and "db" signal:
  parameter real td_d1=45p;  parameter real td_d2=136p;
  ......

end
endmodule
```

Since the charging/discharging current sources in Fig.4.18 are realized by N/PMOS transistors, theses N/PMOS transistors will work out of saturation ranges if the output voltage goes too high or too low. Therefore the generated output current cannot be the same value as the out-sided biased reference current. The two parameters V_max and V_min in the CP model are used to represent the limit of the output voltage. The parameter Mis is used to represent the mismatch current between the charging and the discharging currents.

```
module CP(Iout, Down, N_Down, N_Up, Up, Ibias);
    input Up, N_Up, Down, N_Down;
    output Iout;
    electrical Up, N_Up, Down, N_Down, Iout;
    ......
    parameter real Ip=25.0e-6;   //charge pump’s output current
    parameter real v_max=2.85;   //maximum voltage at output node
                                 //so that PMOS current source in saturation
    parameter real v_min=0.35;   //minimum voltage at output node
                                 //so that NMOS current source in saturation
    parameter real Mis=0.00;     //Mismatch of up and down current
    .....
analog begin
    .....
    // restrict the output voltage range from v_min to v_max
    @(cross(V(Iout)-v_max, 1)) begin
        state =0
    end
    @(cross(V(Iout)-v_min, -1))begin
        state =0;
    end
    I(Iout)<+transition(Ip*state*(1+state*Mis), Delay, TransTime );
end
endmodule
```

In the VCO block, according to [LMC01], the linear VCO model predicts more than 90% of the real VCO characteristics, especially in the operating range. In this experimental PLL’s VCO, when the input voltage is in the input range, the VCO output frequency increases with the linear gain K_{VCO}. When the input voltage exceeds the maximum input level, the VCO output frequency can still increase, but the gain is set to be $0.25K_{\mathrm{VCO}}$ instead of K_{VCO}.

```
module vco(V_tune,VCO_out);
    input V_tune;
    output VCO_out;
    electrical V_tune, VCO_out;
    ......
    parameter real Vmin=0.6;
    parameter real Vmax=2.6 from(Vmin:10e5);
    parameter real Kvco=600e6;
    ......
analog begin
    ......
    // compute the freq from the input voltage
    if  (V(V_tune)<=Vmin)
         freq = Fmin;
    else if (V(V_tune)<=Vmax)
         freq = (V(V_tune)-Vmin)*Kvco+Fmin;
    else
         freq = (V(V_tune)-Vmax)*(0.25*Kvco)+(Vmax-Vmin)*Kvco+Fmin;
    ......
end
endmodule
```

Models for jitter extraction In a transient simulation, step size in simulation setup is a crucial factor for time cost and simulation accuracy. If jitters of CP and of VCO cause two individual events and the two events are displaced at almost same time points but not exactly coincident, then circuit simulators have to spend many more time points (i.e. smaller step size) to resolve the two distinct events, hence the simulation will run much slower. For this reason, it is desirable to reduce the number of jitter events by combining jitter sources compactly. The jitter of CP block, i.e. edge-to-edge J_{CP}, can be embedded into the model of the reference input oscillator (OSC) block. The following parameter *syncJitter* represents the jitter of CP.

```
module OSC(out);
    output out; electrical out;
    parameter real freq=25e6 from (0:10e9);
    parameter real accJitter=0 from [0:0.1/freq);
                     //period jitter from reference OSC
    parameter real syncJitter=0 from [0:0.1*ratio/freq);
                     //edge-to-edge jitter from PFD/CP
    ......
end
endmodule
```

As well known, the higher frequency the circuit runs at, the step size in simulation setup has to be smaller. The output frequency of VCO is N times (divider value) of the internal signal frequency of CPPLL. When divider and VCO are merged together, the simulation can run much faster because the high VCO output frequency is never generated actually. The acceleration of simulation is obvious. If divider is merged into the module of VCO, the period jitter at the VCO output should be $\sqrt{N}$ time larger than the period jitter J_{VCO} [Kun05]. Since there are two transient processes in one period: i.e. one rising edge and one falling edge, the output jitter of the VCO_Div block is therefore equal to

$$J_{\text{VCO\_Div}} = \sqrt{2NJ_{\text{VCO}}}. \tag{4.34}$$

The merged behavioral model for VCO with divider is named as VCO_Div. The Verilog-A model is given in the following.

