TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Entwurfsautomatisierung

On the Structural Analysis of CMOS and Bipolar Analog Integrated Circuits

Tobias Massier

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. habil. Gerhard Rigoll
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. Ulf Schlichtmann
	2.	Univ.-Prof. Dr. rer. nat. Doris Schmitt-Landsiedel

Die Dissertation wurde am 01.07.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 29.03.2010 angenommen.

Fyll inte livet med dagar,
fyll dagarna med liv.

(svenskt ordspråk)

Acknowledgements

From November 2002 to May 2009, I had the opportunity to work as a research assistant in the analog group of the Institute for Electronic Design Automation, Technische Universität München. This thesis summarises the results of my research activities.

First of all, I would like to thank Professor Kurt Antreich and Professor Ulf Schlichtmann for giving me the chance to work at this institute.

I am very grateful to all of my colleagues. In particular, I feel obliged to PD Dr. Helmut Gräb, head of the analog research group, for all his advice, productive discussions, and proofreading my publications and this thesis.

Already before joining the Institute for Electronic Design Automation, Martin Strasser, Christian Michael Pehl, and Michael Eick assisted me in the course of my work with their Bachelor, Master, or Diploma theses. Martin Strasser created a graphical interface for the original version of the structure recognition software. Christian Michael Pehl implemented software for preparation of circuit netlists and laid the foundation for discrete and structural optimisation of analog circuits (Chapter 4.3). Michael Eick implemented the extended structure recognition algorithm (Chapter 4.2) and graphical user interfaces for both the structure recognition and the discrete optimisation programmes.

I would also like to express my gratitude to all other colleagues at the institute, especially of the analog group, namely Dr. Guido Stehr for supervising my diploma thesis and his suggestions during my PhD thesis, Daniel Müller, Husni Habal, Aurélien Tchego Kamdem, and Xin Pan for their advice and discussions. Thanks as well to Tatiana Rousseau and Yuanhang Chen who also assisted me as student workers.

In addition, I would like to thank all partners from industry. This incorporates all the staff at MunEDA, particularly Dr. Volker Glöckel, Dr. Michael Pronath, and Dr. Frank Schenkel for their assistance in several projects involving the tool WiCkeD.

Moreover, I want to express my gratitude to Dr. Regino Montiel-Montoya, Pietro Brenner, and Edmund Götz from Infineon Technologies AG Munich for their support in the field of bipolar transistor technology (Chapter 3.4), and to Dr. Josef Eckmüller, also from Infineon Technologies AG Munich, for providing the institute with industrial circuit examples to test and improve the structure recognition method.

Finally, I want to thank my parents Maria and Franz Massier as well as all my friends for their continuous support during the time of my doctorate.

Contents

Chapter 1

Introduction

In December 1947, William Bradford Shockley, John Bardeen, and Walter Houser Brattain invented the transistor. They were awarded the Nobel Prize in Physics "for their researches on semiconductors and their discovery of the transistor effect" in 1956 [NoC09]. The same year, the first integrated circuits were developed [WW99]. An integrated circuit is an assemblage of circuit elements on one circuit board connected via lines. In the 1950s, integrated circuits consisted of RTL (resistor-transistor logic) and DTL (diode-transistor logic). In the 1960s however, RTL and DTL were increasingly replaced by TTL (transistor-transistor logic) which requires less space on the chip. Until the early 1980s, integrated circuits usually consisted of bipolar transistors only. But with the emergence of the CMOS transistor, integrated circuits were increasingly manufactured with CMOS transistors [WW99]. CMOS transistors are able to operate at lower voltages than bipolar transistors. Hence, the supply voltage for integrated circuits could be reduced to significantly less than 10V. Today, many integrated circuits often require supply voltages of only about 1V. In addition, the integration density of CMOS transistors is much higher, so that more CMOS transistors can be combined on one chip. Today, integrated circuits are both realised in CMOS and BiCMOS, i.e., bipolar and CMOS transistors together in one circuit. Pure bipolar transistor circuits can still be found in high-power or high-frequency applications.

During the past 30 years, the principle of transistor-transistor logic has not changed substantially. However, transistors have become consistently smaller, allowing more transistors to be placed on one chip. In 1965, Gordon E. Moore described the phenomenon that since 1958, the number of transistors in integrated circuits had been reduplicating approximately every year, and this trend would continue in the future [Moo65]. In 1975, he changed his estimation to every second year. This phenomenon is known as Moore's law, though it is actually not a real law. Instead, it is just an estimation which still is still correct today.

However, in recent years, it was questioned if with transistor sizes decreasing to less than 100nm, Moore's law could be held up in the future as well. In April 2008, about half a year after Gordon E. Moore predicted the validity of his law to come to an end within

the next ten to 15 years, Intel's senior vice president Patrick P. Gelsinger estimated at the Intel Developer's Forum that Moore's law could well persist until 2029 [Gel08]. In [Pow08], the theoretical temporal limit of Moore's law has been calculated using physical constants. The result is year 2036 when transistor sizes reach the "characteristic dimension of an electron".

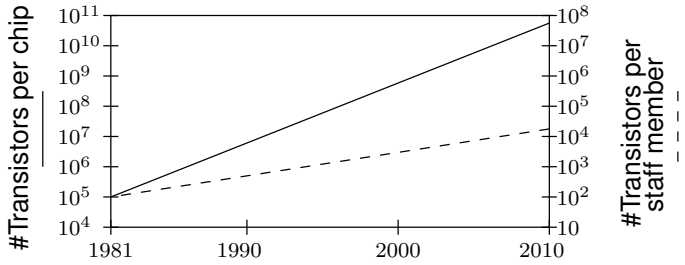

Figure 1.1: Potential Design Complexity and Designer Productivity

In [Sem07], a comparison between the potential design complexity and designer productivity is given. A diagram from this report is shown in Figure 1.1. While the design complexity grows at a rate of about 58%, the growth rate of the designer productivity is only about 21%, which leads to increasing design costs. This applies especially to the design of analog integrated circuits, whose degree of automation lags behind the one of digital circuits.

According to [Hor05], about 75% of all chips include an analog part. Even though the analog circuit part only comprises about 20% of the chip area, the design effort of the analog part is about 40% of the overall design effort. In [Rut10], it is claimed that 66% of all chips are mixed-signal chips, and that the analog circuit part comprises about 25% of the chip area but requires even 80% of the design effort. Hence, analog components often are the bottleneck in the design flow.

Some examples for analog circuits are operational amplifiers, low noise amplifiers (LNAs), mixers, loop filters or oscillators. Circuits like phase-locked loops (PLLs), digital-analog converters (DACs), analog-digital converters (ADCs) are mixed-signal circuits, i.e., they contain both an analog part as well as a digital part. Analog components can provide clock generation or pad driving, for instance.

1.1 Problem Description

While a logical function can directly be transformed into a digital circuit, an analog circuit can not be derived automatically from the given specifications. Specifications are minimum or maximum values of the circuit's performances like, for instance, open-loop gain, transit frequency or phase margin. Additionally, physical effects like variations of operating conditions, process variations, matching constraints, or noise have to be incorporated into analog design. Analog circuit design is a complicated process which

for the most part is still done by hand, hence the high design effort of 40% as mentioned above.

The design process of analog circuits can be divided into three main steps:

1. Topology Selection
   In this first step, the structure of the circuit is chosen. Transistors, resistors and other circuit elements are connected to one another such that the demanded specifications can be fulfilled after circuit sizing. Moreover, the available supply voltage has to be taken into account, i.e., if too many circuit elements are stacked vertically (between supply voltage and ground), a higher supply voltage is required to supply all these circuit elements. Hence, topology selection is dependent on the transistor technology that is used. Both CMOS transistors with lower threshold voltage and more restrictive requirements on the available supply voltage necessarily lead to new topologies.
   Topology selection is hard to automate since it requires deep knowledge of, for instance, analog devices and circuit design. Most approaches, like [DG98], [MV01], or [ST02], use statistical methods or genetic algorithms to randomly generate circuits. Genetic operations such as selection or mutations are applied and a fitness analysis is performed for every circuit. These approaches are time consuming and meeting the specifications is very hard to achieve. Hence, topology selection is usually still done completely manually.
   Even though this thesis mainly focuses on structural analysis of analog integrated circuits instead of synthesis, the preparatory steps described in chapter 4.3 can be used as a foundation for structural optimisation. That means, the initial topology selection would still be executed by the designer but during the optimisation process, topological variations could be included. In Chapter 5.4, the first results of optimisation with discrete parameters and an outlook to structural optimisation are presented.
2. Circuit Sizing
   Having chosen a topology, a circuit does not necessarily fulfil the demanded specifications. And even if the circuit performs well in the nominal case, it might fail under unfavourable operating conditions. Fulfilment of the specifications as well as robustness towards variations of operating conditions and manufacturing tolerances is not guaranteed with the use of a certain topology. Instead, all circuit elements have to be sized properly. Sizing means adjusting transistor geometries or values of, for instance, resistors or capacitors. Replacing single circuit elements or compounds of circuit elements with others to obtain a better design is somewhere between topology selection and circuit sizing. This approach is discussed in Chapters 4.3 and 5.4.
   The first step of circuit sizing is nominal optimisation. During this sub-step, the circuit is sized such that all specifications are fulfilled if variations are not considered. In a second sub-step, the sizing process continues until a design point is found where all specifications are met when variations or manufacturing tolerances are considered. This sub-step is called design centring.

3. Layout Generation
   After the circuit has been designed and sized, the chip layout for the printed circuit board has to be realised. Some optimisation goals for the layout are to minimise area, wire length, and heat generation.

This thesis focusses on circuit sizing. Though automatic circuit sizing is developed quite well, a major obstacle in practice is the often incomplete circuit specification. Specifying circuit performance bounds, e.g., for DC gain, transit frequency, slew rate, power supply rejection ratio or phase margin of an operational amplifier is not sufficient to prevent mathematical optimisers from driving the circuit into technically meaningless regions. Often, the resulting circuit performs regularly in the nominal case, but exhibits increased sensitivity to process and operating variations and to noise [MCR00].

To avoid this, additional constraints in the form of inequalities to restrain the design space or in the form of equalities to reduce the number of degrees of freedom can be formalised. Most of these constraints – so-called sizing rules – refer to dimensions of and voltages at circuit elements. In this thesis, sizing rules are defined for transistors only, since operational amplifiers usually consist mainly of transistors. But such rules could be formulated for other circuit elements like capacitors as well. Examples for sizing rules are the demand for pairs or larger groups of transistors to have equal dimensions, or inequalities to ensure that a CMOS transistor does not leave the saturation region. Experimental results show that most circuits perform almost linearly when sizing rules are fulfilled (see Chapter 5). When sizing rules are violated however, circuits may behave strongly non-linearly which affects their performance.

An additional advantage of considering sizing rules is the speed-up of the optimisation process. A single transient simulation of an analog circuit may take several seconds or even minutes. During optimisation, many simulations have to be performed. Hence, the optimisation process may take days to complete. Sizing rules help to reduce the time effort of circuit optimisation. When the size of the design space is restrained, an optimiser does not have to search the whole design space to find a solution. Instead, only a small area, where – on top of it all – the circuit is more robust, has to be taken into account. With fewer degrees of freedom due to sizing rules formulated as equalities, the number of parameters is reduced which also speeds up the optimisation process.

The aim of this thesis is to identify sizing rules for CMOS and bipolar transistors and basic compounds consisting of two or more transistors, and to develop a reliable structure recognition method to identify these basic compounds in any analog circuit that is given as a flat netlist.

Section 1.2 focusses both on work about circuit synthesis that also mentions additional constraints required for circuit sizing and on other approaches for structure recognition in analog integrated circuits. In Section 1.3, the objectives of this work are presented in more detail.

1.2 State of the Art

There have been many approaches to analog synthesis. A number of them mention additional constraints that have to be satisfied during the synthesis process. Some of these approaches are discussed in the following.

In [VDL+01], a whole environment for analog circuit synthesis from topology selection to layout generation is presented. Topology selection is done by first synthesising sub-blocks, and then blocks on higher levels. Sub-blocks are, for instance, operational amplifiers but not smaller blocks like transistor pairs. Circuit topologies are collected from a library. Depending on the desired application, the most relevant topology is chosen. Circuit sizing is then done by equation-based circuit optimisation. This contains DC- and AC-equations. In that publication, design constraints, stability constraints and geometrical requirements are mentioned, without going into further detail.

In [ST02], an approach for automatic synthesis of analog integrated circuits using genetic algorithms and fitness evaluation is proposed. It distinguishes between "hard constraints" and "soft constraints". Hard constraints are performance specifications, such as open-loop gain, gain-bandwidth product, or phase margin. Soft constraints are additional requirements on the fitness evaluation but they also require that, for instance, CMOS transistors are only supposed to operate in the saturation region. This approach also offers a "current-flow analysis" where amongst others the direction of current and the operation region of transistors can be determined. However, there are no constraints for compounds of transistors formulated.

A geometric programming approach is presented in [MV01]. It focuses on determining the minimum voltages at all circuit nodes, such that transistors perform in the saturation region. In addition, a requirement is made on minimum transistor geometries. Compounds of transistors are not taken into account. Solely symmetric alignment of transistors is considered. For instance, if the drain currents of two transistors that form a simple current mirror is the same, both transistors are automatically in saturation, since the driving transistor whose gate and drain are connected is always in saturation if the gate-source voltage is above the transistor's threshold voltage.

In [DG98], a distinction is made between different "kinds of space". The first and largest one is the "solvability space", which is the space defined by the design parameters (see Chapter 2.1). All other "kinds of space" represent a further reduction of the solvability space. The "manufacturable space" constricts the solvability space by the requirement that transistors must have minimum width, length, and area depending on their technology. By requiring that all transistors perform in the correct operating region, the manufacturable space is reduced to the "operationality space". In the "functionality space", all designs that fulfil the design requirements are contained. Finally, in the "applicability space", it is required that all specifications are fulfilled. Additionally, a "robustness space" which includes performance variations is given. Constraints are only defined for single transistors. Compounds of transistors are not considered.

Geometric programming in order to determine values of circuit elements is used in [dMHBL01]. In that publication, a range of different types of constraints is presented. "Dimension constraints" refer to transistor geometries, both in form of equations (e.g., equal transistor widths and lengths in a differential pair) and inequalities (minimum transistor geometries). Additionally, constraints referring to bias, power, signal swing, and the small-signal transfer function are imposed. Finally, some other constraints are introduced that incorporate the dependency of the performances on the values of circuit elements. The dimension constraints incorporate transistor compounds, but they are just given exemplarily. And there is neither a listing of constraints referring to transistor compounds, nor a systematic method to assign these constraints automatically.

Another circuit synthesis approach using evolutionary algorithms is presented in [PKR+00]. It mentions "component constraints" and a few examples such as the requirement for equal widths and lengths in a differential pair are given. But it does not go into further detail.

In [HRC89], a synthesis approach that is based on designer expertise and that uses fundamental equations for current mirrors, differential pairs, etc. is introduced. An operational amplifier is first considered on block level. The blocks are current mirrors or differential pairs, for example. In another step, a decision is made on how the blocks will be realised. For instance, a current mirror could be realised, for instance, as simple current mirror or cascode current mirror. The transistors are then sized using those fundamental equations and incorporating the circuit specifications. Different kinds of blocks an operational amplifier consists of are introduced. However, additional constraints to ensure the proper operation region of transistors are not incorporated.

All these approaches give examples of different kinds of constraints, but none of them goes into detail or presents a listing of generic constraints for analog circuits or components consisting of only few transistors. In addition, it is not mentioned if these constraints are extracted by hand or automated procedures.

Approaches for structure recognition have been presented in numerous publications. In [Rub06] and preceding publications like [Rub03], the "SubIslands" method was introduced, an optimisation-based approach for sub-circuit recognition in digital circuits is presented. It combines a graph labelling algorithm and a method called "graduated assignment technique" to calculate match probabilities for model graphs in an object graph. It presents a non-linear graph optimisation algorithm for structure recognition. The object graph is created from the given circuit to examine, the model graphs are sub-circuits from a library that has to be set up by the user.

FROSTY [YS03] is a program for automatic hierarchy extraction in digital circuits. The circuit netlist is given on transistor level. The library of sub-circuits that appear in the circuit has to be provided by the user. The algorithm consists of two steps combining structural recognition and pattern matching. In the first step, the netlist on transistor level is transformed into a netlist on gate level. During step two, both the user-defined sub-circuit blocks and the gate level netlist are represented as directed graph. Then these user-defined blocks are recognised in the circuit using pattern matching.

In [HO95], a graph coding algorithm based on a technology file is used. In the technology file, the types of available devices, as well as different configurations of sub-circuits are defined. For each sub-circuit, a unique code is generated which is then compared to the codes of the circuit.

In [CS91], a method to automatically produce a layout of analog integrated circuits is presented. A distinction between different types of circuit nodes is given, and some basic analog circuit "primitives" like current mirror or differential pair are included. However, the paper neither goes into further detail on how the recognition of these "primitives" is being carried out nor deals with sizing.

These approaches go more or less into detail concerning the recognition algorithm itself. All of them mention a library of sub-circuits, but only one of them actually shows some of the library elements. The others leave it to the user to define a library of sub-circuits. None of them defines a library of generic transistor compounds.

The term "sizing rules" meaning constraints that have to be satisfied for analog circuits was first mentioned in [EGG98], where a hierarchical characterisation of analog circuits was presented. No sizing rules were stated explicitly, but the number of sizing rules for some basic transistor pairs like the simple current mirror or the differential pair were given.

The sizing rules for a simple current mirror have first been stated in [ZEG98]. In [Eck98], a more detailed overview of important CMOS transistor pairs and their sizing rules is given.

In [ZEG99], the sizing rules for a differential pair are stated exemplarily. Additionally, one section focusses on the automatic identification of all sizing rules for analog circuits. In this context, it is mentioned that transistor pairs are the most important constituents of analog circuits.

In [AEG+00], the benefit of sizing rules for simulation-based automatic analog synthesis is mentioned. The number of sizing rules for a few elemental transistor compounds are given as well.

The first publication presenting the automatic construction of sizing rules for CMOS transistor circuits in more detail was [GZEA01]. In that article, sizing rules for CMOS transistors and a library of compounds – so-called building blocks – of two or more CMOS transistors were presented. Additionally, a search procedure was proposed that was able to recognise all given building blocks available in the presented library. That algorithm detected the given building blocks "bottom-up", i.e., it started with transistor-pair blocks, moving on to larger blocks. After the search had been performed, sizing rules were assigned "top-down" to the detected blocks.

In [Ziz01], sizing rules for ten CMOS transistor compounds are presented in detail. In addition, more of the underlying theory on the representation of a netlist and how to recognise pairs of transistors or compounds is given.

1.3 Objectives of the Work

With an exhaustive library of basic compounds for analog circuit design and a list of applicable sizing rules for each compound as well as for single transistors, the process of circuit design can be accelerated and the resulting designs will be more robust to variations. In [GZEA01], a library of ten CMOS transistor compounds was introduced.

In this thesis, the library is extended by further compounds. Additionally, a new library for bipolar transistor compounds with respective sizing rules is set up and combined with the library of CMOS transistor compounds.

The recognition algorithm in [GZEA01] is able to recognise ten CMOS transistor compounds consisting of pairs of transistors or pairs of compounds. Bipolar transistor compounds are not included in the library. More importantly, a problem that was not mentioned in [GZEA01] is the large number of ambiguities that can occur during structure recognition. If, for instance, a transistor has been recognised as part of more than one compound, usually a decision between alternative compounds has to be made. Otherwise, sizing rules for all compounds would be assigned to that transistor. If some of these rules exclude one another, a solution where all sizing rules are fulfilled cannot be found. Hence, a posteriori correction of the recognition result by the designer is required.

A new algorithm that reliably recognises both CMOS and bipolar transistor compounds defined in the new library is developed in this thesis. With this new algorithm, the foundation for structural optimisation is laid as well (see Chapter 4.3). In the following sub-sections, the objectives of this work are described in further detail.

1.3.1 Development of a new Library of both CMOS and Bipolar Transistor Compounds

The work in [GZEA01] already contains a library of ten compounds of CMOS transistors plus banks and the single transistor in saturation or triode region. Sizing rules have been derived for all library elements. In this dissertation, a new library is developed based on the one in [GZEA01]. For the new library, additional compounds of CMOS transistors have been added. Like the library in [GZEA01], all compounds in the new library except single transistors are pairs of other library elements. But banks of compounds are no longer treated as pairs, but as special compounds. Furthermore, a library for bipolar transistor compounds is set up. Since most of the bipolar compounds have the same structure as the CMOS compounds, the two libraries are united to a single library of basic compounds – so-called modules – for analog circuit design. This has several advantages. Using a single library makes it possible to recognise hybrid modules consisting of CMOS and bipolar transistors. Moreover, the recognition algorithm will be simplified, such that CMOS and bipolar modules can be detected in one go. Finally, by developing a single library of CMOS and bipolar transistor compounds, a clear overview

of the available compounds and the sizing rules that have to be assigned to them is given.

1.3.2 Development of a new Structure Recognition Algorithm

In this thesis, the problem of structure recognition based on recognition of pairs is formulated using discrete mathematics and UML (unified modelling language) [Gro99]. This helps in developing an efficient structure recognition algorithm and in providing a solution for the fundamental drawback of the method presented in [GZEA01], as previously mentioned: the many ambiguities in assigning transistors to pairs and higher-order groups of transistors.

For this reason, a new recognition algorithm and a novel heuristic methodology for arbitration of assignment ambiguities is developed. Now, all possible compounds are recognised at first. After that, the recognition result is revised and some of the compounds are removed with the help of a so-called domination relation (see Chapter 4.2).

The list of sizing rules is stored in a way such that it can be read by an automatic optimisation tool like, e.g., Wicked [Mun09].

1.3.3 Preparation of Circuit Netlists

Before the structure recognition algorithm can actually be applied on a netlist, it has to be prepared for the recognition first. A past solution that was used in [Ziz01], was to convert the given netlist into a pseudo netlist by discarding all information that exceeded the connectivity of the circuit elements. Unfortunately, this could not handle all netlists and it only worked with few types of CMOS transistors. Additionally, netlists often include additional circuit elements like power-down transistors that will either be shorted or opened during operation or resistors connected to transistors. These additional circuit elements distort the recognition result since the recognition algorithm only takes the connectivity of circuit elements into account. As a consequence, the pseudo netlist or the recognition result had to be edited manually which was error-prone.

Hence, to fully automate the process of structure recognition, it is necessary to provide an automatic method to prepare the netlist before. This is shown in detail in Chapter 4.3.

This preparation of the netlist is also the foundation for discrete and structural optimisation of analog circuits, since it enables the replacement of circuit elements or even compounds with others during the optimisation process.

Six publications arose from this work. In [MSG03], assignment ambiguities during structural recognition are mentioned for the first time, and a first approach to tackle them was introduced. Sizing rules for bipolar transistor components are introduced in [MGS08a] and [MG08]. In [MGS08b], a new heuristic for arbitration of assignment ambiguities

is introduced and sizing rules for both CMOS and bipolar transistor compounds are summarised. The first approaches for discrete optimisation of analog integrated circuits incorporating exchanging circuit elements in the netlist is presented in [PMGS08a] and [PMGS08b] .

This thesis is organised as follows. In Chapter 2, basic terms and definitions are given. Sizing rules for CMOS and bipolar transistor compounds (modules) are presented in Chapter 3. In Chapter 4, a structure recognition method for the modules presented in Chapter 3 is introduced. Results are presented in Chapter 5.

Chapter 2

Problem Formulation

In this chapter, some fundamental terms and definitions are given.

Circuit sizing means that values of circuit elements are adjusted until the circuit fulfils the demanded specifications. The values of circuit elements are part of the circuit parameters. These parameters are mapped to the performances by performance evaluation. Sizing rules confine the available parameter space to exclude unfavourable parameter sets and to speed up the optimisation process.

In the following, the terms circuit parameters, circuit specifications and sizing rules will be discussed in detail.

2.1 Circuit Parameters

Circuit parameters are transistor dimensions, values of circuit elements like resistors or capacitors, physical parameters like oxide thickness in transistors or parasitic resistances, but also the operating temperature, or the applied supply voltage. To summarise, circuit parameters cover all variables that are present through the circuit elements in a given circuit topology and its environment.

These parameters can be classified into three categories.

Design parameters

These parameters can be adjusted during the design process in order to achieve the specifications of a circuit. Design parameters are, for instance, transistor widths and lengths. The design parameters are collected in vector $\mathbf{d} \in \mathbb{R}^{n_d}$, where n_d is the number of design parameters.

For each design parameter, a lower and an upper bound can be given. These bounds can arise from technological limits or requirements concerning the maximum chip area, for

instance. They can also define the range where the given models for the circuit elements are accurate enough.

With a lower and upper bound for each design parameter, the vector of design parameters is enclosed between the vectors $\mathbf{d}_l$ and $\mathbf{d}_u$ representing the bounds:

$$\mathbf{d}_l \leq \mathbf{d} \leq \mathbf{d}_u, \tag{2.1}$$

with

$$\mathbf{x} \leq \mathbf{y} \Longleftrightarrow \bigvee_{1 \leq i \leq |\mathbf{x}|} x_i \leq y_i \tag{2.2}$$

for arbitrary vectors $\mathbf{x}$ and $\mathbf{y}$ with $|\mathbf{x}| = |\mathbf{y}|$.

Operating parameters

These parameters concern the environment the given circuit is employed in. Supply voltage, operating temperature and load are examples for operating parameters. They are subject to variations and even their nominal values can extensively differ depending on where the circuit is employed. For instance, a cell phone or digital camera is expected to perform well both at $+30°C$ and $-10°C$.

During simulation, bounds for operating parameters are specified according to the operating conditions that can be expected in reality. For instance, common upper and lower bounds for operating temperature are $-40°C$ and $80°C$. The operating parameters are collected in vector $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$. This vector is enclosed between vectors $\boldsymbol{\theta}_l$ and $\boldsymbol{\theta}_u$ representing lower and upper bounds:

$$\boldsymbol{\theta}_l \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}_u. \tag{2.3}$$

Statistical parameters

These parameters refer to parameters that are subject to variations that occur during the manufacturing process. This means that if a certain value, e.g., for oxide thickness, is specified for a transistor model, the oxide thickness of all transistors of that model type will more or less differ from the specified value, since the manufacturing process is not 100% accurate.

Statistical parameters are collected in vector $\mathbf{s} \in \mathbb{R}^{n_s}$. Statistical parameters s_k with $1 \leq k \leq n_s$ have a Gaussian distribution with mean value $s_{k,0}$ and standard deviation σ_k, or they can be transformed into a Gaussian distribution.

The probability density function for one statistical parameter s_k is given by:

$$pdf(s_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \cdot \exp\left(-\frac{(s_k - s_{k,0})^2}{2\sigma_k^2}\right) \tag{2.4}$$

For an n-dimensional Gaussian distribution, the probability density function is given by:

$$pdf(\mathbf{s}) = \frac{1}{\sqrt{(2\pi)^{n_s} \cdot \det \boldsymbol{\Sigma}}} \cdot \exp\left(-\frac{1}{2}(\mathbf{s} - \mathbf{s_0})^T \boldsymbol{\Sigma}^{-1} (\mathbf{s} - \mathbf{s_0})\right) \tag{2.5}$$

with the covariance matrix $\mathbf{\Sigma}$.

All circuit parameters are collected in vector $\mathbf{p}$ with

$$\mathbf{p} = \begin{pmatrix} \mathbf{d} \\ \boldsymbol{\theta} \\ \mathbf{s} \end{pmatrix}. \tag{2.6}$$

Vector $\mathbf{p}$ describes the parameter space $\mathcal{P}$.

2.2 Circuit Performances

To obtain the performances of a circuit, it is embedded into a test environment – a so-called testbench – and simulated. The testbench is set up to provide different kinds of simulation. To obtain the characteristic curves of an operational amplifier, DC-, AC-, and transient simulation are usually sufficient. The performance values are then extracted from the characteristic curves. For instance, the open-loop gain of an operational amplifier is the absolute value of the frequency response at frequency 0.

The process of simulating a circuit and extracting the performance values is called performance evaluation. Examples of performances of operational amplifiers are phase margin, slew rate, power supply rejection ratio, and transit frequency.

The performances are collected in vector $\mathbf{f}$. By performance evaluation, a single point $\mathbf{p}_i$ in the parameter space is mapped to one point $\mathbf{f}_i$ in the performance space:

$$\mathbf{p}_i \overset{\text{performance evaluation}}{\longmapsto} \mathbf{f}_i. \tag{2.7}$$

The performance space $\mathcal{F}$, i.e., the set of all performance values that are obtained from any possible set of parameters $\mathbf{p}$ by performance evaluation (denoted by "*eval*"), is given by:

$$\mathcal{F} = \{\mathbf{f} \,|\, \mathbf{f} = eval(\mathbf{p})\}. \tag{2.8}$$

The transformation from the whole parameter space to the whole performance space is called performance exploration which can be written as:

$$\mathcal{P} \overset{\text{performance exploration}}{\longmapsto} \mathcal{F}. \tag{2.9}$$

Specifications are minimum or maximum performance values of performances that have to be achieved. Some specifications, e.g., offset, have both a minimum and maximum value. A specification like this can be replaced by two specifications, one for the minimum, and one for the maximum. By alternating the algebraic sign, every minimum value can be transformed into a maximum value and vice versa. Hence, without loss of

generality, the requirement that all performances $\mathbf{f}$ must fulfil all specifications $\mathbf{a}$ can be written as follows:

$$\mathbf{f} \geq \mathbf{a} \tag{2.10}$$

The part of the design space where all specifications are fulfilled is called the acceptance space with:

$$\mathcal{A} = \{\mathbf{f} \,|\, \mathbf{f} \geq \mathbf{a}\} = \mathcal{F} - \{\mathbf{f} \,|\, \mathbf{f} < \mathbf{a}\}. \tag{2.11}$$

2.3 Sizing Rules

In the first place, the function of an analog circuit is determined by its topology. But the correct function of the circuit is not automatically guaranteed. In fact, the actual values of the design parameters play an important role as well.

When designing an analog circuit like an operational amplifier, a designer often determines the currents that are supposed to flow through the circuit elements. If the deviation of these values becomes too large, the circuit will not work any more.

