THÈSE DE DOCTORAT DE L'UNIVERSITÉ PARIS VI

Spécialité : **Informatique et Micro-Électronique**

Présentée par : **Mohamed DESSOUKY**

Pour obtenir le titre de
Docteur de l'Université Paris VI

Conception en vue de la Réutilisation de Circuits Analogiques. Application: Modulateur Delta-Sigma à très Faible Tension

Soutenue le : **18 Janvier 2001**

Devant le jury composé de :

M. Georges GIELEN	Rapporteur
M. Patrice SENN	Rapporteur
M. Andreas KAISER	Examinateur
M. Hani RAGAI	Examinateur
M. Yves LEDUC	Examinateur
M. Alain GREINER	Examinateur
Mme. Marie-Minerve LOUERAT	Examinateur

Ph.D. THESIS OF THE UNIVERSITY OF PARIS VI

Department : COMPUTER SCIENCE AND MICRO-ELECTRONICS

Presented by : **Mohamed DESSOUKY**

Thesis submitted to obtain the degree of
DOCTOR OF THE UNIVERSITY OF PARIS VI

DESIGN FOR REUSE OF ANALOG CIRCUITS. CASE STUDY: VERY LOW-VOLTAGE DELTA-SIGMA MODULATOR

18 January 2001

Committee in charge :

M. Georges GIELEN
M. Patrice SENN
M. Andreas KAISER
M. Hani RAGAI
M. Yves LEDUC
M. Alain GREINER
Mme. Marie-Minerve LOUERAT

À la mémoire de ma mère.

Remerciements

Je tiens tout d'abord à remercier Madame Marie-Minerve Louërat, Chargée de Recherche CNRS au laboratoire LIP6, d'avoir dirigé mes recherches et d'avoir organisé avec efficacité les contacts avec les autres équipes hors du laboratoire. Je lui suis très reconnaissant pour ses qualités rares tant au niveau humain que scientifique sans lesquels le développement et l'achèvement de ces recherches - et de ce manuscrit - n'auraient été possibles.

Je tiens particulièrement à remercier le professeur Alain Greiner à la fois pour son travail d'encadrement, ses conseils, sa patiente relecture du manuscrit et ses qualités pédagogiques que j'apprécie beaucoup.

Je tiens à remercier vivement Monsieur Andreas Kaiser, Chargé de Recherche CNRS et professeur à l'ISEN-Lille, pour m'avoir accueilli au sein de son équipe pendant la phase de conception et de mesure du circuit. Je lui dois beaucoup, entre autres, pour les nombreuses et instructives discussions.

Je remercie également Monsieur Jacky Porte, Ingénieur d'étude à l'ENST-Paris, pour m'avoir permis d'utiliser son logiciel COMDIAC et pour les enrichissantes discussions que nous avons eu.

Que les membres du jury trouvent ici l'expression de ma gratitude. Tout d'abord, le professeur Georges Gielen, Responsable de l'équipe de Méthodologies de Conception de circuits analogiques et mixtes dans le groupe MICAS à l' Université Catholique de Leuven, Belgique, et Monsieur Patrice Senn, Responsable du laboratoire France Telecom R&D/DIH/OCF, qui m'ont fait l'honneur d'être rapporteurs de cette thèse. Le professeur Hani Ragai Responsable du laboratoire ICL au département d'Electroniques et Télécommunications, Université d'Ain Shams, Egypte, et Monsieur Yves Leduc, de Texas Instruments, pour l'intérêt qu'ils ont porté à mes travaux de recherches.

Je remercie tous les membres de l'équipe ASIM du laboratoire LIP6, et plus particulièrement Abdelhakim Khouas pour son soutien et son amitié pendant ces années, Hassan Aboushady pour les innombrables conversations et "deadlines" que nous avons vécu ensemble, Frédéric Pétrot, Franck Wajsbürt, et Ludovic Jacomme pour m'avoir aidé à comprendre et à utiliser avec efficacité les outils ALLIANCE. Je remercie également tous les étudiants de DEA qui ont contribué à l'évolution de l'outil CAIRO : Ronan Lucas, Frédéric Billereau, Alphonse Chesneau et plus particulièrement Matthieu Lintz et Pierre Nguyen Tuong à qui je souhaite une bonne continuation dans leurs recherches.

Je ne pourrai jamais trouver les mots justes pour exprimer mes sentiments pour ma meilleure amie et sœur Naglaa. Je veux la remercier pour son soutien, ses sacrifices et surtout pour m'avoir toujours cru à chaque fois que je lui disais que je finirai la thèse "l'année prochaine".

Enfin, je ne connais personne qui aurait été plus heureuse d'assister à l'achèvement de cette thèse, et à qui je pensais constamment pendant ces dernières années, que ma mère. Si j'ai eu de la patience, de la volonté et de la persévérance qui m'ont permis de mener ce travail, ce n'étaient que quelque gouttes que j'ai pu récupérer de la rivière abondante de sa vie. Si elle a dû quitter ce monde au milieu de la thèse, son esprit restera présent en moi à tout jamais.

M. Dessouky

Résumé

Le fait de pouvoir réutiliser des circuits analogiques pour différents procédés cibles prend de plus en plus d'importance dans la conception des systèmes intégrés actuels. Pour atteindre les meilleures performances en changeant de technologie cible, un des problèmes clés est la prise en compte des dégradations possibles dues aux masques physiques dans la phase de synthèse électrique.

Pour résoudre ce problème, nous avons proposé une nouvelle méthode de conception, basée sur la synthèse analogique prenant en compte le dessin des masques. Cette méthode permet de conserver le savoir-faire pour une réutilisation future tout en assurant une forte intégration entre la phase du dimensionnement électrique et la réalisation physique. Elle garantit le respect des performances attendues, permet d'optimiser certains aspects de la conception en présence de parasites et réduit le temps total de conception en évitant les itérations fastidieuses entre le dimensionnement et les masques. Cette méthode a été mise en oeuvre grâce à deux outils basés sur le savoir-faire, l'un dédié au dimensionnement des circuits analogiques (COMDIAC) et l'autre au dessin des masques (CAIRO). Ces outils permettent de réutiliser de manière efficace la méthode de conception ainsi que le dessin des masques dans le cas de circuits similaires.

Pour valider cette approche, nous l'avons appliquée à des circuits basse-tension, basse-consommation. Notre étude a conduit à de nouvelles architectures de circuits qui permettent le fonctionnement sous très basse tension d'un circuit à capacités commutées en technologie CMOS standard. En suivant cette approche, nous avons conçu, fabriqué et testé un modulateur analogique-numérique Delta- Sigma pour des applications numériques audio, fonctionnant sous 1V avec 14 bits de précision. Deux circuits similaires ont été resynthétisés dans une autre technologie cible, montrant ainsi que notre méthode est tout à fait appropriée pour des circuits mixtes analogiques-numériques à hautes performances.

Mots Clés

Circuits analogiques réutilisables, Automatisation de la conception analogique, Génération de masques procédurale, Modulation Delta- Sigma, Basse-tension, Capacités commutées.

Abstract

Analog design reuse is becoming more and more important in recent system-on-chip designs. In these designs electrical and physical design integration is a challenging problem specially when designing high performance analog circuits in different technologies.

To solve this problem, we propose a new design methodology based on a layout-oriented synthesis approach that allows to capture design knowledge for eventual reuse with a close interaction between electrical and physical design. This methodology guarantees the fulfillment of the required performance specifications, permits to optimize various design aspects in the presence of parasitics and shortens the overall design time by avoiding laborious sizing-layout iterations. The methodology has been implemented using two knowledge-based tools dedicated to analog circuit sizing (COMDIAC) and layout generation (CAIRO). The tools allow both the design knowledge and the generated layout to be efficiently reused in similar designs.

To validate the previous claims, we have chosen low-voltage low-power analog circuits as an application. Our study has led to new circuit architectures that allow very low-voltage switched-capacitor circuit operation in standard CMOS technologies. Using the above methodology and circuit techniques, we have designed, fabricated, and tested a 1-V 14-bit Delta-Sigma A/D modulator for digital-audio applications. Two similar designs are then resynthesized in another technology demonstrating the suitability of the methodology for very high performance mixed-signal circuits.

Keywords

Analog Design Reuse, Analog Design Automation, Procedural Layout Generation, Delta-Sigma Modulator, Low-Voltage, Switched-Capacitor.

Contents

List of Abbreviations and Symbols

Abbreviations

AC	Alternating Current.
ADC	Analog-to-Digital Converter.
BS	Bootstrapped Switch.
BW	Bandwidth.
CAD	Computer-Aided Design.
CM	Common-mode.
CMFB	Common-mode Feedback.
CMOS	Complementary Metal Oxide Semiconductor.
DAC	Digital-to-Analog Converter.
DA	Design Automation.
DC	Direct Current.
DR	Dynamic Range.
EDA	Electronic Design Automation.
FM	Figure of Merit.
GBW	Gain-Bandwidth product.
HDL	Hardware Design Language.
IP	Intellectual Property.
IC	Integrated circuit.
MOS	Metal Oxide Semiconductor.
NTF	Noise Transfer Function.
Opamp	Operational Amplifier.
OSR	Oversampling Ratio.
OTA	Operational Transconducatne Amplifier.
SC	Switched-capacitor.
SO	Switched-opamp.
SoC	System-on-Chip.
SNR	Signal-to-Noise Ratio.
SNDR	Signal-to-Noise plus Harmonic Distortion Ratio.

SQNR	Signal-to-Quantization Noise Ratio.
SR	Slew Rate.
PM	Phase Margin.
PSRR	Power Supply Rejection Ratio.
VHDL	Very High Speed Integrated Circuit Hardware Description Language.
VLSI	Very Large Scale Integration.
VM	Voltage Multiplication.

Symbols

A_{d0}	Opamp DC gain.
β_i	Integrator feedback factor (integration phase).
β_s	Integrator feedback factor (sampling phase).
C_C	Opamp compensation capacitance.
C_{ip}	Opamp input capacitance.
C_I	Integration capacitance.
C_{Ibp}	Integration capacitor parasitic bottom-plate capacitance.
C_{ds}	Transistor Drain-source capacitance.
C_{db}	Transistor Drain-bulk junction capacitance.
C_{sb}	Transistor Source-bulk junction capacitance.
C_{gb}	Transistor Gate-bulk capacitance.
C_{gd}	Transistor Gate-drain capacitance.
C_{gs}	Transistor Gate-source capacitance.
C_L	Opamp load capacitance.
C_{ox}	Transistor oxide capacitance.
C_S	Sampling capacitance.
$\mathfrak{F}$	Transistor diffusion capacitance reduction factor due to folding.
f_{CL}	Opamp closed-loop 3-dB frequency.
f_l	Lowest signal frequency (=200Hz).
f_m	Maximum signal frequency (signal bandwidth).
f_s	Sampling frequency.
f_t	Opamp gain-bandwidth product frequency.
f_u	Opamp unity-gain frequency.
ϕ_j	pn junction built-in potential.
g_0	Integrator gain.
g_{ds}	Drain-source conductance.
g_m	Input-output transconductance.
g_{mb}	Bulk transconductance.

g_o	Output conductance.
γ_{th}	White noise excess factor.
γ_f	Flicker noise excess factor.
θ_i	Settling error in the integration phase.
I_D	Transistor drain-source current.
I_{Dsat}	Transistor drain-source current.
k	Boltzman constant.
λ	Symbolic layout unit.
M	Number of parallel transistor elements of a given transistor.
N_T	Total noise power.
N_Q	Quantization noise power.
N_{sw}	Switches noise or KT/C noise power.
N_{amp}	Total opamp noise power.
N_{th}	Opamp thermal noise power.
$N_{1/f}$	Opamp flicker noise power.
NTF_{inf}	Out-of-band gain of the noise transfer function.
ρ_i	Closed-loop static error (integration phase).
ρ_s	Closed-loop static error (sampling phase).
T	Absolute Temperature.
T_s	Sampling period.
t_s	Time allowed for sampling.
t_i	Time allowed for integration.
t_{av}	Available time for charging/discharging.
t_{lin}	Time of linear response.
t_{nov}	Non-overlap clock phases time.
t_{slew}	Time of slewing response.
τ	Integrator time constant.
τ_{cmfb}	Opamp CMFB time constant.
ω_t	Gain-bandwith product.
ω_{tcmfb}	Gain-bandwith product of the CMFB amplifier.
ω_u	Unity gain frequency.
U_{max}	Maximium allowable input signal level for which the modulator remains stable.
μ	Transistor channel mobility.
V_{AGND}	Analog ground common-mode voltage.
V_{bias}	Transistor bias voltage.
Vb_{cmfb}	Transistor CMFB circuit bias voltage.
V_{BS}	Transistor bulk-source voltage.
V_{DD}	Supply voltage.

V_D	Transistor drain voltage.
V_{DS}	Transistor drain-source voltage.
V_{dsat}	Transistor saturation voltage.
V_E	Transistor Early voltage.
V_{EG}	Transistor effective gate-source voltage ($V_{GS} - V_{th}$).
V_{GS}	Transistor gate-source voltage.
V_{th}	Transistor threshold voltage.
Vip_{CM}	Opamp input common-mode voltage.
Vop_{CM}	Opamp output common-mode voltage.
Vop_{max}	Opamp maximum output voltage.
Vop_{min}	Opamp minimum output voltage.
Vop_{swing}	Opamp linear output voltage swing.
V_{ref}	Modulator reference voltage.
V_S	Transistor source voltage.
V_{SS}	Supply ground voltage.

List of Figures

List of Tables

Résumé Étendu en Français

Ce chapitre est un résumé étendu de la thèse. Chaque paragraphe de ce résumé correspond à un chapitre de la thèse.

1 Introduction

Comme les progrès technologiques permettent de créer des systèmes intégrés mixtes analogiques-numériques, la complexité des circuits intégrés actuels continue à croître. Du fait que les systèmes intégrés deviennent plus complexes, la seule façon de concevoir des systèmes aussi denses est d'incorporer sur ces circuits des blocs existants, appelés aussi "Intellectual Property" (Propriété Intellectuelle), ou IP. Alors que ce concept obtient un certain succès dans la partie numérique des circuit mixtes, il est encore extrêmement difficile de réutiliser tel quel un bloc IP analogique.

Notre contribution porte sur trois points : Premièrement, nous proposons une méthode de conception pour la réutilisation de circuits analogiques, basée sur la co-conception électrique et physique du circuit. Deuxièmement, nous présentons des outils pour mettre en oeuvre cette méthode ; un outil de dessin des masques procédural, indépendant de la technologie, qui prend en compte les contraintes spécifiques des masques analogiques et un environnement dédié au dimensionnement de circuits, basée sur la réutilisation de savoir-faire. Enfin, l'application de cette méthode à l'aide de ces deux outils a été effectuée sur les circuits à capacités commutées fonctionnant sous très faible tension avec une faible consommation. Nous avons proposé de nouvelles architectures de circuits qui ont permis de concevoir et de réaliser un modulateur Delta-Sigma pour des application numériques audio, fonctionnant sous 1V et consommant 1mW. En dehors du fait que ce circuit est lui même intéressant, il démontre que la méthode que nous proposons, associée aux outils de CAO, sont particulièrement bien adaptés aux circuits mixtes à hautes performances.

2 Problématique

Le chapitre 2 présente le contexte de la thèse en définissant le problème à résoudre et en introduisant les objectifs du travail.

2.1 Les spécificités de la conception analogique

Les caractéristiques de la conception analogique, qui la différencient fortement de la conception numérique sont les suivantes :

- la hiérarchie est mal définie
- les performances sont définies par de nombreux paramètres
- le dimensionnement des composants élémentaires est critique pour la performance
- les composants d'un même circuit peuvent varier de plusieurs ordres de grandeur
- pour une même fonctionnalité, il existe une grande variété de topologies possibles
- la façon de dessiner les masques a une grande influence sur les performances
- les performances du circuit sont plus sensibles aux variations technologiques
- les différents niveaux de la hiérarchie du système intégré interagissent fortement
- les performances visées sont souvent aux limites de la technologie.

2.2 Automatisation et réutilisation de la conception

Le but de la conception assistée par ordinateur est d'automatiser certaines tâches comme le dimensionnement, l'optimisation et la génération des masques [Gielen91]. Le but ultime est de rendre automatique tout le processus de conception depuis la description comportementale jusqu'au dessin des masques. Le degré d'automatisation est mesuré par le rapport entre le temps de conception d'un système réalisé de façon manuelle et celui utilisant les outils de synthèse [Ochotta98].

Par ailleurs le but de la réutilisation est de pouvoir réutiliser efficacement une réalisation existante pour un système dans un autre environnement et/ou un autre procédé de fabrication, tout en conservant à peu près les mêmes performances. Le concept de réutilisation ne se limite pas à reprendre exactement le même schéma électrique, car l'expertise accumulée peut être employée pour concevoir des circuits similaires à l'aide d'une approche bien définie et grâce à quelques cellules du premier circuit. Le degré de réutilisation est mesuré par la quantité d'informations et d'expériences qui sont transmises d'un circuit réussi aux réalisations suivantes.

Cependant, beaucoup de concepts et d'outils sont communs entre l'automatisation et la réutilisation de la conception.

2.3 Modulateur Delta-Sigma très faible tension

Nous avons choisi les modulateurs Delta-Sigma fonctionnant sous très faible tension avec une faible consommation comme exemple de conception d'un bloc IP analogique, d'une part pour l'intérêt que présente leur conception et d'autre part car ils sont très adaptés pour tester l'approche que nous proposons ainsi que les outils de conception.

Les systèmes mono-puce à venir vont nécessiter l'intégration de parties purement numériques, de parties analogiques ainsi que de la mémoire, fonctionnant à faible tension. Le développement de techniques de conception adaptées à la faible tension et à la faible consommation sont ainsi requises pour deux raisons : D'une part, les progrès récents en téléphonie mobile et équipement portable ont accru les besoins en circuits mixtes faible-tension et basse-consommation. D'autre part, les progrès en procédés de fabrication CMOS sont déterminés par la vitesse de fonctionnement des systèmes numériques qui augmente grâce à la réduction continue des longueurs de grille. La diminution des longueurs de grille conduit à baisser les tensions d'alimentation pour éviter le claquage de grille.

Enfin, les convertisseurs analogiques-numériques sont des blocs indispensables comme interfaces des systèmes intégrés mixtes récents. Des techniques particulières comme la modulation Delta-Sigma permettent de réaliser des convertisseurs haute résolution, de vitesse moyenne, robustes, en technologie CMOS standard. Au contraire des circuits numériques, les circuits analogiques voient leur consommation augmenter quand la tension d'alimentation diminue [Sansen98]. Une estimation précise des éléments parasites est alors un bon moyen pour limiter la consommation de ces circuits.

2.4 Objectifs du travail

Les objectifs de ce travail étaient ainsi définis :

- la définition d'une méthode compétitive de conception adaptée aux circuits analogiques réutilisables
- le développement de prototypes d'outils associés à la méthode. Il s'agit de :
  - la création d'un outil dédié au dessin des masques CAIRO, indépendant de la technologie et permettant de prendre en compte les contraintes spécifiques de la conception analogique
  - l'adaptation de l'environnement de dimensionnement COMDIAC, permettant la capture de l'expertise du concepteur
- la validation de la méthode et des outils associés par le biais de la conception d'un bloc IP analogique avec des performances exigeantes. Nous avons choisi les modulateurs Delta-Sigma très faible tension comme démonstrateur de notre méthode, dans une réalisation en capacités commutées, pour plusieurs technologies cibles.

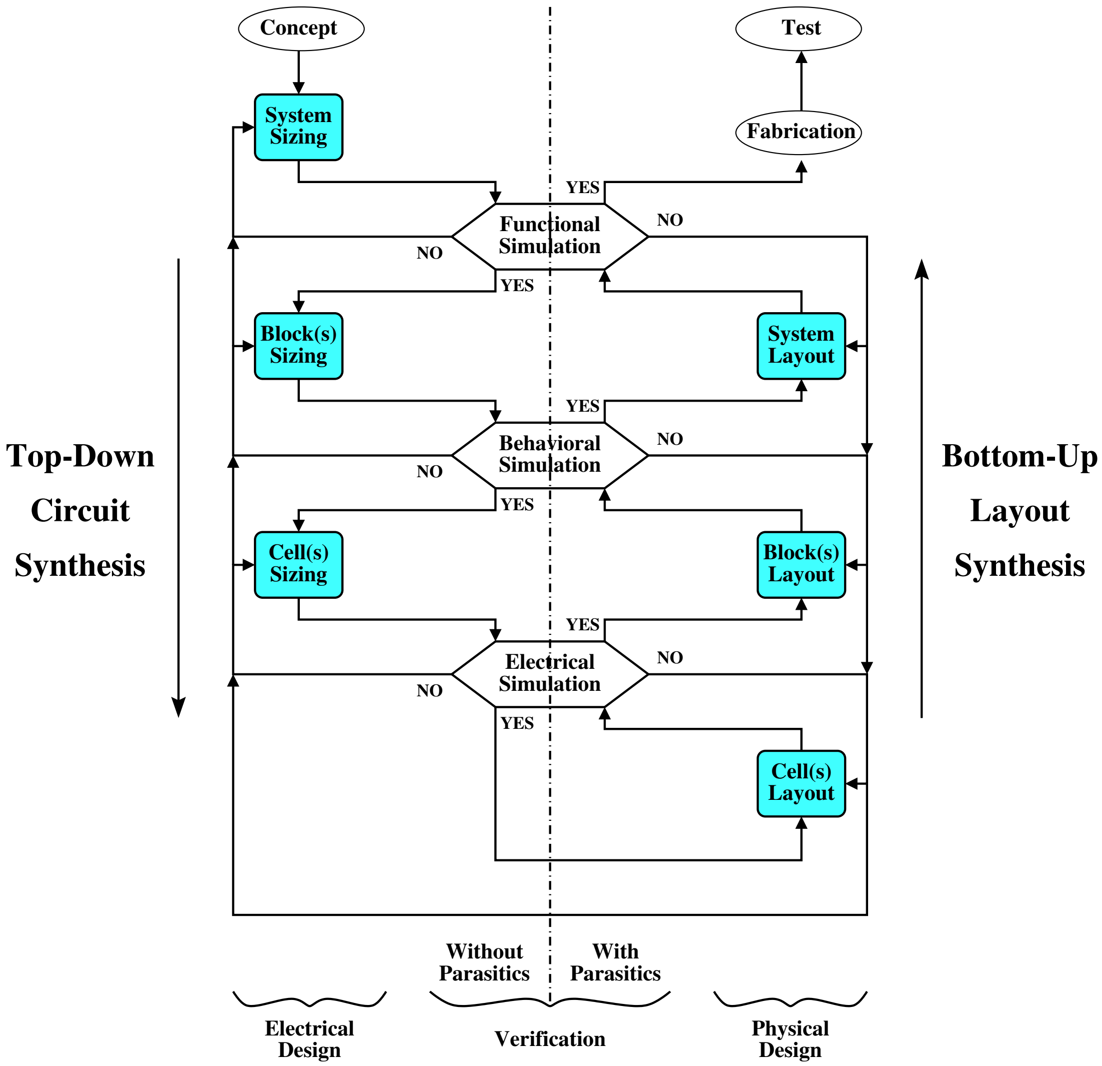


Figure 1: *Étapes principales de la conception mixte.*

3 Etat de l'Art

La figure 1 présente les étapes principales de la conception mixte :

- Dimensionnement du système : Pendant cette phase on choisit l'architecture du système complet. Le système est ensuite décomposé en une interconnexion hiérarchique de blocs fonctionnels dont les spécifications résultent de celles du système complet. L'automatisation de cette phase n'est possible que pour les systèmes dont l'architecture est fixée. L'exemple d'un système de pilote de vidéo est donné dans [Chang97], un autre [Donnay97] présente la synthèse, suivant trois méthodes différentes, d'un système d'acquisition analogique.

- Dimensionnement des blocs : Les blocs sont définis en tant que fonction indépendante avec une interface robuste qui permet de distinguer clairement le bloc de son environnement.

Il s'agit de boucles à verrouillage de phase, de convertisseurs analogiques-numériques ou de convertisseurs numériques- analogiques. Des outils de CAO ont été développés pour accélérer la conception de tels systèmes. Le cas d'un outil dédié à un convertisseur analogique-numérique cyclique est traité dans [Jusuf90], celui de la synthèse d'un convertisseur numérique-analogique en courant commutés CMOS dans [Neff95], et un ensemble d'outils dédiés aux modulateurs Delta-Sigma est présenté dans [Medeiro95] [Medeiro99].

- Dimensionnement des cellules : Les cellules sont définies comme des fonctions de base, de complexité réduite, utilisées pour construire un bloc, comme les amplificateurs et les oscillateurs. Dans cette phase, il s'agit de déterminer les tailles des composants élémentaires (transistor, capacité, résistance) du circuit associé à chaque cellule de manière à atteindre les performances requises pour le bloc. Les méthodes de dimensionnement sont classées en deux catégories : celles utilisant le savoir-faire esixtant, et celles utilisant l'optimisation.

- Les masques au niveau des cellules : Il s'agit ici de générer les masques des cellules à partir de la netlist dimensionnée et d'informations additionnelles sur les éléments parasites, l'appariement et les performances attendues. Les approches existantes peuvent aussi être classées suivant deux catégories : l'une basée sur l'utilisation du savoir-faire, ne pouvant être appliquée qu'à des topologies de circuit fixées, et l'autre, plus générale, basée sur l'optimisation. Dans la première catégorie on trouve que le savoir-faire peut être stocké sous une forme procédurale [Owen95] ou à travers l'utilisation de bibliothèques de topologies [Koh90], ou à l'aide de gabarits [Conway92] ou à travers une série de règles explicites [Bexten93]. Dans la seconde catégorie on trouve des approches qui réalisent le placement à l'aide d'algorithmes d'optimisation, suivi par la phase de routage [Rijmenants89], [Cohn91], [Lampaert95].

- Les masques au niveau des blocs : Quelques outils dédiés à des applications bien définies ont été développés pour automatiser la génération des masques [Jusuf90], [Neff95].

- Les masques au niveau du système : Les masques du système complet sont obtenus par placement et routage des différents blocs qui composent le système.

3.1 Le contrôle des éléments parasites

Deux approches ont été utilisées pour contrôler automatiquement les éléments parasites :

1. La classification des signaux : Dans [Rijmenants89], on trouve que les signaux sont classés suivant leur caractère plus ou moins critique, de manière à minimiser les parasites sur les signaux sensibles et le couplage entre signaux bruités pendant le routage. Dans [Cohn91], le placement et le routage reposent sur la minimisation des divers éléments parasites pondérés

ainsi que sur le respect des contraintes d'appariement intégrés dans une fonction de coût. Cependant, aucune stratégie claire n'est proposée pour fixer les différents poids de la fonction de coût qui doivent être fixés par le concepteur en fonction de son expérience.

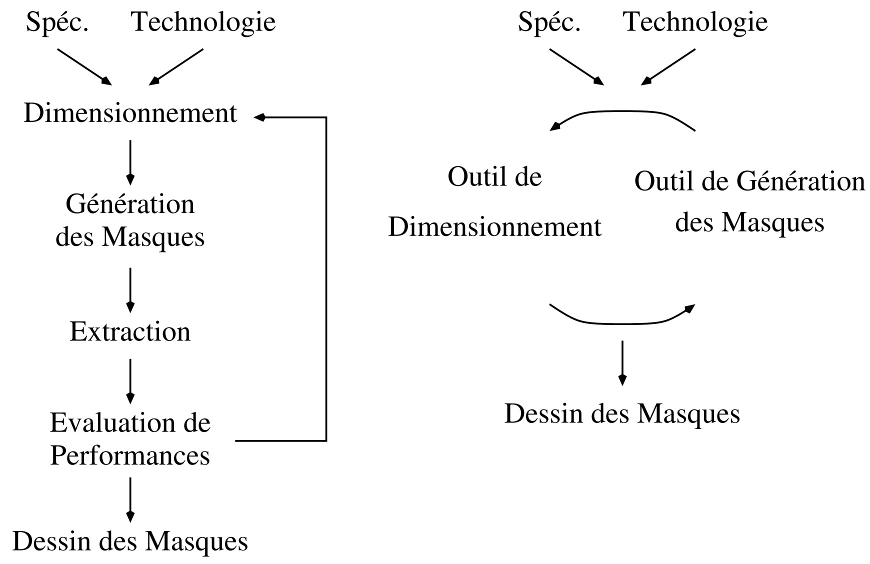
2. L'optimisation sous contrainte : Plus récemment, des outils de génération des masques basés sur la définition de contraintes et utilisant une analyse de sensibilité des performances du circuit [Choudhury90b], [Charbon93] ont été proposés pour le placement [Charbon92] et le routage [Choudhury90a]. La méthode de [Lampaert95] a éliminé la phase intermédiaire d'élaboration de contraintes physiques en optimisant le dessin des masques directement à partir des contraintes sur les performances électriques du circuit. Cependant, le temps CPU requis pour satisfaire les contraintes est toujours grand, ce qui limite l'utilisation de cette méthode à des circuits à faible nombre de composants.

3.2 Propriété intellectuelle analogique et migration technologique

Un des problèmes majeurs qui se pose lorsque l'on souhaite réutiliser un circuit analogique est celui de la migration technologique. On doit toujours passer par une étape de redimensionnement avant de pouvoir porter un circuit analogique d'une technologie à une autre. Une solution est d'utiliser des outils de dimensionnement des cellules. Cependant, du fait qu'il existe une forte interaction entre les différents niveaux hiérarchiques d'un circuit analogique, on est souvent obligé d'effectuer des modifications au niveau des blocs et des compromis au niveau des cellules. La plupart des outils de dimensionnement ne permettent pas l'interaction du concepteur. Une autre approche consiste à développer des outils de synthèse dédiés à une application particulière [Jusuf90] [Neff95] [Medeiro95]. Le développement de générateurs spécifiques nécessite un effort considérable, demande du temps et représente un travail de préparation et de suivi qui suppose que le concepteur du générateur ait acquis une parfaite compréhension du fonctionnement du circuit et puisse améliorer le générateur au fur et à mesure de son expérience. Un des inconvénients de cette approche est qu'un générateur ne prend en compte qu'un certain nombre de paramètres, bornés, en fonction de l'architecture du circuit ce qui limite l'espace des solutions. Plus récemment des expériences pour réutiliser un circuit existant ont été menées en utilisant un raisonnement qualitatif [Francken99] ou un système à base d'optimisation [Phelps00]. Dans les deux cas la plupart des intentions du concepteur original sont perdues dans les réalisations ultérieures.

4 Méthode de conception orientée dessin des masques

Chaque fois que l'on cherche à utiliser à nouveau un bloc analogique existant dans un contexte différent, que ce soit un système différent qui nécessite une modification des performances, ou le même système dans une autre technologie, on doit toujours modifier les tailles des composants



(a) (b)

Figure 2: *Méthodologie de compensation des éléments parasites: (a) traditionelle et (b) proposée.*

du circuit. Vu la quantité de paramètres qu'il est souhaitable de conserver d'une réalisation à une autre, la façon la plus efficace de concevoir des blocs réutilisables est d'inclure l'information adéquate, concernant aussi bien la synthèse que les masques, lors de la conception du premier circuit.

Ce travail propose une méthode de conception analogique pour la réutilisation, basée sur des plans de conception. Un plan global de conception est constitué par une succession d'étapes qui comprennent d'une part des équations analytiques et des modèles fonctionnels/comportementaux pour la synthèse au niveau système, accompagnés d'une méthode permettant d'en déduire les paramètres des blocs dans les niveaux hiérarchiques inférieurs. D'autre part, pour ce qui concerne la synthèse bas niveau et la génération des masques, la conception s'appuie sur des outils de CAO basés sur la définition de gabarits. Par ailleurs, la conception d'un circuit repose sur une forte interaction entre les phases de dimensionnement et de génération des masques afin d'accélérer le cycle de conception, d'en améliorer la qualité et de faciliter la migration dans une autre technologie.

On résout souvent le problème de compensation des éléments parasites résultant du dessin des masques par un processus itératif montré sur le flot de conception de la figure 2(a). Une conception traditionnelle met en oeuvre une suite laborieuse de boucles enchaînant le dimensionnement du circuit, puis la génération des masques, l'extraction de la netlist avec les éléments parasites et l'évaluation des performances du circuit en prenant en compte ces parasites.

Dans ce travail, le dimensionnement et la génération des masques ne sont plus considérés comme deux phases distinctes (voir figure 2(b)). Cette approche est une extension de celle qui a été proposée dans [Onodera90]. Pendant le dimensionnement, l'outil de dessin des masques est utilisé pour calculer les éléments parasites résultant d'une certaine réalisation physique. Cet outil doit être d'une part rapide car il est susceptible d'être appelé plusieurs fois lors du dimensionnement d'un circuit. D'autre part, les solutions de placement obtenues à chaque itération, doivent être proches de manière à aider la convergence des éléments parasites. Ces considérations nous ont conduits à utiliser une approche basée sur la capture du savoir-faire, qui repose sur le concept de gabarits ainsi définis :

- le gabarit éléctrique : il définit une topologie de circuit sans aucune information sur les tailles des composants,
- le gabarit physique : il définit à la fois le placement relatif des éléments et leur routage pour une topologie de circuit fixée, sans aucune information sur la tailles des composants ou le facteur de forme final de circuit.

Un premier dimensionnement est réalisé d'après les spécifications sur les performances, en supposant qu'aucun transistor n'est replié. Après cette phase d'initialisation, les informations quantitatives suivantes sont transmise à l'outil de dessin des masques :

- les tailles des transistors (L et W),
- les courants de drain et de source des transistors,
- des précisions sur la réalisation physique de certains composants (appariement, ...),
- un paramètre physique global définissant le facteur de forme du circuit total.

Muni de ces données ainsi que du gabarit physique du circuit, l'outil de dessin des masques est exécuté dans un mode particulier appelé *mode de calcul des parasites*. Dans ce mode, il n'y a pas de réelle génération de masques, l'outil calcule seulement les éléments parasites et transmet en retour à l'outil de dimensionnement :

- le style du transistor physique (i.e. le nombre de repliements, la surface de diffusion, ...),
- les capacités parasites de routage, y compris les capacités de couplage,
- la surface exacte des caissons pour permettre le calcul des capacités de caissons flottants.

Ce procédé est itéré jusqu'à obtenir la convergence des parasites. L'outil de dessin des masques est alors exécuté dans un mode appelé *génération* où les masques sont réellement créés d'après le gabarit physique.

En comparant les figures 2(a) et (b), on remarque que les itérations visant à compenser les parasites existent toujours. Cependant, la grande différence est que cette boucle de compensation a été automatisée grâce au mode *calcul des parasites* de l'outil de masques, ce qui offre les avantages suivants :

- On peut comparer diverses réalisations physiques des mêmes composants,
- On ne peut pas séparer le comportement des composants de leurs réalisations physiques, ce qui est particulièrement important dans le cas des inductances intégrées.
- On peut optimiser certaines caractéristiques du circuit en exploitant la possibilité de minimiser les capacités parasites.
- On peut prendre en compte certaines contraintes physiques lors du dimensionnement.
- On raccourcit le temps de conception total en supprimant les itérations manuelles laborieuses dimensionnement-masques-extraction-simulation.
- On garantit que le circuit réalisé atteint les spécifications attendues car on a tenu compte des éléments parasites.

Bien que nous ayons remarqué que les niveaux hiérarchiques d'un circuit analogique soient mal définis, l'utilisation de la hiérarchie reste un des moyens les plus efficaces pour gérer la complexité d'un circuit. Il faut noter que les dégradations de performances dues aux parasites peuvent provenir de plusieurs niveaux hiérarchiques. Par ailleurs, afin d'obtenir un circuit final rectangulaire, ce qui préférable pour faciliter le plan de masse au niveau système, il faut contrôler la forme des blocs en fonction de la forme du circuit total, puis la forme des composants plus élémentaires en fonction de la forme des blocs. C'est pourquoi nous avons choisi l'approche descendante aussi bien pour la synthèse que pour le dessin des masques, dans cette approche orientée réalisation physique. Générer les masques d'un circuit suivant une approche descendante suppose que l'optimisation globale de surface puisse jouer sur la forme des cellules dans les différents niveaux hiérarchique grâce à une propagation descendante des contraintes.

5 Dessin procédural des masques avec calcul des parasites

Afin de pouvoir être conforme à notre approche, l'outil de génération des masques doit satisfaire les conditions suivantes :

- Il doit comporter une méthode précise de calcul des parasites.
- Il doit permettre de respecter les contraintes des masques analogiques.
- Il doit permettre différentes réalisations physiques d'un même composant.

L'outil de dessin des masques se présente sous la forme d'un langage, appelé CAIRO, composé d'un ensemble de fonctions écrites en langage C. Ce langage constitue un sur-ensemble de Genlib, ensemble de primitives physiques [Pétrot94].

Les rectangles grisés de la figure 3(a) montrent les différents ressources du langage, il s'agit de :

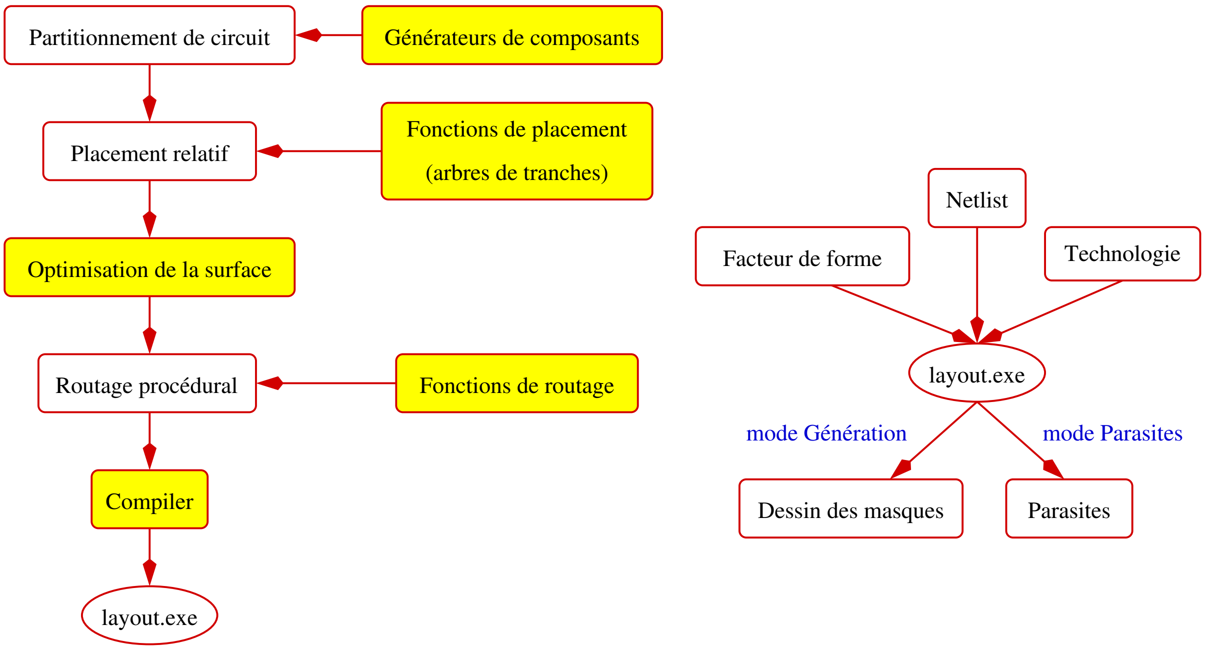


Figure 3: *Langage CAIRO: (a) La description du savoir-faire (b) Génération des masques et calcul des parasites*.

- des générateurs optimisés de composants simples,
- des fonctions de placement relatif,
- un algorithme original d'optimisation de la surface,
- des fonctions de routage relatif,
- un script de compilation basé sur le compilateur du langage C.

La description des masques du circuit, réalisée avec les fonctions de CAIRO, est compilée puis liée avec la bibliothèque de CAIRO. Le programme exécutable permet de prendre en entrée une netlist dimensionnée, une contrainte sur la taille du circuit total et un fichier de paramètres technologiques. Il peut délivrer en sortie soit les éléments parasites associés à la netlist, soit le dessin des masques physiques, voir figure 3(b). Il faut souligner le fait que la description des masques est indépendante des tailles des composants et de la technologie.

5.1 Les contraintes des masques analogiques

S'il est indispensable que l'outil CAIRO soit rapide, il est aussi capital qu'il puisse satisfaire les contraintes analogiques. Il s'agit de :

- Contraintes sur les éléments parasites : Tous les transistors sont construits à partir d'un motif élémentaire qui permet de déterminer la position, la largeur, et la nature des connecteurs et des fils de connexion. Ceci fournit un degré de liberté supplémentaire pour contrôler le couplage entre signaux à l'intérieur même d'un transistor en fonction des applications [Wolf99]. Les transistors dont la grille est très large peuvent être générés sur plusieurs empilements. Le repliement des transistors diminue les capacités parasites de diffusion par rapport au substrat. On peut ainsi minimiser la capacité parasite d'un signal en jouant sur le nombre de repliements connectés sur ce signal. Ce contrôle des éléments parasites permet d'améliorer les caractéristiques fréquentielles d'une réalisation physique.
- Contraintes d'appariement : Les miroirs de courant constituent un des cas où l'appariement entre les transistors est déterminant. Nous avons développé un algorithme dédié au dessin des masques de miroirs de courant. Cet algorithme prend en compte le sens du courant dans la grille et garantit le maximum d'entrecroisements entre transistors centrés, autour du centre de l'empilement.
- Contraintes de fiabilité : Elles sont essentielles pour le fonctionnement du circuit à long terme. Ainsi les largeurs des fils dans chaque composant, des fils de routage et le nombre de contacts sont calculés d'après le courant de polarisation qui les traverse, de manière à respecter la densité de courant maximale permise par la technologie.

5.2 Hiérarchie et optimisation de la surface

Pendant la construction d'un module, CAIRO suit une méthode de placement hiérarchique, basé sur les arbres de tranches [Conway92]. Pour décrire le placement, les composants élémentaires sont assemblés en groupes, tranches et modules.

Le circuit dans son ensemble doit satisfaire une contrainte sur la hauteur ou sur le facteur de forme. Pour respecter cette contrainte, on utilise un algorithme descendant la hiérarchie qui minimise la surface, voir figure 4. Cet algorithme est hiérarchique, ce qui veut dire que les tranches de niveau supérieur peuvent contenir des sous-circuits existants qui, à leur tour, contiennent plusieurs tranches.

5.3 Extraction des parasites

Après la phase d'optimisation de surface, l'emplacement, la forme et la dimension de chaque composant sont connus avec précision. L'outil CAIRO peut alors calculer les éléments parasites associés à chaque composant de base grâce à un modèle basé sur la géométrie des masques. De même, connaissant précisément la position des composants, de leurs connecteurs et des fils de routage, on peut en déduire aisément les capacités parasites par rapport au substrat dues aux fils. Ainsi, dans le mode *calcul des parasites*, tous les élément parasites peuvent être déterminés sans que soient réellement générés les masques physiques.

```
OPTIMIZE_SLICE(HS)
Phase 1:
  FIND the initial set of group heights h_i;
Phase 2:
  DO {
      FIND the widest group j (w_j = WS);
      FIND ΔH such that
          when h_j = h_j + ΔH
              w_j = f_j(h_j) < WS;
      /* Try to compensate ΔH by the other groups */
      FOR each group i ≠ j
          WHILE (ΔH > 0)
          DO {
          h_i = h_i − Δh_i such that w_i = f_i(h_i) < WS;
          ΔH = ΔH − Δh_i;}
      IF (ΔH <= 0)
      /* ΔH is compensated by the other groups */
      THEN
          Conserve the new set of heights;
      ELSE
          Exit;
  };
```

Figure 4: *Algorithme d'optimisation de la surfance.*

5.4 L'independance technologique

On a mis au point une variante de l'approche des masques symboliques sur grille fixe [Greiner90]. L'idée directrice de cette approche est que, si les rapports entre largeurs des rectangles et distances bord à bord diffèrent d'une technologies à une autre, les distances entre axes varient de façon quasi homothétique avec la technologie. Les masques sont construits en utilisant des objets appelés symboles, définis soit par un seul point dans le cas des contacts, soit par deux points dans le cas des segments et des transistors. Les symboles sont placés sur une grille isotropique dont les axes sont distants de 1 λ dans les deux directions. Par ailleurs, on définit une transformation affine pour calculer les dimensions physiques réelles des masques rectangulaire dans une technologie donnée, à partir des dimensions symboliques et d'un fichier de paramètres technologiques [Greiner95]. On a adapté cette méthode aux masques analogiques en introduisant le placement relatif d'objets déformables et en proposant une bibliothèque de générateurs de composants optimisés. Dans les générateurs de composants élémentaires comme le générateur de transistors ou de capacités, on autorise le placement hors grille symbolique ainsi que les dimensions non entières. En fait, on utilise les règles inverses de la transformation symbolique vers réel pour calculer les dimensons symboliques à partir des dimensions réelles.

6 Dimensionnement d'un circuit en présence de parasites

Les objectifs de l'environnement de synthèse analogique COMDIAC sont doubles : D'une part, l'idée est de faciliter la capture du savoir-faire sous la forme d'un enchaînement d'étapes de synthèse. D'autre part, l'idée est de proposer des procédures de synthèse rapides qui permettent une exploration de l'espace des solutions guidée par le concepteur. COMDIAC offre au concepteur beaucoup de degrés de liberté pour tester diverses solutions.

Le dimensionnement d'un circuit analogique ne peut pas se résumer à une simple procédure algorithmique qui conduirait à une solution unique respectant toutes les spécifications. La philosophie de COMDIAC est d'optimiser une ou deux spécifications les plus essentielles au bon fonctionnement du circuit et de laisser le concepteur fixer d'autres paramètres pour satisfaire au mieux *manuellement* le reste des spécifications. Les estimations que peut fournir COMDIAC sont du même ordre de grandeur que la précision des simulateurs électriques car les modèles des composants sont les mêmes. On utilise une approche hiérarchique qui fait appel au dimensionnement des schémas de base implantés dans COMDIAC. Enfin, on peut choisir indépendamment le modèle de calcul des composants élémentaires, la technologie et la procédure de dimensionnement, ce qui permet de dimensionner un même circuit avec différents modèles et pour différentes technologies.

6.1 Exemple de dimensionnement : OTA

La figure 5 décrit la procédure de synthèse d'un amplificateur implémentée dans COMDIAC. Elle prend en entrée un fichier technologique, un gabarit de netlist, un ensemble de spécifications et la polarisation des transistors. Pour effectuer la synthèse, il faut fixer un premier ensemble de spécifications. On utilise un sous-ensemble des entrées. Il s'agit de :

- la tension d'alimentation V_{DD},
- le courant de polarisation I ou le produit gain-bande GBW,
- la marge de phase PM,
- la capacité de charge C_L.
- la polarisation des transistors.