```
module VCO_Div(V_tune,VCO_out);
    input V_tune; output VCO_out;
     electrical V_tune, VCO_out;
    ......
    parameter real N=20; //if N!=1, divider moved into the VCO block
    parameter real Jvco=10e-12;  //Jitter of VCO
    ......
analog begin
    @(initial_step)begin
        seed=160;
        Vout=VSS;
        delta=Jvco*sqrt(2*N);
        // calculating the corresponding jitter on the divider output
    end
    ......
endmodule
```

Based on the above efficient setups in the behavioral models, the system-level simulations for the locking time and the jitter can be finished in some minutes. The reasonable simulation time makes the “simulation-in-a-loop”-based optimization methods feasible for PLL circuits.

4.4.3 Pareto-Optimal Fronts of Building Blocks

The deterministic PSE method mentioned in Chapter 3 is applied to the circuit level of the CP and the VCO respectively. The extracted Pareto fronts presented here are computed through real circuit-level simulation, which is a non-trivial task.

PSE on CP For the CP, small jitter and low power consumption are desired. Fig. 4.21 shows the Pareto-optimal front for the outside-biased current and the output jitter of CP, which represents the lower bound on the combination of both performance values. Right and above the front is the feasible region. The trade-off front indicates that a larger current will result in a smaller jitter.

PSE on VCO Fig. 4.22 shows the feasible ranges bounded by the Pareto fronts for the competing objectives jitter, gain and current of the VCO in 3-D and in the different 2-D projections. We can evaluate e.g. the increasing trend of J_{VCO} with increasing K_{VCO} for constant I_{VCO} from Fig. 4.22(b), or how K_{VCO} increases at the cost of a larger J_{VCO} for constant I_{VCO} from Fig. 4.22(c), or the increasing trend of I_{VCO} with the decreasing of J_{VCO} for constant K_{VCO} from Fig. 4.22(d).

The feasible regions for system-level parameters, i.e. $I_{\mathrm{CP}}, J_{\mathrm{CP}}, I_{\mathrm{VCO}}, K_{\mathrm{VCO}}$ and J_{VCO}, can be extracted from the Pareto-optimal fronts in Fig. 4.21&4.22.

4.4.4 Hierarchical Optimization

Through restricting the system-level parameters on their own Pareto-optimal fronts, an optimal performance of the whole circuit can be obtained [EMG05]. In this experiment, the independent

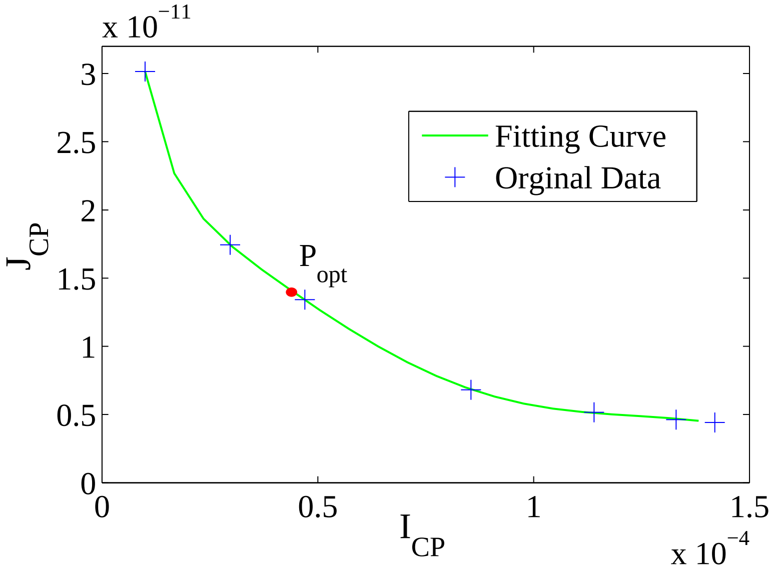


Figure 4.21: Pareto-optimal front of the CP

system-level parameter set is selected as $\mathbf{p} = [I_{\mathrm{CP}}, I_{\mathrm{VCO}}, K_{\mathrm{VCO}}, R, C_1, C_2]$. The parameter J_{CP} is dependent on the parameter I_{CP} and J_{VCO} is dependence on the parameters I_{VCO} and K_{VCO}, as expressed in Equation 4.35. There are two ways to transfer the jitter values into the behavioral models. One way is to directly embed the fitting function into the models given in List D.5 and List D.7. The other way is to calculate the jitter values through a special script file firstly; and then to transfer these values to the jitter parameters of the models by means of circuit netlist mapping:

$$\begin{aligned} J_{\mathrm{CP}} &= \mathrm{fitting\_function\_1}(I_{\mathrm{CP}}) \\ J_{\mathrm{VCO}} &= \mathrm{fitting\_function\_2}(I_{\mathrm{VCO}}, K_{\mathrm{VCO}}). \end{aligned} \tag{4.35}$$

4.4.4.1 File system in WiCkeD

In the thesis, the optimization process is realized in WiCkeD [Wic]. The file system for the CPPLL optimization is briefly listed in Fig. 4.23, and the setup for other optimization tasks in other parts of the thesis is similar to this. The file system can be classified into four levels. At the first level, “CPPLL.def” is the main file for optimization setup, including the following functions:

- The parameters are defined, i.e. design parameters, statistical parameters and operation parameters. And their corresponding upper and lower boundary are also given.
- The circuit performances and the necessary optimization constraints are defined. And their corresponding simulations are set, e.g. "LockingTime.rs" for locking time performance, "Jitter.rs" for jitter performance.

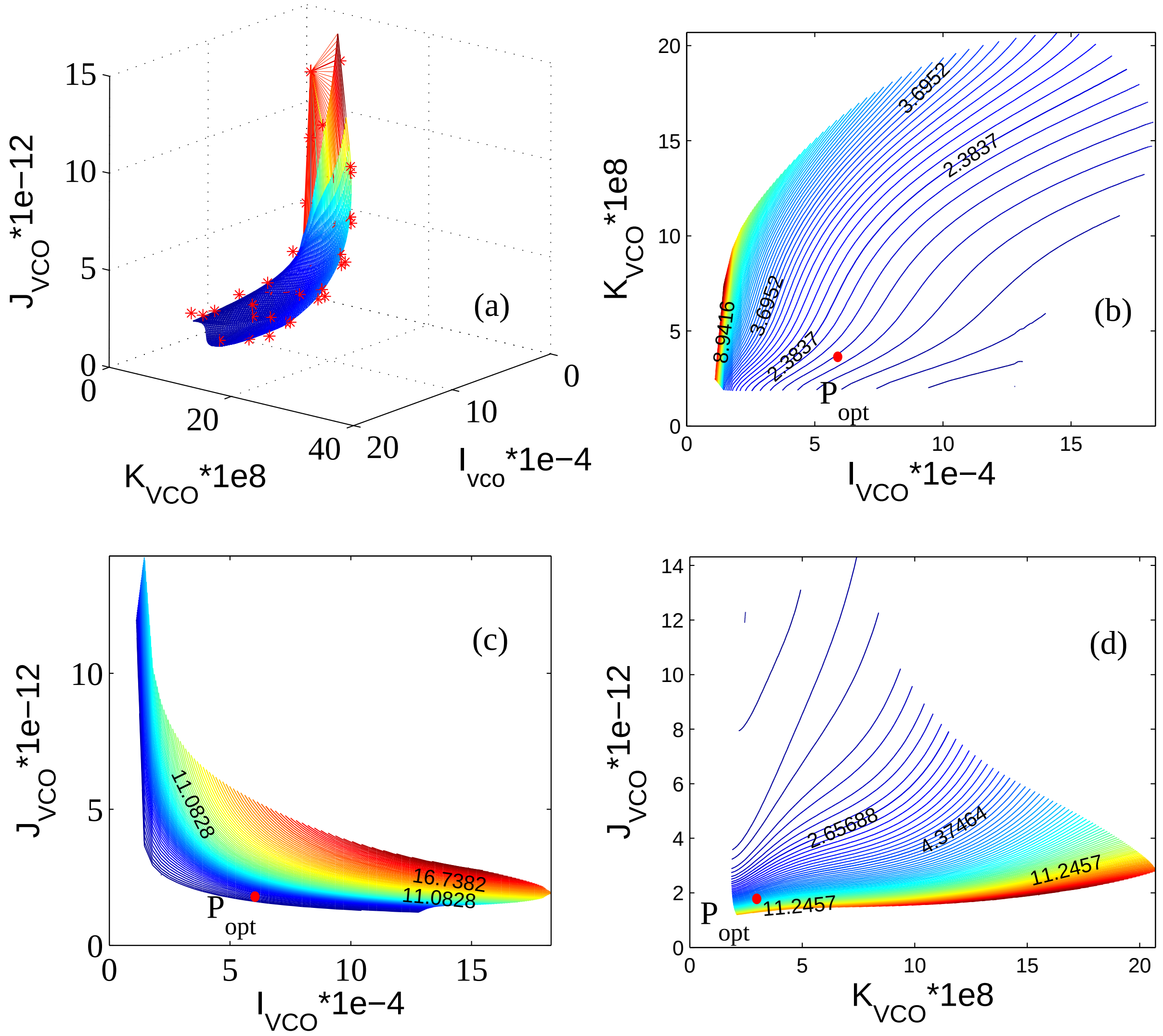


Figure 4.22: Pareto-optimal front of the VCO

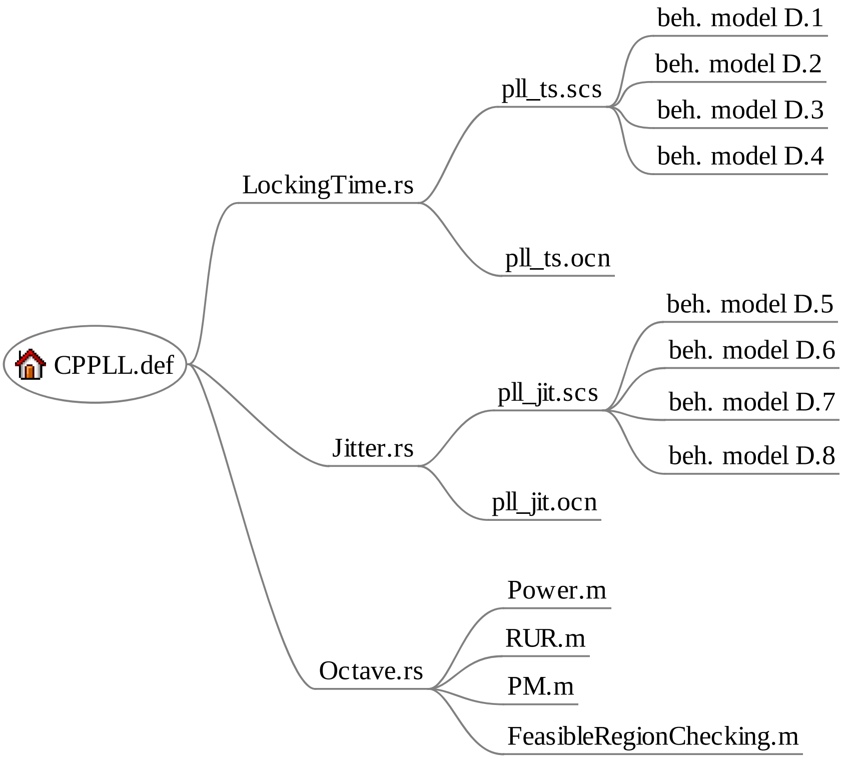


Figure 4.23: File system in WiCkeD

- The simulation tasks are distributed to many machines simultaneously. The performances and constraints can be extracted in parallel.

At the second level, “***.rs” is the setup file for the performances and constraints, which controls the simulation and extraction files at the third level. For example, “Jitter.rs” controls two files: “pll_jit.scs” is the *Spectre*-format netlist of the transient simulation for jitter performance, and “pll_jit.ocn” is in charge of the jitter extraction from primal simulation data bank. The behavior-level modules list D.5-list D.8 are included at the fourth level.

In the optimization of the CPPLL, one transient simulation with *Spectre* is used to obtain the locking time T_s, while another transient simulation is for the jitter J_{SUM}. Based on *s*-domain analysis of the PLL, the linear lock-in state properties, e.g. unity-gain-bandwidth and phase margin, are calculated. A third simulation with octave [Oct] is used for the system-level constraints: the ratio RUR, the PM and the feasible ranges of the system-level parameters.

4.4.4.2 Hierarchical Optimization Results

Tab. 4.3 shows the results of the two optimization steps (top-down automatic sizing and bottom-up verification) in hierarchy. Starting from the system specifications in column 3 and the initial design in column 4, system-level optimization leads to the optimized system performances in the upper half of column 5 and the system-level parameters in the lower half of column 5. The latter, propagated as specifications to the circuit-level optimization, results in the final $\mathbf{P}_{\mathrm{opt}}$, which are listed in the lower half of the last column and depicted in dots in Fig. 4.21&4.22, respectively. A system simulation with the behavioral models calibrated according to the circuit-level design parameters yields the system performances in the upper half of column 6. We can see that by considering the technological feasibility of building blocks a first-time-successful hierarchical optimization has been achieved for a comprehensive set of design aspects. The optimization on system level and circuit level requires about 1.5 hours respectively, which results in a total of 3 hours.

4.5 Pareto-Optimal Front Computation (POFC) of a whole CPPLL

Compared to only one optimization design point in the last experiment example, a comprehensive optimization result, i.e. Pareto-optimal front of the system performance, will be computed in this experiment example. PSE method is applied not only to the building blocks but to the whole CPPLL system as well. The trade-offs in the performance of building blocks, e.g. gain, jitter and power in VCO, and the performance at system level, e.g. bandwidth, locking time and jitter, will be represented as Pareto-optimal fronts. These fronts offer designers a fast way to get insight into the capability of the whole system for a given technology.

The topologies of the building blocks in this experiment are exactly same as those of the last experiment. The technology is 130nm and the normal supply voltage VDD is 1.8 volt. The reference input signal is f_{in}=25MHz. The CPPLL is supposed to generate the output frequency f_{out}=500MHz.

4.5.1 POFC of the CP Block

For the CP block, the feasible range of the out-biased current is set from $5\mu A$ to $65\mu A$. The Pareto-optimal front in Fig. 4.24 represents the trade-off relationship between the two performances, that the output jitter of CP decreases with the increasing current. The front shows the lower bound for both performances, right and above the front is the feasible region.

4.5.2 POFC of the VCO Block

For POFC on the VCO block, two constraints are set:

1. the maximal output frequency is larger than the expected output frequency, so that the VCO is capable of providing the 500MHz output frequency;
2. the nonlinearity limit of the gain is set to 2.5 to guarantee the acceptable linearity of VCO, i.e.

$$\frac{f_{\mathrm{mid}} - f_{\mathrm{min}}}{f_{\mathrm{max}} - f_{\mathrm{mid}}} \leq 2.5, \tag{4.36}$$

   where f_{mid} is the corresponding output frequency of the input voltage $V_{\mathrm{mid}} = (V_{\mathrm{max}} - V_{\mathrm{min}})/2$.

1	2	3	4	5	6
System & blocks	Performances	Specifications	Initial design	System-level optimization	Circuit-level optimization
System: CPPLL	Locking Time T_s (μs)	≤ 2.5	2.85	2.13	2.2
	Jitter J_{sum} ps	≤ 2.5	20.8	2.25	2.21
	Power (mW)	≤ 2.5	3.02	2.42	2.417
	Ratio RUR	≥ 20	30.6@ f_{min} 78.6@ f_{max}	28.5@ f_{min} 72.8@ f_{max}	28.7@ f_{min} 73.5@ f_{max}
	Phase Margin PM(°)	≥ 45	43.6@ f_{min} 48.0@ f_{max}	45.1@ f_{min} 45.1@ f_{max}	45.1@ f_{min} 45.0@ f_{max}
Block: CP	Current I_{CP} (μA)		50	47.3	I_{up}=46.5 I_{down}=-46.5
	Jitter J_{CP} (ps)		55.2	13.34	14.4
Block: VCO	Gain K_{VCO} (MHz/V)		270.3	347.1	348.4
	Current I_{VCO} (μA)		805.9	619.2	619
	Jitter J_{VCO} (ps)		13.7	1.88	1.876
Block: LF	Resistor R (KΩ)		20	16.98	16.98
	Capacitors C_1&C_2(pF)		50 & 8	44.5 & 7.66	44.5 & 7.66

Table 4.3: Hierarchical optimization results of the CPPLL

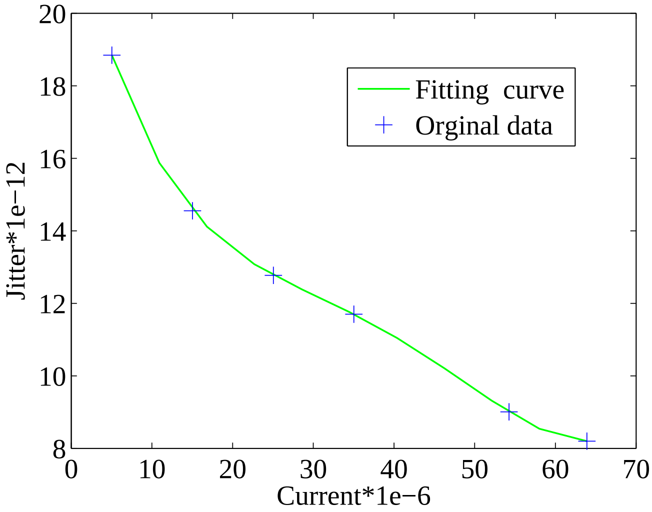


Figure 4.24: Pareto-optimal front of the CP

Large gain, small jitter and low current consumption are preferred in the VCO block. The 3D Pareto-optimal front for these three competing objectives is shown in Fig. 4.25(a). The contours of jitter on the Pareto front surface is shown in Fig. 4.25(b), which represents the jitter is reduced by increasing current consumption for a fixed gain.

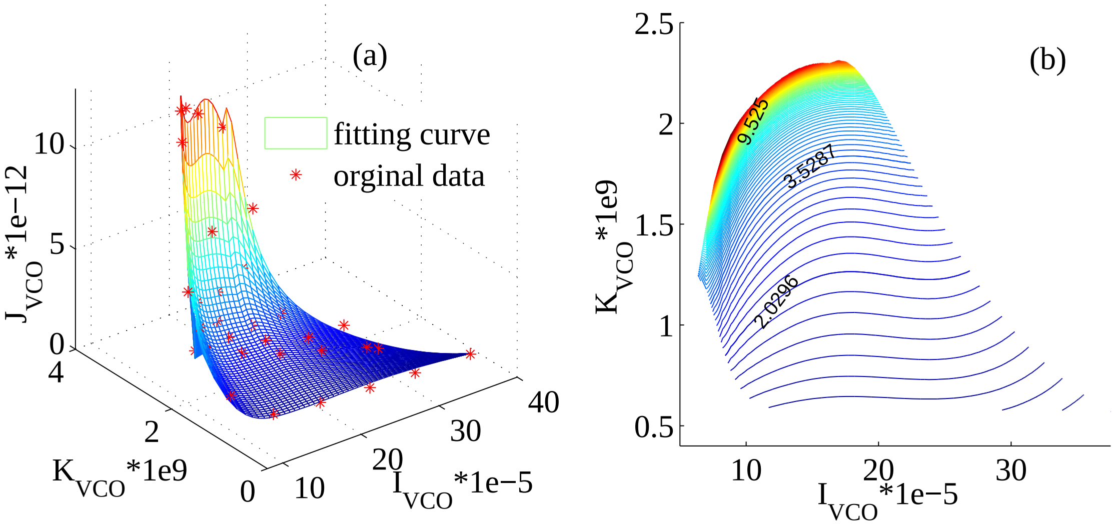


Figure 4.25: (a) 3D Pareto-optimal front of the VCO (b) Contours of jitter in 2D surface (gain vs. current)

4.5.3 POFC of the CPPLL System

For POFC on CPPLL system level, three constraints are set:

1. phase margin is larger than 45° for PLL stability in the lock-in state;
2. RUR is bigger than 20 so that PLL can work properly in the nonlinear acquisition process;
3. the system design parameters are restricted on the Pareto-optimal fronts of the building blocks, which are derived from POFC on circuit level.

In the CPPLL, $\mathbf{p} = [I_{\mathrm{CP}}, K_{\mathrm{VCO}}, I_{\mathrm{VCO}}, R, C_1]$ is selected as independent system-level parameter set, $N = 20$ is determined by the output frequency. C_2 is set to $C_1/10$ for the sake of good matching and safety phase margin.

Here we consider two different application cases. In the first case the dominant noise comes from the reference input, which is the case in clock recovery applications, where the input is random data. In the second case the dominant noise originates from the PLL blocks, which is the case in frequency synthesizers application, where the input is a clear signal source.

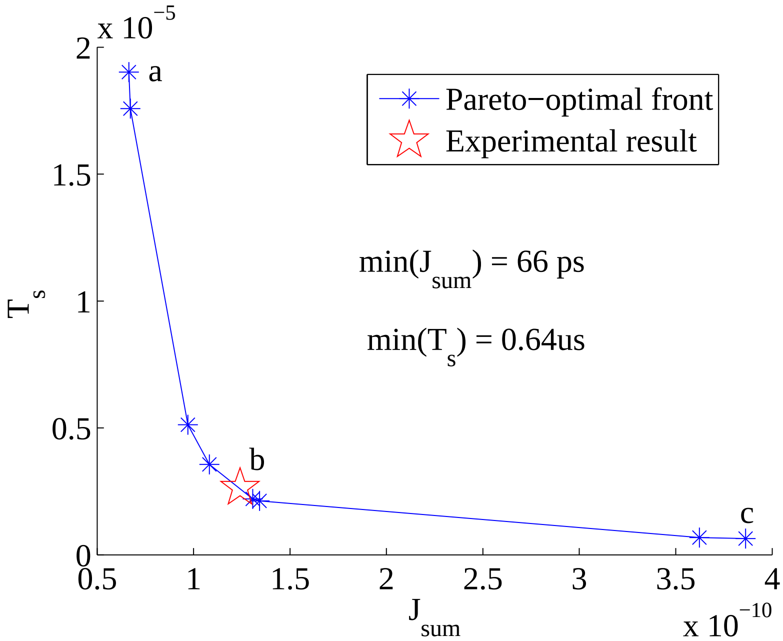


Figure 4.26: Pareto-optimal front of jitter and locking time at input jitter=1ns

In the first case, the loop bandwidth should be chosen to be small in order to suppress the reference noise. Assume the input jitter is 1ns, which corresponds to 4% distortion at the input signal (25MHz). Since the overall output jitter is dominated by the input noise, a clear trade-off exists between the locking time and the output jitter, as depicted in Fig. 4.26. The minimal jitter, point a, is obtained at the minimal loop bandwidth, which is produced by the minimal values of I_{CP} and K_{VCO}. The feasible regions of I_{CP} and K_{VCO} can be found in Figs. 4.24& 4.25. The minimal locking time, point c, is obtained at the maximum loop bandwidth, which is limited in theory by the RUR constraint. In this experiment, the maximum loop bandwidth happens to be generated by the maximal value of I_{CP} and K_{VCO}. The best performance, the constraints and the corresponding design parameters are listed in the upper half of Tab. 4.4.

In the second case, the reference input is clean and so the loop bandwidth should be chosen to be large to suppress the VCO noise. Meanwhile, a large loop bandwidth can provide a

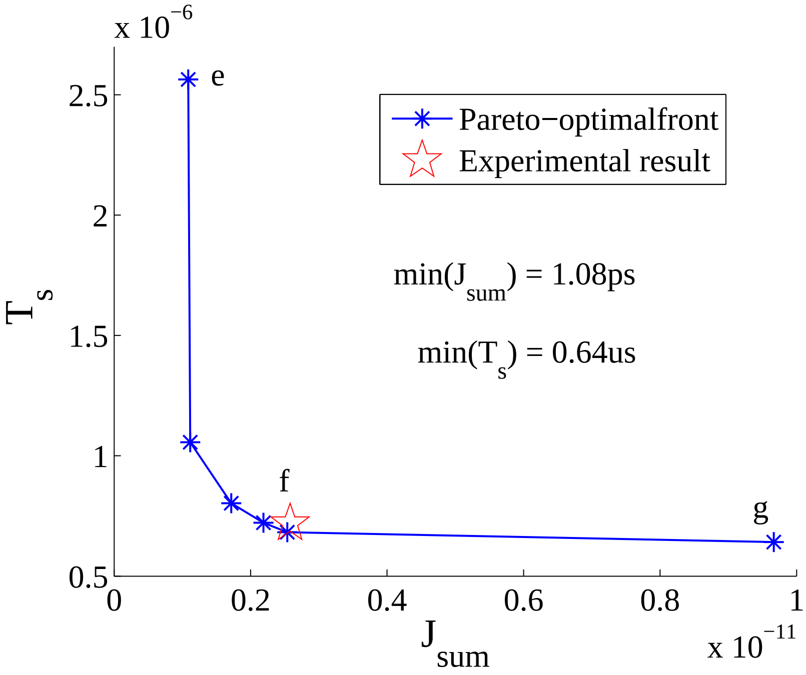


Figure 4.27: Pareto-optimal front of jitter and locking time at input jitter=0 and setting $I_{\mathrm{CP}} = I_{\max}$

quick acquisition process. At a first glance, it seems that there is no contradiction between reducing locking time and output jitter. From Sec. 2.3, the loop bandwidth ω_{BW} is increased by increasing I_{CP}. When I_{CP} reaches the maximum but the ω_{BW} is still not large enough, it can be further enlarged by increasing K_{VCO}. A larger K_{VCO} introduces more VCO noise, which results in a larger output jitter. Fig. 4.27 shows the trade-off between the jitter and the locking time, where the input noise is 0ps and the I_{CP} is set to maximum. The relationship of the loop bandwidth at the points e, f, g is: $\omega_e > \omega_f > \omega_g$. This indicates that the output jitter is increased much more by VCO noise than decreased by the enlarged loop bandwidth. Therefore, minimizing jitter is equivalent to minimizing VCO noise. The best performance, the constraints and the corresponding design parameters are listed in the lower half of Tab. 4.4.

The ultimate sizing goal is to optimize locking time, jitter and power consumption simultaneously. The performance and the corresponding system-level parameter sets for the two different cases introduced in Sec. 4.3 are collected in rows b and f in Tab. 4.4. These obtained values of the system-level parameters can be successfully realized by the final circuit level. Moreover, the two compromise optimization results, b and f, are also denoted by pentagrams in Fig. 4.26 and Fig. 4.27 respectively.

It is worth noting one point here. Based on the discrete computed efficient points, an entire Pareto-optimal front is estimated by means of a smooth fitting curve or surface in this work. Although the residual at certain efficient point ** in the fitting function is not small sometimes, the highest-level optimization results based on the fitting Pareto-optimal fronts can really realized by the final lowest-level circuit in the two optimization experiments. That proves again

** These efficient points might be generated due to non-robust algorithm of performance space exploration.

that the behavior of a circuit is usually well-natured so long as it works in the correct region of operation.

4.6 Summary

This chapter has discussed the basic concept behind of the phase-locked loop. The building blocks and their operation of a charge pump phase-locked loop are briefly explained. Three analysis methods: *s*-domain analysis, impulse invariance method and state space method are used to characterize the system properties of CPPLL in linear lock-in state and nonlinear acquisition process.

The PLL's performances, i.e. locking time, phase noise/jitter are discussed in detail. Additionally, the PLLs' stability criteria for linear lock-in state and for nonlinear acquisition process are discussed individually. The design trade-offs among the loop bandwidth, locking time and jitter are analyzed in theory and visualized by the experimental results.

In this work, to tackle the complex trade-offs in the PLL design, we present an optimization method to find a proper loop bandwidth in order to optimize performance in terms of locking time and jitter, which considers the capability of the building blocks. The experimental results are consistent with the theoretical analysis. The Pareto-front computation on system level provides a comprehensive insight into the circuit's capability. In clock recovery or frequency synthesizer applications, starting from various requirements on the PLL's loop bandwidth to final circuit realization, the automatic sizing process can be accomplished in hours. Moreover, it is a first-time-successful top-down sizing process without iteration. Without the proposed hierarchical approach, a complete CPPLL design could need many iteration steps due to complexity problems.

	Jitter J_{sum} (ps)	Locking Time T_s (µs)	Power (µw)	RUR	PM	VCO	CP	LF
Case 1: Input Jitter=1ns								
a	386	0.64	642	20	45°	K_{VCO}=2.2e9 Hz/V I_{VCO}=161.5µA J_{VCO}=8.41ps	I_{CP}=65µA J_{CP}=8.21ps	C_1=27.2pF C_2=2.72pF R=5.9k
b	124.1	2.64	209	75	46.4°	K_{VCO}=1.537e9 Hz/V I_{VCO}=82.41µA J_{VCO}=5.6ps	I_{CP}=11.3µA J_{CP}=15.74ps	C_1=48pF C_2=2.72pF R=13.4k
c	66.4	19.02	279	244	45°	K_{VCO}=573.6e6 Hz/V I_{VCO}=140.2µA J_{VCO}=1.048ps	I_{CP}=5µA J_{CP}=18.87ps	C_1=80pF C_2=8pF R=24.5k
Case 2: Input Jitter=0								
e	9.67	0.64	652	20	45°	K_{VCO}=2.2e9 Hz/V I_{VCO}=167µA J_{VCO}=7.93ps	I_{CP}=65µA J_{CP}=8.21ps	C_1=27.2pF C_2=2.72pF R=5.9k
f	2.58	0.72	662	23.4	46.6°	K_{VCO}=1.414e9 Hz/V I_{VCO}=173µA J_{VCO}=2.28ps	I_{CP}=65µA J_{CP}=8.21ps	C_1=23pF C_2=2.3pF R=7.98k
g	1.08	2.56	669	50.2	56°	K_{VCO}=573.6e6 Hz/V I_{VCO}=175µA J_{VCO}=1.02ps	I_{CP}=65µA J_{CP}=8.21ps	C_1=80pF C_2=8pF R=11.3k

Table 4.4: Hierarchical optimization results at two different cases

Chapter 5

Hierarchical Optimization of Switched-Capacitor Sigma-Delta Modulators

Data modulator is an indispensable circuit in nowadays chip, it converts analog signal $x(t)$ in time-continuous domain to digital signal $x(n)$ in discrete-time domain or converts the signal in the reverse process. In last decades, oversampling data converters become more popular for high resolution medium-to-low-speed applications such as high-quality digital audio. "Oversampling is the process of sampling a signal with a sampling frequency significantly higher than twice the bandwidth or highest frequency of the signal being sampled" [wik]. The major advantages of oversampling method compared to other converter architectures are that high-selectivity analog filters are not needed and the conversion properties are much less sensitive to the circuitry imperfections and noisy environment.

Switch capacitor (SC) sigma-delta ($\Sigma\Delta$) modulator is an effective topology for high resolution Analog-to-Digital (A/D) conversion. SC $\Sigma\Delta$ modulators use not only the oversampling method but the noise shaping method as well. Hence, SC $\Sigma\Delta$ modulators inherit the oversampling's advantages, such as high tolerance to circuitry non-idealities and reduced accuracy requirements on sample-and-hold circuit. Despite such predominant properties of SC $\Sigma\Delta$ modulators, "it is still governed by the limitations of its analog building blocks. In particular, it is sensitive to circuit non-idealities at the input stage where no noise shaping has yet taken place" [TMG02]. To extract the effects of the non-idealities of analog blocks on the circuit performance, a circuit-level simulation is the direct way generally. However the circuit-level simulation of SC $\Sigma\Delta$ modulators is a huge time-consuming and large memory-capacity required process, which results in that it is unfeasible on such circuit by directly using the "simulation-in-a-loop" based optimization approaches stated in [WRSVT88, AEGP00, HBL01, PKR$^+$00, dPDL$^+$01]. A hierarchical process is needed for optimization on SC $\Sigma\Delta$ modulators.

In this chapter, Sec. 5.1 introduces the fundamentals of $\Sigma\Delta$ oversampling modulators. Sec. 5.2 reviews the topology of a second-order SC $\Sigma\Delta$ modulator and lists out the major non-idealities of the building blocks. Sec. 5.4 shows the proposed hierarchical optimization method applied to the second-order SC $\Sigma\Delta$ modulator. The SNR performance is maximized at the nominal design case and at the worst design case respectively. Finally, Sec. 5.5 concludes.