For instance, the drain current i_d in a CMOS transistor is nearly constant as long as the transistor operates in the saturation region. Only the channel length modulation causes a slight increase of i_d with increasing v_{ds}. However, when the transistor leaves the saturation region and enters the triode region, the drain current i_d changes nearly linearly with v_{ds}. This means that even a slight change in v_{ds} leads to a change in i_d. For instance, a current mirror which is supposed to provide a constant current ratio would hardly work as it should if a transistor left the saturation region and its drain current was therefore much more sensitive to variations of its drain-source voltage.

Thus, if a certain constant current is required to flow through a CMOS transistor, it is required to operate in saturation region. Sizing rules provide that purpose, i.e., the optimiser will be prevented from driving that CMOS transistor out of the saturation region.

But even if the circuit fulfils its function and a design parameter set within the acceptance space $\mathcal{A}$ is found in the nominal case, it is not ensured that the circuit is robust to variations of statistical or operating parameters. Sizing rules help to provide a certain robustness to variations. Results show that a circuit performs nearly linearly and is therefore less sensitive to variations when sizing rules are fulfilled (see Chapter 5).

Sizing rules already arise on transistor level, i.e., there are rules for single CMOS and bipolar transistors as well as for compounds of two or more transistors like current mirrors or differential stages. Even for circuits containing less than 100 transistors, often hundreds of sizing rules apply as the results in Chapter 5 show.

	geometrical	electrical
Function	symmetry requirements	ensure correct operating region
Robustness	reduction of influence of process variations	provision of safety margins to avoid mismatch

Table 2.1: Classification of Sizing Rules with Examples of Application

Sizing rules can be differentiated as follows:

Function and Robustness:
Sizing rules concerning function ensure that a circuit operates as it is supposed to do. These sizing rules are therefore crucial considering the function of a circuit. Sizing rules concerning robustness ensure that a circuit still operates as it should do when variations of of operating conditions or statistical parameters or mismatch are considered.

Electrical and geometrical:
Electrical sizing rules refer to voltages and currents* between circuit elements like transistors (e.g., drain-source voltage v_{ds} of a CMOS transistor). Geometrical sizing rules refer to transistor geometries (width, length, area). These can be rules that require the lengths of two CMOS transistors to be equal or rules for minimum transistor geometries, for instance.

Equalities and inequalities:
Equalities appear between transistor geometries, e.g., channel length of CMOS transistors. Sizing rules that force two parameters to be equal reduce the number of design parameters by one. Hence, these sizing rules come into effect before the optimisation process. They lead to a reduction of the dimension of the design space which leads to a faster optimisation process. Inequalities can appear both between transistor geometries and voltages or currents, e.g., the drain-source voltages of two transistors. They further confine the design parameter space that is bounded by $\mathbf{d}_l$ and $\mathbf{d}_u$.

Table 2.1 shows examples for sizing rules, distinguishing geometrical and electrical sizing rules as well as sizing rules for function and robustness.

The evaluation of sizing rules is usually of low cost since it only requires a DC-simulation which has to be performed anyway to obtain the operating point.

* None of the sizing rules presented in Chapter 3 refers to currents, but there could be such rules for other circuits and basic compounds than those presented in this thesis.

Sizing rules are collected in vector $\mathbf{c}(\mathbf{p})$. All sizing rules can be formulated in a way such that they are fulfilled if their value is greater or equal zero. The part of the parameter space where sizing rules are fulfilled is called the feasible parameter space:

$$\mathcal{P}' = \{\mathbf{p} \,|\, \mathbf{c}(\mathbf{p}) \geq 0\}. \tag{2.12}$$

Based on performance evaluation, the feasible parameter space is mapped to the feasible performance space:

$$\mathcal{P}' \overset{\text{performance exploration}}{\longmapsto} \mathcal{F}'. \tag{2.13}$$

with

$$\mathcal{F}' = \{\mathbf{f} \,|\, \mathbf{c}(\mathbf{p}(\mathbf{f})) \geq 0\}, \tag{2.14}$$

where $\mathbf{p}(\mathbf{f})$ is the parameter vector that was mapped to $\mathbf{f}$ by performance evaluation. A design point that is feasible is not necessarily a design point where all specifications are met. The performance space where both all specifications and all sizing rules are met is called the feasible acceptance space $\mathcal{A}'$ and is given by:

$$\mathcal{A}' = \mathcal{A} \cap \mathcal{F}' = \{\mathbf{f} \,|\, \mathbf{f} \geq \mathbf{a} \wedge \mathbf{c}(\mathbf{p}(\mathbf{f})) \geq 0\}. \tag{2.15}$$

In Chapter 1.2, the process of circuit sizing was divided into two sub-steps, nominal optimisation and design centring. When sizing rules are considered, an additional preparatory step is carried out where a design point is searched, where all sizing rules for the given circuit are fulfilled, regardless of the specifications. This additional step is called feasibility optimisation.

Since even in a small circuit the number of sizing rules can be above 100, their automatic generation is necessary. Manual setup is time consuming and error-prone.

In this thesis, the automatic structural analysis of a given circuit to determine the function of its transistors is presented. If a transistor is part of a compound of transistors that fulfils a certain function, the function of that transistor is clear and hence, sizing rules can be applied to that transistor as well as to the other transistors in that compound. Such a compound can consist of as few as two transistors, like for instance a simple current mirror or a differential pair. The function of that compound is then derived from the connection of the two transistors. The sizing rules are stored automatically and can be accessed during the optimisation process.

Chapter 3

Sizing Rules for Basic Analog Building Blocks

In this chapter, sizing rules for basic compounds in analog design will be presented. First, a definition of the most important terms will be given. Afterwards, a library of basic analog compounds – so called modules – consisting of at least one transistor will be introduced. Finally, sizing rules for both CMOS and bipolar transistor modules will be presented.

3.1 Definitions

An electrical circuit consists of physical devices that are connected to one another via nets. A module can be a such a physical device, but a module can also be a compound of at least two physical devices connected to one another that form a sub-circuit whose usage is common in analog design. Figure 3.2 shows a library of such modules for CMOS and bipolar transistors. This library will be discussed in more detail in Chapter 3.2.

Definition 3.1
A "module" is a physical device or a compound of physical devices common in analog design.

The module library in Figure 3.2 is organised in a way that all modules except single physical devices are built-up of pairs of other modules. This leads to the following two additional definitions.

Definition 3.2
Every module that is not built-up of any other modules is called a "single-module", regardless of its function.

A single-module is indivisible, hence, all physical devices like transistors are single-modules.

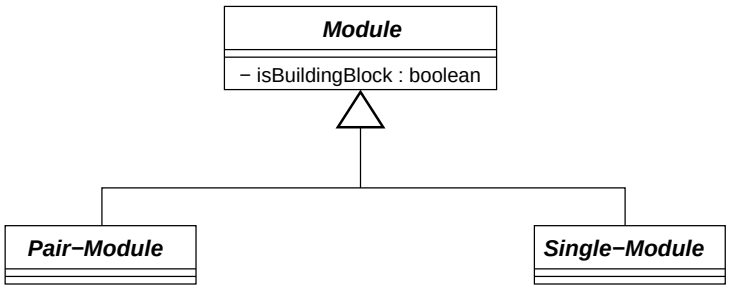

Figure 3.1: Definition of a Module in UML Notation

Definition 3.3
A module that is built-up as a pair of other modules is called a "pair-module".

Figure 3.1 shows a UML (unified modelling language) diagram that illustrates this context.

If a module fulfils a certain function, it is also called a building block. For this purpose, a module has got a property `isBuildingBlock` which can be true or false. For instance, a CMOS transistor that has to operate in saturation is not just a module, it is also a building block.

Definition 3.4
A building block is a module that fulfils a designated function.

The basic modules introduced in the next section allow a simple way to collect all sizing rules for a circuit. In the following, a library of modules in CMOS and bipolar technology will be presented. Sizing rules for these modules will be presented in Chapters 3.3 and 3.4.

3.2 Hierarchical Module Library

Figure 3.2 presents the hierarchical library L of available modules in CMOS and bipolar transistor technology. It is a hierarchical library because it can be divided into hierarchy levels with sub-libraries L_i with

$$L = \bigcup_{i=0}^{h_L} L_i, \tag{3.1}$$

where h_L is the number of hierarchy levels, which is three in the library in Figure 3.2. Modules on higher hierarchy levels consist of modules from lower hierarchy levels. The hierarchy levels are denoted and separated from one another in Figure 3.2. A hierarchical library represents a strictly ordered set. The strictly ordered set $(L, \in)$ describes the character of L, i.e., the strict order is characterised by the $\in$-relation. That means that all elements in L on hierarchy level 1 and above consist of other elements in L.

The upper or left module in a module is sub-module 1, the other module is sub-module 2. On hierarchy level 1, the transistors the modules consist of are marked with (1) or (2) respectively. These indices will be used in the following chapters to refer to one of the modules another module consists of.

A condition for all elements in L except the single transistor which is indivisible is that they must consist of exactly two other elements (sub-modules), i.e.,

$$\underset{i \in \mathbb{N}_0, l \in L}{\forall} ite\left(i = 0, d^-(l) = 0, d^-(l) = 2\right) \tag{3.2}$$

The *ite*-function is the "*i*f-*t*hen-*e*lse"-function. The operator $d^-(l)$ denotes the predecessor degree of l, i.e., the number of elements l consists of. This means that all modules on hierarchy level 0 do not consist of any sub-modules, i.e., their predecessor degree is 0. If $d^-(l) = 2$, which has to be true for all modules above hierarchy level 0, l has to consist of exactly two sub-modules, either of two different ones or two of the same type. For instance, a simple current mirror consists of two transistors, a cascode current mirror consists of a simple current mirror and a level shifter. All modules marked with an asterisk are building blocks, i.e., they deliver sizing rules on their own. The others deliver sizing rules as part of larger building blocks.

The modules in Figure 3.2 are given for NMOS and npn transistors, but they exist for PMOS and pnp transistors as well. Some modules are only shown with CMOS transistors, some only with bipolar transistors. In this case, these modules are only used in the respective technology or there are no building blocks that consist of these modules in the respective technology. For instance, a cascode pair also exists with bipolar transistors. But since a cascode pair is not a building block, and there is no bipolar building block in L that contains a cascode pair, it is not included in L.

Hierarchy level 0 contains the single CMOS or bipolar transistor. Level 1 contains modules that consist of two transistors each, like the simple current mirror, differential pair or Darlington configurations. The building blocks on level 2 consist either of two modules from hierarchy level 1 or a module from level 1 and one from level 0. For instance, a wide swing cascode current mirror consists of two modules from hierarchy level 1: a voltage reference II and a cascode pair. A Wilson current mirror however, consists of a single transistor from hierarchy level 0 and a simple current mirror from level 1. Level 3 contains the differential stage which consists of a differential pair together with an arbitrary current mirror from level 1 or 2. A differential stage can consist of bipolar and CMOS transistors together (see Chapter 3.5). Actually, there are different types of differential stages, depending on which type of current mirror they contain. Hence, one type of differential stage – namely the one consisting of a simple current mirror and a differential pair – should appear on hierarchy level 2. But due to the generalisation that a differential stage consists of a differential pair and an arbitrary current mirror, the differential stage is listed only once and located one level above the highest hierarchy level that contains a type of current mirror.

In bipolar transistor circuits, resistors are often used (e.g., at the emitter pins of a differential pair). Since these resistors do not change the basic structure of the modules,

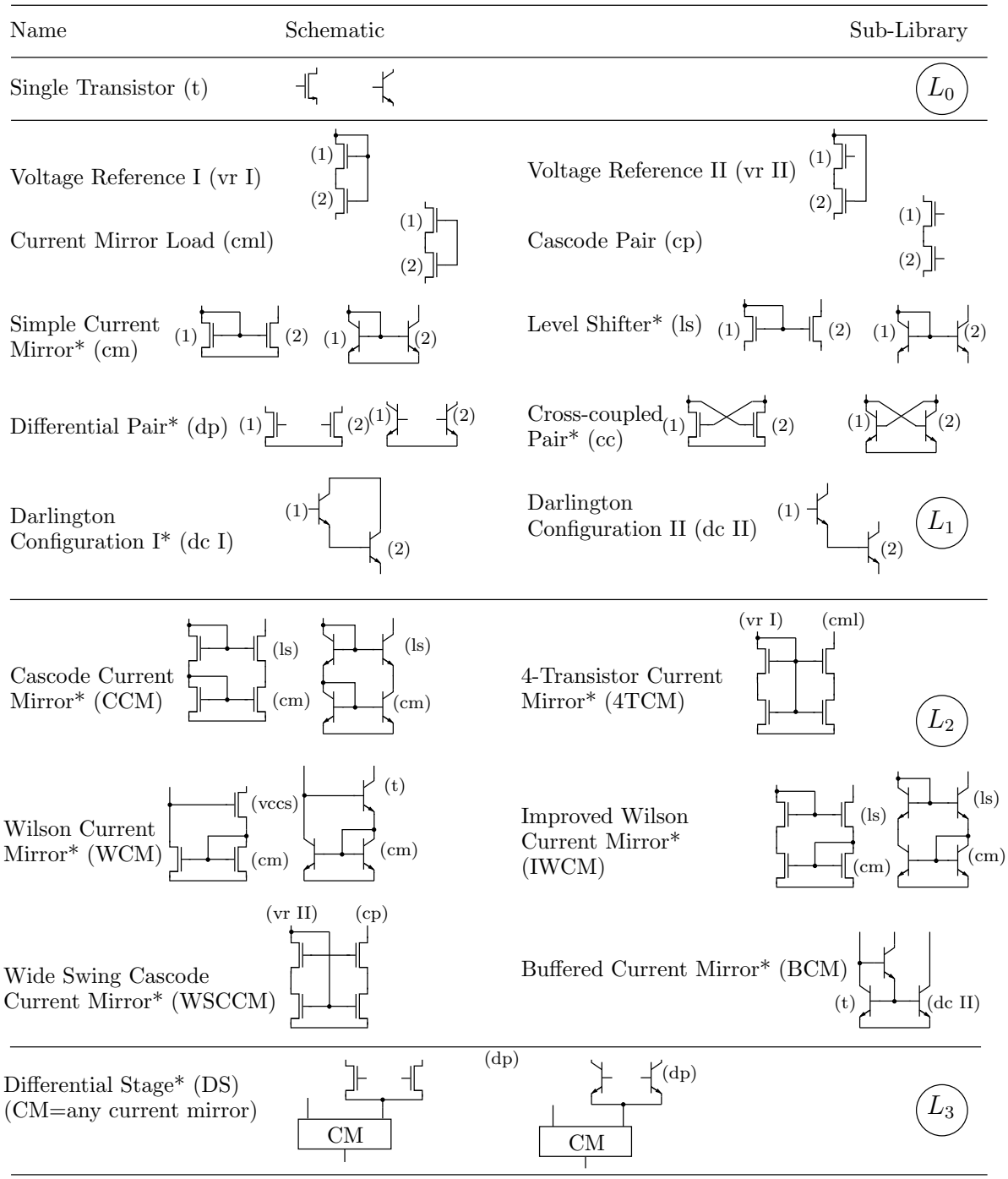

Figure 3.2: Hierarchical Library L of Available Modules (NMOS, npn)

all modules are only shown in their basic form without resistors. In Chapter 3.4, some modules with resistors will be included and discussed in further detail. To be able to recognise modules, the transistor and the resistor are treated like a single circuit element (see Chapter 4.3).

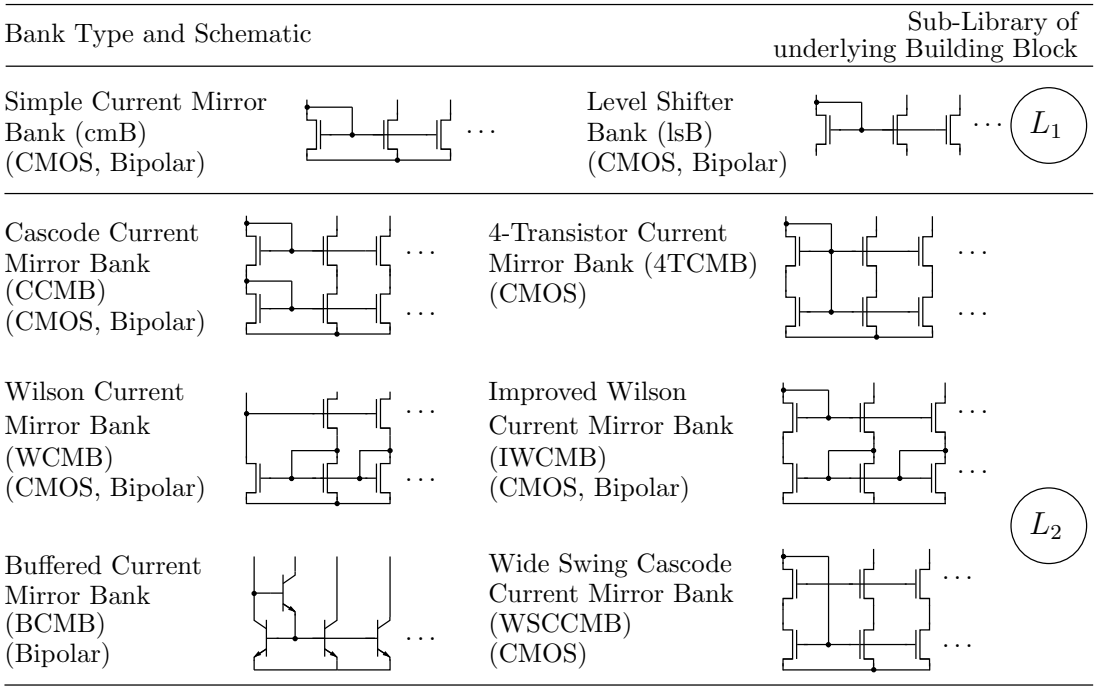

Figure 3.3: Available Banks based on the Modules in Figure 3.2

Considering intra- and inter-connections, there are $B_6 = 203$ possible transistor pair structures, where B_k is the Bell number [Mat93]. The term intra-connection refers to pins of the same transistor being connected, e.g., drain and gate of one CMOS transistor, while the term inter-connection refers to pins of different transistors being connected, e.g., the source pins of two transistors that form a differential pair. The number six in B_6 is the number of pins that two transistors together have. The bell number describes the number of partitions (non-empty subsets) a set can be split into. Hence, the number 203 includes all cases from none of the six pins being connected to all six pins being connected to one another. Most of these 203 possible combinations are technically irrelevant.

On hierarchy level 1, the library in Figure 3.2 contains solely transistor pairs that deliver sizing rules on their own or are contained in building blocks on higher hierarchy levels. In this sense, the list of transistor pairs on level 1 is complete. For hierarchy levels above 1, this library is not complete and a variety of other building blocks could be included. But it represents a majority of typical building blocks and can be considered a standard building block library.

Some of the modules in Figure 3.2 are "bankable", which means that these modules can be combined in a way that they form a bank. Figure 3.3 shows all available banks based on the modules in Figure 3.2. A bank contains at least two modules that share the same driving stage. For banks consisting of transistor pairs, the driving stage is a single transistor that all these pairs have in common, like for instance, the driving transistor of a number of simple current mirrors. In most cases, the driving stage of a bank is

sub-module 1 of the modules the bank contains. One exception is the cascode current mirror, which is because the two transistors on the left do not form a pair-module that is defined in library L. Banks do not produce any additional sizing rules since all rules already result from the modules contained in the banks. But they are included for completeness and they could be useful in future work, for instance, when symmetry constraints are taken into account.

Figure 3.4 shows the library L and the definition of banks in UML notation. It includes the definition of a module, pair-module, and single-module given in Figure 3.1. Figure 3.4 shows all available modules from the transistor as a single-module up to the differential stage and the bank-instance which contains at least two pair-modules.

For a full circuit representation, further circuit elements like resistors, capacitors, voltage and current sources etc. would have to be included. But for the representation of the modules presented in Figure 3.2, these circuit elements do not have to be included in Figure 3.4.

Every module has got an attribute `isBuildingBlock` which is true for those modules that are also building blocks (see Def. 3.4). Every module has also got an attribute `type`, which is a tuple (`transtype, structype`), where `transtype` refers to the type of the transistors the module consists of, and `structype` refers to the structural type. The set of all types Y is defined as

$$Y \subseteq YT \times YS \text{ and } y = (t, s) \text{ with } y \in Y. \tag{3.3}$$

The set YT is the set of all transistor types:

$$YT = \{\text{NMOS, PMOS, npn, pnp, NMOS\_npn, PMOS\_pnp, npn\_NMOS, pnp\_PMOS}\} \tag{3.4}$$

The latter four elements of YT are hybrid types. For instance, `NMOS_npn` means that sub-module 1 of a module consists of NMOS transistors, while sub-module 2 consists of npn transistors. The set YS is the set of all structural types that can be found in Figure 3.2:

$$YS = \{\text{transistor, vr I, vr II, } \ldots, \text{ DS}\}. \tag{3.5}$$

In addition, all modules have got a constant attribute `allowedTypes`, which is a subset of YT containing all transistor types that are valid for the respective module according to Figure 3.2. For instance, for a voltage reference I, `allowedTypes` $= \{NMOS, PMOS\}$. For a differential stage however, `allowedTypes` $= YT$. Which types are allowed for the different modules is evident from Figure 3.2.

All pair-modules have got an attribute `isBankable`. It is set to true for only those pair-modules that can be formed a bank out of. Additionally, these pair-modules have got an attribute `drivingStage` which is a set of modules that contains the sub-modules that are shared by all members of the same bank. For most pair-modules, this set contains only one element, i.e. sub-module 1 of the pair-module.

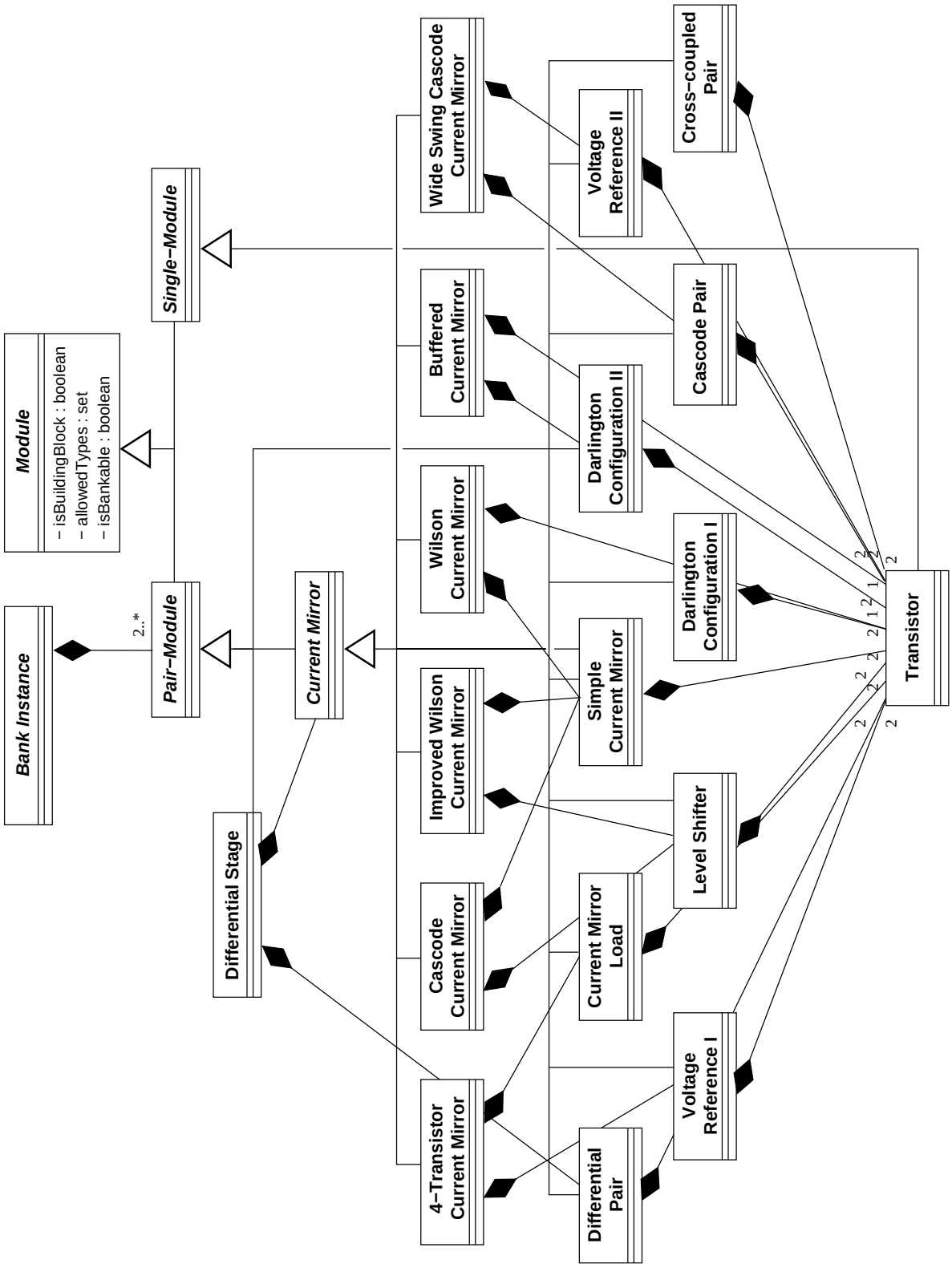

Figure 3.4: Hierarchical Library L plus Banks in UML Notation

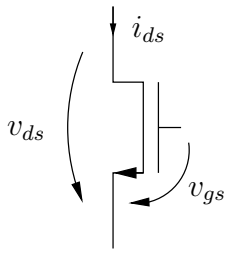

Figure 3.5: NMOS Transistor

3.3 Sizing Rules for CMOS Transistor Building Blocks

In this chapter, sizing rules for all basic CMOS building blocks in Figure 3.2 are presented. All rules will be presented for NMOS building blocks but can be formulated accordingly for their PMOS counterparts.

All constants that appear in the following sub-chapters are technology-specific and have to be determined only once for each technology.

A CMOS transistor's (Figure 3.5) behaviour can be described using the Shichman-Hodges model [SH68]. The three operating region of a CMOS transistor are defined as follows:

Cut-off region: $v_{gs} \leq 0$

Triode region: $0 \leq v_{ds} < v_{gs} - V_{th}$

Saturation region: $v_{gs} - V_{th} \leq v_{ds}$

The transistor's drain current is given by

$$i_d = \begin{cases} 0, & \text{if } v_{gs} \leq 0 \\ K\frac{w}{l}[(v_{gs} - V_{th}) - \frac{v_{ds}}{2}] \cdot v_{ds}\left(1 + \frac{\lambda}{l}v_{ds}\right), & \text{if } 0 \leq v_{ds} < v_{gs} - V_{th} \\ \frac{1}{2}K\frac{w}{l}\left(v_{gs} - V_{th}\right)^2 \cdot \left(1 + \frac{\lambda}{l}v_{ds}\right), & \text{if } v_{gs} - V_{th} \leq v_{ds} \end{cases} \tag{3.6}$$

Here, w and l are the transistor's width and length, $K = \mu_{Si}C_{ox}$ with μ_{Si} being the electron mobility and C_{ox} the oxide capacity, V_{th} is the threshold voltage, and λ the channel length modulation coefficient. The gate current is very small and is therefore neglected. Hence, the source current is approximately equal to the drain current. The design parameters are the transistor geometries w and l.

Figure 3.6 shows the curve family of an NMOS transistor. It shows the drain current i_d against the drain-source voltage v_{ds} for different values of the gate-source voltage v_{gs}. For $v_{gs} < V_{th}$, the transistor is in weak inversion, i.e., there is just some sub-threshold leakage [Bin08].
For higher values of v_{gs}, the transistor is either in the triode or saturation region. The

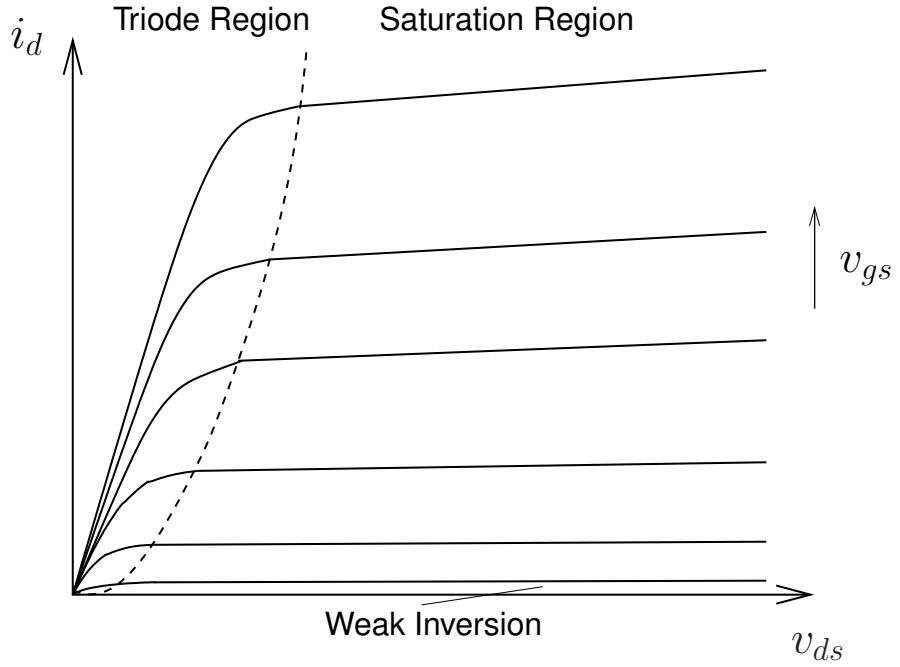

Figure 3.6: Output Curve Family of an NMOS Transistor

dashed line indicates the boundary between triode and saturation region. In the triode region, the transistor behaves like a voltage-controlled resistor (vcr). In the saturation region, the transistor behaves like a voltage-controlled current source (vccs). Here, i_d changes only slightly with v_{ds}. The slope of i_d is equal to $1/(\lambda/l)$. Hence, without the influence of channel length modulation, i_d would be constant for constant v_{gs}. In analog circuits, a transistor is mostly employed in the saturation region. Sometimes, transistors are required to operate in triode region, though. This leads to different sizing rules for the CMOS transistor, depending on the region in which it is supposed to operate. The building blocks it is part of according to the recognition result determine the operation region of the transistor.