Les tensions de polarisations V_{DS} et V_{EG} de chaque transistor sont maintenues constantes dans la boucle de dimensionnement. L'initialisation se fait en fixant les longueurs de tous les transistors à la valeur minimale permise par la technologie. Puis, à chaque itération, on augmente la longueur de chaque transistor, on calcule les largeurs des transistors grâce aux équations analytiques de synthèse et on en déduit les paramètres petits signaux suivant le modèle électrique choisi. La longueur de certains transistors peut être fixée par le concepteur. A la fin de l'itération en cours, on

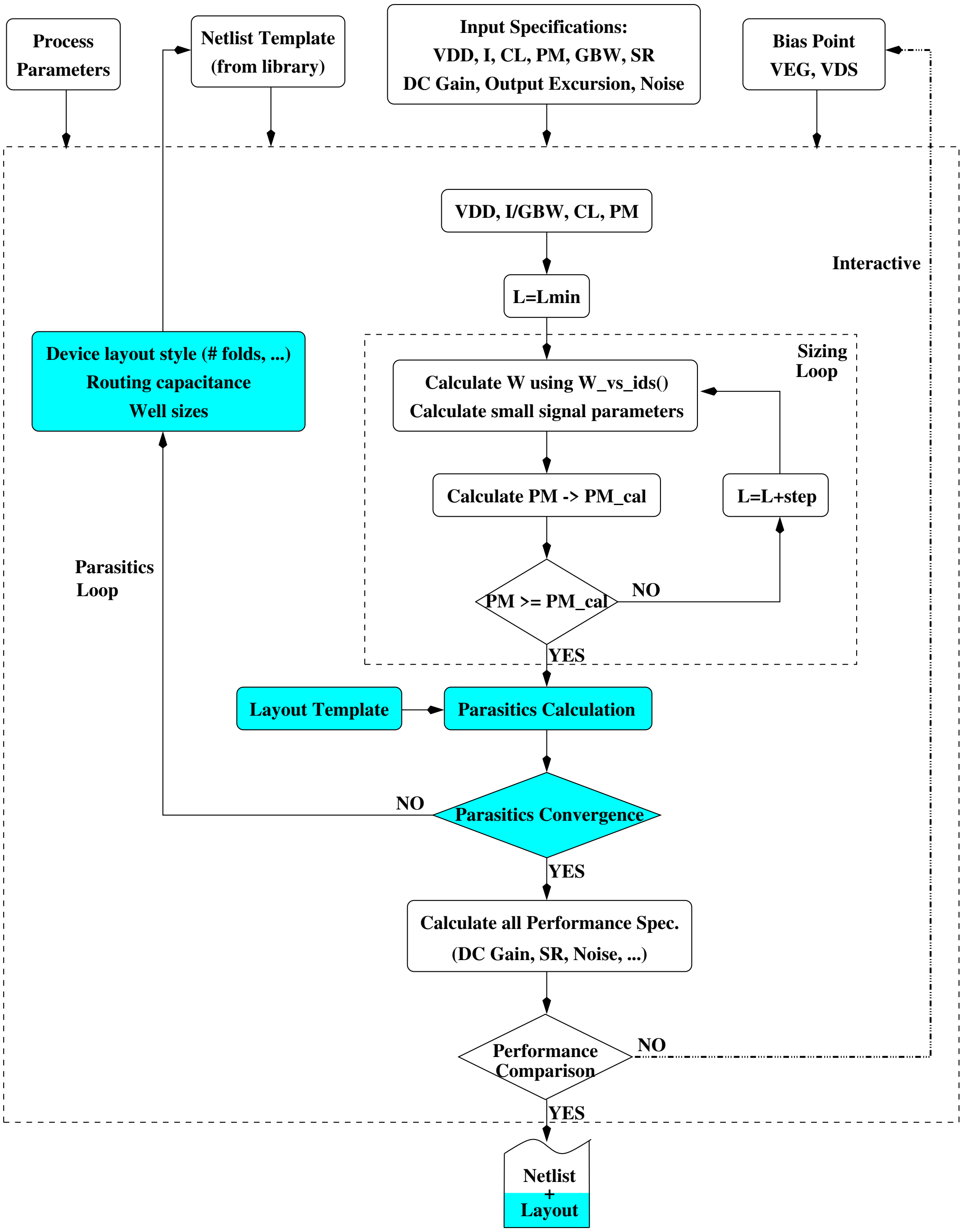


Figure 5: *Procédure de synthèse d'un amplificateur*.

calcule la marge de phase. Lorsque la marge de phase souhaitée est atteinte, on appelle l'outil de masques CAIRO pour calculer les éléments parasites associés au circuit. Tant que la convergence des éléments parasites n'est pas obtenue, on rappelle la boucle de dimensionnement en prenant en compte le nombre de repliements des transistors déterminé par CAIRO. Lorsque la conver-

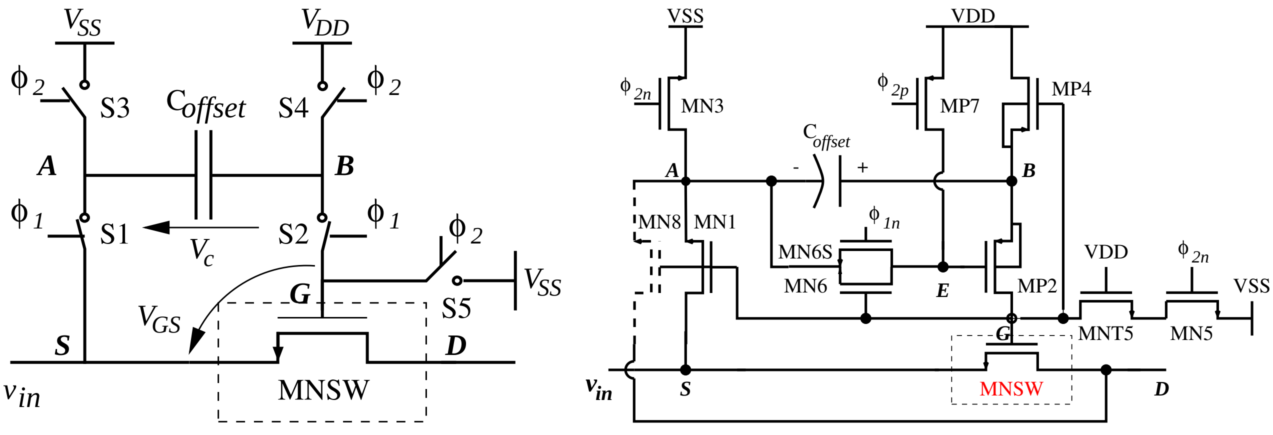


Figure 6: *(a)L'interrupteur bootstrap et (b) réalisation en transistors.*

gence des éléments parasites est obtenue, on effectue l'estimation des caractéristiques restantes. Ces caractéristiques peuvent être optimisées d'une manière interactive par le concepteur en choisissant convenablement les tensions de polarisation des transistors. En effet, en fixant le point de polarisation de chaque transistor d'après des considérations sur l'appariement et la dépendance en température, on augmente la fiabilité des circuits réalisés. Le fait que la procédure de dimensionnement soit très rapide et très précise, permet une exploration interactive par le concepteur d'un grand nombre de solutions.

Dans l'environnement COMDIAC, notre travail a porté sur l'introduction de la boucle sur les parasites et sur l'introduction de nouvelles prcédures décrites dans les paragraphes suivants.

7 Conception d'un circuit à capacités commutées faible tension

Le problème majeur du fonctionnement des circuits à capacités commutées en faible tension est la valeur de la conductance des interrupteurs. Dans ce chapitre on propose deux configurations pour résoudre ce problème. Les deux solutions sont basées sur un circuit d'interrupteur "bootstrap", faible tension[Brandt96] dont le schéma est donné figure 6(a). Les interrupteurs S3 et S4 chargent la capacité C_{offset} pendant ϕ_2 à V_{DD}. Pendant ϕ_1, les interrupteurs S1 et S2 introduisent la capacité préchargée en série avec la tension d'entrée v_{in}, en imposant sur la tension grille-source du transistor MNSW la tension V_C ($\approx V_{DD}$) présente aux bornes de la capacité. Ce montage permet au transistor MNSW de commuter pour v_{in} entre V_{DD} et V_{SS}. La réalisation en transistors de cet interrupteur "bootstrap" est donnée figure 6(b). Les transistors MN1, MP2, MN3, MP4 et MN5 correspondent aux cinq interrupteurs idéaux respectivement S1 à S5. Les autres transistors ont été ajoutés pour étendre le fonctionnement de tous les interrupteurs de V_{SS} à V_{DD} tout en limitant toutes les tensions grille-source à V_{DD}.

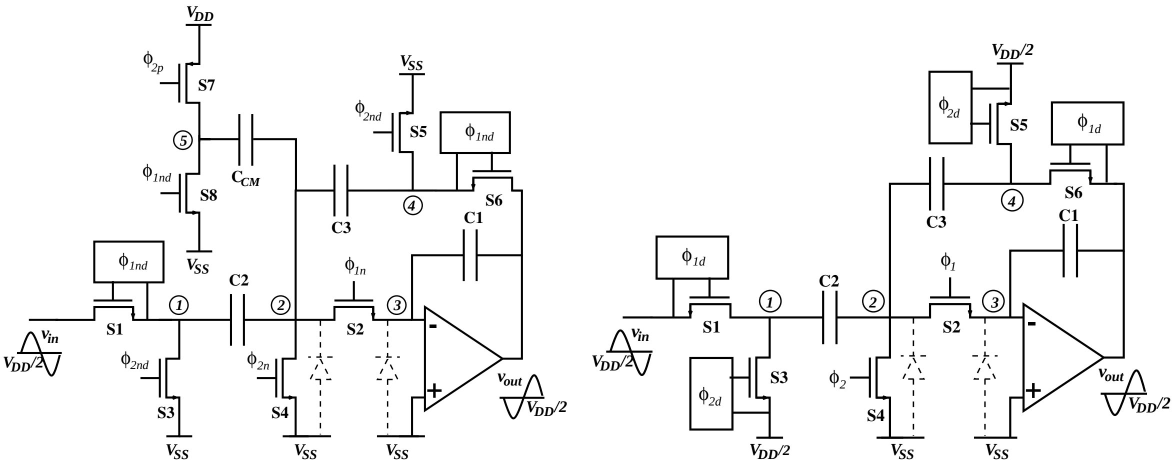


Figure 7: *Implémentations en capacités commutées d'une section passe-bas.*

Nous avons proposé une implémentation en capacités commutées d'une section passe-bas du première ordre fonctionnant sous faible tension (figure 7(a)). Afin de maximiser la conductance des interrupteurs en mode passant, on utilise la tension V_{SS} comme tension de référence ce qui permet d'utiliser des simples transistors NMOS comme interrupteurs. Cependant, la tension de repos à l'entrée et à la sortie du circuit est fixée à $V_{DD}/2$ pour maximiser la dynamique du signal. La différence de tension entre l'entrée et la sortie de l'amplificateur est compensée par l'injection d'une charge constante à travers C_{CM} à chaque cycle d'horloge [Baschirotto97b]. L'inconvénient de cette technique de compensation de charge est d'introduire une nouvelle capacité et donc d'augmenter le niveau de bruit blanc. Par ailleurs, une erreur dans la valeur de C_{CM} crée une tension de décalage, et tout le bruit d'alimentation V_{DD} est injecté sur le chemin du signal.

Nous avons proposé une autre technique qui évite cette capacité de compensation (figure 7(b)). On utilise deux tensions de référence : V_{SS} à l'entrée de l'amplificateur, commutée avec un interrupteur simple NMOS, et la tension $V_{DD}/2$ à la sortie de l'amplificateur et à l'entrée du circuit pour maximiser la dynamique du signal. Pour commuter cette tension, il est nécessaire d'utiliser l'interrupteur bootstrap.

7.1 Amplificateur faible tension

La structure de base de l'amplificateur est montrée figure 8. Elle est composée de deux étages et utilise la méthode de Miller pour effectuer la compensation. Il est nécessaire de régler la tension de mode-commun à la sortie de chacun des deux étages. Ainsi le courant du transistor M5(M6) a été partagé en deux transistors identiques entrecroisés (M51, M52 et M61, M62) avec les grilles connectées aux sorties du premier étage (noeuds n3 et n4). Les transistors agissent comme un

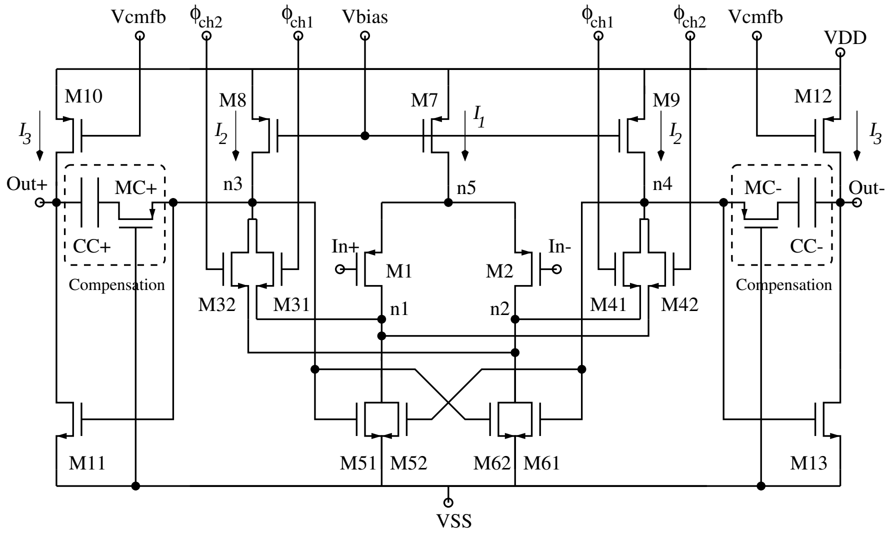


Figure 8: *Amplificateur faible tension*.

circuit de contre réaction de mode-commun qui mesure le mode-commun en sortie du premier étage, le moyenne à travers les transistors en parallèle M51/M52(M61/M62) et régule le mode-commun grâce au courant de polarisation. Le deuxième étage est composé d'un transistor NMOS en source commune M11(M13) avec une charge active M10(M12). Dans ce cas, on peut utiliser un circuit de contre réaction de mode- commun passif [Castello85].

On a également introduit une technique de stabilisation "chopper" [Hsieh81] pour éliminer le bruit en 1/f. La modulation d'entrée peut en effet être réalisée facilement avec quatre interrupteurs. La sortie du premier étage seul est modulée en utilisant deux transistors cascodes supplémentaires M32 et M42 en parallèle avec les transistors existants, mais dont les sources sont connectées aux noeuds n2 et n1 respectivement. Les grilles des deux cascodes sont commandées par deux horloge recouvrantes (ϕ_{ch1} et ϕ_{ch2}) à la fréquence moitiée de la fréquence d'échantillonnage.

8 Conception d'un modulateur Delta-Sigma très faible tension

La conception du modulateur comprend quatre étapes principales :

1. La synthèse haut niveau : On part des performances attendues pour le système et on choisit l'architecture du modulateur la plus appropriée. Enfin, on détermine les coefficients du modulateur.

2. Performances des blocs intermédiaires : Une fois l'architecture choisie, on construit les modèles de fonctionnement non-idéaux des blocs qui composent le modulateur. Ceci permet de trouver les performances que doivent atteindre les différents blocs pour que les performances soient respectées au niveau du système.

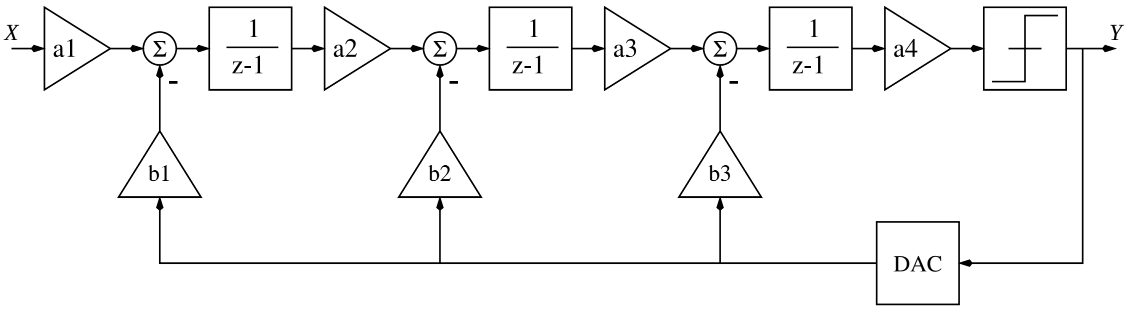


Figure 9: *Schéma bloc du modulateur*.

	Interstage Coeff.	Feedback Coeff.
Premier Integrateur	$a_1 = 0.10$	$b_1 = 0.10$
Second Integrateur	$a_2 = 0.27$	$b_2 = 0.18$
Troisième Integrateur	$a_3 = 0.31$	$b_3 = 0.17$
Comparateur	$a_4 = 4.35$	

Table 1: Coefficients du modulateur

3. Synthèse bas niveau : Il s'agit de dimensionner la netlist de chaque bloc en respectant les performances déterminées à l'étape précédente.

4. Dessin des masques : On génère les masques du circuit complet en utilisant les gabarits des blocs.

8.1 Synthèse haut niveau

On souhaite réaliser un modulateur Delta-Sigma de précision voisine de 14 bits pour une application numérique audio, qui fonctionne sous très faible tension ($V_{DD} = 1V$) en technologie CMOS standard. La figure 9 montre le schéma bloc du modulateur. Il est basé sur une chaîne d'intégrateurs, mono-bit, avec une contre réaction distribuée. On a déterminé les coefficients du modulateur grâce au "Delta-Sigma Toolbox" [Schreier] pour MATLAB, en suivant la procédure donnée dans [Adams96]. Le tableau 1 montre les valeurs obtenues pour les coefficients.

8.2 Performance des blocs intermédiaire

On a construit des modèles pour chacun des blocs en faisant apparaître les effets non-idéaux correspondant à l'implémentation physique du circuit. On a étudié en particulier :

- le gain fini de l'amplificateur utilisé dans l'intégrateur,
- le produit gain-bande de l'amplificateur,

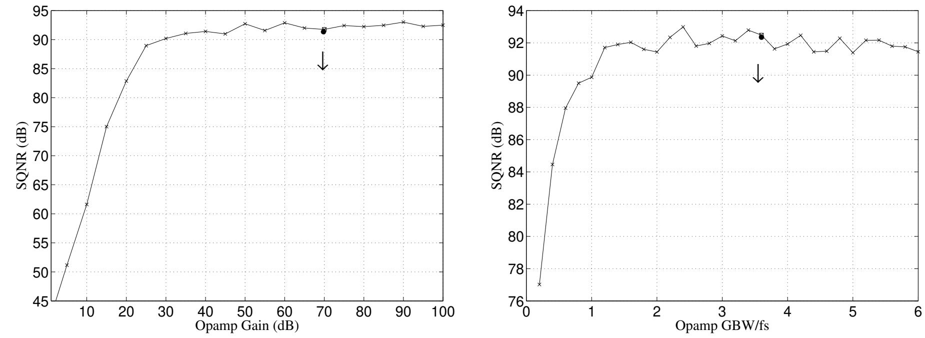


Figure 10: *Le rapport signal-à-bruit fonction (a) du gain de l'amplificateur et (b) du produit gain-bande de l'amplificateur/f_s.*

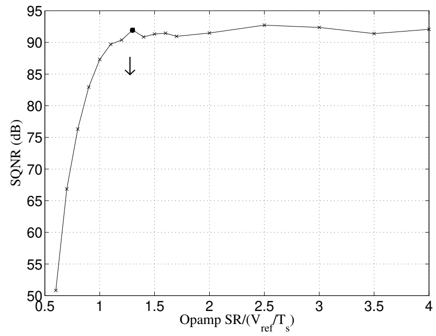


Figure 11: *Le rapport signal-à-bruit fonction du slew-rate de l'amplificateur*.

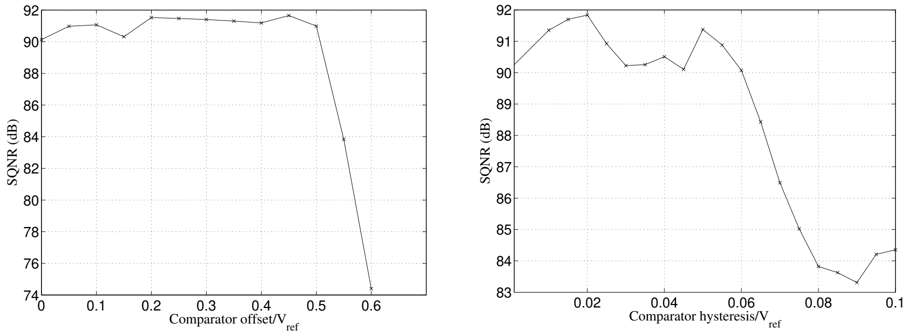


Figure 12: *Le rapport signal-à-bruit fonction (a) de la tension de décalage et (b) de l'hystérésis du comparateur*.

- le slew-rate de l'amplificateur,
- la tension de décalage et l'hystérésis du comparateur,
- la résistance non-nulle des interrupteurs.

Ces modèles sont utilisés pour évaluer la dégradation du niveau de bruit du fait des non-idéalités des blocs. Les figures 10, 11 et 12 présentent les conséquences des défauts des blocs sur le rapport signal-à-bruit du modulateur. On a indiqué par un point les valeurs retenues pour les performances des blocs.

8.3 Synthèse bas niveau

La figure 13 montre la procédure du dimensionnement qui aboutit au schéma en transistors dimensionnés. Les procédures de synthèse relatives à l'intégrateur, à l'amplificateur et à l'interrupteur ont été implémentées dans l'environnement COMDIAC présenté à la section 6. La figure 14(a) montre la procédure de dimensionnement de l'intégrateur qui elle même utilise celle de l'amplificateur. On a porté une attention particulière au bruit de l'intégrateur, à la dynamique du signal et à l'erreur d'établissement. Toutes les caractéristiques de l'amplificateur ont été analysées. Il s'agit du gain, de la fréquence de transition, de la dynamique de sortie, de la capacité d'entrée et du bruit. La procédure de dimensionnement des interrupteurs est résumée à la figure 14(b). On a ainsi dimensionné séparément chaque interrupteur dans le modulateur. Ceci a permis d'améliorer la taille des interrupteurs qui, fonctionnant sous faible tension, doivent être plus grands que ceux que l'on rencontre ordinairement dans les circuits à capacités commutées.

9 Les circuits réalisés

Le modulateur a été implémenté dans une technologie 0.35-μm CMOS standard avec deux niveaux de poly, cinq niveaux de métal et un double caisson. Les masques ont été générés de manière hiérarchique en utilisant le langage CAIRO présenté à la section 5. Le code décrivant chaque bloc a été instantié dans les blocs de niveau hiérarchique plus élevé. Les circuits d'horloge ont été réalisés avec la chaîne ALLIANCE [LIP] et instantiés dans la description en langage CAIRO du modulateur. Comme vérification ultime, le circuit a été extrait au niveau transistor puis simulé sous Eldo. La figure 15(a) montre la photographie du circuit. La figure 15(b) montre le spectre de sortie dans la bande passante pour un signal d'entrée d'amplitude 6dB et de fréquence 3.2kHz. Le tableau 2 résume les performances obtenues pour le modulateur.

9.1 Réutilisation

Le même modulateur a été porté dans une autre technologie 0.35μm avec les mêmes performances attendues. On a utilisé les mêmes résultats de la synthèse haut-niveau. Cependant, il

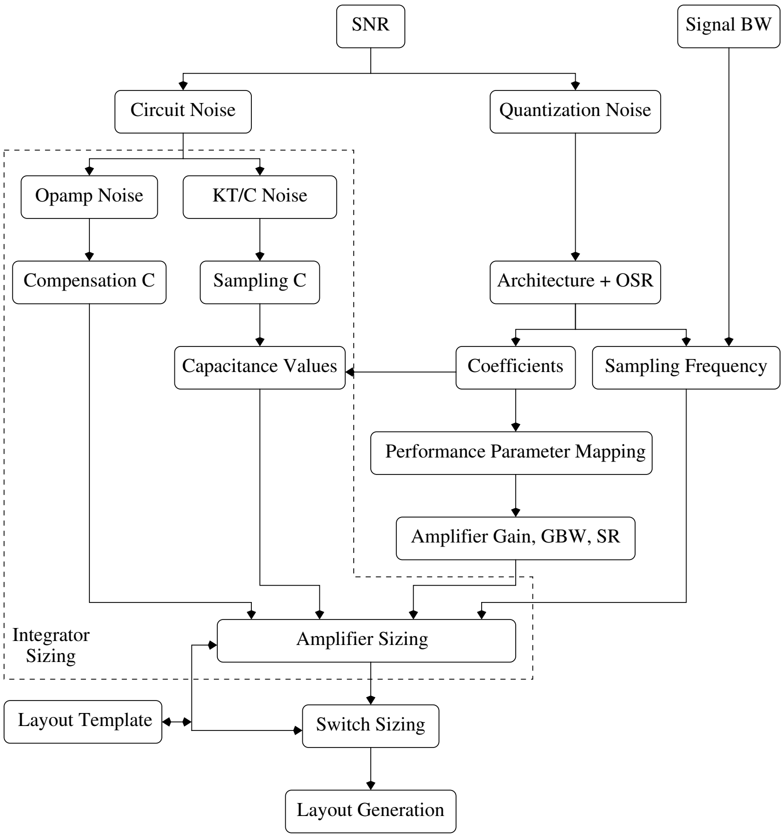


Figure 13: *La procédure du dimensionnement du modulateur*.

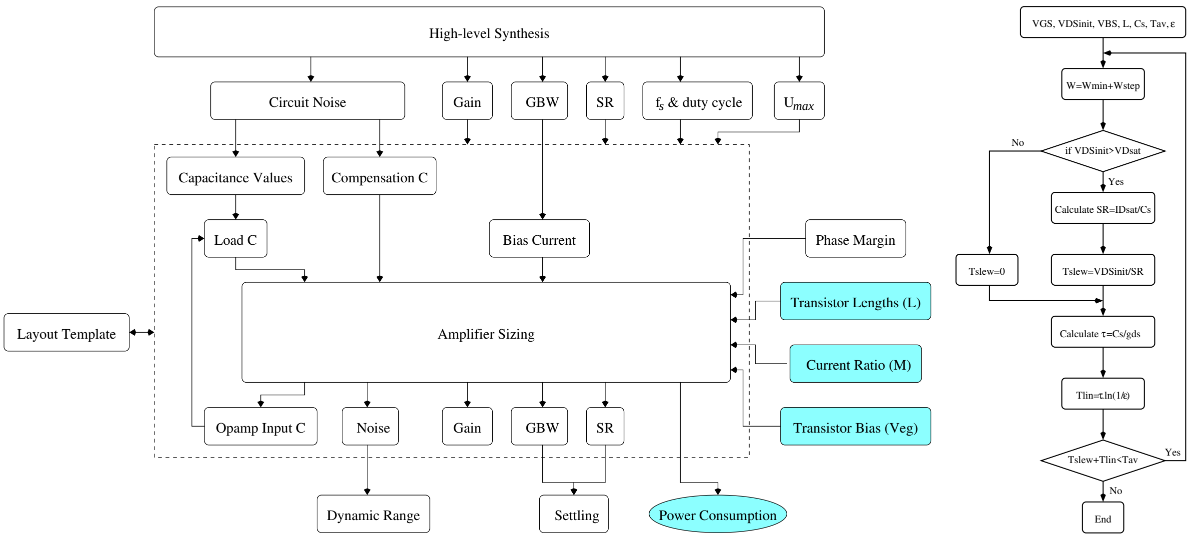


Figure 14: *La procédure de dimensionnement (a) de l'intégrateur et (b) l'interrupteur*.

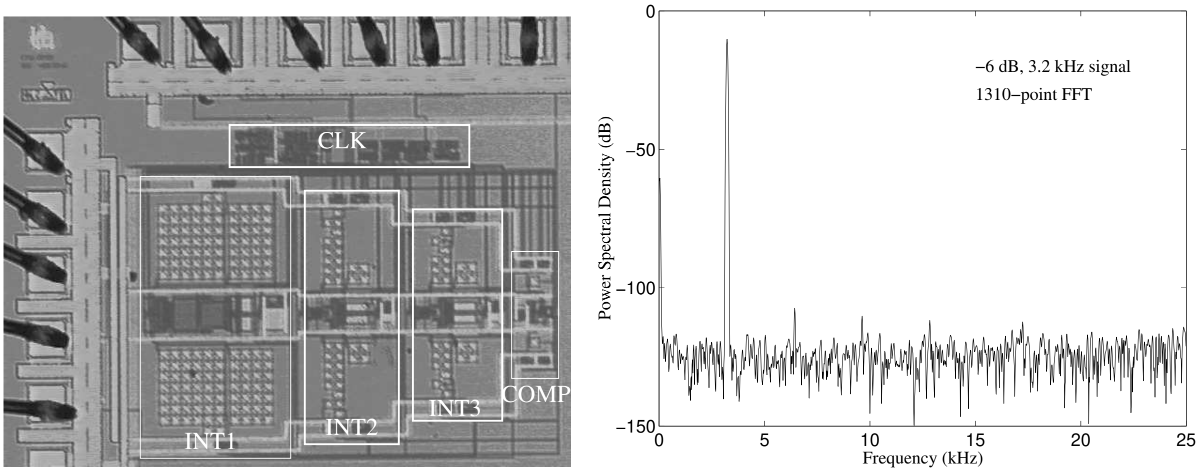


Figure 15: *(a) La photographie du circuit et (b) le spectre de sortie dans la bande passante.*

Tension d'alimentation	1V
Tension de référence	1V
Dynamique d'entrée	88dB
SNR / SNDR max	87dB / 85 dB
Nombre de bits	14
Rapport de suréchantillonnage	100
Fréquence d'échantillonnage	5MHz
Bande passante	25kHz
Consommation	950 μW
Facteur de mérite $\times 10^6$	275
Surface	0.9mm $\times$ 0.7mm
Technologie	0.35-μm CMOS TMDP

Table 2: *Les performances obtenues.*

a été nécessaire de refaire le dimensionnement des intégrateurs et des interrupteurs à cause des changements de paramètres technologiques. Les mêmes gabarits ont été utilisés pour générer les masques avec des modifications mineures. Les masques du modulateur complet sont donnés figure 16(a). La conception du second modulateur a pris seulement une semaine, grâce à la réutilisation des procédures de dimensionnement et des gabarits de masques.

Afin d'examiner un autre mode de réutilisation, nous avons conçu un autre modulateur qui utilise les mêmes blocs faible tension que le précédent, et donc le même savoir-faire. Nous avons choisi un modulateur du quatrième ordre [Coban99]. La figure 16(b) montre les masques de ce

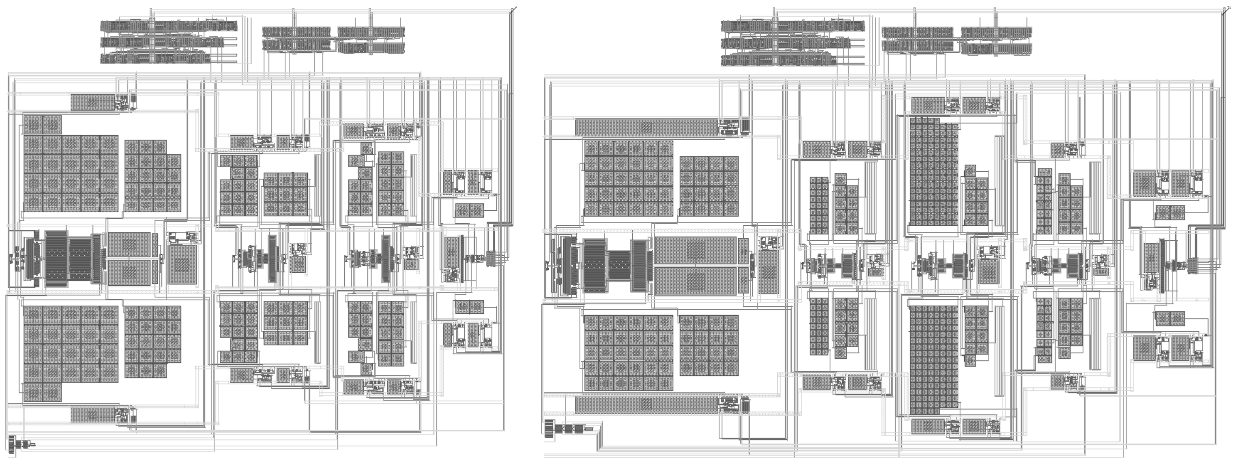


Figure 16: *Le dessin des masques de modulateurs: (a) troisiḿe ordre et (b) quatriéme ordre dans une autre technologie 0.35μm.*

modulateur. Comme on a réutilisé les gabarits de masques, le plan de masse de ce modulateur est similaire au précédent, mis à part l'étage d'intégration supplémentaire. La conception jusqu'au dessin des masques était terminée en deux semaines.

10 Conclusion

Ce travail a présenté un méthode de conception en vue de la réutilisation, basée sur l'intégration des phases de synthèse électrique et physique. La méthode est fondée sur la capture du savoir-faire du concepteur sous la forme de procédure de dimensionnement, utilisant deux outils de CAO : COMDIAC et CAIRO. L'efficacité de la méthode a été démontrée par la réalisation d'un modulateur Delta-Sigma faible tension, faible consommation. Le caractère réutilisable de la conception a été expérimenté de deux manières différentes : D'une part en concevant (des spécifications jusqu'aux masques) le même modulateur dans une autre technologie, et d'autre part en concevant un modulateur du quatrième ordre avec des spécifications plus exigeantes et une topologie différente, mais avec les mêmes blocs de base.

Ce travail a porté sur différents aspects de la conception analogique assistée par ordinateur ainsi que sur la conception analogique elle-même. Parmi les problèmes rencontrés, nous pensons que les points suivants devraient être approfondis :

- La méthode de conception orientée dessin des masques a été utilisée de manière automatisée uniquement au niveau des cellules. La méthode utilise une approche hiérarchique descendante pour permettre la propagation des contraintes physiques ainsi que des parasites d'un niveau hiérarchique à un autre. Cette propriété pourrait être exploitée plus à fond.

- La description des fils de routage avec CAIRO est assez fastidieuse et il est très difficile de décrire un routage qui suive toutes les déformations possibles des composants. Un routeur automatique serait très utile, à condition de donner les informations sur les éléments parasites.

- Explorer l'espace des solutions est possible dans COMDIAC, mais suppose l'interaction du concepteur. Il serait intéressant d'examiner l'apport éventuel d'un outil d'optimisation. Il serait aussi souhaitable d'améliorer l'architecture logicielle de COMDIAC pour faciliter l'introduction de nouvelles procédures.

- Sur le plan des circuits faible tension basse consommation, il serait intéressant d'étudier des applications haute fréquence et de voir leur compatibilité avec la très faible tension. Cependant une fréquence plus haute requiert une consommation plus élevée. Il faudrait donc trouver des techniques pour limiter la consommation.

Chapter 1

Introduction

1.1 Motivation

The complexity of integrated electronic circuits being designed nowadays is continuously increasing as advances in process technology make it possible to create mixed-signal integrated SoC designs. Most parts of these SoC's are completely digital rather than analog blocks. This is because in the digital domain, noise has much less influence on the quality of signal processing than in the analog one. In addition, logic synthesis, layout and verification of digital circuits are highly suited for design automation methodologies which make it easier for the designer to implement his/her function and reduces the overall time-to-market. But since the real world is an analog place, true SoC designs must include at least some analog interfacing functions. Analog design automation lags behind its digital counterpart and becomes in many cases a limiting factor in accelerating SoC time-to-market.

In addition, as SoC's are becoming larger, the only way to efficiently design such dense SoC's is by embedding cores, also called IP blocks, on these chips. Ideally, these cores should be reusable, pre-characterized and pre-verified. This means that the same core can be used on different chip designs and in different technologies after migration. While this concept is currently having *some* success on the digital side of mixed-signal systems, it is still extremely difficult to reuse an analog IP block in its actual form.

Design reuse of analog IP blocks will thus gain more importance in the coming few years especially with the rapid changes in fabrication technologies led by the digital system needs [Association99]. Analog cells would have to be migrated to these new technologies with minimal manual modifications. While analog design automation methodologies are not yet widely accepted by analog designers, design reuse will soon be a huge driving force.

1.2 Contribution

The contribution of this work is threefold: First, we propose a design methodology for analog circuit design reuse based on the integration of both electrical and physical design. Secondly, we present the tools supporting the methodology; namely, a technology independent procedural layout tool that takes into account analog-specific layout constraints, and a knowledge-based circuit sizing environment. Finally, as a case study, very low-voltage low-power switched-capacitor circuits are considered. Design solutions are proposed leading to the design and implementation of a 1-V, 1-mW $\Delta\Sigma$ modulator for digital audio applications. Besides being of a particular interest from the design point of view, the circuit also demonstrates the suitability of the proposed methodology and CAD tools for high-performance mixed-signal circuits.

Methodology: In order to promote analog design reuse, the presented work proposes a new design methodology based on a layout-oriented circuit synthesis approach. A global design plan is constructed. This plan contains design choices, steps, heuristics and main tradeoffs from high-level system considerations down to layout. During circuit sizing, on one side, the methodology is based on interactive circuit *sizing plans* that allow rapid design space exploration. On the other side, it relies on a technology- and size-independent *layout templates* that contain physical layout information related to the circuit. These templates can then be used:

- During circuit sizing, to calculate both rapidly and accurately all parasitics that appear during physical realizations without any layout generation.
- To generate the layout once all specifications have been satisfied.
- In another re-design of the same circuit topology for different specifications or/and different technology.

The methodology contributes both during the design phase and in future design reuse. Key points are:

- Electrical and physical design integration which is becoming more and more important with advances in fabrication technologies and the continuous increase in operation speed. This guarantees the fulfillment of the required performance specifications.
- Optimization of various design aspects in the presence of parasitics.
- Shortening the overall design time by avoiding laborious sizing-layout generation iterations.
- Analog design reuse, since design knowledge is stored in design plans and layout templates.

CAD Tools: The implementation of this methodology is then studied. This has led to the development of the layout language, CAIRO[1], and the evolution of the circuit sizing environment, COMDIAC[2] [Porte97]. At the layout level, the CAIRO language is characterized by:

- Efficient algorithms taking into account analog layout constraints such as matching, parasitics control and reliability considerations.
- The problem of top-down area optimization subjected to large device size variations under fixed topology is treated.
- The resulting layout generator is independent of device sizes and fabrication technology.

On the circuit level, the circuit sizing environment COMDIAC is characterized by:

- Knowledge-based analog cell sizing approach.
- Hierarchical sizing.
- Detailed Spice-like device models.

Case Study: A Low-voltage low-power $\Delta\Sigma$ modulator has been selected as a design application. This study has led to new circuit architectures and building blocks that allow very low-voltage, robust, SC circuit operation in standard CMOS technologies. This includes a special locally-bootstrapped low-voltage switch that allows rail-to-rail signal switching while avoiding any gate-oxide overstress. The switch constant overdrive also enhances considerably circuit linearity. Further reduction in power consumption is obtained through a modified two-stage low-voltage differential opamp. We have then designed, fabricated, and tested a third-order, 1-V $\Delta\Sigma$ modulator. Measurements show that for an OSR of 100 the modulator achieves a dynamic range of 88 dB, a peak SNR of 87 dB and a peak SNDR of 85 dB in a signal bandwidth of 25 kHz, and dissipates 1 mW. Obtained results show the feasibility of very low-voltage SC circuits using the proposed circuit techniques.

The complete design methodology for the modulator IP block is presented. As a result of using the proposed layout-oriented methodology and the developed CAD tools, parasitics are accurately taken into account during the design phase. The sizing tool has also allowed to investigate a wide range of design space points. In addition, the time needed to re-design another similar $\Delta\Sigma$ modulator is greatly reduced. This has been investigated in two different ways: First, by re-designing (from specifications to layout) the same modulator in a different fabrication process, and secondly, by re-designing a fourth-order one with more demanding specifications. These two designs, however, were not fabricated.

[1] CAIRO stands for "**C**ircuits **A**nalogiques **I**ntégrés **R**éutilisables et **O**ptimisés".
[2] COMDIAC stands for "**COM**pilateur de **DI**spositifs **AC**tifs".

1.3 Outline

This section gives a brief overview of the contents of the following chapters:

After a brief introduction in chapter 1, chapter 2 defines the context of the thesis. Motivations are introduced and the main objectives are clearly assigned.

Chapter 3 introduces the mixed-signal design process, it also contains a brief presentation of the state of the art CAD tools and methodologies towards analog design automation and design reuse.

In chapter 4, the proposed design methodology is presented. Advantages of the layout-oriented synthesis is then discussed and its impact on analog IP reuse is investigated.

Chapter 5 discusses the implementation of the layout generation language CAIRO. First, the requirements imposed by the proposed methodology are presented. A thorough discussion of various analog layout constraints follows, together with the corresponding algorithms. Different implementation decisions are then presented. The chapter terminates with a layout example.

Chapter 6 introduces the modifications introduced in the circuit sizing environment COMDIAC. It starts with the presentation of the main characteristics of the tool. The sizing method is presented and applied on an op-amp. Extensions necessary for the proposed methodology are also presented. Finally, a detailed sizing example is given using different parasitics considerations followed by a comparison between each case.

Chapter 7 begins the application part of this work. It starts by introducing the main problems of SC circuit operation under very-low voltages together with the existing techniques to overcome them. The proposed technique is then presented accompanied with original circuit architectures.

Chapter 8 presents the design methodology used to design a reusable very low-voltage $\Delta\Sigma$ modulator. First, high level synthesis is performed to determine system-level parameters from the required specifications. Then, mapping of system requirements to building block specifications is performed based on block behavioral modeling and discrete-time simulations. Finally, low-level synthesis of each block is done using the tools presented in chapters 5 and 6.

Chapter 9 presents the implementation of the $\Delta\Sigma$ modulator together with some design trade-offs. Measurement results of the fabricated modulator are presented and compared with similar prototypes. Finally, design reuse is investigated through the design of two additional modulators.

Chapter 10 includes some concluding remarks together with possible directions for future work.

Chapter 2

Problem Definition and Motivation

2.1 Introduction

This chapter defines the context of the thesis by defining the problem then by introducing work motivations and objectives.

Design automation and design reuse are two faces of the same CAD coin, both aim to shorten design times and share a lot of tools. First, in order to appreciate the analog CAD challenge, major analog design issues are presented in section 2.2. Differences between analog and digital design are clearly identified.

In section 2.3, SoC design based on IP blocks is introduced with emphasis on analog design reuse. Different forms of an IP are discussed and compared.

Some basic concepts and definitions related to design automation and design reuse, that distinguish similarities and major differences, are given in section 2.4.

In order to validate this work, a high-performance IP design example must be investigated. Section 2.5 introduces the motivations behind choosing very low-voltage low-power $\Delta\Sigma$ modulators as a case study.

In section 2.6, the objectives of this work are carefully assigned.

2.2 Analog Design Particularities

Before discussing any issue related to analog CAD or design reuse, analog design characteristics and particularities that distinguish it from the heavily automated digital design must be clearly identified. Analog design differs to a great extent from the digital one, mainly in the following aspects [Gielen91]:

- **Loose form of hierarchy**: Hierarchical levels are not so as strictly defined and certainly not as generally accepted as in the digital domain. Voltages, currents and impedances must all be considered at all levels of hierarchy. This leads to a close interaction between all levels.

- **Large spectrum of performance specifications**: The set of performance characteristics for analog circuits contains many more specifications than digital ones. In addition, the importance as well as the value of a given specification can widely vary depending on the application.

- **Critical device sizing**: In spite of the small number of transistors per cell compared to digital circuits, device sizing is more involved because there is a much stronger interaction between the electrical characteristics of each individual device and the performance of the global circuit.

- **Large variations in device sizes**: Device sizes in the same circuit can vary over wide ranges [Koh90]. Depending on the performance specifications, it is not uncommon to see two orders of magnitude variations in device sizes. During the layout phase, these large devices can be designed in many different shapes and/or be divided into sub-devices if necessary, and their terminal configuration may vary, too.

- **Large range of circuit schematics**: The same function can be implemented with various circuit topologies each suited to a class of applications.

- **Big influence of layout**: Because of the rather wide range of parameter spreads in IC fabrication, the behavior of the circuit depends largely on its corresponding layout. While some circuit techniques exist that cancel out first-order effects caused by variations in key parameters, second-order effects (such as matching, symmetry, device orientation, ...) dominate performance [Chang97]. Thus layout techniques that enforces matching and symmetry of some critical devices are of major importance. A good understanding of the circuit behavior is inevitable during layout synthesis. Consequently, as opposed to digital layout, minimum area is not among the first concerns, other considerations, related to circuit performance after fabrication, are of more importance to analog layout.

- **Large influence of technology**: Process, biasing, temperature variations and layout parasitics strongly influence the circuit performance and can even change the functionality of the circuit. The matching precision obtainable in a given process is also an important parameter during physical design.

- **Interactions on the system level**: The accuracy of analog circuits is very sensitive to interactions at the system level. This includes crosstalk and thermal feedback. Layout precautions to isolate analog circuits from sources of noise and heat are thus essential.

- **High-performance applications**: Analog design has been pushed to applications with aggressive performance specifications where a digital implementation becomes difficult. Such applications require a careful design and layout to meet the required performance.

For these reasons analog design does not render itself easily to CAD methodologies. There are some published tools that aim to automate both circuit sizing and layout generation, however, a few number of them has reached the commercial level, not to mention the designers' community acceptance. Some of such approaches will be discussed in the following chapter with emphasis on their ability to handle analog design reuse.

2.3 IP and Analog Cores

Design reuse is not a new concept. The first step undertaken by a design engineer facing a new problem is to look over old designs to find a circuit or at least a topology which may be applied to his problem. Most analog circuits designed nowadays are optimized for a certain application and are rarely used without modification in another one. Some re-design steps are always needed to satisfy the needs of the new application and/or the new technology. Since this *adaptation* is not considered during the first design, it is often a laborious task that takes a long time both for the new design optimization and physical layout creation. This design philosophy will have to change soon to cope with the needs of the emerging SoC's, where embedding multiple IP blocks from different providers is considered as one of the most efficient way to reduce the time to market. In the same time, this allows the system designer to manage the growing complexity of the chip. This is analogous to the split between the board world and the IC world, such that the reuse of IP's in SoC's is analogous to the reuse of IC's on boards.

The term IP is used to describe pre-designed functions that have been protected through patents, copyrights or trade secrets and that are bought and sold in abstract form for incorporation into larger ASIC's [VSIA97]. IP's can be simple cells such as op-amps, voltage controlled oscillators and comparators, or more sophisticated blocks like ADC's, DAC's and PLL's. The system designer is, however, interested by blocks with complex functionality, or in other words a *core*, and so is this work. The actual form for an IP core can vary depending on the way the IP designer provides his core to the system designer. Three different ways of defining a block can be identified [VSIA97] [Lipman98]:

1. **Soft cores**: They are specified by behavioral descriptions. They are more adapted to digital cores where the description, e.g. in VHDL, is process-independent and can be synthesized on the gate level, using pre-characterized cell libraries. The main drawback of this kind of cores is that their performance is not guaranteed since implementing in different processes can result in performance variations.

2. **Hard cores**: They are optimized for a given performance and have a fixed layout in a specific process. They have the advantage of being much more predictable. This is, however, only useful when the same circuit with exactly the same performance specifications is to be used in the same technology. They are thus less flexible.

3. **Firm cores**: They combine some attributes of both soft and hard cores. These cores have some predefined information concerning the structure and topology to make performance more predictable. They, generally, do not include routing. Firm cores offer a compromise between soft and hard ones being more flexible and portable than hard cores, yet more predictive than soft ones.

Designing reusable and interchangeable analog cores is still questionable and not as widely spread as the digital ones due to the following reasons:

- Process migration and layout dependent parasitics urge to resize the design to take into account the new technology parameters and to compensate for the modified parasitics. For example, the simple act of transferring the layout of an amplifier from one process to a different one could result in an unstable design, due to the change of the gain and phase margin, leading to oscillations.

- Changing design performance requirements which usually accompanies the new circuit physical environment. For example, on a SoC, the noise performance requirement of an ADC may be more stringent if parts of the digital electronics run asynchronously to the ADC [Zwan97] due to the additional injected asynchronous noise.

- New block specifications. For example, moving a 12-bit ADC to 16-bit, means, in most cases, re-considering a lot of design choices and tradeoffs both on the architecture and transistor sizing levels. Compare that with moving from a 32-bit digital core to a 64-bit one, which will basically mean doubling the gate count, while essentially keeping the same architecture.