5.1 ΣΔ Oversampling A/D Converters

In this section, the basic topology of conventional converters and of ΣΔ oversampling A/D converters are briefly described as follows. More details can be found in [FM98, Par]. A conventional Nyquist A/D converter consists mainly of three blocks, as shown in Fig. 5.1(a.1):

- *Anti-aliasing filter*: It is a low-pass filter and prevents the input signals above Nyquist frequency to bring new components in the base-band, which don't belong to the original signal.
- *Sample-hold circuit*: By using a sampling signal with frequency f_{s1}, the sample-hold circuit samples a continuously changing input signal each $1/f_{s1}$ time interval and provides a constant signal for the subsequent quantization process. If the sampled signal has a f_b frequency in Nyquist converters, the sampling frequency has to be twice times of f_b in order to avoid any aliasing, i.e. Nyquist frequency $f_{s1} = 2 \cdot f_b$.
- *Multi-level quantizer*: The sampled analog signals are compared with the predefined reference levels, and then the comparison results will be digitalized as the output signals.

In a N-bit Nyquist converter, the number of the quantization level is 2^N and the number of the level interval is $2^N - 1$. The interval value q between two successive levels is referred to as least-significant-bit (LSB) of the converter:

$$q = \frac{1}{2^N - 1}. \tag{5.1}$$

This conversion has a quantization error up to $\pm q/2$ with same probability and the quantization noise power is equally spread over the entire signal bandwidth. Assumed the above three building blocks are ideal, a perfect classical N-bit Nyquist A/D converter has quantization noise power spectral density of $q/(\sqrt{12} \cdot f_{s1})$ uniformly distributed within the signal bandwidth. The noise spectrum is presented only at one side here, as shown in Fig. 5.1(a.2). The *signal-to-noise ratio* (SNR) * performance of an ideal N-bit Nyquist converter is [Bak04]

$$\mathrm{SNR}_{\mathrm{ideal}} = (6.02N + 1.76) \quad \mathrm{dB}. \tag{5.2}$$

However the A/D conversion is not ideal in practice. For example, the undesired signal above the Nyquist frequency cannot be attenuated enough to below the noise floor by the anti-aliasing filter, or the output of the sample-and-hold circuit varies during the quantization process. Consequently, the quantization noise will be larger than its theoretical minimum value, then its effective resolution will be less than N-bit. The actual resolution, i.e. *Effective Number of Bits* (ENOB) is defined by

$$\mathrm{ENOB} = \frac{\mathrm{SNR} - 1.76\mathrm{dB}}{6.02\mathrm{dB}}. \tag{5.3}$$

In oder to achieve high-resolution by Nyquist converter, high performances for each building block are required.

In oversampling converters, a much higher sampling rate f_{s2} (=$R \cdot 2 f_b$) is applied in sample-hold circuit. The factor R is generally referred to as *oversampling ratio* (OSR). By the price of

* SNR is defined as the ratio between the signal power and the noise power: $\mathrm{SNR} = P_{\mathrm{signal}}/P_{\mathrm{noise}}$. SNR includes all noise sources, both thermal and quantization.

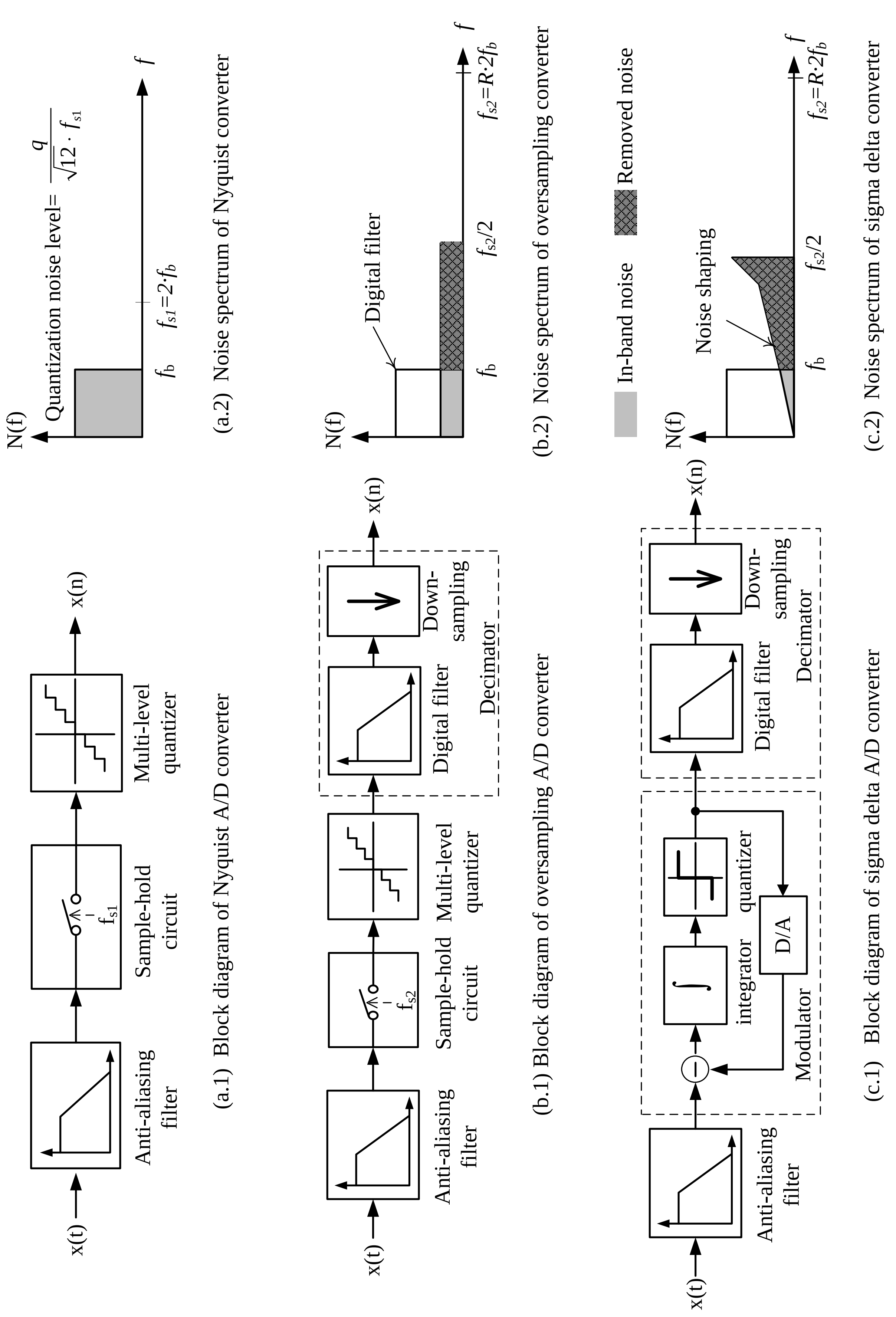


Figure 5.1: Block diagram of A/D Converters and Noise spectrum

high sampling rate at the input signal, oversampling method brings two benefits. One is that the requirement of anti-aliasing filter in oversampling converter is much more lower than that in Nyquist converter. A simple passive first-order filter suffices mostly. The another benefit is that a high resolution A/D conversion is accomplished with a low-bit A/D converter. Assumed a full precision quantizer, the total noise power of oversampling converter is same as that of Nyquist converter. However, the quantization noise is distributed over a wider bandwidth to $f_{s2}/2$, i.e. $R \cdot f_b$ (this phenomenon is normally called *noise averaging*), as shown in Fig. 5.1(b.2). If a digital low-pass filter (LPF) is added at the output, then most part of the quantization noise can be removed, moreover the wanted in-band signal is not affected. Through the noise averaging, the power of quantization noise in the bandwidth of interest is decreased by factor R. The ideal SNR using oversampling method is calculated by [Bak04]

$$\mathrm{SNR}_{\mathrm{ideal}} = (6.02N + 1.76 + 10\log R) \quad \mathrm{dB}. \tag{5.4}$$

If OSR is 2, then $\mathrm{SNR}_{\mathrm{ideal}}$ can be increased by 3dB or the ENOB can be improved by 0.5 bits. In oversampling converters, decimator is needed besides the three blocks mentioned above, as shown in Fig. 5.1(b.1).

- *Decimator*: It filters all the signal components out of the signal band, which includes a big part of the quantization error power. The filtered signal is downsampled to the Nyquist rate without degrading SNR performance. The collective operation of the low-pass filtering and the downsampling is known as decimation, which can be realized by purely digital circuit.

If only using oversampling method to increase N-bit resolution more, the sample rate has to be faster by a factor of 2^{2N}. The ΣΔ converter does not need such a high oversampling ratio because the operation of modulator (in Fig. 5.1(c.1)) not only reduces the quantization noise power within the signal band, but also pushes quantization noise power from the signal band to outside of the signal band (this phenomenon is normally called *noise shaping*), as shown in Fig. 5.1(c.2). Consequently, the ideal SNR by oversampling plus one-order noise shaping process is [Bak04]

$$\mathrm{SNR}_{\mathrm{ideal}} = (6.02N + 1.76 - 5.17 + 30\log R) \quad \mathrm{dB}. \tag{5.5}$$

The details of the noise shaping process in modulator will be discussed in Sec. 5.3.

- *Modulator*: In this block the signal is over-sampled and quantized. Same as oversampling converter, it spreads the whole quantization noise power (a fixed value) over the frequency range with bandwidth of $R \cdot f_b$ in order to reduce the quantization noise power within the signal band. Furthermore, the modulator attenuates the quantization noise within the signal band and amplifies it outside the signal band by means of noise shaping process. In consequence, most of the noise power lies out of the signal band. These out-of-band quantization noise could be filtered by decimator later.

Through noise averaging and noise shaping methods, the requirements on some building blocks are quite low in ΣΔ A/D converter. The anti-aliasing filter can be realized by a simple RC low-pass filter. The decimator is a pure digital block and can be designed with the help of mature CAD tools. In contrast, the modulator is in charge of the simultaneous implementation of the noise averaging and the noise shaping. The enclosed intrinsic errors of modulator, e.g. inherent quantization errors and imperfections (non-idealities) of circuit performances, will degrade the converter performance [FAB99]. Therefore, the modulator block is taken as the to be optimized circuit in this work.

A large amount of ΣΔ modulator architectures have been developed since 1980s. Tab. 5.1 [FAB99] summarize the respective advantages and disadvantages of the most used ΣΔ modulator architectures. Based on system stability, oversampling ratio, circuit complexity and circuit sensitivity, second-order ΣΔ modulators are most selected architectures for high-resolution application. Their effectiveness has already been illustrated in a variety of applications, such as digital speech processing systems and voice-band telecommunication in [KHE+86, BW88].

	Advantages	**Disadvantages**
Single-loop, 1-bit low-order [LNGB88]	⋆ guaranteed stability ⋆ simple circuitry ⋆ maximum useful input range	○ need of high OSR ○ presence of noise patterns
Single-loop, 1-bit high-order [AP87]	⋆ large SNR by using low OSR ⋆ smaller noise pattern	○ potential instability ○ useful input range smaller than full-scale range ○ need of low gain integrators
high-order cascade [HIUK86]	⋆ guaranteed stability ⋆ large SNR by using low OSR ⋆ maximum useful input range	○ sensitivity to circuit imperfections ○ larger complexity of the digital part
Multi-bit [CSC97]	⋆ better stability ⋆ large SNR by using low OSR ⋆ smaller noise patterns	○ more complex digital and analog circuitry ○ sensitivity to multi-bit DAC nonlinearity

Table 5.1: Summary of ΣΔ modulator architectures based on [FAB99]

5.2 Second-Order Switched-Capacitor Sigma-Delta Modulators

The ΣΔ modulator can be implemented as a discrete-time system (switched-capacitor, switched OP AMP) [BW88, GSS00] or as a continuous-time one (active RC or transconductor-C) [SZ96, CS00]. Compared to the continuous-time realization, the implementation using switched-capacitor circuit is compatible with standard CMOS process and is insensitive to clock jitter. Moreover, the frequency response of the noise shaping filter can be accurately set by the relative capacitor ratios. For this reason, a second-order SC ΣΔ modulators is taken in this thesis.

5.2.1 Building Blocks of a second-order SC ΣΔ Modulator

A topology of the second-order SC ΣΔ modulator [BW88] is shown in Fig. 5.2, consisting of two switched-capacitor integrators, a comparator and a 1-bit D/A converter.

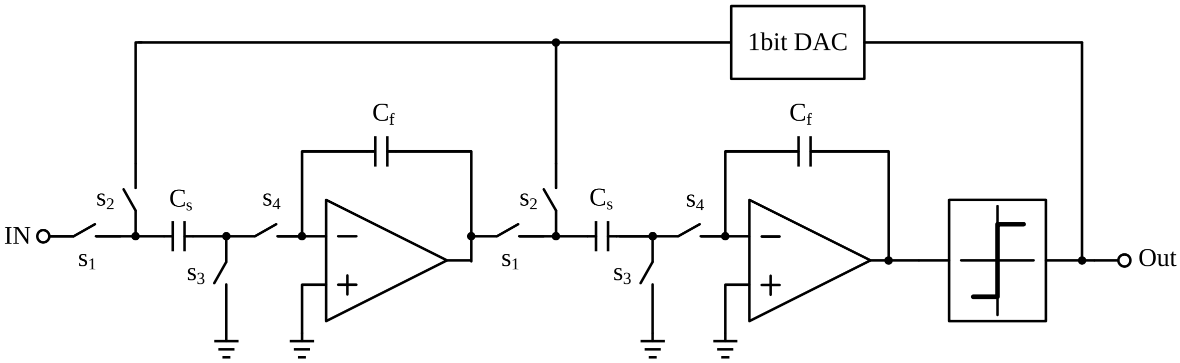


Figure 5.2: A second-order SC ΣΔ modulator [BW88]

5.2.1.1 Switched-Capacitor Integrators

The two switched-capacitor integrators in Fig. 5.2 are identical. Each consists of an operation amplifier (OP AMP), a sampling capacitor C_s and an integrating capacitor C_f. Compared to a standard RC integrator, switched-capacitor (SC) integrator has two major advantages:

- Much reduced area to realize the same functionality: as seen in Fig. 5.3, a resistor is replaced by the sampling/integrating capacitors and the periodical turn-on/off switches. For a same time constant ($\tau = R_\mathrm{eq}C_f$), a meg-ohm resistance can be replaced by a pico-farad MOS capacitance with a proper sampling rate.
- More stable time constant: the time constant of the RC integrator is closely related to the absolute values of the resistance and the capacitance, while the time constant of the SC integrator is determined by the relative ratio of the capacitances. The absolute values of resistance and capacitance can vary $\pm 20 \sim 30\%$ in silicon realization. In contrast, capacitances can superiorly match with each other, its relative ratio is much more constant than its absolute values. Hence, the time constant is insensitive to the variations of process and of temperature.

In this experiment, the ratio of $C_\mathrm{s}/C_\mathrm{f}$ is set as Equation 5.6 in oder to realize the gain of 0.5, i.e.

$$b = \frac{C_\mathrm{s}}{C_\mathrm{f}} = 0.5. \tag{5.6}$$

A SC integrator has two operation phases, which are controlled by two non-overlapping clocks. The switches are closed when the controlling clocks are high and vice versa. The corresponding time diagram for all switches in the modulator is shown in Fig. 5.3. During ϕ1, switches s_1&s_3 are closed while s_2&s_4 are open, and the capacitor C_s of the integrator is charged by the input V_{in}. During ϕ2, switches s_2&s_4 are closed while s_1&s_3 are open, the charge stored on C_s is transferred to C_f. Switches s_1&s_2 are closed after a little delay Δt of switches s_3&s_4 respectively in order to avoid charge injection on C_s.

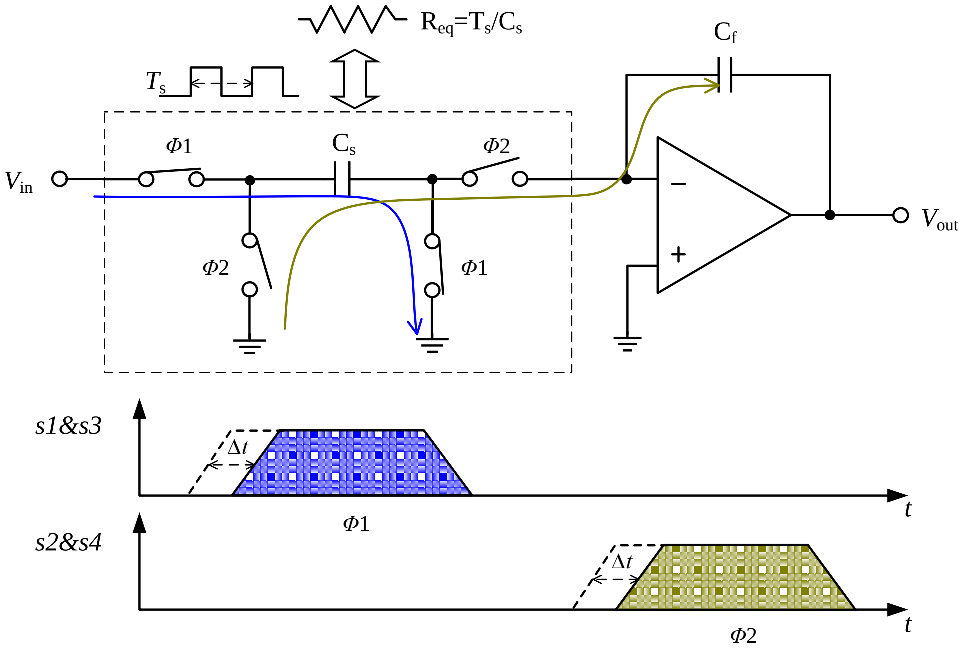


Figure 5.3: Single-ended SC Integrator

The SC integrator is a key building block in ΣΔ modulators. Its non-idealities, e.g. switch thermal noise, finite DC gain and slew rate of OP AMP, have strong impact on the SNR performance of ΣΔ modulators. We will discuss that in detail in Sec. 5.3.1.3.

5.2.1.2 Comparator

A comparator is used to quantize an analog signal in the loop. The generated digital output will be directly taken as the output signal and be fed back to the D/A converter at the same time. As shown in Fig. 5.2, the comparator lies after the integrator block, its non-idealities are shaped by the loop in the same way that quantization noise is shaped. Therefore, the impact of the comparator's non-idealities can be ignored. Generally, "a simple regenerative latch without pre-amplification or offset cancellation fulfills the comparator requirements" [BW88].

5.2.1.3 1-bit D/A Converter

The D/A converter transforms the digital output signal back into an analog form, which acts as another input of the modulator in addition to the reference input signal. Any D/A conversion error will result in distortion at the modulator's output. Since the conversion error is not attenuated, the DAC's non-linearity conversion will considerably hurt the performance of modulators.

In order to minimize the impact of non-idealities D/A converter on the ΣΔ modulators, 1-bit D/A converter is selected here. The advantage of an 1-bit D/A converter is that it is inherently linear and guarantees the D/A conversion from differential nonlinearity, because there are only two output values and only one step size.

5.3 Analysis on ΣΔ modulator in z-domain

In principle, quantization process is an intrinsic non-linear procedure, so a ΣΔ modulator is also an inherently non-linear system. To analyze the behavior of a ΣΔ modulator, it can still be implemented by a linear operation if the quantization error is assumed to be additive white noise by fulfilling the following four conditions [Mal02]:

I. "All quantization levels are exercised with equal probability.

II. The quantization steps are uniform.

III. The quantization error is not correlated with the input signal.

IV. A large number of quantization levels are used †".

A general linear model for a modulator is shown in Fig. 5.4. It is a two-input ($x(n)$ & $e(n)$) and one-output ($y(n)$) system, whose z-domain expression is represented by

$$\begin{aligned} Y(z) &= STF(z)X(z) + NTF(z)E(z) \\ &= \frac{H(z)}{1+H(z)}X(z) + \frac{1}{1+H(z)}E(z), \end{aligned} \tag{5.7}$$

where $X(z)$ and $E(z)$ presents the input signal and the quantization noise respectively; and $STF(z)$ and $NTF(z)$ are the respective z-domain transfer functions of the input and the quantization noise.

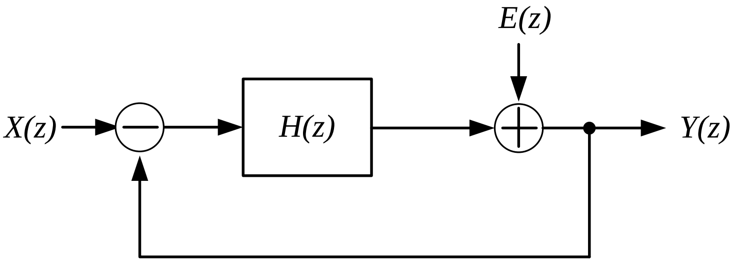


Figure 5.4: Linear model of the modulator with an injected quantization noise

For the 2$^{\text{nd}}$-order SC ΣΔ modulator topology in Fig. 5.2, the corresponding linear model is shown in Fig. 5.5. The ideal integrator's z-domain transfer function including the necessary closed loop gain are given by $H_{1,2}(z)$, i.e.

$$H_1(z) = H_2(z) = 0.5\frac{z^{-1}}{1-z^{-1}}. \tag{5.8}$$

† Although only one-level quantization is used in the second-order SC ΣΔ modulator, the quantization error is still assumed to be white noise, since the quantization noise will be firstly sharpened.

Thus, the ideal signal transfer function is given by

$$STF(z) = \frac{H_1H_2}{1+H_1H_2+H_2} = \frac{0.25}{z^2-1.5z+0.75}. \tag{5.9}$$

Whereas, the noise transfer function is to be

$$NTF(z) = \frac{1}{1+H_1H_2+H_2} = \frac{z^2-2z+1}{z^2-1.5z+0.75}. \tag{5.10}$$

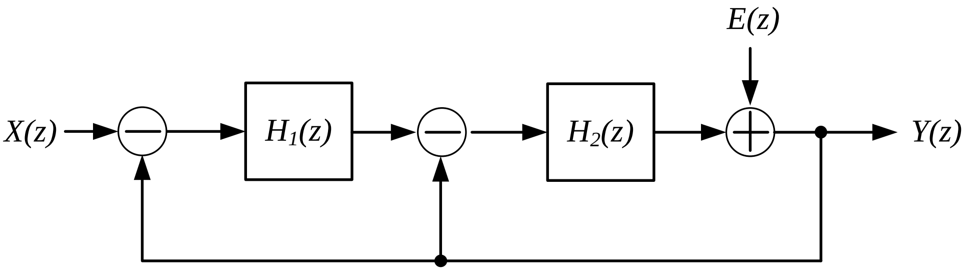


Figure 5.5: Ideal linear Model of the 2nd-order SC ΣΔ modulator

As can be seen, the $STF(z)$ in Equation 5.9 is a low-pass function, while $NTF(z)$ in Equation 5.10 is a high-pass function. Therefore, the 2nd-order SC ΣΔ modulator can suppress the quantization noise in the signal band and will not attenuate the input signal.

Fig. 5.6 shows the power spectral density (PSD) of an ideal 2nd-order SC ΣΔ modulator when the OSR is set to 256. As seen in this figure, the PSD of noise power is attenuated in the signal

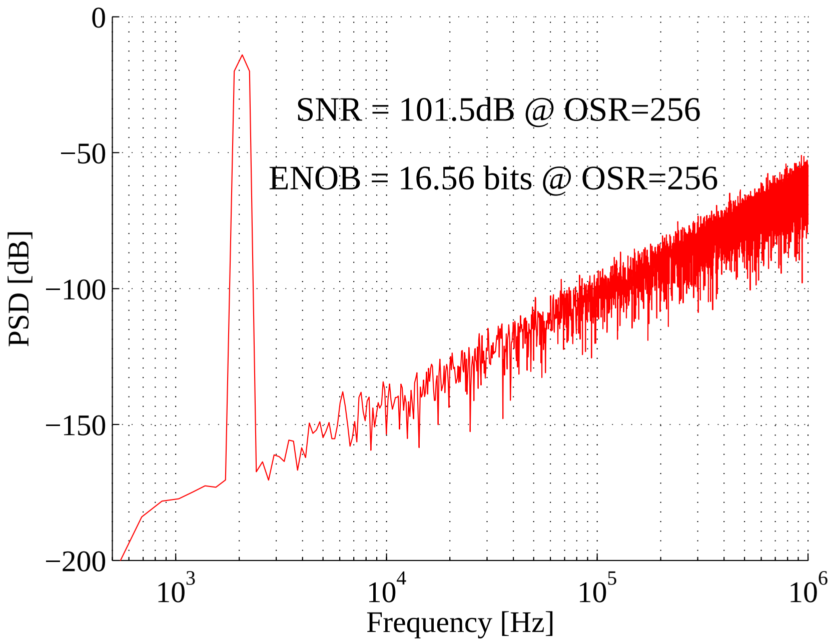


Figure 5.6: PSD plot of the ideal 2nd-order SC ΣΔ modulator @ OSR=256:

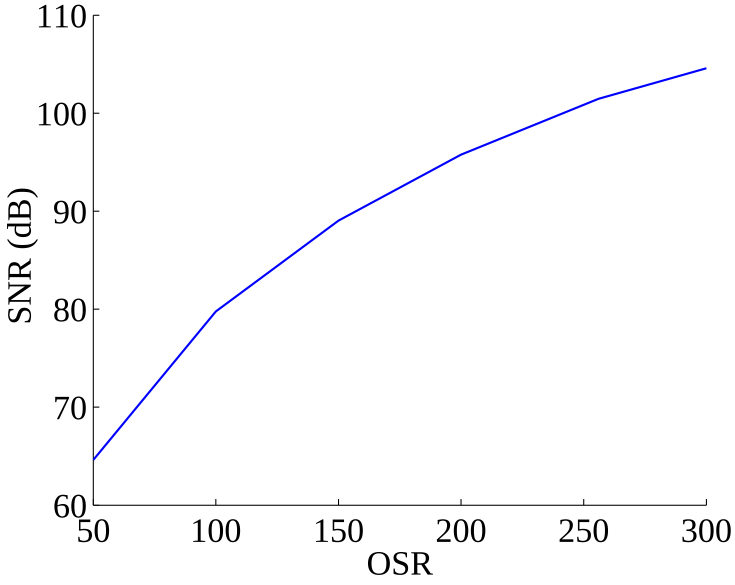


Figure 5.7: SNR vs. OSR in the ideal 2^{nd}-order SC $\Sigma\Delta$ modulator

band and is amplified outside the signal band. With 256 oversampling ratio, we can obtain the maximal SNR is 101.5dB and the corresponding ENOB is 16.56bit. The higher OSR is, the lower PSD quantization noise is in the signal band. Consequently, SNR increases with the increasing OSR as shown in Fig. 5.7.

5.3.1 Effects of Non-idealities

Sole quantization noise is taken into account in the ideal linear model of $\Sigma\Delta$ modulator in the last section. Although $\Sigma\Delta$ modulator is usually well-known for its robustness to the non-idealities of the building block compared to other data conversion architectures [CT92], it is still necessary to take into account such non-idealities of the electrical implementations and the corresponding introduced conversion error. Actually, the impact associated with such imperfections increases when the modulator specifications are highly demanded because they become the dominant error sources [BKMA88,DLM92]. The detailed analysis on the non-idealities of building blocks are described in [BW88, FAB99]. Here, the effects of non-idealities on the SNR performance are presented and the corresponding modeling of these non-idealities are built. The main non-idealities of SC $\Sigma\Delta$ modulators [BFM$^+$99, MBF$^+$03] can be classified into three categories, shown as follows:

- clock jitter
- integrator noise
  - switches thermal noise
  - OP AMP noise
- integrator non-idealities

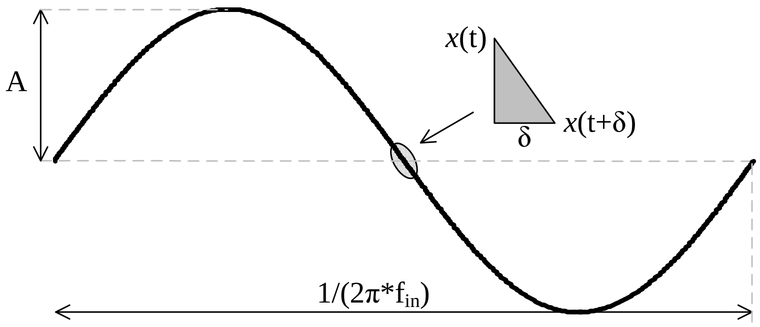


Figure 5.8: Clock jitter on sampling input signal

- DC gain
- bandwidth
- slew rate
- saturation (output range)

5.3.1.1 Clock Jitter

In SC ΣΔ modulators, the property of SC circuits is closely dependent on the charge transfer process during each clock phase. As seen in Fig. 5.3, the four clock signals are used to trigger the switches in order to sample the input analog signal. Due to the clock jitters on the four clock signals, a non-uniform sampling time sequence is applied on the input signal, which results in nonlinear distortions of the sampled signal. In consequence, the noise floor or the total harmonic distortion of modulator is increased.

The distortion of the sampled signal is a function related to the statistical property of the clock jitter and the modulator input signal. If the input signal is a sinusoidal signal $x(t)$ with amplitude A and frequency f_{in}, the distortion introduced by a sampling with an instant clock jitter (δ) is

$$x(t+\delta)-x(t)=\frac{d}{dt}x(t)\delta \approx 2\pi f_{\text{in}}Acos(2\pi f_{\text{in}}t)\delta. \tag{5.11}$$

Fig. 5.8 illustrates the relationship between distortion and clock jitter. Since clock jitter is assumed as white noise in this thesis, so the oversampling process is helpful to reduce the signal distortion introduced by the clock jitter. The total power of the distortion is $(2\pi f_{\text{in}}\delta\tau A)^2/2$ and uniformly distributed from 0 to $f_s/2$ [BW88].

Fig. 5.9 shows the PSD of the 2$^{\text{nd}}$-order SC ΣΔ modulator with the clock jitter when the OSR is set to 256. The SNR performance is reduced from 101.5dB (only considering the inherent quantization noise) to 99.95dB (considering additional clock jitter with standard deviation $\Delta\tau = 4e-9$). This effect can be simulated with SIMULINK by using the model in Fig. B.1 in Appendix B, which represents Equation 5.11.

5.3.1.2 Noise Sources

In the real circuit implementation of ΣΔ modulators, the signal is corrupted not only by the intrinsic quantization noise, but by various electrical noises in the building blocks as well. A

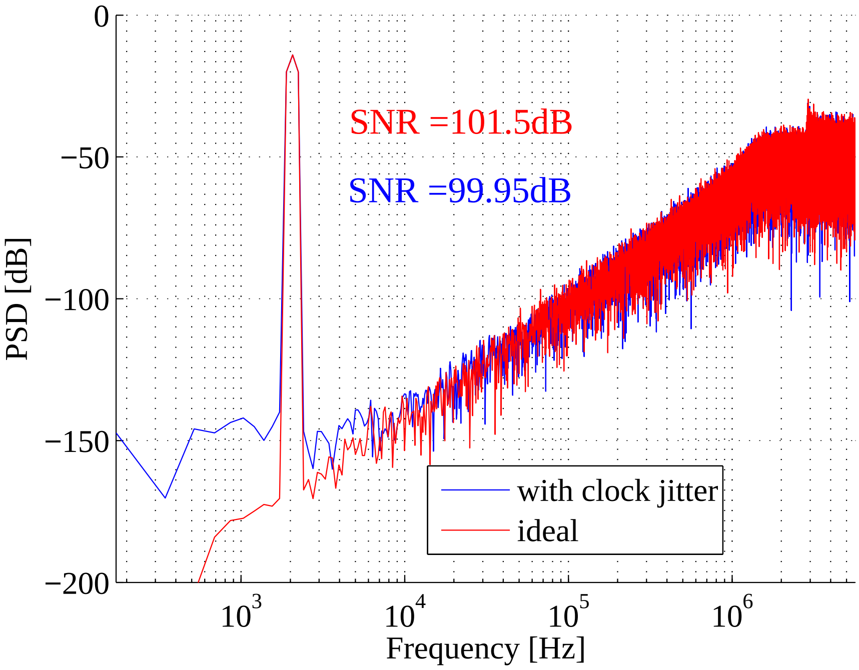


Figure 5.9: Effect of clock jitter

typical switched-capacitor integrator is already shown in Fig. 5.3. The most important noise sources affecting the operation of the SC integrator are thermal noise associated with the sampling switches and the intrinsic noise of the operational amplifier [MBF$^+$03].