3.3.1 Building Blocks on Hierarchy Level 0

3.3.1.1 CMOS transistor as Voltage Controlled Current Source (vccs)

A transistor working as voltage controlled current source operates in saturation. From the equation for i_d in (3.6), it follows that v_{ds} must be greater than $v_{gs}-V_{th}$ (3.8). The distance from the boundary to the triode region is adjusted with the constant $V_{sat_{min}}$. To ensure strong inversion, v_{gs} must be greater than V_{th} (3.10). Unless $V_{sat_{min}}$ is negative, (3.10) is redundant. If not, (3.9) assures that v_{ds} is positive. Furthermore, from [SH68], [LHC86], and [PDW89], it follows that the drain-source current variation depends on variations of channel width and length, threshold voltage, electron mobility and specific gate oxide capacitance with factors $1/w^2$, $1/l^2$ and $1/(w \cdot l)$. Additionally,

	geometric	electric
Function	———	$v_{ds} - (v_{gs} - V_{th}) \geq V_{sat_{min}}$ (3.8) $v_{ds} \geq 0$ (3.9) $v_{gs} - V_{th} \geq 0$ (3.10)
Robustness	$w \cdot l \geq A_{min_{sat}}$ (3.11) $w \geq W_{min_{sat}}$ (3.12) $l \geq L_{min_{sat}}$ (3.13)	———

Table 3.1: Sizing Rules for an NMOS Transistor as Voltage Controlled Current Source

	geometric	electric
Function	———	$(v_{gs} - V_{th}) - v_{ds} \geq V_{lin_{min}}$ (3.14) $v_{ds} \geq 0$ (3.15) $v_{gs} - V_{th} \geq 0$ (3.16)
Robustness	———	———

Table 3.2: Sizing Rules for an NMOS Transistor as Voltage Controlled Resistor

$1/f$ noise is also proportional to $1/(w \cdot l)$. In [LHC86], the variations of V_{th}, μ_{Si}, and C_{ox} were derived. Based on these, in [Ziz01], the variation of the drain current was derived as

$$\frac{\sigma_{i_d}^2}{i_d^2} = \frac{\sigma_W^2}{W^2} + \frac{1 + \frac{2\lambda v_{ds}}{L}}{1 + \frac{\lambda v_{ds}}{L}} \cdot \frac{\sigma_L^2}{L^2} + \frac{4}{(v_{gs} - V_{th})^2 \cdot \frac{A_{V_{th}}}{W \cdot L}} + \frac{A_{\mu_{Si}}}{W \cdot L} + \frac{A_{\mu_{C_{ox}}}}{W \cdot L} \tag{3.7}$$

Hence, for robustness, certain values for width, length and area are required which are sufficiently larger than L_{min} and W_{min} defined for the technology that is used ((3.11)–(3.13)).

The sizing rules for a transistor in saturation are summarised in Table 3.1.

3.3.1.2 Voltage Controlled Resistor (vcres)

A transistor as a voltage controlled resistor operates in the linear region. In this case, $v_{gs} - V_{th}$ has to be larger than v_{ds} (3.14). The distance from the boundary to the saturation region is adjusted with the constant $V_{lin_{min}}$. A larger value of $V_{lin_{min}}$ is

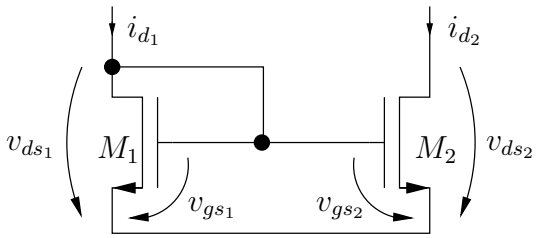

Figure 3.7: NMOS Simple Current Mirror

	geometric	electric
Function	$l_1 = l_2$ (3.17)	$\lvert v_{ds_2} - v_{ds_1} \rvert \leq \Delta V_{ds_{max(cm)}}$ (3.18)
Robustness	———	$v_{gs_{1,2}} - V_{th_{1,2}} \geq V_{gs_{min}}$ (3.19)
Additionally, the rules in Table 3.1 must be fulfilled for both transistors.		

Table 3.3: Sizing Rules for an NMOS Simple Current Mirror

required if the transistor is supposed to operate in the deep ohmic region. Strong inversion is assured through (3.15) and (3.16). The sizing rules for a transistor operating as voltage controlled resistor are summarised in Table 3.2.

3.3.2 Building Blocks on Hierarchy Level 1

3.3.2.1 Simple Current Mirror (cm)

The function of a simple current mirror (Figure 3.7) is to produce a constant ratio between the drain currents of the two transistors. Since the gate currents are assumed to be zero, the drain currents of the two transistors are equal if the transistors are identical and the drain-source voltages are equal. From (3.6), it follows that if both transistors operate in saturation, the ratio between the drain currents of the two transistors M_1 and M_2 is given by:

$$\frac{i_{d_2}}{i_{d_1}} = \frac{K_2(w_2/l_2)}{K_1(w_1/l_1)} \cdot \frac{(v_{gs_2} - V_{th_2})^2}{(v_{gs_1} - V_{th_1})^2} \cdot \frac{1 + \frac{\lambda_2}{l_2} v_{ds_2}}{1 + \frac{\lambda_1}{l_1} v_{ds_1}} \tag{3.20}$$

Both transistors have to operate in saturation because the drain current i_d is only weakly dependent on v_{ds} in saturation region. From the last fraction of (3.20), it can be seen that both transistors have to have the same length and that the channel modulation coefficients λ_1 and λ_2 should be equal. The requirement for equal lengths can easily

be fulfilled by choosing two transistors with the same length. However, due to manufacturing tolerances, two transistors are never absolutely identical. Thus, the channel modulation coefficients aren't equal either. Hence, to reduce the influence of channel length modulation, it has to be assured that the two transistors have equal length in the nominal case (3.17) and that the difference between the drain-source voltages at the two transistors is small (3.18). It can also be seen that greater lengths generally reduce the influence of the drain-source voltages. More requirements to the process of manufacture arise from the dependency of i_{d_2}/i_{d_1} on the parameters $K = \mu_{Si}C_{ox}$ and V_{th}.
The gate-source voltages of both transistors are equal due to the connection of both their gate and their drain pins. But to avoid mismatch due to local process variations, the effective gate-source voltage has to be sufficiently large.

If both transistors are identical and their lengths are set equal, the ratio between the drain-current simplifies to

$$\frac{i_{d_2}}{i_{d_1}} = \frac{w_2}{w_1} \tag{3.21}$$

In some applications, e.g., high-frequency applications, it is recommended to use transistors with identical widths. In this case, the current ratio can be adjusted by the ratio of the number of transistors in parallel on each side of the current mirror. The sizing rules for a simple current mirror are summarised in Table 3.3.

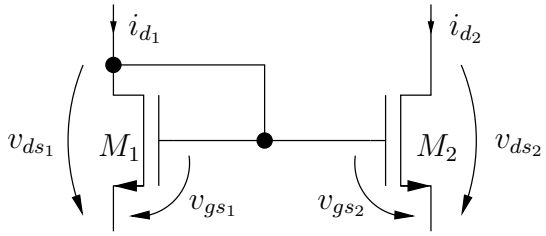

Figure 3.8: NMOS Level Shifter

	geometric	electric
Function	$l_1 = l_2$ (3.22)	———
Robustness	———	$v_{gs_{1,2}} - V_{th_{1,2}} \geq V_{gs_{min}}$ (3.23)
Additionally, the rules in Table 3.1 must be fulfilled for both transistors.		

Table 3.4: Sizing Rules for an NMOS Level Shifter

3.3.2.2 Level Shifter (ls)

The function of a level shifter is to produce a constant voltage difference. Basically, the level shifting function can be processed using a single transistor, e.g., a source follower or a transistor with its drain and gate terminal connected (diode connection). The function of the building block in Figure 3.8 is either to provide a constant differential voltage between or equal voltages at the source pins of the two transistors. This depends on the application of this building block. Some of the current mirrors on hierarchy level 2 include a level shifter to form a cascode pair which assures equal drain-source voltages at the other transistors. This building block is often also called impedance converter.

Both transistors operate in saturation and their lengths are set equal to reduce the dependency of i_d on v_{ds}. To reduce the influence of threshold voltage mismatch, $v_{gs} - V_{th}$ must be sufficiently large. A difference in the drain-source voltages of the two transistors is only of negligible influence on the performance of this building block as long as both transistors operate in saturation.
The sizing rules for a level shifter are summarised in Table 3.4.

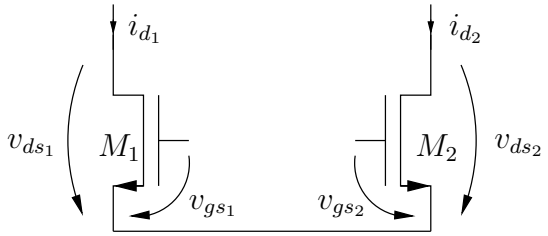

Figure 3.9: NMOS Differential Pair

	geometric	electric
Function	$l_1 = l_2$ (3.25) $w_1 = w_2$ (3.26)	$\lvert v_{ds_2} - v_{ds_1} \rvert \leq \Delta V_{ds_{max(cm)}}$ (3.27)
Robustness	———	$\lvert v_{gs_2} - v_{gs_1} \rvert \leq \Delta V_{gs_{max}}$ (3.28)
Additionally, the rules in Table 3.1 must be fulfilled for both transistors.		

Table 3.5: Sizing Rules for an NMOS Differential Pair

3.3.2.3 Differential Pair (dp)

A differential pair (Figure 3.9) transforms a difference between the gate-source voltages of the two transistors into a difference between their drain currents. If both transistors operate in the saturation region, the difference between the drain currents is given by:

$$\Delta i_d = \left| K_2 \frac{w_2}{l_2} \cdot (v_{gs_2} - V_{th_2})^2 \cdot (1 + \frac{\lambda_2}{l_2} v_{ds_2}) - K_1 \frac{w_1}{l_1} \cdot (v_{gs_1} - V_{th_1})^2 \cdot (1 + \frac{\lambda_1}{l_1} v_{ds_1}) \right| \tag{3.24}$$

Both transistors have to operate in saturation to reduce the dependency of i_d on v_{ds}. Due to manufacturing tolerances, the values of K, V_{th} and λ can be different which distorts the result. To reduce the effect of channel length modulation, the difference of the drain-source voltages must not exceed a certain value $\Delta V_{ds_{max(dp)}}$ and the lengths of the transistors have to be equal. Since Δi_d also depends on the ratio between width and length of the transistors, the widths have to be equal, too. For too large differences between the gate-source voltages, the differential pair's behaviour becomes non-linear. Therefore, the difference between the gate-source voltages must not exceed a certain value $\Delta V_{gs_{max}}$. Table 3.5 summarises the sizing rules for a differential pair.

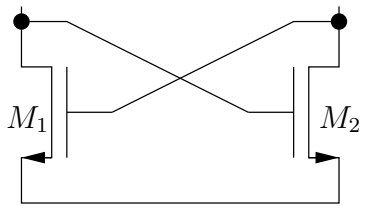

Figure 3.10: NMOS Cross-Coupled Pair

	geometric	electric
Function	$l_1 = l_2$ (3.29) $w_1 = w_2$ (3.30)	———
Robustness	———	———

Table 3.6: Sizing Rules for a Cross-Coupled Pair

3.3.2.4 Cross-Coupled Pair (cc)

This building block which is shown in Figure 3.10 can be found in VCOs operating as a negative resistor, for instance. But it can also form a simple memory consisting of two transistors. For symmetry reasons, both transistors have to be identical. Table 3.6 shows the sizing rules for this building block.

3.3.2.5 Further Building Blocks on Hierarchy Level 1

For the remaining CMOS transistor pairs in Figure 3.2, no sizing rules can be applied immediately. But these pairs form larger building blocks on hierarchy level 2 to which sizing rules apply.

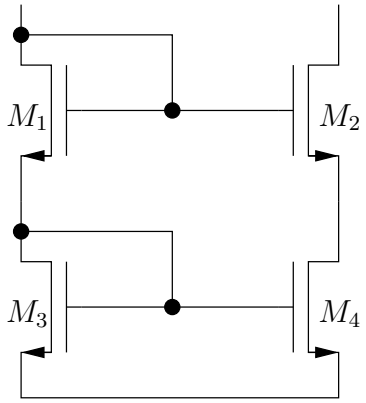

Figure 3.11: NMOS Cascode Current Mirror

	geometric	electric
Function	$w_1 = w_3$ (3.31) $w_2 = w_4$ (3.32)	———
Robustness	———	———
For the level shifter (M_1, M_2), the rules in Table 3.4 hold additionally.		
For the simple current mirror (M_3, M_4), the rules in Table 3.3 hold additionally.		

Table 3.7: Sizing Rules for a Cascode Current Mirror

3.3.3 Building Blocks on Hierarchy Level 2

3.3.3.1 Cascode Current Mirror (CCM)

A cascode current mirror consists of a simple current mirror (transistors M_3 and M_4) together with a level shifter (transistors M_1 and M_2). In this configuration, the voltage difference between the level shifter's source pins is supposed to be zero. This leads to equal voltages at the drain pins of M_2 and M_4 and thus to equal drain-source voltages of M_2 and M_4. In this manner, one of the drawbacks of the simple current mirror is remedied. A cascode current mirror also has a higher output impedance than a simple current mirror. In the ideal case, its current ratio is equal to the ratio between the widths of the simple current mirror's transistors:

$$\frac{i_{d_2}}{i_{d_1}} = \frac{w_4}{w_3} \tag{3.33}$$

Since the level shifter has to produce equal voltages at its source pins to obtain equal drain-source voltages at the transistors of the simple current mirror, both transistors

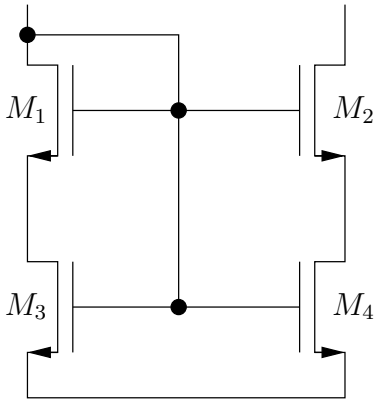

Figure 3.12: NMOS 4-Transistor Current Mirror

	geometric	electric
Function	$w_1 = w_3$ (3.34) $w_2 = w_4$ (3.35)	$\lvert v_{ds_3} - v_{ds_4} \rvert \leq \Delta V_{ds_{max(4TCM)}}$ (3.36)
Robustness	———	———
For the level shifter (M_1, M_2), the rules in Table 3.4 hold additionally.		
For transistors M_3 and M_4, the rules in Table 3.2 hold additionally.		

Table 3.8: Sizing Rules for a 4-Transistor Current Mirror

on the left and both transistors on the right must have the same width. It is useful to use these absolute equalities instead of ratios between the transistor widths, because this reduces the number of degrees of freedom from 4 to 2 instead of from 4 to 3. The fewer design parameters, the faster the optimiser manages to find a solution where all specifications are met. The sizing rules for a cascode current mirror are summarised in Table 3.7.

3.3.3.2 4-Transistor Current Mirror (4TCM)

A 4-transistor current mirror consists of a voltage reference I (transistors M_1 and M_3) and a current mirror load (transistors M_2 and M_4). It has the same advantage over a simple current mirror as the cascode current mirror and the additional advantage of a lower drain-source voltage drop, which is important for lower supply voltages. The two upper transistors are also recognised as a level shifter, hence the respective sizing rules have to be fulfilled by those two transistors. Additionally, the symmetry requirements of the cascode current mirror hold for the transistor widths. Due to the structure of this

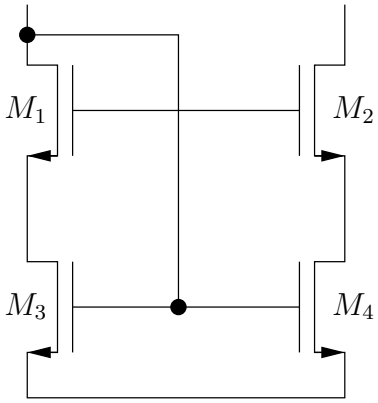

Figure 3.13: NMOS Wide Swing Cascode Current Mirror

	geometric	electric
Function	$w_1 = w_3$ (3.37) $w_2 = w_4$ (3.38)	———
Robustness	———	———
For transistor pair (M_1, M_2), the rules in Table 3.4 hold additionally.		
For transistor pair (M_3, M_4), the rules in Table 3.3 hold additionally.		

Table 3.9: Sizing Rules for a Wide Swing Cascode Current Mirror

building block, transistors M_3 and M_4 operate as voltage controlled resistors, i.e., in the linear region. The difference of their drain-source voltages has to be small to reduce the influence of the drain-source voltage and the channel length modulation. This current mirror is more sensitive to variations since M_3 and M_4 operate in the linear region where the dependency of the drain current on the drain-source voltage is higher. Table 3.8 summarises the sizing rules for a 4-transistor current mirror.

3.3.3.3 Wide Swing Cascode Current Mirror (WSCCM)

A wide swing cascode current mirror (Figure 3.14) consists of a voltage reference II (transistors M_1 and M_3) and a cascode pair (transistors M_2 and M_4). This type of current mirror is usually driven by a diode-connected CMOS transistor or a voltage reference I. It has also got a high output impedance. In addition, the voltage drop along the two transistors M_1 and M_3 is equal to just the gate-source voltage of transistor M_3, while in the cascode current mirror, the voltage drop is the sum of the two left transistors' gate-source voltages. The sizing rules for a wide swing cascode current

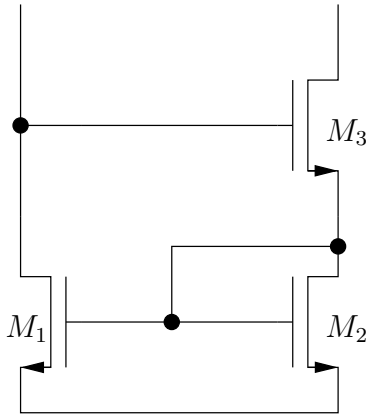

Figure 3.14: NMOS Wilson Current Mirror

For the simple current mirror (M_1, M_2), the rules in Table 3.3 must be fulfilled.
For transistor M_3, the rules in Table 3.1 must be fulfilled.

Table 3.10: Sizing Rules for a Wilson Current Mirror

mirror do not arise from its sub-modules. In contrast, the sizing rules are the same as for a cascode current mirror as given in Table 3.9.

3.3.3.4 Wilson Current Mirror (WCM)

A Wilson current mirror (Figure 3.14) consists of a simple current mirror (transistors M_1 and M_2) and a single transistor M_3. The third transistor forms a cascode pair with transistor M_2, providing a higher output impedance. If transistors M_1 and M_2 are identical, the current ratio is given by:

$$\frac{i_2}{i_1} = \frac{w_2}{w_1}. \tag{3.39}$$

It can be seen, that the role of driving and driven transistor in the simple current mirror are reversed.

For the lower two transistors, the rules for a simple current mirror apply. The third transistor has to operate as a voltage-controlled voltage source, i.e., in saturation. Table 3.10 summarises the sizing rules for a Wilson current mirror.

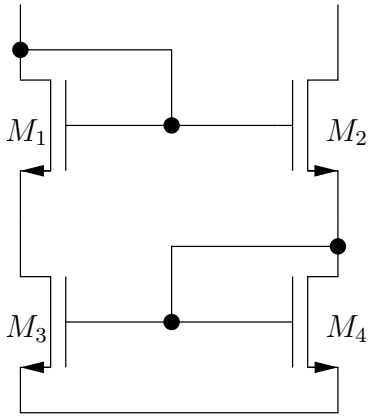

Figure 3.15: NMOS Improved Wilson Current Mirror

	geometric	electric
Function	$w_1 = w_3$ (3.41) $w_2 = w_4$ (3.42)	———
Robustness	———	———
For the level shifter (M_1, M_2), the rules in Table 3.4 hold additionally.		
For the simple current mirror (M_4, M_3), the rules in Table 3.3 hold additionally.		

Table 3.11: Sizing Rules for an Improved Wilson Current Mirror

3.3.3.5 Improved Wilson Current Mirror (IWCM)

A drawback of the Wilson current mirror is the drain-source voltage mismatch between the transistors of the simple current mirror. To remedy this, a fourth transistor is added, so that M_1 and M_2 act as a level shifter that provides equal drain-source voltages at M_3 and M_4. Since an improved Wilson current mirror consists of the same sub-blocks as a cascode current mirror, the sizing rules are the same (see Table 3.7). If M_3 and M_4 are identical, the current ratio is given by

$$\frac{i_2}{i_1} = \frac{w_4}{w_3}. \tag{3.40}$$

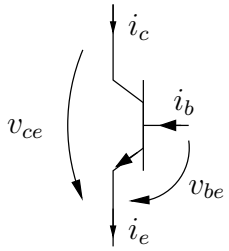

Figure 3.16: NPN Transistor

3.4 Sizing Rules for Bipolar Transistor Building Blocks

In this section, sizing rules for bipolar transistor building blocks will be derived. All sizing rules will be presented for npn transistors but hold analogously for pnp transistors. As for CMOS building blocks, the constants that appear in the following are technology-specific. Hence, they have to be determined only once for each technology.

The three operating regions* of a bipolar transistor (Figure 3.16) are defined as follows:

Cut-off region:	$v_{be} \leq 0 \wedge v_{be} \leq v_{ce}$
Saturation region:	$0 \leq v_{be} \wedge v_{ce} < v_{be}$
Forward-active region:	$0 \leq v_{be} \wedge v_{be} < v_{ce}$

In all presented building blocks, the bipolar transistor is supposed to operate in forward active region. In [Jae04], the collector current of a bipolar transistor operating in forward active region is given by

$$i_c = I_S e^{\frac{v_{be}}{V_T}} \left(1 + \frac{v_{ce}}{V_A}\right). \tag{3.43}$$

Here, $V_T = \frac{k_B T}{q_0}$ is the thermal voltage with k_B being the Boltzmann constant, T the operating temperature and q_0 the electron charge, and V_A is the Early Voltage. The saturation current is denoted by I_S. It depends on the transistor area A and several other parameters like the diffusion constant [LS94], for instance. The transistor area is the design parameter in bipolar transistor technology. Instead of scaling transistors, it is recommended to use identical transistors for each building block and to connect transistors in parallel, e.g., to produce a certain current ratio. The number of transistors connected in parallel will be denoted by N.

In contrast to CMOS transistors whose gate current is usually neglected, the base current i_b of a bipolar transistor has to be considered. The forward current gain β is given by

$$\beta = \frac{i_c}{i_b}. \tag{3.44}$$

* The reverse-active region is not considered in this thesis.

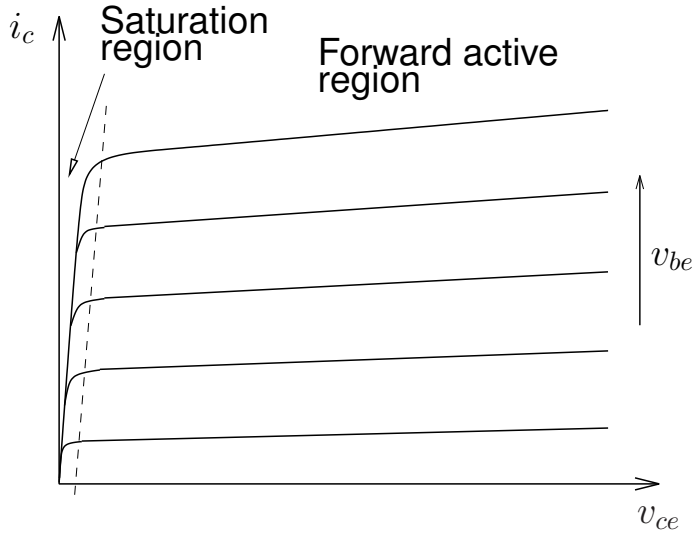

Figure 3.17: Output Curve Family of an npn Transistor

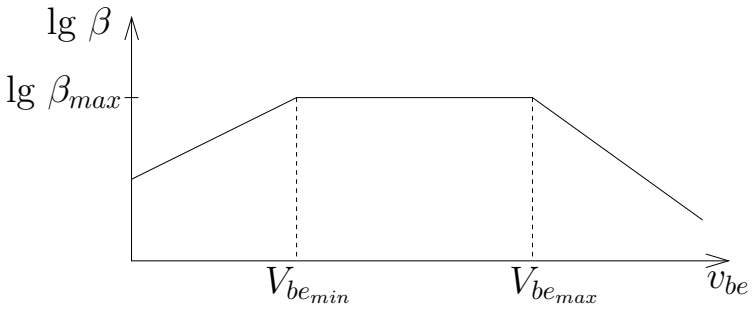

Figure 3.18: Forward Current Gain (logarithmic) against Base-Emitter Voltage in a Bipolar Transistor [LS94]

Hence, the emitter current i_e is given by

$$i_e = i_c + i_b = i_c(1 + \frac{1}{\beta}) = i_b(\beta + 1)\,. \tag{3.45}$$

The value of β depends on v_{be}. Figure 3.18 [LS94] illustrates this. Only as long as v_{be} stays in a range bounded by $V_{be_{min}}$ and $V_{be_{max}}$, the forward current gain β is maximal ($\beta = \beta_{max}$). The lower the value of β becomes, the lower the collector current becomes. This leads to worse performance of transistors, and equal collector currents in different transistors of a building block are practically unachievable when β drops. Hence, for all presented building blocks, $\beta = \beta_{max}$ has to be fulfilled. This leads to the following additional sizing rule for all presented bipolar transistor building blocks:

$$RE: \quad V_{be_{min}} \leq v_{be} \leq V_{be_{max}} \tag{3.46}$$

	geometric	electric
Function	———	$v_{be} - V_{ci} \geq 0$ (3.47) $v_{ce} - (v_{be} - V_{ci}) \geq V_{ce_{SAT}}$ (3.48)
Robustness	$A \geq A_{min_{fwd}}$ (3.49) (3.50)	———

Table 3.12: Sizing Rules for an npn Transistor in Forward Active Region

3.4.1 Building Blocks on Hierarchy Level 0

Transistor in forward active region

All transistors contained in the library in Figure 3.2 have to operate in the forward active region. Hence, v_{be} has to be larger than the cut-in voltage V_{ci} of the base-emitter diode (3.47), and the collector-emitter voltage has to be sufficiently larger than the difference between the base-emitter voltage and V_{ci} (3.48). Experimental results in [MK95] and [DIY95] show that the low-frequency noise in the base current and consequently in the collector current in bipolar transistors is reciprocally dependent on the base-emitter area. Hence, for robustness, the area of the bipolar transistor should be above a certain technology-dependent A_{min} (3.49).

The sizing rules for a bipolar transistor in forward active region are summarised in Table 3.12.

3.4.2 Building Blocks on Hierarchy Level 1

3.4.2.1 Simple Current Mirror (cm)

The function of a simple current mirror (Figure 3.19) is to produce a constant ratio between i_{in} and i_{out} which depends on the number of transistors in parallel, which is N_1 or N_2 respectively. Omitting the base currents and assuming that both transistors operate in the forward active region, the ratio between i_{out} and i_{in} is given by:

$$\frac{i_{out}}{i_{in}} = \frac{N_2}{N_1} \cdot \frac{I_{S_2}}{I_{S_1}} \cdot \frac{e^{\frac{v_{be_2}}{V_{T_2}}}}{e^{\frac{v_{be_1}}{V_{T_1}}}} \cdot \frac{1 + \frac{v_{ce_2}}{V_{A_2}}}{1 + \frac{v_{ce_1}}{V_{A_1}}} \tag{3.51}$$

Manufacturing tolerances which lead to different values of the saturation current and the Early voltage, influence the current ratio. The base-emitter voltages are equal due

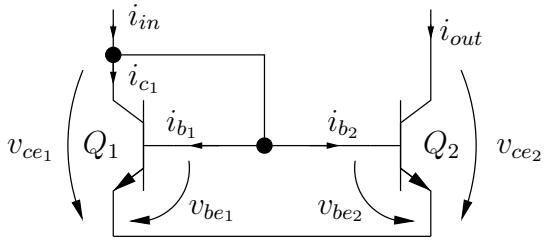

Figure 3.19: NPN Simple Current Mirror

	geometric	electric
Function	$A_1 = A_2$ (3.52)	$\lvert v_{ce_2} - v_{ce_1}\rvert \leq \Delta V_{ce_{max(cm)}}$ (3.53)
Robustness	———	$V_{be_{min}} \leq v_{be_{1,2}} \leq V_{be_{max}}$ (3.54)
Additionally, the rules in Table 3.12 must be fulfilled for both transistors.		

Table 3.13: Sizing Rules for an npn Simple Current Mirror

to the structure of a simple current mirror. The value of V_T is the same for both transistors if the operating temperature is the same. This should be considered during layout generation. The influence of the early voltage can be reduced by keeping the collector-emitter voltages of the two transistors equal, which can be seen from the last fraction in (3.51).