- With the actual RF-design trend, the effect of layout parasitics on the design performance is continuously increasing due to the continuous decrease of device sizes and the corresponding increase in system speed. In spite of the overall shrinking in the circuit dimensions, faster circuit operation often results in relatively higher parasitics.

- Loss of the original designer intentions and considerations. Since in most cases, some parts of the design is based on heuristics acquired through the designer's own experience, it is usually difficult to re-design the given block without knowing a priori these considerations. For example, the layout may include certain matching structures which are key to the circuit performance and must be respected in any re-designing process.

- Analog design can not be separated from the original global design methodology. While for each type of digital circuits, the design methodology hardly changes from one circuit to another, there exists no standard approach for the design of analog circuits. This includes, for example, defining the levels of hierarchy, levels of abstraction for simulations at each hierarchical level and employed behavioral models.

In addition to difficulties in the design, characterization, and reuse of analog IP's mentioned above, other design constraints must be addressed when mixing analog and digital cores on a common chip. Besides the adverse effect of digital cores on analog blocks which includes crosstalk, ground noise and thermal interaction, fabrication technologies are also driven by the optimization of digital designs being the major concern of actual implementations. Analog performance in a digital process is thus worse than in a process geared toward high-performance analog circuits. This makes mixed analog/digital multicore-chip even more difficult.

2.4 Design Automation and Design Reuse

The main objective of electronic CAD is the creation of methodologies and tools for the design of electronic systems, helping designers build functionality while satisfying intended performance specifications [Chang97]. This assistance can be in the form of design automation or design reuse methodologies. While both aim to shorten design times and reduce design costs by improving productivity, each focuses on a different aspect of the design process.

When designing a system for the first time, the designer resorts to every possible CAD tool to support him in acquiring a better understandings and insights of the system, and to efficiently handle every design aspect, this could be a simulator or even an automatic synthesis tool that sizes parts of the system. Once designed a successful system or block that exactly matches the required performance specifications should be well documented. It is often desirable to keep the same design choices, tradeoffs and heuristics for a similar new project, so as not to restart the whole design process from scratch. This significantly increases the probability of a first-pass silicon when starting new projects.

The purpose of design automation is to automate some design tasks, such as automatic sizing, circuit optimization and layout generation [Gielen91]. The ultimate goal is to automate the whole design process from behavioral system description down to layout. The degree of automation is measured as the ratio of the time it takes to design a system for the first time manually to the time it takes with the synthesis tool [Ochotta98].

On the other hand, the purpose of design reuse is to be able to efficiently reuse a previous successful design experience in another system environment and/or in a different fabrication process, with either the same or slightly modified performance requirements. Design reuse is not limited to reusing exactly the same circuit topology, since design knowledge can also be employed to build similar designs using the same approach and cells of the original design. The degree of reuse is measured by the amount of information and experience that is transfered from the successful first design to subsequent ones.

Many concepts and tools are, however, used interchangeably between design automation and design reuse methodologies. For example, the same sizing tool could be well used for both synthesizing the circuit for the first time as well as, after feeding it with the appropriate reuse infor-

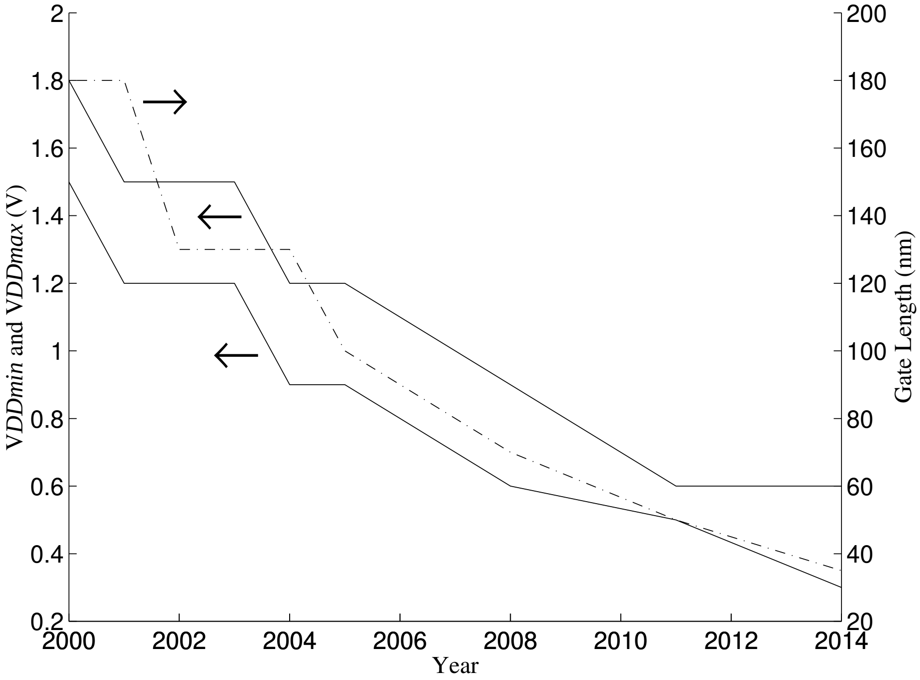


Figure 2.1: *Power supply and Gate length evolution.*

mation, in a design reuse methodology.

2.5 Very Low-voltage $\Delta\Sigma$ Modulator

Very low-voltage low-power $\Delta\Sigma$ modulators have been chosen as a design example of an analog IP block, both for their design interest as well as their adequacy to test the proposed approach and tools.

Future SoC's will require integration of logic, analog and memory on the same chip at low power supply voltages. The need for the development of low-voltage and low-power analog circuit techniques is, thus, twofold: First, the demand on low-voltage low-power mixed-signal circuits is significantly increasing in order to cope with modern advances in portable and battery operated systems. In these systems, low-voltage allows to use fewer batteries for size and weight considerations, while low-power permits to ensure reasonable battery lifetime. On the other side, advances in CMOS technology are driven by the digital system need to enhance the circuit speed performance and increase the integration density by continuously reducing the channel length. Lower channel lengths lead to lower supply voltages. Fig. 2.1 shows the power supply and minimum gate length evolution in the coming years as predicted by the Semiconductor Industry Association technology roadmap [Association99]. A fast and continuous decrease of both the supply voltage and channel length is obvious. The supply voltage is given as a range; maximum V_{DD} corresponds to maximum speed performance while minimum V_{DD} corresponds to the minimum power consumption. The supply voltage is expected to drop down to around 0.6 V for a channel length of 0.05 μm by 2011. The threshold voltage, however, must remain relatively constant to keep the off transistor leakage within tolerable limits.

On the other hand, ADC's are becoming unavoidable building blocks in modern mixed-signal SoC's for interfacing. They are often considered as one of the performance limiting blocks in the system. Special techniques such as $\Delta\Sigma$ modulation allow to build robust high-resolution medium-speed converters in modern IC technologies. For the circuit implementation SC circuits are still good candidates even under very low-voltage (Here, the term very low-voltage is used for circuits that are able to operate on a minimum supply voltage of one gate-source voltage and a saturation voltage $V_{DDmin} = V_{GS} + V_{dsat}$ [Hogervorst96]. This value is around 1V for current technologies). The SC technique is characterized by its robustness and compatibility with modern VLSI fabrication methods. It has been successfully used over the past decades to fabricate most of the analog circuits on the market today. It is thus very well understood by the designers and easy to use to build new circuits. However, under low-voltage, the widely-used clock voltage multiplication technique [Rabii97] can not be employed anymore for critical switches in the circuit due to the gate dielectric reliability limitation [Abo99a]. New circuit techniques are thus needed to allow SC circuit operation under these challenging conditions and maintain, at least, the same SNR. In addition, in contrast to digital circuits, the power consumption of analog circuits increases as the supply voltage decreases [Sansen98]. A good estimate of the circuit parasitics is thus critical for saving power consumption. Over-estimation of layout parasitics may mislead the designer and urges him to increase the drain current to compensate for these parasitics, leading to wasted power and area.

2.6 Work Objectives

Considering both the nature of analog circuits and the needs of multi-core SoC designs, the objectives of this work were defined to be:

- The definition of a design methodology for analog design reuse.
- Prototype development of the associated software tools. This includes
  - An appropriate technology independent layout tool that is able to take into account analog-specific physical design constraints.
  - The adaptation of the knowledge-based circuit sizing environment COMDIAC.
- The validation of the approach and tools through the design of a reusable analog IP block for a challenging application.

As a case study for the proposed methodology, very low-voltage $\Delta\Sigma$ modulators are chosen. For circuit implementation, the SC technique has been selected. In addition to the above objectives, others were then added:

- Study of SC low-voltage techniques and definition of appropriate solutions.
- Implementation of a very low-voltage low-power $\Delta\Sigma$ modulator using the proposed methodology, tools and circuit techniques.
- Re-design the above circuit in a different technology.

Chapter 3

State of the Art

3.1 Introduction

Due to the nature of analog design, analog design automation and design reuse share a lot of concepts and tools. This chapter contains a brief presentation of the state of the art CAD tools and methodologies.

In section 3.2, the mixed-signal design process is previewed. Each phase in the design is clearly identified accompanied with an overview of the associated CAD research work.

One of the important issues is layout parasitics compensation during the design. In section 3.3, the effect of layout parasitics on circuit design is investigated together with solutions proposed in the literature.

Design reuse cannot be discussed separately from technology migration since in most cases reuse is targeted to a new technology. In section 3.4, IP cores and technology migration of analog circuits are discussed.

3.2 The Mixed-Signal Design Process

Fig. 3.1 shows the major steps of the mixed-signal design procedure. Starting from the initial concept and system specifications three major top-down *design* phases can be distinguished, namely: system sizing, block sizing and cell sizing together with their corresponding bottom-up *layout* phases, namely: system layout, block layout and cell layout. After each step, verification-by-simulation must be accomplished. The type of simulation, however, depends on the level of abstraction. Infeasibility results after any step may lead to going-back one or more steps in the design plan to modify previous choices. The simulation is always repeated after layout synthesis, at each step, in order to watch for the effect of layout parasitics on circuit performance. This process is repeated till all system specifications are satisfied. In the following sections, each step is defined in more details together with some corresponding published approaches and CAD tools towards design automation.

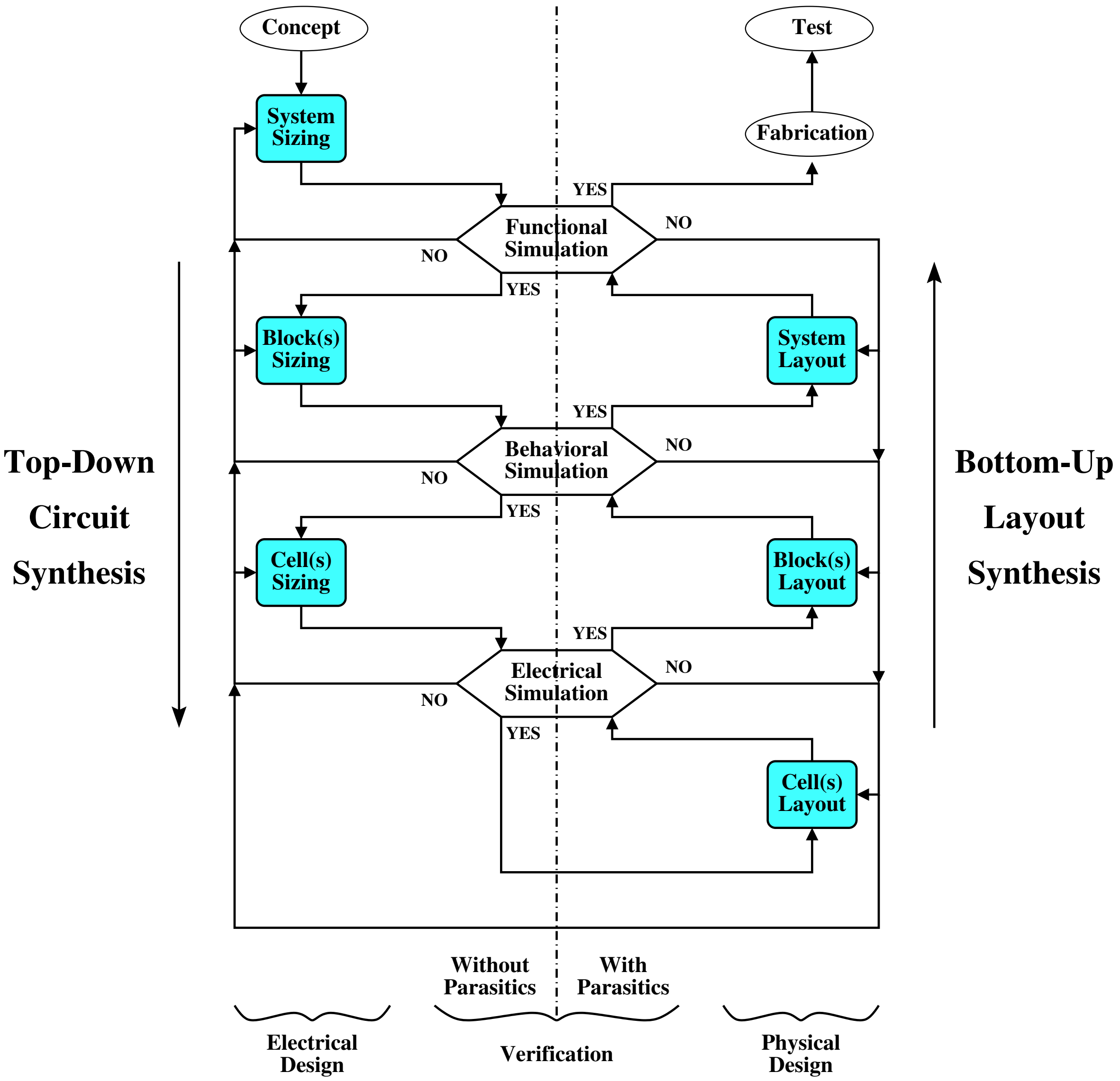


Figure 3.1: *Mixed-signal design process.*

3.2.1 System Sizing

In this phase, an adequate system architecture is chosen. The system is then decomposed into a collection of functional blocks and the specifications for each block are derived from those of the system. This is the one of the most difficult steps to automate since, usually, it has no unique solution and, often, many trade-offs exist which need human expert guidance. In addition, each mixed-signal system has different design objectives and considerations according to the application under consideration and the system environment. Automation of this step is only possible for fixed system architectures. For example, in [Chang97] a video driver system design methodology is presented. The system constraints can be immediately decomposed into DAC and frequency synthesizer constraints. Since the silicon area of low-level blocks cannot be determined at this level, an alternate measure of optimality, namely a defined *flexibility* function, is used. High-level optimization is then done using the supporting hyperplane algorithm. As another example, consider [Donnay97] where an analog sensor interface front-end system is synthesized comparing three different methods; two optimization-based methods using simulations in the loop and equations, and a library-based approach. Furthermore, in [Vandenbussche98] the simulated annealing-based optimization loop is retained. At each iteration a set of block specifications is proposed, the corresponding system performance is simulated using behavioral models for the blocks, and the estimated implementation cost is calculated based on power/area estimators.

A simulation is then performed to verify that the system-level specifications are satisfied with the determined block-level ones using behavioral or functional models. Functional modeling is the most abstract modeling level, it is used to describe complex systems with little accuracy. The connection points between blocks are not conservative but rather indicate a transfer of information as in a signal-flow model.

3.2.2 Block Sizing

Blocks are defined as stand-alone functions with a robust interface that can be easily distinguished from its environment, such as a PLL, an ADC or a DAC. Starting from block specifications determined in the previous step, functional blocks are then synthesized each separately. IP blocks can be used at this level of design for some building blocks in order to shorten the design time. For other blocks where complete sizing is needed, the performance specifications of each block is then mapped to specifications of lower level blocks till the basic cell-level is reached.

Analog and mixed-signal system sizing needs different algorithms for different types of blocks. For example, different ADC architectures (pipeline, flash, delta-sigma, ...) would need different sizing algorithms. Circuit-specific CAD tools have been developed to speed-up the design of such systems. In [Jusuf90], a cyclic ADC synthesis tool is presented. Depending on the supplied specifications, a particular *netlist module generator* is selected. A *subblock requirement generator* is responsible for the generation of all necessary requirements for all components that build a particular type of cyclic ADC's. These requirements are then fed into the *customized routines* to synthesize and generate the complete device sizing and netlist. This process relies heavily on analog-design expertise. Converter performance is verified using a simple customized behavioral simulator and re-sizing is allowed. In [Neff95], automatic synthesis of CMOS current-switched DAC's is addressed using a nonhierarchical approach. A constrained optimization method is coupled with combination of circuit simulation (using HSPICE [Met96]) and DAC design equations. In [Medeiro95] [Medeiro99], a set of dedicated tools for $\Delta\Sigma$ modulator design is presented. The tool uses statistical optimization and a design equation database to calculate cell specifications. A dedicated behavioral simulator is then used for verification.

On this level of abstraction, behavioral simulation is the most efficient way for block verification [Moser97]. During this simulation, all cells are replaced by an appropriate behavioral model. Behavior models can be built in an HDL using algorithmic sequences of statements and differential equations. The connection points between behavioral models represent physical continuous-time signals. They are governed by generalized conservation laws. For this reason, some typical behavior like input/output impedance and power supply can be included in the model. Another way of modeling is by the use of macromodels. Macromodels make use of ideal components, e.g. resistors, capacitors, independent and dependent source, to build a circuit which mimics the behavior of the cell.

3.2.3 Cell Sizing

Cells are defined as smaller basic functions that are used to build a block, such as op-amps and oscillators. During *cell sizing*, a detailed circuit-level schematic is created for each cell, such that all block requirements are satisfied. Many design automation approaches have been proposed on this design level. In order to be able to compare between them, we begin by defining the basic metrics for cell sizing tools:

- **Accuracy**: the discrepancy between the tool results and those of a detailed circuit simulator.
- **Generality**: the range of circuits and performance specifications handled by the tool.
- **Sizing time**: the CPU time required for sizing.
- **Preparatory expertise**: the design expertise and effort required to prepare a new circuit to be sized by the tool.
- **User interaction**: the designer may wish to add new design constraints or fix some design parameters based on his design experience.
- **Variation tolerance**: the ability of the tool to create circuits that are tolerant of manufacturing process and operating point variations.
- **Technology independence**: the ability to easily change the used technology.

There are two main approaches used for cell sizing:

- **Knowledge-based**: in which detailed analog design knowledge are exploited to perform circuit sizing. This includes topological and analytical knowledge, rules of thumb, heuristics and simplified models. The internal representation of this knowledge can be in the form of rules [El-Turky89], design plans [Harjani89] or hard-coded procedures [Rezania95].
- **Optimization-based**: in which the the analog design problem is formulated in the form of a mathematical routine as a constrained optimization problem, which aims to determine a

Method	Example	Accuracy	Generality	Sizing time	Preparatory expertise	Variation tolerance
K-B	OASYS[Harjani89]	+	- - -	+ + +	- - -	+ +
E-D	OPASYN[Koh90]	+	- -	+ +	- -	+
E-S	ARIANDE[Gielen93]	+	-	-	-	+
S-D	DELIGHT[Nye88]	+ +	+	- -	+	-
S-S	FRIDGE[Medeiro94]	+ +	+ +	- - -	+ +	-

+ : Better

- : Worse

K-B: Knowledge-Based.

E-D: Optimization-based: Equation evaluation - Deterministic update.

E-S: Optimization-based: Equation evaluation - Statistical update.

S-D: Optimization-based: Simulation evaluation - Deterministic update.

S-S: Optimization-based: Simulation evaluation - Statistical update.

Table 3.1: *Comparison of cell sizing strategies*.

specified vector of design parameters (transistor length/width, bias voltages, ...) in order to minimize/maximize some design objectives (power, area, ...) subjected to some constraints (gain, settling time, ...). The optimization process minimizes a *determined* cost function. During this process a loop of two main actions is executed, namely:

– **Update**: during which the vector of design parameters is updated using either a *deterministic updating* [Vanderplaats84] method such as the steepest descent algorithm that follows gradients to the nearest local minima, or a *statistical updating* [Laarhoven87] one such as simulated annealing that uses random movements which are accepted or rejected based on a specific probability function. The main disadvantage of deterministic methods is that they can be trapped in local minima depending on the starting point, whereas statistical ones usually lead to the global minimum.

– **Evaluate**: during which the circuit performance is evaluated after the updating action. This evaluation can be either using *symbolic equations* previously derived for the circuit or a *circuit simulator*. This allows then to calculate the optimization cost function.

Table 3.1 compares these approaches in terms of the above criteria[1]. Since optimization using evaluation-by-simulation uses directly the same simulator used for verification, they have the best accuracy and are more general since generally few circuit-specific information are supplied to the tool. However, optimization-based methods have the highest computational cost and makes it difficult for the user to interact with the tool. This inability to use the designer experience makes the produced design more susceptible to process variations. In order to fix this problem, a new method to include variation tolerance during synthesis has been proposed in [Mukherjee94].

Usually a detailed transistor-level electrical simulation is needed to verify the cell performance specifications.

3.2.4 Cell Layout

Starting from the sized cell netlist, additional information about parasitics, matching and performance constraints, the corresponding *cell layout* is generated. Various layout automation tools have been reported in order to automate the cell layout generation phase. They can also be classified into two main groups:

- **Knowledge-based**: in which the circuit topology is always fixed. A *sound* topological arrangement for the building blocks of the circuit is stored based on traditionally accumulated design experience. Knowledge storage can either be in the form of a procedural layout [Owen95] or through the use of topology libraries [Koh90], by employing a design by example principle (layout templates) [Conway92], or through stored rules [Bexten93].

- **Optimization-based**: employs an optimization algorithm to generate a suitable placement configuration followed by a routing phase [Rijmenants89], [Cohn91], [Lampaert95]. It is fully automated and strives to take a large number of specific analog constraints into account. Information on critical nodes, matching and symmetrical constraints still must be supplied by the user.

As knowledge-based approaches offer short layout generation times, in addition to a reuse of expert knowledge and experience (which seems to be indispensable to the analog domain), they suffer from their high design cost and thus are best suited for frequently used circuits. On the other hand, optimization-based approaches offer *automatic* layout generation which tries to optimize certain aspects of the layout, but they suffer from the complexity of the optimization problem and the difficulty of the appropriate cost function determination which may differ according to the application, this is besides a long layout generation time. They are thus best suited for circuits with small number of devices.

In any of the above approaches, the quality of the final layout depends heavily on the ability of the available *device* generators to take into account analog-specific constraints such as matching,

[1]For a more comprehensive presentation of the existing tools, refer to [Gielen91], [Ochotta98].

symmetry and capacitance minimization by merging [Lampaert99]. Procedural layout generators have been developed that generate a layout for a fixed configuration of devices [Owen95], taking into account common-centroid, interdigitated device pairs and passive components [Bruce96]. Reliability constraints have also been treated in [Wolf99]. They rely on the designer to construct an adequate circuit mapping to available device generators. However, due to the limited set of devices, situations could arise where a potential geometry sharing situation in a circuit topology does not match one of the pre-defined device generators. An alternate approach was proposed in [Cohn91]. It relies on a simple set of procedural device generators of *single* devices. Merging of these primitive devices is then allowed through the use of a sophisticated placement algorithm. However, some commonly used structures can never be constructed simply by merging, e.g. interdigitated multiple transistors. Based on the fact that transistors in a given topology are always placed in stacks due to merging, in [Malavasi95] a stack generator is used to partition a given circuit to find different alternative sets of device merging. All possible sets are generated by the algorithm and a cost function based on critical parasitics and area is used to select the best alternatives. In [Naiiknaware99], more focus has been placed on the stack quality by taking into account area and diffusion as well as routing parasitic capacitance optimization.

A detailed transistor-level electrical simulation is also needed after this step to measure cell performance degradation due to layout parasitics.

3.2.5 Block Layout

Application-specific system-level tools have been constructed to automate the layout of well defined applications. For example, in the cyclic ADC generator [Jusuf90] described in section 3.2.2 the partitioning of blocks are fixed. The layout generation is then performed in a hierarchical bottom-up manner. While in the current DAC [Neff95], the regular nature of current-switched DAC's has lead to the use of a procedural cell tiling layout approach.

Behavioral simulation is used again to verify the block performance. Behavioral models must also include parasitics effects in addition to normal behavior.

3.2.6 System Layout

The whole *system layout* is created by global placement and routing of the individual block layouts. On the system level, cells may include digital as well as analog blocks. Among the issues that must be addressed are crosstalk [Mitra92], substrate noise injection [Mitra96], and power grid distribution [Stanisic94].

Functional or a more detailed behavioral simulation including parasitics must be finally done before system fabrication.

3.3 Layout Parasitics Control

During the design of high performance analog cells, device matching, parasitics, reliability design rules, thermal and substrate effects must all be taken into account. All of these effects can be controlled with a good layout design performed either manually by an expert layout designer or using a dedicated automatic tool. However, the nominal values of performance specifications are subject to degradation due to a large number of parasitics which are generally difficult to estimate accurately before the actual layout is complete. Over-estimation of layout parasitics results in wasted power and area, while under-estimation of parasitics leads to circuits that do not meet the required specifications.

This means that layout parasitics have a strong influence on the behavior and performance optimization of the fabricated circuit. Their effect must be carefully treated both during the design and future design reuse since it varies from one technology to another. Parasitics compensation must then be included in any design methodology. In order to have a sufficient design margin, designers often largely over-estimate layout parasitics. The amount of wasted power and area, however, depends on the experience of the designer and his knowledge of the process. Some re-sizing-layout generation iterations, that include detailed layout extraction and simulation, are needed to fine tune the design. In order to minimize the number of these iterations, some automatic layout tools try to impose parasitics constraints during layout generation. Historically, there exist two main approaches for automatic parasitics control:

1. **Classified Nets**: In [Rijmenants89], nets are classified based on its criticality, trying to minimize parasitics on sensitive nets and coupling between noisy nodes during routing (based on a gridless channel router). The router routes net-by-net in a given priority order: power nets are normally routed first, followed by sensitive nets to ensure the shortest path on the preferential layer to minimize parasitics. Noisy nets are routed last after noncritical ones. The cost function to rank paths includes distance and penalties for crossing or running adjacent to noisy or sensitive nets. In [Cohn91], placement and area routing rely on weighted parasitics minimization and matching constraint enforcement which are integrated in the algorithm's cost function. During placement, device shaping and abutment are performed on MOS transistors in order to minimize diffusion capacitance. However, no clear strategy is indicated for the definition of parasitics weights. This information must be supplied by the designer on the basis of his experience.

2. **Constraint-Driven**: More recently, constraint-driven layout generation tools for placement [Charbon92] and routing [Choudhury90a] have been proposed, generally based on sensitivity analysis of circuit performance [Choudhury90b], [Charbon93]. In [Malavasi96], a methodology for performance-driven layout synthesis is presented, based on the previous tools. High-level constraints are automatically translated into a set of low-level bounds on the parameters (parasitics and geometry) that can be controlled during

layout synthesis. If the layout tools fail to meet one of the derived parasitics constraints, one or more iterations with another set of constraints are needed. In [Lampaert95], performance constraints are used to drive directly the layout tools thus eliminating the intermediate constraint generation step. The tools will either yield a correct layout or will flag the specifications as being impossible to meet, without iterations. However, the CPU time needed to satisfy these constraints is always large, thus limiting the applicability of this method to small cells. In addition, this calculation time could be avoided if small modifications and re-considerations are allowed in the circuit design. If some constraints could not be satisfied, the whole process must be repeated.

3.4 Analog IP and Technology Migration

Due to reasons discussed in sections 2.2 and 2.3, most actual analog and mixed-signal IP cores of complex functionalities are hard ones. In other words, they have a fixed layout for a specific process, thus having well-known performance characteristics. IC foundries and IP providers offer layouts for commonly-used blocks. The problem is that the analog space cannot be fully covered by any finite block library. One of the main problems facing analog design reuse of IP's is technology migration. Technology migration is the ability to port a circuit previously designed and fabricated in a certain process to another one. Due to the ever-shrinking minimum device dimensions and the associated improvement of digital circuit performance, analog and mixed-signal blocks optimized and targeted for a given technology are always required to be migrated to new processes. Function libraries and technology porting have been successfully employed for digital circuits where it mostly suffices to appropriately scale the corresponding layout [Mead80] [Pétrot94]. However, analog circuit performance cannot be guaranteed using the same approach due to reasons discussed in section 2.2, in particular analog critical device sizing, and the large influence of technology on performance characteristics. Some re-sizing must always be performed in order to be able to port any analog circuit to a new technology.

Cell sizing tools (see section 3.2.3) can be used to re-size the different building cells of an IP block given a new technology. However, due to the close interaction between all levels of hierarchy of analog circuits (see section 2.2), block-level modifications and tradeoffs in cell specifications are, in most cases, inevitable. Also, most of the cell sizing tools do not allow user interaction which may become an important consideration for the designer who wants to reuse his own experience. An alternative approach consists of developing specific block synthesis tools as those presented in sections 3.2.2 and 3.2.5 [Jusuf90] [Neff95] [Medeiro95]. For a given class of circuits, the generator first selects appropriate cell circuit topologies from a predefined topology library and then performs transistor-level sizing in order to fulfill the required performance specifications. In this case, design knowledge is stored in the form of a design equation database and heuristics integrated in the software. Process information is usually an input to the generator. All the generated circuits share the same knowledge *source* but are completely independent of each other. Specific block generators can be used by IP providers to generate circuits with different specifications in a relatively short period of time. Block generator development takes a considerable effort and time represented by a heavy preparatory and maintenance work which includes acquiring an excellent understanding of the circuit functionality and formulating the gained experience. This must be justified by an extensive use of the generator. However, the main weakness is that a generator can be only parameterized within certain boundaries on a given architecture. In many practical cases, depending on the application, special requirements on the system specifications may prevent the generated design from being used without modifications that may rise to the architectural level. In addition, since IP's represent in most cases state-of-the-art techniques, designers often don't have the time nor the software competence to work on developing generators.

More recently, design reuse based on an original *working* design has been investigated both through qualitative reasoning [Francken99] and optimization-based synthesis [Phelps00]. The basic assumption is that full access to a working design and the usual documentation archived with such designs is available, but with no access to the designer. In [Francken99], an example of technology porting of analog circuits taking into account the original circuit sizing is presented. Re-sizing and layout generation are done separately. During re-sizing, first an initial *guided* scaling step is performed to produce a starting point which is then fine-tuned, by using qualitative reasoning, to correct for possible violations of certain performance specifications. This is done through a special dependency matrix which describes qualitatively the dependency of each performance specifications on each design parameter. During layout, relative positions as well as aspect ratios of the building blocks are kept constant. All blocks part of the floorplan are then generated automatically [Lampaert95]. The complete layout is regenerated hierarchically in a bottom-up manner by synthesizing each block separately, while assembling is still done manually. In [Phelps00] an equalizer/filter block has been resynthesized from scratch in several design styles. This has been done based on an optimization-based simulation-evaluation cell synthesis retargeted to the block level. The key idea is a hierarchical decomposition in which cell-level macro-models are used to search for an optimal block-level design, while concurrently a full transistor-level design evolves for each cell. This was made possible through a sophisticated workstation-level parallelism using a compute farm of 20 to 30 Sun UltraSparcs. In both cases, the original designer is substituted either by another designer who constructs the dependency matrix [Francken99], or by extensive computing optimization [Phelps00]. Most of the working design intentions and considerations are thus lost in subsequent designs. In addition, the above approaches treat only the sizing phase of the design, the layout is considered as a separate phase handled with dedicated tools. Also, the effect of the new process parasitics on the circuit behavior is not explicitly treated.

3.5 Conclusions

An overview of the analog and mixed-signal design process was presented together with the research work carried out to automate each of its stages. This includes mixed-signal system sizing, block sizing and cell sizing, together with the associated layout phases. Emphasis was made on the verification method after each step.

The problem of layout parasitics compensation was then studied. Parasitics over-estimation and resizing-layout generation are often used by designers. Some automatic layout tools were shown to be able to handle this problem either by classifying the nets according to their criticality or by driving the tools with parasitics constraints.

Finally, analog IP cores and technology porting of analog circuits were discussed. The large influence of technology on the performance characteristics of analog circuits hinders the use of digital scaling approaches. Some resizing is always necessary to tune the performance.

The aim of studying the design automation tools presented in this chapter was to investigate their application in an eventual design reuse methodology. The next chapter introduces the proposed design reuse methodology based on a close interaction between sizing and layout generation. The following two chapters then describes the CAD implementation of this methodology.

Chapter 4

Layout-Oriented Design Methodology

4.1 Introduction

In any attempt to reuse an analog block in a different context, either in a different system with modified performance specifications or a different fabrication process, circuit resizing is unavoidable. During circuit sizing, it is very important to account for layout parasitics as shown in sections 3.3 and 3.4. In this chapter, a layout-oriented approach is presented.

Section 4.2 introduces the design reuse approach adopted in this work and the associated design flow.

In section 4.3, a layout-oriented synthesis method is introduced. Parasitics are considered early in the design phase.

In section 4.4, the advantages of the proposed approach are discussed. Its impact on both the design process and eventual design reuse is emphasized.

In section 4.5, the use of hierarchy in the proposed methodology is analyzed.

4.2 Analog Design Reuse

While the concept of reusable IP's is inherited from the digital world, the concept of an analog IP is not really established yet and may evolve in the coming few years. Taking into account analog design particularities discussed in section 2.2, analog circuit performance depends heavily on process parameters and physical implementation. Analog design reuse requires CAD tools to perform synthesis from a given behavioral description to a sized transistor-level netlist and then generate the corresponding layout in a target fabrication process. For some basic analog functions (such as OTA's, comparators, ...), specific block generators may represent a kind of firm IP's, but for complex functional blocks (such as complete ADC's, PLL's, ...), it is difficult to synthesize a reliable circuit using such generators without a detailed knowledge of the internal architecture and major tradeoffs of the complete circuit.

Due to the huge amount of design experience that has to be transferred from one design experience to another, the most efficient way to generate reusable analog IP's is by incorporating appropriate information, concerning both circuit synthesis and layout, in the original design method. In addition, the effect of process dependent parameters such as layout parasitics must be treated explicitly.

In this work, a design method for analog design reuse, based on design plans, is presented. A global design plan includes analytical equations and functional/behavioral models for high-level synthesis. Low-level synthesis is based on CAD tools that allow the designer to capture design plans for sizing as well as for layout generation. In the sizing tool, interactive design plans are in the form of analytical equations and procedures developed for pre-defined *schematic templates*, both for performance evaluation and sizing. In the layout tool, design plans are in the form of *layout templates*, associated with each *schematic template*. Design plans and templates are developed, in a hierarchical manner, starting from leaf cells till the complete functional block. Knowledge is thus efficiently captured in a modular way that can be easily updated, if necessary. Design plans are not intended in any way to replace the designer, but as an approach to hold valuable information of successful design experiences, in order to guarantee first-pass silicon of future designs. This approach is more general, more flexible and more easily maintained than special-purpose, rigid block generators. The proposed method is founded on a close interaction between sizing and layout in order to accelerate the design cycle, improve the design quality and facilitate design reuse in different processes.

This chapter introduces the layout-oriented design method, the two following chapters present the associated CAD tools.

4.3 Layout-Oriented Design Methodology

All of the systems cited in section 3.2 consider the layout as a step which *follows* the circuit sizing process. The layout generation tool does not interact with synthesis. So the circuit sizing tool has *no* information about the parasitics that the physical implementation is going to generate during the layout phase. The problem of compensating layout parasitics is usually solved by an iterative procedure as demonstrated by the design flow shown in Fig. 4.1(a). The design process follows laborious iteration loops during which circuit sizing is followed by generating the layout, extracting the circuit netlist with layout parasitics and evaluating the effect of those parasitics. Some layout tools consider parasitics control during layout generation in order to avoid performance degradation as mentioned in section 3.3, however, usually one needs to resize the circuit in order to compensate for those parasitics. The resizing modifies the parasitics and the loop is repeated till a satisfying performance is obtained.

The design method presented in this work is based on a close interaction between circuit sizing and layout generation that are no longer considered as two separate tasks. Fig. 4.1(b) shows

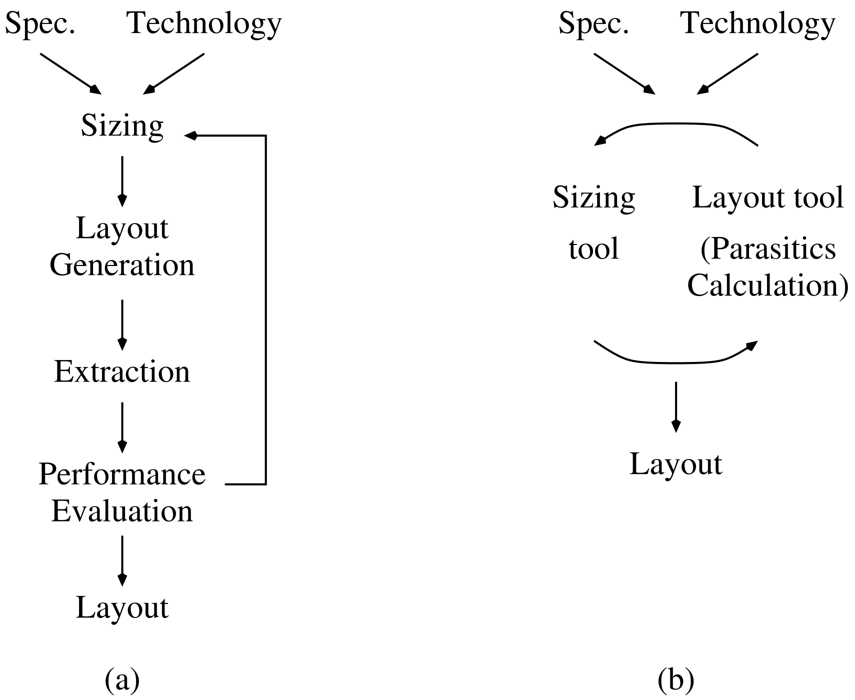


Figure 4.1: *Parasitics Compensation Methodology: (a) traditional and (b) proposed.*

the proposed layout-oriented methodology. The approach is an extension to that first presented in [Onodera90]. The layout tool is used to directly calculate parasitics related to the physical implementation during the sizing procedure. The layout generator must be fast as it is normally called several times during circuit sizing. In the same time, close placement solutions must be obtained at each iteration so as to help parasitics convergence. It is clear from the previous conditions that optimization-based layout generation approaches [Cohn91], [Malavasi96], [Lampaert95] can't be used due to their high computational cost. The knowledge-based approach is thus retained, it relies on the following *template* definitions:

- **Schematic template**: defines a fixed circuit topology and connectivity for a given function, without any information on device sizes or component values (transistor W/L, capacitance value, . . .) which are considered as design parameters.

- **Layout template**: defines both physical device *relative placement* and *relative routing* paths for a given schematic template, without any information on component sizes nor the final layout aspect ratio.

For example, Fig. 4.2 shows the *schematic template* of a folded cascode OTA containing 11 transistors. Template sizing means the determination of all transistor lengths and widths, as well as all biasing voltages, given a set of performance specifications. Fig. 4.3(a) shows the corresponding *layout template* as the relative placement of 9 cells. The actual layout of two different sizing

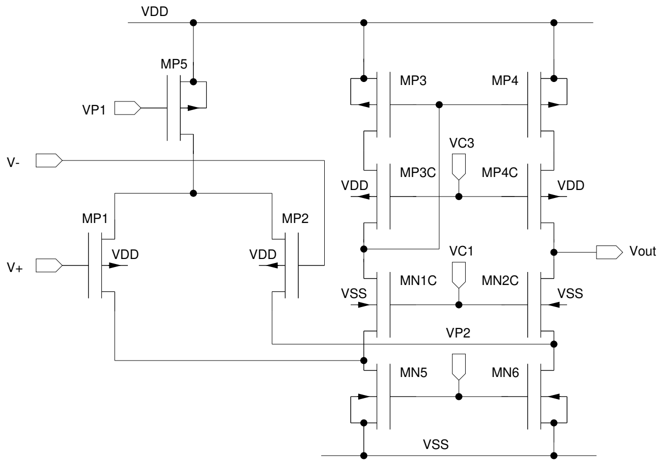


Figure 4.2: *Schematic template: Folded cascode OTA.*

	MP1/2	MP5	MN5/6	MN1C/2C	MP3/4	MP3C/4C
GBW=5 MHz	14.4	33.6	9.6	4.4	2.1	16
GBW=50 MHz	156	352	102.6	45.2	15.9	168

Table 4.1: *Transistor widths in μm for two sizings of the OTA shown in Fig. 4.2 for two different GBW's of 5 and 50 MHz, all L's are set to 1 μm.*

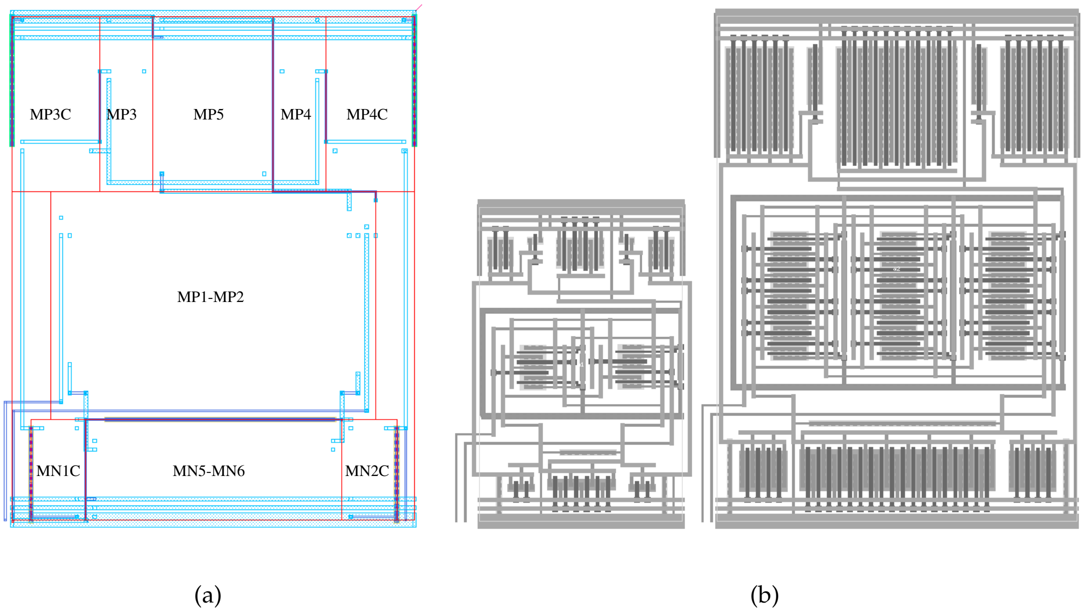


Figure 4.3: *(a) Layout template and (b) two generated layouts for the folded cascode OTA.*

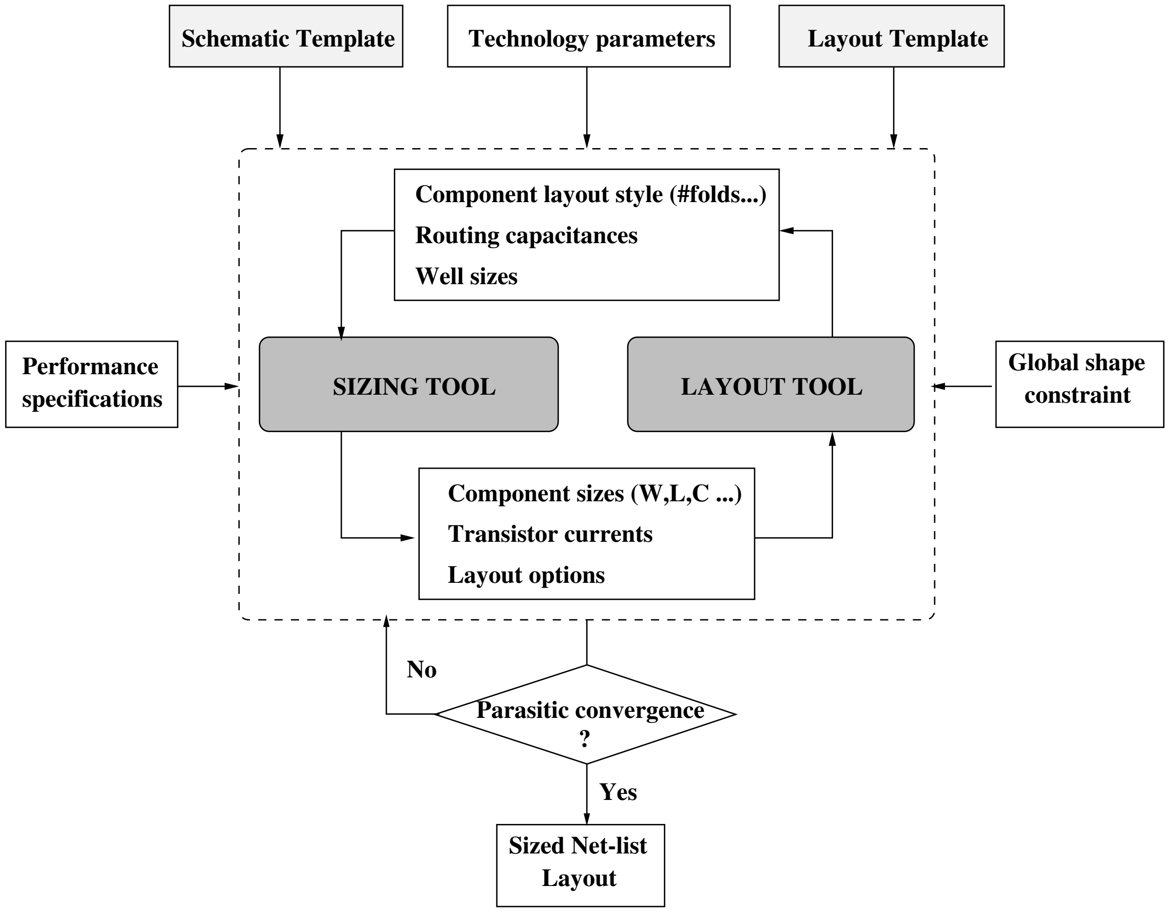


Figure 4.4: *The proposed methodology.*

examples are shown in Fig. 4.3(b) for a gain-bandwidth product frequencies of 5 and 50 MHz respectively while fixing all other specifications. Resulting transistor sizes are given in table 4.1. As shown in Fig. 4.3, in spite of the great difference between the obtained transistor sizes in both cases (for example, the differential pair W/L are $14.4/1\mu$ and $156/1\mu$ respectively), relative placement and routing are the same following the pre-defined template. This layout-oriented approach has the following property: For a given sizing, it becomes possible to make an early evaluation of the exact shape and area of each device, and consequently the length of each routing wire can be easily derived, both resulting in an accurate evaluation of the associated layout parasitics.

The method is presented in details in Fig. 4.4. Starting from the given performance specifications a first circuit sizing is performed. As an example of parasitic capacitances, consider transistor diffusion capacitances. Wide transistors are often folded to have practical aspect ratios for physical implementation, this also reduces the diffusion capacitance due to source/drain sharing, see section 5.4.1. During the initial sizing, all transistors are assumed to have single-folds. This assumption over-estimates transistor diffusion capacitances as it neglects diffusion sharing, however, it offers a good starting point as will be shown in section 6.4. After this initial sizing the following information is transfered to the layout tool:

- Calculated transistor sizes.
- Calculated transistor currents.
- Layout options for each device.
- A global shape constraint for the OTA such as the layout aspect ratio.