Switch Thermal Noise

Switches in SC integrators are implemented with CMOS transistors. When a switch is on, the CMOS transistor works in triode region and has a finite on-resistance that introduces thermal noise. Thermal noise is an electronic noise generated by the thermal agitation of the charge carriers (usually the electrons) and usually presents in equilibrium. Thermal noise has a white spectrum and wide band, is limited only by the time constant of the switched capacitors or the bandwidth of the OP AMP [MBF$^+$03].

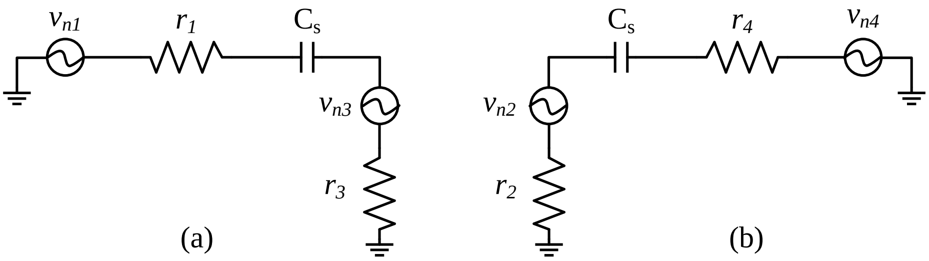


Figure 5.10: Equivalent SC circuits in (a) ϕ1 (b) ϕ2

Referring to the SC integrator shown in Fig. 5.3, the sampling capacitor C_s is in series with switches, that periodically open and close with finite resistance. The equivalent circuit is shown in Fig. 5.10. r_1, r_2, r_3 and r_4 are the on-resistances of switches s_1, s_2, s_3 and s_4, respectively. v_{n1}, v_{n2} v_{n3} and v_{n4} are white Gaussian noise sources, which modulate the thermal noise of r_1,

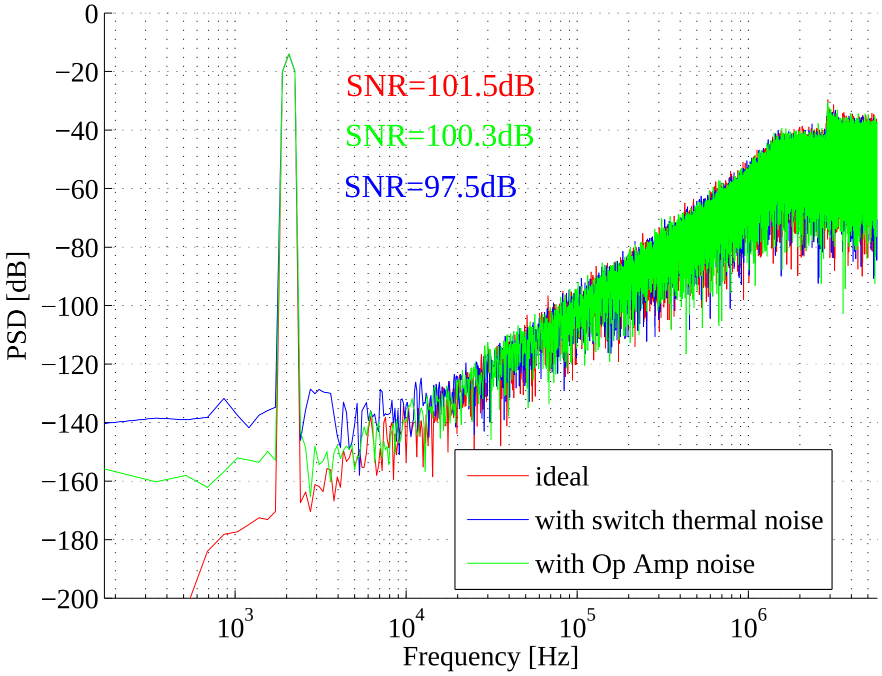


Figure 5.11: Effect of noise

r_2, r_3 and r_4, respectively. Thus noise sources are sampled periodically onto C_s. The equivalent circuit is a simple RC circuit at both phases ϕ_1 and ϕ_2. Therefore, the total noise power can be evaluated by [NST97]

$$e_T^2 = \int_0^\infty \frac{4kTR_{\text{on}}}{1+(2\pi R_{\text{on}}C_s)^2}df = \frac{kT}{C_s}, \tag{5.12}$$

where k is the Boltzmann's constant, T is the absolute temperature, and $4kTR_{\text{on}}$ is the noise PSD associated with the switch on-resistance. The corresponding model for switch thermal noise in simulink is shown in Fig. B.2.

Operational Amplifier Noise

The intrinsic noise of operational amplifier includes thermal noise, flicker (1/f) noise, shot noise etc. For these purely random noises, it is very hard to predict their instantaneous values at any time. The usual method to calculate random noise is the average mean-square value of the noise. When having multiple noise sources in a circuit, all noises can be represented together by a total root mean square (rms) noise source, which is the square root of the sum of the average mean-square values of each individual source. In the behavioral model of OP AMP, an input-referred noise source with rms noise voltage V_n is used to present the intrinsic noises in OP AMP:

$$y(t) = b \cdot [x(t) + n_{\text{opamp}}(t)], \tag{5.13}$$

where

$$n_{\text{opamp}} = V_n \cdot RN(t). \tag{5.14}$$

b is the integrator gain, and $RN(t)$ is a Gaussian random number with zero mean and unity standard deviation. In this experiment, only thermal noise is considered, while other noises are neglected. The corresponding model for thermal noise is shown in Fig. B.3.

Fig. 5.11 shows the PSD of the 2nd-order SC ΣΔ modulator with the switch thermal noise and the operation amplifier noise respectively, when the OSR is set to 256.

5.3.1.3 Operational Amplifier Non-idealities

In Sec. 5.3, the transfer functions of input signal and quantization noise ($STF(Z)$ and $NTF(Z)$, respectively) are based on an ideal integration process. In practice, the real behavior of an integrator deviates from this ideal behavior due to several non-idealities of the analog circuit implementation. The non-ideal effect of the integrator is a consequence of the OP AMP's non-idealities. Such as finite DC gain and bandwidth, slew rate and limited voltage region will result in an incomplete transfer of charge in SC integrators. These non-idealities will be discussed separately in the following.

DC Gain To simplify analysis here, the effect of an infinite DC gain of OP AMP in a first-order ΣΔ modulator are formulated here, and the following equations can be extended to the higher-oder modulators. The transfer function of an ideal integrator with unity coefficient is

$$H_{\text{ideal}}(z) = \frac{z^{-1}}{1-z^{-1}}. \tag{5.15}$$

For a first-order ΣΔ modulator, the ideal z-domain transforms of the input signal and the quantization noise are expressed as

$$\begin{aligned} Y(z) &= STF_{\text{ideal}}(z)X(z) + NTF_{\text{ideal}}(z)E(z) \\ &= \frac{H_{\text{ideal}}(z)}{1+H_{\text{ideal}}(z)}X(z) + \frac{1}{1+H_{\text{ideal}}(z)}E(z) \\ &= z^{-1}X(z) + (1-z^{-1})E(z), \end{aligned} \tag{5.16}$$

where

$$\begin{aligned} |STF_{\text{ideal}}(z)| &= 1 \\ |NTF_{\text{ideal}}(z)| &\to 0 \end{aligned} \quad \text{for} \quad z \to 1 \quad \text{i.e.} \quad f \to 0. \tag{5.17}$$

As we can see from Equation 5.16, the output of the modulator is the delayed input signal plus the in-band shaped quantization noise. In practice, a real integrator built by an OP AMP with a finite DC gain A_0 provides a real transfer function:

$$H_{\text{real}}(z) = \frac{z^{-1}}{1-(1-\alpha)z^{-1}} \quad \text{with} \quad \alpha = \frac{1}{A_0}. \tag{5.18}$$

Therefore, the real z-domain transforms of the input signal and the quantization noise can be expressed as

$$\begin{aligned} Y(z) &= STF_{\text{real}}(z)X(z) + NTF_{\text{real}}(z)E(z) \\ &= \frac{H_{\text{real}}(z)}{1+H_{\text{real}}(z)}X(z) + \frac{1}{1+H_{\text{real}}(z)}E(z) \\ &= \frac{z^{-1}}{1+\alpha z^{-1}}X(z) + \frac{1+(1-\alpha)z^{-1}}{1+\alpha z^{-1}}E(z), \end{aligned} \tag{5.19}$$

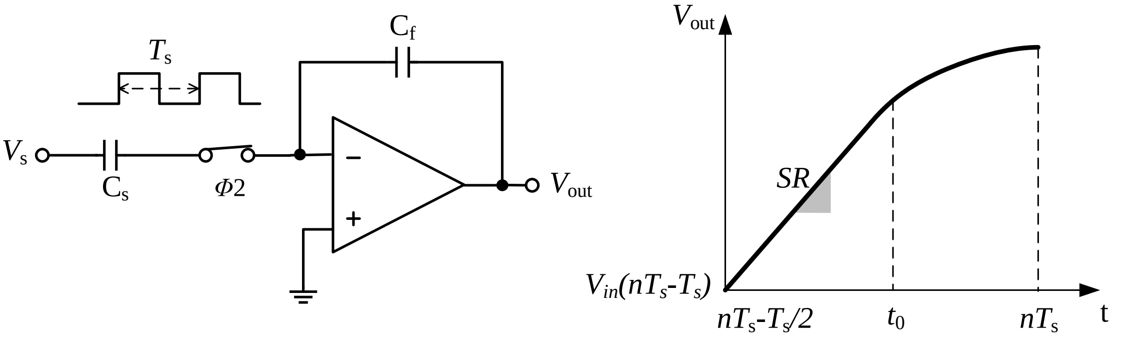


Figure 5.12: Integrator output response

where

$$\begin{array}{ll} |STF_{\text{real}}(z)| = |\frac{1}{1+\alpha}| \cong 1-\alpha & \\ & \text{for} \quad \alpha \ll 1 \quad \& \quad z \to 1. \\ |NTF_{\text{real}}(z)| = |\frac{1+(1-\alpha)z^{-1}}{1+\alpha}| \cong |(1-\alpha)(1-z^{-1})+\alpha z^{-1}| \to \alpha & \end{array} \tag{5.20}$$

An OP AMP with a finite DC gain affects the integration process by two aspects. Based on the comparison between $|STF_{\text{ideal}}(z)|$ and $|STF_{\text{real}}(z)|$, only a $(1-\alpha)$ fraction of the previous output of the integrator is added to the new input sample every time. This phenomenon is usually referred as "*leaky integration*". Based on the comparison between$|NTF_{\text{ideal}}(z)|$ and $|NTF_{\text{real}}(z)|$, quantization noise in the signal band is weakly attenuated. Hence the SNR performance is consequently degraded by the incomplete integration process.

Bandwidth and Slew Rate Besides the finite DC gain of OP AMP, a finite bandwidth or a finite slew rate could also produce an inaccurate charge transfer within each clock cycle, which leads a non-ideal transient response in SC circuits. The effect of the imperfections on bandwidth and slew rate are correlated to each other and can be interpreted as a nonlinear gain [MPVARVH94].

Referring to the SC integrator shown in Fig. 5.12 with a sampling period T_s, the evolution of the output node during the n-th integration period (when Φ_2 is on) is given by

$$V_{\text{out}}(t) = V_{\text{out}}(nT_s - T_s) + (1-\alpha)V_s(1-e^{-\frac{t}{\tau}}), \quad 0 < t < T_s/2, \tag{5.21}$$

where $V_s = V_{\text{in}}(nT_s - T_s/2)$, $(1-\alpha)V_s$ presents the leaky integration due to the finite DC gain. $\tau = 1/(2\pi \cdot GBW)$ is the time constant of the integrator. GBW is the unity-gain bandwidth of the OP AMP when loaded by C_{f}. From Equation 5.21, the maximal slope of the integration curve happens at the beginning of each integration process, i.e. $t = 0$, resulting in

$$\frac{d}{dt}V_{\text{out}}(t)|_{\text{max}} = (1-\alpha)\frac{V_s}{\tau}. \tag{5.22}$$

This can also be explained by that the SC integrator usually starts the integration process with the maximal slew rate. The later integration process can be divided into two separate cases according to the slew rate capability of OP AMP [MBF$^+$03]:

1. If the slew rate of OP AMP is larger than the value in Equation 5.22, there exists no limitation for the integration process. Therefore, the integration process of V_{out} can be described entirely by Equation 5.21.

2. If the slew rate of OP AMP is smaller than the value in Equation 5.22, the output signal of OP AMP cannot linearly follow the changes of the input signal. V_{out} can be characterized in a piecewise function. The first part of the temporal evolution of $V_{\text{out}}(t \leq t_0)$ is linear with the slope *SR*, while the second part of $V_{\text{out}}(t > t_0)$ is non-linear with the slope *SR*.

In summary, the integrator output V_{out} can be characterized as

$$V_{\text{out}}(t) = \begin{cases} \begin{cases} V_{\text{out}}(nT_s - T_s) + SR \cdot t & \text{if} \quad t \leq t_0 \\ V_{\text{out}}(t_0) + ((1-\alpha)V_s - SR \cdot t_0) \times (1 - e^{-\frac{t-t_0}{\tau}}) & \text{if} \quad t > t_0 \end{cases} & \text{if} \quad (1-\alpha)\frac{V_s}{\tau} > SR \\ V_{\text{out}}(nT_s - T_s) + (1-\alpha)V_s(1 - e^{-\frac{t}{\tau}}) & \text{if} \quad (1-\alpha)\frac{V_s}{\tau} \leq SR, \end{cases} \tag{5.23}$$

where t_0 is set by

$$t_0 = \frac{(1-\alpha)V_s}{SR} - \tau \tag{5.24}$$

in order to get the continuity of the derivatives of $V_{\text{out}}(t)$.

Saturation (Output Voltage Range) Another non-ideality of OP AMP is the output saturation, i.e. the limited output voltage range. Clipping will occur when the OP AMP is asked to produce an voltage exceeding its own output voltage range. If this occurs in the integration process, the output signal would fail to follow the ideal output voltage waveform, and is instead a distorted waveform. Therefore, the output range of OP AMP has also to be taken into account.

The corresponding model for a real integrator is shown in Fig. B.4. Fig. 5.13 shows the PSD of the 2^{nd}-order SC $\Sigma\Delta$ modulator with a limited DC gain, a limited slew rate and a limited bandwidth and a limited output voltage range of the OP AMP respectively, when the OSR is set to 256.

In this section, we have made the analysis on how the non-idealities of electrical implementations result in the error mechanisms, which consequently worsen the performance of modulators. Moreover, the degraded SNR performances by these non-idealities are shown in Fig. 5.9, 5.11 and 5.13.

5.4 Example: Hierarchical Optimization of a 2^{nd}-order SC $\Sigma\Delta$ Modulator

The SNR performance determines the eventual resolution of A/D and D/A converters. For SNR measurement, one simulation requires 10,000-100,000 clock samples depending on the oversampling ratio and desired accuracy of the SNR estimation. The standard analog *spice-like* simulations are usually used to generate the raw sample data for the later fast Fourier transform (FFT) or discrete Fourier transform (DFT) postprocessing. If a DFT/FFT is used for calculation of SNR, a $\Sigma\Delta$ modulator with 64*OSR and 1024 in-band FFT bins would require 131,072 clock cycles plus the initial cycle [NST97]. To achieve an accuracy on the order of 90dB, circuit simulators need typically 100-1000 time steps per clock cycle. Thus, over a million time steps are needed just to acquire a single data point on the SNR-versus-input amplitude plot. Such

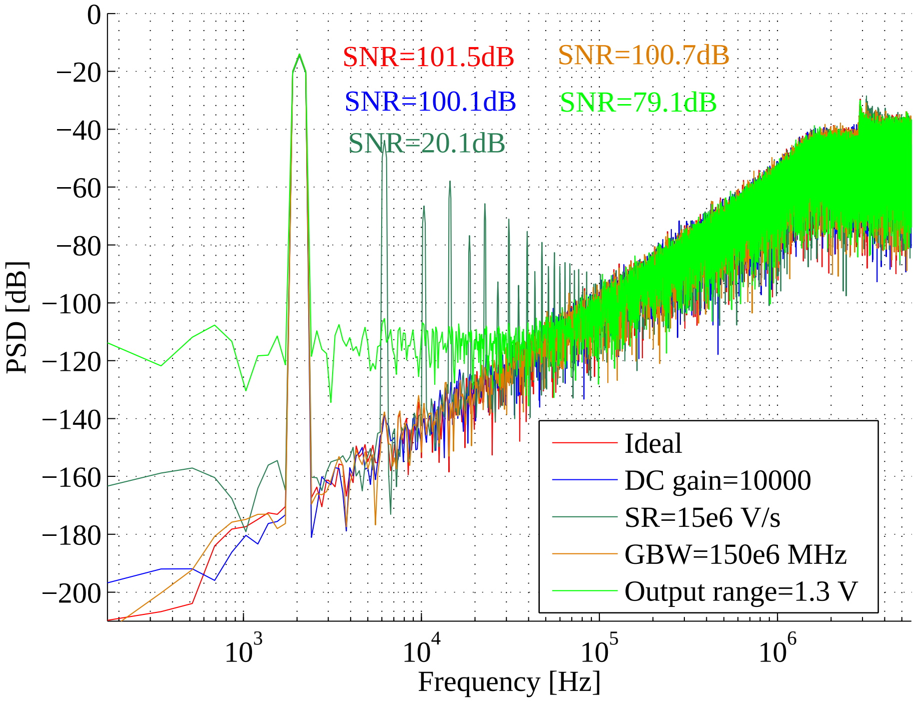


Figure 5.13: Effects of non-identities in OP AMP

a simulation can last a day even running on today's fastest computer machine. Therefore, in order to optimize SNR performance, a hierarchical optimization process is needed for SC ΣΔ modulators.

5.4.1 SC ΣΔ Modulator Hierarchical Modeling

A system-level model in [MBF+03] for SC ΣΔ modulators enables a quick and accurate SNR estimation. Based on the efficient module, a transient simulation only needs several minutes. It makes the "simulation-in-a-loop" based optimization approach feasible for SC ΣΔ modulators.

A system-level module for the 2^{nd}-order SC ΣΔ modulators in Fig. 5.2 is realized in SIMULINK, as shown in Fig. 5.14. Since SC ΣΔ modulators are sensitive to circuit's non-idealities at the input stage where no noise shaping has yet taken place [BW88], only the non-idealities of the first integrator are considered here, while the second integrator, comparator and 1-bit DAC are assumed as ideal. The detailed realization of each sub-model is described in Appendix B. In this model, the following non-idealities of building blocks are considered:

1. clock jitter at the input sampler: δ;
2. switch thermal noise in the SC structure: KT/C_s;
3. operational amplifier noise: $RN(t)$;
4. operational amplifier finite DC gain: A;
5. operational amplifier gain bandwidth: GBW;

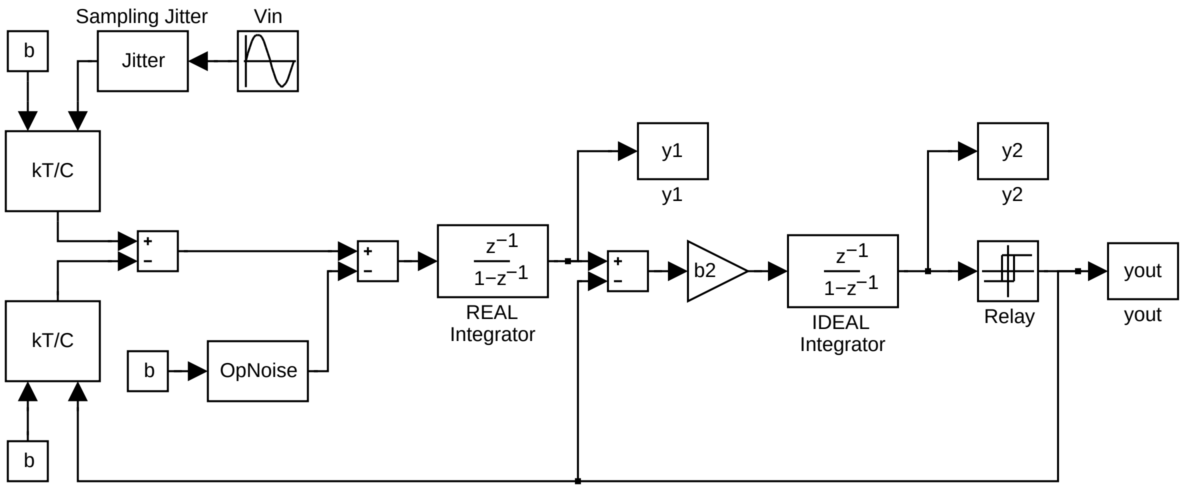


Figure 5.14: A second-order SC ΣΔ modulator model in Simulink

6. operational amplifier slew rate: *SR*;
7. operational amplifier output voltage: V_{out};

Item 1 clock jitter and item 3 operational amplifier noise are random noises, which won't be considered in this work. Since K is the Boltzmann's constant and T is the absolute temperature, item 2 switch thermal noise KT/C_s is fixed for a predetermined capacitor. The SNR performance of a given topology SC ΣΔ modulator can be optimized by properly choosing the design parameters of OP AMP, i.e. items 4-7 A, GBW, SR and V_{out}. Other system information is listed in Tab. 5.2.

Table 5.2: Parameters of the 2$^{\text{nd}}$-order SC ΣΔ modulator

Parameter	Value
Signal bandwidth	BW = 22.05Hz
Sampling frequency	f_s = 11.29MHz
Oversampling ratio	R = 256
Number of samples considered	N = 66536
Integrator coefficients	$b = b2 = 0.5$

In the hierarchical modeling of the SC ΣΔ modulator, the SNR is the sole system performance to be optimized, i.e.

$$f = \text{SNR}. \tag{5.25}$$

The SNR performance is dependent of the system-level parameters **p** of OP AMP, i.e.

$$\mathbf{p} = [A, GBW, SR, V_{\text{out}}]. \tag{5.26}$$

These system-level parameters are determined by the circuit-level design parameters **x**, i.e.

$$\mathbf{x} = [W_1, L_1, W_2, L_2, \ldots]. \tag{5.27}$$

where W_i and L_i are the width and length of the *i-th* transistor shown in Fig. 5.16. The hierarchical modeling of the second-order SC ΣΔ modulator is presented in Fig. 5.15.

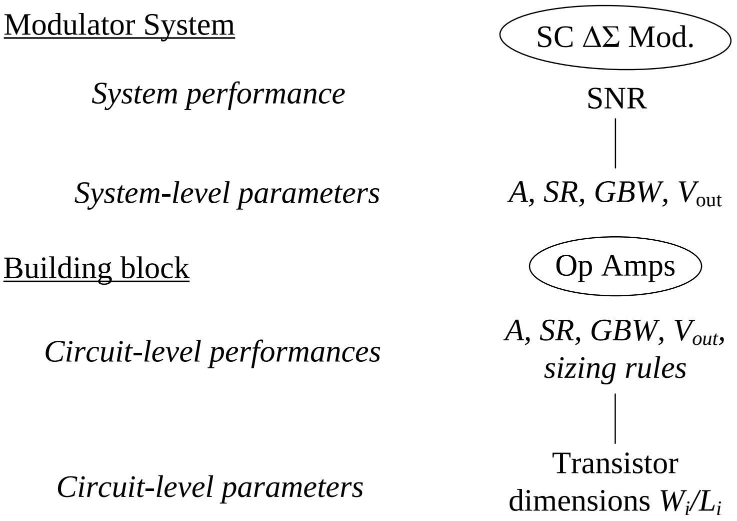

Modulator System
SC ΔΣ Mod.
System performance
SNR
System-level parameters
A, SR, GBW, V_{out}
Building block
Op Amps
Circuit-level performances
A, SR, GBW, V_{out}, sizing rules
Circuit-level parameters
Transistor dimensions W_i/L_i

Figure 5.15: Hierarchical performance modeling of the SC ΣΔ modulator

5.4.2 PSE on OP AMP

A folded-cascode OP AMP is used for the second-order SC ΣΔ modulator, which is realized in 180nm technology with a supply voltage of 3V. The schematic is shown in Fig. 5.16.

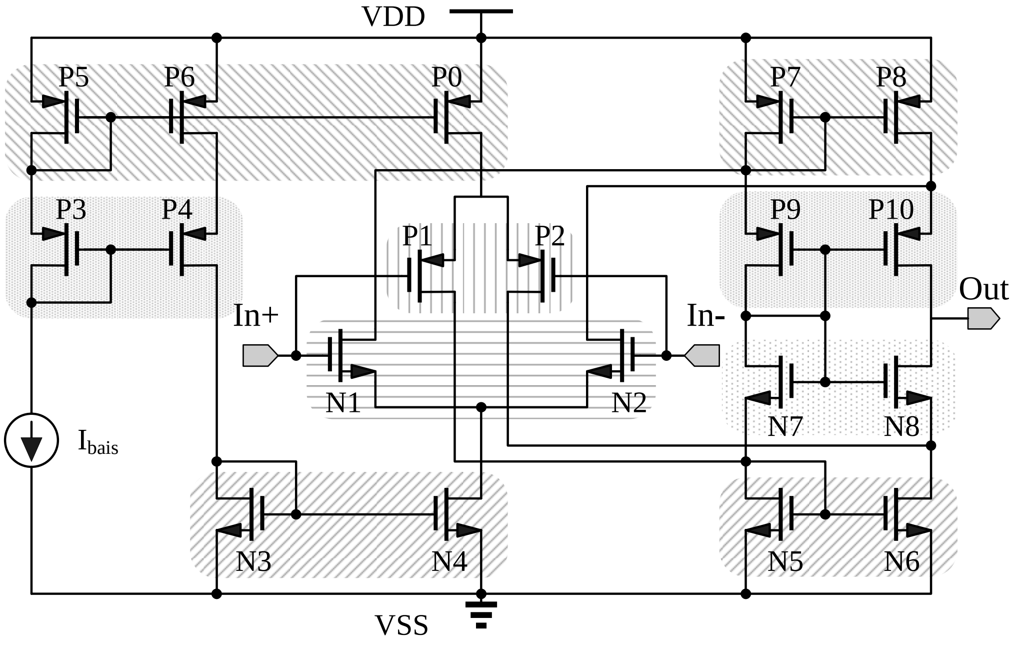


Figure 5.16: Schematic of a fold cascode OP AMP

For PSE on the OP AMP, sizing rules [GZEA01], i.e. geometric and electric constraints, are considered to guarantee all transistors to work properly in the expected analog function. Tab. 5.3 summarizes the total 152 sizing rules for the N/PMOS in the folded-cascode OP AMP. The detailed sizing rules for each basic analog structure can be found in Tab. A.2. In addition to the sizing rules, two performance constraints are taken into account: a) phase margin (*PM*) larger than 45° for the stability of OP AMP; b) the output voltage region V_{out} larger than 2.5V, so that the 2nd-order SC ΣΔ modulator will work properly and the SNR performance will not be degraded by the limited output voltage range.

Table 5.3: Sizing rules for the folded-cascode OP AMP

N/PMOS	Analog function	Number of sizing rules
N1 & N2	N-type differential pair	2*7
N3 & N4	N-type simple current mirror	2*7
N5 & N6	N-type simple current mirror	2*7
N7 & N8	N-type lever shift	2*9
P1 & P2	P-type differential pair	2*7
P3 & P4	P-type lever shift	2*9
P5 & P6	P-type simple current mirror	2*7
P0 & P6	P-type simple current mirror	2*7
P7 & P8	P-type simple current mirror	2*7
P9 & P10	P-type lever shift	2*9
		152 (sum)

Large DC gain (A), large unity-gain bandwidth (GBW) and large slew rate (SR) are desired in the OP AMP circuit. Through PSE method, e.g. NBI in [SGA03], a 3D nominal Pareto-optimal front of A, GBW and SR, i.e. PF, is obtained, as shown in Fig. 5.17(a). From the distribution of these efficient points on each 2D surface, a design trade-off between A and GBW, denoted as $PF1$ in Fig. 5.17(b), and a trade-off between A and SR, denoted as $PF2$ in Fig. 5.17(c) are clearly seen. The increasing trend of SR is proportional to GBW, denoted as the solid line with arrow in Fig. 5.17(d). However, there still exists a Pareto-optimal front between SR and GBW, denoted as $PF3$, when both performances approach their limits. The maximum A, point x, the maximum GBW, point y, and the maximum SR, point z, are denoted as pentagrams in Fig. 5.17. And the achieved corresponding maxima and minima of the three performances are listed in Tab. 5.4.

Table 5.4: Maxima and minima of A, GBW and SR

	maximum	minimum
A (dB)	105.23	76.54
GBW (MHz)	47.05	1.831
SR (V/μs)	33.01	1.44

5.4.3 Nominal Optimization of SC ΣΔ Modulator

The ideal SNR value of the modulator is 101.5dB. Practically, SNR will be degraded by the finite A, GBW and SR of the OP AMP. To compute the maximal SNR performance, a hierarchical optimization process based on the capability of the OP AMP can be formulated as

$$\max_{A,GBW,SR} \mathrm{SNR}(A,GBW,SR) \quad s.t. \quad c(A,GBW,SR) = PF, \tag{5.28}$$

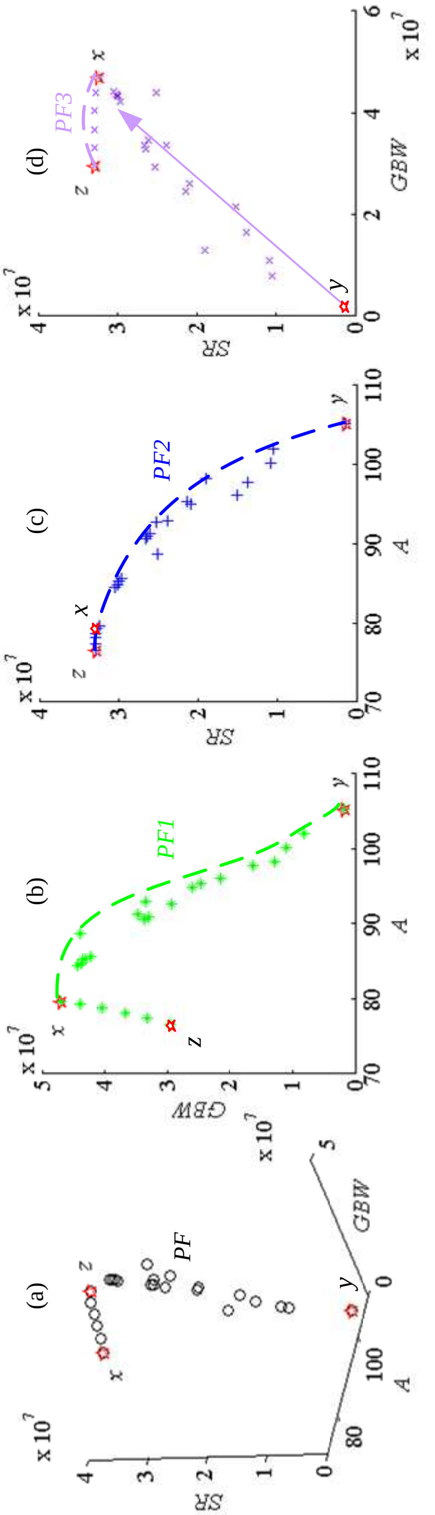


Figure 5.17: Pareto-optimal front of OP AMP (a)3D dimension; Projection on 2D surface (b) *A* vs. *GBW*; (c) *A* vs. *SR*; (d) *SR* vs. *GBW*

where $c(*)$ describes the feasible ranges of the system-level design parameters: A, GBW and SR, as shown in Fig. 5.17(a). However, the trade-off between GBW and SR is not obvious, which results in a narrow 3D Pareto-optimal front as shown in Fig. 5.17(a). It is difficult to fitting the PF. Moreover, an inaccurate fitting could result in a false design result. Hence, we try to reduce the number of the system-level design parameters first. Six system-level sweep-analysis processes of SNR are executed at design points x,y,z, respectively. For example a sweep-analysis at point x, (i.e. maximal GBW), one of the other two parameters (A and SR), varies from their own minimum to maximum individually and the other parameters keep constant at the corresponding design values.

At point x, the red lines in Fig. 5.18 represent the variation of SNR related to A and SR respectively. As can be seen, SNR increases with SR, while SNR stays almost constant with A. At point y, the blue lines represent the variation of the SNR related to SR and GBW respectively. Worth to note, SNR keeps almost 0 with GBW, because SR is minimal at the design point. At point z, the green lines represent the variation of the SNR related to SR and A respectively. Based on these results, we can find that A doesn't have any impact on SNR in this experiment, while SNR is very sensitive to the value of SR. Additionally, a larger GBW can enlarge SNR when SR is maximal.

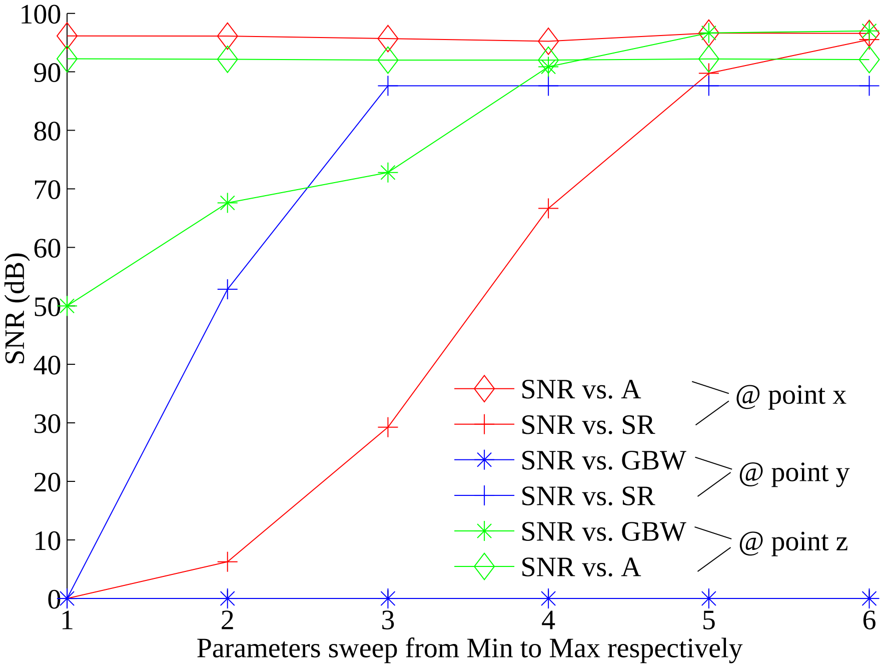


Figure 5.18: SNR vs Parameter Sweep

Based on the analysis above, SNR is mostly dominated by SR. And the main increasing trend of SR is proportional to GBW. Therefore, the maximal SNR is achieved by a design point, which lies on the nominal Pareto-optimal front $PF3$, zoomed in Fig. 5.19. The optimization of SNR in Equation 5.28 can be simplified to

$$\max_{GBW,SR} \text{SNR}(GBW,SR) \quad s.t. \quad c(GBW,SR) = PF3, \tag{5.29}$$

where the system-level design parameters are reduced to *GBW* and *SR*, *PF*3 is the optimization constraint for system-level optimization. The maximal SNR is 98.89dB generated by the design point q_1, which is listed in Tab. 5.6 and denoted as diamond in Fig. 5.19.

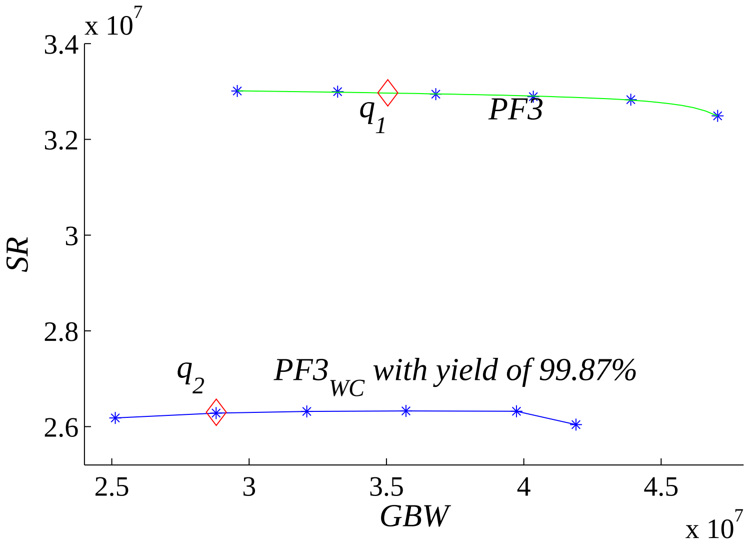


Figure 5.19: Pareto-optimal front of GBW vs. SR

5.4.4 Worst-Case Analysis of SC ΣΔ Modulator