However, the base currents are not zero. Assuming that transistors Q_1 and Q_2 are absolutely identical and $v_{ce_1} = v_{ce_2}$, the base currents i_{b_1} and i_{b_2} as well as the collector currents i_{c_2} and i_{c_1} are equal, and $\beta_1 = \beta_2 = \beta$. Applying Kirchhoff's current low at the common base, the current ratio i_{out}/i_{in} is derived as follows:

$$\begin{aligned}
& i_{in} = N_1 i_{b_1} + N_2 i_{b_2} + N_1 i_{c_1} \\
\iff \quad & i_{in} = (N_1 + N_2) i_{b_2} + N_1 i_{c_2} \\
\iff \quad & i_{in} = \frac{N_1 + N_2}{\beta} i_{c_2} + N_1 i_{c_2} \\
\iff \quad & i_{in} = \frac{N_1 + N_2}{\beta} \cdot \frac{i_{out}}{N_2} + N_1 \frac{i_{out}}{N_2} \\
\iff \quad & i_{in} = i_{out} \frac{N_1 + N_2 + N_1\beta}{N_2\beta} \\
\iff \quad & \frac{i_{out}}{i_{in}} = \frac{N_2}{N_1 + \frac{N_1+N_2}{\beta}}
\end{aligned} \tag{3.55}$$

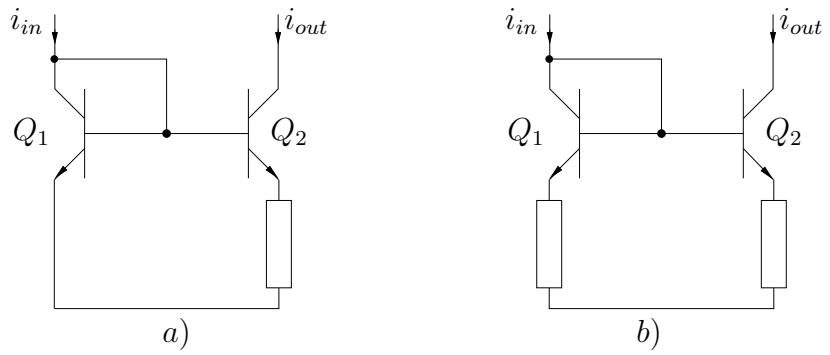

Figure 3.20: Modifications of the Simple Current Mirror

For $N_1 = N_2 = 1$, this becomes

$$\frac{i_{out}}{i_{in}} = \frac{1}{1 + \frac{2}{\beta}}. \tag{3.56}$$

Hence, there is an error in the current ratio dependent on the forward current gain β, which means that β has to maximal, i.e., (3.46) has to be fulfilled for both transistors.

From (3.55), it also follows that $i_{out}/i_{in} \rightarrow \beta$ for $\frac{N_2}{N_1} \rightarrow \infty$. Thus, there is an upper bound for the current ratio that is adjustable using different numbers of transistors in parallel on each side.

Other current mirrors also suffer from an inaccuracy caused by the base-currents dependent on the transistor β, but different structures lead to a weaker dependency with β occurring in quadratic form.

The sizing rules for a bipolar simple current mirror are summarised in Table 3.13.

Modifications of the simple current mirror by adding resistors

Figure 3.20 shows two different modifications of the simple current mirror to raise the output resistance which arise from adding a resistor to one or both branches.

Modification a) is called a Widlar current mirror. According to [LS94], the current ratio is given by

$$\frac{i_{in}}{i_{out}} = \frac{1}{B} exp\left(\frac{R i_{out}}{V_T}\right). \tag{3.57}$$

In this equation, B denotes the number of transistors on the right, but B can be set to N_2/N_1 as well. The current ratio depends not just on the ratio of the number of transistors connected in parallel on each side but also on the value of resistor R and the operating temperature that influences V_T. The difference between the collector-emitter voltages has only little effect on the current ratio, hence, (3.53) does not have to be fulfilled for this building block. All other sizing rules for a simple current mirror have to be fulfilled though.

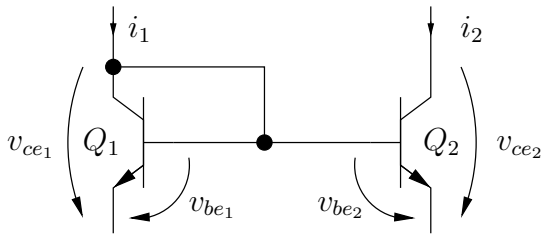

Figure 3.21: NPN Level Shifter

	geometric	electric
Function	$A_1 = A_2$ (3.60)	———
Robustness	———	$V_{be_{min}} \leq v_{be_{1,2}} \leq V_{be_{max}}$ (3.61)
Additionally, the rules in Table 3.12 must be fulfilled for both transistors.		

Table 3.14: Sizing Rules for an npn Level Shifter

In modification b), a resistor is added on both sides. In [LS94], the current ratio is given for equal number of transistors in parallel on both sides. For a different number of transistors in parallel on both sides, the following equation arises:

$$\frac{i_2}{i_1} = \frac{N_2}{N_1} \cdot \frac{R_1}{R_2} \left(\frac{1 + \frac{1}{g_{m_1} R_1}}{1 + \frac{1}{g_{m_2} R_2}} \right) = \frac{N_2}{N_1} \cdot \frac{g_{m_2}(1 + g_{m_1} R_1)}{g_{m_1}(1 + g_{m_2} R_2)}, \tag{3.58}$$

where g_m is the transconductance of a bipolar transistor.
For $g_{m_1} = g_{m_2} = g_m$, this simplifies to

$$\frac{i_2}{i_1} = \frac{N_2}{N_1} \cdot \frac{1 + g_m R_1}{1 + g_m R_2} \approx \frac{N_2}{N_1} \cdot \frac{R_1}{R_2} \quad (\text{for } R_1, R_2 \gg 1). \tag{3.59}$$

Hence, if the number of transistors on both sides is equal, the current ratio is determined by the ratio between the resistor values R_1 and R_2. The sizing rules for this modification are the same as for the original simple current mirror.

3.4.2.2 Level Shifter (ls)

Like its CMOS counterpart (Chapter 3.3.2.2), the function of a bipolar level shifter (Figure 3.21) is to produce either a constant voltage difference between or equal voltages at its emitter pins depending on the collector currents. If the level shifter together with a

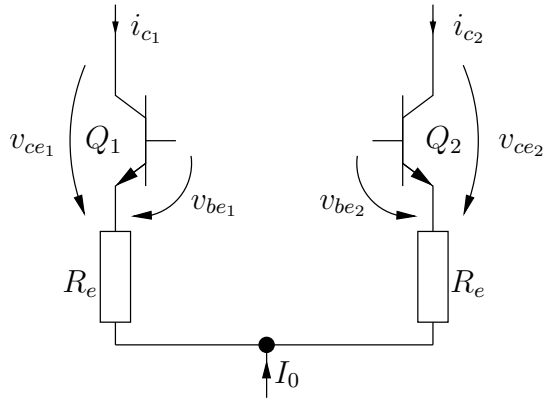

Figure 3.22: NPN Differential Pair with Resistors

	geometric	electric
Function	$A_1 = A_2$ (3.62)	$\lvert v_{ce_2} - v_{ce_1}\rvert \leq \Delta V_{ce_{max(dp)}}$ (3.63)
Robustness	———	$V_{be_{min}} \leq v_{be_{1,2}} \leq V_{be_{max}}$ (3.64) $\lvert v_{be_2} - v_{be_1}\rvert \leq \Delta V_{be_{max}}$ (3.65)
Additionally, the rules in Table 3.12 must be fulfilled for both transistors.		

Table 3.15: Sizing Rules for an npn Differential Pair

simple current mirror is part of a larger current mirror, the voltages of the level shifter's emitter pins should be equal to ensure that the collector-emitter voltages of the simple current mirror are equal.
To ensure that the collector currents do not vary too much, the forward current gain of both transistors has to be maximal and both transistors operate in the forward active region. If this is the case, the influence of the difference between the collector-emitter voltages of the two transistors is very little. Table 3.14 summarises the sizing rules for a bipolar level shifter.

3.4.2.3 Differential Pair (dp)

The function of a differential pair (Figure 3.22, with additional resistors) is to produce a constant difference $\Delta i_c = |i_{c_1} - i_{c_2}|$ between the collector currents dependent on the base-emitter voltages of the two transistors:

$$\Delta i_c = \left| I_{S_1} e^{\frac{v_{be_1}}{V_T}} (1 + \frac{v_{ce_1}}{V_A}) - I_{S_2} e^{\frac{v_{be_2}}{V_T}} (1 + \frac{v_{ce_2}}{V_A}) \right| \tag{3.66}$$

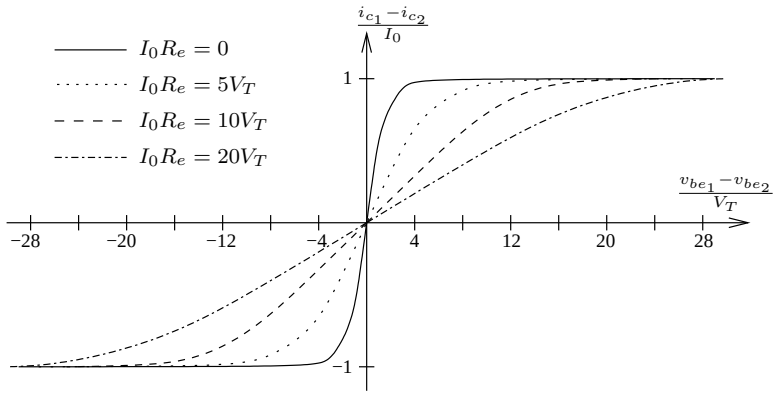

Figure 3.23: Normalised Difference of Collector Currents against Normalised Difference of Base-Emitter Voltages in a Bipolar Differential Pair for Different Values of I_0R_e

To keep the base currents small and to reduce the dependency of i_c on v_{ce}, both transistors have to operate in the forward active region and their forward current gain β has to be maximal. For symmetry reasons, the number of transistors in parallel on both sides has to be the same. The difference of the collector-emitter voltages has to be small to reduce the influence of the Early effect. Both transistors have to be identical to avoid mismatch between the saturation currents. Additionally, the difference between the base-emitter voltages must be smaller than $\Delta V_{be_{max}}$ to keep the differential pair in the linear region. The linear region of a bipolar differential pair is much smaller than that of a CMOS differential pair. The linear region can be enlarged though, using additional resistors at the transistors' emitter pins, as will be explained in the next paragraph. Table 3.15 summarises the sizing rules for a differential pair.

To enlarge the linear region of a bipolar differential pair, a resistor $R_e \neq 0$ is added at each of the transistors' emitter pins. The value of $\Delta V_{be_{max}}$ in (3.65) depends on the value of these resistors and the current flowing into the common emitter terminal. Figure 3.23 shows the normalised difference between the collector currents against the normalised difference of the base-emitter voltages for different values of $R_e \cdot I_0$, whereas I_0 is the value of the current flowing into the common-emitter. For $R_e = 0$, the linear region is smaller than $3V_T$. The higher the product I_0R_e, the larger is the linear region, and thus the value of $\Delta V_{be_{max}}$.

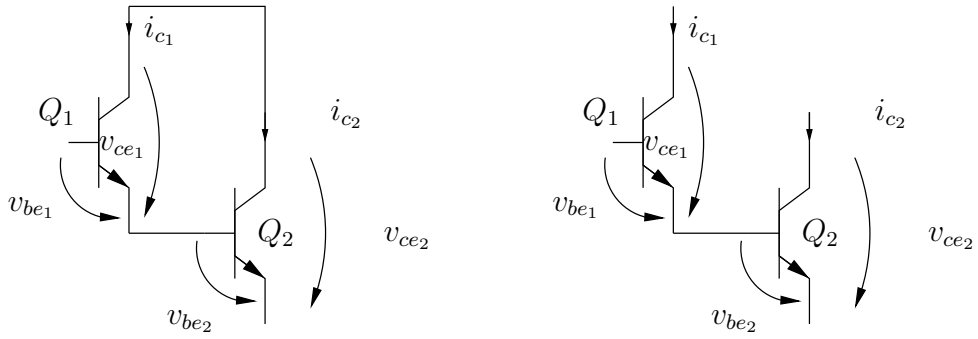

Figure 3.24: NPN Darlington Configuration I and II

	geometric	electric
Function	———	———
Robustness	$A_2 = \beta_1 \cdot A_1$ (3.67)	$V_{be_{min}} \leq v_{be_{1,2}} \leq V_{be_{max}}$ (3.68)
Additionally, the rules in Table 3.12 must be fulfilled for both transistors.		

Table 3.16: Sizing Rules for npn Darlington Configurations I and II

3.4.2.4 Darlington Configuration I / II (dc I / II)

The configurations that are shown in Figure 3.24 are also called Darlington pairs. These configurations act like a single transistor with $\beta_{new} = \beta_1\beta_2$. Configuration II is also used as sub-block of a buffered current mirror (Chapter 3.4.3.1). Both transistors have to operate in the forward active region and for both, the forward current gain has to be maximal to ensure proper function and to require low input currents. Since the collector current flowing into transistor Q_2 is β_1 times higher than the current flowing into Q_1, the area of transistor Q_2 should be β_1 times higher than the area of transistor Q_1. The sizing rules for the Darlington configurations are summarised in Table 3.16.

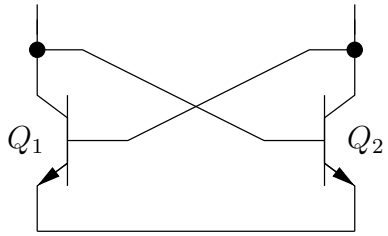

Figure 3.25: NPN Cross-Coupled Pair

	geometric	electric
Function	$A_1 = A_2$ (3.69) $N_1 = N_2$ (3.70)	———
Robustness	———	———

Table 3.17: Sizing Rules for a Cross-Coupled Pair

3.4.2.5 Cross-Coupled Pair (cc)

Like its CMOS counterpart, this building block requires identical transistors with equal area and equal number of transistors connected in parallel on both sides. The sizing rules are summarised in Table 3.17.

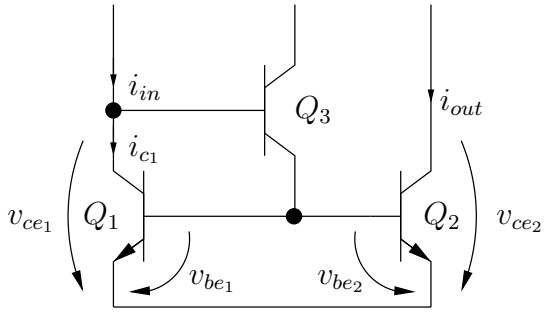

Figure 3.26: NPN Buffered Current Mirror

For the transistor pair (Q_1, Q_2), the rules in Table 3.13 must be fulfilled.
For transistor Q_3, the rules in Table 3.12 must be fulfilled.

Table 3.18: Sizing Rules for a Buffered Current Mirror

3.4.3 Building Blocks on Hierarchy Level 2

3.4.3.1 Buffered Current Mirror (BCM)

A buffered current mirror is another modification of a simple current mirror. To reduce the error in the current ratio caused by the base current, an additional transistor Q_3 is added between the input and the common base. To model the buffered current as a combination of two building blocks, it is not necessary to define a new building block consisting of transistors Q_1 and Q_2. Instead, Q_2 and Q_3 that form a Darlington configuration II are combined with the single transistor Q_1.

If all three transistors are identical, $v_{ce_1} = v_{ce_2}$, $i_{b_1} = i_{b_2}$, and $i_{c_1} = i_{c_2}$, $\beta_1 = \beta_2 = \beta_3$ is obtained. Thus, the current ratio of this current mirror can be derived as follows:

$$
\begin{aligned}
& i_{in} = N_1 i_{c_1} + i_{b_3} \\
\Longleftrightarrow \quad & i_{in} = \frac{N_1}{N_2} i_{out} + \frac{i_{e_3}}{1+\beta} \\
\Longleftrightarrow \quad & i_{in} = \frac{N_1}{N_2} i_{out} + \frac{N_1 i_{b_1} + N_2 i_{b_2}}{1+\beta} \\
\Longleftrightarrow \quad & i_{in} = \frac{N_1}{N_2} i_{out} + \frac{(N_1+N_2) i_{b_2}}{1+\beta} \\
\Longleftrightarrow \quad & i_{in} = \frac{N_1}{N_2} i_{out} + \frac{N_1+N_2}{1+\beta} \cdot \frac{i_{out}}{\beta N_2} \\
\Longleftrightarrow \quad & i_{in} = i_{out} \left(\frac{N_1+N_2}{\beta N_2 + \beta^2 N_2} + \frac{N_1}{N_2} \right) \\
\Longleftrightarrow \quad & i_{in} = i_{out} \left(\frac{N_1+N_2+N_1(\beta+\beta^2)}{N_2(\beta+\beta^2)} \right) \\
\Longleftrightarrow \quad & \frac{i_{out}}{i_{in}} = \frac{N_2}{N_1 + \frac{N_1+N_2}{\beta+\beta^2}}
\end{aligned}
\tag{3.71}
$$

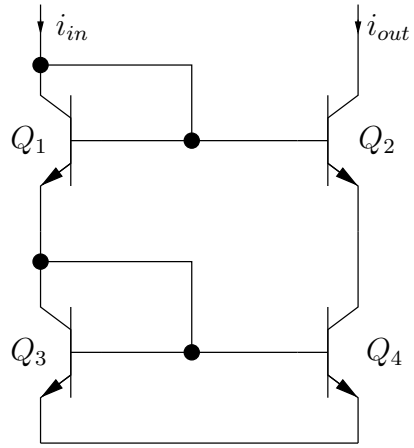

Figure 3.27: NPN Cascode Current Mirror

	geometric	electric
Function	$N_1 = N_3$ (3.72) $N_2 = N_4$ (3.73)	———
Robustness	———	———
For the level shifter (Q_1, Q_2), the rules in Table 3.14 hold additionally.		
For the simple current mirror (Q_4, Q_3), the rules in Table 3.13 hold additionally.		

Table 3.19: Sizing Rules for a Cascode Current Mirror

With i_{out}/i_{in} depending on β^2, the error caused by the transistor β is much smaller than in a simple current mirror, as long as the value of β does not become too small. Hence, (3.46) has to be fulfilled for all three transistors.

The sizing rules for transistors Q_1 and Q_2 are the same as for a simple current mirror. Transistor Q_3 has to operate in forward active region. Table 3.18 summarises the sizing rules for a buffered current mirror.

3.4.3.2 Cascode Current Mirror (CCM)

A cascode current mirror consists of a simple current mirror (Q_3, Q_4) together with a level shifter (Q_1, Q_2). It works the same way as its CMOS counterpart (see 3.3.3.1). Hence, the voltage difference between the level shifter's emitter pins is supposed to be zero which leads to equal collector-emitter voltages at the simple current mirror's transistors.

Like in the other current mirrors, the current ratio i_{out}/i_{in} is distorted due to the non-zero base currents, whereby (3.46) must hold for all transistors. This is already required for the cascode current mirrors sub-modules.

From (3.55), it follows that

$$\frac{N_2 i_{e_2}}{N_1 i_{e_1}} = \frac{N_4}{N_3 + \frac{N_3+N_4}{\beta}}.$$

Hence, the current ratio i_{out}/i_{in} can be derived as follows:

$$\begin{aligned}
& i_{in} = N_1(i_{c_1} + i_{b_1}) + N_2 i_{b_2} \\
\iff \quad & i_{in} = N_1 i_{e_1} + N_2 \frac{i_{c_2}}{\beta} \\
\iff \quad & i_{in} = N_1 i_{e_1} + N_2 \frac{i_{out}}{\beta} \\
\iff \quad & i_{in} = N_2 i_{e_2} \frac{N_3 + \frac{N_3+N_4}{\beta}}{N_4} + \frac{i_{out}}{\beta} \\
\iff \quad & i_{in} = N_2 i_{c_2}(1 + \frac{1}{\beta}) \frac{N_3 + \frac{N_3+N_4}{\beta}}{N_4} + \frac{i_{out}}{\beta} \\
\iff \quad & i_{in} = i_{out}(1 + \frac{1}{\beta}) \frac{N_3 + \frac{N_3+N_4}{\beta}}{N_4} + \frac{i_{out}}{\beta} \\
\iff \quad & i_{in} = i_{out} \left(\frac{\beta N_3 + N_3 + N_4}{\beta N_4} + \frac{\beta N_3 + N_3 + N_4}{\beta^2 N_4} + \frac{1}{\beta} \right) \\
\iff \quad & i_{in} = i_{out} \frac{\beta^2 N_3 + 2\beta(N_3 + N_4) + N_3 + N_4}{\beta^2 N_4} \\
\iff \quad & \frac{i_{out}}{i_{in}} = \frac{N_4}{N_3 + 2\frac{N_3+N_4}{\beta} + \frac{N_3+N_4}{\beta^2}}
\end{aligned} \tag{3.74}$$

So, on the one hand, the error caused by a difference in the collector-emitter voltages between transistors Q_3 and Q_4 is reduced, but on the other hand, the error caused by the base currents is larger than in a simple current mirror.

Since the level shifter has to produce equal voltages at its emitter pins to obtain equal collector-emitter voltages at the transistors of the simple current mirror, the number of transistors in parallel on the left has to be the same as well as the number of transistors in parallel on the right.
The sizing rules for a cascode current mirror are summarised in Table 3.19.

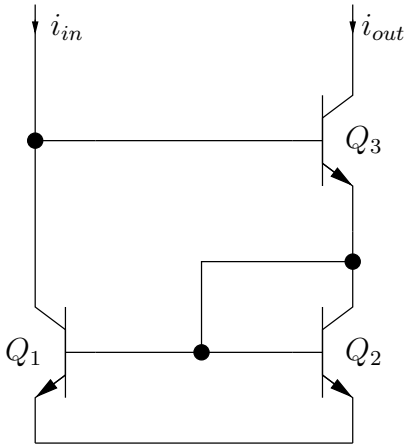

Figure 3.28: NPN Wilson Current Mirror

For the simple current mirror (Q_2, Q_1), the rules in Table 3.13 must be fulfilled.
For transistor (Q_3), the rules in Table 3.12 must be fulfilled.

Table 3.20: Sizing Rules for a Wilson Current Mirror

3.4.3.3 Wilson Current Mirror (WCM)

A Wilson current mirror (Figure 3.28) consists of a simple current mirror (Q_1, Q_2) and a single transistor Q_3. As for its CMOS counterpart, the current ratio is determined by transistors M_1 and M_2 that form the simple current mirror, but the role of driving and driven transistor are reversed.

With (3.55) it follows:

$$\frac{N_1 i_{c_1}}{N_2 i_{e_3}} = \frac{N_1}{N_2 + \frac{N_1+N_2}{\beta}}.$$

The current ratio i_{out}/i_{in} is derived as follows:

$$\begin{aligned}
& i_{in} = N_1 i_{c_1} + N_3 i_{b_3} \\
\iff \quad & i_{in} = N_2 i_{e_3} \frac{N_1}{N_2 + \frac{N_1+N_2}{\beta}} + N_3 \frac{i_{c_3}}{\beta} \\
\iff \quad & i_{in} = i_{out}(1 + \frac{1}{\beta}) \frac{N_1}{N_2 + \frac{N_1+N_2}{\beta}} + \frac{i_{out}}{\beta} \\
\iff \quad & i_{in} = i_{out} \left(\frac{\beta N_1}{\beta N_2 + N_1 + N_2} + \frac{N_1}{\beta N_2 + N_1 + N_2} + \frac{1}{\beta} \right) \\
\iff \quad & i_{in} = i_{out} \frac{\beta^2 N_1 + \beta N_1 + \beta N_2 + N_1 + N_2}{\beta^2 N_2 + \beta(N_1 + N_2)} \\
\iff \quad & \frac{i_{out}}{i_{in}} = \frac{N_2 + \frac{N_1+N_2}{\beta}}{N_1 + \frac{N_1+N_2}{\beta} + \frac{N_1+N_2}{\beta^2}}
\end{aligned} \tag{3.75}$$

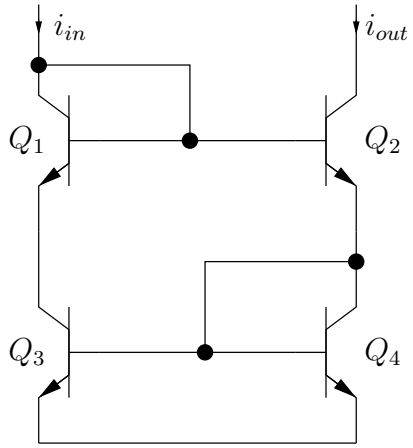

Figure 3.29: NPN Improved Wilson Current Mirror

	geometric	electric
Function	$N_1 = N_3$ (3.76) $N_2 = N_4$ (3.77)	———
Robustness	———	———
For the level shifter (Q_1, Q_2), the rules in Table 3.14 hold additionally.		
For the simple current mirror (Q_4, Q_3), the rules in Table 3.13 hold additionally.		

Table 3.21: Sizing Rules for an Improved Wilson Current Mirror

Since numerator and denominator only differ by one term which is proportional to $1/\beta^2$, the error caused by the base currents is low.

For Q_1 and Q_2, the rules for a simple current mirror apply, but the roles of driving and driven transistor are reversed. Transistor Q_3 also has to operate in forward active region and for matching purposes, the number of transistors in parallel has to be the same as for Q_2. Table 3.20 summarises the sizing rules for a Wilson current mirror.

3.4.3.4 Improved Wilson Current Mirror (IWCM)

An improved Wilson current mirror arises from a Wilson current mirror by adding a forth transistor to remedy the error caused by difference between the collector-emitter voltages of the two transistors of the simple current mirror. Hence, an improved Wilson current mirror can be modelled as a combination of a level shifter and a simple current mirror, just like its CMOS counterpart.

From (3.55), it follows that

$$\frac{N_1 i_{e_1}}{N_2 i_{e_2}} = \frac{N_3}{N_4 + \frac{N_3+N_4}{\beta}}.$$

Thus, the current ratio is derived as follows:

$$\begin{aligned}
& i_{in} = N_1(i_{c_1} + i_{b_1}) + N_3 i_{b_3} \\
\Longleftrightarrow \quad & i_{in} = N_1 i_{e_1} + N_3 \frac{i_{c_3}}{\beta} \\
\Longleftrightarrow \quad & i_{in} = N_1 i_{e_2} \frac{N_3}{N_4 + \frac{N_3+N_4}{\beta}} + \frac{i_{out}}{\beta} \\
\Longleftrightarrow \quad & i_{in} = i_{out}(1 + \frac{1}{\beta}) \frac{N_3}{N_4 + \frac{N_3+N_4}{\beta}} + \frac{i_{out}}{\beta} \\
\Longleftrightarrow \quad & \frac{i_{out}}{i_{in}} = \frac{N_4 + \frac{N_3+N_4}{\beta}}{N_3 + \frac{N_3+N_4}{\beta} + \frac{N_3+N_4}{\beta^2}}
\end{aligned} \tag{3.78}$$

It can be seen that the current ratio is exactly the same as for the Wilson current mirror, i.e., the error caused by the base currents is also low.

Since an improved Wilson current mirror consists of the same sub-modules as a cascode current mirror (3.4.3.2), the sizing rules are the same (see Table 3.19).

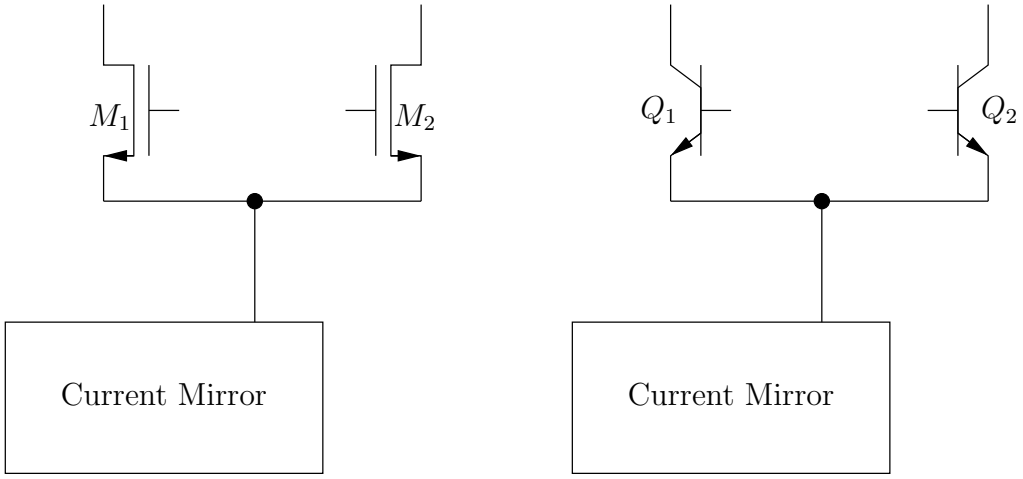

Figure 3.30: Differential Stage with NMOS or npn Differential Pair

3.5 Hybrid Building Blocks

Differential Stage (DS)

A differential stage consists of an arbitrary current mirror and a differential pair. It does not produce any sizing rules itself, but it is important to produce an error-free recognition result. During the recognition process (Chapter 4), usually many differential pairs are recognised, since the only requirement is two transistors connected solely via their source pins. As soon as a differential pair is recognised as part of a differential stage, it is clear that the sizing rules for a differential pair must apply for the two respective transistors.

A differential stage is a hybrid building block. That means the differential pair and the current mirror do not have to be of the same transistor family. But it is important that all transistors are either of n-type or p-type.

Chapter 4

Structure Recognition

4.1 Automatic Module Recognition

Figure 3.4 on page 23 shows the hierarchical library L in UML notation. All modules above level 0 are pairs of modules on lower hierarchy levels. The task of automatic module recognition is to find pairs of modules that have the same connectivity as those defined in L. For this purpose, a new algorithm has been developed that searches for these pairs bottom-up. At the beginning, only transistor pairs, i.e., modules on hierarchy level 1 are searched. After that, modules on higher hierarchy levels can be detected.

During the process of structure recognition, assignment ambiguities can occur. For instance, a transistor may be part of two or more different modules defined in the library in Figure 3.2 at the same time due to its connectivity with two other transistors. Sometimes, it is intended that a transistor is part of more than one larger module, but usually, this is not the case. Hence, a decision has to be made, which module to keep. In this case, the modules are mutually exclusive, i.e., only one of them is valid. The arbitration of such assignment ambiguities will be covered in Chapter 4.2.

In theory, the recognition algorithm corresponds to a search for sub-graph isomorphisms, which is known to be NP-complete. The sub-graphs correspond to the schematics of the module set defined in L. The computational cost of the recognition algorithm is low since both the number of transistors in analog circuits and the number of modules in the library in 3.2 is rather small and only pairs of modules are being searched. In addition, the recognition algorithm is organised such that the number of pairs to compare is reduced. The number of transistor pairs to be checked when searching for modules on hierarchy level 1 is limited to $\binom{n}{2} < \frac{1}{2}n^2$ for $n \geq 2$, with n being the number of transistors in the circuit to examine. Since there are different types of transistors in a circuit, this number is even smaller. In case that 50% of the transistors in a circuit are NMOS and 50% are PMOS transistors, it is just $2\binom{0.5n}{2} < \frac{1}{4}n^2$. More on the complexity of the structure recognition algorithm can be found in Chapter 4.4.