Based on this information coupled with the OTA pre-defined layout template, the layout tool is executed in a *parasitics calculation* mode. In this mode an area optimization step determines the shape of each device in the layout template in order to satisfy a given aspect ratio. This is followed by the calculation of routing paths and wire widths specified by the allowable current density. No actual physical layout is generated in this mode. The layout tool returns the following information to the sizing tool:

- The number of folds for each transistor and their widths, in addition to the number of source/drain diffusions which are external, internal to the transistor or shared with other transistors. This allows exact calculation of diffusion capacitances.
- Parasitic routing capacitance.
- Exact well sizes so that floating well capacitance can be calculated.

Multiple calls to the layout tool in the *parasitics calculation* mode are allowed as the iteration loop progresses. This allows the sizing tool to accurately account for parasitics (resulting from a particular physical realization) during circuit sizing. In other words, the sizing-layout iterative process is efficiently automated. When parasitics convergence is reached, the layout tool is called in a *generation* mode where the actual layout is physically generated based on the same template. It should be noted that since the *initial* sizing starts already with a rather *good* estimate of the parasitics accompanying each device (one fold per transistor, lower plate capacitance of two-plate capacitors, ...), global parasitics convergence is reached with a limited number of iterations which depends on both the circuit behavior with respect to parasitics and the operation frequency.

As stated above, the idea of parasitics compensation in the sizing loop was first presented in [Onodera90]. However, in our approach, we not only try to estimate the parasitics but we also try to optimize certain aspects of the layout before the extraction, for example the minimization of the transistor diffusion capacitance on certain nodes to enhance the frequency behavior, using different possibilities for device implementations which have different matching-parasitics compromises, layout constraints enforcement through complex module generators, considering reliability rules, and a special area optimization algorithm (see chapter 5). Also, in [Onodera90] sizing is performed on two steps: First a knowledge-based one during which no parasitics information is available, and whose main purpose is to provide a starting point to the *detailed* sizing step which follows and is based on a simulation-based optimization algorithm where the extracted parasitics are added to the simulated netlist. In the proposed method, a knowledge-based sizing approach is investigated where the extracted parasitics are injected directly in the design equations (see chapter 6). In addition, an emphasis is made on extending the approach on several levels of hierarchy.

As can be seen, the methodology depends heavily on the implementation of the layout and sizing generators. This will be the subject of chapters 5 and 6 respectively.

4.4 Advantages

Comparing figures 4.1(a) and 4.1(b), we recognize that the iterative loop for parasitics compensation is conserved. However, automating this procedure using a special parasitics mode in the layout tool offers the following advantages:

- For a given circuit, different layout styles can be investigated for the same device. Their corresponding parasitics contribution is calculated and sent back to the circuit sizing program. Their effect on the overall circuit performance can be compensated by the sizing procedure. For example, different sophisticated techniques exist for transistor matching. This may increase the parasitic capacitance. A compromise is often needed between matching and performance degradation [Malavasi95].
- Some components behavior can't be separated from their physical implementation. For example, integrated inductors, used heavily in recent RF IC's, have different possible shapes, each with a different associated equivalent electrical behavior which can be used during sizing, by the sizing procedure.
- Layout techniques that minimize parasitic capacitances can be exploited. For example, folding large transistors allows to decrease their source and drain diffusion capacitances. This can be used to optimize transistor sizes and to reduce power consumption for a given frequency and noise specifications.
- Global layout constraints such as the global aspect ratio and circuit reliability design rules can be taken into account *during* circuit sizing.
- The proposed approach guarantees that the circuit will satisfy the performance specifications in the presence of layout parasitics. The accuracy is largely dependent on the precision of parasitics calculations by the layout tool, as well as its capability to take analog layout constraints into consideration, see section 5.4.1.

This method can be applied at the cell sizing level as well as at the block sizing level in a hierarchical manner as will be explained in the following section.

4.5 Top-Down or Bottom-Up

Although it has been stated in section 2.2 that analog circuits have a loose form of hierarchy, hierarchy is still one of the most efficient ways to manage design complexity. It permits to decompose the global, complicated design task into smaller, more manageable subtasks, and allows an eventual reuse of existing knowledge for each sub-block [Gielen91]. An important decision in any hierarchical approach is whether it should start from the top-level and proceed to the lower, smaller and less complicated levels (Top-Down approach), or starts by building the leaf cells and assembles them to build larger ones till the whole system is completed (Bottom-Up). As shown in Fig. 3.1, circuit sizing often follows a top-down approach while layout generation follows a bottom-up one.

An important consideration while designing in the presence of layout parasitics is that parasitics degradations may result from several levels of hierarchy. For example, while designing an opamp in a switched-capacitor implementation, the opamp is usually considered as a separate cell. At the opamp output, in addition to the internal parasitic capacitance, there exist also those of the routing capacitance and some lower-plate capacitance of capacitors connected to the output, that all load the amplifier. If these parasitic capacitances are not taken into account during sizing large discrepancies would result between expected and measured performances. This means that some layout information are also needed from higher levels of hierarchy to be fed to lower levels.

In addition, in order to facilitate system-level floorplanning, some global shape information that controls the aspect ratio of each block must also be transmitted to lower level cells. Since analog devices are characterized by having large variations in device sizes (section 2.2), this often leads to a large degree of freedom in controlling the layout shape.

For these reasons, we have chosen the *top-down* approach for both circuit and layout synthesis. While this hierarchy is somewhat easy to understand for the sizing tool, it needs some explanations for the layout generation one. Top-down layout generation means that global area optimization could be done influencing the shape of cells in different levels of hierarchy by means of top-down shape constraint propagation (see section 5.6). However, allowing the cells to change its shape must be limited in the same time by satisfying the required design constraints and requirements imposed by the designer so as not to degrade the circuit performance. It should be noted that this overall area optimization could only be made possible using a fixed layout topology defined in the corresponding layout *template*, otherwise, the time needed for automatic layout floorplanning, device generation and optimization would become prohibitive.

4.6 Conclusions

In this chapter a layout-oriented design method intended to enhance design quality and facilitate design reuse of analog circuits was presented. This approach is based on the definition of sizing plans for predefined schematic and layout templates. By integrating both electrical and physical synthesis, the proposed method contributes to the capture of reusable designs. Since the design must be resized when reused in another fabrication process, the proposed approach also helps in process retargeting.

The methodology is used hierarchically in a top-down fashion. Shape and parasitics constraints on the layout are propagated from the highest level to smaller ones as will be shown in the following chapters.

The next two chapters present a layout *language* that allows both to capture layout templates and eventually generate the corresponding layout, and a knowledge-based sizing environment for schematic template and design plan capture. Since the main CAD contribution of this work has been in the layout generation phase, emphasis is made on layout issues.

Chapter 5

Procedural Layout with Parasitics Calculation

5.1 Introduction

In this chapter, the CAIRO language that is used to describe *layout templates* for both layout generation and parasitics calculation is introduced. This includes device generators, hierarchy, area optimization, routing, parasitics extraction, and technology independence. Algorithms to take into account several analog-specific layout constraints are presented.

In section 5.10, an example showing the use of CAIRO is given.

5.2 Overview

In section 4.3, the layout approach was chosen to be knowledge-based one. Any layout generator must satisfy the following conditions:

- It must support a fast and accurate method for parasitic calculation which is the foundation of the method.
- It must support specific analog layout constraints so as to preserve the quality of the produced layout.
- It must support different layout styles for each device.

The layout language allows the designer to easily describe both relative placement and routing, and provides a set of predefined device generators which are part of the language.

CAIRO is implemented in the form of a documented superset of C functions. The language is constructed on top of a set of pre-existing functions (**Genlib**) for procedural layout [Pétrot94]. Genlib has been successfully used in the ALLIANCE CAD System [LIP] for the development of

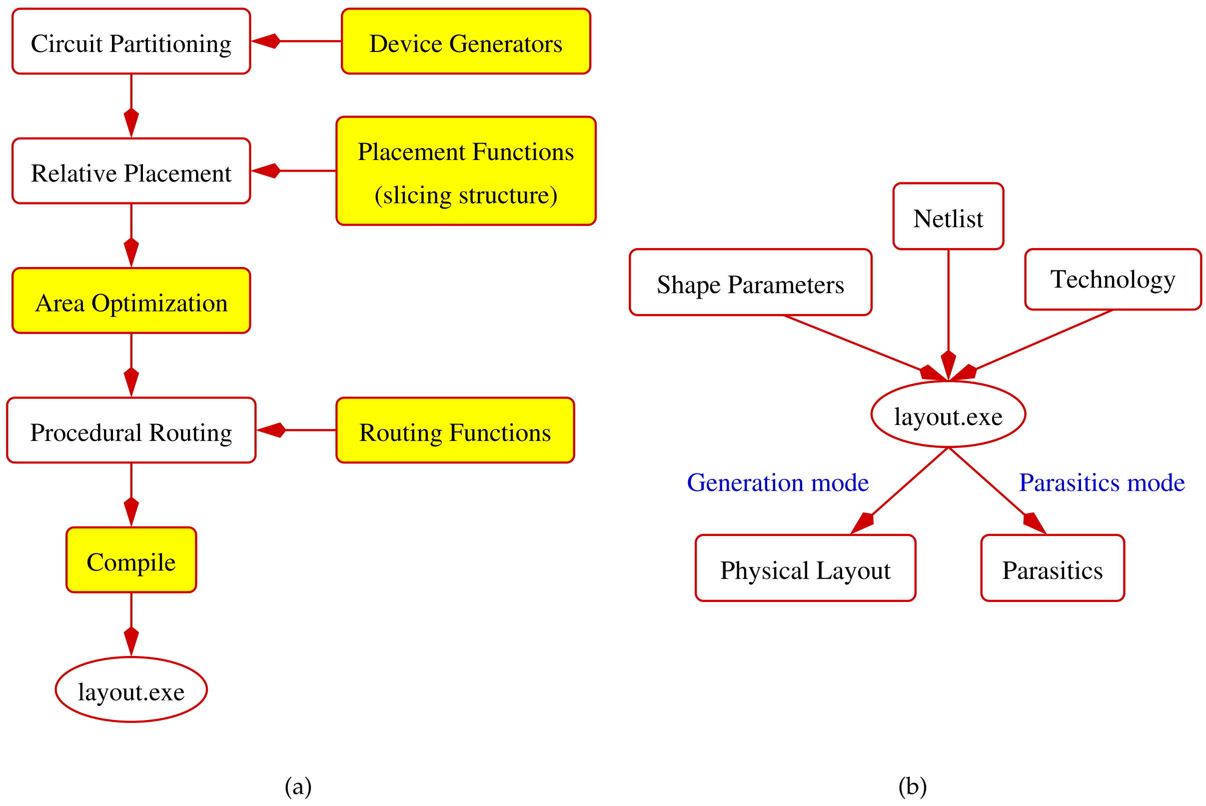


Figure 5.1: *CAIRO implementation: (a) Layout knowledge capture phase. (b) Layout generation and Parasitics calculation phase.*

parameterized digital module generators [Greiner94]. For a complete description of the CAIRO language refer to appendix A.

Shaded boxes in Fig. 5.1(a) show the main components constituting the language, namely:

- Complex device generators which include transistors, differential pairs, multi-capacitor arrays and resistors (section 5.3), that respect the corresponding analog layout constraints (section 5.4).
- Placement functions that allow to define relative placement, based on a fixed slicing structure (section 5.5).
- An original, area optimization algorithm (section 5.6).
- Routing functions that allow relative routing description using predefined reference points (section 5.7).

Circuit partitioning, *relative placement* and *procedural routing* steps represent the *knowledge capture* process. First, the circuit components are mapped to the available device generators. Using placement functions, the designer then describes relative placement of devices and modules (subcircuits). Finally, the designer describes relative routing. In other words, for each circuit the corresponding physical implementation based on an expert knowledge is stored in a specific C program.

This program is then compiled and linked to a set of static libraries containing the language functions and device generators.

It should be noted that the program is independent of both device sizes and fabrication process. Both information are provided only during the execution phase. The parameters of the program are:

- a SPICE netlist describing the actual device sizes.
- a global shape constraint.
- a technology file describing the target fabrication process.

For a given template, the layout generator can be used to either generate the layout or calculate the associated parasitics for a given fabrication process.

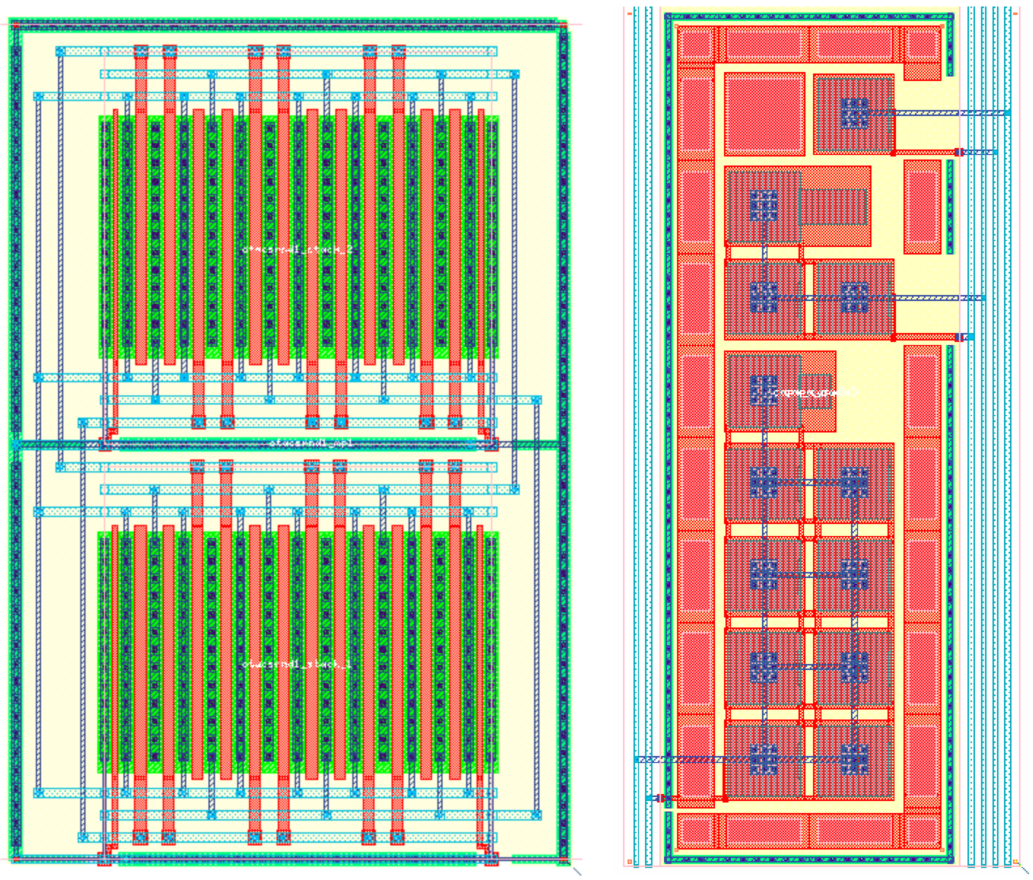
5.3 Device Generators

The method relies on a set of predefined device generators of commonly used devices, this includes:

- **MOS transistor**. There are actually several generators for different MOS transistor combinations: a simple transistor, a differential pair, a simple or a multiple transistor current mirror with different current ratios. The main parameters are the transistor length and width. Transistors are generated respecting analog layout constraints as described in section 5.4. Fig. 5.2(a) shows a layout example of a common-centroid interdigitated differential pair.

- **Capacitor**. There are actually two capacitor generators depending on contact position. Contacts could either be centered on the top plate or on the sides. Capacitor armatures could be any metal or poly levels. The main parameter is the capacitance value.

- **Capacitor array**. This parameterized generator [Chesneau98] places several capacitors with a given capacitance ratio after dividing them into *unit capacitors* in a rectangular array. The generator supports non-integer ratios while preserving a fixed perimeter-over-area ratio for all unit capacitors in order to reduce capacitance ratio errors due to edge effects. The generator also handles the placement of dummy capacitors around the array and in vacant array positions. An example of a 3-element capacitor non-integer array with dummy capacitors and well contacts is shown in Fig. 5.2(b).



(a) (b)

Figure 5.2: *Layout example of a (a) common-centroid interdigitated differential pair and a (b) 3-element capacitor array.*

- **Resistors**. There are actually two generators for simple and two-resistor interleaved array (for maximum matching) in resistive POLY. The main parameter is the resistance value.

The physical shape of a device is variable: For a given fabrication process and a given circuit sizing, the total device area is roughly constant, but the aspect ratio is highly variable (depending on the number of folds for a transistor, or the number of rows for a capacitor array). The actual shape is automatically determined by the global shape constraints and the instantiation context during area optimization. It can also be fixed by the designer.

All devices share the following characteristics (refer to Fig. 5.3):

- Each device is surrounded by two rectangular boxes: The first contains all the physical objects, it is thus called the *bounding box*. The second surrounds the first one and is used to place devices relative to each other by abutment, it is thus called the *abutment box*. The distance

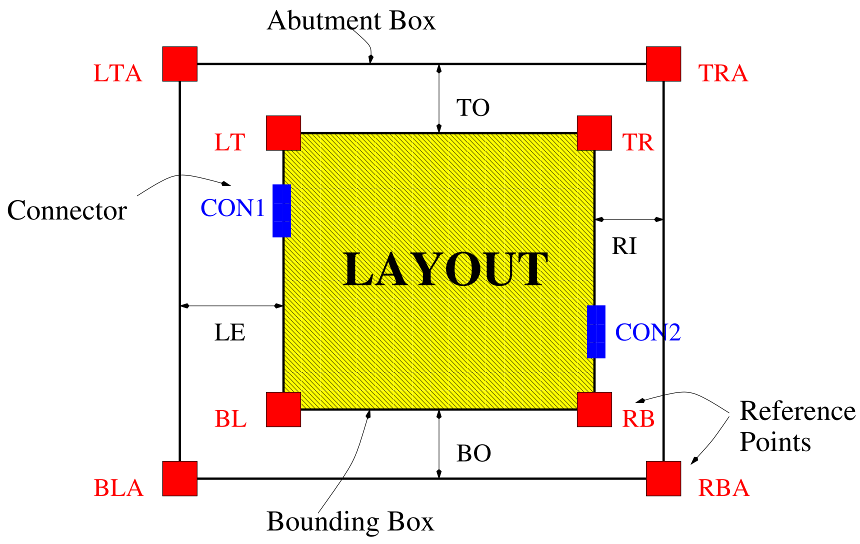


Figure 5.3: *Bloc Characteristics*.

Device Generator	Device	Status
CAIRO_TRANSISTOR	MOS transistor	Available
CAIRO_DIFFPAIR	MOS Differential pair	Available
CAIRO_BIASPAIR	MOS Simple current mirror	Available
CAIRO_CURRENT_MIRROR	MOS Multi-transistor mirror	Prototype
CAIRO_CAPACITOR	Single capacitor	Available
CAIRO_MULTIPLE_CAPACITOR	Capacitor array	Available
CAIRO_RESISTOR	Single resistance	Available
CAIRO_MULTIPLE_RESISTOR	Resistance array	Under development

Table 5.1: *CAIRO device generators*.

between the two boxes (the *surround*) can be adjusted by means of four parameters namely: *LE, TO, RI, BO*, each responsible for one side. Since devices are placed by abutment, this is used to allocate free space for routing.

- Attached to each device are eight *reference points* (see section 5.7), one on each corner of both the bounding and the abutment boxes. Those references are used to define routing.

- Each device has a set of physical *connectors* used to connect the corresponding device to other ones. More than one physical connector can be defined for each terminal. All connectors lies *on* the bounding box.

Table 5.1 shows the list of CAIRO device generators including those currently under development.

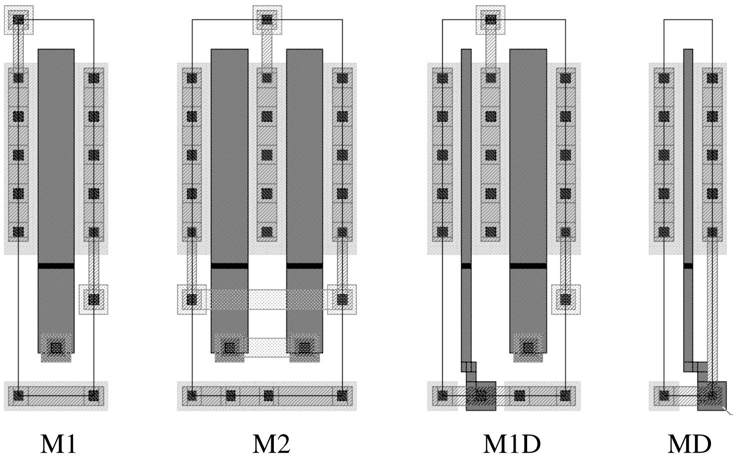


Figure 5.4: *Motifs used in building transistors.*

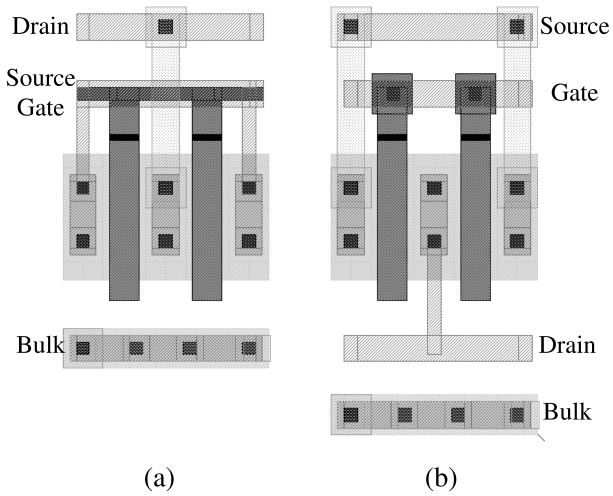


Figure 5.5: *Different transistor overlapping terminals.*

5.4 Analog Layout Constraints

In this section analog-specific layout constraints taken into account in the device generators are presented together with the algorithms developed to control them.

5.4.1 Parasitics Constraints

All transistors are built using four motif generators: A single-transistor module *M1*, a double-transistor module *M2*, a single-transistor module with a dummy one *M1D* and a dummy transistor *MD*, all shown in Fig. 5.4. The gate of the dummy transistor is connected to the bulk to keep the transistor off. Motifs are stacked or interleaved in order to create larger transistors. The generator allows the designer to control coupling parasitic capacitance between wires [Wolf99].

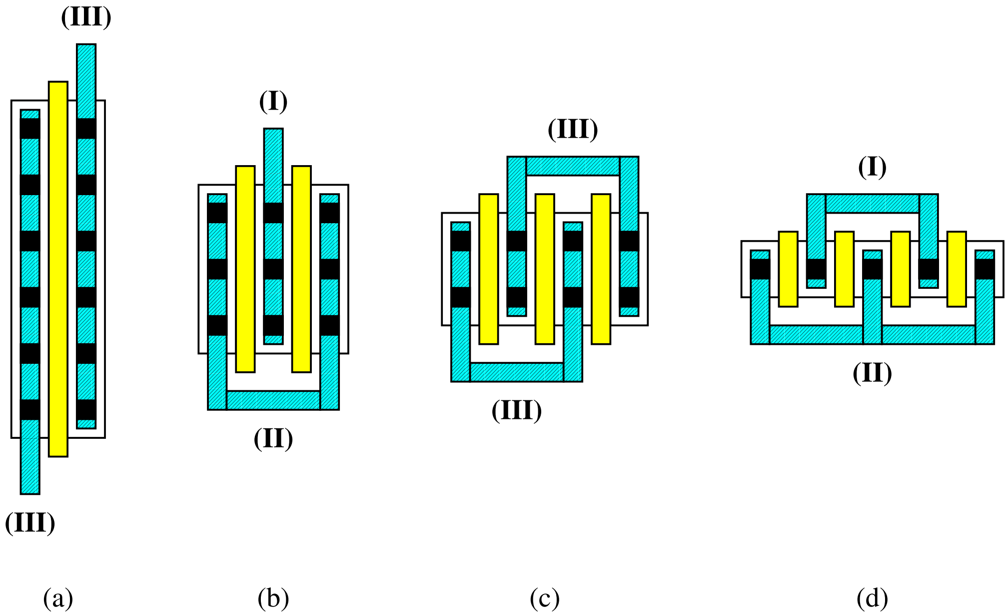


Figure 5.6: *Transistor folding: (a)* $N_f = 1$ *(b)* $N_f = 2$ *(c)* $N_f = 3$ *(d)* $N_f = 4$*. Cases (I), (II) and (III) correspond to those in equation (5.1).*

Fig. 5.5 shows two different implementations of the same transistor. In 5.5(a) the gate and source are superimposed, while the drain passes over both of them. This configuration can be used in a low frequency application. In 5.5(b) the gates are joined by the first metal layer close to the active region to reduce RC effects, the source passes over it using the second metal layer and the drain is separated downwards. This module is best suited to high frequency applications. Both of them are generated using the *same* module generator with different parameters.

Very wide transistors can be generated on multiple stacks. This allows to insert more bulk contacts, as shown in Fig. 5.2(a), in order to avoid latch-up and to reduce substrate coupling noise. However, increasing the number of stacks also increases the routing capacitance. The number of stacks is also a parameter of the transistor device generator.

Transistor folding reduces the diffusion-bulk parasitic capacitance (drain-bulk and source-bulk capacitances). This is due to the sharing of these diffusion areas between folds as shown in Fig. 5.6. The total effective diffusion width W_{eff} applied to calculate the diffusion capacitance is usually a fraction $\mathfrak{F}$ of the transistor width W, i.e. $W_{eff} = \mathfrak{F}.W$, where $\mathfrak{F}$ is the capacitance *reduction factor* due to folding. In case of a non-folded transistor $\mathfrak{F} = 1$. While for a folded one, $\mathfrak{F}$ depends on the number of folds N_f and the position of the diffusion (for alternate source/drain diffusions) as

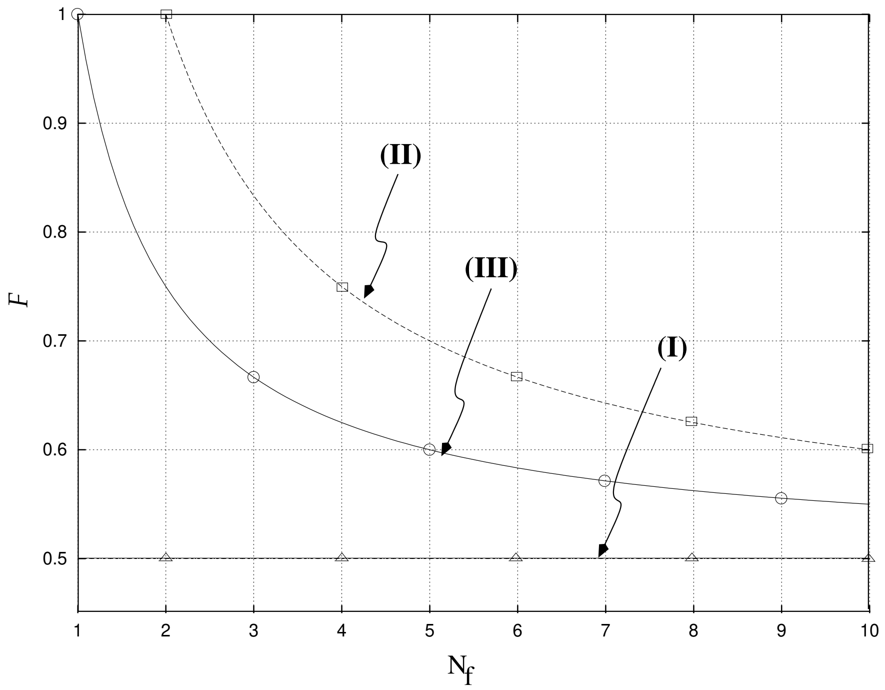


Figure 5.7: *Equation (5.1): Diffusion capacitance reduction factor $\mathfrak{F}$ with the number of transistor folds.*

follows:

$$\mathfrak{F} = \begin{cases} \frac{1}{2} & \text{if } N_f \text{ even \& internal diffusion} \quad \text{(I)} \\ \frac{N_f+2}{2N_f} & \text{if } N_f \text{ even \& external diffusion} \quad \text{(II)} \\ \frac{N_f+1}{2N_f} & \text{if } N_f \text{ odd} \quad \text{(III)} \end{cases} \tag{5.1}$$

An example for each case is shown in Fig. 5.6 for N_f equal to 1, 2, 3 and 4 respectively. As shown in Fig. 5.7, the reduction factor $\mathfrak{F}$ decreases significantly for the first few folds for cases (II) and (III). It is clear that this parasitic capacitance is minimized in case (I).

5.4.2 Matching Constraints

Special layout styles are used in order to minimize device mismatch based on considerations of process gradients, temperature gradients, anisotropic and boundary effects. Interleaving and common centroid configurations are shown to be effective in reducing the mismatch due to linearly varying parameters across the chip surface [Bastos96]. Combined with parasitics constraints (both diffusion and routing), several configurations of critical transistors in the circuit could be investigated and a good compromise between matching and parasitics effects could be found [Naiiknaware99].

The mismatch between transistors is also dependent on their relative channel orientation. Consider two MOS transistors M_i and M_j, respectively split into n_i and n_j identical folds, all in the same stack and carrying the same nominal current I. The *current mismatch* F_{ij} between transistors M_i and M_j is given by [Malavasi95]

$$F_{ij} \triangleq \frac{\epsilon_I}{I} \left| \frac{\Delta n_i}{n_i} - \frac{\Delta n_j}{n_j} \right| \tag{5.2}$$

where ϵ_I is the maximum error of the difference between currents flowing through channels with opposite orientations, $\Delta n_i (\Delta n_j)$ is the difference between the number of motifs oriented in opposite directions of transistor $M_i (M_j)$. For N transistors in the same stack, the *current mismatch* F_N is defined as

$$F_N = \sum_{i=1}^{N} \sum_{j=i+1}^{N} F_{ij} \tag{5.3}$$

Current mirrors are a special case where tight matching between transistors is usually critical to the circuit operation. An algorithm dedicated to the physical layout of current mirrors has been developed. It takes into account channel orientation and guarantees maximum interleaving between transistors all centered around the stack mid-point (common-centroid).
Given the current ratio of a mirror with N transistors, the corresponding stack is generated on two steps:

1. Assign for each transistor the appropriate motifs that minimize F_N given by equation (5.3). The algorithm is based on the *M2* and the *M1D* motifs shown in Fig. 5.4. Since all transistors in the current mirror have a common source, by assigning the source to the external diffusion and the drain to the internal one, these two motifs (*M2* and *M1D*) can be freely interleaved by sharing the external source diffusion area. Each transistor M_i is thus composed of nm_{1i} motifs of type *M1D* and nm_{2i} motifs of type *M2*. The total number of motifs in each transistor is

$$nm_{ti} = nm_{1i} + nm_{2i} \tag{5.4}$$

and the total number of transistor modules is

$$n_i = nm_{1i} + 2.nm_{2i} \tag{5.5}$$

Since the *M2* motif has two transistor modules with opposite channel orientations while *M1D* has only one module oriented to the right, then by definition

$$\Delta n_i = nm_{1i} \tag{5.6}$$

Motif assignment is done by an exhaustive trial of all possible motif combinations and selecting the one with minimum F_N. A trivial solution that leads to $F_N = 0$ is to take all motifs of the type *M1D*. In this case $\Delta n_i = n_i$, i.e. all transistors have the same channel orientation

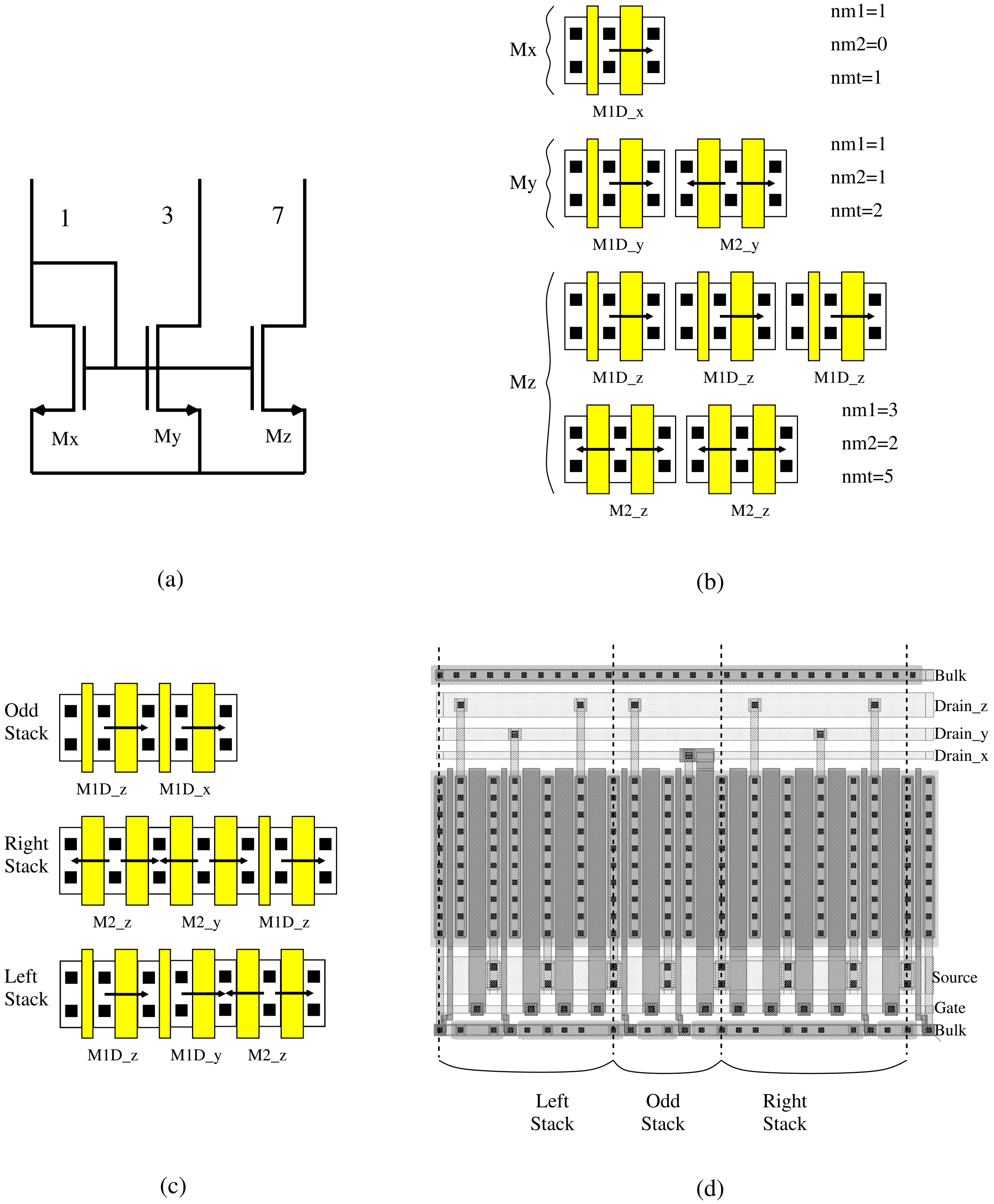


Figure 5.8: *Current mirror, (a) schematic, (b) transistor motifs, (c) elementary stacks, and (d) final layout.*

with dummy transistors inserted in between. This solution however increases the distance between transistors which is another important mismatch factor. It also leads to an overall excessive area. Thus solutions of more than one transistor with *all* channel orientations in one direction are rejected.

2. Interleave the motifs of all transistors such that each group of motifs belonging to a given transistor is centered around the middle of the stack. In order to achieve this, three elementary stacks are constructed: An *odd* stack containing one motif from each transistor with an odd number of motifs nm_{ti}, and two symmetrical stacks, a *left* and a *right* stack, constructed at the same time by placing one motif from each transistor alternatively between both stacks till all motifs are exhausted. This ensures maximum interleaving between transistors. The required current mirror stack is then composed of the *odd* stack placed at the middle, and the other two stacks abutted one at each side. This places the centroid of all transistors near the middle of the final stack.

As an example, consider a current mirror composed of three transistors $Mx : My : Mz = 1 : 3 : 7$ shown in Fig. 5.8(a). Applying the previous algorithm, the following motifs are found: $nm_{1x}/n_x : nm_{1y}/n_y : nm_{1z}/n_z = \Delta n_x/n_x : \Delta n_y/n_y : \Delta n_z/n_z = 1/1 : 1/3 : 3/7$ which minimizes equation (5.3). The assigned motifs of each transistor are shown in Fig. 5.8(b). Arrows show the direction of current flow. Since the number of motifs of Mx and Mz (nm_{tx} and nm_{tz}) are both odd, one motif from each is placed in the *odd stack*. The other two stacks are then composed by taking one motif alternatively from each transistor as shown in Fig. 5.8(c). Fig. 5.8(d) shows the physical layout of the current mirror stack after abutting the three elementary stacks shown in Fig. 5.8(c). Dummy transistors are then added on both sides if not present.

5.4.3 Reliability Constraints

Reliability design rules are important for the long-term functionality of the circuit. DC current information is used to adjust wire widths inside each device as well as routing wires between devices in order to respect the maximum current density allowed by the technology. This prevents electromigration from taking place which may lead to open circuits in wires subjected to high current densities [Wolf99]. The number of contacts are also increased for wide wires in order to decrease their resistance according to reliability design rules. This is clearly shown in the current mirror shown in Fig. 5.8(d) where wire widths and contact numbers have been adjusted separately for each transistor assuming high current densities. The widest wire is that of the *source* where the sum of all transistor currents flows.

Parameter	Description
Name	Transistor instance name
Type	Transistor type
W	Gate width
L	Gate length
Bulk	Bulk connection
I*	Drain current
M*	Number of transistor fingers
STACKS*	Number of transistor stacks
DUMMY*	Dummy transistor placement
DIFF_CAP*	Diffusion capacitance minimization
GATE_Y*, DRAIN_Y*, SOURCE_Y*, BULK_Y*	Vertical order of terminal routing
GATE_X*, DRAIN_X*, SOURCE_X*, BULK_X*	Horizontal order of terminal routing
GATE_Wx*, DRAIN_Wx*, SOURCE_Wx*, BULK_Wx*	Width of different routing wires (x)
GATE_TYPE_x*, DRAIN_TYPE_x*, SOURCE_TYPE_x*, BULK_TYPE_x*	Layer of different routing wires (x)

* Optional parameter.

Table 5.2: *Transistor device generator parameters*.

All the above layout constraints are taken into account inside the corresponding device generators. As an example, table 5.2 describes the different parameters of the single transistor device generator. All optional parameters have default values except the number of fingers (M) which is determined by the area optimization algorithm (see section 5.6). The optional current parameter (I) is used to automatically adjust wire widths according to process reliability design rules. These widths can also be directly imposed by the TERMINAL_Wx parameters.

For example, transistor layouts shown in Figs. 5.5 (a) and (b) are generated using the following CAIRO statements respectively:

```
CAIRO_TRANSISTOR("MN1", NTRANS, 8.0, 1.0, B_O, DRAIN_TYPE_V1, ALU2,
SOURCE_Y, 1, GATE_Y, 1, DRAIN_Y, 2, BULK_Y, -1, C_END);
```

```
CAIRO_TRANSISTOR("MN2", NTRANS, 8.0, 1.0, B_O, SOURCE_TYPE_V1, ALU2,
GATE_TYPE_H, ALU1, SOURCE_Y, 2, GATE_Y, 1, DRAIN_Y, -1, BULK_Y, -2,
C_END);
```

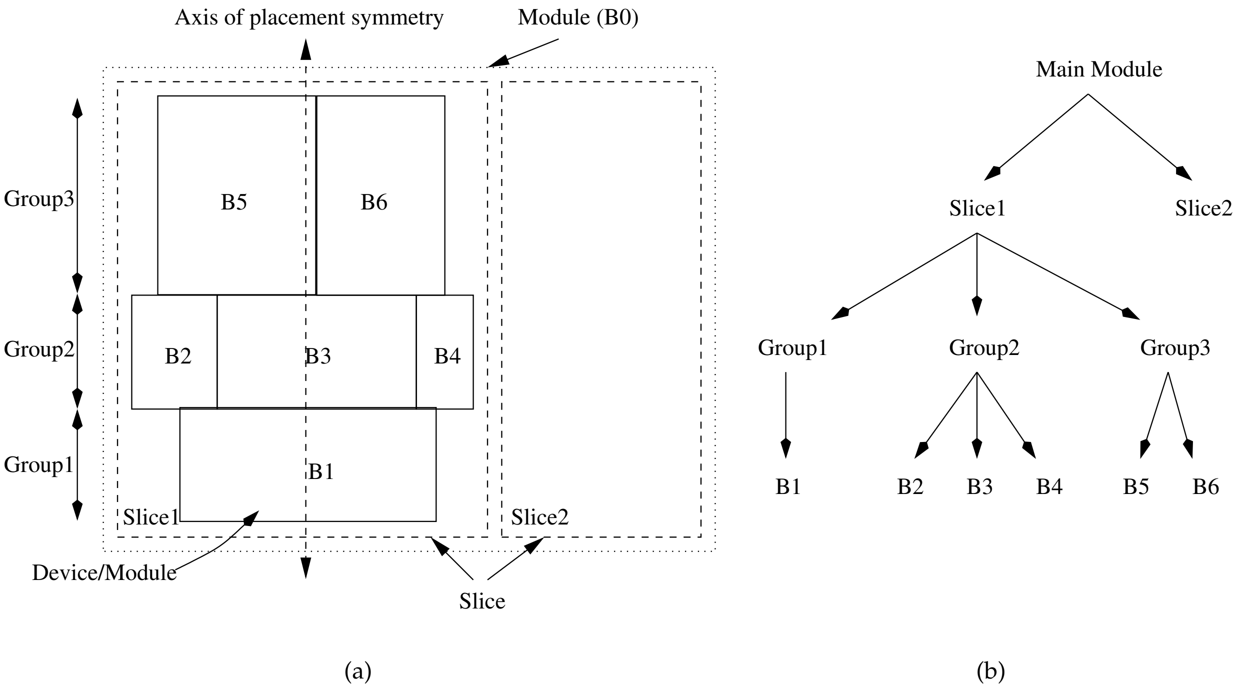


Figure 5.9: *(a) CAIRO predefined hierarchy. (b) the corresponding slicing tree.*

5.5 Hierarchical Placement

CAIRO supports a hierarchical placement approach based on *slicing trees* [Conway92].

The predefined hierarchy of a *module* is shown in Fig. 5.9. The basic elements of a *module* are:

The Device: This is the leaf cell of the tree. It is one of the built-in parameterized device generators (section 5.3). It can also be another module. A previously user-defined module is treated like a built-in device. This means that the placement is completely hierarchical.

The Group: This is composed of a horizontal arrangement (physical row) of devices and/or modules, placed besides each other in a specific order.

The Slice: This is composed of a vertical arrangement (physical column) of groups, placed on top of each other in a specific order. After layout generation, each slice preserves a vertical axis of symmetry passing by its center.

The Module: This is composed of a horizontal arrangement (physical row) of slices, placed besides each other in a specific order. A module is considered as a building block (a sub-circuit) that can be used to construct other modules, till the complete layout (*main module*) is described.

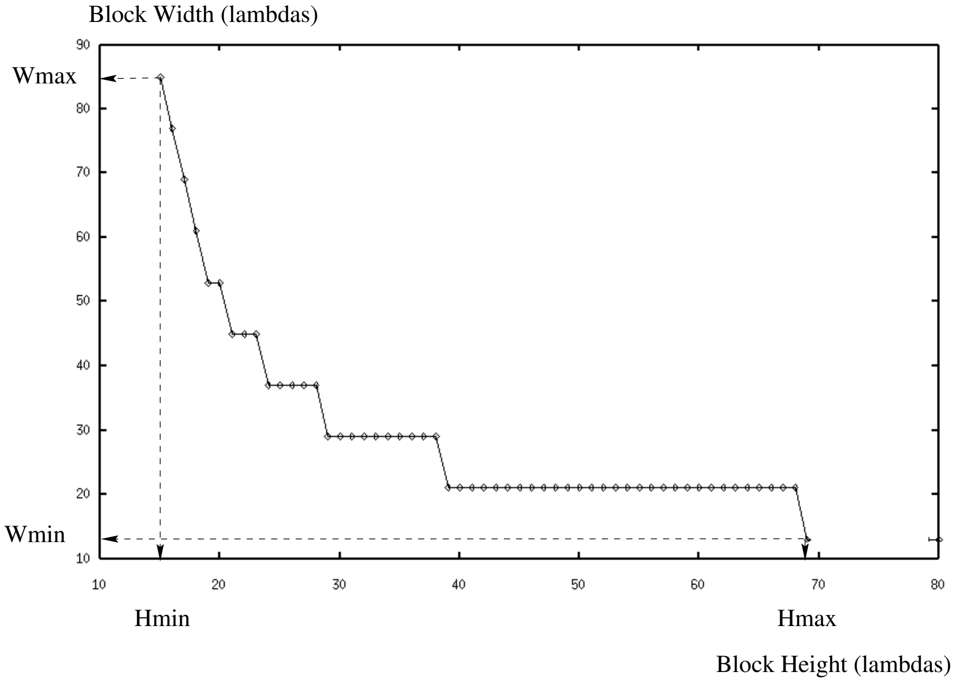


Figure 5.10: *Folded MOS transistor Shape function*.

5.6 Area Optimization

The layout is usually driven by a global shape constraint (a given height or aspect ratio). Given this constraint, area optimization is performed using an efficient hierarchical top-down algorithm based on *shape functions* and *slicing structures* [Conway92].

5.6.1 Shape Functions

As described in section 5.3, for a given fabrication process and a given electrical sizing, there are several possible shapes for a device. Shape functions associated with each device generator calculate the overall dimensions of alternative shapes for the corresponding built-in device. For a given device height, the function returns the corresponding device width. Since the $height \times width$ product is roughly constant for a given device size, this function is a discrete monotonic decreasing one. An example of this function is illustrated in Fig. 5.10 for the case of a folded MOS transistor. Each step in the shape function corresponds to a change in the number of transistor folds. The minimum and maximum values allowed for the independent side, h_{max} and h_{min}, are also calculated to avoid infeasible device implementations. The core of the algorithm consists of calculating the shape function of a given *module* starting from its child devices [Koh90] through shape function propagation as described in the following sections.

The effect of analog layout constraints, discussed in section 5.4, on the area optimization problem is handled inside device generators and the accompanying shape functions. This allows a complete separation between the optimization algorithm and device specific layout generators.