What is the SNR performance with the nominal optimized device sizes actually after fabrication? Through a Monte Carlo analysis on the statistical parameters of the OP AMP, about 52% circuits can generate the q_1 performance values, i.e. *GBW*=35.05MHz and *SR*=32.97V/*µs*. Additionally, two operation parameters are considered here, the supply voltage, which varies from 2.9V to 3.1V, and the temperature, which varies from 0°*C* to 100°*C*. The worst-case operation conditions for *SR* and *GBW* are listed in Tab. 5.5. Consequently, only 0% circuits can achieve the q_1 performance values in terms of both variation of statistical and operation parameters. Therefore, the SNR of 98.89dB, which is generated by design point q_1, is not an actual capability of the circuit after fabrication.

Table 5.5: Worst-case operation conditions for *GBW* and *SR*

	Temperature	Supply voltage
GBW	100°C	2.9V
SR	0°C	2.9V

To get the real maximum of SNR after fabrication, a yield-aware optimization is needed. According to varied target yields, different worst-case-aware Pareto-optimal fronts can be extracted. Fig. 5.19 shows a front $PF3_{\text{WC}}$ with a target yield of 99.87% when the worst-case distance β_w is set to 3, according to Tab. 3.1. The optimization of SNR with the 99.87% yield can be formulated as

$$\max_{GBW,SR} \text{SNR}(GBW,SR) \quad s.t. \quad c(GBW,SR) = PF3_{\text{WC}}. \tag{5.30}$$

The achieved maximal SNR is 89.4dB, the corresponding design point q_2 is listed Tab. 5.6 and denoted as diamond in Fig. 5.19. The maximal value of SNR is obviously degraded by the statistical and operation parameters. Finally, the robust design results that we are searching for are the device sizes which generate q_2, not q_1.

Table 5.6: Nominal and robust optimization results of the Modulator

	SNR (dB)	Yield	GBW (MHz)	SR (V/us)
Ideal	101.5		infinite	infinite
Nominal q_1	98.8	0%	35.05	32.97
Robust q_2	89.5	99.87%	28.32	26.3

5.5 Summary

This chapter describes the basic advantages by using ΣΔ modulators compared to Nyquist-frequency modulators. Noise averaging and noise shaping is simultaneously implemented in the ΣΔ modulators. The efficiency of the 2^{nd}-order SC ΣΔ modulator is discussed compared to other modulator topologies.

Based on the linear z-domain model of the 2^{nd}-order SC ΣΔ modulator topology, the maximal theory value of SNR can be evaluated. Due to the non-idealities of the analog building blocks, the actual maximum of SNR will be less than the theoretical value. Most non-idealities of the 2^{nd}-order SC ΣΔ modulator are listed here and their effects on the SNR performance are presented individually. And the corresponding models in SIMULINK are built including these non-idealities.

Since the simulation of a SC ΣΔ modulator on circuit level is very time-consuming, the proposed hierarchical optimization methodology in Chapter 3 is applied on the modulator. The SNR performance is maximized based on the real capability of a fold-cascode OP AMP circuit. In the optimization process, a performance space exploration method is firstly applied to the OP AMP so that a nominal Pareto-optimal front is extracted, and then the worst-case analysis is conducted on each efficient point of the nominal Pareto-optimal front, in order to obtain a worst-case-aware Pareto-optimal front. Based on that, the practical maximum of SNR with a target yield has been computed. It gives designers a real insight into the capability of the circuit after fabrication. The presented method is not limited to the 2^{nd}-order SC ΣΔ modulator but applicable to other modulators as well.

Chapter 6

Conclusion

The Semiconductor Industry Association (SIA) has announced that worldwide semiconductor sales reached totaling \$267.5 Billion in 2007, a new industry record. Analog circuits are one of the fastest-growing segments of the market. In the next five years, nearly 70 percent of all ICs will have analog signal components, which are widely used in communication, computer, consumer, automotive and industrial applications. The gradually shrinking process technologies make analog circuits smaller, faster and more power efficient, but introduce more challenges and difficulties as well. The increasing time-to-market pressure drives the progress of design methodologies, CAD tools and design flows. In contrast to the significant increasing density and complexity of analog circuits, design methodologies on analog circuits have achieved a little progress over the past decades. Compared to the well developed and widely available digital CAD tools, few robust commercial CAD tools are available except for *spice*-like numerical simulators. Consequently, the design of analog and mixed-signal ICs remains still a long and error prone process, which tends to be a bottleneck point in design process of SoC systems. Therefore, an efficient design methodology for large-scale analog/mixed-signal circuit design will be valuable.

Chapter 1 presented the challenges of analog design. Compared to digital design, designers have to be much more involved in the design process of an analog design than that of a digital design, and consequently the design results are highly dependent on the knowledge and experiences of individual designers. The current analog design flow starts with the manual selection of circuit topology. Then designers assign the transistor sizes and the values for resistors and capacitors, and simulate the circuit. The tuning process on design parameters and the circuit verification have to been repeated until the desired circuit performances are achieved. Some commercial automatic sizing tools have been mentioned, which are capable of accelerating some analog designs. As analog circuits become larger, more complex and include digital parts gradually, two different optimization strategies for large-scale analog/mixed-signal circuits have been introduced: i.e. flat and hierarchical optimization methodology. Additionally, the methods for performance space exploration have been also summarized, which provide the useful information of the lower-level realization for the design on a higher-level in hierarchy.

Chapter 2 explained the two automatic processes for analog design, i.e. automatic sizing and performance space exploration. The fundamental concepts of analog design, e.g. design parameter, circuit performance/specification/yield, circuit simulation, have been introduced. Nominal design aims to optimize the often conflicting circuit performances at the same time, while robust design aims to maximize the circuit yield. The automatic sizing process is a mapping from circuit specifications to design parameters, and is usually referred to as a nominal design process. The additional sizing rules for CMOS transistors intend to reduce the degree of freedom in analog design, so that the automatically sized results are guaranteed to stay in the technically meaningful regions. Performance space exploration is a mapping from the feasible space of the design parameters to the feasible space of the circuit performances. A deterministic and simulation-based PSE method, i.e. Normal-Boundary Intersection, has been introduced to generate a Pareto-optimal front (a part boundary of the entire feasible performance region), which represents the performance capabilities of the circuit.

Chapter 3 proposed a hierarchical optimization methodology for large-scale analog/mixed-signal circuits. The methodology is a ***"simulation-in-a-loop"-based hierarchical optimization*** methodology, which consists of four main steps. (1) Performance space exploration is applied to each building block on the circuit level individually. Their respective Pareto-optimal fronts can be obtained. (2) Efficient behavioral models are built in HDLs or SIMULINK. The models include not only the description of circuit functionalities, but the description of Pareto-optimal fronts as well. (3) Based on these behavioral models, an automatic sizing process is conducted on the system level. During the optimization process, the system-level parameters are restricted to these Pareto-optimal fronts. (4) The specifications for each building block are propagated from the optimized results on the system level. And then, the automatic sizing process on each building block is conducted individually and in parallel. The whole hierarchical optimization process can be characterized by a bottom-up extraction of circuit capability and a top-down hierarchical sizing process. Additionally, through worst-case analysis on the efficient points of the nominal Pareto-optimal front, a worst-case-aware Pareto-optimal front can be extracted. Based on that, the obtained optimization results represent the actual circuit capability with a target yield after fabrication.

In this thesis, the proposed hierarchical optimization methodology has been applied to two typical large-scale analog/mixed-signal circuits: a charge-pump phase-locked loop (CPPLL) and a switched-capacitor sigma-delta (SC ΣΔ) modulator. In Chapter 4, the fundamental of a CPPLL has been discussed, including PLL's building blocks, PLL's performances and PLL system's analysis methods. The complex trade-offs in PLLs show the difficulties and the challenges of the optimization task by manual design. The time-consuming simulation of PLL's performance is the main obstacle to the “simulation-in-a-loop”-based optimization method. To tackle the problem, efficient behavioral models in Verilog-A have been developed. Based on performance space exploration on the circuit level, a first-time-successful top-down sizing process without iteration has been realized. The obtained Pareto-optimal fronts of building blocks and system represent the capability of circuit and visualize the conflicting relationship among performances, which give designers a detailed insight into the circuit. In Chapter 5, a second-order SC ΔΣ modulator has been taken as example circuit because of its efficiency. Based on the linear z-domain model, the theoretic maximum SNR performance can be evaluated. However due to the non-idealities of the analog building blocks, the actual SNR performance cannot achieve the theoretical value. By applying performance space exploration on building blocks, the SNR performance has been maximized while considering the capabilities of OP AMP. Moreover, worst-case analysis has been applied to the nominal efficient points in order to extract a worst-case-aware Pareto-optimal front. Based on it, the actual maximum of the SNR with a target yield has been extracted. The final optimized result show the eventual capability of the circuit after fabrication. The “simulation-in-a-loop”-based optimization method has been realized based on the efficient behavioral modeling in SIMULINK.

With the proposed hierarchical approach, the complete CPPLL and the SC ΣΔ modulator have been sized algorithmically despite their design complexity. Moreover, the whole design process can be accomplished in a reasonable time cost, which is obviously shorter than the design period by using the traditional analog design method. The presented approach is not limited to the two kinds of circuits but applicable to other large-scale analog/mixed-signal circuits as well.

Appendix A

Analog Sizing Rules

For analog circuits using CMOS technology, a lot of fundamental analog building blocks are identified based on [GZEA01]. There are five levels of hierarchy as depicted in Tab. A.1 [Ste05]. *

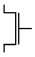
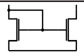
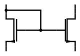
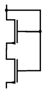
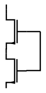
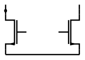
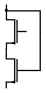
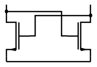
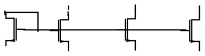
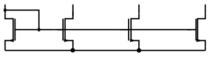
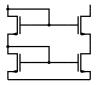
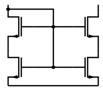
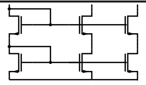
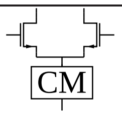
- At the lowest hierarchy level 0, the atomic building block is a single transistor. A transistor can act as a voltage-controlled current source(VCCS) in its saturation operation range or a voltage-controlled resistor in its linear operation range.
- At the hierarchy level 1, seven transistor pairs are defined to present basic analog functionalities. For example, a simple current mirror is used to copy a current from one path to another path. A level shift is used to shift a voltage to a higher or a lower voltage level.
- At the hierarchy level 2, four “pairs of transistor pairs” are defined to present more complex or more accurate analog function. These pairs are consisted of the structures from level 0 and level 1. For example, a 4-transistor can be modeled as a combination of a voltage reference and a current mirror load.
- At the hierarchy level 3, a cascode current mirror bank is modeled as a level shifter bank and a current mirror bank.
- At the hierarchy level 4, a differential stage is modeled as a combination of a differential pair and a generic current source. The current source can be implemented as a simple current mirror or a cascode mirror structure.

As can be seen, the block at a certain hierarchy level is composed of blocks at lower levels. Moreover, the structure library is not complete and a variety of other building blocks can be added.

A set of sizing rules will be given for each building block. Theses sizing constraints guarantee the dedicated analog function and strengthen its robustness, e.g. reduce mismatch effect or channel length modulation. These constraints refer to not only transistor geometry parameters (width, length and area) and electrical transistor quantities (e.g. transistor drain/source voltages) as well. Since each block on level *i* consists of the building blocks at the lower levels in hierarchy, therefore the sizing rules for each identified block at level *i* include all sizing rules

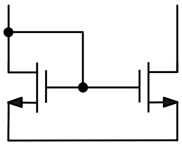
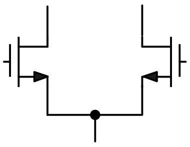
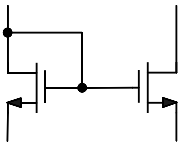
* In this table, fundamental analog building blocks are presented schematically through NMOS transistors and analogously for PMOS counterparts.

Table A.1: Library of analog basic building blocks [Ste05]

Function	Schematic (NMOS) (PMOS analogously)	Hierarchy Level
Voltage-controlled resistor (res) Voltage-controlled current source (cs)		0
Simple current mirror (cm)		1
Level shifter (ls)		
Voltage reference 1 (vr1)		
Current mirror load (cml)		
Differential pair (dp)		
Voltage reference 2 (vr2)		
Flip-flop (ff)		
Level shifter bank (LSB)		2
Current mirror bank (CMB)		
Cascode current mirror (CCM)		
4-Transistor current mirror (4TCM)		
Cascode current mirror bank (CCMB)		3
Differential stage (DST) (CM ∈ { cm, CCM, 4TCM, CCMB })		4

of these fundamental blocks and the additional sizing rules for the block itself. Tab. A.2 [mun] lists the detailed sizing rules for the recognized basic NMOS structures: simple current mirror, differential pair and level shifter. The safety margin m is set according to the CMOS realization technology.

Table A.2: Sizing rules for NMOS basic structures [mun]

Structure	Constraint	Safety margin	Reason
NMOS Current Mirror 	$V_{GS} - V_{th} \geq m$	$m = 100mV$	inversion
	$V_{DS} - (V_{GS} - V_{th}) \geq m$	$m = 100mV$	saturation
	$L \cdot W \geq m$	$m = 1\mu m^2$	limit V_{th} mismatch
	$L \geq m$	$m = 0.5\mu m$	limit relative variance of the transconductance factor
	$W \geq m$	$m = 0.5\mu m$	
	$-m \leq V_{DS1} - V_{DS2} \leq m$	$m = 200mV$	reduce the influence of the channel length modulation factor on the current transmission coefficient
	equal length		limit systematic mismatches
NMOS Differential Pair 	$V_{GS} - V_{th} \geq m$	$m = 10mV$	inversion
	$V_{DS} - (V_{GS} - V_{th}) \geq m$	$m = 10mV$	saturation
	$L \cdot W \geq m$	$m = 1\mu m^2$	limit V_{th} mismatch
	$L \geq m$	$m = 0.5\mu m$	limit relative variance of the transconductance factor
	$W \geq m$	$m = 0.5\mu m$	
	$V_{GS} - V_{th} \leq m$	$m = 1.0V$	reduce the influence of transconductance mismatch on the input offset
	$-m \leq V_{DS1} - V_{DS2} \leq m$	$m = 200mV$	reduce the influence of the channel length modulation factor on the current transmission coefficient
	equal length		avoid transconductance mismatch and input offset voltage mismatch
	equal width		
NMOS Level Shifter 	$V_{GS} - V_{th} \geq m$	$m = 10mV$	inversion
	$V_{DS} - (V_{GS} - V_{th}) \geq m$	$m = 10mV$	saturation
	$L \cdot W \geq m$	$m = 1\mu m^2$	limit V_{th} mismatch
	$L \geq m$	$m = 0.5\mu m$	limit relative variance of the transconductance factor
	$W \geq m$	$m = 0.5\mu m$	
	$-m \leq i_{DS2}/i_{DS2} - W_2/W_1 \leq m$	$m = 0.2$	avoid a difference between the effective voltage V_{GS}
	equal length		avoid transconductance mismatch

Appendix B

Modeling ΣΔ Modulator Non-Idealities in SIMULINK

In this chapter, the behavioral model of the second-order ΣΔ modulators is realized in Simulink module, which are based on the modules in [BFM$^+$99, MBF$^+$03].

Sampling Jitter According to Equation 5.11, a random sampling jitter model can be built as shown in Fig. B.1. The input signal $x(t)$ and its derivative du/dt are continuous-time signals. They are sampled with a sampling period T_s by a zero-order hold. The sampling uncertainty, i.e. clock jitter δ, is implemented by a Gaussian random process $n(t)$ with a standard deviation $\Delta\tau$.

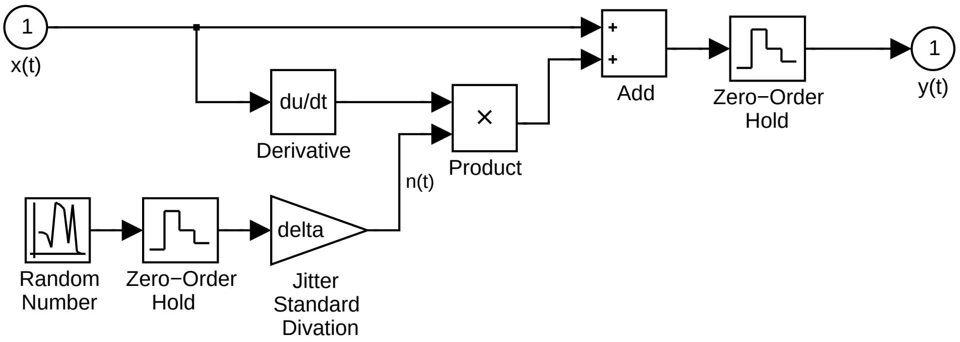


Figure B.1: Modeling a random sampling jitter [MBF$^+$03]

Switches Thermal Noise The switch thermal noise voltage e_T can be evaluated by Equation 5.12. And the thermal noise is superimposed to the input voltage $x(t)$, expressed as

$$y(t) = [x(t) + e_T(t)]b = [x(t) + \sqrt{\frac{kT}{C_\mathrm{s}}}n(t)]b, \tag{B.1}$$

where $n(t)$ denote a Gaussian random process with unity standard deviation, and $b = C_\mathrm{s}/C_\mathrm{f}$ is the coefficient of the integrator. Equation B.1 is implemented by the model shown in Fig. B.2.

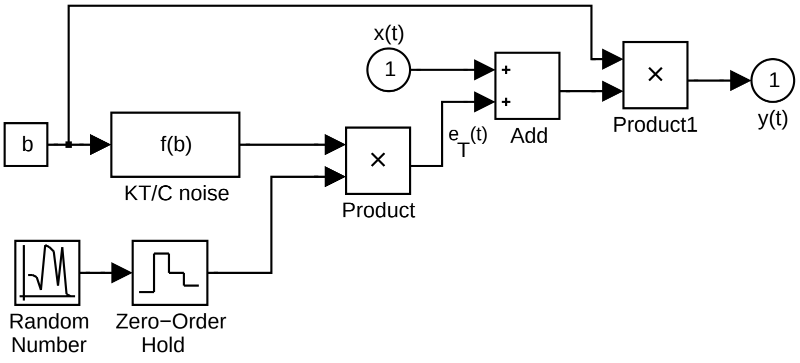


Figure B.2: Modeling switches thermal noise KT/C block [MBF+03]

Operation Amplifier Noise The model of OP AMP noise is shown in Fig. B.3. The intrinsic noise of operational amplifier includes thermal noise, flicker (1/f) noise, shot noise etc. These various noises contribute together to the total OP AMP noise power V_n^2, whose value can be evaluated through simulation on the circuit in Fig. 5.3 during phase $\phi 2$. "The resulting output referred noise PSD has to be integrated over the whole frequency spectrum, eventually taking into account the degradation of the thermal noise PSD introduced by the auto-zero or correlated double sampling techniques" [ET96], and then divided by b^2 in order to calculate the corresponding input-referred rms noise voltage V_n.

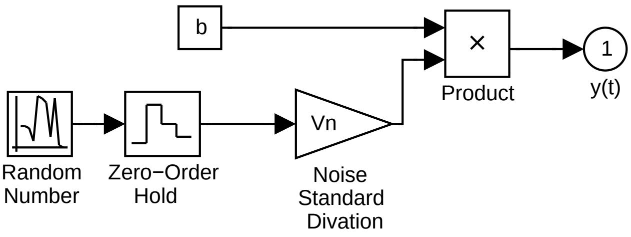


Figure B.3: Modeling operational amplifier noise [MBF+03]

Real Integrator Fig. B.4 shows the model of a real integrator including all the non-idealities listed in Sec. 5.3.1.3. Only a fraction $(1-\alpha)$ of the previous output of the integrator is added (fed back) to each new input sample, which is modeled the parameter 'alpha' in Fig. B.4, where

$$\text{alpfa} = 1-\alpha = \frac{A_0-1}{A_0}. \tag{B.2}$$

The finite bandwidth and slew rate of the OP AMP are implemented by a MATLAB function block which is placed in front of the integrator. According to Equation 5.23, the corresponding description in C-code is shown as follows. The limited output range of the OP AMP can be simply realized by using a saturation block to limit the final output range.