A circuit is represented by a netlist. Among other things, the netlist comprises all physical devices contained in the circuit. For every physical device, the nets its pins

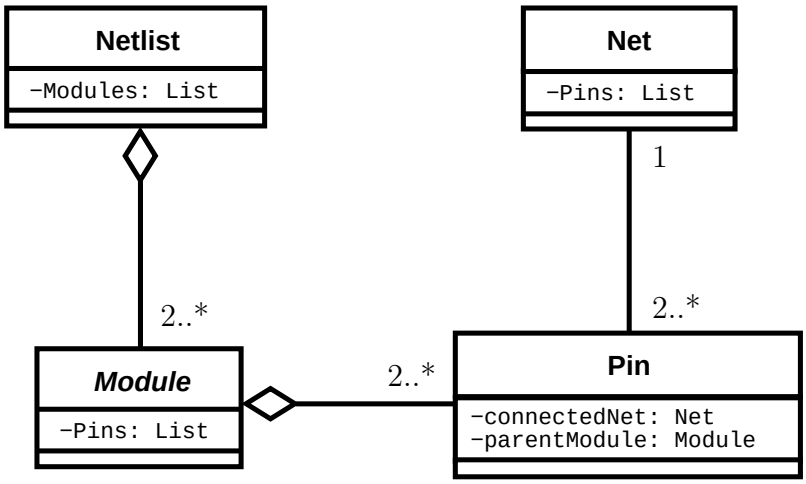

Figure 4.1: Representation of a Netlist in UML Notation

are connected to are listed. Figure 4.1 shows the representation of a netlist in UML notation. A netlist contains at least two modules. A module has at least two pins. Every pin is connected to exactly one net, while a net is connected to at least two pins as long as there are no floating nets or floating pins in the circuit.

The netlist is usually available as a text file and can be parsed to extract the names of all devices and nets present in the circuit. Based on this information, automatic module recognition can be performed. From the netlist, a set M of modules contained in the circuit is instantiated first. At this point, all modules in M are single-modules, namely physical devices. During the recognition process, set M is extended by each new module that is recognised. The given netlist has to be flat, so, if a hierarchical netlist is given, all sub-circuits have to be flattened out. A first way of defining M is as a set of numbered modules m_μ:

$$M = \{m_\mu \,|\, \mu = 1, 2, \ldots, |M|\}. \tag{4.1}$$

Set M can also be considered a union of subsets like the hierarchical library L in Figure 3.2:

$$M = \bigcup_{i=0}^{h_L} M_i. \tag{4.2}$$

In this case, the modules in M are grouped into subsets M_i according to their levels of hierarchy.

Additionally, M can be formulated as the union of subsets that only contain modules of a certain type as defined by set Y:

$$M = M_{(NMOS,t)} \cup M_{(PMOS,t)} \cup \cdots \cup M_{(pnp\_PMOS,DS)}. \tag{4.3}$$

All these three forms of representation will be used in this thesis.

Set N of nets connecting the modules is defined as a set of numbered nets n_ν:

$$N = \{n_\nu \,|\, \nu = 1, 2, \ldots, |N|\}. \tag{4.4}$$

The set of pins P_μ of a single module m_μ is defined as:

$$P_\mu = \{m_\mu.p_\psi \,|\, \psi = 1, 2, \ldots, |P_\mu|\},\ p_\psi \in \{d, g, s, c, b, e, \ldots\}. \tag{4.5}$$

The names of the pins are used for illustration here. For instance, $m_1.p_\psi = m_1.d$ denotes the drain pin of a CMOS transistor that is module m_1. For modules on higher hierarchy levels, special pin names like "common source" are used. Set P of all pins is the union of all sets P_μ:

$$P = \bigcup_{\mu=1}^{|M|} P_\mu. \tag{4.6}$$

With these definitions, a formal representation of the circuit netlist can be given. A heterogeneous relation C that represents the pin-to-net connections of all modules in the given circuit is given by:

$$C \subseteq P \times N. \tag{4.7}$$

In graph notation, graph $G_C = (P \cup N, C)$ is a bipartite graph, i.e., there are only edges between vertices of set P and N, not between vertices of the same set. Since set P is the union of all sets P_μ referring to the pins of module m_μ, relation C can be defined as the union of all relations C_μ which refer to the pin-to-net connections of a single module m_μ:

$$C_\mu \subseteq P_\mu \times N\,, \qquad C = \bigcup_{\mu=1}^{|M|} C_\mu. \tag{4.8}$$

Consequently, the connectivity of two modules m_κ and m_λ is represented by relation $C_{\kappa,\lambda}$ as follows:

$$C_{\kappa,\lambda} = C_\kappa \cup C_\lambda \tag{4.9}$$

Hence, if $C_{\kappa,\lambda}$ is equal to $C_l := C_{l(1)} \cup C_{l(2)}$ with l being any module in L, the modules m_κ and m_λ form a new module m_μ of the same type as l.

The recognition algorithm is depicted in Figure 4.2. The attached line numbers assist in the following detailed explanation of the algorithm.

At the beginning, set M is initialised with all circuit elements contained in the netlist, i.e., available modules from level 0. In addition, two relations R_1 and R_2 are initialised. Every time a new module is detected, a new ordered pair will be added to both relations. The upper or left sub-module (according to Figure 3.2), i.e., sub-module with index 1, and the new module are stored as an ordered pair in R_1. The lower or right sub-module (with index 2) and the new module are stored as an ordered pair in R_2. This information is needed to handle recognition ambiguities (Chapter 4.2).

The outermost loop (line 3) iterates bottom-up through all library elements $l \in L - L_0$, i.e., pair-modules only. This means, each library element is only examined once during the whole process. It does not matter in what order the library elements on each

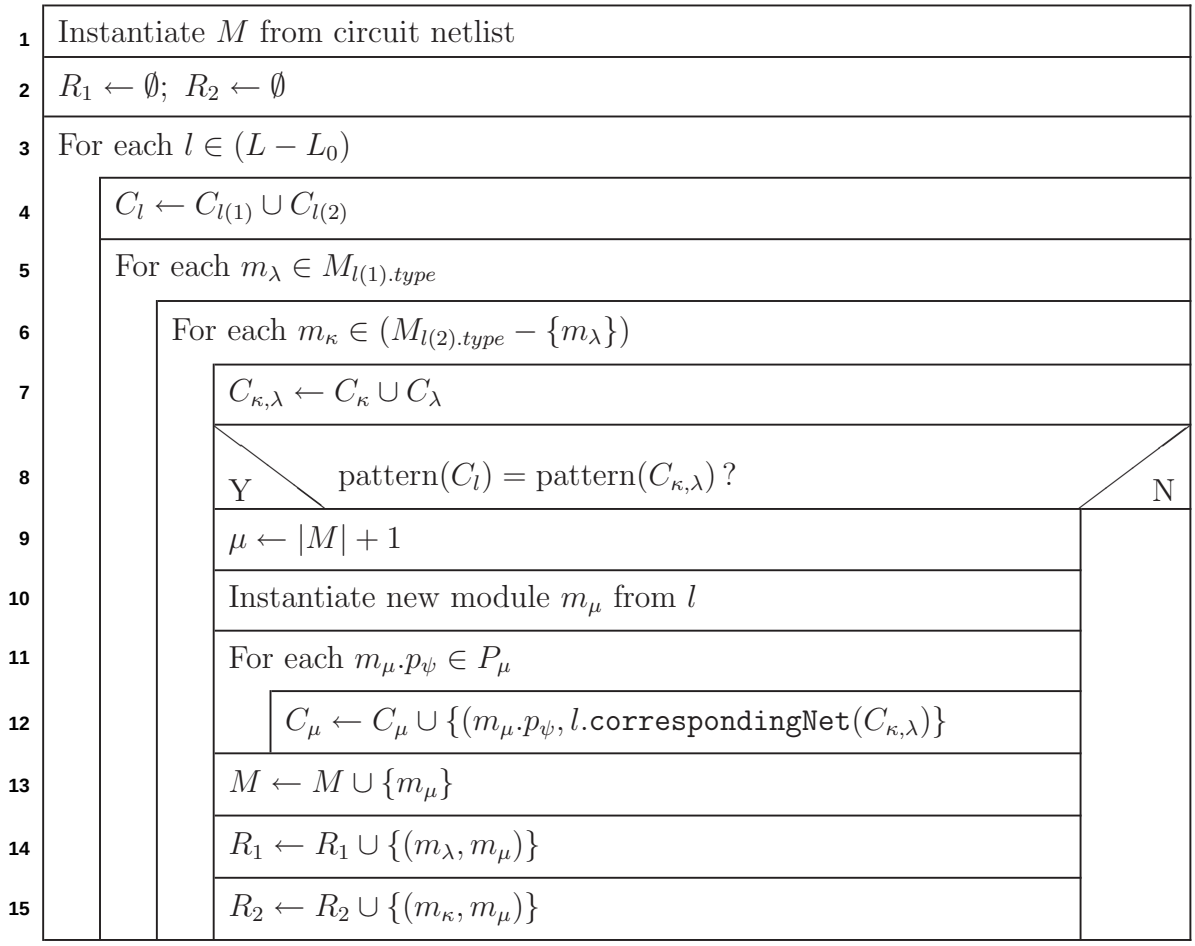

Figure 4.2: Building Block Recognition Algorithm

hierarchy level are examined, since the sub-modules of newly recognised modules will not be removed from M during the recognition process.

Relation C_l is built for each library element (line 4). In the inner loops (lines 5 and 6), all possible pairs of appropriate modules m_κ and m_λ are examined. Appropriate means that all m_κ and m_λ that are selected are of the same type as $C_{l(1)}$ or $C_{l(2)}$. For instance, when NMOS cascode current mirrors are searched, only NMOS simple current mirrors and NMOS level shifters that were detected before are selected for m_κ and m_λ. If the pattern $C_{\kappa,\lambda}$ (line 7) matches the pattern C_l (line 8), the pair (m_κ, m_λ) forms a new module m_μ. This module with $\mu = |M| + 1$ is instantiated based on the current library element l (lines 9 and 10).

Next, the pins of the new module are connected to the appropriate nets of its sub-modules (lines 11 and 12). The number of pins of m_μ is equal to the number of nets in $C_{\kappa,\lambda}$. For each net, m_κ and m_λ are connected to, a pin for m_μ is instantiated. This new pin has to be connected to the appropriate net, e.g., in a simple current mirror, a pin "common source" will be instantiated and connected to the net, the source or emitter pins of the two transistors are connected to. For each library element, a specific mapping

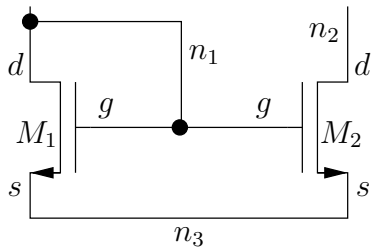

Figure 4.3: NMOS Simple Current Mirror with Pins and Nets

exists which picks out the proper net from $C_{\kappa,\lambda}$. The function `correspondingNet` is applied on each pin of m_μ to connect it to a representative pin of m_κ or m_μ that is selected by a mapping. A small example which shows the mapping for a simple current mirror is given after the description of the recognition algorithm.

After that, the new module m_μ is added to set M (line 13). Consequently, m_μ can be recognised as part of other modules in the next run of the outer loop (line 3). Its sub-modules are also kept in M, since a module can be contained in several modules in some cases. In addition, it cannot be determined beforehand if the detected module is really meaningful and removing a sub-module would rule out other possible modules containing that sub-module. Ambiguities that may occur, will be treated afterwards (Chapter 4.2).

Finally, relations R_1 and R_2 are extended by one ordered pair each, consisting of the first or second sub-module respectively, as well as the new module (lines 14 and 15).

Example for pin mapping

Figure 4.3 shows two transistors M_1 and M_2 forming an NMOS simple current mirror. The pins and the three nets they are connected to are labelled in Figure 4.3. The adjacency matrix of circuit relation $C_{1,2} := C_1 \cup C_2$ looks as follows:

$$\mathbf{C_{1,2}} = \begin{array}{c} \\ M_1.d \\ M_1.g \\ M_1.s \\ M_2.d \\ M_2.g \\ M_2.s \end{array} \begin{array}{c} \begin{array}{ccc} n_1 & n_2 & n_3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

The columns in matrix $\mathbf{C_{1,2}}$ show the connections that can also be seen in Figure 4.3: drain and gate of M_1, and the gate of M_2 are connected, as well as both transistors' sources, while the drain of M_2 is not connected to any other pins. This matches exactly

the pattern of a simple current mirror with M_1 as driving transistor and M_2 as driven transistor, i.e., $C_{1,2} = C_{cm}$.

Thus, a new simple current mirror cm_1 with sub-modules M_1 and M_2 will be instantiated. The function `correspondingNet` makes sure that each of the simple current mirror's pins will be connected to the appropriate net. In detail, with $m_\kappa = M_1$, $m_\lambda = M_2$, and $m_\mu = cm_1$, the mapping looks as follows:

$$m_\kappa.d \longmapsto m_\mu.\texttt{drivingIn}$$
$$m_\lambda.d \longmapsto m_\mu.\texttt{drivenIn}$$
$$m_\kappa.s \longmapsto m_\mu.\texttt{commonSource}$$

This means, the `drivingIn`-pin of the simple current mirror is connected to n_1, its `drivenIn`-pin to n_2, and its `commonSource`-pin to n_3. □

Banks are treated differently in the recognition process. If a pair of equal modules (e.g., a pair of simple current mirrors) shares one common driving driving stage (e.g., the driving transistor of both simple current mirrors), these two modules belong to a bank. If one of those modules is already part of a bank, the other modules will be added to that bank. If not, a new bank will be instantiated from these two modules. Since banks do not produce sizing rules, the recognition of banks will be performed after the whole recognition process including the arbitration of assignment ambiguities.

The hierarchical library representation allows a redundancy-free storage of generic sizing rules. Since each module on level i consists of modules from lower hierarchy levels, in most cases only the additional rules that arise on level i have to be stored in the libraries. Experimental results of the structure recognition are presented in Section 5. The procedure in Figure 4.2 holds as well for any other hierarchy of modules that is organised like the one shown in Figure 3.2.

4.2 Arbitration of Assignment Ambiguities

Using the algorithm in Figure 4.2 to recognise modules in a circuit, all possible modules are detected purely from a structural analysis. This can lead to ambiguities. Without employing additional rules to resolve possible conflicts, the number of detected modules in each of the presented circuits would be too high in most cases. The recognition algorithm only checks type and connectivity to determine if a pair of modules forms a module. This could result in modules being recognised as part of different modules at the same time which is not correct in certain cases.

In this thesis, two types of ambiguities are distinguished. The first type is "recognition conflicts". These can occur when modules are recognised as sub-modules of more than one module at the same time. This is sometimes intended, but it often leads to

contradictory recognition results that have to be resolved to avoid the assignment of non-applicable sizing rules which could even cause the feasible region to be empty. The second type is about "uncertain building blocks", i.e., the connectivity alone does not contain sufficient information to determine if a detected pair of modules really fulfils the function of the corresponding building block. In case of a single transistor, this means that the operating region of a transistor cannot be determined automatically if it was not recognised as part of any building block at all.

In the following, both types of ambiguities will be discussed in detail and solutions how to arbitrate these ambiguities will be presented.

4.2.1 Recognition Conflicts

4.2.1.1 Problem Description

Especially on hierarchy level 1, where the structural type of the sub-modules is always "transistor", it is possible that one transistor is recognised as part of several modules at the same time during the recognition process. In some cases, this is intended. For instance, a transistor can be part of both a voltage reference I and a level shifter, or the driving transistor is contained in several simple current mirrors that form a simple current bank. But in most cases, only the recognition of one of the modules was intended and the others have to be discarded from the set of recognised modules.

Figure 5.1 on page 82 as well as Figure 4.4 show a folded cascode circuit. In Figure 5.1, all recognised modules above level 0 are shaded. This figure presents the recognition result after the arbitration of assignment ambiguities. Originally, twelve* additional transistor pair modules were recognised by the algorithm in Figure 4.2. Figure 4.4 shows only these additionally recognised modules. These are eight differential pairs, namely $\{MN3, MN11\}$, $\{MN5, MN11\}$, $\{MP1, MP5\}$, $\{MP1, MP7\}$, $\{MP1, MP9\}$, $\{MP3, MP5\}$, $\{MP3, MP7\}$, and $\{MP3, MP9\}$ which are labelled from $dp2$ to $dp9$ and four level shifters $\{MP2, MP3\}$, $\{MP2, MP6\}$, $\{MP2, MP8\}$, $\{MP2, MP10\}$, which are labelled from $ls5$ to $ls8$. From the schematic of the folded cascode circuit, it can be seen that these assignments are wrong. The transistors contained in $dp2$ to $dp9$ are actually part of current mirrors. Transistor $MP3$ is part of a current mirror load and transistors $MP6$, $MP8$ and $MP10$ belong to wide swing cascode current mirrors, not to level shifters. But as the automatic recognition algorithm only checks the connectivity of the modules in the netlist, each pair of transistors that are only connected via their source pins is recognised as a differential pair. Accordingly, the four additional level shifters are recognised wrongly, based on the connectivity of the corresponding transistors. Thus, many additional sizing rules would be assigned to each of these pairs. For all transistors that are solely connected via their source pins, the sizing rules for a CMOS differential pair (3.3.2.3) would apply.

* Cascode pairs are not considered in this example, since they do not deliver any sizing rules.

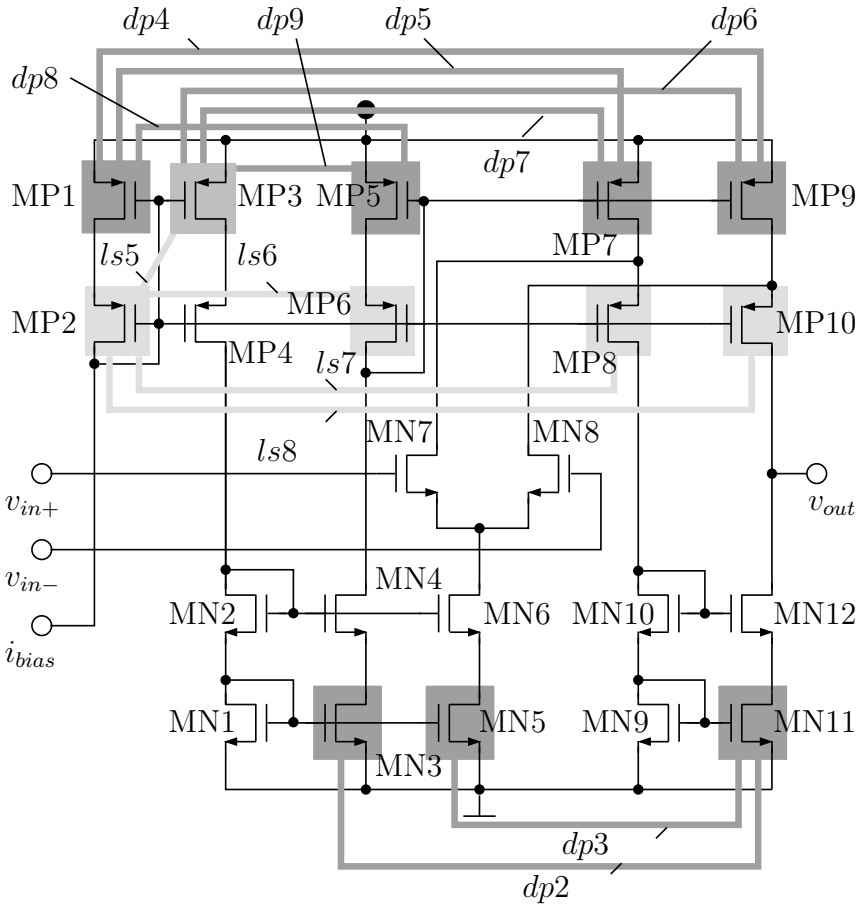

Figure 4.4: Wrongly Detected Modules in a Folded Cascode Operational Amplifier

As a first consequence, all these transistors would have to operate in saturation region. This would not be a problem in most cases, since the rules for the simple current mirrors include saturation for both transistors they consist of. But transistors $MP1$ and $MP2$ would have to operate in saturation and triode region at the same time, as the latter is preset through the rules of a four transistor current mirror (see Chapter 3.3.3.2). This is contradictory and as a result, the feasible region would be empty. As a second consequence, the difference between the drain-source and gate-source voltages of the transistor pairs in *dp*2 to *dp*9 would not be allowed to exceed a certain maximum value (see (3.27) and (3.28)). A rule for maximum difference between the drain-source voltages of two transistors also exists for the simple current mirror (see (3.18)). This could at least reduce the feasible region to a very small space.
The results in Chapter 5 show that assignment ambiguities occur in all of the examined circuits. In one of the examined circuits, a total of 34 modules were removed after the recognition process. Hence, the arbitration of these ambiguities is crucial.

4.2.1.2 Resolving Recognition Conflicts

A recognition conflict occurs whenever a module is being recognised as part of at least two modules that are mutually exclusive. In this case, it has to be decided which of these modules to accept. Hence, a set of rules has been set up to determine which module to prefer in case of a conflict. A way to realise this is to sort the modules by preference. Every module "dominates" other modules with lower preferences and is dominated by modules with higher preference.

In this context, two questions arise:

1. Is it possible to include this resolution completely into the recognition algorithm in Figure 4.2, or is it necessary to post-process the recognition result after running the recognition algorithm?
2. Can the modules in the library in Figure 3.2 be sorted by preference such that all conflicts between different modules can be resolved?

The first question can only be answered yes if there are no conflicts between modules on different hierarchy levels or if modules on lower levels are always to be preferred over modules on higher levels, since the recognition algorithm works bottom-up. In this case, the recognition of modules with lower preference would be prevented if their sub-modules to-be were already part of another module that was in conflict with the current one.

The following two examples illustrate that conflicts between modules on different hierarchy levels can actually occur and that some modules on higher hierarchy levels dominate some on lower ones. This means, the first question cannot be answered yes.

Figure 4.5 shows a wide swing cascode current mirror with a voltage reference I, that supplies the bias voltage at the gate of transistor $M3$. Additionally, the recognition algorithm recognises a level shifter bank consisting of transistors $M1$, $M3$ and $M5$, which is wrong. Neither $M1$ and $M3$ nor $M1$ and $M5$ form a level shifter. However, this conflict cannot be resolved on hierarchy level 1. If the cascode pair was generally preferred over the level shifter, many level shifters could not be recognised, since a cascode pair appears quite frequently. On the other hand, if the level shifter was preferred over the cascode pair, the cascode pair $(M5, M6)$ could not be recognised and the wide swing cascode current mirror would not be recognised at all. The same type of conflict exists between level shifter $(M1, M3)$ and voltage reference II $(M3, M4)$. Hence, the only way to obtain the correct recognition result is to accept the level shifters, the voltage reference I, and the cascode pair first, and remove the wrongly detected level shifters afterwards.

Figure 4.6 shows an extract of a bipolar operational amplifier with a buffered current mirror and another transistor $Q4$ that is not supposed to be part of any module. Transistors $Q2$ and $Q3$ are recognised correctly as Darlington configuration II. Transistors $Q1$ and $Q4$ however, are wrongly recognised as a differential pair. As a consequence, the buffered current mirror consisting of $Q1$, $Q2$, and $Q3$ cannot be recognised any more,

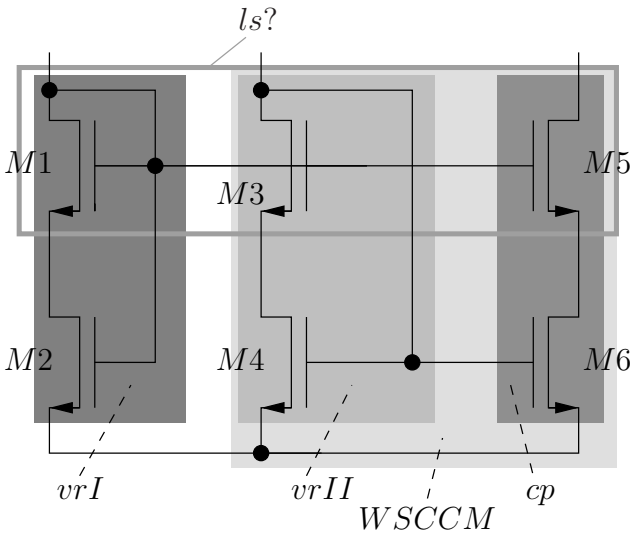

Figure 4.5: Wide Swing Cascode Current Mirror and Voltage Reference I

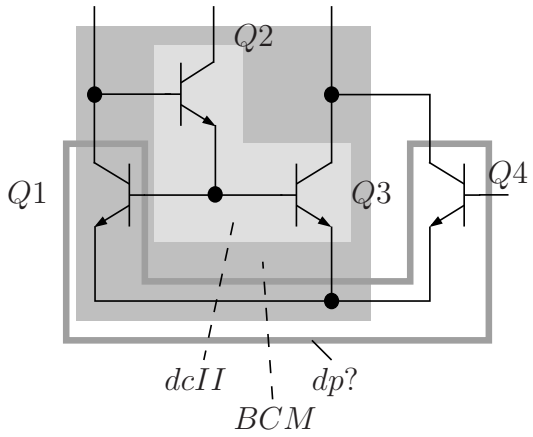

Figure 4.6: Buffered Current Mirror and Differential Pair

since $Q1$ has already been recognised as part of a differential pair. A resolution of this conflict during the recognition process is impossible, because the buffered current mirror is located on hierarchy level 2 in the library in Figure 3.2, but like all transistor pairs, the differential pair is located on level 1. Hence, unless transistor $Q1$ is not recognised as part of any transistor pair, it can never be recognised as part of the buffered current mirror. For that reason, there must be a way to enable the recognition of the buffered current mirror even when one of its transistors has been recognised as part of a different module. Hence, the differential pair and the buffered current mirror have to be accepted first, and the wrongly recognised differential pair including $Q1$ has to be removed after the recognition process.

The assignment of the sizing rules of both a differential pair and a buffered current mirror would be very problematic here. For $Q1$ and $Q3$, the sizing rules for a simple current mirror apply. One rule (3.53) forces the difference between the collector-emitter

voltages of $Q1$ and $Q3$ to be low. Rules (3.63) and (3.65) however, force the difference between the collector-emitter voltages and the base-emitter voltages between $Q1$ and $Q4$ which does not belong to the buffered current mirror to be low. The assignment of these sizing rules could result in a very small or empty feasible region in this case.

These examples show that the first question cannot be answered yes. The second question however, can be answered yes for the modules given in Figure 3.2. For this purpose, a so-called dominance relation has been developed in this thesis and tested on a number of operational amplifiers. This relation contains information about modules dominating other modules. This means that if a module was recognised as part of more than one module and one of these modules dominates the others, the other modules will be removed after the recognition process. In the following, a few examples on modules dominating others are given, including those in Figures 4.5 and 4.6.

In the example in Figure 4.5, the wide swing cascode current mirror has to dominate the level shifter to prevent the wrong recognition of a level shifter. Thus, if a transistor that has been recognised as part of a level-shifter is also part of a wide swing cascode current mirror, the level shifter will be removed after the recognition process.
To resolve the recognition conflict in Figure 4.6, the buffered current mirror ranks before the differential pair. Hence, if a transistor that has been recognised as part of a differential pair is also recognised as part of a buffered current mirror – either as single transistor or as part of the Darlington configuration II that is contained in the buffered current mirror – the differential pair will be removed from the recognition result.

A more complex case where the dominance between modules in general is not enough is the recognition conflict between a level shifter and a current mirror load. Sub-module 1 of a current mirror load may be part of a level shifter, sub-module 2 must not be part of a level shifter. The connectivity is the same, however. For instance, in the folded cascode circuit in Figures 5.1 and 4.4, $MP2$ and $MP4$ form a level shifter, but $MP2$ and $MP3$ do not. Since $MP3$ and $MP4$ form a current mirror load, it is not sufficient to define that the current mirror load dominates the level shifter. This way, both level shifters would be removed after the recognition process. Hence, it has to be taken into account which sub-module the conflict refers to. For instance, only sub-module 2 of the current mirror load dominates the level shifter, but sub-module 1 does not.

The other conflicts in Figure 4.4 can be resolved similarly. All wrongly detected differential pairs are also part of current mirrors. Hence, the differential pair is dominated by all kinds of current mirrors.
The recognition of level shifter $ls6$ can be prevented by ranking the voltage reference II before the level shifter. The recognition of $ls7$ and $ls8$ is prevented by ranking sub-module 2 of the wide swing cascode current mirror, i.e., the cascode pair, before the level shifter. Ranking the cascode pair alone before the level shifter would not work, since there are usually many cascode pairs detected easily in every circuit which would prevent the recognition of many level shifters.

These are just some of the rules that can be derived by examining practical circuits. More can be derived in a similar way. In addition, some rules can be established by

taking the connectivity of the pair-modules into account. For instance, the connectivity of the two transistors in a cross-coupled pair is very unique. Hence, it is very unlikely that one of the two transistors is meant to be part of another pair-module. Thus, the cross-coupled pair can be assigned a high preference.

All the rules are stored in a so-called "dominance relation" S. This homogeneous relation and its elements $s \in S$ are defined as follows:

$$S \subseteq \big(YS \times \{1,2\}\big)^2; \qquad s := \big((y,i),(z,j)\big) \tag{4.10}$$

Each tuple $s \in S$ consists of two tuples itself. The first component of each of these tuples is a structural type, i.e., $y, z \in YS$. The second component is either the natural number 1 or 2, i.e., $i, j \in \{1,2\}$. It denotes whether sub-module 1 or 2 is referred to. If $\big((y,i),(z,j)\big) \in S$, then a module that is the ith sub-module of another module of structural type y is dominated by a module that is the jth sub-module of another module of structural type z. For instance, one element in S is $\big((dp,2),(BCM,1)\big)$. This means that sub-module 1 of the buffered current mirror dominates sub-module 2 of the differential pair. Hence, a differential pair whose second transistor has also been recognised as part of a buffered current mirror has to be removed after the recognition process.

To avoid contradictions S has to be an order relation. Since it is not necessary to include reflexive pairs in S, i.e., no module dominates itself, S is defined as a strict order relation. This means that S is both transitive and asymmetric. The former ensures that if module 'A' dominates module 'B', and 'B' dominates 'C', then 'A' also dominates 'C'. The latter ensures that cycles are prevented. Hence, a case where 'A' dominates 'B', 'B' dominates 'C', and 'C' dominates 'A' cannot occur.