5.6.2 Slice Area Optimization

Problem Formulation: The hierarchy, described in section 5.5 has been chosen to facilitate the process of area optimization. For a slice of n groups, let HS be the given slice height, WS the corresponding calculated slice width, $\underline{h_g}$ be the set of group heights and $\underline{w_g}$ the set of the corresponding calculated widths, then the problem of slice area optimization can be formulated as follows:

$$\text{given} \quad HS\text{, and}$$

$$w_{gi} = f_{gi}(h_{gi}) \qquad \text{for } i = 1..n \tag{5.7}$$

$$\underset{\underline{h_g}}{\text{minimize}} \quad WS = max(\underline{w_g}) \tag{5.8}$$

$$\text{subjected to} \quad \begin{cases} \sum_i h_{gi} \leq HS \\ h_{gimin} \leq h_{gi} \leq h_{gimax} \\ \text{Layout constraints (section 5.4)} \end{cases} \tag{5.9}$$

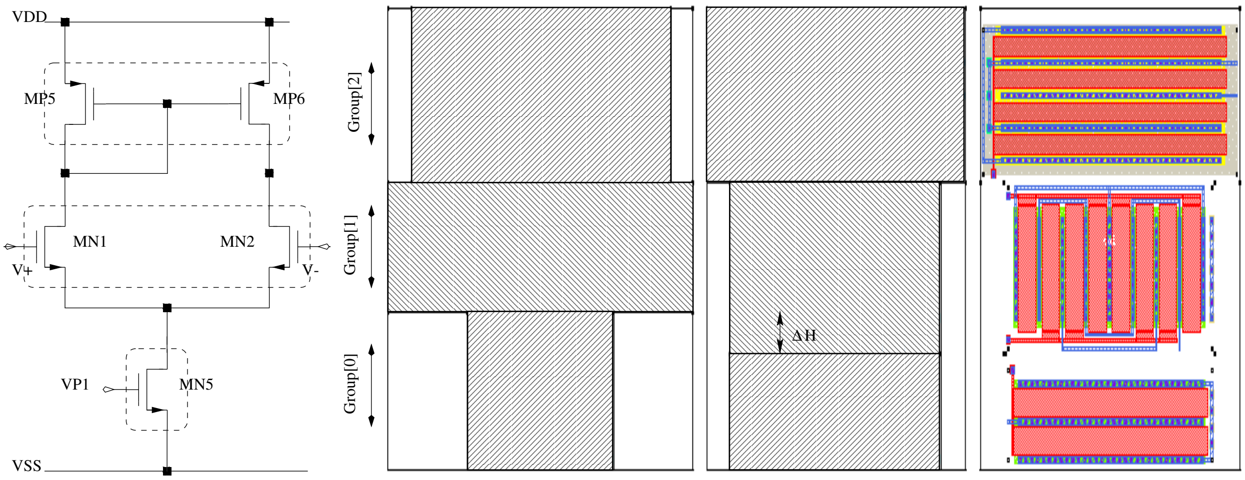
where f_{gi} is the group i shape function calculated using the shape functions of the constituting devices as follows

$$f_g(h_g) = \sum_j w_{dj} = \sum_{j=1}^{j=k} f_{dj}(h_g) \tag{5.10}$$

where k is the number of horizontal devices each of width w_{dj} constituting the group and f_{dj} is the device shape function described in the previous section. Since the slice height HS is given as a global shape constraint, area minimization reduces to the minimization of the slice width WS as given by equation (5.8). Equation (5.9) represents the optimization constraints.

The Algorithm: The proposed algorithm is illustrated by the example shown in Fig. 5.11. Fig. 5.11(a) shows the schematic of a simple OTA, and its corresponding hierarchy. The selected hierarchy is composed of three groups and only one slice. Optimization starts with a given desired slice height and proceeds in two phases:

1. **Area Estimation Phase**: An initial estimate of group heights $\underline{h}$ is calculated such that the given slice height HS is divided between the groups in proportion to their *estimated* surface areas. The area of each device is estimated by calling the corresponding device generator



(a) (b) (c) (d)

Figure 5.11: *Area optimization steps:(a) schema, (b) estimation, (c) calculation and (d) generation.*

OPTIMIZE_SLICE(HS)
Phase 1 (Estimation phase):
    FIND the initial set of group heights h_{gi};
Phase 2 (Optimization phase):
    DO {
        FIND the widest group j ($w_{gj} = WS$);
        FIND ΔH such that
            when $h_{gj} = h_{gj} + \Delta H$
                $w_{gj} = f_{gj}(h_{gj}) < WS$;
        /* Try to compensate ΔH by the other groups */
        FOR each group $i \neq j$
            WHILE ($\Delta H > 0$)
            DO {
            $h_{gi} = h_{gi} - \Delta h_{gi}$ such that $w_{gi} = f_{gi}(h_{gi}) < WS$;
            $\Delta H = \Delta H - \Delta h_{gi}$;}
        IF ($\Delta H <= 0$)
        /* ΔH is compensated by the other groups */
        THEN
            Conserve the new set of heights;
        ELSE
            Exit;
    };

Figure 5.12: *Slice width optimization algorithm.*

in an *estimation* mode starting only from electrical information, e.g. the W and L of a transistor. The corresponding device generators are then used in a *calculation* mode to calculate the device width starting from its height using its shape function. This step is shown in Fig. 5.11(b). It is clear that the slice width WS is determined by the width of the widest group (i.e. group[1]) as shown in equation (5.8), while the slice height is given by summing the heights of all groups.

2. **Optimization phase**: In order to decrease WS, the height of the widest group, h_1 in Fig. 5.11(b), must be increased by a certain amount ΔH according to the corresponding shape function f_{g1}. This ΔH must then be subtracted from the heights of the other groups $h_{g0} + h_{g2}$ in order to keep the total slice height $HS \leq H$. This is shown in Fig. 5.11(c), where the height of group[1] has been increased by ΔH and the height of group[0] was reduced to compensate for this ΔH. The width of group[2] then becomes the new slice width WS. This process is then repeated till the smallest width is reached.

This algorithm is summarized in Fig. 5.12. Fig. 5.11(d) shows the final slice layout.

5.6.3 Module Area Optimization

Consider the module hierarchy shown in Fig. 5.13, the main module `Module1` is composed of n slices (`Slice11` to `Slice1n`). Given a global module heigth HM, a module area optimization function (`OPTIMIZE_MODULE(`HM`)`) is executed, which in turn calls the slice area optimization function (`OPTIMIZE_SLICE(`HM`)`) once per slice. The area of each slice is, thus, optimized separately according to the algorithm described in the previous section, such that after optimization, all slices would have the same height HM. The `OPTIMIZE_MODULE()` function returns the total module width calculated by adding the widths of all slices.

5.6.4 Multilevel Hierarchical Top-Down Area Optimization

In Fig. 5.13, `Module1` instanciates besides built-in devices (`B11, ...`), another user-defined module `Module2`. In order that `Module2` behaves as a built-in device, it must be able to supply the same information to the optimization algorithm as a built-in device. As described in section 5.6.2, each device can be called in two modes, namely the estimation mode and the calculation mode. In the estimation mode, an estimation of the module total surface area is calculated by calling its child devices in the same mode and simply adding all surface areas. In the calculation mode, however, the exact module width is required for a given height. The module area optimization function (`OPTIMIZE_MODULE()`) is thus called for `Module2`. Module width is calculated by adding the widths of all constituting slices. Fig. 5.13 shows that `Module2` contains in turn another user-defined module `Module3`. `Module3` is handled like its parent `Module2`. The `OPTIMIZE_MODULE()` function is thus used in a recursive manner. This corresponds to dynamically constructing the shape function for all user-defined modules. In other words, equation (5.10)

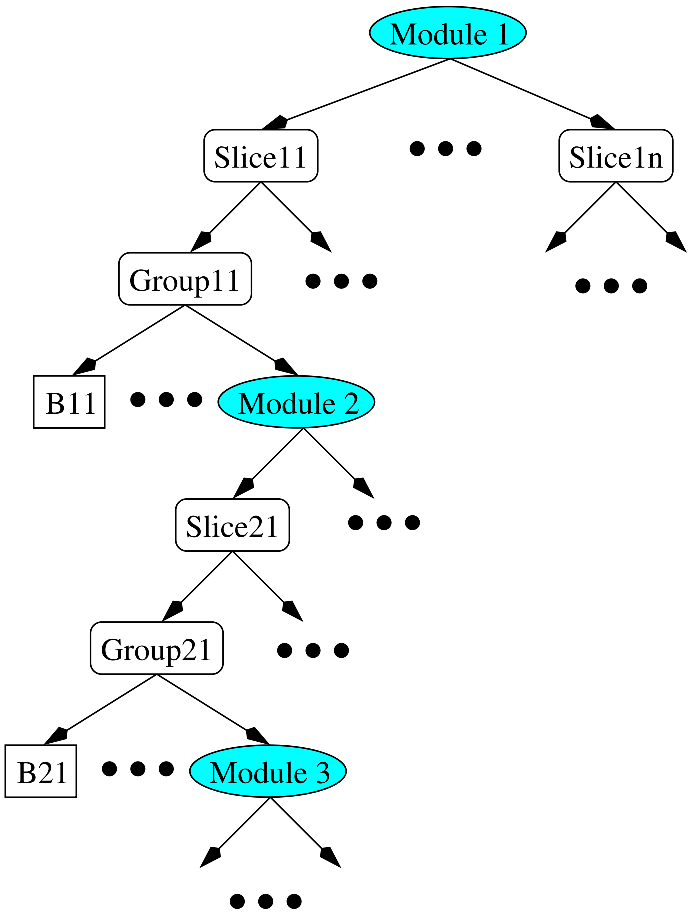


Figure 5.13: *Multilevel hierarchical optimization*.

used to calculate the group shape function f_g (refer to Fig. 5.12) becomes

$$f_g(h_g) = \sum_{j=1}^{j=k} \begin{cases} f_{dj}(h_g) & \text{if built-in device} \\ \texttt{OPTIMIZE\_MODULE}(h_g) & \text{if user-defined module} \end{cases} \tag{5.11}$$

5.7 Routing

Routing is done explicitly by the designer, i.e. the designer has to describe each physical wire, using the language primitives.

In order to support variations in the module shape due to variable device sizes, CAIRO makes use of *reference points* attached to each device (see Fig. 5.3), that allow the designer to symbolically describe routing wires. The multi-segment routing functions do not depend on the absolute coordinates of terminals, all coordinate values are automatically retrieved from terminal names and reference points in the instantiated devices. This allows routing flexibility with respect to different shapes of the same layout template.

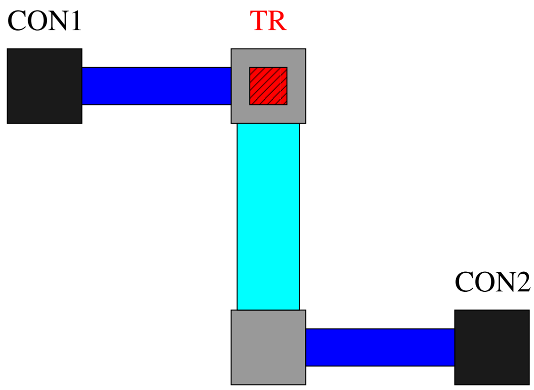


Figure 5.14: *A three-segment routing wire.*

As an example, the routing wire shown in Fig. 5.14 is generated using the following CAIRO statement:

```
CAIRO_WIRE3(ALU1, ALU2, ALU1, SW_ALU1, SW_ALU2, SW_ALU1, "MN1", "CON1",
0, "MN2", "CON2", 1, CAIRO_GET_X("MN1",TR), HOR);
```

It connects the two connectors *CON1.0* and *CON2.1*, belonging to the instances *MN1* and *MN2* respectively with three segments; the first is in *ALU1* of width *SW_ALU1*, the second is in *ALU2* of width *SW_ALU2*, and the third is in *ALU1* of width *SW_ALU1*. The first wire is horizontal (the *HOR* parameter). Note the use of the function *CAIRO_GET_X()* for capturing the x-coordinate of the reference point *TR* to assure relative routing. For a description of routing functions and reference points, refer to section A.5.

Routing is the most time consuming task while describing a given module. Several possibilities are now under investigation in order to improve the efficiency of this step.

5.8 Parasitics Extraction

In the *parasitics calculation* mode, parasitic capacitances are calculated. This takes place on two steps.

5.8.1 Device parasitics

After the determination of the device shape in the area optimization step, each device generator calculates directly the values of the associated parasitic components in a predefined attached parasitics model. Since all rectangles are generated procedurally by the *device generator*, their shape, position and exact dimensions are well-determined.
There are two types of layout parasitics, namely geometry dependent parasitics and voltage dependent ones.

- **Geometry dependent parasitics**. Their values depend only on the geometrical characteristics. For example, in the case of a simple transistor, this includes device wiring capacitance with respect to the substrate: C_{gbw}, C_{sbw} and C_{dbw}, and wire coupling capacitance: C_{gsw}, C_{gdw} and C_{sdw}. The capacitance between two overlapping layers i and j is calculated using

$$C_{ij} = C_A \times A_{ij} + C_P \times P_{ij} \tag{5.12}$$

where C_A and C_P are two technology dependent parameters denoting the capacitance per unit area and length respectively, A_{ij} and P_{ij} are the overlapping area and perimeter respectively. This equation accounts for both area and fringe capacitances. More accurate models which also account for the lateral capacitance between conductors on the same layer can be found in [Choudhury91] [Arora96] [Sakurai83], however, they need additional technology characterization steps. As we only need a rough estimate, we decided to keep it simple since the parameters used in equation (5.12) can be found in any process documentation.

- **Voltage dependent parasitics**. Their values depend on both the geometrical structure and the voltage. An example is the transistor source-bulk and drain-bulk junction capacitances which in case of the source is given by

$$C_{sb} = \frac{A_S C_j}{\left(1 + \frac{|V_{BS}|}{\phi_j}\right)^{m_j}} + \frac{P_S C_{jsw}}{\left(1 + \frac{|V_{BS}|}{\phi_j}\right)^{m_{jsw}}} \tag{5.13}$$

where, C_j and C_{jsw} are two technology dependent parameters denoting the bottom junction capacitance per unit area and sidewall junction capacitance per unit length respectively at $V_{BS} = 0$, A_S and P_S are the source diffusion area and perimeter respectively, ϕ_j is the built-in junction potential, m_j and m_{jsw} are technology parameters depending on the doping profile of the diffusion junction. The bias information (V_{BS}) is only available during sizing. Thus in our approach, the layout generator supplies the geometrical information to the circuit sizing tool which in turn uses the same electrical models implemented in circuit simulators to calculate voltage-dependent parasitics.

5.8.2 Routing capacitance

For each layout template, routing is determined by the template designer and it is not known a priori. However, it is explicitly described by the routing functions (see section 5.7). Only wire parasitic capacitance with respect to the substrate is calculated in this case. Each routing function calculates the value of the associated wire parasitic capacitances after the area optimization step.

Thus, in the parasitics calculation mode, all parasitics can be retrieved without any layout generation.

5.9 Fabrication Process Independence

Fabrication process independence is an important consideration in design reuse methodologies. A variation of symbolic layout on a fixed grid approach [Greiner90] is used as described in the following sections.

5.9.1 Symbolic Layout Approach

Symbolic layout has been successfully used in the ALLIANCE CAD system [Greiner92] to maintain a process independent library of digital cells. Careful examination of over twenty different processes ranging from 2 to 0.6 μm has led to the definition of a generic set of symbolic design rules. The basic idea is that while minimum widths and spacings of physical rectangles are quite different through these sample technologies, pitches (axis to axis distances) vary more homogeneously. Any set of physical design rules can then be mapped to the generic symbolic rules by using one basic parameter λ which is responsible to ensure correct spacing. The layout is described using structured objects called *symbols*, which are either defined by a single point, like contact primitives, or by two points, like segments. Symbols are placed on an isotropic grid with a spacing of 1 λ in both directions, such that all symbol axes lie on this grid. In addition, some fixed linear translation rules are used for width translation from the corresponding symbolic ones to the target process physical dimensions using technology dependent coefficients that depends on process layers [Greiner95]. Starting from the symbolic layout, a fully automatic symbolic-to-real translation tool (**s2r**) is responsible for the conversion towards the target process. The value of λ is chosen to respect the most critical spacing in the design rules. This is usually determined by the spacings between metal layers. As a result, respecting the generic design rules in the symbolic layout ensures an error-free physical layout after translation to any target process. This is achieved at the expense of some area overhead estimated between 10 to 15 percent larger of a corresponding layout performed directly without passing by the symbolic phase [Pétrot94].

5.9.2 Symbolic Layout for Analog Circuits

Area overhead is not a critical issue in analog design, since the most important is to satisfy performance constraints. Analog circuit performance depends directly on the sizes of all devices in the layout. Thus sizes could not be limited to multiples of λ. In addition, in some cases, rectangles could not also be kept on the symbolic grid without a significant increase of the parasitic capacitances. As an example, Fig. 5.15 shows the symbolic layout of a transistor gate besides a diffusion contact, both placed with their axes *on* the symbolic grid, shown by the vertical dashed lines, and respecting the minimum symbolic rule distance *Lmin*, which is 1.5λ. Three different cases for different transistor gate lengths are shown. It is clear that depending on the transistor gate length, there is an area overhead due to the symbolic grid placement constraint.

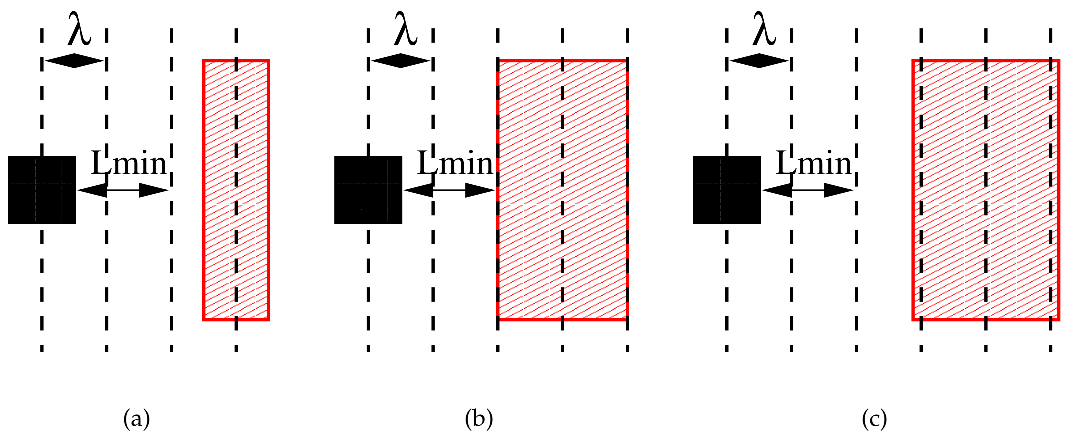


Figure 5.15: *Contact-Gate distance overhead due to symbolic layout on a fixed grid: (a) Symbolic gate length=1λ. (b) Symbolic gate length=2λ. (c) Symbolic gate length=2.2λ.*

Therefore, the symbolic approach is limited to device placement and inter-device routing. Inside device generators, all segments are drawn directly respecting the minimum target process design rules. However, the device abutment box (used for relative device placement) and external connectors respect the symbolic grid placement. Moreover, all routing wires also lie on the symbolic grid. This ensures compatibility between analog cells and digital ones using the same symbolic approach.

Design rules are transparent to the designer and are substituted by symbols in the language syntax (see appendix A). CAIRO layout templates are thus independent of the fabrication process.

5.9.3 Limitations

It is sometimes impossible to migrate the layout between different fabrication processes without modifying the template due to the following reasons:

- Some special devices such as resistors and capacitors are implemented using specific layers: For example, the layers constituting a parallel plate capacitor could either be two poly layers (the standard poly layer in addition to another special one for capacitors), standard metal-poly layers, two metal-layers, or metal sandwich capacitor using multiple layers.

- The number of available metal layers used for interconnection can also differ across fabrication processes. This limitation, however, can be avoided by proper template parameterization. Routing layers can be kept as external parameters that are determined when the layout is generated. Via handling is done automatically by the routing functions during physical layout generation.

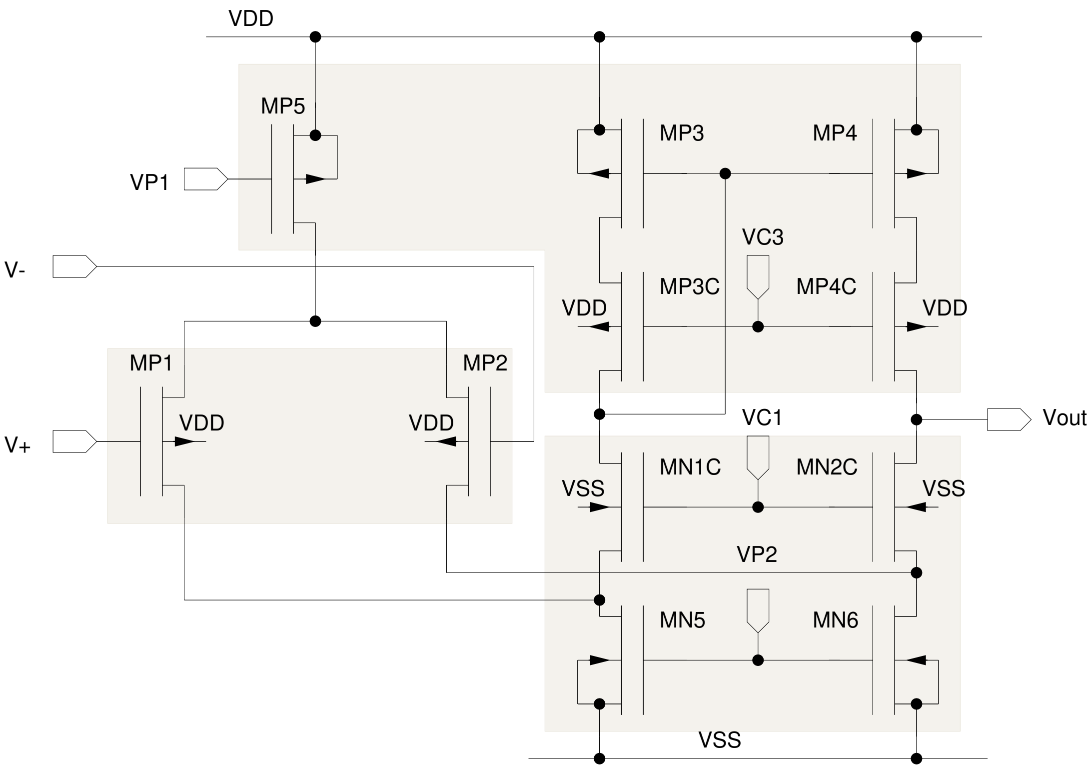


Figure 5.16: *Folded Cascode OTA.*

5.10 Example

As an example consider the folded cascode OTA shown in Fig. 5.16. The three dark areas correspond to three horizontal *groups* chosen for the corresponding slicing structure. Fig. 5.17 shows the main sections of the corresponding layout generator. The file starts as a normal C file with a special included file *cairo.h* and the standard *main* function definition.

In section (1) a netlist file in *spice* format is opened, using the `CAIRO_OPEN_SPICE_FILE()` function. The file name is kept as a run-time argument so that different sized netlists could be used. Device sizes and currents as well as special comments for additional device layout options (number of stacks for transistors, dummy structures, . . .) are defined by the SPICE file. This file is normally generated by the sizing tool. Module definition then starts with the `CAIRO_OPEN_MODULE()` function.

In section (2), the `CAIRO_OPTIMIZE()` function chooses the mode of operation, refer to Fig. 5.1(b), which could be either parasitics calculation or physical layout generation. The argument to the function (either TRUE for parasitics calculation, or FALSE for layout generation) is usually a command-line one. Module definition then starts with the `CAIRO_OPEN_MODULE()` function.

A device declaration section follows which defines *devices* according to the required partition-

```
#include <cairo.h>
main(argc,argv)
int argc;
char **argv;{
/***** (1) Open a SPICE file  *************************/
CAIRO_OPEN_SPICE_FILE(argv[1]);
/***** (2) Begin Module Definition  *******************/
CAIRO_OPTIMIZE(argv[3]);
CAIRO_OPEN_MODULE("OTA");
/***** (3) Device Declaration *************************/
CAIRO_DIFFPAIR_SPI("DP1",PTRANS,"MP1","MP2",B_O,
       "DUMMY","DIFF_CAP",MIN_D,C_END);
CAIRO_TRANSISTOR_SPI("DP5",PTRANS,"MP5",B_S,
      "GATE_TYPE_H",ALU2,"DIFF_CAP",MIN_D,C_END);
...
/***** (4) Placement (slicing structure)  *************/
/***** (4.1) Building Groups **************************/
CAIRO_ADD_DEVICE("TP5","group_2","DP5",SYM_X,C_END);
CAIRO_ADD_DEVICE("TP1","group_1","DP1",ROT_P,C_END);
...
/***** (4.2) Building Slices  *************************/
CAIRO_ADD_GROUP("group_0","slice_0","TO",2*PITCH,C_END);
CAIRO_ADD_GROUP("group_1","slice_0",C_END);
...
/***** (4.3) Building the Module  *********************/
CAIRO_ADD_SLICE("slice_0",C_END);
CAIRO_CLOSE_MODULE("OTA");
/***** (5) Area optimization  *************************/
CAIRO_RESHAPE("OTA",H,argv[2],TRUE);
CAIRO_PLACE("OTA");
/***** (6) Routing  ***********************************/
CAIRO_BEGIN_ROUTE("OTA","OTA");
CAIRO_WIRE3("2",ALU2,ALU2,ALU1,CURRENT_W,CURRENT_W,
        CURRENT_W,"TP1","source",0,"TP5","drain",0,
        CAIRO_GET_Y("MP5",TRA,REF),VER);
...
/***** (7) Defining the module interfaces *************/
CAIRO_PLACE_CON_H("TP5","gate",0,"evp1",ALU2,CURRENT_W);
...
CAIRO_END_ROUTE("OTA");
/***** (8) Verification & Statistics ******************/
CAIRO_DRC("OTA");
CAIRO_STATISTICS("OTA");}
```

Figure 5.17: *Language description of the OTA circuit shown in Fig. 5.16.*

ing. In this step, the designer allocates for each element or group of elements a single device from the language available device library and chooses the corresponding layout options. A *p*-transistor differential pair `DP1` is called in the first line. This device contains the two transistors `MP1` and `MP2` defined in the netlist. It has a separate well connection to be connected later to the supply potential (parameter `B_O`). Dummy transistors (the `DUMMY` option) are employed at both ends, and the drain diffusion capacitance is minimized (the `"DIFF_CAP", MIN_D` option), see section 5.4.1. Layout styles concerning terminal positions (section 5.4.1) can be changed in an external default *style* file (refer to Appendix A.8 for a description of the style file format) or directly in the language code as shown in the second line. This line declares a *p*-transistor device `DP5` with the second metal level is used for gate connections (the `"GATE_TYPE_H", ALU2` option). The device has its bulk connected to the source (parameter `B_S`), and the corresponding drain diffusion capacitance is also minimized.

Section (4) constructs the slicing structure. The `CAIRO_ADD_DEVICE()` function builds the horizontal *groups*. It instantiates the previously declared devices as needed. The same device can be instantiated more than once in any group. The resulting instances share the same element sizes and layout options but may have completely different shapes according to its position, orientation, and the overall shape parameter. The `CAIRO_ADD_GROUP()` function builds vertical *slices* from the previous groups. Finally, the `CAIRO_ADD_SLICE()` function adds the constructed *slices* to current module.

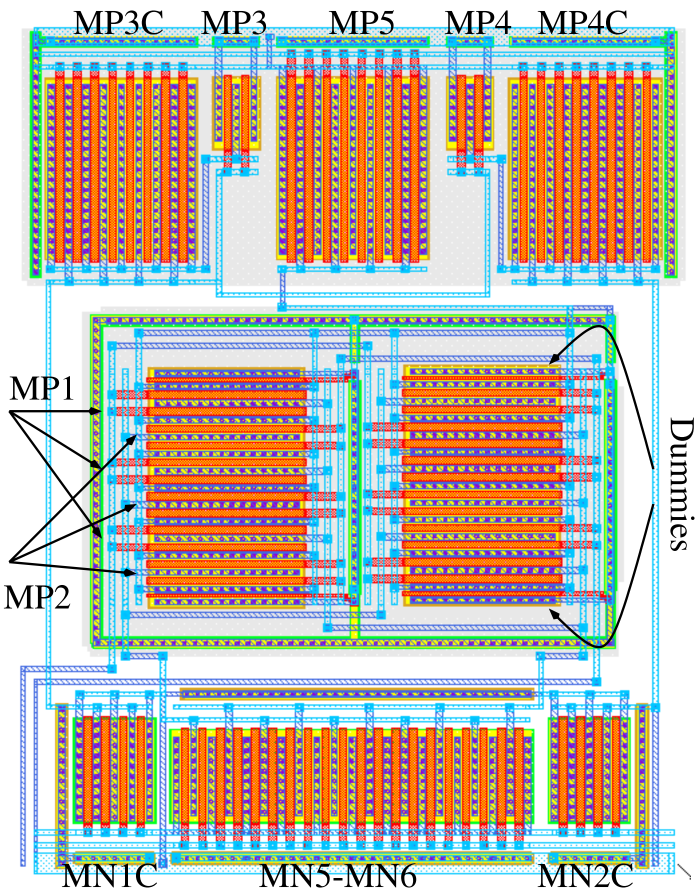
Area optimization is performed using the `CAIRO_RESHAPE()` function in section (5). In this example the global shape parameter is the layout height (`H` option) which is passed as a run-time argument when calling the compiled circuit layout description, see Fig. 5.1(b).

Sections (6) and (7) contain the routing and module terminal definitions. Routing functions support multi-layer routing with appropriate via placement. The width of each wire is determined according to the corresponding layer type and the current passing in the modules connected to it. This is achieved through the `CURRENT_W` option which captures device currents and adjusts wire widths accordingly. During routing, special functions (`CAIRO_GET_X()` and `CAIRO_GET_Y()`) capture the coordinates of device connectors and reference points.

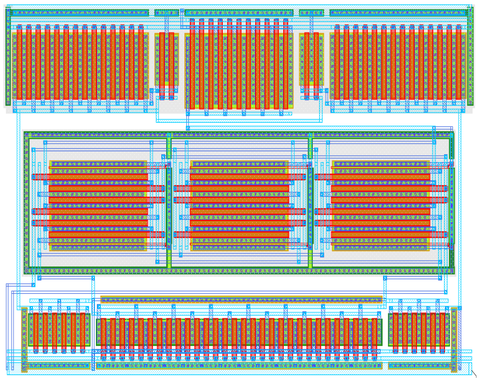
Finally in section (8) layout verification with respect to symbolic design rules is performed together with some layout information. For a detailed description of the language syntax, refer to appendix A.

Fig. 5.18(a) shows the generated layout in a 0.6-μ, 3.3-V process. As can be seen from the layout, all transistor folds are chosen such that *drains* are internal diffusions to minimize drain capacitance and enhance frequency behavior. The input differential pair is interleaved in a common centroid style over two stacks with dummy transistors placed at the end in order to avoid boundary effects and improves matching.

The OTA was then re-sized for a 0.25-μ, 2.5-V process given the same performance specifications, and the layout has been also generated using the *same* template description shown in



(a) 0.6-μ process



(b) 0.25-μ process

Figure 5.18: *Folded Cascode OTA Layout in two different processes.*

Fig. 5.17, in addition to the appropriate technology file and the sized netlist. This process is described in Fig. 5.1(b). Fig. 5.18(b) shows the resulting layout. A different aspect ratio was specified. The differential pair `MP1–MP2` has a different layout option; it has been generated on three stacks to avoid very wide transistors.

5.11 Conclusions

The dedicated language for analog layout generation, CAIRO, which aims to closely couple circuit sizing and layout generation has been presented.

Analog layout constraints are encapsulated inside procedural device generators using efficient algorithms. A slicing-tree placement structure is chosen in order to facilitate hierarchical, top-down area optimization using a fast algorithm. Simple geometrical methods are used for parasitics extraction, since they combine both computational efficiency and reasonable accuracy. Process independence has been achieved through the use of a symbolic layout approach for layout template placement and routing. Finally, a layout example of a folded cascode OTA is given to demonstrate the use of the tool.

The proposed tool thus allows to account for constraints related to the physical implementation of a given circuit such as parasitics and reliability *during* the design optimization phase. In the same time, it offers efficient solutions to improve the quality of the produced layout.

Compared to the existing symbolic layout approach for digital circuits (Genlib) that handles rigid devices, CAIRO handles deformable devices.

The next chapter introduces the circuit sizing environment COMDIAC.

Chapter 6

Circuit Sizing with Layout Parasitics

6.1 Introduction

In this chapter, the circuit sizing environment COMDIAC[1] [Porte97] is presented. This environment has been enhanced to be used in the design flow presented in section 4.3, i.e. to take into account layout parasitics during sizing.

In section 6.2, the general sizing method is introduced together with some characteristics of the COMDIAC environment.

In section 6.3, the sizing procedure of an opamp is taken as an example to demonstrate the sizing approach.

In section 6.4, the effect of parasitics is analyzed through an example.

6.2 Sizing Approach

From the discussion in sections 3.2.3 and 4.2, a knowledge-based approach was chosen. COMDIAC facilitates the capture of circuit sizing knowledge in the form of *guided* design plans.

COMDIAC is essentially composed of a set of C functions that allow the designer to accurately calculate different device parameters. For example, given transistor bias voltages, drain-source current and channel length, COMDIAC supplies the transistor width and all small-signal parameters. Using those functions, routines have been developed for basic sub-circuits sizing that can be, in turn, used in a hierarchical manner. For example, a differential pair is handled as an entity which contains two identical common-source transistors and a biasing current source. This hierarchical approach simplifies the addition of new topologies by reusing specific design knowledge.

Choosing a particular fabrication process is completely decoupled from the sizing procedure itself, such that the same circuit can be easily sized in multiple fabrication processes using different transistor models. Advanced transistor model equations like *BSIM3V3* [Liu99] and

[1]This tool is originated and maintained by Jacky Porte (porte@enst.fr) at the Ecole Nationale Supérieure des Télécommunications, Paris, France. A close co-operation with the author has allowed to complete this work.

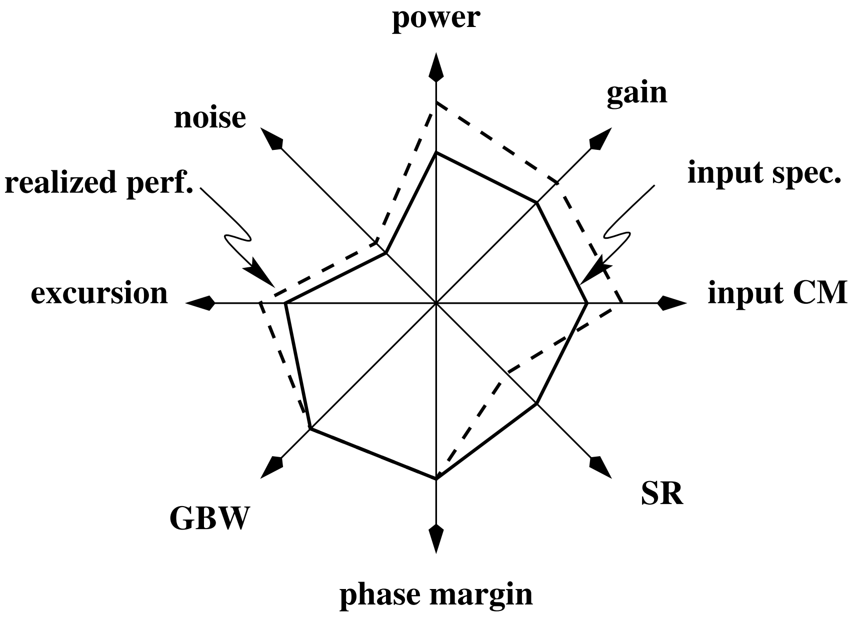


Figure 6.1: *Opamp design specification space.*

MM9 [Velghe93] as well as traditional SPICE MOS levels 1, 2 and 3 are incorporated in the tool. In addition, based on these equations, a guided user interface allows the designer to easily characterize different technologies by plotting transistor small signal parameters such as transconductances and capacitances with different bias voltages, transistor sizes and operating temperatures. This helps the designer to choose the most suitable fabrication process for a given application.

6.3 Sizing Procedure

The complexity of analog design resides in its multi-dimensional specification space. Fig. 6.1 shows a part of this space for an opamp. During the design, one must consider all these specifications in the same time. The continuous line polygon represents the input specifications, while the dashed one represents the obtained performance after sizing. Trying to move a vertex of the dashed polygon on the corresponding axis will also affect other vertices so that a compromise is always needed to obtain an overall satisfactory performance. Depending on the application, some specifications may also be added or removed from this specification space. An important step in building a sizing procedure is thus to define the input specification set.

The philosophy of sizing plans in COMDIAC is to focus on the most significant performance characteristics while leaving the possibility to the designer to control interactively design details, thanks to a fast and accurate performance evaluation based on pre-derived equations that are defined by the design plan.

6.3.1 Device Sizing

As stated above, module sizing is based on basic device sizing. There is actually one basic device implemented in COMDIAC which is the MOS transistor.

COMDIAC offers a set of functions for transistor sizing, or in other words to calculate different parameters associated with the transistor. All functions share the following input parameters:

- process parameters,
- transistor type (N or P),
- layout style, which includes the number of fingers, number of shared source and drain diffusions between fingers, and source/drain diffusion width.

There exists two types of functions. First, functions that determine transistor DC parameters:

1. Vth() calculates the threshold voltage given the gate length and width, and bias voltages V_{DS} and V_{BS}.
2. Ids_Vgs() calculates the DC drain-source current given the gate length and width, and bias voltages V_{GS}, V_{DS} and V_{BS}.
3. Ids_Veg() calculates the DC drain-source current given the gate length and width, and bias voltages V_{EG}, V_{DS} and V_{BS}.
4. W_Vgs() calculates the gate width given the current, gate length, and bias voltages V_{GS}, V_{DS} and V_{BS}.
5. W_Veg() calculates the gate width given the current, gate length, and bias voltages V_{EG}, V_{DS} and V_{BS}.
6. L_Vgs() calculates the gate length given the current, gate width, and bias voltages V_{GS}, V_{DS} and V_{BS}.
7. L_Veg() calculates the gate length given the current, gate width, and bias voltages V_{EG}, V_{DS} and V_{BS}.

Once transistor dimensions and quiescent point have been determined, a second set of functions calculates all small-signal parameters, i.e. g_m, g_{ds}, C_{gs}, C_{ds}, ...

6.3.2 Sub-circuit Sizing

The differential pair shown in Fig 6.2 is a typical reusable sub-circuit. In this schematic template, COMDIAC sizes all three transistors starting from the following input parameters:

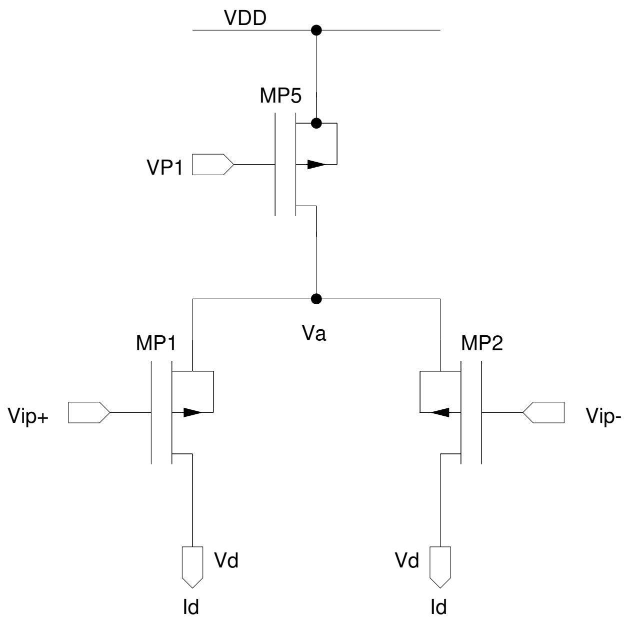


Figure 6.2: *Differential pair*.

- The supply voltage V_{DD}.
- The biasing current I_d.
- The drain voltage V_d of the differential pair.
- Input common-mode voltage Vip_{CM}.
- Gate-source bias effective voltage ($V_{EG1} = V_{EG2}$, V_{EG5}) and transistor bulk-source connections.
- Transistor lengths ($L_1 = L_2$, L_5).

The function renders

- Transistor widths ($W_1 = W_2, W_5$).
- Bias voltage V_{P1}.
- Transistor small-signal parameters.

Assuming that the bulk of transistors MP1 and MP2 is tied to their common source such that $V_{BS1} = V_{BS2} = 0$, the sizing procedure can be summarized as follows:

1. The only unknown voltage needed for transistor width calculation is V_a. From Fig. 6.2, it is obvious that $V_a = Vip_{CM} - V_{GS1} = Vip_{CM} - V_{th1} - V_{EG1}$. (Note that for p-transistors V_{GS}, V_{EG} and V_{th} are all negative quantities). Thus, V_{th1} needs to be evaluated. The threshold

voltage, however, has a certain dependency on transistor width and V_{DS}. As an initial guess, the voltage V_a is arbitrary set to V_{DD}+2 V_{EG5}

2. Calculate $V_{DS1} = V_d - V_a$ and $V_{DS5} = V_a - V_{DD}$.
3. Calculate transistor widths $W_1 = W_2$ and W_5 using function [5] above.
4. Calculate the threshold voltage V_{th1} using function [1] above.
5. Based on V_{th1}, re-calculate V_a.
6. Repeat steps (2-5) till $V_{a(i+1)} - V_{a(i)} < \epsilon$, where ϵ is the permissible voltage error.
7. Calculate the bias voltage $V_{P1} = V_{DD} + V_{GS5} = V_{DD} + V_{EG5} + V_{th5}$.
8. Knowing the dimensions and quiescent point of all transistors, small signal parameters are then calculated.

Two variants of the above procedure exist. The first considers the bulk of transistors MP1 and MP2 connected to V_{DD}. Since V_{BS} is a dominant factor in the threshold voltage, the calculation of V_a (step (5)) is done numerically by dichotomy in this case. The second variant considers a cascoded current source MP5 with an additional transistor.

6.3.3 OTA Sizing

In this section, as an example of circuit sizing procedures integrated in COMDIAC, the procedure of a simple OTA, shown in Fig. 6.3, is described. The following equations represent a simplified set showing the dependency of the performance shown in Fig. 6.1 on various design parameters:

$$\text{Power} = V_{DD} I_5 \tag{6.1}$$

$$\text{Gain} = \frac{g_{m1}}{g_{ds2} + g_{ds4}} \approx \frac{L_2.L_4.V_E}{V_{EG1}(L_2 + L_4)} \tag{6.2}$$

$$\text{GBW} = \frac{g_{m1}}{C_2} \approx \frac{I_1}{V_{EG1}} \frac{1}{C_L} \tag{6.3}$$

$$\text{PM} = 90^0 - arctan\frac{GBW}{\omega_1} + arctan\frac{GBW}{2\omega_1} \tag{6.4}$$

$$\approx 90^0 - arctan\frac{g_{m1}.C_1}{g_{m3}.C_2} + arctan\frac{g_{m1}.C_1}{2.g_{m3}.C_2} \tag{6.5}$$

$$\approx 90^0 - arctan\frac{4.V_{EG3}.W_3.L_3.C_{ox}}{3.V_{EG1}.C_L} + arctan\frac{2.V_{EG3}.W_3.L_3.C_{ox}}{3.V_{EG1}.C_L} \tag{6.6}$$

$$\text{SR} = \frac{I_5}{C_2} \tag{6.7}$$

(6.8)

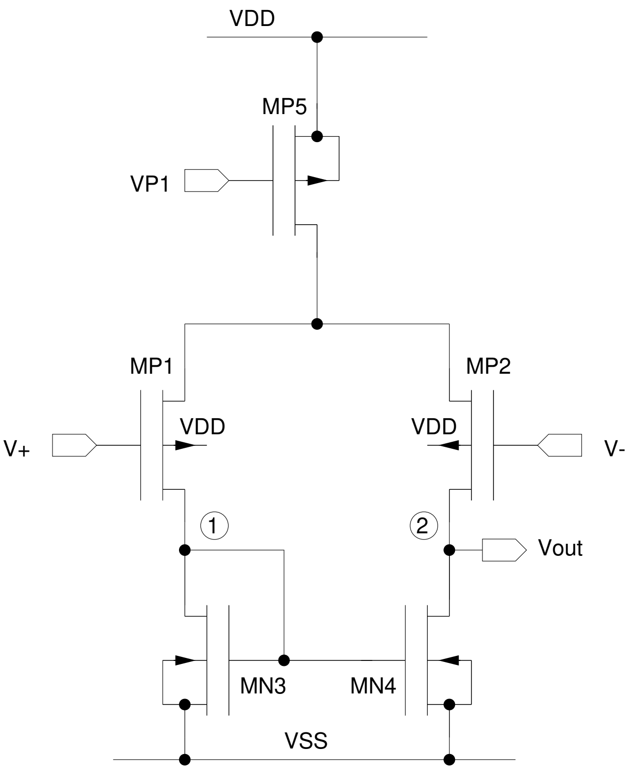


Figure 6.3: *Simple OTA.*

$$\text{Noise} \propto \frac{1}{g_{m1}} \approx \frac{V_{EG1}}{I_1} \tag{6.9}$$

$$\text{Excursions} \propto V_{DD}, \frac{1}{V_{EG}}, \frac{1}{V_{DS}} \tag{6.10}$$

where

$$C_1 = C_{gd1} + C_{db1} + C_{gs3} + C_{gb3} + C_{db3} + C_{gs4} + C_{gb4} \approx 2C_{gs3} \tag{6.11}$$

$$C_2 = C_{gd2} + C_{db2} + C_{gd4} + C_{db4} + C_L \approx C_L \tag{6.12}$$

$$\omega_1 = \frac{g_{m3}}{C_1} \tag{6.13}$$

and

I_5	bias current,
V_{EG}	transistor effective gate-source voltage($V_{GS} - V_{TH}$),
V_{DS}	transistor drain-source voltage,
L	transistor gate length,
g_m, g_{ds}	transistor transconductance and output conductance,

C_{gs}	transistor gate-source capacitance,
μ	transistor carrier mobility,
V_E	transistor Early voltage,
C_L	amplifier load capacitance.
C_1, C_2	capacitance at nodes 1 and 2 (refer to Fig. 6.3).

From these equations, four types of independent parameters can be distinguished:

1. the bias current I_5,
2. the biasing voltages V_{EG} and V_{DS},
3. process parameters V_E and μ, and
4. transistor lengths L.

Thus, by fixing the current and biasing voltages, only transistor lengths could be varied in order to satisfy a given specification. The most important performance characteristic of an opamp is its small-signal frequency behavior, i.e. the gain-bandwidth product (GBW) and the phase margin (PM). The phase margin has been chosen as the main characteristic to be optimized by iterating on transistor lengths. In addition, the GBW can be satisfied by adding an additional iteration loop on the bias current based on equation (6.3). Other specifications such as the SR, noise performance, etc., are then calculated. If obtained results are not satisfactory, the designer can modify transistor bias voltages to control these specifications. This means that simple sub-circuit sizing can be fully automated, but complex multi-dimensional circuits such as an OTA are not fully automated.

Fig. 6.4 describes the sizing procedure. According to the discussion above, a minimal set of performance specifications are selected. This includes:

- the supply voltage V_{DD},
- the gain-bandwidth product GBW or the bias current I_5,
- the phase margin PM,
- the load capacitance C_L, and
- the bias potentials V_{DS}, V_{EG} and V_{BS} of each independent transistor. Since the threshold voltage V_{TH} changes with transistor lengths during sizing, the effective gate-source voltage $V_{EG} = V_{GS} - V_{TH}$ is chosen as input parameter rather than V_{GS}.