```
function out = slew(in,alpha,sr,GBW,Ts)
% Models the operational amplifier finite bandwidth and slew rate
% for a discrete time integrator
% out = slew(in,alpha,sr,GBW,Ts)
% in:       Input signal amplitude
% alpfa:     Effect of finite gain (ideal amplifier alpha=1)
% sr:       Slew rate in V/s
% GBW:      Gain-bandwidth product of the integrator loop gain in Hz
% Ts:       Sample time in s
% out:      Output signal amplitude

tau=1/(2*pi*GBW);                   % Time constant of the integrator
Tmax = Ts/2;
slope=alpha*abs(in)/tau;

if slope > sr                       % Op-amp in slewing
    tsl = abs(in)*alpha/sr - tau;    % Slewing time
    if tsl >= Tmax
        error = abs(in) - sr*Tmax;
    else
        texp = Tmax - tsl;
        error = abs(in)*(1-alpha) + (alpha*abs(in) - sr*tsl) * exp(-texp/tau);
    end
else                                % Op-amp in linear region
    texp = Tmax;
    error = abs(in)*(1-alpha) + alpha*abs(in) * exp(-texp/tau);
end

out = in - sign(in)*error;
```

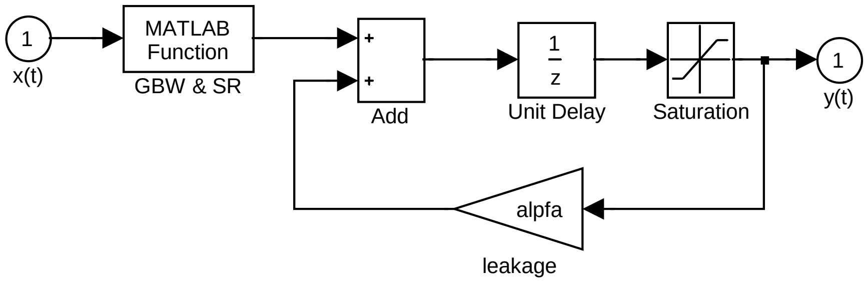


Figure B.4: Modeling real integrator [MBF$^+$03]

Appendix C

Phase Noise & Jitter

C.1 Relationship between Phase Noise and Jitter

Analog engineers prefer using phase noise, while digital designers prefer using jitter. The relationship between the two parameters is briefly described here. More details can be found in [Raz96, DMR00, HL98, HLL99, LH00, Meh02].

Timing jitter is defined as the standard deviation of the time uncertainty [HLL99]:

$$\sigma^2_{\Delta T} = \frac{1}{\omega_0^2} \cdot E\{[\phi(t+\Delta T) - \phi(t)]^2\} = \frac{1}{\omega_0^2} \cdot \{E[\phi^2(t+\Delta T) + \phi^2(t) - 2\phi(t+\Delta T) \cdot \phi(t)]\}, \quad \text{(C.1)}$$

where $E[\cdot]$ represents the expected value, the time uncertainty is $\phi(t+\Delta T) - \phi(t)$. Since $E[\phi(t) \cdot \phi(t+\Delta T)]$ is equal to the autocorrelation of $\phi(t)$, i.e. $R_\phi(\Delta T)$. Therefore, the jitter in Equation C.1 can be written as

$$\sigma^2_{\Delta T} = \frac{2}{\omega_0^2}[R_\phi(0) - R_\phi(\Delta T)]. \quad \text{(C.2)}$$

The relationship between the autocorrelation and the power spectrum is given by the Khinchin theorem [Gar90], i.e.

$$R_\phi(\tau) = \int_0^\infty S_\phi(f) e^{j2\pi f\tau} df, \quad \text{(C.3)}$$

where $S_\phi(f)$ represents the power spectrum of $\phi(t)$. Replacing the autocorrelation of Equation C.3 into Equation C.2 result in

$$\sigma^2_{\Delta T} = \frac{8}{w_0^2} \int_0^\infty S_\phi(f) \sin^2(\pi f \Delta T) df. \quad \text{(C.4)}$$

Equation C.4 describes the relationship between timing jitter and noise power spectral density (psd). Therefore, the timing jitter can be calculated from the phase noise by using Equation C.4. As ΔT goes to infinity, timing jitter is calculated from Equation C.2:

$$\sigma^2_{\Delta T\to\infty} = \frac{2}{\omega_0^2}[R_\phi(0)] \quad \text{(C.5)}$$

$$\sigma^2_{\Delta T\to\infty} = \frac{2}{\omega_0^2} \int_0^\infty S_\phi df. \quad \text{(C.6)}$$

As can be seen from the above analysis, since time jitter has less information than phase-noise spectrum, the inverse process (from jitter to phase noise) is normally not feasible, unless the extra information on the shape of phase-noise spectrum is available.

C.2 Extracting Jitter from Phase Noise Analysis on PFD/CP and VCO Blocks

Simulator SpectreRF [Cadb] is used in this thesis to compute the phase noise of the CPPLL. The proper frequency range in periodic-steady-state (PSS) analysis and PNoise analysis is chosen so that the noise out of the frequency range is neglected*. In PSS and PNoise analysis, it linearizes the circuit at each time step in a given period and accumulates the contributions from every noise sources and over each time point to compute the total phase noise.

The jitter of PFD/CP can be derived from the following steps. First of all, according to the edge-to-edge jitter definition, i.e.

$$var(\delta t_n) = E[(\delta t_n - \overline{\delta t_n})^2], \tag{C.7}$$

where $\overline{\delta t_n}$ is the mean value of δt_n. The value is equal to 0 for a white noise. Therefore, the $var(\delta t_n)$ can be reformulated as

$$var(\delta t_n) = E[(\delta t_n)^2] = R_{\delta t_n}(0). \tag{C.8}$$

Applying the Wiener-Khinchin Theorem in Equation C.3 to determine

$$var(\delta t_n) = \int_0^{\infty} S_n(f)df, \tag{C.9}$$

where $S_n(f)$ is the power spectral density of the δt_n sequence.

To reduce the simulation time on jitter extraction, the noise source of PFD/CP is moved forward at its input, which is called input-referred jitter of PFD/CP. The corresponding input-referred jitter can be calculated by dividing the effective gain of the PFD/CP:

$$J_{\mathrm{ee}_{\mathrm{PFD/CP}}} = \frac{T}{K_{\mathrm{det}}}\sqrt{\frac{var(\delta n)}{2}}, \tag{C.10}$$

where K_{det} is the gain of the PFD/CP, in units of ampleres/cycle, T is in units of seconds/cycle. The ratio 2 comes from the modeling on two transition edges in a cycle. †

The jitter in VCO, is almost completely due to oscillator phase noise. According to the Equation 76 in [Kun05], i.e.

$$c = L(\Delta f)\frac{\Delta f^2}{f_0^2}, \tag{C.11}$$

* The noise should be at least -40dB down and dropping at the highest frequency simulated.

† In a cycle with two transition, the sum jitter is calculated by $J_{\mathrm{sum}} = \sqrt{J_1^2 + J_2^2} = \sqrt{2}J$, where $J_1 = J_2 = J$

and Equation 73 in [Kun05], i.e.

$$J = \sqrt{cT} = \sqrt{\frac{c}{f_0}}, \tag{C.12}$$

the jitter of VCO can be extracted by

$$J_{\mathrm{VCO}} = \frac{\Delta f}{f^{1.5}} 10^{\frac{L(\Delta f)}{20}}. \tag{C.13}$$

$L(\Delta f)$ means the phase noise on the offset frequency Δf in unit of dBc/Hz. Note the Equation C.13 is only valid when the phase noise at Δf, which is in the $1/f^2$ region.

Appendix D

CPPLL’s Verilog-A Models

In this chapter, the Verilog-A models for building blocks of the CPPLL are listed in the following, which are based on the modules in [Kun05].

Lists D.1- D.4 are for the extraction of the locking time performance.

Listing D.1: Behavioral model of PFD in Verilog A

```
//PFD

‘include "constants.h"
‘include "discipline.h"

module PFD (ref, feedback, u, ub ,db ,d);
   input ref, feedback;   //input and feedback are the reference and feedback clock
   output u, ub, db,d;
   electrical ref, feedback, u, ub, db, d;

   parameter real v_high=3;
   parameter real v_low=0;
   parameter real ttol=10p;
   parameter real tt=120p from (0:1000000);       //rise time and fall time
   // for "u" and "ub" signal: delay time extracted from circuit level simulation
   // ———————— rise dealy —————————— fall delay
   parameter real td_u1=37p;  parameter real td_u2=190p;
   // for "d" and "db" signal:
   parameter real td_d1=45p;  parameter real td_d2=136p;

   integer state;                          //state=-1 for down, state=1 for up.
   real td_u,td_ub,td_d,td_db;       //delay time

analog begin
   @(cross(V(ref)-v_high/2.0, 1, ttol)) begin
        if (state<1)
               state = state+1;
                   //$strobe("current time %g ns and state=", $realtime*10e8, state);
                   if (V(u) >v_high/2)  td_u =td_u1;    else td_u =td_u2;
                   if (V(ub)>v_high/2)  td_ub=td_u1;    else td_ub=td_u2;
                   if (V(d) >v_high/2)  td_d =td_d1;    else td_d =td_d2;
                   if (V(db)>v_high/2)  td_db=td_d1;    else td_db=td_d2;
        end