A descriptive graphic representation of a strict order relation is a Hasse diagram which arises from the arrow diagram of the relation by omitting all transitive edges, i.e., $H(S) := S - S^2$. The Hasse diagram of S is depicted in Figure 4.7. The arrow on the right indicates the direction of the arrows in the original arrow diagram. Considering transistor pairs, it can be seen that those with many connections are found at the top. The more connections a transistor pair has got, the more likely it is that these two transistors do actually fulfil the function of the module that their connectivity represents. Hence, due to its unique structure, the cross-coupled pair is located above the other transistor pairs in the library in Figure 3.2, which means it dominates all other transistor pairs. The voltage references I and II are located right beneath it. It can be seen that not between every arbitrary pair of modules, a connection exists and that sometimes only sub-module 1 or 2 is mentioned, but not both. For instance, the current mirrors that consist of at least three transistors are not dominated by the cross-coupled pair or voltage references, since these current mirrors also have many inter-connections and hence, a conflict is very unlikely. Similarly, sub-module 1 of a simple current mirror does not dominate the differential pair. The reason is the intra-connectivity which often prevents a wrong recognition result itself. For instance, sub-module 1 of a simple current mirror is a transistor with its gate and drain or base and collector connected. A

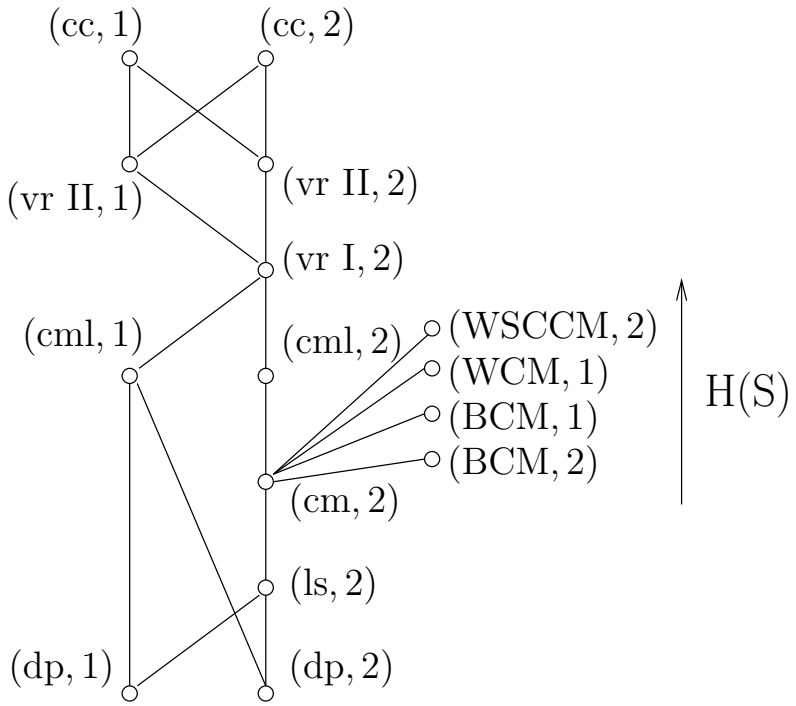

Figure 4.7: Hasse Diagram of Dominance Relation S

transistor with such an intra-connection can never be recognised as part of a differential pair since none of the two transistors of a differential pair has got any intra-connections. The cascode pair is not included in S since it does not produce any sizing rules on its own. Furthermore, the main purpose of S is to avoid the assignment of non-applicable sizing rules, not the detection of design faults. Therefore, S was compiled such that it contains as few pairs as possible.

To formalise the dominance of modules over others, relations R_1 and R_2 that were first introduced with the recognition algorithm in Figure 4.2 are needed. For each module m_μ that has been recognised as part of a module m_λ, an ordered pair (m_μ, m_λ) is stored in R_1 if m_μ is sub-module 1 of m_λ, or in R_2 respectively if m_μ is sub-module 2 of m_λ. With R_1, R_2, $R := R_1 \cup R_2$ and S, the following propositional form to check if there are dominated modules in M has been set up as a rule. If it is false, there is at least one module in set M that is dominated by another one and therefore has to be removed from the overall recognition result.

$$\underset{m_\mu \in M}{\forall} \; \underset{m_\kappa, m_\lambda \in des(m_\mu)_\mathrm{R}}{\forall} \; \underset{i,j \in \{1,2\}}{\forall} \left[\big((m_\kappa.structype, i), (m_\lambda.structype, j) \big) \in S \wedge \underset{x \in des^*(m_\mu)_\mathrm{R}}{\exists} (x, m_\lambda) \in R_j \longrightarrow \neg \underset{y \in des^*(m_\mu)_\mathrm{R}}{\exists} (y, m_\kappa) \in R_i \right] \tag{4.11}$$

In (4.11), $des(m_\mu)_R$ denotes the set of descendants of m_μ in R, i.e., all modules, m_μ is contained in – either as direct sub-module or as part of another sub-module. Addi-

tionally, $des^*(m_\mu)_R := des(m_\mu)_R \cup \{m_\mu\}$, i.e., $des^*(m_\mu)$ is the set of all modules m_μ is contained in and m_μ itself. The rule reads: For every module $m_\mu \in M$ and all modules m_κ and m_λ it is contained in, and all $i, j \in \{1, 2\}$, it holds that if $(m_\kappa.structype, i)$ is dominated by $(m_\lambda.structype, j)$ and there is at least one module x in $des^*(m_\mu)$ that is sub-module j of m_λ, there exists no module y in $des^*(m_\mu)$ that is sub-module i of m_κ. Hence, if a module m_κ exists in that case, the rule is violated, i.e., the propositional form is false and m_κ has to be removed from the overall recognition result. In addition, all other modules on higher hierarchy levels it is contained in have to be removed too, since module m_κ was one part of them which is then no longer existent in the list of modules.

Using relation S and the rule (4.11), an algorithm to resolve recognition conflicts has been set up. This algorithm is applied after the recognition process. It collects all modules from set M that are not "dominated" by any other modules into a new set F containing the final recognition result. The algorithm is depicted in Figure 4.8. First, set F is instantiated as a copy of M, and relation R as the union of R_1 and R_2 (line 1). The outer loop (line 2) iterates over a counter μ from 1 to the cardinality of set M. Then all modules m_μ in F are checked (line 3). At the beginning, F is equal to M, but with every conflict occurring, modules are removed from F. These modules do not have to be checked any more because if a module was removed from F, all the modules it was contained in had been removed from F, too. For the current module m_μ, set E of m_μ itself and all modules in F that m_μ is contained in as a sub-module is compiled (line 4). For this purpose, R^+ – the transitive closure of R [GT96] – is used. The expression $m_\mu R^+ m_{\mu'}$ is true for all m_μ that are contained in $m_{\mu'}$, i.e., for all descendants $m_{\mu'}$ of m_μ. If $m_{\mu'}$ is no longer contained in F, this means that it has been removed in one of the previous runs of the outer loop due to a conflict and is therefore no longer relevant for resolving new conflicts. In line 5, all possible pairs of different modules in E without the identity relation I_E, i.e., no pairs that consist of the same element of E twice are assigned to U. The following three loops (lines 6-8) iterate through each possible pair (u, v) with $u \neq v$ and both u and v contained in U and R_1 or R_2. The second element of u or v, which is $u^{(2)}$ or $v^{(2)}$, respectively, denotes the module that module $u^{(1)}$ or $v^{(1)}$, is contained in. To find out if m_μ is allowed to be contained in both $u^{(2)}$ and $v^{(2)}$, it is checked in line 9, if the structural type of $v^{(2)}$ dominates the structural type of $u^{(2)}$ considering if their sub-modules $u^{(1)}$ and $v^{(1)}$ are sub-module no. 1 or 2, i.e., if u and v are in R_1 or R_2. This is done by checking if $\big((u^{(2)}.structype, i), (v^{(2)}.structype, j)\big) \in S$. If this is the case, module $u^{(2)}$ and all modules that $u^{(2)}$ is contained in are removed from F (line 10).

Example

The resolution of recognition conflicts will be illustrated using the example of the folded cascode circuit with the wrongly recognised modules in Figure 4.4. Two conflicts and their resolution will be illustrated in detail. The constants t and f represent the logical values *true* and *false*.

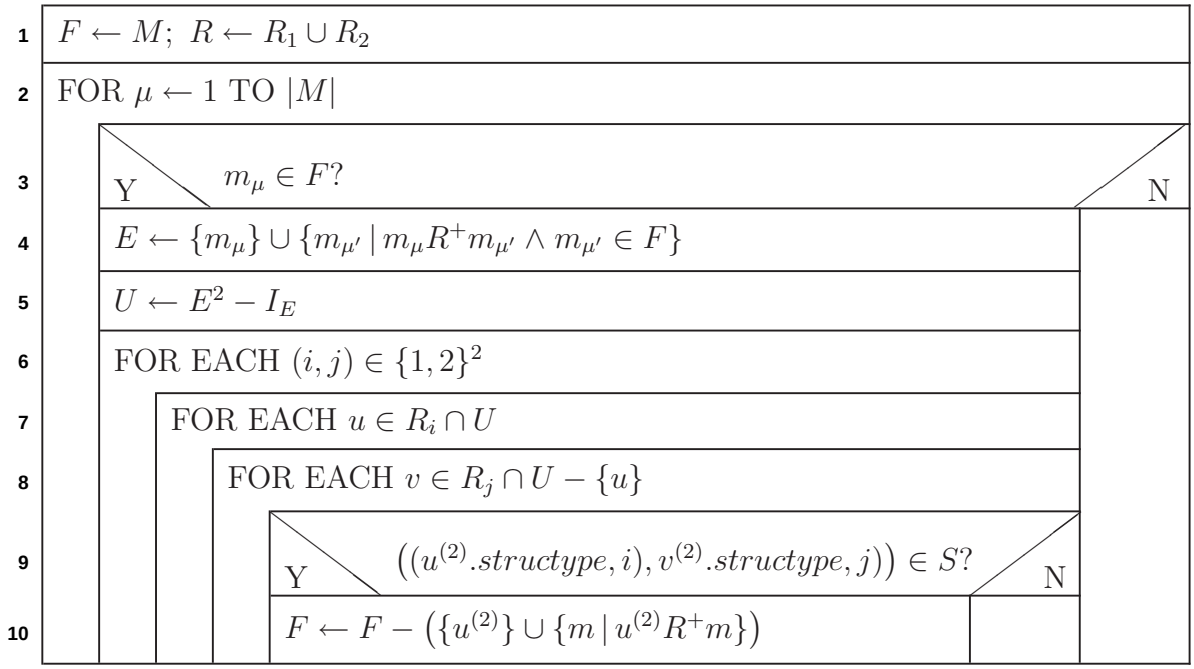

Figure 4.8: Algorithm for Arbitration of Assignment Ambiguities

For convenience, only transistors $MP2$, $MP4$, $MP5$, $MP6$, $MP7$, and $MP8$ are considered. With this limitation, the set of modules looks as follows after the recognition process:

$$M = \{MP2, MP4, MP5, MP6, MP7, MP8, vrII1, ls4, ls6, ls7, cp1, WSCCM1\}.$$

Figure 4.9 shows the arrow diagrams of relations R_1 and R_2. As lists of ordered pairs, relations R_1, R_2, and $R^+ := (R_1 \cup R_2)^+$ are given by:

$$\begin{aligned}
R_1 =& \{(MP2, ls4), (MP2, ls6), (MP2, ls7), (MP6, vrII1), (MP8, cp1),\\
& (vrII1, WSCCM1)\}\\
R_2 =& \{(MP4, ls4), (MP5, ls6), (MP5, vrII1), (MP6, ls6), (MP7, cp1),\\
& (MP8, ls7), (cp1, WSCCM1)\}\\
R^+ =& \{(MP2, ls4), (MP2, ls6), (MP2, ls7), (MP6, vrII1), (MP8, cp1),\\
& (vrII1, WSCCM1), (MP4, ls4), (MP5, ls6), (MP5, vrII1), (MP6, ls6),\\
& (MP7, cp1), (MP8, ls7), (cp1, WSCCM1), (MP5, WSCCM1),\\
& (MP6, WSCCM1), (MP7, WSCCM1), (MP8, WSCCM1)\}
\end{aligned}$$

There are two recognition conflicts to resolve: The level shifters $ls6$ and $ls7$ have been recognised wrongly. The conflict involving $ls6$ will be treated first because it occurs between two modules on the same hierarchy level and is therefore easier to resolve than the one involving $ls7$.

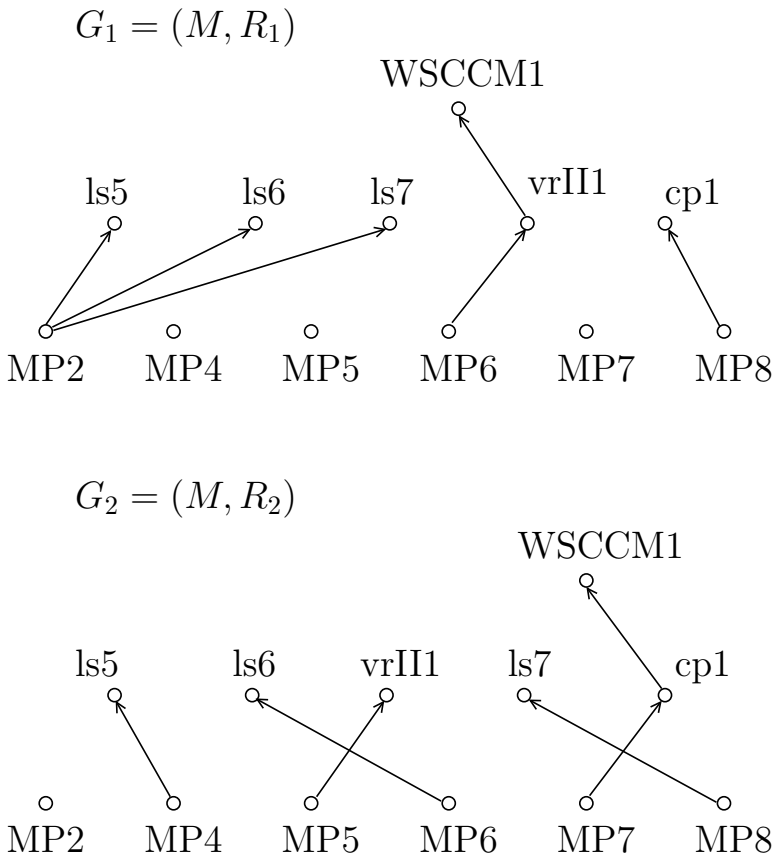

Figure 4.9: Relations R_1 and R_2

First, it is shown that the rule (4.11) is violated: Let $m_\mu = MP6$, $m_\kappa = ls6$, $m_\lambda = vrII1$, and $(i, j) = \{2, 1\}$.
It is obtained:

$$\big((ls, 2), (vrII, 1)\big) \in S \wedge \underset{x \in des^*(MP6)_\mathrm{R}}{\exists} (x, vrII1) \in R_1 \longrightarrow \neg \underset{y \in des^*(MP6)_\mathrm{R}}{\exists} (y, ls6) \in R_2$$

with

$$des^*(MP6) = \{MP6, ls6, vrII1, WSCCM1\}.$$

The expression $\big((ls, 2), (vrII, 1)\big) \in S$ is true which is apparent from Figure 4.7.
For $x = MP6$,

$$\underset{x \in des^*(MP6)_\mathrm{R}}{\exists} (x, vrII1) \in R_1 \Longleftrightarrow t$$

since $(MP6, vrII1) \in R_1 \Longleftrightarrow t$.
But for $y = MP6$,

$$\neg \underset{y \in des^*(MP6)_\mathrm{R}}{\exists} (y, ls6) \in R_2 \Longleftrightarrow f$$

since $(MP6, ls6) \in R_2 \Longleftrightarrow t$.
Hence, $t \wedge t \longrightarrow f \Longleftrightarrow f$ is obtained [GT96], which means that the rule is violated.

Next, it is shown how this conflict is resolved using the algorithm in Figure 4.8. With relations R_1, R_2, and R^+, relation E looks as follows when $MP6$ is examined within the outer loop.

$$E = \{MP6, vrII1, ls6, WSCCM1\}.$$

Thus, relation U is instantiated as

$$\begin{aligned} U = \{&(MP6, vrII1), (MP6, ls6), (MP6, WSCCM1), (vrII1, ls6), \\ &(vrII1, WSCCM1), (ls6, WSCCM1), (vrII1, MP6), (ls6, MP6), \\ &(WSCCM1, MP6), (ls6, vrII1), (WSCCM1, vrII1), (WSCCM1, ls6)\} \end{aligned}$$

Hence, $R_1 \cap U$ and $R_2 \cap U$ look as follows:

$$R_1 \cap U = \{(MP6, vrII1), (vrII1, WSCCM1)\}, \qquad R_2 \cap U = \{(MP6, ls6)\}$$

For $(i, j) = (2, 2)$, $R_j \cap U - \{u\}$ is empty, so the if-clause inside the three loops will not be executed. For $(i, j) = (1, 1)$, $R_j \cap U - \{u\}$ is $\{(MP6, vrII1)\}$ or $\{(vrII1, WSCCM1)\}$. In this case, the if-clause will be executed with $u^{(2)} = vrII1$ and $v^{(2)} = WSCCM$ or $u^{(2)} = WSCCM$ and $v^{(2)} = vrII1$. In both cases, no conflict occurs since the wide swing cascode current mirror does not dominate the voltage reference II or the other way round.

For $(i, j) = (1, 2)$ or $(i, j) = (2, 1)$, $R_j \cap U - \{u\}$ is again not empty, so the if-clause inside the three loops will be executed. For $(i, j) = (1, 2)$, it is checked if $\big((vrII, 1), (ls, 2)\big) \in S$ or if $\big((WSCCM, 1), (ls, 2)\big) \in S$ which is both false because the level shifter does not dominate the voltage reference II.
However, for $(i, j) = (2, 1)$,

$$R_2 \cap U = \{(MP6, ls6)\} \text{ and } R_1 \cap U - \{u\} = \{(MP6, vrII1), (vrII1, WSCCM)\}.$$

Since $\big((ls, 2), (vrII, 1)\big) \in S$ is true, the voltage reference II $vrII1$ is the dominating module and the level shifter $ls6$ will be removed from set F of all remaining modules.

The conflict involving $ls7$ occurs between two modules on different hierarchy levels, namely the level shifter $ls7$ and the wide swing cascode current mirror $WSCCM1$. Rule (4.11) is violated again. Let $m_\mu = MP8$, $m_\kappa = ls7$, $m_\lambda = WSCCM1$, and $(i, j) = (1, 2)$.
It is obtained:

$$\big((ls, 2), (WSCCM, 2)\big) \in S \wedge \underset{x \in des^*(MP8)_\mathrm{R}}{\exists} (x, WSCCM1) \in R_2 \longrightarrow \neg \underset{y \in des^*(MP8)_\mathrm{R}}{\exists} (y, ls7) \in R_2$$

with

$$des^*(MP8) = \{MP8, ls7, cp1, WSCCM1\}.$$

The expression $\big((ls, 2), (WSCCM, 2)\big) \in S$ is true which is apparent from Figure 4.7.
For $x = cp1$,

$$\underset{x \in des^*(MP8)_{\mathrm{R}}}{\exists} (x, WSCCM1) \in R_2 \Longleftrightarrow t$$

since $(cp1, WSCCM1) \in R_2 \Longleftrightarrow t$.
But for $y = MP8$,

$$\neg \underset{y \in des^*(MP8)_{\mathrm{R}}}{\exists} (y, ls7) \in R_2 \Longleftrightarrow f$$

since $(MP8, ls7) \in R_2 \Longleftrightarrow t$.
Hence, $t \wedge t \longrightarrow f \Longleftrightarrow f$ is obtained once again which means that the rule is violated.

When $MP8$ is examined, set E is instantiated as

$$E = \{MP8, cp1, ls7, WSCCM1\}.$$

Thus, relation U is instantiated as

$$\begin{aligned} U = \{&(MP8, cp1), (MP8, ls7), (MP8, WSCCM1), (cp1, ls7), \\ &(cp1, WSCCM1), (ls7, WSCCM1), (cp1, MP8), (ls7, MP8), \\ &(WSCCM1, MP8), (ls7, cp1), (WSCCM1, cp1), (WSCCM1, ls7)\} \end{aligned}$$

Thus, $R_1 \cap U = \{(MP8, cp1)\}$ and $R_2 \cap U = \{(MP8, ls7), (cp1, WSCCM1)\}$. Hence, for all i and j except $(i, j) = (1, 1)$, the inner if-clause of the algorithm in Figure 4.8 is executed. But as the cascode pair is not included in S, the only case where a conflict occurs, is $i = j = 2$. In that case, for $u = (MP8, ls7)$ and $v = (cp1, WSCCM1)$, it is checked if $\big((ls, 2), (WSCCM, 2)\big) \in S$. This is true. Hence, the level shifter $ls7$ will be removed from F.

4.2.2 Uncertain Building Blocks

The function of a detected module is not always "certain", i.e., it is not clear if the module is also a building block in the given circuit. The resolution of recognition conflicts only removes modules that are dominated by others. But even if there is no recognition conflict, it cannot be guaranteed that a module actually fulfils the function that is assigned to it. The reason is once again that the recognition algorithm only considers the connectivity between modules. Hence, modules that do not have many connections, like a differential pair that only has one between the transistors' source or emitter pins respectively, are easily recognised. For this reason, these modules will be classified as uncertain. That means that it is up to the designer to decide whether the corresponding sizing rules should be assigned or not. The single transistor which has not been recognised as part of any larger module is also classified uncertain since its operation region cannot be determined automatically in that case.

Some modules can become uncertain within the resolution of recognition conflicts. For instance, if a differential pair was removed from the sets of recognised modules, the operation region of the transistors it consisted of is not defined any more.

The following building blocks are classified as "uncertain" if they have not been recognised as part of a larger building blocks.

Single transistor: The single transistor is the only module that does not consist of two sub-modules. Since the concept of building blocks and sizing rules is based on the connectivity of pairs of modules, no sizing rules can be assigned to a module that is a single circuit element unless it has been recognised as part of a building block. For a single transistor, this means, the required operating region of a transistor cannot be determined before it has been recognised as part of a larger building block.

Differential pair: Based on the connectivity, two transistors form a differential pair if only their source (or emitter) pins are connected. Hence, many differential pairs are usually detected during the recognition process. Even though a lot of them are removed again due to recognition conflicts, some differential pairs that are not meant to be differential pairs might remain. Since the rules for a differential pair are very strict, all differential pairs that were not recognised as part of a larger building block, are classified as uncertain.

Darlington configuration II: Similar to the differential pair, this building block may be recognised quite often since the only connection between the two involved transistors is between the emitter of transistor (1) and the base of transistor (2). For this reason, this building block is also always classified as uncertain as long as it was not recognised as part of a larger building block.

Cascode pairs are never classified as uncertain since they do not produce any sizing rules on their own.

A module is classified as uncertain when it is of one of the structural types mentioned above and it is not contained in any larger building block. Mathematically, this can be formalised as follows:

$$\underset{m_\mu \in F}{\forall} \left[m_\mu.\text{structype} \in \{trans, dcII, dp\} \wedge \left(\neg \underset{m_\lambda \in F}{\exists} m_\mu R \text{m}_\lambda \right) \longrightarrow Z m_\mu \right] \quad (4.12)$$

$$\text{with } Zx :\Longleftrightarrow x \text{ is uncertain}$$

Only modules in set F are considered, i.e., modules that were not removed due to recognition conflicts. Figure 4.10 shows the algorithm to classify building blocks as uncertain. At the beginning, set Q of uncertain building blocks is instantiated as the empty set. Since all building blocks that can be classified as uncertain are located on the two lowest hierarchy levels, the algorithm only checks modules on these levels, i.e., modules in sets M_0 and M_1. The algorithm runs top-down through these two hierarchy levels. For instance, if a differential pair is classified as uncertain, the transistors it consists of also have to be classified as uncertain. But if these transistors were examined before the differential pair they would not be classified as uncertain since they were still part of that differential pair that is included in F but not yet in Q. In the if-clause modules in F are examined. If the module that is examined is not part of any other

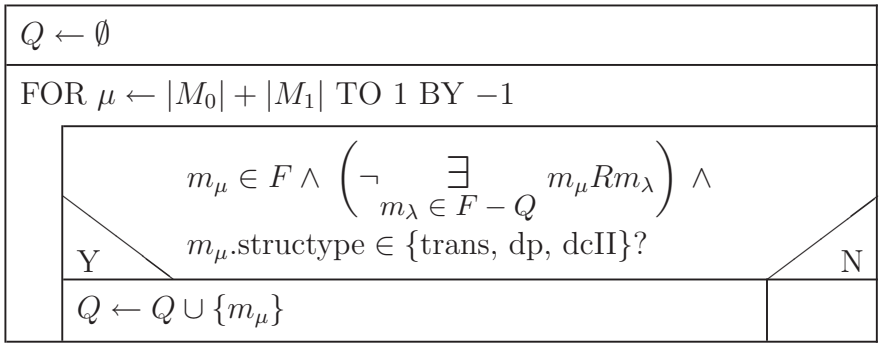

Figure 4.10: Algorithm to Determine "Uncertain Building Blocks"

module that is in F but not in Q and its type is transistor, differential pair or Darlington configuration II, it is classified as uncertain and it will therefore be added to Q.

Set Q is available to the designer for further actions, like assigning the respective sizing rules to the modules stored in Q.

4.3 Preparation of Circuit Netlists

In the two previous sections, structure recognition including the arbitration of assignment ambiguities was presented in detail. It was not described how the circuit netlist has to be prepared to enable structure recognition as it was presented in Chapter 4.1.

The preparation of the netlist in this thesis covers the following:

a) Treatment of circuit elements connected in parallel as one circuit element

b) Treatment of transistors with a resistor connected to their emitter or source pin as one circuit element

c) Removal of circuit elements like power-down transistors (optional)

Item a) is important to avert multiple recognition of the same module due to circuit elements connected in parallel. For instance, without preparation of the netlist, a simple current with m transistors in parallel on the driving side and n transistors in parallel on the driven side, the structure recognition algorithm would recognise $m \cdot n$ simple current mirrors instead of one.

The requirement in b) issues from Chapter 3.2, where it was mentioned that especially in bipolar transistor circuits, resistors are often connected to the emitter pins of transistors. Without preparation of the netlist, the resistor would prevent the recognition of a module in most cases. For instance, none of the modifications of the simple current mirror with additional resistors presented in 3.4.2.1 could be recognised.

The solution to items a) and b) that has been chosen in this thesis is to transform the physical netlist into a so-called meta-netlist. The structure recognition algorithm then

operates on this meta-netlist. During the transformation, physical circuit elements are transformed into meta-elements. To transform the physical netlist into a meta-netlist, a technology file is required that contains information about the netlist syntax that is simulator-specific and all types of circuit elements or models, e.g., whether transistors are scalable or not.

The meta-elements contain the type of the original element as a parameter. Thus, when an NMOS transistor is transformed into a meta-element, that parameter will be assigned to "NMOS transistor". Identical devices connected in parallel are replaced by a single meta-element. The number of devices is also stored as a parameter of the meta-element. It is necessary though, that all devices in parallel are not just of the same type. They also have to have exactly the same design parameters assigned to them. This means that, for instance, two NMOS transistors connected in parallel but with different design parameters for the widths will not be transformed into a single meta-element. Finally, every transistor with a resistor connected to its emitter or source pin will be transformed into a single-module that stores the information "transistor-resistor pair" in it. However, since this transformation reduces the number of nets by one, it will only be performed if no other device is connected to that net.

This solution also incorporates an advantage to accelerate the recognition algorithm. When iterating over all elements of library L, modules of equivalent structural type but different transistor type – with or without additional resistors at the source or emitter pins – can be recognised in one go. For instance, all types of simple current mirror (NMOS, PMOS, npn, pnp) and the modifications presented in 3.4.2.1 can be recognised in one run of the outer loop of the recognition algorithm in Figure 4.2. The library in Figure 3.2 is organised such that all elements are given defined by the connections of their sub-modules and they have a list of allowed transistor types they may consist of (see Chapter 3.2).

Item c) is an optional interactive step. The user can remove circuit elements, too. One example are power-down transistor which can be used to switch off parts of a circuit. These transistors can also distort the recognition result. The removal of these transistors requires the name of the power-down net and – if present – the inverted power-down net in the circuit. After the user has provided this information, all transistors whose gate is connected to one of these two nets will be shorted or opened, depending on whether they are of type NMOS or PMOS and whether they are connected to the power-down net or the inverted power-down net.

Figure 4.11 shows three transformations using the example of three current mirrors: an NMOS simple current mirror with one transistor on each side, an npn simple current mirror with n transistors on the right, and a Widlar current mirror. However, the structure on the right is always the same: There are two meta-elements X_1 and X_2 connected as a simple current mirror. These elements carry the information about the physical devices as denoted in the figure. This way, all these structures will be recognised correctly – (a) and (b) as simple current mirrors, (c) as Widlar current mirror, and the appropriate sizing rules can be assigned correctly to all transistors involved.