There are basically four loops in the sizing procedure which are from the inner to the outer one as follows:

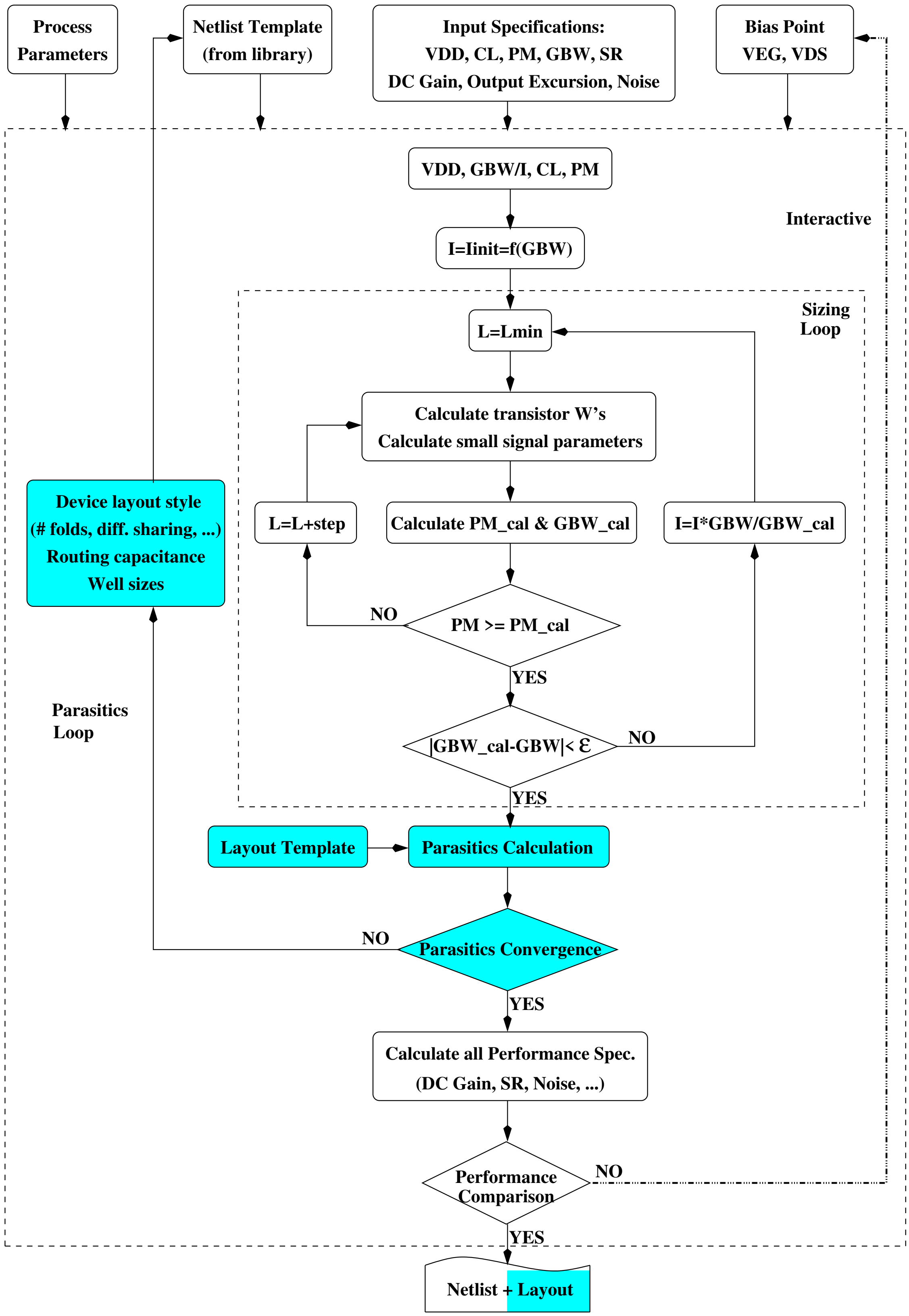


Figure 6.4: *OTA design procedure.*

1. Transistor lengths loop:
   - Initially, all transistor lengths are set to their minimal value determined by the process.
   - Using the functions described in section 6.3.1, all transistor widths and small-signal parameters (g_m, g_{ds}, C_{gs}, C_{sb}, ...) are then calculated.
   - The phase margin (PM) and gain-bandwidth (GBW) are calculated using equation (6.4). The obtained PM is often larger than the required one, since minimal transistor lengths are employed which correspond to minimal small-signal capacitances.
   - All transistor lengths are incremented and the loop is repeated till the required PM is obtained (It should be noted that lengths of some transistors can be fixed to a certain value, if desired, during this iteration loop).
2. Gain-bandwidth loop:
   - First, an initial bias current is *estimated* from the given GBW.
   - The first loop is then executed which besides satisfying the PM, calculates the exact GBW.
   - The GBW is compared with the required one. A new value for the bias current is calculated by linear interpolation and the whole process is repeated till the required GBW is satisfied.
3. Parasitics loop:
   Based on the obtained transistor sizes, the layout tool is called in the parasitics estimation mode to calculate the corresponding parasitics information (see sections 4.3 and 5.8). This step renders the exact layout style of each transistor (number of fingers, diffusion width, ...), which allows exact determination of diffusion capacitance. In addition, routing capacitance is added to the capacitances C_1 and C_2, which modifies both the GBW and PM as shown by equations (6.3) and (6.4) respectively. Loops (1) and (2) are then repeated taking into account these modified capacitances (both diffusion and routing) till parasitics convergence is reached, i.e. the calculated parasitics remains unchanged.
4. Other performance loop:
   Finally, the procedure calculates the rest of the obtained performance characteristics. These characteristics can be controlled by modifying interactively transistor bias voltages. Fixing the operating point of each transistor taking into account considerations like matching and temperature dependence increases the reliability of the circuits. The fact that the sizing process is very fast and highly accurate allows interactive exploration of a wide variety of design space points by the designer. Additional on-line documentation shows the dependency of the resulting performance characteristics on the bias information. Performance is evaluated using this derived mathematical description of circuit behavior, see equations (6.1)-(6.10).

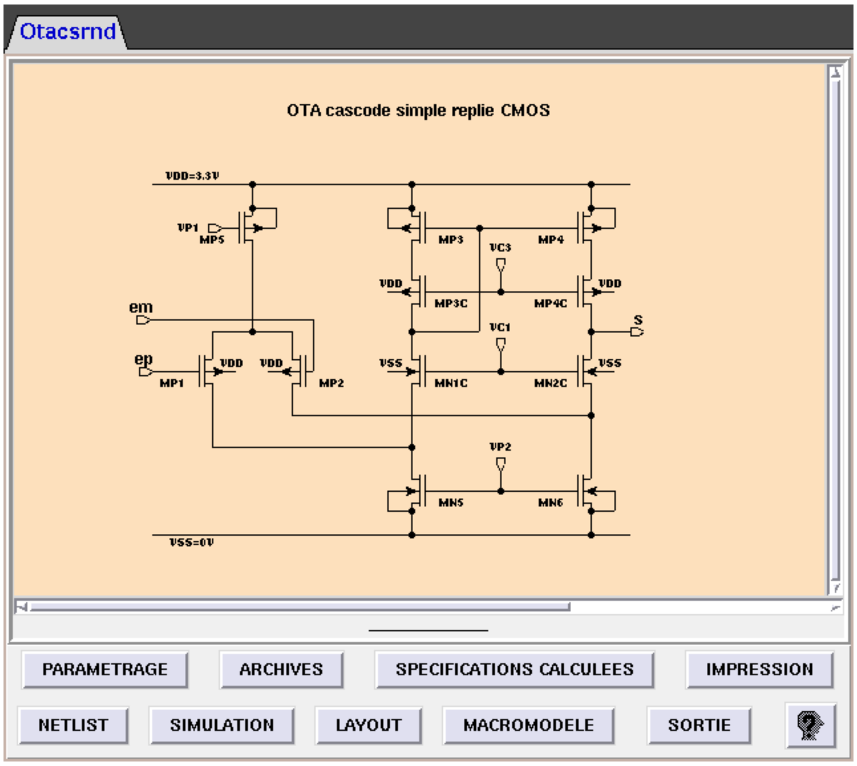
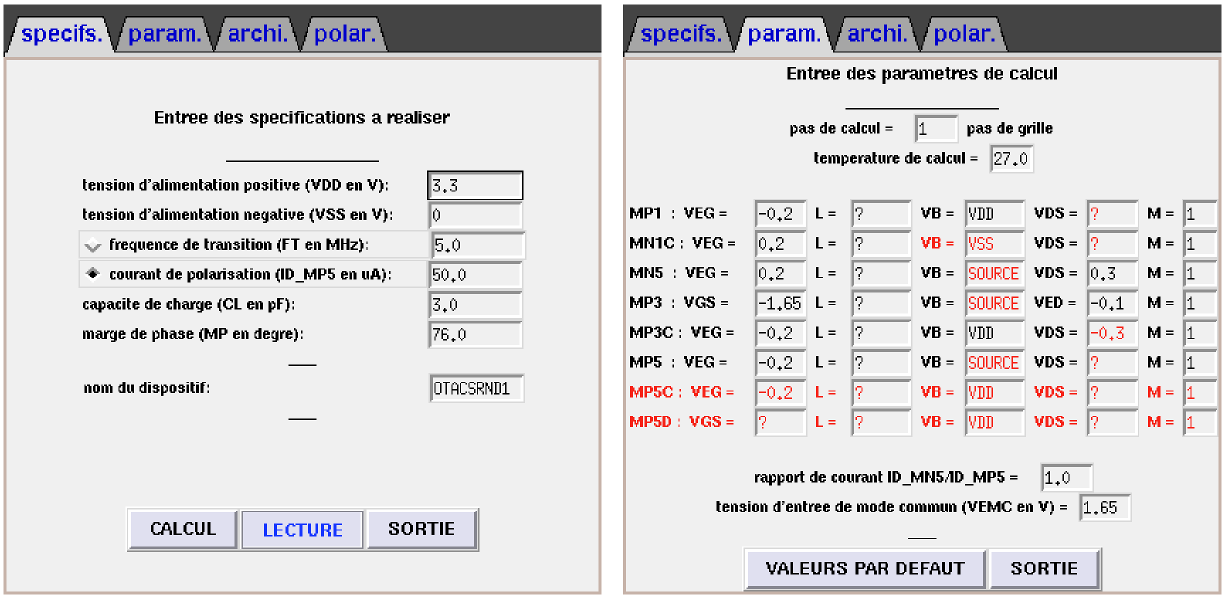


Figure 6.5: *OTA window in COMDIAC.*

The main contribution of this work with respect to COMDIAC has been the introduction of real physical parasitics in the sizing procedures (loop (3)), and the generation of the associated layout, as shown by the shaded boxes in Fig. 6.4.

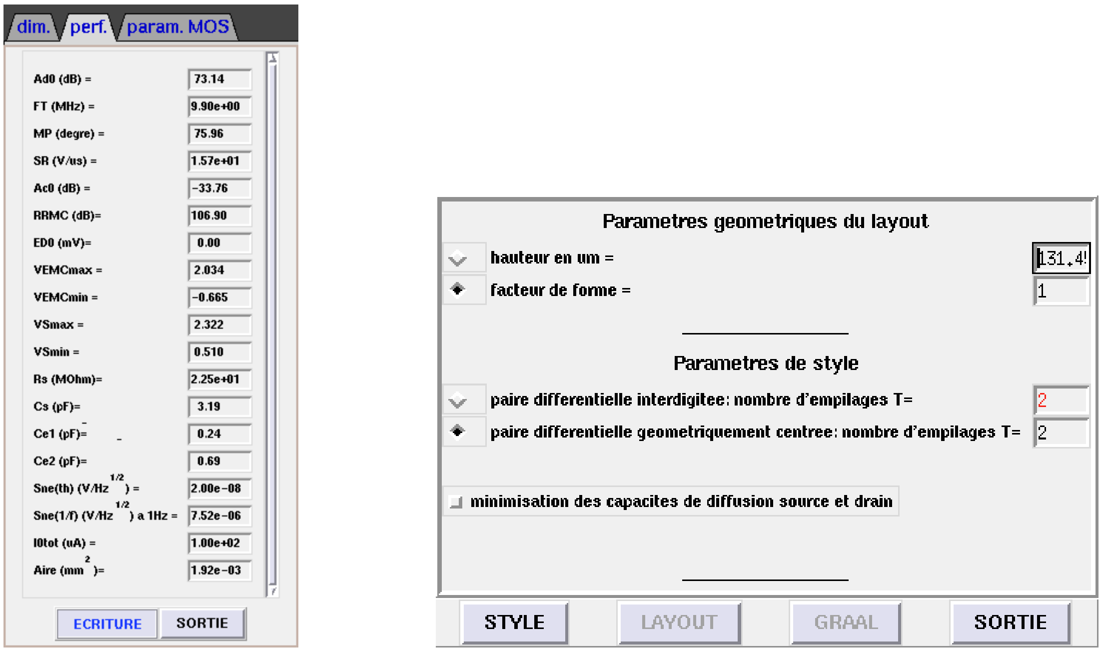
6.3.4 Interactive Graphical Interface

Fig. 6.5 shows the main window corresponding to the folded cascode OTA implemented in COMDIAC. This window controls the circuit sizing, layout and verification processes. Fig. 6.6(a) shows the specification window discussed above, while Fig. 6.6(b) shows the parameter window controlling the bias as well as the length of each independent transistor. The number of fingers (M) of all transistors are first taken to be unity. After sizing, the window shown in Fig. 6.7(a) is obtained with the realized performance. Using the same window, the obtained sizes as well as detailed transistor parameters can be displayed. The layout window in Fig. 6.7(b) is responsible for physical implementation. It allows to take into account layout parasitics during sizing. A layout shape parameter (either the layout height or the aspect ratio) is chosen, together with some layout options. Using the **STYLE** button the sizing-parasitics loop is executed which results in the exact number of fingers for each transistor, see section 4.3. Using the **LAYOUT** button the final layout is physically generated. Finally, a verification-by-simulation step is accomplished. A special



(a) (b)

Figure 6.6: *OTA: (a) main parameters and (b) transistor bias.*



(a) (b)

Figure 6.7: *OTA: (a) obtained results and (b) layout parasitics calculation and generation.*

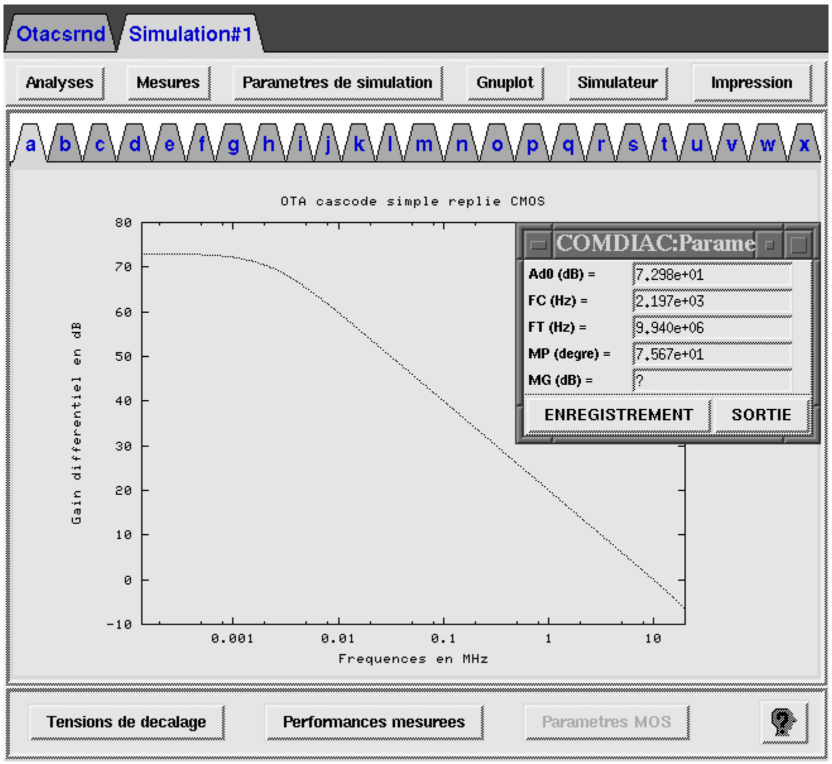


Figure 6.8: *OTA: verification by simulation.*

window, shown in Fig. 6.8, allows the designer to perform a series of simulations to measure the obtained performance in order to compare it with calculated results. It generates the appropriate netlists with the required test sources and simulation cards, calls the desired simulator, and finally does the necessary post processing for data extraction and curve display.

The graphical interface has been built using the Tcl/Tk language [Ousterhout94].

6.4 Impact of Parasitics

In this section, the effectiveness of the layout-oriented approach is demonstrated through an example, that quantifies the impact of parasitics on circuit sizing. Consider the folded cascode OTA shown in Fig. 6.5. The circuit has been synthesized using different layout parasitics considerations in a 0.6-μm technology. The OTA is sized for a V_{DD} of 3.3V, a GBW of 65MHz, a PM of 65^o and a load capacitance of 3pF. For comparison, the input CM voltage range as well as the output voltage range are kept the same for all cases by adjusting transistor bias voltages. Channel lengths of all transistors in the OTA are also restricted to be the same.

Table 6.1 shows the obtained sizing results for each case of parasitics consideration, it also shows simulation results of the corresponding extracted netlist with all parasitics taken into account (diffusion, routing and coupling capacitances) between brackets. Extraction has been done

Obtained Results	Case (1)	Case (2)	Case (3)	Case (4)
Transistor Lengths (μ)	1.8	0.9	1.4	1.3
DC gain (dB)	70.1(70.1)	55.0(56.59)	66.1(66.1)	64.7(64.7)
Fcutoff (KHz)	21.3(19.7)	122.0(107.0)	32.9(33.1)	39.4(39.4)
GBW (MHz)	64.9(58.1)	66.5(71.2)	65.0(62.6)	65.8(66.1)
Phase margin (degrees)	65.3(56.3)	65.4(72.4)	65.4(64.4)	65.15(65.4)
Slew rate ($V/\mu s$)	94.0(86.5)	103.0(98.1)	93.3(93.3)	93.0(94.4)
CM gain (dB)	-30.6(-30.6)	-22.1(-22.0)	-27.82(-27.82)	-26.8(-26.8)
CMRR (dB)	100.7(100.7)	77.1(78.6)	93.9(93.9)	91.6(91.6)
Offset voltage (mV)	0.0(0.0)	0.0(-0.1)	0.0(0.0)	0.0(0.0)
Output Resistance ($Mohm$)	2.4(2.4)	0.38(0.47)	1.5(1.47)	1.23(1.23)
Input Capacitance$^+$ (pF)	0.83(1.29)	0.27(0.20)	0.54(0.71)	0.52(0.42)
Input Capacitance$^-$ (pF)	2.57(2.61)	0.73(0.84)	1.64(1.68)	1.48(1.49)
Input noise voltage (μV)	83.9(96.1)	101.6(85.6)	83.3(87.8)	82.7(85.8)
Thermal noise density ($nV/\sqrt{Hz}$)	7.2	6.98	7.15	7.13
Power dissipation (mW)	2.0(2.0)	2.4(2.2)	2.1(2.1)	2.1(2.1)

Input specifications: $V_{DD} = 3.3$V, $GBW = 65$MHz, phase margin = 65degrees, $C_{load} = 3$pF,
Input CM range = $[-0.55, 1.84]$V, Output range = $[0.51, 2.31]$V (determined by transistor bias voltages).
Case 1: Sizing with no layout capacitances (Neither diffusion nor routing).
Case 2: Sizing with diffusion capacitance assuming single transistor folds and no routing capacitance.
Case 3: Sizing with calculation of exact diffusion capacitance and neglecting routing capacitances.
Case 4: Sizing considering all layout parasitics.
Values between brackets are obtained after layout generation, extraction and simulation.

Table 6.1: *Sizing, layout and simulation results.*

using the commercial Cadence design system. In case (1) no layout capacitances (neither diffusion nor routing) have been taken into consideration, only gate capacitances and transistor folding are considered. It can be seen that all DC characteristics match the extracted layout simulation results, while for the GBW and PM, a considerable discrepancy can be recognized. In case (2) diffusion capacitance has been taken into consideration, while assuming only one fold per transistor and neglecting routing capacitance, i.e. no layout information is used during synthesis. Results show that the GBW and PM exceed the required specifications. In fact, since diffusion capacitance sharing is ignored leading to an over-estimation of diffusion capacitance, the obtained transistor lengths are too small. This implies that other specifications like the input noise, the DC gain and the output resistance could not be optimized. Note also the resulting offset voltage after folding due to slight modification of transistor widths after rounding to the fabrication process grid. Case (3) shows sizing results with layout information concerning *exact* diffusion capacitance, no routing capacitance is considered. We notice only a slight difference in the GBW and PM between synthesized and extracted netlist simulations. However, both specifications could not be satisfied. Case

(4) shows results with all parasitic capacitance information being considered during the synthesis phase. All results match the extracted netlist simulations.

Three calls of the layout tool were needed during the iteration loop before parasitics convergence. The sizing time for each case does not exceed two minutes.

Fig. 5.18 shows the generated layout for case (4). As can be seen from the layout, all transistor folds are chosen such that *drains* are internal diffusions to minimize drain capacitance and enhance the frequency behavior. The input differential pair is in a common centroid style with dummy transistors at the end in order to improve transistor matching.

6.5 Conclusions

The knowledge-based circuit sizing environment (COMDIAC) has been presented. This tool is a set of sizing procedures for a fixed library of schematic templates.

Introducing a new template must be preceded by a thorough study of the circuit leading to the definition of *all* performance characteristic equations. Using a hierarchical sizing approach permits to reuse the stored knowledge related to frequently used sub-circuits.

All basic sizing procedures of COMDIAC have been modified to take into account layout parasitics through multiple calls to the layout tool described in the previous chapter.

Finally, in order to demonstrate the efficiency of the proposed method, a high performance OTA has been sized using different parasitics considerations.

Chapter 7

Low-voltage Switched-Capacitor Circuit Design

7.1 Introduction

This chapter starts the application part of the thesis. The layout-oriented design methodology described in section 4.3 together with the layout tool described in chapter 5 and the circuit sizing tool described in chapter 6 are used in a challenging application to demonstrate their use efficiency during both the design and future design reuse. Now, we'll move to the designer position who is facing a certain design problem and starts with a thorough study. The purpose of this chapter is thus to introduce the design problem and to study it. Solutions to some design problems are also proposed.

After fixing very low-voltage circuits as a target application, and the SC technique for the implementation in section 2.5, section 7.2 introduces the problems associated with these types of circuits together with the existing techniques to deal with them.

In section 7.3, two proposed schemes to allow SC circuits under very low-voltage are presented. Both configurations are based on a special low-voltage bootstrapped switch which is introduced in section 7.4. Emphasis is made on the compatibility of the circuitry with modern low-voltage technologies.

In section 7.5, a fully-differential very low-voltage opamp structure is presented to be used in the previous configurations.

Finally, the chapter ends with some concluding remarks.

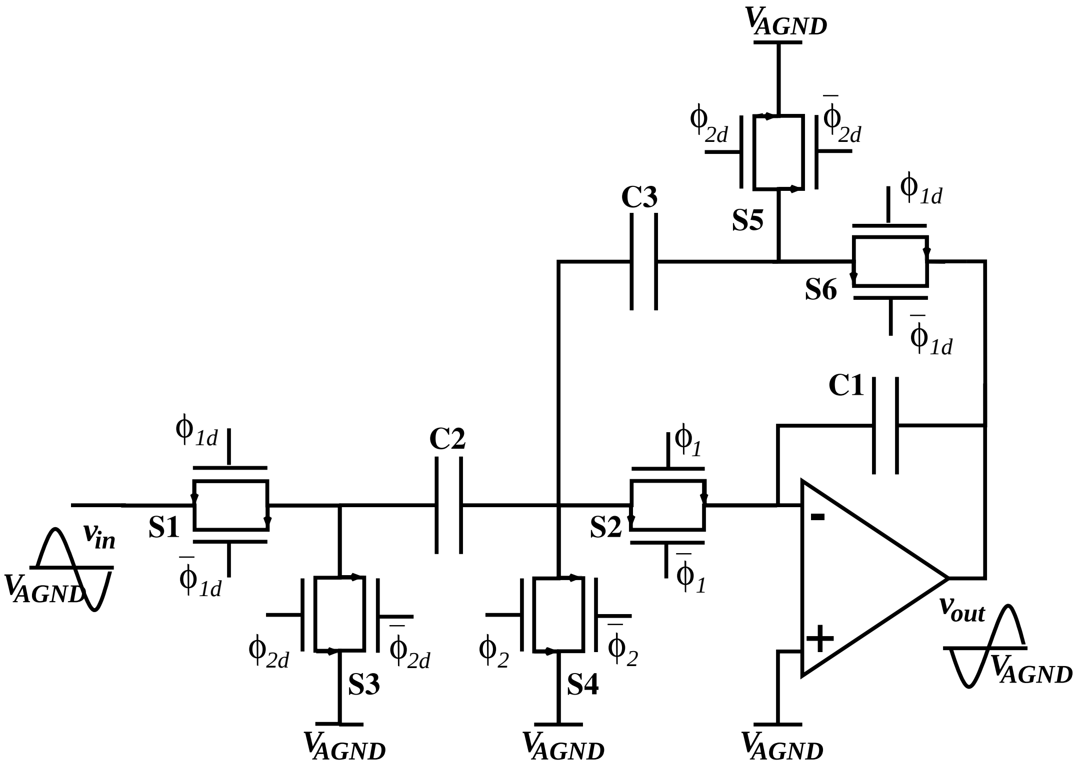


Figure 7.1: *A first-order Switched-capacitor low-pass section.*

7.2 Low-voltage Switched-Capacitor Problems

The main problem under low-voltage operation of SC circuits is the switch on-conductance. The conductance of an n-transistor in the linear region of operation is given by [Laker94]

$$\begin{aligned} g_{ds} &= \mu C_{ox}\frac{W}{L}(V_{GS} - V_{thn}) \\ &= \mu C_{ox}\frac{W}{L}(V_{DD} - V_S - V_{thn}) \end{aligned} \tag{7.1}$$

where the source potential V_S represents the signal value to be switched. The threshold voltage V_{th} for standard CMOS technologies is around 0.7V. Thus for supply voltages around 1V, the signal swing would be severely limited to very small and unpractical values.

Fig. 7.1 shows a typical first-order low-pass SC section. The analog ground potential V_{AGND} is usually set to $V_{DD}/2$ to maximize signal excursions. The CMOS switch is fully operational for supply voltages higher than the sum of the threshold voltages of both transistors $V_{thn} + V_{thp}$. The n-transistor has higher conductance for signals near to V_{SS}, as shown by equation 7.1, while the p-transistor has higher conductance for signals near to V_{DD}. For lower supply voltages, a conductance gap begins to appear around the middle of the supply range as will be shown in section 7.4. This means that under very low-voltage operation, this configuration does not work anymore.

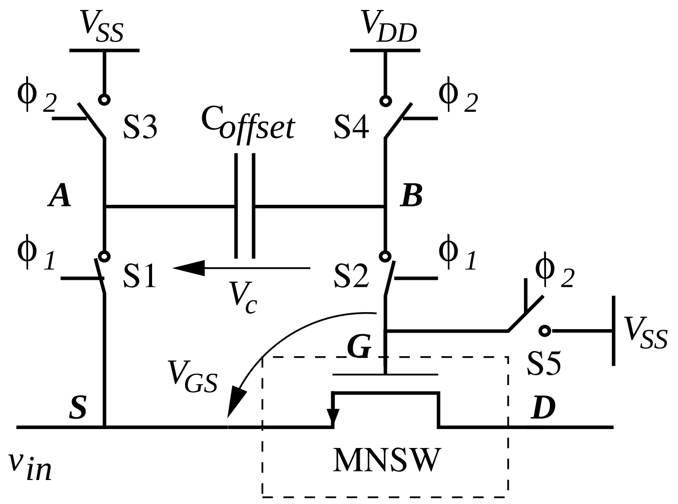


Figure 7.2: *Basic switch bootstrapping circuit.*

Some solutions to this problem can be found in the open literature. Special fabrication processes have been modified to have, besides standard transistors, low-threshold transistors. These processes, however, are more expensive since processing steps must be added during circuit fabrication. Another technique which is widely used in standard CMOS technologies is clock voltage multiplication (VM). A high voltage (usually twice V_{DD}) is generated on chip using an additional VM circuitry which is then used to drive critical switches [Rabii97]. This technique, however, is not power efficient and not compatible with very advanced low-voltage CMOS processes where gate oxide breakdown becomes an issue.

Alternatively, the switched-opamp (SO) technique [Crols94], [Baschirotto97a] has been proposed to get around this problem. Reference voltages are set to V_{DD} and V_{SS}, the remaining critical switches at the outputs of each opamp, e.g. switch S6 in Fig. 7.1, are then eliminated by switching the opamp itself. However, the switch S1 connected to the input of the circuit remains, and it restricts severely the maximum allowable amplitude of the input signal. The switching of the opamp may also affect the speed of the circuit. In addition, serious non-linearity problems emerge which limit the attainable peak SNDR [Peluso98b]. This results in relatively poor performance of the SO circuits.

In the rest of this chapter, a modified SC arrangement is introduced to allow very low-voltage operation. The design of the corresponding building blocks are then discussed.

7.3 Proposed Technique

In this section two configurations are proposed to solve the problem of very low-voltage operation of SC circuits. Both solutions are based on a special low-voltage *bootstrapped* switch [Brandt96] whose basic operation is demonstrated in Fig. 7.2. This figure shows the signal switch MNSW together with five additional switches (S1-S5) and an additional capacitor C_{offset}. Switches S3 and S4 charge the capacitor during ϕ_2 to V_{DD}. During ϕ_1 switches S1 and S2 add the pre-charged capacitor in series with the input voltage v_{in} such that the gate-source voltage of transistor MNSW is equal to the voltage V_C ($\approx V_{DD}$) across the capacitor. Switch S5 fixes the gate voltage of MNSW to V_{SS} during ϕ_2 to make sure that the transistor is in the off state.

This switch arrangement allows rail-to-rail signal switching, since the gate-source voltage is always constant independently of the input signal. A transistor-level implementation of the bootstrapped switch, which is fully compatible with modern low-voltage CMOS processes, is given in section 7.4. A trivial solution to the low-voltage operation of the SC section shown in Fig. 7.1 is to use the bootstrapped switch everywhere by replacing all CMOS switches. However, in order to minimize the number of bootstrapped switches and maintain circuit simplicity two configurations are proposed in the following sections.

7.3.1 The Charge Cancellation Scheme

In Fig. 7.1, there are only two switches that are always switching a varying voltage signal, namely S1 and S6. All other switches are connected to a fixed reference potential V_{AGND}. Consequently, the minimum number of bootstrapped switches could be possibly reduced to only two switches. Based on this remark, Fig. 7.3 proposes another SC implementation of the low-pass section shown in Fig. 7.1. In order to maximize switch conductance, V_{SS} is used as the reference potential such that a simple n-transistor suffices. However, the input DC voltage and the opamp output quiescent DC voltage is set to $V_{DD}/2$ to maximize signal swing. The voltage difference between the opamp input and output is compensated by injecting a fixed amount of charge, through C_{CM}, at the opamp input every clock cycle [Baschirotto97b]. This allows the simultaneous optimization of switch operation and output signal swing.

From Fig. 7.3, at no input signal the charge injected at the opamp input virtual ground (node 3) at the end of ϕ_1 is given by

$$Q_{inj} = C_2 \left(\frac{V_{DD}}{2} - V_{SS} \right) + C_3 \left(\frac{V_{DD}}{2} - V_{SS} \right) - C_{CM} \left(V_{DD} - V_{SS} \right) \tag{7.2}$$

where V_{SS} is set to zero potential. In steady state conditions no charge transfer should occur, i.e. no charge injection in the opamp virtual ground, then by choosing

$$C_{CM} = \frac{1}{2}(C_2 + C_3) \tag{7.3}$$

Q_{inj} reduces to zero.

For reliability reasons, the bootstrap circuit *must* always be connected between the gate and the terminal of the transistor (S1 and S6) having the lower voltage just before the switch is turned on, the source in this case, such that when the gate voltage is raised by V_{DD} with respect to the source, the gate-drain voltage remains always below V_{DD}.

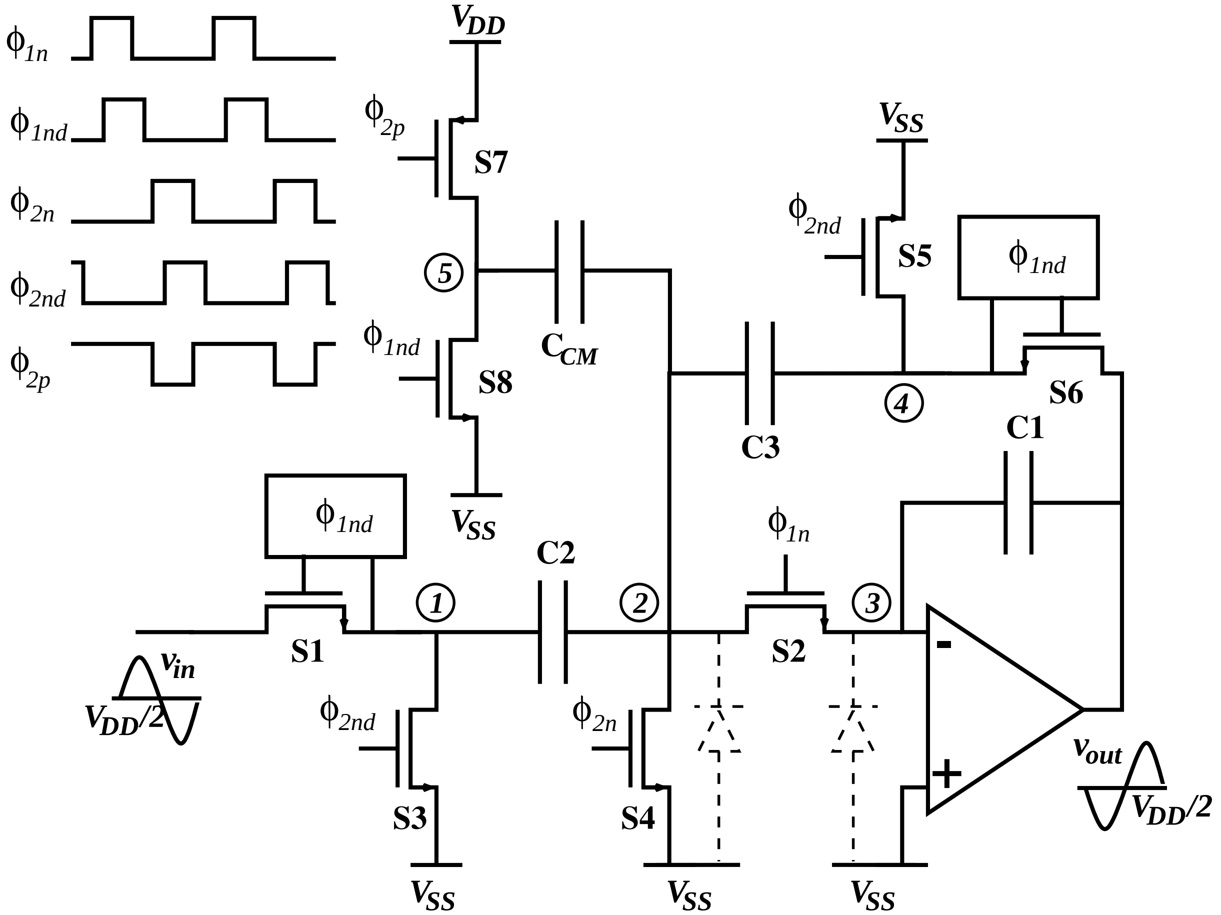


Figure 7.3: *The SC section shown in Fig. 7.1 using the charge cancellation scheme.*

Charge injection is a potential problem in SC circuits. It takes place at the turn-off of switches when the channel charge, which depends directly on V_{GS}, flows from under the gate out through the source and drain terminals and is injected into nearby capacitors causing charge errors. For switches with varying large signals, namely S1 and S6, the channel charge is modulated by the signal, inducing signal dependent distortion. However, the use of bootstrapped switches contributes to minimizing this effect by maintaining a signal-independent gate-source voltage. Delayed clock phases are also employed to further reduce charge injection. For example, in Fig. 7.3, switches S1 and S6 at the signal potential should be driven with a delayed clock phase (ϕ_{1nd}) with respect to that driving S2 (ϕ_{1n}).

Using V_{SS} at the opamp input eases the biasing of input transistors of the low-voltage opamp, however, it may cause charge leakage due to negative transient spikes [Baschirotto97a]. Reverse-biased diodes corresponding to drain/source-bulk junctions of switch transistors exist on all nodes of the SC circuit. Large negative voltage spikes could then forward bias these diodes leading to charge loss to the substrate. These current spikes also may cause noise coupling

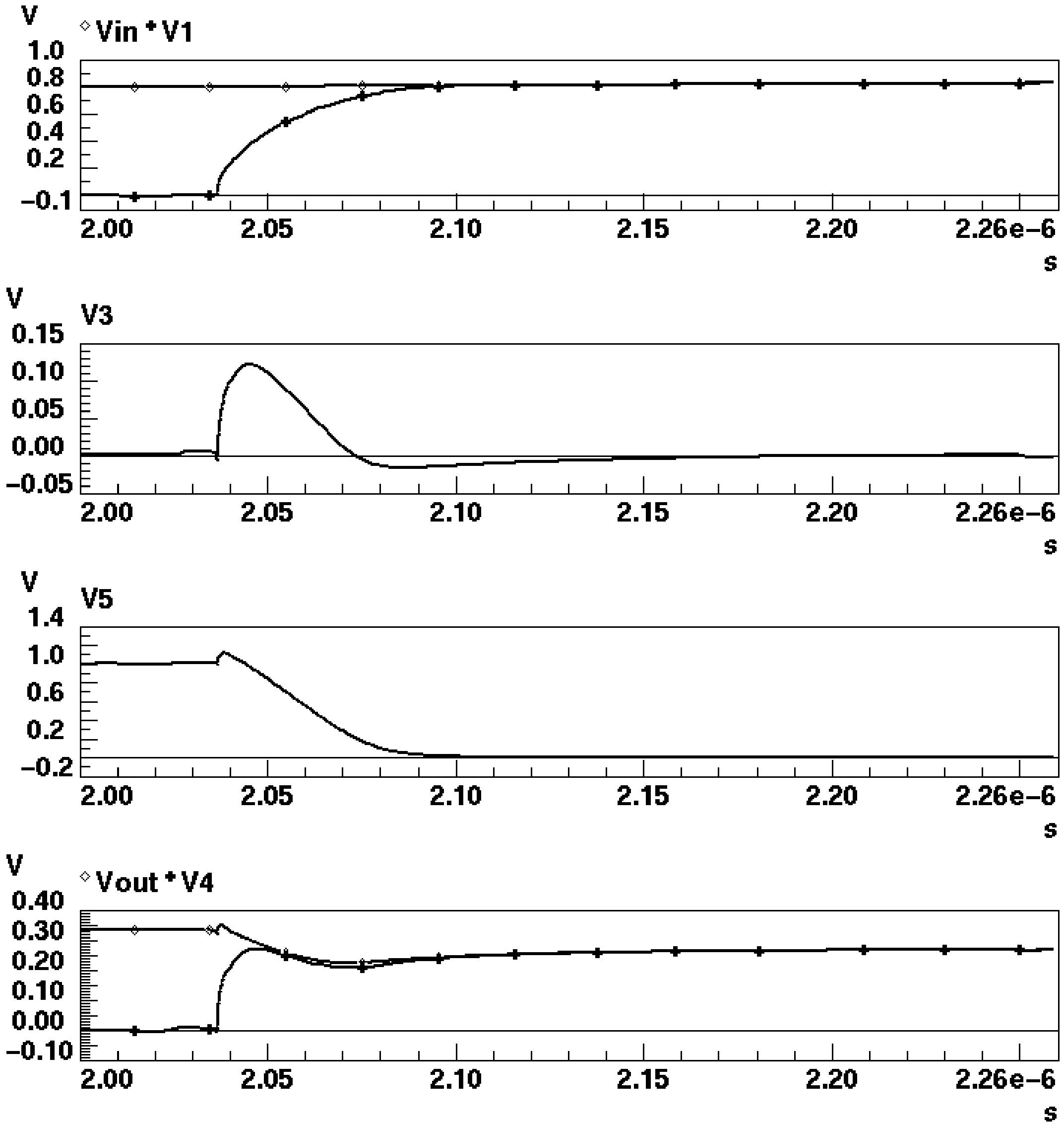


Figure 7.4: *Simulation of the integration phase* ϕ_1 *transition of Fig. 7.3.* $C_1 = 1.41pF$, $C_2 = C_3 = C_{CM} = 1pF$, $W_1 = W_6 = 2W_8$, $f_s = 2MHz$, *and the opamp GBW=*$3f_s$.

to other parts of the circuit. Nodes 2 and 3 in Fig. 7.3 are subjected to such spikes, and the drain/source diodes associated to these nodes are shown.

At the beginning of the integration phase ϕ_1, the following three voltage steps are applied to node 2 through the charge carrying capacitors: $v_{in} - V_{SS}$ through C_2 and S1, $v_{out} - V_{SS}$ through C_3 and S6, and $-(V_{DD} - V_{SS})$ through C_{CM} and S8. The opamp is responsible to keep this node at V_{SS} through S2. However, due to the opamp finite bandwidth, spikes may appear as a result of these voltage steps. The two steps through C_2 and C_3 are always positive while that at C_{CM} is always negative. The resultant spike can be kept positive by ensuring that the positive voltage steps are injected before the negative one. This can be achieved in two ways [Baschirotto97b]: Assuming S2 is not conducting, which is actually not true, the switch on-resistances can be adjusted to yield complete voltage cancellation at node 2, or at least keep the voltage spike always in the positive direction (refer to appendix B). Delayed clocks can also be used such that positive steps are applied before negative ones. The resistance of switch S2 should also be kept low enough in order to allow the opamp to rapidly restore the voltage at node 2 to V_{SS}. Fig. 7.4 shows simulation results of voltage transitions during ϕ_1. For this example, $C_1 = 1.41pF$, $C_2 = C_3 = C_{CM} = 1pF$, and the opamp GBW=$3f_s$. At the beginning of ϕ_1, assuming the opamp is not fast enough, the voltage on node 2 is given by (refer to section B.2)

$$v_2 = \frac{R_8 - R_1}{2R_8 + R_1} V_{DD} e^{-3t/(2R_8+R_1)C_2} \tag{7.4}$$

where R_1 and R_8 are switches S1 and S8 on-resistances respectively. Hence, in order to control the spike at node 2, the condition $R_8 \geq R_1$ must be satisfied, so transistor widths are adjusted such that $W_1 = W_6 = 2W_8$. The spike on node 3 is shown to be positive. Also shown is the opamp output settling transient.

At the beginning of ϕ_2, node 2 is also subjected to negative spikes. These spikes are harmless to the signal as they occur in the reset phase. However, large substrate current spikes are not desirable as they may induce noise to sensitive nodes elsewhere in the circuit. The same techniques discussed above could also be employed. In this case, delayed clock phases are used as shown in Fig. 7.3. The resistance of S4 should also be kept low to speed up voltage recovery.

A disadvantage of this charge compensation scheme by using an extra capacitor C_{CM} switched between V_{DD} and V_{SS} is the increase of the white noise level. In addition, an error in the C_{CM} size results in extra offset, while all the noise present on V_{DD} is injected into the signal path. The latter problem is, however, greatly alleviated when using a fully differential structure [Baschirotto97b].

7.3.2 The Double Reference Voltage Scheme

Another technique that avoids the extra charge compensation capacitor is shown in Fig. 7.5. Two reference voltages are used: V_{SS} at the opamp input where a normal n-switch can be used to switch the ground voltage. And a $V_{DD}/2$ quiescent DC voltage at the opamp output and at the circuit input to maximize signal swing. The bootstrapped switch is used to switch signals at this voltage level. In this case charge injecting capacitors $C2$ and $C3$ are not reset to V_{SS}, but to $V_{DD}/2$.

The negative spikes problem described in section 7.3.1 also exists on nodes 2 and 3. In this structure, however, injected spikes may have both positive and negative values depending on the signal direction. Thus, controlling the step sequence is not effective in this case. But since both injected steps are always *opposite* in direction, adjusting switches S1 and S2 resistances (refer to appendix B) could help to reduce this effect. In addition, the maximum step value is limited to $V_{DD}/2$ which is around 0.5V for very low-voltage operation. This represents the maximum spike height for an open circuit at S2. In practical, the spike height is much lower than this value due to

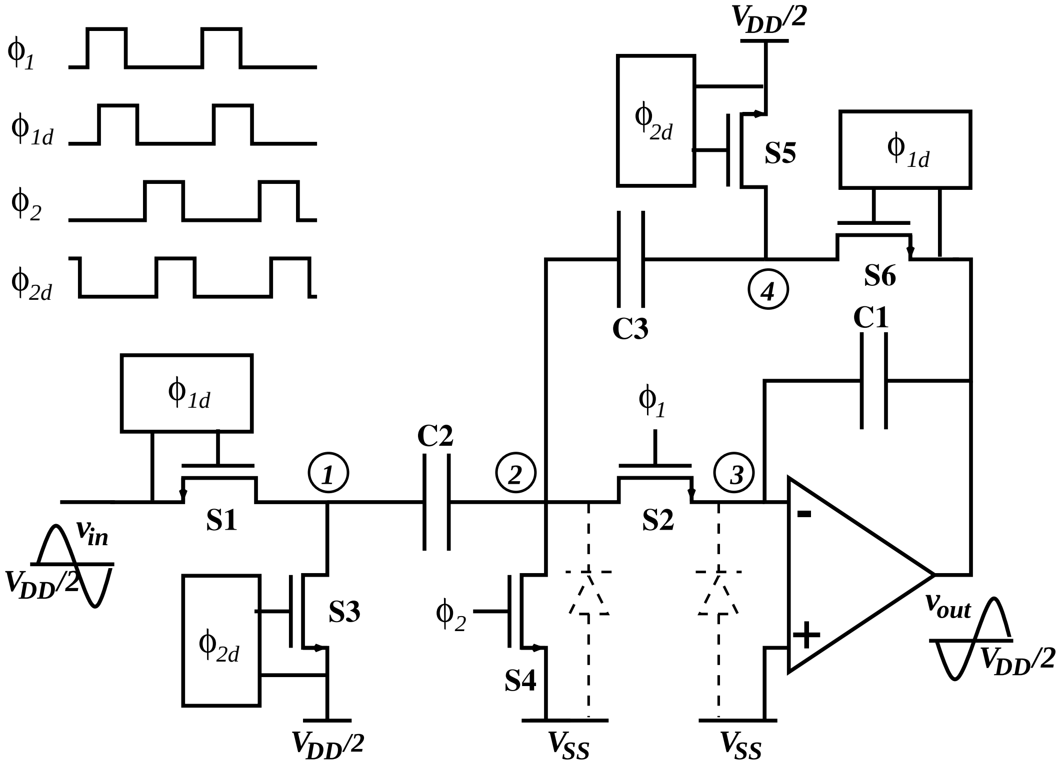


Figure 7.5: *The SC section shown in Fig. 7.1 using the double reference voltage technique.*

the finite switching time and the fact that the node is already connected to the opamp which has a certain speed. As a result, this spike is usually not sufficient to forward bias the source/drain-bulk diode. The on-resistance of switches S2 and S4 should also be kept low enough in order to allow fast voltage recovery to V_{SS} on node 2. Fig. 7.6 shows simulation results. For this example, $C_1 = 1.41pF$, $C_2 = C_3 = 1pF$, and the opamp GBW=$3f_s$. The widths of transistors S1 and S6 are adjusted to be equal. The spike on node 3 is shown to be limited to very small values which are insufficient to turn on the reverse-biased diodes.

As in the previous section, besides the use of bootstrapped switches, delayed clock phases further reduce signal dependent charge injection.

In contrast to the scheme shown in the previous section, the switch terminal with the lower potential can not be determined a priori. This means that at the switching moment, the voltage drop around the gate oxide at one terminal of the transistor might exceed V_{DD} leading to an oxide overstress on this side and causing a transistor reliability problem on the long term. This could be a serious problem. However, as will be shown in section 7.4, it is actually simple to make the bootstrapped switch fully symmetrical. In this case, the gate potential is referenced either to the source or the drain, whichever is lower, thus eliminating any potential oxide overstress.