      @(cross(V(feedback)-v_high/2.0, 1, ttol)) begin
                if (state>-1)
                   state = state -1;
                   if (V(u) >v_high/2)  td_u =td_u1;    else td_u =td_u2;
                   if (V(ub)>v_high/2)  td_ub=td_u1;    else td_ub=td_u2;
                   if (V(d) >v_high/2)  td_d =td_d1;    else td_d =td_d2;
                   if (V(db)>v_high/2)  td_db=td_d1;    else td_db=td_d2;
        end

      V(u)  <+transition((state==+1) ? v_high : 0.0, td_u,  tt);
      V(ub)<+transition((state==+1) ? 0.0 : v_high, td_ub, tt/2.0);
      V(d)  <+transition((state==-1) ? v_high : 0.0, td_d,  tt);
      V(db)<+transition((state==-1) ? 0.0 : v_high, td_db, tt/2.0);

end
endmodule
```

Listing D.2: Behavioral model of CP in Verilog A

```
//CP

`include "constants.vams"
`include "disciplines.vams"

module CP(Iout, Down, N_Down, N_Up, Up, Ibias);
   input Up, N_Up, Down, N_Down;
   output Iout;
   electrical Up, N_Up, Down, N_Down, Iout;

   parameter real v_high = 3;
   parameter real v_low = 0;
   parameter real v_th =(v_high-v_low)/2;          //threshold voltage
   parameter real TransTime=10p from (0:1000000);
   parameter real Delay=1p from (0:1000000);
   parameter real Ip=25.0e-6;       //charge pump's output current
   parameter real v_max=2.85;           //maximum voltage at output node so that
                                        // Pmos current source in saturation
   parameter real v_min=0.35;           //minimum voltage at output node so that
                                        // Nmos current source in saturation
   parameter real Mis=0.00;             //Mismatch of up and down current

   integer state; // CP state: "-1->charge","1->discharge","0->no output current"

analog begin
   @(initial_step) begin
       state=0;
   end
   @(cross(V(Up)-v_th, 1)) begin        // current charge
       state = -1;
   end
   @(cross(V(Down)-v_th, 1)) begin      // current discharge
       state = 1;
   end
   @(cross(V(Up)-v_th, -1)) begin       // no output current
       state =0;
   end
   @(cross(V(Down)-v_th, -1)) begin
       state =0;
   end

   // restrict the output voltage range from v_min to v_max
   @(cross(V(Iout)-v_max, 1)) begin
         state =0
   end
   @(cross(V(Iout)-v_min, -1))begin
       state =0;
   end

  I(Iout)<+transition(Ip*state*(1+state*Mis), Delay, TransTime );

end
endmodule
```

Listing D.3: Behavioral model of VCO in Verilog A

```
//VCO

`include "discipline.h"
`include "constants.h"

module vco(V_tune,VCO_out);
  input V_tune;
  output VCO_out;
  electrical V_tune, VCO_out;

  parameter real VSS=0, VDD=3;
  parameter real Vmin=0.6;
  parameter real Vmax=2.6 from(Vmin:10e5);
  parameter real Fmin=50e6 from(0:10e9);
  parameter real Kvco=600e6;
  parameter real tt=0.0001/Fmin from(0:10e3);
  parameter real ttol=1e-8/Fmin from(0:1/Fmin);

  real freq, phase, Vout;

  analog begin
     @(initial_step)begin
        Vout=VSS;
     end

    // compute the freq from the input voltage
    if  (V(V_tune)<=Vmin)
          freq = Fmin;
    else if (V(V_tune)<=Vmax)
          freq = (V(V_tune)-Vmin)*Kvco+Fmin;
    else
          freq = (V(V_tune)-Vmax)*(0.25*Kvco)+(Vmax-Vmin)*Kvco+Fmin;

    // ideal function
    // freq=(V(V_tune)-Vmin)* Kvco + Fmin;

    //phase is the integral of the frequency modulo 1
     phase=idtmod(freq, 0.0, 1.0, -0.5);

     //update jitter twice per period
     @(cross(phase-0.25,1,ttol))begin
        Vout=VDD;
     end
     @(cross(phase+0.25,1,ttol))begin
        Vout=VSS;
     end
     V(VCO_out)<+transition(Vout,0,tt);

end
endmodule
```

Listing D.4: Behavioral model of divider in Verilog A

```
// divider

`include "constants.h"
`include "discipline.h"

module divider(clock_out, clock_in);
   input clock_in;
   output clock_out;
   electrical clock_in,clock_out;

   parameter real Nmin = 6 ;           //minimum divider value
   parameter real Nmax = 20 ;          //maximum divider value
   parameter real v_high = 3;
   parameter real v_low  = 0;
   parameter real v_th = 1.5 ;
   parameter real tt = 10p;            // time of rise and fall time
```

```
    parameter real td = 0;
    parameter real JumpTime=8u;  // output frequency jump from minimum maximum

    integer count,n,M;

analog begin

   @(cross((V(clock_in)-v_th),1)) begin
        if ($abstime>=JumpTime) M=Nmax;
        else     M=Nmin;

        count=count+1;
        if (count>=M) count=0;
        n = (2*count >= M);
   end

   V(clock_out)<+ transition(n? v_high : v_low, td, tt);

end
endmodule
```

Lists D.5- D.8 are for the extraction of the jitter performance.

Listing D.5: Behavioral model of oscillator in Verilog A

```
//Fixed-frequency oscillator (OSC)with accumulating and synchronous jitter

`include "disciplines.h"

module OSC(out);
    output out;
    electrical out;

    parameter real freq=25e6 from (0:10e9);
    parameter real ratio=1 from (0:10e9);
    parameter real VSS=0, VDD=3;
    parameter real tt=0.01*ratio/freq from (0:10e9);
    parameter real accJitter=0 from [0:0.1/freq);
                                  //period jitter for reference osc and divider
    parameter real syncJitter=0 from [0:0.1*ratio/freq);
                                  //edge-to-edge jitter, for divider and PFD/CP

    integer n, accSeed, syncSeed;
    real next, dT, dt, accSD, syncSD, Jcp;

analog begin
   @(initial_step) begin
        accSeed=286;
        syncSeed=-459;
        accSD=accJitter*sqrt(ratio/2);
        syncSD=syncJitter;
        next=0.5/freq+$abstime;
   end

    // calculation jitter value from fitting fucntion
    Jcp=fittingfunction1(Icp);
    syncSD=Jcp;

   @(timer(next+dt))begin
        n=!n;
        dT = accSD *$dist_normal(accSeed,0,1);
        dt = syncSD*$dist_normal(syncSeed,0,1);
        next=next+0.5*ratio/freq+dT;
   end

   V(out)<+transition(n?VDD:VSS,0,tt);
end
endmodule
```

Listing D.6: Behavioral model of PFD/CP in Verilog A

```
// Phase-Frequency Detector with Charge Pump
//
// pfd_cp1: a simple three state phase-frequency detector
// Version 1a, 12 July 03
//
// Ken Kundert
//
// Downloaded from The Designer's Guide (www.designers-guide.org).
// Post any questions to www.designers-guide.org/Forum

‘include "disciplines.vams"
‘include "constants.vams"

//
// This model exhibits no jitter
// The jitter of PFD/CP is integrated into OSC block

module pfd_cp1 (Iout, ref, vco);

output Iout; electrical Iout;    // current output
input ref; voltage ref;          // positive input (edge triggered)
input vco; voltage vco;          // inverting input (edge triggered)
parameter real iout=20u;         // maximum output current
parameter real VDD=1.8;          // output voltage in high state
parameter real VSS=0;            // output voltage in low state
parameter real Vth=(VDD+VSS)/2;  // threshold voltage at input
parameter integer dir=1 from [-1:1] exclude 0;
                                 // dir=1 for positive edge trigger
                                 // dir=-1 for negative edge trigger
parameter real tt=1n from (0:inf);      // transition time of output signal
parameter real ttol=1p from (0:inf);
integer state;

analog begin
    @(cross(V(ref)-Vth, dir, ttol)) begin
        if (state > -1) state = state - 1;
    end
    @(cross(V(vco)-Vth, dir, ttol)) begin
        if (state < 1) state = state + 1;
    end

    I(Iout) <+ transition(iout*state, 0, tt);

end
endmodule
```

Listing D.7: Behavioral model of VCO with jitter in Verilog A

```
//VCO&D together exhibits Jitter
‘include "discipline.h"
‘include "constants.h"

module VCO_Div(V_tune,VCO_out);
  input V_tune;
  output VCO_out;
  electrical V_tune, VCO_out;
  parameter real VSS=0, VDD=3;
  parameter real Vmin=0.6;
  parameter real Vmax=2.6 from(Vmin:10e5);
  parameter real Kvco=600e6;
  parameter real Fmin=50e6 from(0:10e9);
  parameter real tt=0.0001/Fmin from(0:10e3);
  parameter real ttol=1e-8/Fmin from(0:1/Fmin);
  parameter real N=20;                  //if N!=1, divider moved into the VCO block
  parameter real Ivco=400e-6;           //current comsumption
  parameter real Jvco=10e-12;           //Jitter of VCO
  real freq, phase, dT, delta, Vout;
  integer seed, fvco;
  analog begin
     @(initial_step)begin
        seed=160;
        Vout=VSS;
        delta=Jvco*sqrt(2*N); // calculating the corresponding jitter on the divider output
     end

     //calculate the freq from the input voltage
     if (V(V_tune)<Vmin)
          freq=Fmin;
     else if (V(V_tune)<=Vmax)
          freq=(V(V_tune)-Vmin)* Kvco + Fmin;
     else
          freq=(V(V_tune)-Vmax)*(0.25*Kvco)+(Vmax–Vmin)*Kvco+Fmin;

     //apply the frequency divider, add the phase noise
     freq= (freq/N) / (1 + dT * freq / N);

     //calculation of Jvco through fitting function
     Jvco= fittingfunction2(Ivco,Kvco);

     //phase is the integral of the frequency modulo 1
     phase=idtmod(freq, 0.0, 1.0, -0.5);

     //update jitter twice per period
     @(cross(phase-0.25,1,ttol))begin
        dT=delta*$dist_normal(seed,0,1);
        Vout=VDD;
     end
     @(cross(phase+0.25,1,ttol))begin
        dT=delta*$dist_normal(seed,0,1);
        Vout=VSS;
     end
     V(VCO_out)<+transition(Vout,0,tt);
end
endmodule
```

Listing D.8: Behavioral model of PLL outputs’period measurement in Verilog A

```
// VerilogA for PLL_JitterMod, PLLoutput, veriloga
`include "constants.vams"
`include "disciplines.vams"

module PLLoutput (VCO_out);
   input VCO_out;
   electrical VCO_out;
   parameter real tstart = 20e-6;   // start to write an output file
   parameter real VDD = 3;
   parameter real VTH = VDD/2;
   integer fp;
   real prev;