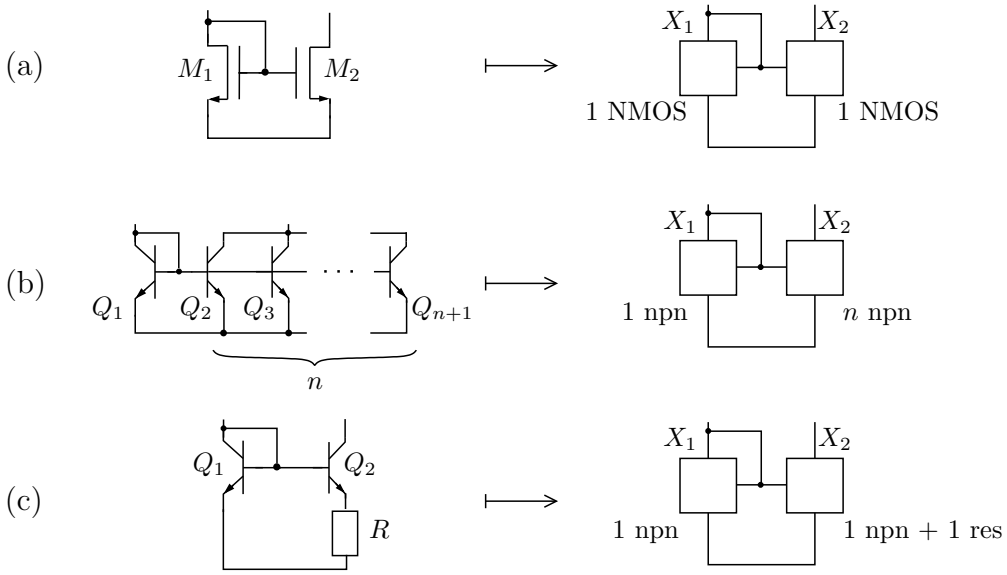

Figure 4.11: Different Transformations from the Physical Netlist into the Meta-Netlist

Generate meta-netlist by transforming physical circuit elements into meta-elements
Perform structure recognition (algorithm in Figure 4.2)
Perform arbitration of assignment ambiguities (algorithm in Figure 4.8)
Perform collection of uncertain building blocks (algorithm in Figure 4.10)

Figure 4.12: Overall Structure Recognition Process

In the example in Figure 4.11(a), an NMOS simple current mirror is transformed into a simple current mirror with meta-modules X_1 and X_2. The meta-modules contain the information about the number of transistors connected in parallel, which is one here, and the type of the transistors, which is NMOS.

Figure 4.11(b) shows an npn simple current mirror with one transistor on the driving side and n transistors on the driven side. In the meta netlist, there are just two circuit elements X_1 and X_2 carrying the information about the transistor type which is npn and the number of transistors connected in parallel, which is 1 on the left and n on the right.

A Widlar current mirror is given in Figure 4.11(c). The according transformation looks as follows: Transistor Q_1 is transformed into circuit element X_1 which contains one npn transistor. Transistor Q_2 and resistor R are combined into one circuit element X_2 that contains one npn transistor and one resistor.

Including this preparatory step, the complete process of structure recognition is summarised in Figure 4.12.

Using the concept of transforming the physical netlist into a meta-netlist allows the treatment of "number of physical devices connected in parallel" as well as "type of the physical devices" as parameters. This can also be used for new ways of circuit optimisation. Instead of scaling transistors, transistors can be replaced by others. And not just transistors, even current mirrors or other modules could be replaced by others during circuit optimisation. A few aspects of this will be discussed in Chapter 5.4.

4.4 On the Complexity of the Structure Recognition Algorithm

The structure recognition algorithm detects pairs of modules by comparing all possible candidates to the modules in the library. In the worst case – which is not relevant in practice – a circuit consists of transistors of all the same type, and every possible combination of modules produces a new module. With n as the number of transistors in a circuit, and each possible pair of transistors forming a module given in the library, the number of comparisons c_0 on hierarchy level 0 is

$$c_0 = \binom{n}{2}. \tag{4.13}$$

Since no modules are discarded before the end of the whole recognition process, all modules are then considered for building blocks on hierarchy level 2. For building blocks consisting of three transistors, all transistor pairs have to be checked with all transistors that are not part of the respective pair, hence the number of comparisons for these building blocks is $(n-2) \cdot c_0$. The total number of comparisons c_1 on hierarchy level 1 is

$$c_1 = (n-2) \cdot c_0 + \binom{c_0}{2}. \tag{4.14}$$

On hierarchy level 3, there are only differential stages. If all two-transistor modules could form a differential stage with all other modules that consist of at least two transistors, the number of comparisons on level 2 would be

$$c_2 = \binom{c_0}{2} + c_0 \cdot c_1 \tag{4.15}$$

As a function of n, the total number of comparisons is given by

$$\sum_{i=0}^{2} c_i = \frac{1}{16}(n^6 + n^5 - 11n^4 + 23n^3 - 30n^2 + 16n). \tag{4.16}$$

Hence, in theory, the number of comparisons grows with n^6. In practice, this growth is much weaker as the results show. The main reason for this is that not every possible pair of modules is checked for whether it forms one of the building blocks in the library

Circuit	Number of transistors of type NMOS	PMOS	npn	pnp	in total
Figure 5.1	12	10	0	0	22
Figure 5.2	0	4	4	3	11
Figure 5.3	21	21	0	0	42
Figure 5.4	0	0	16	11	27
Figure 5.5	2	3	6	5	16
MI refbuffer	10	13	0	0	23

Table 4.1: Number of Transistors Contained in Six Example Circuits

Circuit	Number of comparisons on level 0	1	2	in total	theoretical maximum
Figure 5.1	111	42	27	180	$7.26 \cdot 10^6$
Figure 5.2	15	7	4	26	$1.12 \cdot 10^5$
Figure 5.3	420	205	187	812	$3.49 \cdot 10^8$
Figure 5.4	175	166	33	374	$2.48 \cdot 10^7$
Figure 5.5	29	17	3	49	$1.07 \cdot 10^6$
MI refbuffer	123	39	46	208	$9.48 \cdot 10^6$

Table 4.2: Number of Comparisons Required to Detect All Building Blocks in the Given Example Circuits

or not. Instead, the algorithm browses through all library elements and checks all pairs of appropriate modules only whether they form the respective building block or not. For instance, it is of no use to check pairs of single transistors when current mirrors consisting of four transistors are searched. And the fact that the differential stage is the only hybrid building block reduces the number of comparisons even further.

Table 4.1 shows the number of transistors partitioned by type contained in six different circuits. The results of structure recognition for these circuits will be presented in Chapter 5. In Table 4.2, the number of comparisons that was actually carried out applying the structure recognition algorithm on these six circuits is given. Four of these circuits contain two different types of transistors, and the total number of comparisons ranges between approximately $0.4n^2$ and $0.5n^2$. Hence, the complexity is only quadratic in practice. In the two BiCMOS operational amplifiers, the number is much smaller since they contain three or four different types of transistors, respectively.

Analog circuits are usually small, but in applications like the recognition of modules for digital circuits, the number of transistors can be much higher, so that the computational effort for structure recognition becomes relevant. In [SS09], the structure recognition approach presented in this dissertation, has been adopted and extended in order to perform structure recognition in digital circuits. In that work, some additional measures have been brought about to ensure that even for large circuits, the time effort

stays acceptable. For instance, parts of the dominance relation are applied during the recognition process, not at the end. In addition, the comparisons are carried out in a way, that characteristic comparisons are checked first. For instance, in analog circuits, simple current mirrors could be checked, by first checking if there's a combination between the drain of the first and the gate of the second transistor, since this connection does not appear in most of the other transistor pairs. If this was not the case, further checks would be obsolete. The results in [SS09] show a quadratic dependence of the computational effort on the number of transistors which coincides with the result in this thesis.

4.5 Summary

In this chapter, an algorithm for structure recognition including the arbitration of assignment ambiguities was presented. The module library in Figure 3.2 contains a single transistor on hierarchy level 0. Modules on hierarchy levels above 0 are always modelled as pairs of other modules. As a consequence, the recognition algorithm has been based on the recognition of pairs of modules.

The basic structure recognition algorithm only checks for the connectivity between two modules to determine if these two modules form another module that is contained in the library in Figure 3.2. As a result, assignment ambiguities may occur in the form of recognition conflicts, such that building blocks are recognised which are mutually exclusive. Additionally, some modules are classified as uncertain building blocks, i.e., it cannot be determined automatically if a module is also a building block.

The computational effort of the structure recognition algorithm is negligible since the number of library elements and the number of transistors in analog circuits is small. In addition, some measures have been taken to reduce the number of comparisons during structure recognition. Experimental results show that there is a quadratic dependence of the number of required comparisons on the number of transistors in the given circuit.

To deal with recognition conflicts, a dominance relation and an algorithm to resolve these conflicts have been developed. The dominance relation formalises the dominance of modules over others. The algorithm in Figure 4.8 processes the recognition result using this relation and removes dominated modules.

In addition, an algorithm to classify building blocks as uncertain has been developed. All uncertain building blocks are stored in a set. Since a decision whether to apply the respective sizing rules or not cannot be done automatically for uncertain building blocks, that set is provided to the user.

In a preparatory step, the netlist is prepared to be able to treat both circuit elements connected in parallel as well as transistors together with resistors connected to their emitter or source pins as one circuit element. Moreover, this allows the recognition of modules that are structurally equivalent but consist of different types of transistors.

This is attained by transforming the physical netlist into a meta-netlist before the actual structure recognition process is being carried out.

Additionally, further circuit elements like power-down transistors can be removed. This requires user input.

Chapter 5

Results

In this chapter, the results of the presented structure recognition method applied on six different analog circuits are shown. To demonstrate the benefit of sizing rules, the optimisation results for three of these six circuits – once with and once without consideration of sizing rules – are presented in detail.

The first circuit is a CMOS folded cascode operational amplifier consisting of twelve NMOS and ten PMOS transistors (Figure 5.1). The second circuit is a simple BiCMOS operational amplifier (Figure 5.2). It contains four PMOS transistors, as well as four npn and three pnp transistors. The third circuit is a CMOS buffer amplifier consisting of 21 NMOS and 21 PMOS transistors (Figure 5.3). Circuit number four is a bipolar operational amplifier containing 16 npn and eleven pnp transistors (Figure 5.4). The fifth circuit is a BiCMOS amplifier for automotive consisting of two NMOS, three PMOS, six npn, and five pnp transistors (Figure 5.5). Finally, the sixth circuit is a CMOS operational amplifier with ten NMOS and 13 PMOS transistors. Due to copyright reasons, this circuit cannot be depicted in this thesis.

For all circuits, the automatic structure recognition was performed. After assigning the sizing rules, the first three of the given circuits were optimised performing feasibility optimisation, nominal optimisation, and design centring. The results show that using sizing rules, the resulting designs were more robust and the overall number of simulations of the optimisation process was usually much smaller. Moreover, the optimisation algorithms often terminated before a result was obtained when no sizing rules were considered.

5.1 Structure Recognition

The modules that were detected in the first five example circuits are shaded and labelled in Figures 5.1–5.5. Detailed results of the automatic structure recognition for all six circuits are collected in Table 5.1. The number of transistors in each circuit is given in

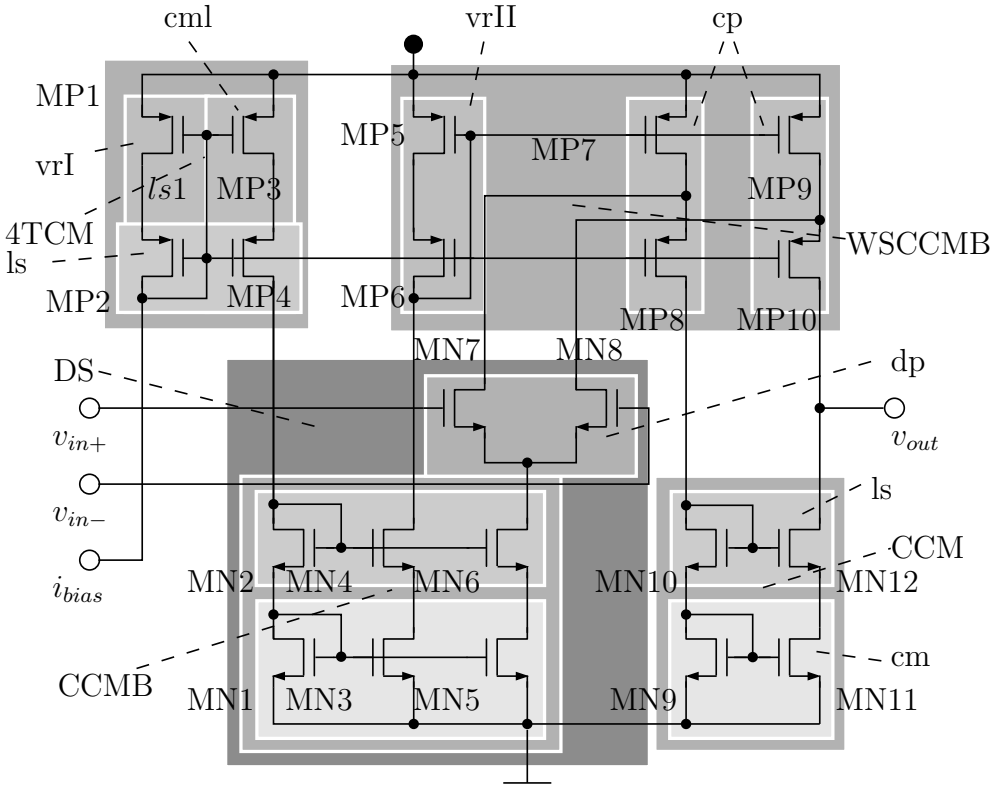

Figure 5.1: Detected Building Blocks of a Folded Cascode Operational Amplifier [AH87]

the second column. In the following columns, the number of detected building blocks, i.e., modules fulfilling a function, for the four levels of hierarchy and in total is listed. The number of building blocks on hierarchy level 0 is equal to the number of transistors if all transistors were recognised as part of a building block, such that an operating region can be determined. For completeness, the number of detected banks in the circuits is shown in the last column of Table 5.1.

These results present the final result of the automatic structure recognition method, including the arbitration of assignment ambiguities. This means, that dominated and uncertain building blocks have already been removed. In addition to the final number of building blocks that were recognised, Table 5.2 shows the number of building blocks that were removed based on the dominance relation S, i.e., that would have been recognised wrongly otherwise. The last column shows the number of uncertain building blocks.

In the CMOS circuits, the number of detected building blocks is a bit lower than twice the number of the transistors. Only few uncertain building blocks remained. In the circuit in Figure 5.3, these are six transistors that have not been recognised as part of any building block. Additionally, about a third of the number of building blocks that was originally recognised is dominated by others and has been removed. In the circuit "MI refbuffer", one transistor has been classified as uncertain. The BiCMOS operational amplifiers in Figures 5.2 and 5.5, are very small and compact. Hence, the number of building blocks removed and classified as uncertain is very low or zero. In the bipolar

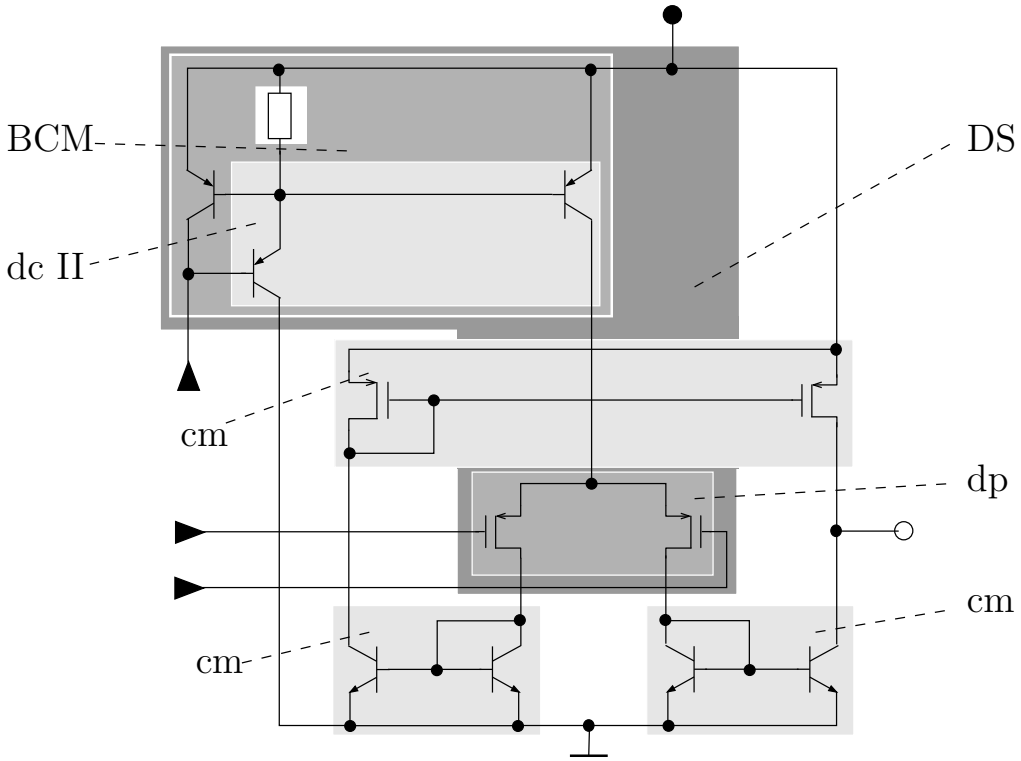

Figure 5.2: Detected Building Blocks of a BiCMOS Operational Amplifier [LS94]

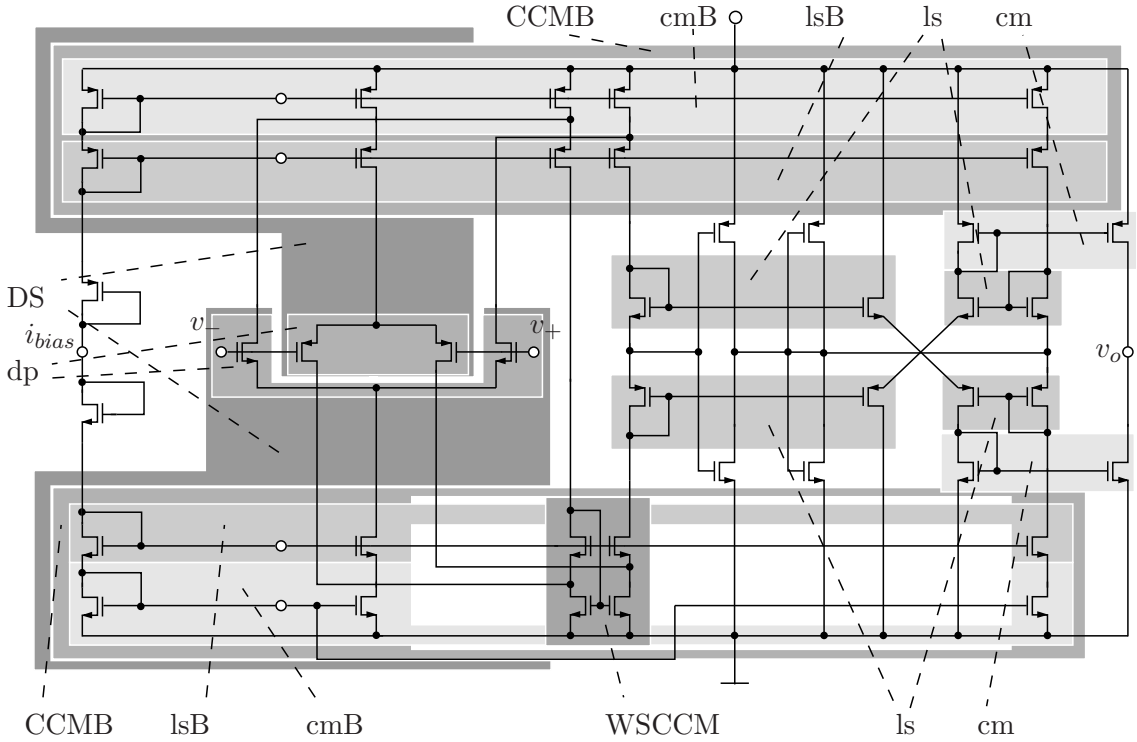

Figure 5.3: Detected Building Blocks of a CMOS Buffer Amplifier [FK87]

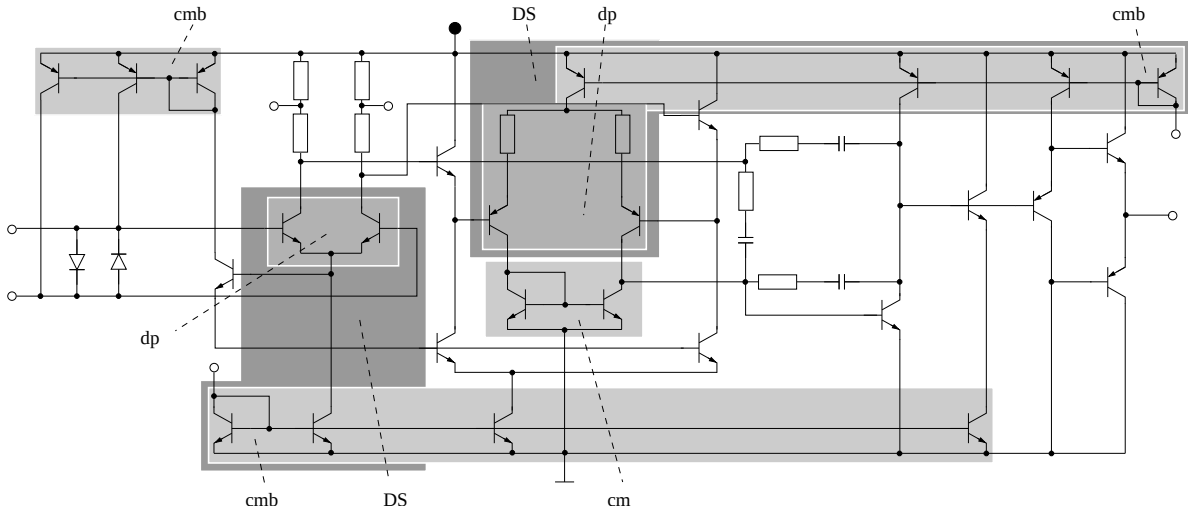

Figure 5.4: Detected Building Blocks of a Bipolar Buffer Amplifier [LS94]

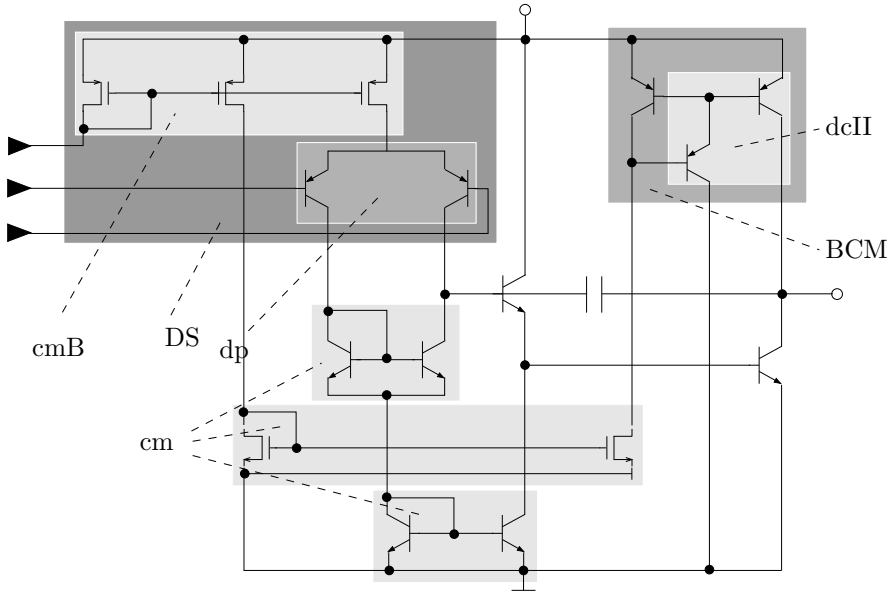

Figure 5.5: Detected Building Blocks of a BiCMOS Operational Amplifier [SGS+04]

operational amplifier "OP-27", ten transistors were not recognised as part of a building block which has lead to a high number of uncertain building blocks in total. The other four uncertain building blocks are of type Darlington configuration II. The results in Table 5.2 clearly show the importance of the arbitration of assignment ambiguities.

5.2 Sizing Rules

Table 5.3 shows the total number of automatically assigned sizing rules for each of the circuits. Although the list of generic sizing rules is fairly small, the overall number of

Circuit	Number of transistors	Number of building blocks on level 0	1	2	3	in total	Number of banks
Figure 5.1	22	22	13	6	1	42	4
Figure 5.2	11	11	5	1	1	18	0
Figure 5.3	42	36	22	11	4	73	6
Figure 5.4	27	17	11	0	2	30	3
Figure 5.5	16	14	7	1	1	23	1
MI refbuffer	23	22	12	4	1	39	2

Table 5.1: Number of Detected Building Blocks for Six Different Operational Amplifiers

sizing rules for each circuit in the examined circuits is quite large. Hence, the automatic construction of these rules on circuit level is of large benefit even for small circuits.

The small number of generic sizing rules is a result of the presented hierarchical building block libraries on transistor-pair level. Every building block inherits the sizing rules of its sub-blocks. In this way, redefinition of existing sizing rules is not necessary. If the generic rules would be established on circuit level, the preparatory effort for analog synthesis would be significantly higher.

Not all sizing rules given as inequalities necessarily become active during the optimisation process though. Some of the sizing rules can be redundant and never become active. A sizing rule is redundant if there are one or more other sizing rules that confine the design parameter space even further. Redundant sizing rules do not effect the optimisation process, since one DC-simulation is enough to determine all sizing rules. They just have to be calculated from the simulation results.

The following section shows the results of circuit sizing for the three circuits given in Figures 5.1–5.3. Based on the result of automatic structure recognition, the list of sizing rules given as inequalities to confine the design space was passed to the optimiser. In this way, a feasibility optimisation could be performed first, in order to find a design point where all sizing rules were fulfilled. The equalities reduce the number of design parameters. They were utilised during the set up of the optimisation. Only the remaining design parameters are passed to the optimiser. With fewer design parameters to adjust, the optimisation process is accelerated.

5.3 Automatic Circuit Sizing

Circuit sizing was performed for the circuits depicted in Figures 5.1–5.3. In the following, the results will be presented for each circuit in detail. As optimisation tool, WiCkeD [Mun09] was used. To find a design point where all sizing rules are fulfilled, a feasibility optimisation was performed first. From that point, nominal optimisation was performed to meet the specifications. The nominal optimisation algorithm in WiCkeD

Circuit	#Final no. of building blocks	#No. of removed building blocks	#No. of uncertain building blocks
Figure 5.1	42	12	0
Figure 5.2	18	1	0
Figure 5.3	73	34	6
Figure 5.4	30	12	14
Figure 5.5	23	0	2
MI refbuffer	39	21	1

Table 5.2: Recognition Results with Removed and Uncertain Building Blocks

Circuit	Number of sizing rules		
	Equalities	Inequalities	in total
Figure 5.1	25	156	181
Figure 5.2	6	68	74
Figure 5.3	47	261	308
Figure 5.4	25	100	125
Figure 5.5	8	94	102
MI refbuffer	28	98	126

Table 5.3: Number of Sizing Rules for the Given Circuit Examples

stops as soon as a design point is found where all specifications are met, regardless of the circuit yield. Finally, a yield optimisation (design centring) was carried out to maximise the overall circuit yield.

At the beginning, a design point where some sizing rules were violated at least one specification was not met was chosen. The task was to find a design point where all specifications were met and the circuit was robust to variations.

To show the benefit of sizing rules, nominal optimisation and yield optimisation were performed once with and once without sizing rules. When no sizing rules were considered for nominal optimisation, no feasibility optimisation was required. This saved some simulation cost. Nevertheless, the results show that with consideration of sizing rules, a higher yield or better performance was achieved at the end of the optimisation process, mostly at a lower simulation effort. Each optimisation was performed on a computer with eight cores to parallelise the simulations. Depending on the number of required simulations, the overall time effort for optimisation ranged approximately between one and five hours.

5.3.1 Folded Cascode Amplifier (Figure 5.1)

In this circuit, the number of design parameters was 14. At the beginning, twelve sizing rules were violated. When sizing rules were considered, the feasibility optimisation

	sizing rules considered?	
	no	yes
#simulations for feasibility optimisation	0	272
#simulations for automatic sizing	failed	420
#violated sizing rules	9	0
#simulations for design centring after automatic sizing without sizing rules	n/a	n/a
overall yield	0%	0%
#simulations for design centring after automatic sizing with sizing rules	3428	3844
overall yield	99.46%	99.58%
#violated sizing rules	2	0

Table 5.4: Optimisation Results for the Circuit in Figure 5.1

needed 272 simulations and automatic sizing required another 420 simulations. When no sizing rules were considered, no design point where the specifications were met was found at all and still nine sizing rules were violated. Thus, the resulting yield was 0% and the consideration of sizing rules was crucial for this circuit.

From the point that was found with consideration of sizing rules, design centring was performed, again once with consideration of sizing rules and once without. The results show that from this point, design centring could be performed successfully in both cases and a high yield was achieved. Table 5.4 summarises the results for this circuit. When no sizing rules were considered, the design centring algorithm terminated after two iterations. When they were considered, the algorithm continued for another iteration, reaching a slightly higher yield at a higher simulation cost.

5.3.2 BiCMOS Operational Amplifier (Figure 5.2)

For the circuit in Figure 5.2, the number of design parameters was eight and two sizing rules were violated at the beginning. A design point where the specifications were met was found both when sizing rules were considered and when they were not considered. The number of required simulations for the nominal optimisation was about the same. When sizing rules were considered, 27 additional simulations were required to find a feasible point. However, when no sizing rules were considered, seven sizing rules were violated after automatic sizing.

Next, design centring was performed from the two design points that were found by automatic sizing. Once again, this was done once with consideration of sizing rules and once without. When no sizing rules were considered for design centring, the algorithm terminated prematurely with a linearisation error, so that no solution could be found. Hence, for the BiCMOS amplifier, the consideration of sizing rules was crucial for design centring.

	sizing rules considered?	
	no	yes
#simulations for feasibility optimisation	0	27
#simulations for automatic sizing	154	145
#violated sizing rules	7	0
#simulations for design centring after automatic sizing without sizing rules	failed	2475
overall yield	71.21%	99.98%
#violated sizing rules	7	0
#simulations for design centring after automatic sizing with sizing rules	failed	1365
overall yield	78.88%	$> 99.99\%$
#violated sizing rules	1	0

Table 5.5: Optimisation Results for the Circuit in Figure 5.2

When sizing rules were considered for design centring, a high yield was achieved for both design points that were found by automatic sizing. In the design centring run that started from the point that was found by automatic sizing with consideration of sizing rules, the yield was nearly 100%. In the run where sizing rules were not considered for automatic sizing, the yield was comparable and the simulation cost was about 80% higher. Table 5.3.2 summarises the results for this circuit.