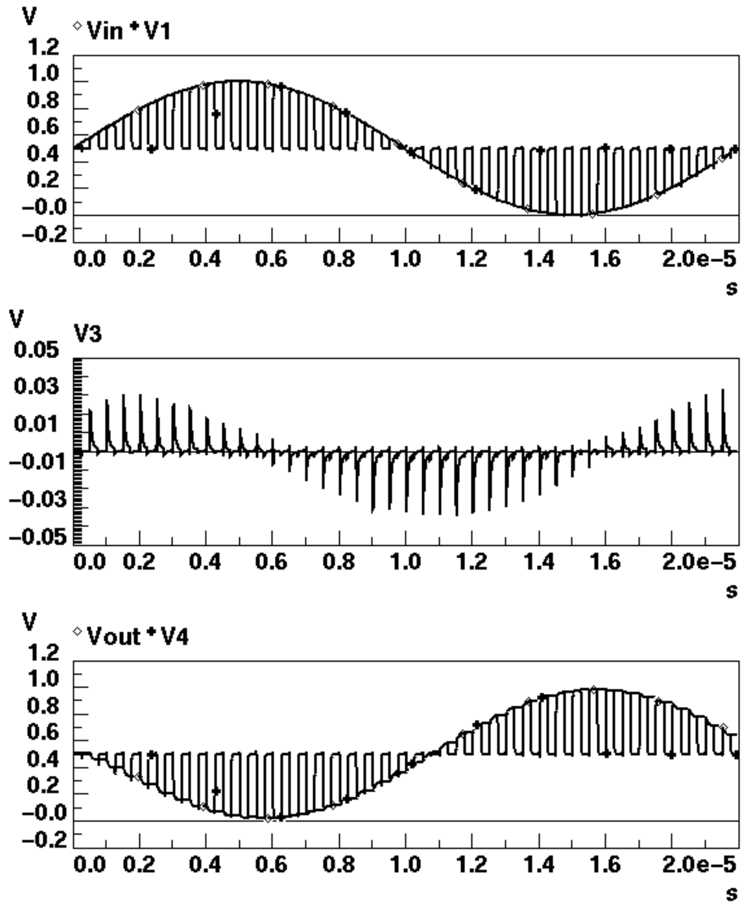


Figure 7.6: *Simulation of the low-pass SC section shown in Fig. 7.5.* $C_1 = 1.41pF$, $C_2 = C_3 = 1pF$, $W_1 = W_6$, $f_s = 2MHz$, *and the opamp GBW=*$3f_s$.

7.4 Low-voltage Bootstrapped Switch

This section describes the circuit implementation of the bootstrapped switch shown in Fig. 7.2. The transistor level circuit of the bootstrapped switch is illustrated in Fig. 7.7. Transistors MN1, MP2, MN3, MP4 and MN5 correspond to the five ideal switches S1-S5 respectively. Additional transistors and modified connectivity shown in Fig. 7.7 were introduced to extend all switch operation from rail-to-rail while limiting all gate-source voltages to V_{DD}. It is evident that the worst case input signal (with respect to switch operation) is that of $v_{in} = V_{DD}$, which is the value attributed to v_{in} in the discussion hereafter.

The n-transistor MN1 which has to switch V_{DD} to make the circuit fully efficient. For this reason, its gate voltage is also bootstrapped, i.e. connected to the gate of MNSW.

Additional critical problems arise on nodes B and G as their voltages reach $2V_{DD}$ due to bootstrapping: First of all, transistor MP4 must remain OFF during ϕ_1 in order not to loose the charge stored on C_{offset} during ϕ_2. If the clock is used to drive it as shown in Fig. 7.2, its gate-source

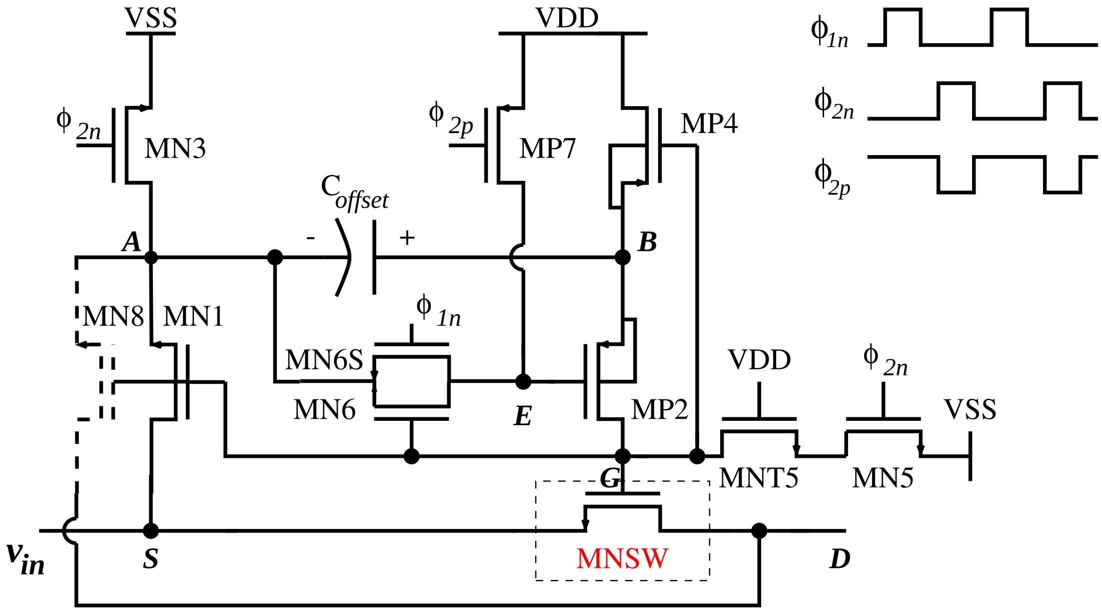


Figure 7.7: *Proposed implementation of the switch bootstrapping circuit.*

voltage would be $-V_{DD}$ and the transistor can't be turned OFF. That's why its gate is connected to node G which provides a voltage of $2V_{DD}$ during ϕ_1 cutting-off the transistor, and a voltage of V_{SS} during ϕ_2 which ensures its high conductivity.

Secondly, the gate-source voltage of transistor MP2 could reach $-2V_{DD}$ during ϕ_1 causing reliability problems. In Fig. 7.7 a solution is proposed. Transistor MN6 is used to connect the gate of MP2 (node E) to node A thus keeping its gate-source voltage equal to $-V_{DD}$ (the voltage across C_{offset}) during ϕ_1. During ϕ_2 transistor MP7 connects it to V_{DD} turning it OFF.

The gate of the n-transistor MN6 is tied to node G to keep it conducting as the voltage on node A rises to V_{DD} during ϕ_1. There is thus a dependency loop inhere; In order to turn on MN6, it must have a sufficient gate-source voltage i.e. MP2 must then be conducting! Transistor MN6S is then necessary as a *startup* to force transistor MP2 to conduct. This on transition is depicted in Fig. 7.8 for a V_{DD} of 1V: when ϕ_1 goes high, transistor MN6S is turned on since its source (node A) is discharged to V_{SS} at the beginning of ϕ_1. Node E thus goes from V_{DD} to V_{SS} through transistor MN6S, this turns on transistor MP2 and consequently the voltage on node G begins to rise turning on transistors MN1, MN6 and MNSW. Node A is, hence, connected to the input and point G rises to $V_{DD} + v_{in}$.

It should be also noted that for an NWELL process, the bulk of transistors MP2 and MP4 must be tied to the highest potential i.e. node B, and not to V_{DD} in order to prevent latch-up.

Lastly, transistor MNT5 has been added in series with MN5 in order to prevent the gate-drain voltage of the latter from reaching $2V_{DD}$ during ϕ_1. The bulk of MNT5 is, however, tied to V_{SS}. During ϕ_1 when it is off, its drain-bulk diode junction voltage reaches a reverse bias voltage of

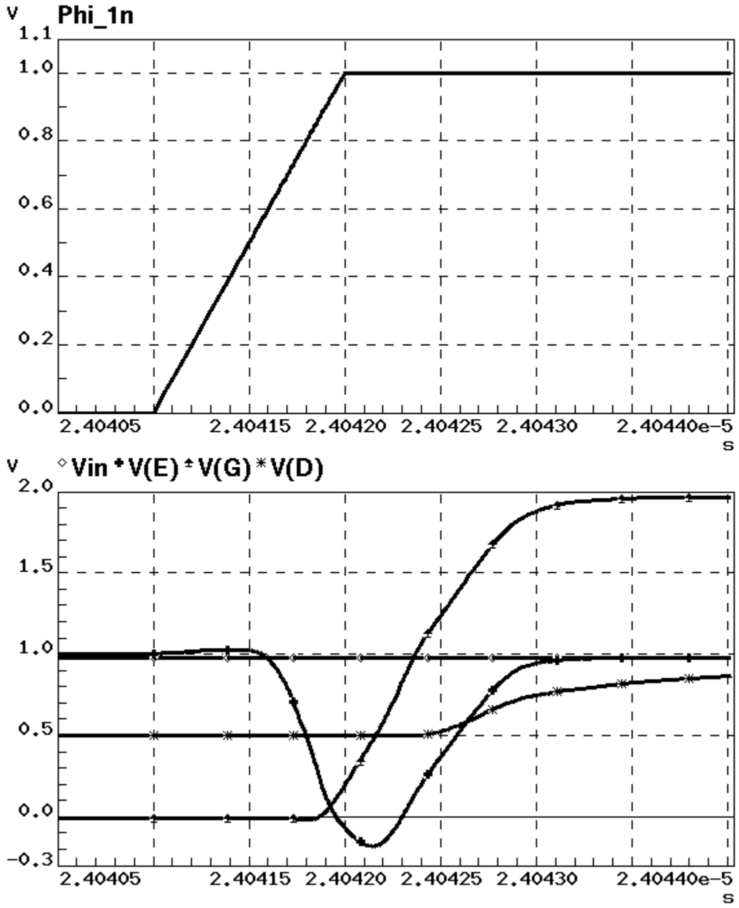


Figure 7.8: *Simulation of the turn on transition of the bootstrapped switch for a V_{DD} of 1V.*

$2V_{DD}$. Typically a CMOS technology is designed such that the reverse breakdown of a stand-alone n+/p- junction is approximately $3V_{DD}$ [Abo99b]. However, for an n-transistor, an n+/p+ junction is formed between the drain (or source) and the p+ field implant[1]. This junction has a lower breakdown voltage. In [Abo99b], a circular MOS layout with the drain at the middle is used such that the channel-stop implant is completely removed around the drain. In addition, the lightly-doped drain region can also be extended into the drain, this will increase the series drain resistance but will also increase the drain break-down voltage. By combining the circular drain layout and the extended lightly-doped drain, the drain break-down voltage can typically be increased 2-4V [Abo99b].

Fig. 7.9 demonstrates the operation of the bootstrapped switch S1[2] used in the low-pass section shown in Fig. 7.5 for a V_{DD} of 1V. The first graph shows that the switch conducts the input signal from rail-to-rail. In the second graph, the voltages on nodes A and B around the offset capacitor

[1]Surface concentration in areas which are *not* active devices (called field regions) is increased to properly isolate active devices.

[2]Transistor MNSW in Fig. 7.7 corresponds to transistor S1 in this case.

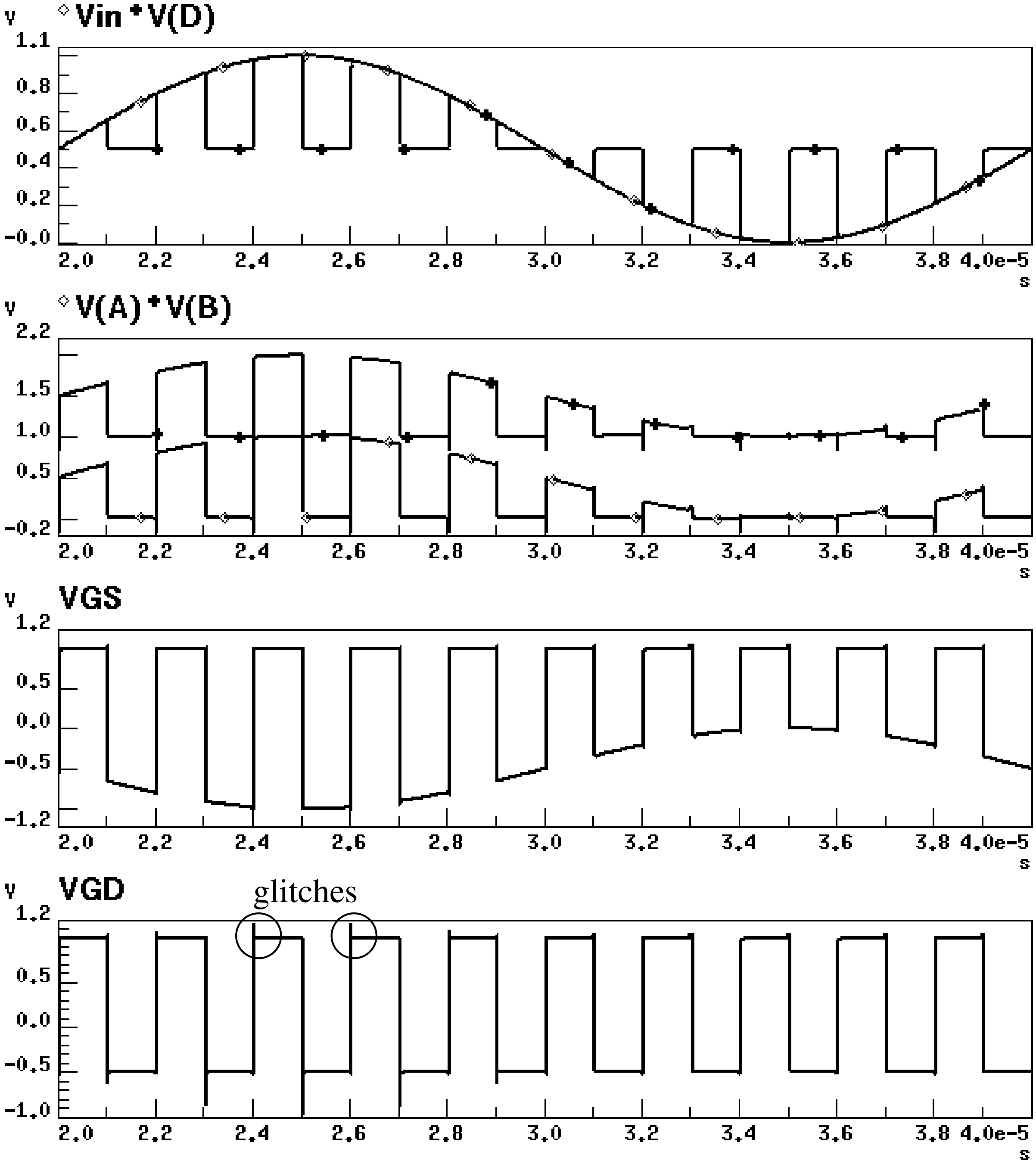


Figure 7.9: *Simulation of the bootstrapped switch S1 circuit operation in Fig. 7.5 for a V_{DD} of 1V.*

are shown, their difference is limited to V_{DD}. The third and the forth graphs shows the voltage difference V_{GS} and V_{GD} of transistor MNSW. It is clear that both are limited to a maximum of V_{DD}. The reliability problem on the drain side discussed in section 7.3.2 can be seen on the V_{GD} curve in the first half of the input signal where the source potential (equal to the input signal)

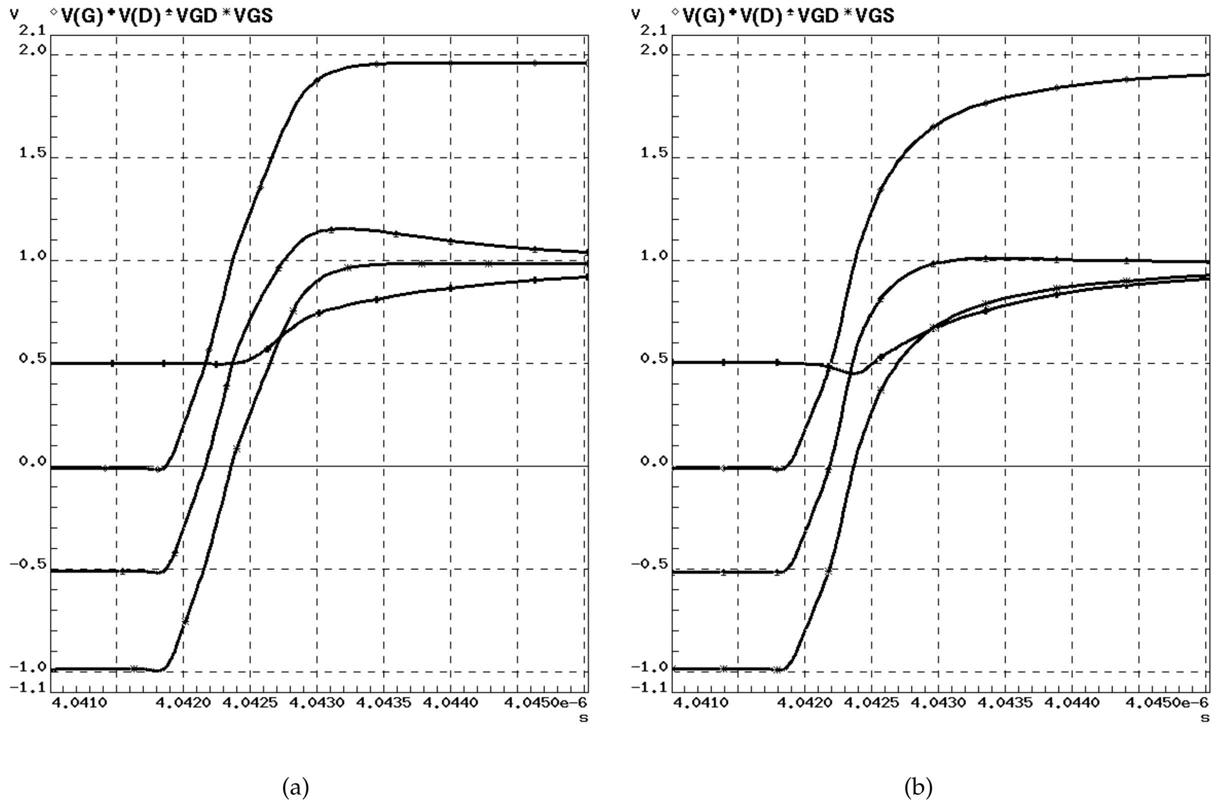
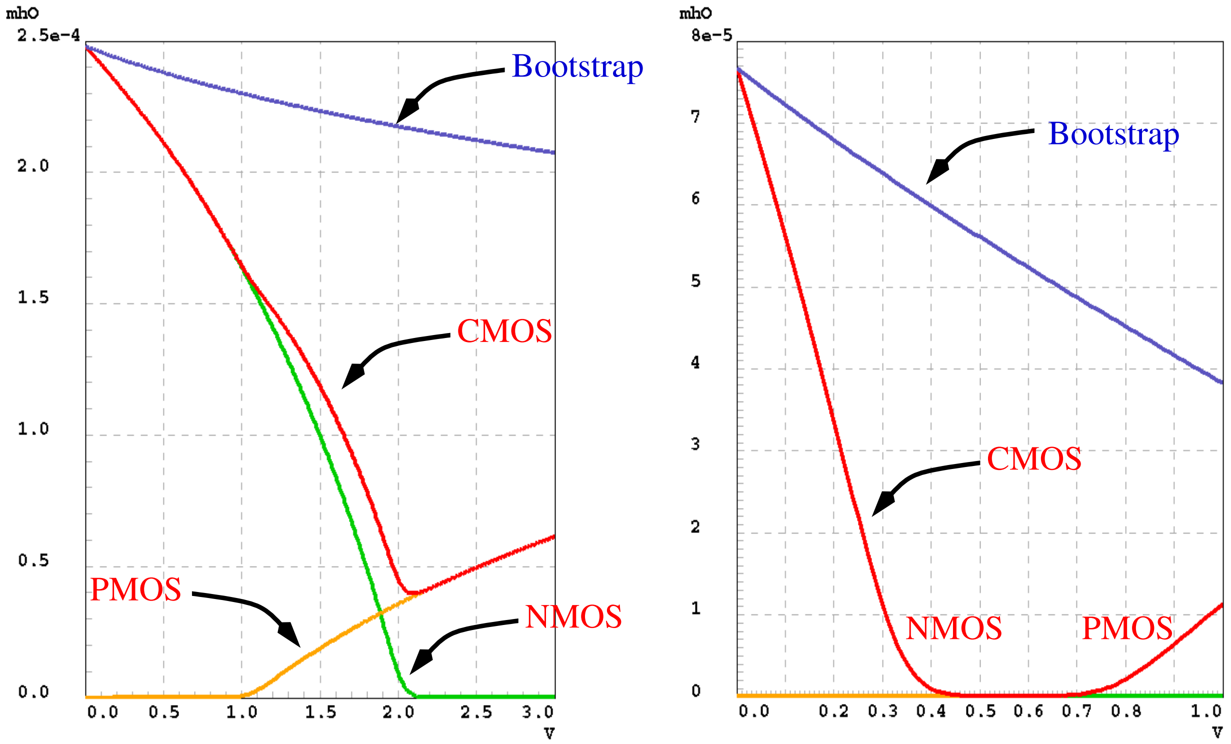


Figure 7.10: *(a) Gate-drain oxide transition overstress simulation and (b) same simulation with the MN8 transistor and* $V_{GD} < 1V$.

is higher than the drain one (discharged to $V_{DD}/2$ during ϕ_2) at the beginning of the transition. This results in a positive glitch that exceeds V_{DD}. This transition glitch has been magnified in Fig. 7.10-a which shows the simulation results of the potential at nodes G and D as well as the voltage difference V_{GS} and V_{GD} at the beginning of the transition. In order to remove this glitch, an additional transistor MN8, shown dashed in Fig. 7.7, has been added on the drain side, such that the switch MNSW becomes completely symmetrical. The gate voltage is thus clamped to a voltage V_{DD} higher than the terminal of the lowest voltage. This is depicted by the simulation results shown in Fig. 7.10-b which shows the same plots of Fig. 7.10-a after the addition of the extra transistor. It can be seen that now the gate-drain voltage is also limited to V_{DD}.

This bootstrapping circuit, thus, allows switch operation (transistor MNSW) from rail-to-rail while limiting all gate-source/drain voltages to V_{DD} avoiding any oxide overstress. The switch also guarantees maximum conductance independent of the input signal thus enhancing considerably the switch linearity. Fig. 7.11 shows the bootstrapped switch conductance versus the source potential for two different supply voltages. Also shown is the conductance of a CMOS switch. For the 3-V case (Fig. 7.11-a), the switch conductance has less variations than that of the CMOS



(a) (b)

Figure 7.11: *Comparison of the bootstrapped and the CMOS switch conductance vs. the source potential for a V_{DD} of (a) 3V and (b) 1V using minimal size transistors W/L=*$0.5\mu/0.35\mu$.

case. This means that using this switch would reduce significantly harmonic distortion effects related to non-linear switch conductance and charge injection. For the 1-V case (Fig. 7.11-b), while the CMOS switch fails due to the conductance gap at the middle, the bootstrapped switch allows rail-to-rail operation. In spite of the fact that the gate-source potential is held constant for the bootstrapped switch, the conductance drops with the source voltage due to the source-bulk potential which increases the threshold voltage. In [Steensgaard99], a solution to this bulk effect is proposed through the use of a separate well p-transistor as the main switch and controlling its bulk potential.

The bottom plate of the C_{offset} capacitor is always connected to the source side S, as shown in Fig. 7.7, to reduce voltage reduction on the gate G due to capacitance devision, see section 8.5.3.3. This adds an extra parasitic loading capacitance to the SC circuit.

It is to be mentioned that other bootstrapped switch implementations have been proposed. In [Sauer96], a MOS only implementation was presented, however, no attention has been paid to reliability problems. In [Abo99a], [Abo99b] reliability problems have been addressed, however, the bootstrap implementation presented in this work has the merit of being much simpler and to address the transient reliability problem.

7.5 Low-voltage Opamp

Reducing the supply voltage puts more constraints on the design of the amplifier. Since reduced supply voltage forces the power consumption to increase [Sansen98], the amplifier topology plays a critical role in low-voltage, low-power SC design. Since differential structures are often employed, additional circuitry for output CM stabilization must also be considered during the design so as not to degrade the overall amplifier performance.

In [Baschirotto97b] a 1V two-stage amplifier is designed for the SO technique. It is based on a p-type folded-cascode two stage Miller-compensated structure. An additional CM amplifier, performing the necessary signal inversion, is used in the CMFB circuitry. The overall opamp operates at a minimum supply voltage of $V_{GS} + V_{DSsat}$, but speed and power consumption are both limited by the additional CMFB amplifier. In [Waltari98], the same opamp is used but with the CMFB in the first stage implemented using a cross-coupled transistor stage. CMFB in the second stage is achieved using a simple passive circuit suitable only for SO circuits. The minimum supply voltage needed is however increased by one V_{DSsat}.

This section describes a differential very low-voltage opamp which incorporates further modifications to the above cross-coupled connection so as to reduce the minimum supply voltage. Bootstrapped switches described in the section 7.4 allow a simple SC CMFB circuit to be used. Two compensation schemes are considered and compared with respect to the amplifier performance. As the signal level is reduced for low supply voltages, the noise level becomes more critical. Special noise reduction techniques are also discussed for the modified architecture.

7.5.1 Opamp Structure

For practical SC circuits, the opamp has to be differential. Fully differential circuitry has superior power supply noise rejection, as compared to single-ended designs, and also provides twice the output swing for a given supply voltage. In addition, the symmetry of a fully differential circuit leads to the cancellation of even-order distortion components, regardless of their cause [Brandt96]. The only limitation of differential amplifiers is the accompanying CMFB circuitry that must be considered in parallel with the amplifier design. Fig. 7.12 shows a fully differential SC integrator using the double reference technique described in section 7.3.2.

In very low-voltage SC circuits, stacked transistors at the amplifier output can't be used to achieve a high DC gain. Thus usually a multi-stage amplifier is needed [Hogervorst96].

In SC circuits, the opamp input voltage is fixed at a given CM voltage, thus relaxing the requirements on the CM input range of the amplifier. In this case, it is fixed at V_{SS} as shown in Fig. 7.12. An input PMOS differential pair allows the use of V_{SS} as the opamp CM input voltage

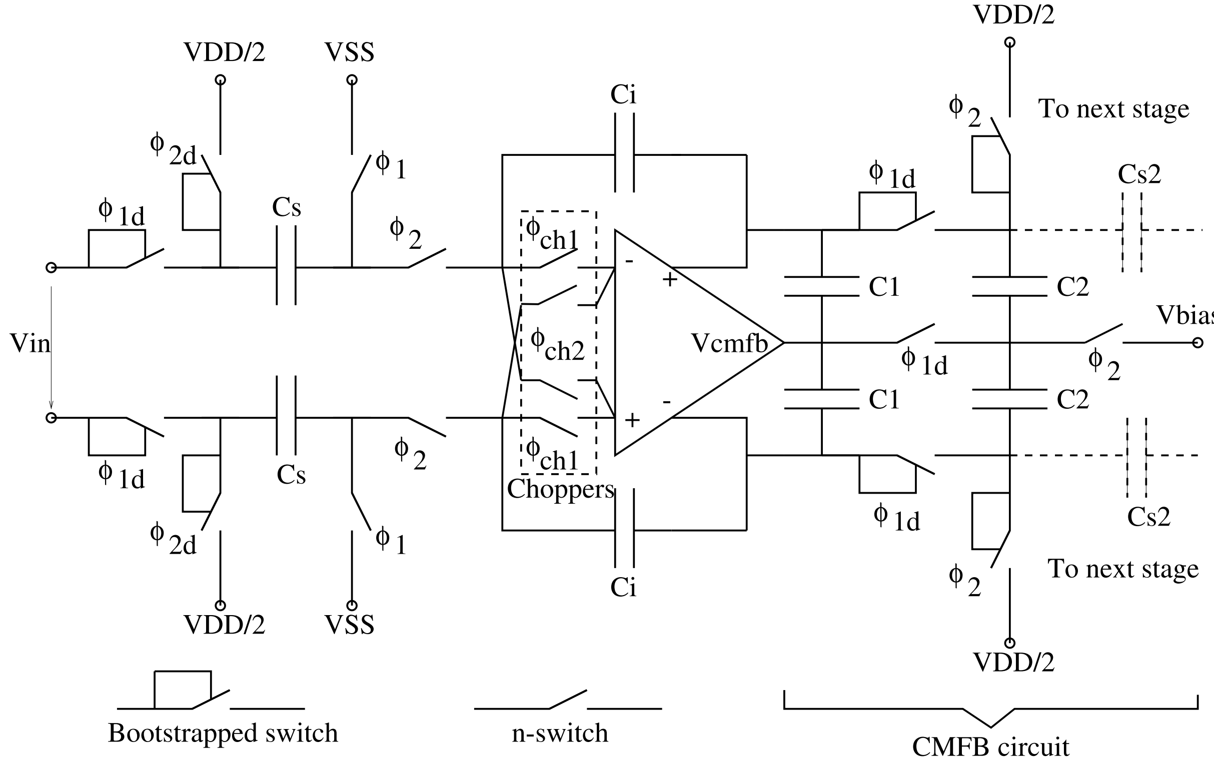


Figure 7.12: *Fully differential integrator using bootstrapped switches.*

V_{cm-in}. The output CM voltage of the opamp is fixed to $V_{DD}/2$ to maximize the available signal swing as described in section 7.3.

Based on the above discussion, the basic opamp structure is shown in Fig. 7.13. It is based on a two stage amplifier with the first stage (M1-M9) folded to adjust the quiescent output voltage. The very low supply voltage allows the cascode transistors M3 and M4 to be biased through V_{DD}, which reduces the number of needed biasing voltages. However, this causes the V_{DD} noise to be injected into the signal path. This effect is greatly reduced through the fully differential structure where this noise results in a CM signal which is rejected by the fully differential operation. The amount of the cancellation is limited by the mismatch of the two differential paths.

The second stage (M10-M13) is a common-source amplifier with active load which allows a large output swing.

7.5.2 Common-Mode Feedback

Due to the differential structure of the two-stage amplifier, the CM output voltage of *both* stages needs to be regulated using CMFB. Biasing of class-A amplifiers is typically accomplished with a CMFB circuit that senses the output CM voltage in order to control the tail current source via a current mirror in the first stage. However, owing to stability considerations, the gain and bandwidth

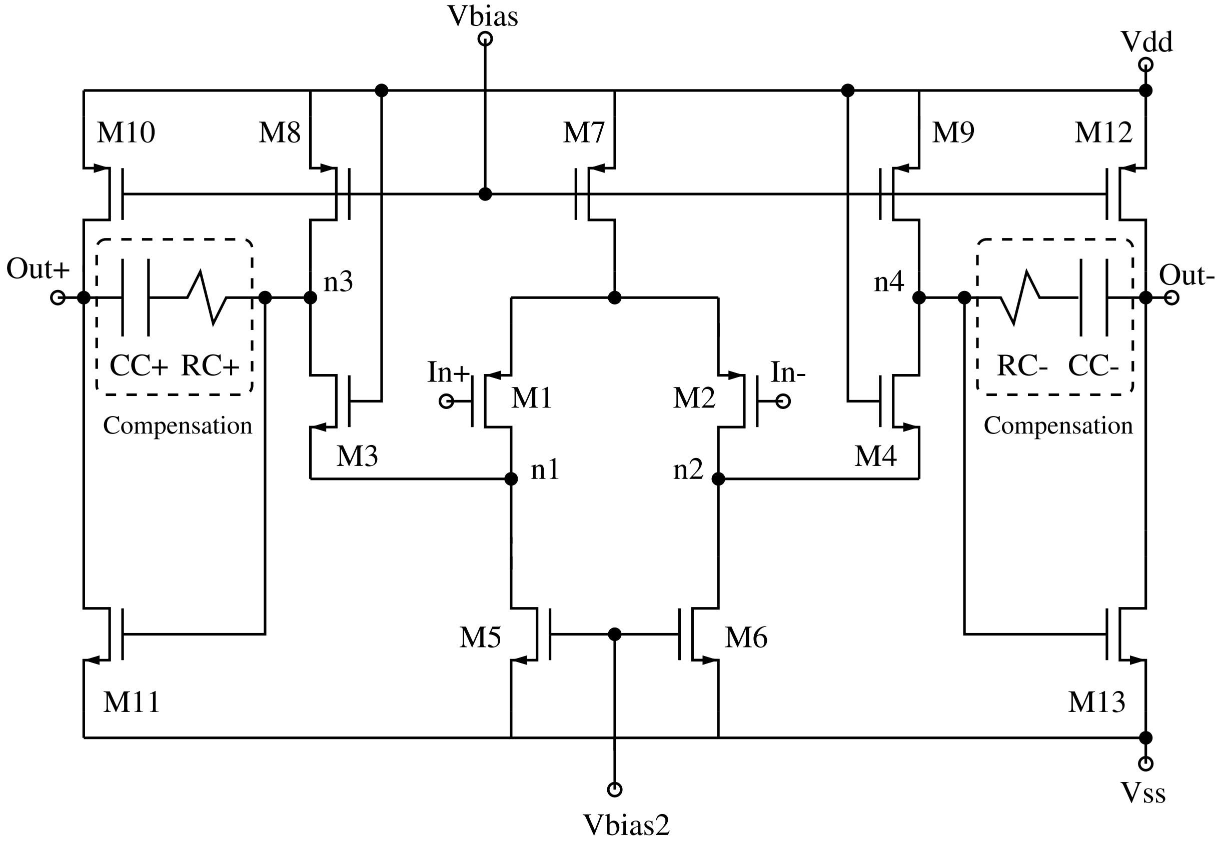


Figure 7.13: *Basic opamp structure with Miller compensation.*

of the CMFB loop are limited to at most those of the differential mode signal path. Moreover, since the gain from the first stage to the output is positive, an additional inverting amplifier is needed to achieve a stable CMFB, increasing power consumption. This CMFB amplifier also limits the output swing of the original differential amplifier. In this work, CMFB of each stage is handled separately.

Fig. 7.14 shows a technique to eliminate this additional CMFB amplifier. The NMOS current source M5(M6) has been split into two equally-sized, cross-coupled devices (M51, M52 and M61, M62) with their gates connected to the two outputs of the first stage (nodes $n3$ and $n4$). This negative feedback connection causes the differential signal at the output of the first stage (nodes $n3$ and $n4$) to see a high load impedance given by the reciprocal of

$$g_{dsout1} = g_{ds8} + (g_{m51} - g_{m52}) + \frac{g_{ds3}(g_{ds1} + 2g_{ds51})}{g_{ds3} + g_{ds1} + 2g_{ds51} + g_{m3} + g_{mb3}} \tag{7.5}$$

The conductance g_{m51} seen at the gate of transistors M51 and M61 is thus canceled by the opposite action of the parallel transistors M52 and M62 respectively. Proper matching of these transistors, together with other terms in equation (7.5) prevent the output resistance from going negative. The total conductance g_{dsout1} is thus limited by g_{ds8}.

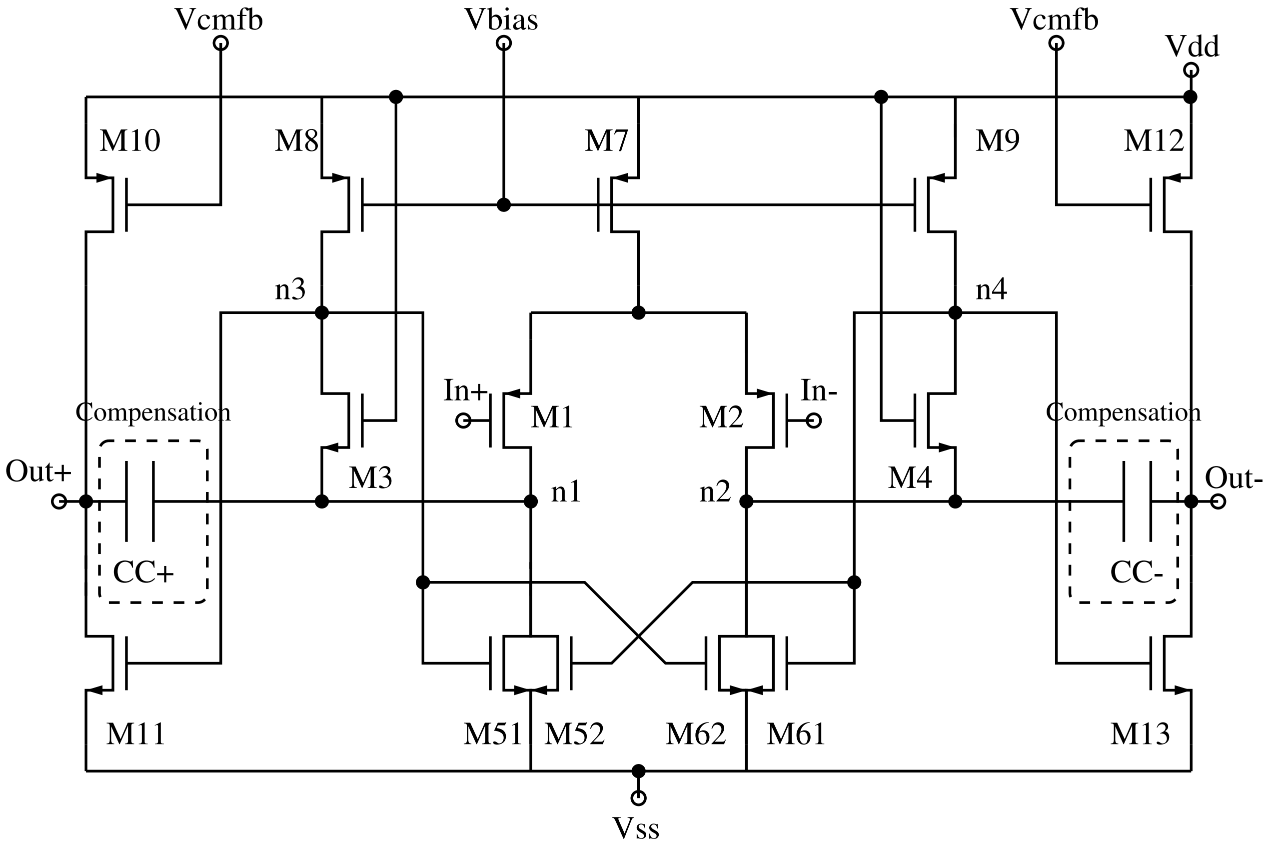


Figure 7.14: *Modified opamp schematic with cascode compensation.*

On the other hand, for the CM signal, the output conductance is also given by equation (7.5) but with the negative term turned positive. The total conductance g_{dsout1} in this case is limited by $g_{m51} + g_{m52}$. This impedance is a low one and thus the first stage does not require an additional CMFB circuit. In fact the cross-coupled devices act like a built-in CMFB circuit that senses the CM output of the first stage, averages it through the parallel transistors M51/M52 (M61/M62) and regulates its CM voltage through the biasing current. This connection also allows a minimum supply voltage of $V_{GS} + V_{DSsat}$.

The second stage is composed of the NMOS common source amplifier M11(M13) with active load M10(M12). A simple passive SC CMFB circuit [Castello85], shown in Fig. 7.12, can be used in this case. The DC voltage across C_1 is determined by capacitor C_2 which is switched between being in parallel with C_1 and $V_{cm-out} - V_{bias}$, where V_{bias} is the desired biasing voltage for the current source p-transistors M10 and M12 as shown in Fig. 7.13. Since the potential V_{bias} is close to V_{SS}, n-transistors can be used to switch it. However, bootstrapped switches must be used in the CMFB circuit for those switches that have to switch the V_{cm-out} potential. These bootstrapped switches can be shared with the sampling network connecting the integrator output to subsequent stages as shown in Fig. 7.12, such that only four bootstrapped switches are needed per differential integrator stage.

7.5.3 Opamp Compensation

Two possible compensation schemes are possible for this two-stage opamp structure: The first one is the standard Miller compensation scheme shown in Fig. 7.13, which consists of connecting the compensation capacitor C_C in series with a nulling resistance R_C between the output nodes and the outputs of the first stage (nodes $n3$ and $n4$). Analysis of the amplifier shows that the transfer function has five poles and two zeros that can be placed in the left half-plane. A sort of pole-zero cancellation is also possible by properly choosing the value of R_C. This further enhances the PM. The second compensation scheme is shown in Fig. 7.14. This is done by connecting the compensation capacitor C_C to the source of the cascode devices (nodes $n1$ and $n2$) [Ribner84]. These low impedance points decouple the gate of the output stage amplifier (transistors M11 and M13) from the compensation capacitor. This technique offers a much improved high-frequency PSRR and moves the right half-plane zero resulting from Miller compensation into high frequencies. It can be shown that this type of compensation results in two complex poles besides the dominant one [Ribner84]. It is thus quite possible to obtain a design with adequate PM, which suffers from insufficient gain margin due to gain peaking beyond the unity-gain bandwidth frequency, caused by a high pole quality factor Q_p. This pole quality factor Q_p is given by [Ribner84]

$$Q_p \approx \left[\frac{g_{m11}C_L}{(g_{m3}+g_{mb3})C_{n3}}\right]^{1/2} \frac{C_C}{C_L+C_C} \tag{7.6}$$

It can be kept low by making the transconductances of the cascode transistor M3(M4) large compared to the output driver M11(M13). In addition a moderate value of C_C is required.

It should be noted that a right half-plane zero exists in this configuration, but its value is high compared to the unity-gain frequency and does not degrade the PM.

Both schemes ensure stability. Low-Q_p considerations for the second one usually impose more constraints on the design of the amplifier. High cascode transconductance implies either lower $V_{GS} - V_{th}$ or higher current. Lower $V_{GS} - V_{th}$ leads to higher parasitic capacitance which will reduce the amplifier bandwidth. This implies an optimum $V_{GS} - V_{th}$ value. Higher cascode current implies larger power consumption and higher input referred thermal noise, see section 8.5.2.2. In addition, the value of the compensation capacitance C_C in the second case is limited by the required Q_p. Since the aliased input referred in-band white noise in SC circuits is inversely proportional to the value of the compensation capacitor of the amplifier, see section 8.5.2.1, this restricts the white noise performance optimization.

7.5.4 Noise Reduction

For low-noise input front-ends, the input amplifier noise optimization is an important step in the overall system design. Thermal noise can be reduced using higher input current in the input differential pair. On the other hand, flicker (or 1/f) noise can be reduced using larger areas for those transistors contributing to flicker noise (namely M1, M2, M5, M6, M8 and M9). This causes higher parasitic capacitance on the internal nodes and thus increases the amplifier power consumption.

Techniques such as chopper stabilization [Hsieh81] can be used to get rid of the 1/f noise: The signal at the amplifier input is modulated to a certain chopper frequency (usually at half the sampling frequency) separating it from the low frequency 1/f noise. At the amplifier output, the signal is demodulated restoring the input signal and moving the 1/f noise to around the chopper frequency. Input and output modulation can be easily done using four switches on both sides as shown in Fig. 7.15. ϕ_{ch1} and ϕ_{ch2} are non-overlapping clock phases used to drive the switches. This is equivalent to multiplying the input and output by a stream of 1 and -1 performing the necessary modulation/demodulation. Since the input CM voltage is at V_{SS}, input chopper switches can be implemented using n-switches as shown in Fig. 7.12. However, since the used opamp is a two-stage amplifier with a compensation capacitor which acts like a memory element, the outputs can't be switched instantaneously[3].

A solution to the output chopping problem is shown in Fig. 7.16. The output of only the *first* stage is chopped using two additional cascode transistors M32 and M42 in parallel with the existing ones, but with their sources connected to nodes $n2$ and $n1$ respectively. The gates of both cascodes are then driven by two *overlapping* chopper clocks (ϕ_{ch1} and ϕ_{ch2}) at half the sampling frequency. The two chopper clocks must *overlap*, see Fig. 9.2, to avoid the simultaneous cutoff of both cascodes in parallel which would increase the settling time of the opamp.

This arrangement reduces the $1/f$ noise for all transistors but M8 and M9 where a larger transistor length must be used. In addition, it can not be used with the cascode compensation scheme shown in Fig. 7.14 since the compensation capacitor is connected inside the first stage.

It should be noted that the input chopper switches, shown in Fig. 7.12, create an additional pole together with the input capacitance of the opamp. This pole must be considered during the amplifier design.

7.5.5 Simulation Results

As an example of the proposed architecture, two schematics have been sized. The first using the cascode compensation scheme, and the second using the Miller scheme with a nulling resistor to compensate for the right half-plane zero. Both circuits are sized for the same supply voltage, unity-gain frequency, PM, SR, and output voltage range in order to be able to compare them. A 0.35-μ technology is used with a p- and n-transistor thresholds of 0.63 V and 0.6 V respectively. Table 7.1 shows the obtained transistor sizes, and table 7.2 shows the simulated results of the two sized netlists.

[3]It should be noted that even if it were possible to use the output switching arrangement shown in Fig. 7.15 (like in the case of a single stage amplifier), bootstrapped switches should have been used since the output CM voltage is at $V_{DD}/2$

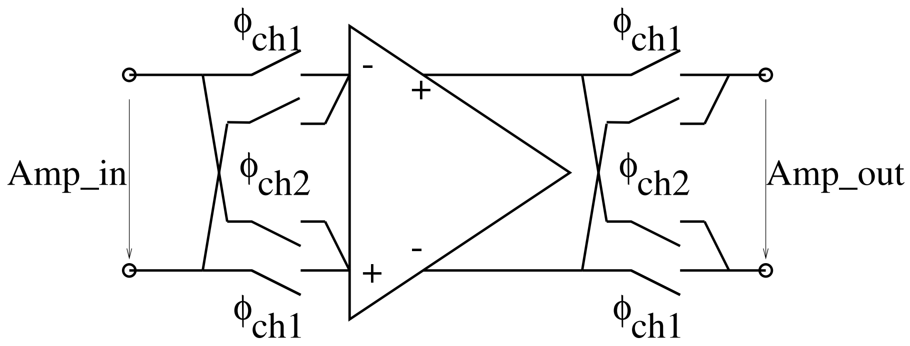


Figure 7.15: *Chopper stabilization, circuit implementation.*

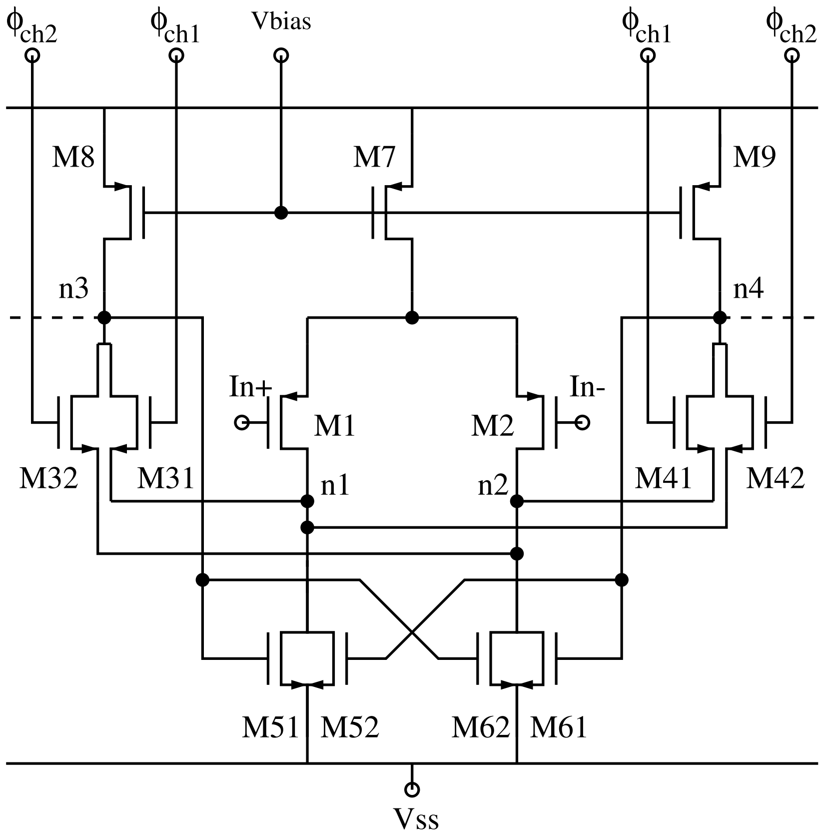


Figure 7.16: *Output chopping using the cascode transistors.*

Transistor	Cascode compensation	Miller compensation
M1, M2	15.8/1.4	43.6/2.3
M3, M4	45.2/1.0	28.6/1.0
M51(2), M61(2)	5.4/1.4	10.6/2.3
M7	115.8/0.35	250.8/0.4
M8, M9	96.5/0.35	125.4/0.4
M10, M12	41.6/0.5	45.1/0.5
M11, M13	13.0/0.6	14.8/0.6

Table 7.1: *Opamp sizes W/L in μm for the cascode and Miller compensation schemes.*

Performance	Simulated Value
V_{DD}	1 V
GBW	10 MHz
PM	75 ^{o}
SR	7.0 V/μs
Load Capacitance (C_L)	5 pF
Compensation Capacitance (C_C)	1.4(2.3) pF
CMFB Loading Capacitance (C_1)	1.0 pF
DC gain	63 (70) dB
Pole Quality factor (Q_p)	1.0
Power Dissipation	175 (213) μW
Total Input Thermal Noise	122 (84) μV
Max. output voltage	0.82 V
Min. output voltage	0.15 V

Table 7.2: *Opamp simulation results for the cascode and Miller (between brackets) compensation schemes.*

For the cascode compensation scheme, a lower compensation capacitance value could be used, this reduces the overall power consumption. However, due to low-Q_p considerations it is more difficult to obtain a satisfactory gain. The high-frequency PSRR is also better for cascode compensation. Careful layout can further enhance the PSRR performance for both cases as supply noise is considered as a CM signal and is cancelled at the differential output of the amplifier.