analog begin
    @(initial_step)begin
        fp=$fopen("PLLPeriods.tcl","w");
    end
    @(cross((V(VCO_out)-VTH),1)) begin
        if ($abstime>=tstart) $fstrobe(fp,"%3.10e",$abstime-prev);
        prev=$abstime;
    end
    @(final_step) begin
        $fclose(fp);
    end
end
endmodule
```

Bibliography

[Abi97] Asad Abidi. Analog circuit design-rf analog-to-digital converters sensor and actuator interfaces; low-noise oscillators, plls and synthesizers. chapter How Phase Noise Appears in Oscillators. Kluwer Academic Publishers, 1997.

[AEGP00] K. Antreich, J. Eckmueller, H. Graeb, and M. Pronath. Wicked: Analog circuit synthesis incorporating mismatch. In *IEEE Custom Integrated Circuits Conference (CICC)*, pages 511–514, May 2000.

[AETL96] Brian A. A. Antao, Fatehy M. El-Turky, and Robert H. Leonowich. Behavioral modeling phase-locked loops for mixed-mode simulation. *Analog Integrated Circuits and Signal Processing*, (10):45–65, 1996.

[AGC94] K. Antreich, H. Graeb, and U. W. Claudia. Circuit analysis and optimization driven by worst-case distance. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 13(1):57–71, Jan. 1994.

[AH06] Bashir Al-Hashimi. *System-on-chip: Next Generation Electronics*. IET, 2006.

[AN96] O. Aaserud and I. Ring Nielsen. Trends in current analog design¡ªa panel debate. *Analog Integrated Circuits and Signal Processing*, (7):5–9, 1996.

[AP87] S. Ardalan and J. Paulos. An analysis of nonlinear behavior in delta-sigma modulators. *IEEE Transactions on Circuits and Systems*, 34(6):593– 603, Jun. 1987.

[APT02] Peter J. Ashenden, Gregory D. Peterson, and Darrell A. Teegarden. *The System Designer's Guide to VHDL-AMS*. Morgan Kaufmann Publishers, San Francisco, 2002.

[Ass] Semiconductor Industry Association. *Inernational Technology Roadmap for Semiconductors*. 1999 edition.

[Bak04] R. Jacob Baker. *CMOS: Mixed Signal Circuit Design*. Wiley-IEEE Press, 2004.

[Bes03] Roland E. Best. *Phase-Locked Loops Design, Simulation, and Applications*. Mc-Graw-Hill, 2003.

[BFM$^+$99] S. Brigati, F. Francesconi, P. Malcovati, D. Tonietto, A. Baschirotto, and F. Maloberti. Modelling sigma-delta modulator nonidealities in simulink. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 384–387, Jul. 1999.

[BGH04] O. Bajdechi, G. E. Gielen, and J. H. Huijsing. Systematic design exploration of delta-sigma adcs. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, pages 86–95, Jan. 2004.

[BGV$^+$04] F. De Bernardinis, S. Gambini, R. Vincis, F. Svelto, A. Sangiovanni-Vincentelli, and R. Castello. Design space exploration for a umts front-end exploiting analog platforms. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 923– 930, Nov. 2004.

[BJS03] F. De Bernardinis, M. I. Jordan, and A. SangiovanniVincentelli. Support vector machines for analog circuit performance representation. In *ACM/IEEE Design Automation Conference (DAC)*, Jun. 2003.

[BKMA88] B. E. Boser, K.-P. Karmann, H. Martin, and B. A.Wooley. Simulating and testing oversampled analog-to-digital converters. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 7(6):668–674, June 1988.

[BM04] Bernhard Boser Boris Murmann. Digitally assisted analog integrated circuits. In *SPECIAL SECTION: Q focus: DSP*, Mar. 2004.

[BNSV05] F. De Bernardinis, P. Nuzzo, and A. Sangiovanni-Vincentelli. Mixed signal design space exploration through analog platforms. In *ACM/IEEE Design Automation Conference (DAC)*, pages 875–880, Jun. 2005.

[BNSV06] F. De Bernardinis, P. Nuzzo, and A. Sangiovanni-Vincentelli. Robust system-level design with analog platforms. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2006.

[BSI] *http://www-device.eecs.berkeley.edu/ bsim3/latenews.html*.

[BW88] B. E. BOSER and B. A. Wooley. The design of sigma-delta modulation analog-to-digital converters. *IEEE Journal of Solid-State Circuits*, 23(6):1298–1308, Dec. 1988.

[CA05] M. Chu and D. J. Allstot. Phase-locked loop synthesis using hierarchical divide-and-conquer multi-optimization. In *Asia and South Pacific Design Automation Conference*, pages 675–678, Jan. 2005.

[Cada] *Cadence Design Systems (www.cadence.com)*.

[Cadb] Cadence Design Systems (www.cadence.com). *VIRTUOSO SPECTRE RF SIMULATION*.

[Cad02] Cadence. The rise of digital/mixed-sginal semiconductors and system-on-a-chip. White Paper, Jun. 2002.

[CB99] Ernst Christen and Kenneth Bakalar. Vhdl-ams-a hardware description language for analog and mixed-signal applications. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(10):1263–1272, Oct. 1999.

[CCC+97] H. Chang, E. Charbon, U. Choudhury, A. Demir, E. Felt, E. Liu, A. Sangiovanni-Vicentelli, and I. Vassiliou. *A Top-Down, Constraint-Driven Design Methodology for Analog Integrated Circuit*. Kluwer Academic Publishers, 1997.

[CHC+05] Yu-Tsun Chien, Li-Ren Huang, Wen-Tzao Chen, Gin-Kou Ma, and T. Mukherjee. Speed: synthesis of high-performance large scale analog/mixed-signal circuit. In *VLSI Design, Automation and Test*, pages 112–115, Apr. 2005.

[CPH94] K. W. Current, J. F. Parker, and W. J. Hardaker. On behavioral modeling of analog and mixed-signal circuits. In *Signals, Systems and Computers, 1994. 1994 Conference Record of the Twenty-Eighth Asilomar Conference on*, pages 264 – 268, 1994.

[CPH+03] D. M. Colleran, C. Portmann, A. Hassibi, C. Crusius, S. S. Mohan, S. Boyd, T. H. Lee, and M. D. M. Hershenson. Optimization of phase-locked loop circuits via geometric programming. In *IEEE Custom Integrated Circuits Conference (CICC)*, pages 377– 380, Sep. 2003.

[CS00] J. A. Cherry and W. M. Snelgrove. *Continuous-Time Delta-Sigma Modulators for High-Speed A/D Conversion: Theory, Practice and Fundamental Performance Limits*. Kluwer Academic Publishers, 2000.

[CSC97] L. R. Carley, R. Schreier, and G. C.Temes. Delta-sigma data converters: Theory, design, and simulation. chapter Delta-Sigma ADCs with Multibit Internal Converters. IEEE Press, 1997.

[CSVM03] Henry Chang, Alberto Sangiovanni-Vincentelli, and Enrico Malavasi. *A Top-Down, Constraint-Driven Design Methodology for Analog Integrated Circuits*. Springer, 2003.

[CT92] J. C. Candy and G. C. Temes. *Oversampling Delta-Sgima Converters*. IEEE Press, 1992.

[DGS+96] S. Donnay, G. Gielen, W. Sansen, W. Kruiskamp, D. Leenaerts, S. Buytaert, K. Marent, M. Buckens, and C. Das. Using top-down cad tools for mixed analog/digital asics: a practical design case. *Analog Integrated Circuits and Signal Processing*, 10(1-2):101–117, Jan. 1996.

[DLG+98] G. Debyser, F. Leyn, G. Gielen, W. Sansen, and M. Styblinski. Efficient statistical analog ic design using symbolic methods. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 21–24, 1998.

[DLM92] V. F. Dias, V. Liberali, and F. Maloberti. Design tools for oversampling data converters needs and solutions. *Microelectronics Journal*, 23:641–650, 1992.

[DMR00] A. Demir, A. Mehrotra, and J. Roychowdhury. Phase noise in oscillators: A unifying theroy and numerical methods for characterization. *IEEE Transactions on Circuits and Systems CAS*, 47(5):655–674, May 2000.

[DND87] M. GR Degrauwe, O. Nys, and E. Dijkstra. Idac: An interactive design tool for analog cmos circuits. *IEEE Transactions on Circuits and Systems*, 22(6):1106–1116, Dec. 1987.

[dPDL$^+$01] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. Gielen, W. Sansen, P. Veselinovic, and D. Leenaerts. Amgie-a synthesis environment for cmos analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 20(9):1037–1058, Sep. 2001.

[dPGS02] G. Van der Plas, G. Gielen, and W. Sansen. *A Computer-Aided Design and Synthesis Environment for Analog Integrated Circuits*. Kluwer, Dordrecht, The Netherlands, 2002.

[DR69] S. Director and R. Rohrer. Automated network design -the frequency domain case. *IEEE Trans. Citcuit Theory*, 16:330–337, Aug. 1969.

[Dra01] B. Drakhlis. Calculate oscillator jitter by using phase-noise analysis. *Microwaves & RF*, pages 82–90, Jan. 2001.

[Eld] *Mentor Grafic Inc. (http://www.mentor.com/products/. . . /eldo)*.

[EMG05] T. Eeckelaert, T. McConaghy, and G. G. E. Gielen. Efficient multiobjective synthesis of analog circuits using hierarchical pareto-optimal performance hyper-surfaces. In *Design, Automation and Test in Europe (DATE)*, pages 1070–1075, Feb. 2005.

[ESG$^+$06] T. Eeckelaert, R. Schoofs, G. Gielen, M. Steyaert, and W. Sansen. Hierarchical bottom-up analog optimization methodology validated by a delta-sigma a/d converter design for the 802.11a/b/g standard. In *ACM/IEEE Design Automation Conference (DAC)*, pages 25–30, Jul. 2006.

[ET96] C. C. Enz and G. C. Temes. Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling and chopper stabilization. *Proceedings of the IEEE*, 84(11):1583–1614, Nov. 1996.

[FAB99] Medeiro Fernando, Perez-Verdu Angel, and Pirez-Verdz Belen. *Top-down design of high-performance sigma-delta modulators*. Springer, 1999.

[FM98] Angel Rodriguez-Vazquez Fernando Medeiro, Belen Perez-Verdu. *Top-Down Design of High-Performance Sigma-Delta Modulators*. Springer, 1998.

[FRV96] Francisco V. Fernandez and Angel Rodriguez-Vazquez. Symbolic analysis tools the state-of-the-art. In *Proc. IEEE Int. Conf. Circuits Systems*, pages 798–801, 1996.

[Gar90] William A. Gardner. *Introduction to Random Process: With Applications Signals and Systems*. Mc-Graw-Hill, New York, 1990.

[GM92] Paul R. Gray and Robert G. Meyer. *Analysis and Design of Analog Integrated Circuits*. John Wiley & Sons, 1992.

[GR00] G. G. E. Gielen and R. A. Rutenbar. Computer-aided design of analog and mixed-signal integrated circuit. *Proceedings of the IEEE*, 88(12):1825–1852, Dec. 2000.

[Gra93] H. Graeb. *Schaltungkreisoptimierung mit Worst-Case-Abstaenden als Zielgroessen*. Dissertation, Institute of Electrical Automation Design at Technical University Munich, 1993.

[GSS00] Y. Geerts, M. S. J. Steyaert, and W. Sansen. A high-performance multibit $\delta\sigma$ cmos adc. *IEEE Journal of Solid-State Circuits*, 35:1829–1840, Dec. 2000.

[GWS90] G. Gielen, H. Walscharts, and W. Sansen. Analog circuit design optimization based on symbolic simulation and simulated annealing. *IEEE Journal of Solid-State Circuits*, 25:707–713, Jun. 1990.

[GWS94] G. Gielen, P. Wambacq, and W. M. Sansen. Symbolic analysis methods and applications for analog circuits: atutorial overview. In *Proceedings of the IEEE*, pages 287–304, 1994.

[GZEA01] H. Graeb, S. Zizala, J. Eckmueller, and K. Antreich. The sizing rules method for analog integrated circuit design. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 343–349, Nov. 2001.

[GZMS07] H. Graeb, J. Zou, D. Mueller, and U. Schlichtmann. Hierarchische optimierung einer phasenregelschaltung. In *ASIM/GI Fachgruppen-Tagung: Simulation technischer Systeme und Grundlagen und Methoden in Modellbildung und Simulation*, Feb. 2007.

[HBL98] M. D. M. Hershenson, S. P. Boyd, and T. H. Lee. Gpcad: A tool for cmos opamp synthesis. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 296–303, 1998.

[HBL01] M. D. M. Hershenson, S. P. Boyd, and T. H. Lee. Optimal design of a cmos op-amp via geometric programming. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 20(1):1–21, Jan. 2001.

[HBMM04] P. K. Hanumolu, M. B., K. Mayaram, and U. K. Moon. Anaylsis of charge-pump phase-locked loops. *IEEE Transactions on Circuits and Systems CAS*, 51(9):1665–1674, Sep. 2004.

[HEL92] J. P. Harvey, M. I. Elmasry, and B. Leung. Staic: an interactive framework for synthesizing cmos and bicmosanalog circuits. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 11(11):1402–1417, 1992.

[Her02] M. D. M. Hershenson. Design of pipeline analog-to-digital converters via geometric programming. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 317–324, 2002.

[Her03] M. D. M. Hershenson. Efficient description of the design space of analog circuits. In *ACM/IEEE Design Automation Conference (DAC)*, pages 970–973, Jun. 2003.

[HIUK86] T. Hayashi, Y. Inabe, K. Uchimura, and T. Kimura. A multistage delta-sigma modulator without double integration loop. In *Solid-State Circuits Conference, Digest of Technical Papers (ISSCC)*, pages 182–183, 1986.

[Hja03] Emil Hjalmarson. *Studies on Design Automation of Analog Circuits - The Design Flow*. Dissertation, Department of Electrical Engineering, Linkpings University, Sweden, 2003.

[HKH00] M. Hinz, I. Koeenkamp, and E. H. Horneber. Behavioral mdoeling and simulation of phase-locked loops for rf front ends. In *IEEE Midwest Symp. On Circuits and Systems*, pages 194–194, Aug. 2000.

[HL98] A. Hajimiri and T. H. Lee. A gereral theroy of phase noise in electrical oscillators. *IEEE Journal of Solid-State Circuits*, 33(2):179–194, Feb. 1998.

[HL99] A. Hajimiri and T. H. Lee. *The Design of Low Noise Oscillators*. Springer, 1999.

[HLL99] A. Hajimiri, S. Limotyrakis, and T. H. Lee. Jitter and phase noise in ring oscillators. *IEEE Journal of Solid-State Circuits*, 34(6):790–804, Jun. 1999.

[HM79] C. L. Hwang and A. S. M. Masud. *Multiple Objective Decision Making*. Springer, 1979.

[HMBL99] M. D. M. Hershenson, S. Mohan, S. Boyd, and T. Lee. Optimization of inductor circuits via geometric programming. In *ACM/IEEE Design Automation Conference (DAC)*, pages 994–998, Jun. 1999.

[HRC89] R. Harjani, R. Rutenbar, and L. R. Carley. Oasys: a framework for analog circuit synthesis. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 8:1247–1265, Dec. 1989.

[HS88] J. P. Hein and J. W. Scott. z-domain model for discrete-time pll's. *IEEE Transactions on Circuits and Systems CAS*, 35(11):1393–1400, Nov. 1988.

[HS96] R. Harjani and J. Shao. Feasibility and performance region modleing of analog and digital ciruits. *Analog Integrated Circuits and Signal Processing*, pages 23–43, Jun. 1996.

[jit] *http://www.mwrf.com/Articles/.*

[JYM+07] Lin Jia, Kiat Seng Yeo, Jian Guo Ma, Manh Anh Do, and Xiao Peng Yu. Noise transfer characteristics and design techniques of a frequency synthesizer. In *Analog Integrated Circuits and Signal Processing*, Sep. 2007.

[KCJ+00] K. Kundert, H. Chang, D. Jefferies, G. Lamant, E. Malavasi, and F. Sendig. Design of mixed-signal system-on-a-chip. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 9(12):1561–1571, Dec. 2000.

[KHC+01] K. Kundert, H, F. L. Chang, D. Jefferies, G. Lamant, E. Malavasi, and F. Sendig. Design of mixed-signal system-on-a-chip. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 19(12):1561–1571, Dec. 2001.

[KHE+86] R. Koch, B. Heise, F. Eckbauer, E. Engelhardt, J. A. Fisher, and F. Parzefall. A 12-bit sigma-delta analog-to-digital converter with a 15-mhz clock rate. *IEEE Journal of Solid-State Circuits*, 21(6):1003–1010, Dec. 1986.

[KLK05] J. Koy, W. Lee, and S. W. Kim. 2.5ghz pll with current matching charge-pump for 10gbps transmitter design. In *Great Lakes Symposium on VLSI (GLS-VLSI)*, pages 122–125, Apr. 2005.

[KPH+01] M. J. Krasnicki, R. Phelps, J. R. Hellums, M. McClung, R. A.Rutenbar, and L. R.Carley. Asf: a practical simulation-based methodology for the synthesis ofcustom analog circuits. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 350–357, 2001.

[KPRC99] M. Krasnicki, R. Phelps, R. A. Rutenbar, and L. R. Carley. Maelstrom: efficient simulation-based synthesis for custom analog cells. In *ACM/IEEE Design Automation Conference (DAC)*, pages 945–950, Jun. 1999.

[KSpG90] H. Koh, C. Sequin, and p. Gray. Opasyn: a compiler for cmos operational amplifiers. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 9:113–125, Feb. 1990.

[Kun05] Ken Kundert. *Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizer*. The Gesigner's Guide Community (www.designers-guide.org), 2005.

[LH00] T. H. Lee and A. Hajimiri. Oscillator phase noise: A tutorial. *IEEE Journal of Solid-State Circuits*, 35(3):326–336, Mar. 2000.

[LMC01] Chang-Hyeon Lee, K. McClellan, and J. Jr. Choma. Supply noise insensitive pll design through pwl behavioral modelingand simulation. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 48(12):1137–1144, Dec. 2001.

[LNGB88] B. N. Leung, R. Neff, P. R. Gray, and R. W. Brodersen. A four-channel cmos oversampled pcm voiceband coder. In *Solid-State Circuits Conference, Digest of Technical Papers (ISSCC)*, pages 106–107, 1988.

[LTZ79] M. Lightner, T. Trick, and R. Zug. Circuit optimization and design, circuit analysi, simulation and design part 2 (a. ruehli). *Advances in CAD for VLSI 3*, pages 333–391, 1979.

[LWT$^+$05] Xin Li, Jian Wang, L. T.Pileggi, Tun-Shih Chen, and Wanju Chiang. Performance-centering optimization for system-level analog design exploration. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 422– 429, Nov. 2005.

[Mal02] F. Maloberti. *Sigma Delta converters, basic issues*. UT Dallas, 2002.

[Man96] J. G. Maneatis. Low-jitter process-independent dll and pll based on self-baised techniques. *IEEE Journal of Solid-State Circuits*, pages 1723–1732, Nov. 1996.

[Man03] Mozhgan Mansuri. *Low-Power Low-Jitter On-Chip Clock Generation*. Dissertation, Universtiy of California Los Angels, 2003.

[Mat] *MATLAB®, The Mathworks (www.mathworks.com)*.

[MBF$^+$03] P. Malcovati, S. Brigati, F. Framcesconi, F. Maloberti, P. Cusinato, and A. Baschirotoo. Behavioral modeling of switched-capacitor sigma-delta modulators. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 50(3):352–364, 2003.

[Meh02] Amit Mehrotra. Noise analysis of phase-locked loops. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(9):1309–1316, Sep. 2002.

[Men] *Mentor Grafic Inc. (http://www.mentor.com/)*.

[M.G80] Floyd M.Gardner. Charge-pump phase-locked loops. *IEEE Transactions on Communications*, 28(11):1849–1858, Nov. 1980.

[Mis04] Ajay R. Mishra. *Fundamentals of Cellular Network Planning and Optimisation*. John Wiley and Sons, 2004.

[Moo] *The Lives and Death of Moore's Law, www.firstmonday.org/issues/issue7_11/tuomi*.

[MPVARVH94] F. Medeiro, B. Perez-Verdu, A-Rodriguez-Vazquez, and J. L. Huertas. Modeling opamp-induced harmonic distortion for switched-capacitor $\delta\sigma$ modulator design. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 445–448, 1994.

[MSG03] T. Massier, G. Stehr, and H. Graeb. Ein beitrag zur automatisierung der strukturanalyse and der implziten spezification von analogen integrierten schaltungen. In *GMM/ITG Diskussionssitzung Entwirf von Analogschaltung (Analog'03)*, Jun. 2003.

[MSGS05] D. Mueller, G. Stehr, H. Graeb, and U. Schlichtmann. Deterministic approaches to analog performance space exploration. In *ACM/IEEE Design Automation Conference (DAC)*, pages 869–874, Jun. 2005.

[MT95] C. A. Makris and C. Toumazou. Analog ic design automation. ii: Automated circuit correction by qualitative reasoning. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 14(2):239–254, Feb. 1995.

[mun] *MunEDA GmbH (www.muneda.com)*.

[Neo] *Neolinear Inc. (acquired by Cadence in 2004) (www.cadence.com/ neolinear)*.

[NST97] Steven R. Norsworthy, Richard Schreier, and Gabor C. Temes. *Delta-Sigma Data Converters:(Theory, Design, and Simulation)*. Wiley-IEEE Press, 1997.

[Oct] *http://www.gnu.org/software/octave/*.

[OHSA04] T. Oura, Y. Hiraku, T. Suzuki, and H. Asai. Modeling and simulation of phase-locked loop with verilog-a description for top-down design. In *Asia and South Pacific Design Automation Conference*, pages 549–552, Dec. 2004.

[ORC96] E. S. Ochotta, R. A. Rutenbar, and L. R. Carley. Synthesis of high performancee analog circuits in astrx/oblx. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, pages 4271–294, Mar. 1996.

[Pap91] A. Papoulis. *Probability, Raqndom Variables and Stochastic Processes*. McGraw-Hill, 1991.

[Par] Sangil Park. *Principles of Sigma-Delta Modulation for Analog-to-Digital Converters*. Motorola Digital Signal Processors.

[PKR$^+$00] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley, and J. R. Hellums. Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 19(6):703–717, Jun. 2000.

[Raz96] Behzad Razavi. A study of phase noise in cmos oscillators. *IEEE Journal of Solid-State Circuits*, pages 331–343, Mar. 1996.

[Raz97] Behzad Razavi. *RF Microelectronics*. Prentice Hall, 1997.

[Raz01] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*. Mc-Graw-Hill, 2001.

[Rob03] Neil Roberts. Phase noise and jitter-a primer for digital designers. In *EE Design*, Jul. 2003.

[RSA99] H. Graeb R. Schwencker, J. Eckmueller and K. Antreich. Automatische nominalpunktdimensionierung analoger cmos schaltung mit parameterabstaeden als zielgroessen. In *GME/ITG-Dissussionssitzung Entwicklung von Analogschaltungen mit CAE Methoden*, Feb. 1999.

[Sab] *Synopsys Inc. (http://www.synopsys.com/products/mixedsignal/saber)*.

[Sch04] Frank C. Schenkel. *Tolerance Analysis and Design Centering of analog Circuits, with Considering of Mixmatch*. Dissertation, Institute of Electrical Automation Design at Technical University Munich, 2004.

[SCP05] A. Somani, P. P. Chakrabarti, and A. Patra. Mixing global and local competition in genetic optimization based design space exploration of analog circuits. In *Design, Automation and Test in Europe (DATE)*, pages 1064– 1069, Feb. 2005.

[SG99] B. D. Smedt and G. Gielen. Models for systematic design and verification of frequency synthesizers. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(10):1301–1308, Oct. 1999.

[SG03] B. D. Smedt and G. G. E. Gielen. Watson: Design space boundary exploration and model generation for analog and rf ic design. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 22(2):213–224, Feb. 2003.

[SGA03] G. Stehr, H. Graeb, and K. Antreich. Performance trade-off analysis of analog circuits by normal boundary intersection. In *ACM/IEEE Design Automation Conference (DAC)*, Jun. 2003.

[SGA04] G. Stehr, H. Graeb, and K. Antreich. Analog performance space exploration by fourier-motzkin elimination with application to hierarchical sizing. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2004.

[Spe] *Cadence Inc. (http://www.cadence.com/products/custom_ic/spectre/index.aspx)*.

[SS88] R. Spence and R. S. Soin. *Tolerance Design of Electronic Circuits*. Addision Wesley, England, 1988.

[Ste05] Guido Stehr. *On the Performance Space Exploration of Analog Integrated Circuit*. Dissertation, Institute of Electrical Automation Design at Technical University Munich, 2005.

[Syn] *Synopsys Inc. (www.synopsys.com)*.

[SZ96] R. Schreier and B. Zhang. Delta-sigma modulators employing continuous-time circuitry. 43:324–442, Apr. 1996.

[TM95] C. Toumazou and C. A. Makris. Analog ic design automation. i. automated circuit generation: newconcepts and methods. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 14(2):218–238, Feb. 1995.

[TMG02] C. Toumazou, G. Moschytz, and B. Gilbert. *Trade-offs in Analog Circuit Desig: The Designer's Companion*. Kluwer Academic Publishers, Boston, 2002.

[TP89] FATEHY EL TURKY and E. E. PERRY. Blades: an artificial intelligence approach to analog circuit design. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 8(6):680–691, Jun. 1989.

[TTR06] S. K. Tiwary, P. K. Tiwary, and R. A. Rutenbar. Gerneration of yield-aware pareto surfaces for hierarchical circuit design space exploration. In *ACM/IEEE Design Automation Conference (DAC)*, pages 31–36, Jul. 2006.

[TVRM04] S. K. Tiwary, S. Velu, R. A. Rutehnbar, and T. Mukherjee. Pareto-optimal modeling for efficient pll optimization. *Modeling and Simulation of Microsystems, Nanotech*, pages 195–198, 2004.

[VCD$^+$96] Iasson Vassiliou, Henry Chang, Alper Demir, Edoardo Charbon, Paolo Miliozzi, and Alberto L. Sangiovanni-Vincentelli. A video driver system designed using a top-down, constraint-driven methodology. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 463–468, 1996.

[Vera] *http://www.designers-guide.org/VerilogAMS/*.

[Verb] *http://www.verilog.org/verilog-ams/htmlpages/public-docs/lrm/VerilogA/verilog-a-lrm-1-0.pdf*.

[VHD] *http://www.eda.org/vhdl-ams/*.

[WFG$^+$95] P. WAMBACQ, F. V. FERNANDEZ, G. GIELEN, W. SANSEN, and A. RODRIGUEZ-VAZQUEZ. Efficient symbolic computation of approximated small-signal characteristics of analog integrated circuits. *IEEE Journal of Solid-State Circuits*, 30(3):327–330, Mar. 1995.

[Wic] *MunEDA (www.muneda.com)*.

[wik] *http://en.wikipedia.org/wiki/Oversampling*.

[Wil] Nicolas Williams. *Facing the challenges in analog design*. www.eetimes.com/showArticle.jhtml?articleID=181502541.

[WRSVT88] W.Nye, D.C. Riley, A. Sangiovanni-Vincentelli, and A.L. Tits. An optimization-based system for the design of integrated circuits. *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, 7(4):501–519, 1988.

[WTHN04] S. Williams, H. Thompson, M. Hufford, and E. Naviasky. An improved cmos ring oscillator pll with less than 4ps rms accumulated jitter. In *IEEE Custom Integrated Circuits Conference (CICC)*, pages 151– 154, Oct. 2004.

[ZMG$^+$05] J. Zou, D. Mueller, H. Graeb, U. Schlichtmann, E. Hennig, and R. Sommer. Fast automatic sizing of a charge pump phase-locked loop basd on behavioral models. In *IEEE/VIUF International Workshop on Behavioral Modeling and Simulation (BMAS)*, pages 100–105, Sep. 2005.

[ZMGS06] J. Zou, D. Mueller, H. Graeb, and U. Schlichtmann. A cppll hierarchical optimization methodology considering jitter, power and locking time. In *ACM/IEEE Design Automation Conference (DAC)*, pages 19–24, Jul. 2006.

[ZMGS07a] J. Zou, D. Mueller, H. Graeb, and U. Schlichtmann. Optimization of switch-capacitor sigma-delta modulators based on worst-case-aware pareto-optimal front. In *IEEE Custom Integrated Circuits Conference (CICC)*, Sep. 2007.

[ZMGS07b] J. Zou, D. Mueller, H. Graeb, and U. Schlichtmann. Pareto-front computation and automatic sizing of cpplls. In *IEEE International Symposium on Quality Electronic Design (ISQED)*, Mar. 2007.

Nomenclature

General Conventions

β_w	Worst-case distance	37
$\mathbf{c}(\cdot)$	Sizing constraints	19
$\mathcal{D}$	Feasible parameter space	23
$\mathcal{F}$	Feasible performance space	23
$\mathbf{f}$	Circuit performances	16
$\mathbf{f}_\mathrm{l}$	Low specifications	17
$\mathbf{f}_\mathrm{u}$	Upper specifications	17
L	Transistor Length	19
$\mathbf{m}(\cdot)$	Mapping process from circuit parameters to circuit performances	17
$\mathbf{p}$	System-level parameters	27
$\mathbf{s}$	Statistical parameters	16
θ	Operational parameters	16
$\mathbf{w}$	Worst-case parameter	37
W	Transistor width	19
w_i	Weight factor	26
$\mathbf{x}$	Circuit-level parameters	29
Y	Yield	17

Switched-Capacitor Sigma-Delta Modulator

A	DC Gain of Op Amp	89
α	Actual fraction of the previous output of the integrator transferred to next stage	89
b	Integrator gain	88
C_f	Integrating capacitor	82
C_s	Sampling capacitor	82
e_T	kT/C noise	88
f_s	Sampling frequency	79

GBW	Unity-gain bandwidth of Op Amp	90
K	Boltzmann’s constant	88
N	Conversion resolution	79
R	Oversampling ration	79
$RN(t)$	Gaussian random number with zero mean and unity standard deviation	88
on$T$	Finite resistance of switch	88
SR	Slew rate of Op Amp	90
T	Absolute temperature	88
T_s	Sampling period	109
V_n	Total rms noise voltage of the operation amplifier	88
$E(z)$	z-transform of the quantization noise	83
$NTF(z)$	z-transfer function of the quantization noise	83
$STF(z)$	z-transfer function of the input signal	83
$X(z)$	z-transform of the input signal	83
$Y(z)$	z-transform of the out signal	83

Charge-Pump Phase-Locked Loop

C_1	Capacitor 1 in Loop filter	42
C_2	Capacitor 2 in Loop filter	42
f_0	Carrier frequency in spectrum	50
Δf	Offset frequency from carrier frequency	50
f_{max}	Maximal output frequency	43
f_{min}	Minimal output frequency	43
f_{out}	Output frequency	40
f_{ref}	Reference input frequency	40
$F_{\mathrm{LF}}(s)$	Transfer function of loop filter in s-domain	45
$H(s)$	Closed loop transfer function in s-domain	45
$H(z)$	Closed loop transfer function in z-domain	46
I_{CP}	Out-biased current of charge pump	42
J_{cc}	cycle-to-cycle jitter	52
J_{ee}	edge-to-edge jitter	52
J_k	long-term jitter	52
K_{VCO}	Gain of VCO in unit of Hz/V	43
K_{c}	Ratio of C_1 to C_2	46

$L(\Delta f)$	Phase noise on the offset frequency Δf in unit of dBc/Hz	50
$LG(s)$	Open loop transfer function in *s*-domain	45
$LG(z)$	Open loop transfer function in *z*-domain	46
N	Divider value	40
ϕ_{PM}	Phase margin	54
ϕ_{fb}	Phase of feedback signal	40
ϕ_{ref}	Phase of input reference signal	40
R	Resistor in Loop filter	42
$S_{\phi}(f)$	Noise power spectral density	53
$\sigma_{\Delta\phi}$	Phase jitter	52
$\sigma_{\Delta T}$	Timing jitter	52
T_{s}	Locking time	49
v_{max}	Maximal input voltage	43
v_{min}	Minimal input voltage	43
v_c	Voltage across the loop filter capacitor C_1	48
v_{ctrl}	VCO control voltage	48
ω_1	Loop filter time constant 1	46
ω_2	Loop filter time constant 2	46
ω_{p3}	PLL's third pole	54
ω_{UGB}	PLL's Unity-gain bandwidth	54
ω_{z}	PLL's zero	54
ω_i	Reference input frequency	46

List of Figures

List of Tables

Abstract in German

Die vorliegende Arbeit stellt ein automatisiertes hierarchisches Dimensionierungsverfahren für komplexe Analog-/Mixed-Signal-Schaltungen vor. Es wird von einem Systemmodell auf der Verhaltensebene und einer Modellierung der Systemblöcke auf der Transistorebene ausgegangen. Bei dem vorgeschlagenen zweistufigen Dimensionierungsprozess werden zunächst Systemparameter dimensioniert und als Spezifikation an die nachfolgende Transistordimensionierung der Systemblöcke weitergereicht. Durch eine Vorab-Charakterisierung des Eigenschaftspotentials der Systemblöcke mittels Pareto-Optimierung wird eine realistische Systemdimensionierung sichergestellt und so ein iterationsfreier Top-Down-Entwurfsprozess erzeugt. Das vorgestellte Verfahren wird anhand des Entwurfs von Phasenregelschleifen und Sigma-Delta-Modulatoren demonstriert.