Even though a high yield was achieved in both cases, there are also essential differences in the quality of the design points found after nominal optimisation. Since a design point was found both when sizing rules were considered and when they were not considered, the results of the nominal optimisation of both cases can be compared. As mentioned, nominal optimisation in WiCkeD stops as soon as a design point is found where all specifications are fulfilled. Thus, some of the optimal values were very close the lower or upper bound.

To compare the quality of the designs, a sweep over the operating parameters supply voltage (V_{dd}) between $1.5V$ and $4.0V$ and temperature (ϑ) between $-40°C$ and $80°C$ was performed. The nominal values were $2.1V$ and $27°C$. Exemplarily, the sweep result for the power supply rejection ratio $PSRR$ and the settling time t_s over these operating parameters is shown in Figures 5.6 and 5.7. The solid line shows the parameter sweep with sizing rules being considered, the dashed line shows the sweep with sizing rules not being considered.

The sweep over V_{dd} in Figure 5.6 shows a very strong contrast in the behaviour for the two cases. When sizing rules were not considered, the $PSRR$ diminished – first smoothly, later drastically – to a value of about 20dB at V_{min}. When sizing rules were considered, the $PSRR$ was only slightly reduced and stayed above 80dB throughout the specified operating range of V_{dd}. The two curves for the sweep over ϑ are more similar to each other. However, when sizing rules were not considered, the specification was

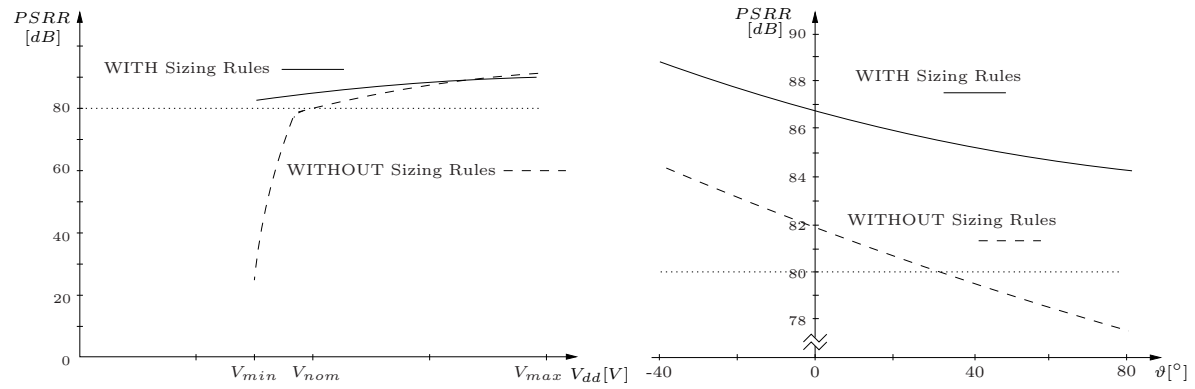

Figure 5.6: $PSRR$ against Supply Voltage and Temperature for the Circuit in Figure 5.2

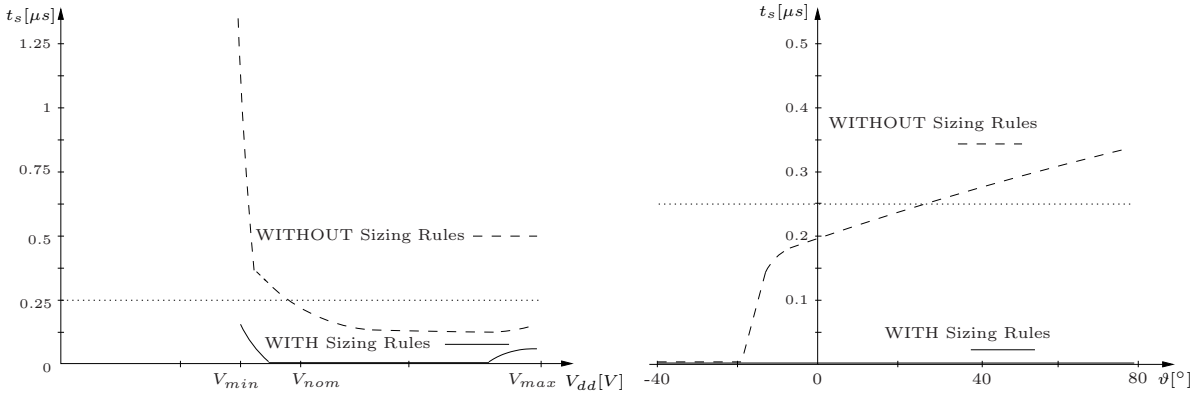

Figure 5.7: t_s Against Supply Voltage and Temperature for the circuit in Figure 5.2

violated when the temperature got higher than $35°C$. When sizing rules were considered, the $PSRR$ stayed well above the lower bound for the given range of ϑ.

Figure 5.7 shows the corresponding sweeps for the settling time. By definition, the settling time is zero when the output signal follows the input signal such that the value of the overshoot stays below a certain bound. The sweep over the supply voltage shows that the settling time stayed 0 when sizing rules were considered. Only for very low or very large values of V_{dd}, the settling time rose a little. But it stayed below the upper bound of $0.25\mu s$. When sizing rules were not considered, the settling time always stayed above zero, and when V_{min} was approached, the settling time leaped.

The sweep over temperature shows that the settling time was always zero for any temperature between $-40°C$ and $80°C$ when sizing rules were considered. When sizing rules were not considered, t_s was zero for temperatures below $-20°C$, but rose to more than $0.25\mu s$ for a temperature higher than $30°C$.

All sweeps show that the design was much less sensitive to variations of operating conditions when sizing rules were considered.

	sizing rules considered?	
	no	yes
#simulations for feasibility optimisation	0	167
#simulations for nominal optimisation	148	199
#violated sizing rules	30	0
#simulations for design centring after automatic sizing without sizing rules	3810	5158
overall yield	51.10%	99.34%
#violated sizing rules	30	0
#simulations for design centring after automatic sizing with sizing rules	4217	3885
overall yield	99.73%	99.94%
#violated sizing rules	6	0

Table 5.6: Optimisation Results for the Circuit in Figure 5.3

5.3.3 CMOS Buffer Amplifier (Figure 5.3)

The number of design parameters for the circuit in Figure 5.3 was 26. The number of sizing rules violated at the beginning was 21. This circuit showed very non-linear behaviour related to the supply voltage, even when all sizing rules were fulfilled. Hence, it was very hard to find a design point with high yield.

For nominal optimisation, a total of 366 simulations was needed when sizing rules were considered. Without consideration of sizing rules, a design point where the specifications were met was found after only 148 simulations. But as for the BiCMOS operational amplifier, there were essential differences regarding the quality of the design points. When sizing rules were neither considered for nominal optimisation nor design centring either, a yield of just above 51% was achieved. When sizing rules were not considered during nominal optimisation but for design centring, a yield of over 99% was achieved, but at a very high cost of over 5000 simulations. This is similar to the result obtained for the BiCMOS amplifier.

When sizing rules were considered for nominal optimisation, the result of design centring was much better. The yield optimisation required 3885 or 4217 simulations respectively, when sizing rules were considered for design centring or not. The resulting yield amounted to nearly 100% in both cases. Hence, similar to the folded cascode amplifier, after finding a design point where all specifications as well as all sizing rules are fulfilled, design centring lead to much higher yields, even when no sizing rules were considered for design centring. The violated sizing rules occur all in level shifters, and it is always rule (3.8) that is violated, i.e., the transistors are close to moving into the triode region. In a level shifter, a variation of the drain currents is not as critical as in a simple current mirror. In cascode current mirrors, which some of these level shifters are part of, the current ratio is determined by the two lower transistors, not by the ones that form the

level shifter. Hence, the effect is not too substantial either. The results are summarised in Table 5.6.

As shown in Table 5.2, there are six transistors in this circuit which were classified as uncertain building blocks, i.e., no operating region was assigned automatically. For completeness, nominal optimisation and design centring was performed once again with forcing these transistors into the saturation region. For feasibility optimisation, 140 simulations and for nominal optimisation, 255 simulations were needed. That is a total of 29 more simulations than for the case where these transistors were not forced into saturation region. The yield optimisation algorithm reported a successful result with a yield of 98.98% after just 2503 simulations. Hence, the process was further accelerated with about the same yield at the end.

To compare the quality of the design points found by nominal optimisation, parameter sweeps over V_{dd} and ϑ were carried out for this circuit as well. The nominal supply voltage was $4.2V$, the nominal temperature $27°C$. The bounds for the supply voltage were $2.0V$ and $5.0V$, the bounds for the temperature were the same as for the BiCMOS operational amplifier. Exemplarily, the results for DC gain and slew rate (SR) are shown in Tables 5.8 and 5.9. The sweep over V_{dd} shows very non-linear curve progressions. The specification of $50dB$ was only reached within a small range of V_{dd}. When sizing rules were considered, the DC gain was about 30 to $40dB$ higher than when sizing rules were not considered. Only for values of V_{dd} close to its upper bound, the DC gain was lower when sizing rules were considered. The curve progression over temperature looks similar when sizing rules were not considered. But with consideration of sizing rules, the DC gain stayed above the lower bound throughout the given temperature range, dropping nearly linearly for increasing temperature.

The results for the slew rate are similar. For low values of the supply voltage, the circuit could not follow the rectangular input signal. Hence, the slew rate could not be determined. The slew rate reached the specification of $50V/\mu s$ for $V_{dd} \approx 3.6V$ or $V_{dd} \approx 4.1V$ respectively. With consideration of sizing rules, higher values of the slew rate were achieved. The sweep over temperature shows smoother curve progressions again. When sizing rules were considered, the specification was fulfilled over the whole temperature range. When they were not considered, the specification was violated when the temperature exceeded $35°C$.

Once again, the solution that was found after nominal optimisation with consideration of sizing rules was much more robust than the one without consideration of sizing rules.

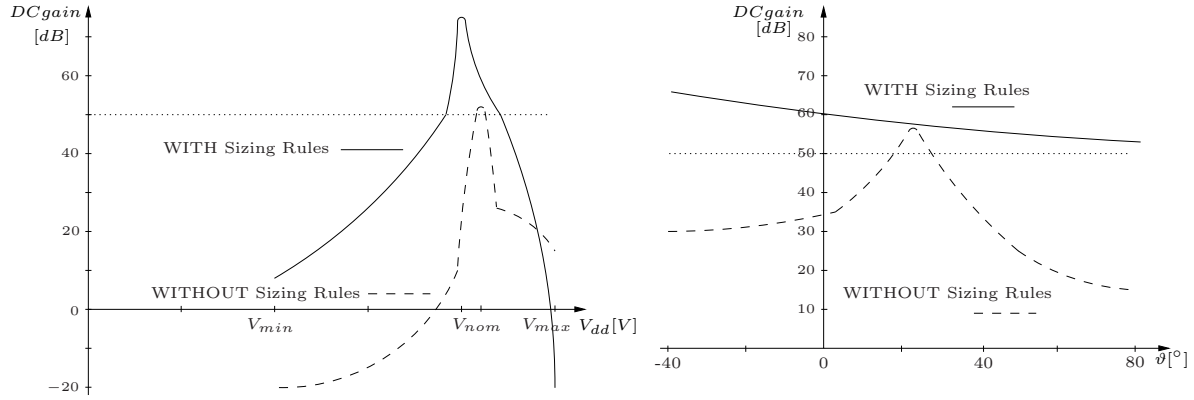

Figure 5.8: DC Gain against Supply Voltage and Temperature for the Circuit in Figure 5.3

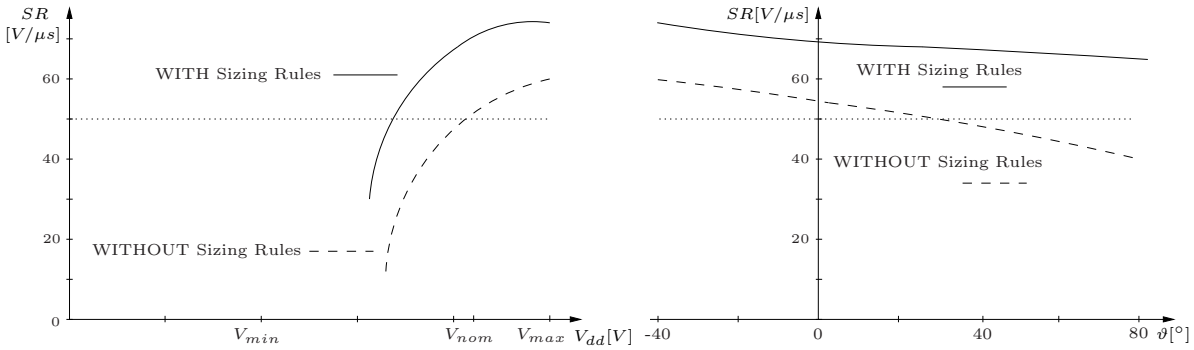

Figure 5.9: Slew Rate against Supply Voltage and Temperature for the Circuit in Figure 5.3

5.4 Further Applications

Several aspects of this thesis have been incorporated into other work. This comprises

a) the usage of the preparation of the netlist for discrete optimisation,

b) the extension of the structure recognition method by symmetry recognition for analog layout,

c) and structure recognition in digital circuits.

The preparation of the netlist as described in Chapter 4.3 transforms every circuit element into a meta element. These meta elements contain information about the circuit element's type and model (e.g., transistor model) and the number of devices connected in parallel. Hence, type and number of devices connected in parallel can be used as design parameters. They are design parameters of the meta elements and can be passed to an optimiser like all other design parameters. However, there are fundamental differences: These parameters are discrete and cannot be altered continuously. Classic gradient-based methods can only work with continuous parameters and round the result after the optimisation. After rounding, fulfilment of the specifications and sizing rules is no longer

guaranteed. While the parameter "number of devices in parallel" can be ordered, this is not the case for the element type without any problems. Replacing one circuit element with another one involves more than just altering one design parameter like width or length. In the case of structural optimisation, where compounds of circuit elements, e.g., current mirrors, are replaced by other ones, even more factors are involved. For instance, replacing a simple current mirror with a current mirror that consists of four transistors, the voltage drop at the current mirror increases which leads to completely different circumstances. In [PMGS08a] and [PMGS08b], a method for discrete optimisation of analog circuits is presented. For convenience, the parameter 'type' was not incorporated. Instead, RFCMOS transistors that only differ from one another by their length were mentioned. But in the actual experiments, different RFCMOS transistors were used instead of just different lengths. In practice, RFCMOS transistors of different length also show a considerable difference in their physical behaviour, which is why it is very hard and therefore expensive to generate continuous models for these transistors.

In circuit layout, symmetry constraints are of high importance. In this thesis, such constraints are not taken into account. By analysing the symmetry of a circuit, more equalities come into play, further reducing the number of design parameters. In addition, symmetry analysis provides useful additional information for automatic layout tools. For instance, in the circuit in Figure 5.2, the two pnp simple current mirrors are connected to the same differential pair. For symmetry reasons, both the driving and the driven transistors of these simple current mirrors are sized equally. Symmetry constraints and symmetry recognition have first been covered in [Eic08].

The presented module library contains basic modules for analog design. Within the scope of a new research topic at the Institute for Electronic Design Automation, a library of basic logic blocks has been introduced in [SS09]. In that work, a new recognition algorithm and a new dominance relation have been set up to handle digital circuits. A new challenge that comes with the analysis of digital circuits is the high number of transistors. First results show that even for circuits with 100000 transistors, the duration of structure recognition stays below two minutes, and the computational effort depends quadratically on the number of transistors (see also Chapter 4.4)

The new structure recognition algorithm presented in this dissertation has been re-implemented by Cadence® in their Design Framework II [Cad09].

Another application of sizing rules is Response Surface Modelling (RSM) which serves to replace performance evaluation by computationally expensive circuit simulation models with cheaper performance evaluation by analytical functions. In practical applications, RSM has to select basis functions for the analytical model, select test points where the "true" circuit is evaluated, and compute the coefficients of the analytical model.

Another problem is the definition region of the analytical model. Sizing rules provide an accurate and technically relevant definition region of an analytical model. The region where test points have to be simulated is much smaller than the original region defined by simple box constraints. In addition, the performance behaviour is near to linear in

the region where sizing rules are satisfied. This results in an increased accuracy of the analytical models [ZEG98].

5.5 Summary

The number of sizing rules determined in the six given circuits ranges between 67 and 308. This shows that the manual determination of sizing rules is time-consuming and error-prone. Without the arbitration of assignment ambiguities, up to 34 wrong building blocks would have been recognised and a high number of wrong sizing rules would have been assigned. As a consequence, the optimisation of the given circuits would not have lead to a satisfactory result. Hence, the arbitration of assignment ambiguities is crucial.

The results for the three circuits in Figures 5.1–5.3 show that already after nominal optimisation, sizing rules lead to more robust designs. Especially in the second step, design centring, the consideration of sizing rules was crucial. Consideration of sizing rules led to a much higher yield for the two CMOS circuits. The results also show that when nominal optimisation of the CMOS circuits was performed with sizing rules, design centring always led to high yields, even when sizing rules were not considered during design centring. But in the case of the circuit in Figure 5.3, this claimed a very high simulation cost. For the BiCMOS operational amplifier, the simulation effort for the yield optimisation was about 80% higher when it started from the point that was determined by nominal optimisation without sizing rules.

Aspects of the introduced structure recognition method were incorporated into other work which also confirms the relevance of this thesis.

Chapter 6

Conclusion

Analog circuit synthesis is a complicated process which for the most part is still done manually. The main reasons for this are:

- A circuit topology cannot be derived directly from the given specifications.
- Topology selection alone is not sufficient. Instead, an analog circuit has to be sized properly to ensure its correct function.
- Variations of operating conditions and process parameters have to be taken into account to ensure robustness.

There are approaches to automated circuit synthesis, usually using genetic algorithms. Circuit sizing is automated to a certain extent. Modern optimisers can adjust design parameters such as transistor geometries in an intelligent way to quickly find a design point where all specifications are fulfilled. However, optimisation requires numerous circuit simulations. Each simulation requires computational effort and is time-consuming. Circuit sizing and design centring comprise hundreds or even thousands of simulations.

To reduce the number of overall simulations, additional constraints can be specified to confine the design parameter space such that technically meaningless regions are excluded. These constraints were first mentioned under the term "sizing rules" in [EGG98]. In [GZEA01], a list of generic sizing rules for basic CMOS transistor compounds as well as an algorithm for their automatic generation was presented. A drawback of the automatic structure recognition algorithm presented in [GZEA01] is the numerous ambiguities that requires extensive manual post-processing.

In this dissertation, a library of both CMOS and bipolar modules, i.e., single transistors or compounds consisting of at least two transistors has been set up. Seventeen different modules consisting of more than one transistor have been defined. Some of these are defined for CMOS transistors only, others for bipolar transistors only, while others are defined for both. A differential stage can even consist of both CMOS and bipolar transistor sub-modules. The library is organised hierarchically such that all modules except the single transistor and banks consist of two sub-modules. For all of these

modules, a set of sizing rules has been given. Sizing rules given as equalities reduce the dimension of the available parameter space, whereas sizing rules given as inequalities bound the parameter space. As the results show, both leads to an acceleration of the design process and the designs are more robust when sizing rules are fulfilled. The threshold values given in the inequality sizing rules have to be given just once for a technology.

The large number of sizing rules of a circuit clearly shows that these cannot be established manually for each circuit. Therefore, a new two-stage structure recognition algorithm has been developed. This new algorithm is capable of recognising all defined modules and resolving ambiguities that might occur during the recognition process. For this purpose, a dominance relation has been set up. If two or more modules contain the same sub-module and one of them dominates the other ones, these modules will be removed. This produces very reliable results and reduces manual post processing to a minimum. A list of the removed modules as well as a list of uncertain modules that were not recognised as part of a larger module is passed to the designer afterwards. The complexity of the structure recognition algorithm is only quadratic in practice due to additional measures, a small library of modules and the fact that analog circuits are usually small. But even in digital circuits with more than one hundred thousand transistors, structure recognition only takes a few minutes, even though a few additional measures had to be taken.

The preparation of the netlist that is performed before the recognition process is also the foundation for structural optimisation of analog circuits. Furthermore, the new structure recognition method with the arbitration of assignment ambiguities is already being used for structure recognition in digital circuits. In the future, these new applications will be further extended.

An additional future topic could be a classification of sizing rules into degrees of necessity, similar to the distinction between sizing rules for function and robustness. The design centring results in Chapter 5 show, that from a point that was reached by nominal optimisation with consideration of sizing rules, design centring without consideration of sizing rules can still lead to a high yield. Hence, a small or moderate violation of some of the sizing rules could be permitted during the optimisation process, at least in few building blocks.

Bibliography

[AEG+00] K. Antreich, J. Eckmueller, H. Graeb, M. Pronath, F. Schenkel, R. Schwencker, S. Zizala: *WiCkeD: Analog circuit synthesis incorporating mismatch*, IEEE Custom Integrated Circuits Conference (CICC), pages 511–514, May 2000.

[AH87] P. E. Allen, D. R. Holberg: *CMOS Analog Circuit Design*, Oxford University Press, 1987.

[Bin08] D. M. Binkley: *Tradeoffs and Optimization in Analog CMOS Design*, John Wiley and Sons Ltd., 1st edition, June 2008.

[Cad09] Cadence Design Framework II, *www.cadence.com*, 2009.

[CS91] D. J. Chen, B. J. Sheu: *Automatic custom layout of analog ICs using constraint-based module generation*, IEEE Custom Integrated Circuits Conference (CICC), pages 5.5/1–5.5/4, May 1991.

[DG98] G. Debyser, G. Gielen: *Efficient analog circuit synthesis with simultaneous yield and robustness optimization*, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 1998.

[DIY95] M. J. Deen, J. Ilowski, P. Yang: *Low frequency noise in polysilicon-emitter bipolar junction transistors*, Journal of Applied Physics, volume 77(12), pages 6278–6288, June 1995.

[dMHBL01] M. del Mar Hershenson, S. P. Boyd, T. H. Lee: *Optimal design of a CMOS op-amp via geometric programming*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume 20(1), pages 1–21, January 2001.

[Eck98] J. Eckmüller: *Zur rechnergestützten Dimensionierung analoger integrierter Schaltungen unter besonderer Berücksichtigung von Struktureigenschaften*, Ph.D. thesis, January 1998.

[EGG98] J. Eckmueller, M. Groepl, H. Graeb: *Hierarchical characterization of analog integrated CMOS circuits*, Design, Automation and Test in Europe (DATE), pages 636–643, Paris, France, February 1998.

[Eic08] M. Eick: *Layoutsynthese analoger integrierter Schaltungen mit automatisch generierten Platzierungsregeln*, Master's thesis, Technische Universität München, March 2008.

[FK87] J. Fisher, R. Koch: *A highly linear CMOS buffer amplifier*, IEEE Journal of Solid-State Circuits SC, volume 22, pages 330–334, 1987.

[Gel08] P. P. Gelsinger: *From peta FLOPS to milli watts*, http://download.intel.com/pressroom/kits/events/idfspr_2008/2008_0402_IDF-PRC_Gelsinger_EN.pdf, April 2008.

[Gro99] O. M. Group: *OMG Unified Modeling Language Specification*, Object Management Group, Inc., 1999.

[GT96] W. K. Grassmann, J.-P. Tremblay: *Logic and Discrete Mathematics*, Prentice-Hall, Inc., 1996.

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

[HO95] K.-T. Huang, D. A. Overhauser: *A novel graph algorithm for circuit recognition*, in: IEEE International Symposium on Circuits and Systems (ISCAS), volume 3, pages 1695–1698, April 1995.

[Hor05] J. Horgan: *Analog*, http://www10.edacafe.com/nbc/articles/view_weekly.php?run_date=21-Mar-2005, March 2005.

[HRC89] R. Harjani, R. Rutenbar, L. Carley: *OASYS: A framework for analog circuit synthesis*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume 8, pages 1247–1266, 1989.

[Jae04] R. C. Jaeger: *Microelectronic Circuit Design*, McGraw-Hill Professional, 2004.

[LHC86] K. Lakshmikumar, R. Hadaway, M. Copeland: *Characterization and modeling of mismatch in MOS transistors for precision analog design*, IEEE Journal of Solid-State Circuits SC, volume 21, pages 1057–1066, 1986.

[LS94] K. R. Laker, W. Sansen: *Design of Analog Integrated Circuits and Systems*, McGraw-Hill, New York, 1994.

[Mat93] H. F. Mattson: *Discrete Mathematics with Applications*, John Wiley & Sons, Inc., 1993.

[MCR00] T. Mukherjee, L. R. Carley, R. A. Rutenbar: *Efficient handling of operating range and manufacturing line variations in analog cell synthesis*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume 19(8), pages 825–839, August 2000.

[MG08] T. Massier, H. Gräb: *Dimensionierungsregeln für analoge Bipolarschaltungen*, pages 107–112, 10. GMM/ITG Diskussionssitzung Entwurf von Analogschaltungen (ANALOG '08), April 2008.

[MGS08a] T. Massier, H. Graeb, U. Schlichtmann: *Sizing rules for bipolar analog circuit design*, Design, Automation and Test in Europe (DATE), March 2008.

[MGS08b] T. Massier, H. Graeb, U. Schlichtmann: *The sizing rules method for CMOS and bipolar analog integrated circuit synthesis*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume 27(12), pages 2209–2222, December 2008.

[MK95] H. A. W. Markus, T. G. M. Kleinpenning: *Low-frequency noise in polysilicon emitter bipolar transistors*, IEEE Transactions on Electron Devices ED, volume 42(4), pages 720–727, April 1995.

[Moo65] G. E. Moore: *Cramming more components onto integrated circuits*, Electronics, volume 38(8), April 1965.

[MSG03] T. Massier, G. Stehr, H. Gräb: *Ein Beitrag zur Automatisierung der Strukturanalyse und der impliziten Spezifikation von analogen integrierten Schaltungen*, 7. GMM/ITG Diskussionssitzung Entwurf von Analogschaltungen (ANALOG '03), pages 31–36, June 2003.

[Mun09] WiCkeD, *www.muneda.com*, 2009.

[MV01] P. Mandal, V. Visvanathan: *CMOS Op-Amp sizing using a geometric programming formulation*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume 20(1), pages 22–38, January 2001.

[NoC09] Official web site of the Nobel Foundation, *http://nobelprize.org*, 2009.

[PDW89] M. Pelgrom, A. Duinmaijer, A. Welbers: *Matching properties of MOS transistors*, IEEE Journal of Solid-State Circuits SC, volume 24, pages 1433–1440, 1989.

[PKR+00] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley, J. R. Hellums: *Anaconda: Simulation-based synthesis of analog circuits via stochastic pattern search*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume 19(6), pages 703–717, June 2000.

[PMGS08a] M. Pehl, T. Massier, H. Gräb, U. Schlichtmann: *Optimierung analoger Schaltungen mit geordneten diskret veränderlichen Parametern*, M.- u. F. G. VDE/VDI-Gesellschaft Mikroelektronik (ed.), GME/ITG-Diskussionssitzung Entwicklung von Analogschaltungen mit CAE-Methoden, 10, pages 171–176, VDE, VDE Verlag GmbH, April 2008.

[PMGS08b] M. Pehl, T. Massier, H. Graeb, U. Schlichtmann: *A random and pseudo-gradient approach for analog circuit sizing with non-uniformly discretized parameters*, IEEE International Conference on Computer Design (ICCD), pages 188–193, October 2008.

[Pow08] J. R. Powell: *The quantum limit to moore's law*, Proceedings of the IEEE, volume 96(8), pages 1247–1248, August 2008.

[Rub03] N. Rubanov: *SubIslands: The probabilistic match assignment algorithm for subcircuit recognition*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2003.

[Rub06] N. Rubanov: *A high-performance subcircuit recognition method based on the nonlinear graph optimization*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume 25(11), pages 2353–2363, November 2006.

[Rut10] R. Rutenbar: *Analog layout synthesis: What's missing?*, International Symposium on Physical Design (ISPD), page 43, March 2010.

[Sem07] ITRS, *International Technology Roadmap for Semiconductors: 2007 Edition – Design*, http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_Design.pdf, 2007.

[SGS+04] U. Schlichtmann, H. Gräb, R. Sommer, E. Hennig, F. Schenkel, T. Ifström: *Systematic analog/mixed-signal design – yield optimization of analog circuits with WiCkeD*, MEDEA+ Forum 2004, November 2004.

[SH68] H. Shichman, D. A. Hodges: *Modeling and simulation of insulated-gate field-effect transistor switching circuits*, IEEE Journal of Solid-State Circuits SC, volume 3(3), pages 285–289, September 1968.

[SS09] L. Stolberg-Stolberg: *Automatische Gatterextraktion für digitale CMOS Schaltungen*, Studienarbeit, Technische Universität München, February 2009.

[ST02] T. Sripramong, C. Toumazou: *The invention of cmos amplifiers using genetic programming and current-flow analysis*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002.

[VDL+01] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. Gielen, W. Sansen, P. Veselinovic, D. Leenaerts: *AMGIE–A synthesis environment for CMOS analog integrated circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, volume 20(9), pages 1037–1058, September 2001.

[WW99] J. G. Webster, R. E. Webster: *Wiley Encyclopedia of Electrical and Electronics Engineering*, Wiley & Sons, 1st edition, March 1999.

[YS03] L. Yang, C.-J. R. Shi: *FROSTY: A fast hierarchy extractor for industrial CMOS circuits*, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pages 741–746, November 2003.

[ZEG98] S. Zizala, J. Eckmueller, H. Graeb: *Fast calculation of analog circuits' feasibility regions by low level functional measures*, IEEE Int. Conf. on Electronics, Circuits and Systems, pages 85–88, September 1998.

[ZEG99] S. Zizala, J. Eckmüller, H. Gräb: *Effiziente Modellierung integrierter analoger CMOS-Schaltungen durch Berücksichtigung von Struktureigenschaften*, GME/ITG-Diskussionssitzung Entwicklung von Analogschaltungen mit CAE-Methoden, pages 46–47, 1999.

[Ziz01] S. Zizala: *Numerische Verhaltensmodellierung analoger CMOS-Schaltungen unter Berücksichtigung von Dimensionierungsregeln*, Ph.D. thesis, 2001.

List of Figures

List of Tables