Additional transistors for chopper stabilization are taken into account during sizing. Thus, flicker noise can be neglected in case of the Miller compensation case.

Fig. 7.17 shows the simulated open-loop gain for both cases, gain peaking can be easily identified for the cascode compensation case. Special care has been taken during sizing such that the

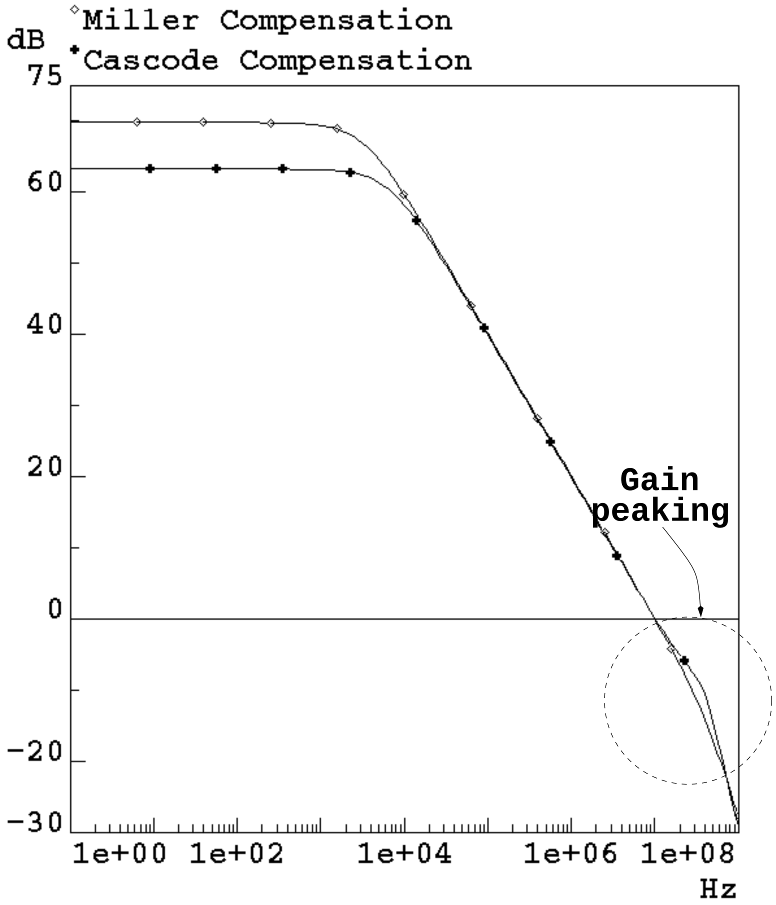


Figure 7.17: *Simulated Open-loop Gain for the two compensation schemes.*

complex pole quality factor Q_p does not exceed unity.

The Miller compensation scheme has been retained for circuit implementation in this work as it offers a good compromise between performance and design optimization complexity. In addition it allows to use the chopper stabilization scheme described in section 7.5.4 for 1/f noise reduction.

7.6 Conclusions

In this chapter, low-voltage low-power SC circuits have been chosen as an application to validate the proposed design tools. This choice has been justified both on the application level by the increasing demand on this type of circuits, as well as on the design level by the need to accurately estimate the layout parasitics in order to optimize power consumption.

Thus. the problem of very low-voltage SC design has been tackled from the designer point of view. First by studying circuit limitations as well as the corresponding existing solutions.

Two SC configurations have been then proposed to allow very low-voltage operation. Both are based on a special bootstrapped switch which allows rail-to-rail signal switching. Key advantages of the proposed circuit is its simplicity, a significant increase in signal-to-noise ratio while the extra power and area requirements are very modest. It is fully compatible with advanced low-voltage CMOS as all gate-source and gate-drain voltages are limited to V_{DD} thus preventing gate-oxide overstress. This low voltage switch also preserves a nearly constant switch conductance, leading to the reduction of harmonic distortion.

A modified opamp architecture was then presented. The proposed fully differential opamp allows very low supply voltage operation and minimizes the additional CMFB circuitry thus reducing overall power consumption. Conventional Miller and cascode compensation schemes are compared using a design example. Minor modifications allow the chopper-stabilization technique to be used for noise reduction.

Having now the necessary elements, the implementation of a very low voltage Delta-Sigma A/D modulator is going to be described in the next chapters.

Chapter 8

Design of a Very Low-voltage Delta-Sigma Modulator

8.1 Introduction

As a case study of the SC very low-voltage operation problem studied in the previous chapter, this chapter presents the design process of a $\Delta\Sigma$ modulator based on the developed circuit techniques.

In section 8.2, design steps and tools used in each phase are presented. It should be noted that these steps are usually common in the design of any analog or mixed-signal system, refer to section 3.2.

In section 8.3, the performance goal of the design is first fixed. The modulator architecture is then studied and the corresponding coefficients are finally determined.

In section 8.4, the effect of cell non-idealities on the overall modulator performance are investigated. This also leads to mapping the global system specifications to building block specifications.

In section 8.5, starting from cell specifications, the circuit level of each cell is synthesized. This section also includes a detailed analytical study of these cells in order to be incorporated in the knowledge-based sizing tool COMDIAC presented in chapter 6.

Finally the chapter ends with some concluding remarks.

8.2 Methodology and Tools

The modulator design flow is shown in Fig. 8.1. The main design goals of an ADC is the signal-to-noise ratio and the input signal bandwidth. Noise is mainly due to signal quantization in addition to noise added by circuit components. Quantization noise is limited by the modulator architecture chosen during high-level synthesis and is often measured by the SQNR. On the other hand, circuit noise depends directly on the circuit implementation. The modulator design process contains four major steps:

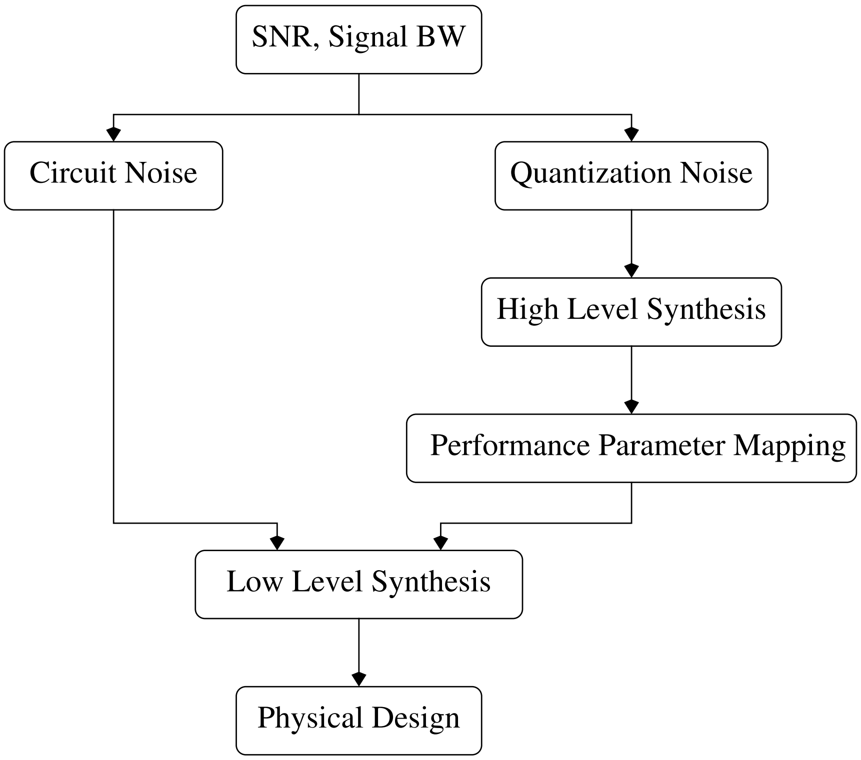


Figure 8.1: *Design flow*.

1. **High-Level Synthesis**:
   Starting from the performance goal, the most suitable modulator architecture and oversampling ratio OSR are chosen. The internal modulator coefficients are then determined. Owing to the nature of $\Delta\Sigma$ modulation, it is difficult to describe analytically its operation and guarantee its stability. Thus, during this phase usually a large number of simulations are done on the functional level. This is the most abstract modeling level where ideal models are used for the building blocks. Connection points indicate a transfer of information as in a signal-flow model. High-level simulations are performed using the MATLAB [Mat97] software.

2. **Performance Parameter Mapping**:
   Now that the architecture has been fixed, models that describe the non-ideal behavior of the modulator building blocks are built and used to investigate the feasibility of the chosen architecture on the circuit level. This also leads to performance parameter mapping from the system level to the building block's transistor level.

3. **Low-Level Synthesis**:
   In this step each block is handled separately and is designed according to the performance specifications determined in the previous step. Synthesis of the building blocks are done using the CAD tools and methodology described in chapter 4. For each block:

- The complete design procedure is incorporated in the knowledge-based sizing tool COMDIAC described in chapter 6. Hierarchical sizing facilitates this step by re-using the existing circuit building blocks such as differential pairs and OTA's.
- The layout code is written using the layout language CAIRO described in chapter 5. The code is independent of transistor sizes and the used technology. A parasitics calculation mode allows layout parasitics to be taken into account during sizing.

4. **Physical Design**:
   Where the complete layout is generated physically using CAIRO. Each block previously described is instantiated in the higher block until the complete layout is constructed. The same code can be used either in the parasitics calculation mode or in the layout generation mode.

8.3 High-Level Synthesis

8.3.1 Performance Goal

The goal of this circuit is to achieve very low-voltage ($V_{DD} = 1V$), low-power operation of a high resolution $\Delta\Sigma$ modulator (around 14 bits) for a digital-audio signal (with a bandwidth of 16 kHz) in a standard CMOS technology.

Topology For low-voltage low-power applications, a single loop $\Delta\Sigma$ modulator topology is preferable over a cascaded one because it has more relaxed requirements on linear amplifier non-idealities, such as the DC gain and the gain-bandwidth product. Also, since noise injected at the internal nodes is reduced so much by the large gain of the preceding integrators, integrators inside the feedback loop can be scaled down resulting in a lower power dissipation [Peluso98b].

OSR and Order The noise transfer function (NTF) of a single loop $\Delta\Sigma$ modulator is given by [Adams96]

$$NTF(z) = \left(1 - z^{-1}\right)^n \tag{8.1}$$

where n is the modulator order. Fig. 8.2(a) shows the theoretical SQNR of a modulator having a NTF given by equation (8.1) vs. the OSR defined by

$$OSR = \frac{f_s}{2f_m} \tag{8.2}$$

where f_s is the sampling frequency and f_m is the signal BW. However, due to stability problems, the practical achievable SQNR is much lower than that predicted by Fig. 8.2(a). Fig. 8.2(b) shows the maximum achievable practical SQNR values taking into consideration stability of the modulator [Adams96]. From Fig. 8.2(a) it is obvious that in order to increase the SQNR one should

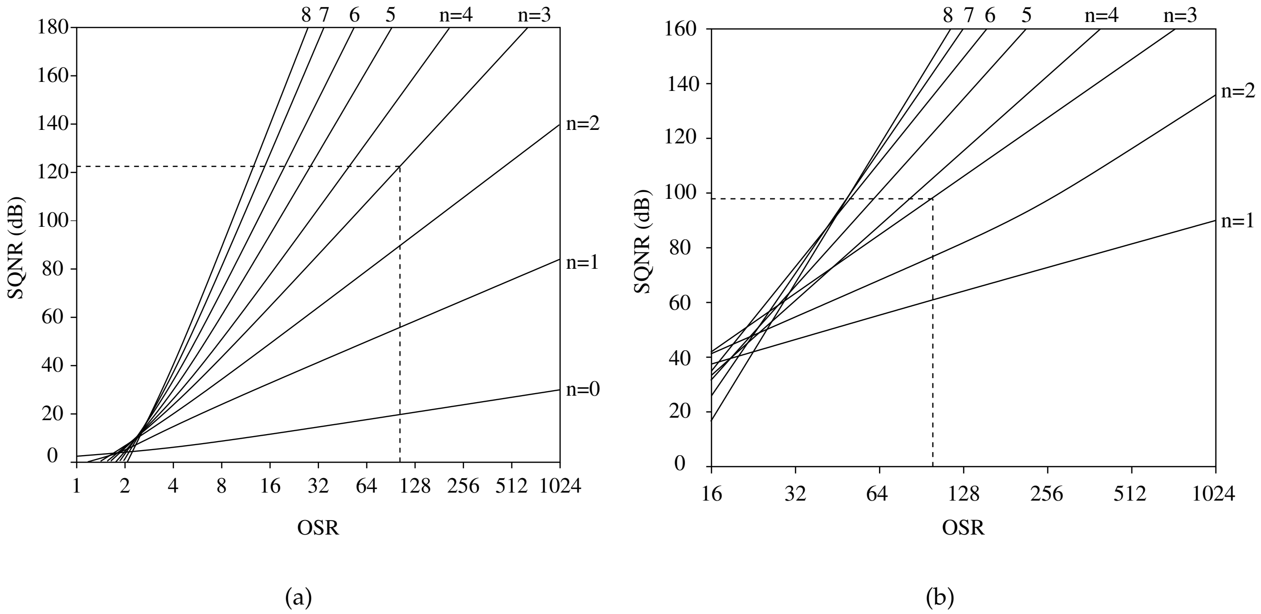


Figure 8.2: *(a) Theoretical SQNR for a modulator having a NTF given by equation (8.1) and (b) the maximum achievable SQNR vs. the oversampling ratio [Adams96].*

increase either the modulator order n or the OSR. Increasing the OSR, however, requires faster settling time for the integrators, i.e. higher amplifier SR and GBW, which means higher power consumption. Thus, in order to keep the oversampling ratio relatively low, the loop order must be increased. In spite of the fact that this means additional integrators, i.e. additional power consumption, the overall power consumption is reduced due to integrator scaling [Peluso97]. Based on the above arguments and the data from Fig. 8.2(b), a *third*-order modulator with an oversampling ratio of 100 is thus chosen. It has a maximum achievable SQNR of around 98 dB.

8.3.2 Coefficient Determination

Fig. 8.3 shows the block diagram of the modulator. It is based on a one-bit chain of integrators with distributed feedback topology. Modulator coefficients have been determined with the help of the *Delta-Sigma Toolbox* [Schreier] for MATLAB [Mat97], according to the design procedure described in [Adams96].

Noise transfer function The first step is to determine the loop NTF. A key parameter in the NTF design is its out-of-band gain (NTF_{inf}). Increasing NTF_{inf} would increase the achievable SQNR but would drive the modulator to the edge of instability for large inputs or small parameter shifts. The noise transfer function has been synthesized using different values of NTF_{inf}. The corresponding modulator coefficients were then determined for each case, followed by discrete-time

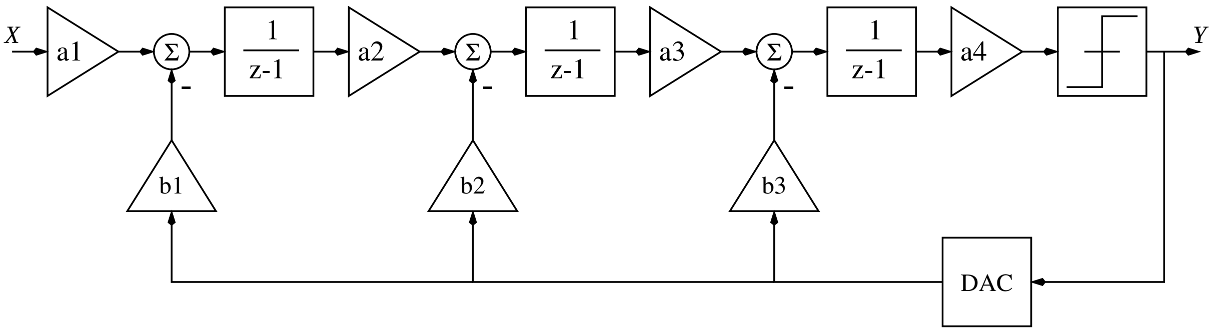


Figure 8.3: *Modulator topology*.

	Interstage Coeff.	Feedback Coeff.
First Integrator	a_1 = 0.10	b_1 = 0.10
Second Integrator	a_2 = 0.27	b_2 = 0.18
Third Integrator	a_3 = 0.31	b_3 = 0.17
Comparator	a_4 = 4.35	

Table 8.1: *Modulator Coefficients*.

simulations to measure the attained performance and stability range. Fig. 8.4 shows MATLAB simulation results showing the peak SQNR achieved for different NTF_{inf} values. Fig. 8.5 shows the corresponding maximum allowable input (U_{max}) that avoids modulator instability. It is quite apparent that while the peak SQNR increases with NTF_{inf}, the allowable input decreases. A trade-off between the peak SQNR and the maximum stable input range U_{max} thus exists. In this work, this trade-off was based on power consumption considerations as follows.

The total noise power is composed of quantization and circuit noise. For low-power implementations, the modulator noise performance should be limited by the circuit white thermal noise, as concluded in section 8.5.2.1. White noise can be reduced by increasing the input sampling capacitor and by reducing the opamp circuit noise, both leading to an increase in power consumption. As will be shown later, the first integrator is the major contributor to the overall power dissipation. In the same time, the first integrator gain (a_1) has a direct impact on its power consumption, due to two reasons: First, for a fixed sampling capacitance (fixed noise), reducing a_1 leads to an increase in the integration capacitance which increases the bottom plate parasitic capacitance at the opamp output and tends to increase power dissipation. Secondly, the amplifier input referred thermal noise is proportional to $(1 + 1/a_1)^2$ (see equation (8.34)). Since the maximum input signal power is proportional to the square of U_{max}, the peak SNR is then proportional to the ratio $(U_{max}/(1 + 1/a_1))^2$. Fig. 8.6 shows this ratio vs. NTF_{inf}. The fluctuations shown in this figure are due to the fact that while U_{max} decreases monotonically as shown in Fig. 8.5, a_1 must have rounded values for practical circuit implementations. Thus, in spite of the fact that a_1

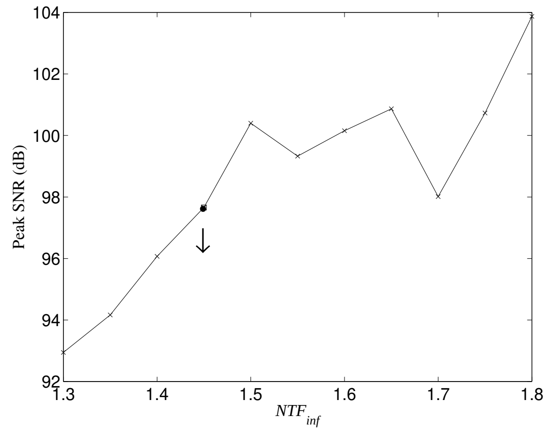


Figure 8.4: *Peak SQNR vs. the noise transfer function out-of-band gain.*

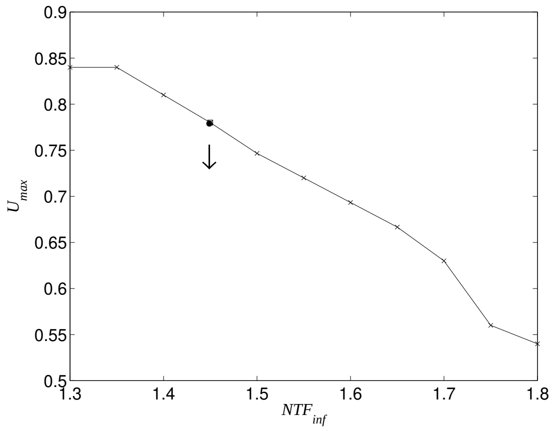


Figure 8.5: *Maximum allowable input vs. the noise transfer function out-of-band gain.*

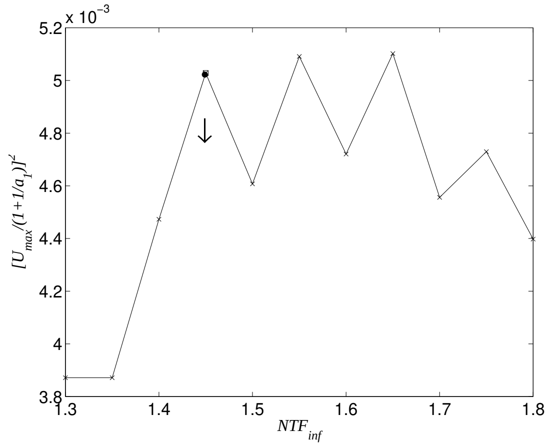


Figure 8.6: $(U_{max}/(1 + 1/a_1))^2$ *vs. the noise transfer function out-of-band gain.*

increases with NTF_{inf}, for some subsequent simulations shown in Fig. 8.6 a_1 has the same value due to rounding. A value of 1.45 for the out-of-band gain has been chosen as a compromise between thermal noise performance and modulator stability. The corresponding SQNR, U_{max}, and $(U_{max}/(1+1/a_1))^2$ ratio is shown with a square marker in Figs. 8.4, 8.5, and 8.6 respectively.

Modulator scaling and Reference voltage During high-level synthesis, it is necessary to perform dynamic-range scaling. This is done to ensure that all integrator outputs have approximately the same power level, so that all nodes will clip near the same level, and there will be no unnecessarily large noise gains from nodes with small signal levels [Johns97]. Maximum levels at integrator outputs are determined using discrete-time simulations. To increase the level of the output of integrator i by a factor k, the coefficients a_i and b_i are multiplied by k, while a_{i+1} is divided by k to keep the same transfer function. Since all voltage levels are normalized to the reference voltages, the output range of each integrator must lie between the positive and negative reference voltages. This means that, in the circuit implementation, these reference voltages are thus determined by the available output swing of the opamps.

Here, for simplicity reasons, the modulator reference voltages have been taken equal to V_{DD} and V_{SS} in order to avoid generating additional reference voltages on chip. In this case, modulator scaling should limit the integrator outputs to the linear output swing of the opamps (Vop_{swing}) which becomes a fraction of the reference voltages. As the limiting levels are reduced, the scaled modulator coefficients are reduced as well and so is a_1 which leads to an increase in the thermal noise level as discussed above. However, maximizing the reference voltages increases the maximum allowable input signal U_{max} which leads to an increase of the input signal power that compensates the previous increase in the noise level. Table 8.1 shows the obtained scaled modulator coefficients.

8.4 Performance Parameter Mapping

The above analysis assumes that all building blocks are ideal. Practically, the behavior of each block is usually accompanied with non-ideal effects related to the corresponding circuit implementation. These non-idealities lead to quantization noise leakage and degrade the overall signal-to-noise ratio. Before passing to the circuit design phase, it is mandatory to investigate whether the performance goal is satisfied in the presence of these non-idealities, and to what extent the modulator can tolerate their presence. This study also provides the mapping of high-level performance parameters to individual block performance parameters. Investigated non-idealities include:

- The finite gain of the amplifier used in the integrator.
- The frequency limitation of the amplifier.
- The SR of the amplifier.

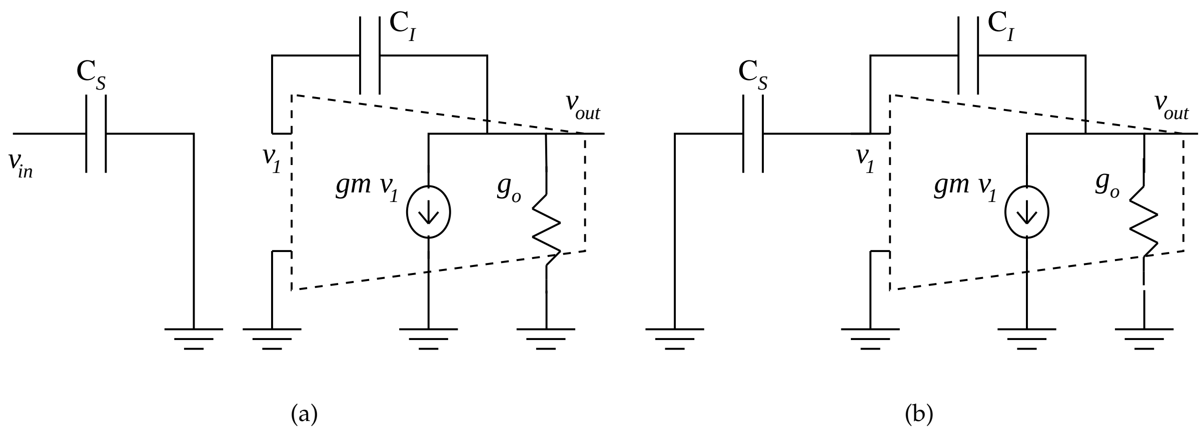


Figure 8.7: *Switched capacitor integrator using a simple one-pole opamp model: (a) Sampling and (b) Integration phases.*

- The comparator offset and hysteresis.
- The switch finite resistance.

The above effects are modeled and their effect on the overall modulator performance are studied using discrete time MATLAB [Mat97] simulations. Since many simulations are to be performed, a compromise must be done between the accuracy and speed of the developed models. The finite switch resistance is treated separately in section 8.5.3.

8.4.1 Opamp Finite Gain and Frequency Performance

An integrator using a simple one-pole amplifier model is shown in Fig. 8.7. The amplifier is modeled by an input-output transconductance g_m and a finite output conductance g_o. This model allows to account for the amplifier gain given by

$$A_{d0} = \frac{g_m}{g_o} \tag{8.3}$$

and for the amplifier frequency limitation caused by the dominant closed-loop pole resulting from the transconductance g_m and the load capacitance, C_S in this case.

According to [Marques99] [Geerts99], the transfer function of such integrator can be described by

$$H(z) = \frac{gz^{-1}}{1 - pz^{-1}} \tag{8.4}$$

where

$$g = \frac{C_S}{C_I}\rho_i(1-\theta_i) \tag{8.5}$$

$$p = \frac{\rho_i}{\rho_s}\left(1-\theta_i\left(1-\frac{\rho_s}{\rho_i}\right)\right) \tag{8.6}$$

where ρ_s and ρ_i are the closed-loop static errors during the sampling and integration phases respectively and are given by

$$\rho_s = \frac{A_{d0}\beta_s}{1+A_{d0}\beta_s} \tag{8.7}$$

$$\rho_i = \frac{A_{d0}\beta_i}{1+A_{d0}\beta_i} \tag{8.8}$$

where β_s and β_i are the corresponding feedback factors given by

$$\beta_s = 1 \tag{8.9}$$

$$\beta_i = \frac{C_I}{C_I+C_S} \tag{8.10}$$

The parameter θ_i represents the settling error in the integration phase. It is expressed by

$$\theta_i = exp\left(-\frac{g_m}{C_S}.\frac{t_i}{\rho_i}\right) \tag{8.11}$$

where t_i is the time available for integration. The factor g_m/C_S represents the closed loop dominant pole p_{CL} of the amplifier during the integration phase given by [Johns97]

$$p_{CL} = \beta_i\omega_t = \left(\frac{C_I}{C_I+C_S}\right)\left(\frac{g_m}{C_L}\right) = \left(\frac{C_I}{C_I+C_S}\right)\left(\frac{g_m(C_I+C_S)}{C_IC_S}\right) = \frac{g_m}{C_S} \tag{8.12}$$

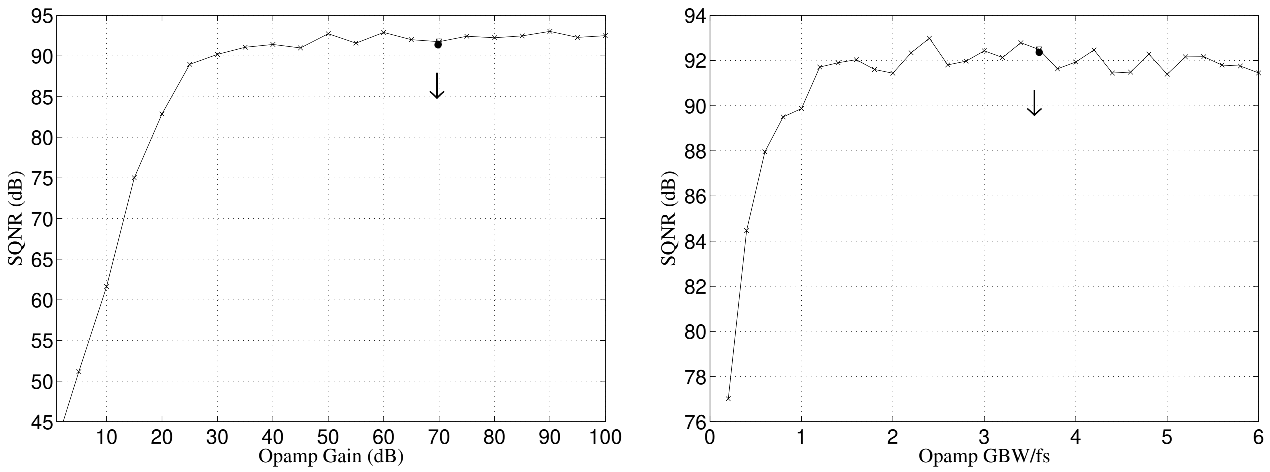
The above model does not include neither the input parasitic capacitance of the amplifier C_{ip} nor the parasitic output capacitance. Furthermore, the used amplifier is actually a two-stage one having its closed-loop pole determined by the internal compensation capacitance (C_C). However, if one tries to model these effects, the analysis becomes very complicated. An approach similar to [Geerts99] is followed: One can preserve the previous simple model and change only the most inaccurately modeled factors. Taking the effect of parasitic capacitances, the feedback factors become

$$\beta_s = \frac{C_I}{C_I+C_{ip}} \tag{8.13}$$

$$\beta_i = \frac{C_I}{C_I+C_{ip}+C_S} \tag{8.14}$$

The input opamp capacitance C_{ip} can be estimated given the opamp GBW (ω_t) as follows: First, C_{ip} can be approximated using the gate-source capacitance of the input transistors C_{gs1}. In saturation this capacitance is given by [Laker94]

$$C_{ip} = C_{gs1} = \frac{2}{3}W_1L_1C_{ox} \tag{8.15}$$



(a) (b)

Figure 8.8: *Discrete-time simulation results showing the SQNR vs. the (a) amplifier gain and (b) amplifier GBW/f_s.*

W_1 can be, in turn, calculated from the input transistor current using

$$I_{D1} = \frac{1}{2}\frac{W_1}{L_1}\mu C_{ox}(V_{GS1} - V_{th1})^2 \tag{8.16}$$

As will be shown in section 8.5.2.2, during opamp synthesis, given ω_t, the values of transistor lengths L_i and transistor gate effective voltage ($V_{EGi} = V_{GSi} - V_{thi}$) are held fix and are chosen according to noise and matching constraints. Eventually, for a two-stage Miller-compensated amplifier, the input transistor current I_{D1} can be determined from [Laker94]

$$\omega_t = \frac{g_{m1}}{C_C} = \frac{I_{D1}}{(V_{GS1} - V_{th1})C_C} \tag{8.17}$$

where C_C is the compensation capacitance of the two-stage amplifier.

The exponential factor in equation (8.11) is strongly affected by the parasitic capacitances. The same equation (8.12) can still be used with ω_t replaced by that of the two-stage amplifier instead. Obviously, these approximations neglect the effect of high frequency poles and zeros on the settling performance of the amplifier. For a sufficiently high PM ($> 70^o$), this can be safely done.

The above model is then employed in the discrete time simulations. The same non-idealities are considered in all amplifiers. A sinusoidal input of amplitude 0.5 and frequency 3.2 kHz is used.

First assuming an infinite opamp GBW frequency (f_t), i.e. $\theta_i = 0$, the effect of the amplifier gain on the overall SQNR is studied. Fig. 8.8(a) shows the results of such simulations. A gain of 40 dB is then sufficient for preserving the SQNR. In our design a gain of 70 dB has been chosen

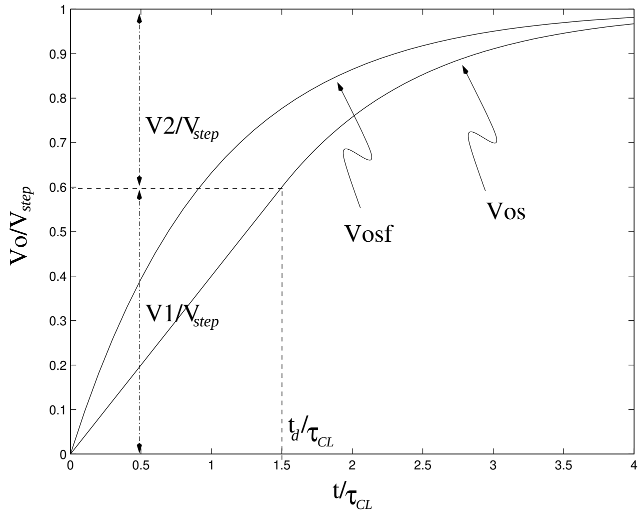


Figure 8.9: *Integrator slew-free output (Vosf) and slewing output (Vos).*

for the first opamp and 60 dB for the second and third ones. This high gain is chosen to avoid any performance degradation and to reduce the effect of non-linearities.

Using the above amplifier gains, Fig. 8.8(b) shows the variation of the SQNR with f_t. Simulations show that an $f_t > 2f_s$ is sufficient, where f_s is the sampling frequency. To have some margin, a ratio of 3.5 is chosen. Selected values are marked with small boxes on the corresponding figures.

8.4.2 Opamp Slew Rate

According to the integrator first-order model presented in the previous section, the time domain response of the integrator output during the integration phase is given by

$$V_{on}(t) = V_{o1n}(t) + V_{o2n}(t) \qquad\qquad 0 < t < t_i \tag{8.18}$$

where V_{on} is the integrator output after n clock cycles,

$$V_{o1n}(t) = \frac{\rho_i}{\rho_s}\left[1 - \theta_i(t)\left(1 - \frac{\rho_s}{\rho_i}\right)\right] V_{on}(0) \tag{8.19}$$

represents the leakage of the integrator stored value due to the amplifier finite gain and BW, and

$$V_{o2n}(t) = \frac{C_S}{C_I}\rho_i(1 - \theta_i(t))V_{in}(0) \tag{8.20}$$

represents the integrator response to the n^{th} integrator input. $\theta_i(t)$ is given by equation (8.11) and can be re-written as

$$\theta_i(t) = exp\left(-\frac{1}{\rho_i}\frac{t}{\tau}\right) \tag{8.21}$$

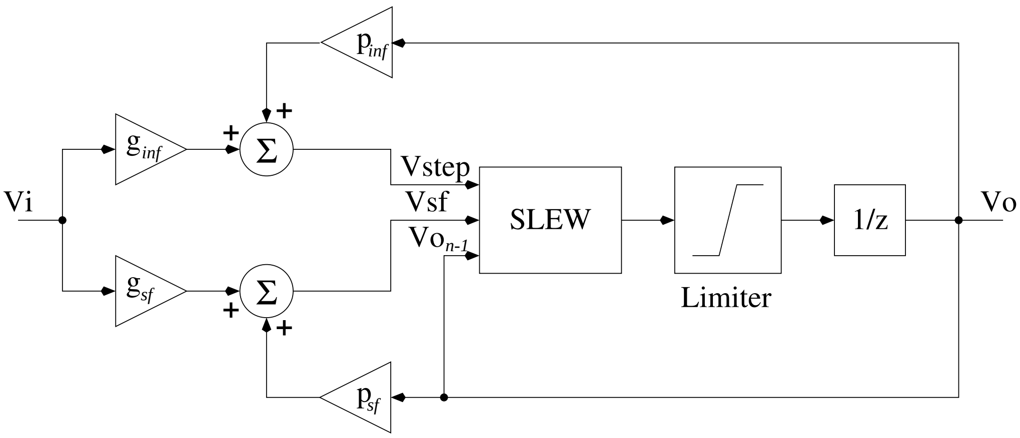


Figure 8.10: *Integrator model.*

where τ is the integrator time constant given by

$$\tau = \frac{1}{p_{CL}} = \frac{1}{\beta_i \omega_t} \tag{8.22}$$

Assuming $V_{o1n} \approx const = V_{o1}$ for the SR analysis, this is true for a high amplifier gain since both ρ_s and ρ_i approach unity, a slewing-free integrator output can be then formulated as

$$V_{osf}(t) = V_{o1} + V_{step}\left[1 - exp\left(-\frac{1}{\rho_i}\frac{t}{\tau}\right)\right] \tag{8.23}$$

where

$$V_{step} = V_{in}(0)\frac{C_S}{C_I}\rho_i \tag{8.24}$$

is the change in the integrator output assuming infinite ω_t.

However, the rate of change of the output can not exceed a certain limit fixed by the amplifier SR. For rapidly changing input, the output slews for a certain time t_d then enters the linear region as shown in Fig. 8.9. The slewing output can then be described by

$$V_{os}(t) = \begin{cases} SR.t & 0 < t < t_d \\ V1 + V2\left(1 - exp\left(-\frac{1}{\rho_i}\frac{t-t_d}{\tau}\right)\right) & t_d < t < t_i \end{cases} \tag{8.25}$$

where $V1$ and $V2$ are defined as shown in Fig. 8.9. Equating the slope of both sections at $t = t_d$, we have

$$V2 = SR.\tau.\rho_i \tag{8.26}$$

and using $V_{step} = V1 + V2 = SR.t_d + V2$, we obtain

$$t_d = \frac{V_{step}}{SR} - \tau.\rho_i \tag{8.27}$$

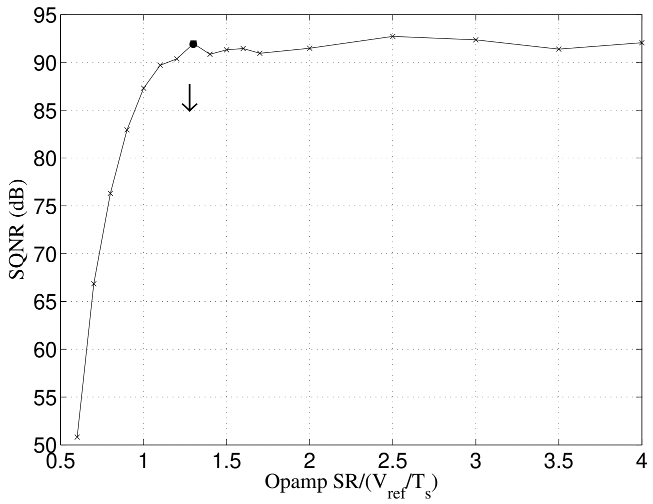


Figure 8.11: *Discrete-time simulation results showing the SQNR vs. amplifier SR.*

Fig. 8.10 shows an integrator model based on the above analysis. V_{step} represents the integrator output considering no frequency limitations. The parameters g_{inf} and p_{inf} are given by equations (8.5) and (8.6) respectively with $\theta_i = 0$. V_{sf} represents the slewing-free integrator output taking into account frequency limitations. The parameters g_{sf} and p_{sf} are given by equations (8.5) and (8.6) respectively. The *SLEW* block models the slewing behavior of the integrator: By calculating t_d, the output can be determined according to the slewing state:

- if $t_d < 0$, then the output is slew free and the integrator remains in the linear region,
- if $0 < t_d < t_i$, then slewing occurs but the integrator re-enters eventually in the linear region. The output can be calculated by equation (8.25),
- if $t_d > t_i$, the integrator remains slewing during the whole integration period and the output is given by $SR.t_i$.

A hard limiter is used to model the integrator output saturation levels.

Using the amplifier gain and GBW frequency calculated in the previous section, Fig. 8.11 shows the variation of the SQNR with the amplifier SR using the above integrator model. A SR of $1.3V_{ref}/T_s$ has been chosen.

8.4.3 Comparator Offset and Hysteresis

The most important characteristics of the comparator are:

- The offset voltage V_{offset} defined in Fig. 8.12.
- The hysteresis voltage V_{hys} defined in Fig. 8.12.

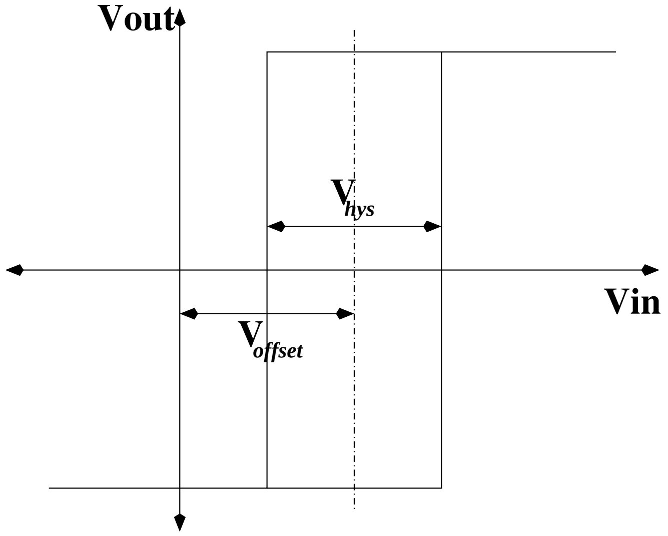
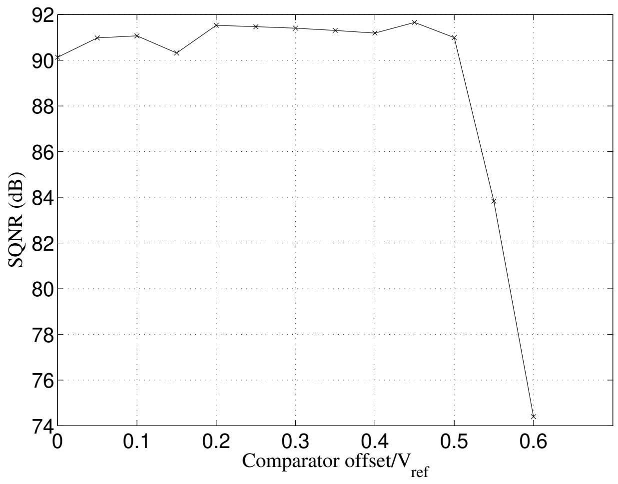
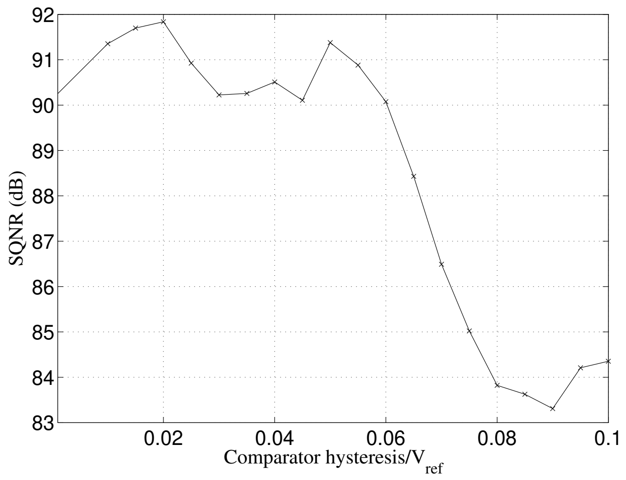


Figure 8.12: *Comparator offset and hysteresis.*



(a)



(b)

Figure 8.13: *Discrete-time simulation results showing the SQNR vs. the comparator (a) offset and (b) hysteresis.*

- The comparison speed. The right decision must be made available in time to the feedback DAC such that the DAC output is subtracted from the input. In this design, comparison and subtraction take place in two different clock phases (see section 8.5.1), this means that this time must be less than one clock phase, i.e. one-half a clock cycle.

The comparator model described by Fig. 8.12 is used to evaluate the effect of these non-idealities on the overall SQNR of the modulator. Fig. 8.13(a) shows the effect of the offset voltage. The effect of the offset is greatly reduced by the feedback loop of the modulator such that an offset of half of the reference voltage can still be tolerated. However, hysteresis is more critical. Fig. 8.13(b) shows that the ratio between V_{hys} and the reference voltage must be kept below 0.05.

8.5 Low-Level Synthesis and Design

In section 8.4, the effect of the major non-idealities on the overall modulator performance has been investigated. This analysis has also led to defining the corresponding limits imposed on the performance specifications of the building blocks. In this section, the analytical equations used to size the building blocks starting from the determined performance specifications are derived. In order to be used in COMDIAC, these equations need to be as accurate as possible. All transistor currents, transconductances and capacitances are calculated using the same model equations as that used in the circuit simulator and implemented in COMDIAC.

8.5.1 Switched-Capacitor Implementation

This work uses the very low-voltage SC scheme described in section 7.3.2 to build the $\Delta\Sigma$ modulator. Fig. 8.14 shows the circuit schematic of the switched capacitor implementation of the modulator shown in Fig. 8.3. The modulator is controlled by a two-phase, non-overlapping clocks ϕ_1 (sampling phase) and ϕ_2 (integration phase) together with their delayed versions ϕ_{1d} and ϕ_{2d} to reduce charge injection on the integration capacitance. Capacitors are connected such that upper plates are connected to noise sensitive nodes (opamp inputs), since lower plates are subjected to picking-up substrate noise. Capacitor lower plates are thus connected to opamp outputs. This, however, adds a considerable parasitic capacitance which loads the amplifier and consequently increase the power dissipation.

This integrator structure permits to set the input and output CM voltages independently. Three types of switches are used: a bootstrapped switch to switch signals around the opamp output CM voltage ($Vop_{CM} = V_{DD}/2 = 0.5V$) and at the circuit input, simple n-switch to switch the opamp input CM voltage ($Vip_{CM} = V_{SS} = 0V$), and CMOS switches to switch the modulator reference voltages which are either V_{DD} or V_{SS}.

A simple feedback DAC [Boser88] is used. The CMOS switches connect the left sides of the sampling capacitors to the reference voltages during the integration phase. This action performs both the D/A conversion and subtraction functions. Since the interstage and feedback coefficients are different for the second and third integrators (see table 8.1), the sampling capacitor is divided into two parallel capacitors where only one of them is connected to the reference voltages. The two corresponding parallel bootstrapped switches at the opamp outputs, however, share the same bootstrapping circuit as shown in Fig. 8.14.

The comparator is reset during ϕ_2 (the integration phase) in preparation for the next comparison, in the same time the latched DAC output is subtracted from the input. Comparison takes

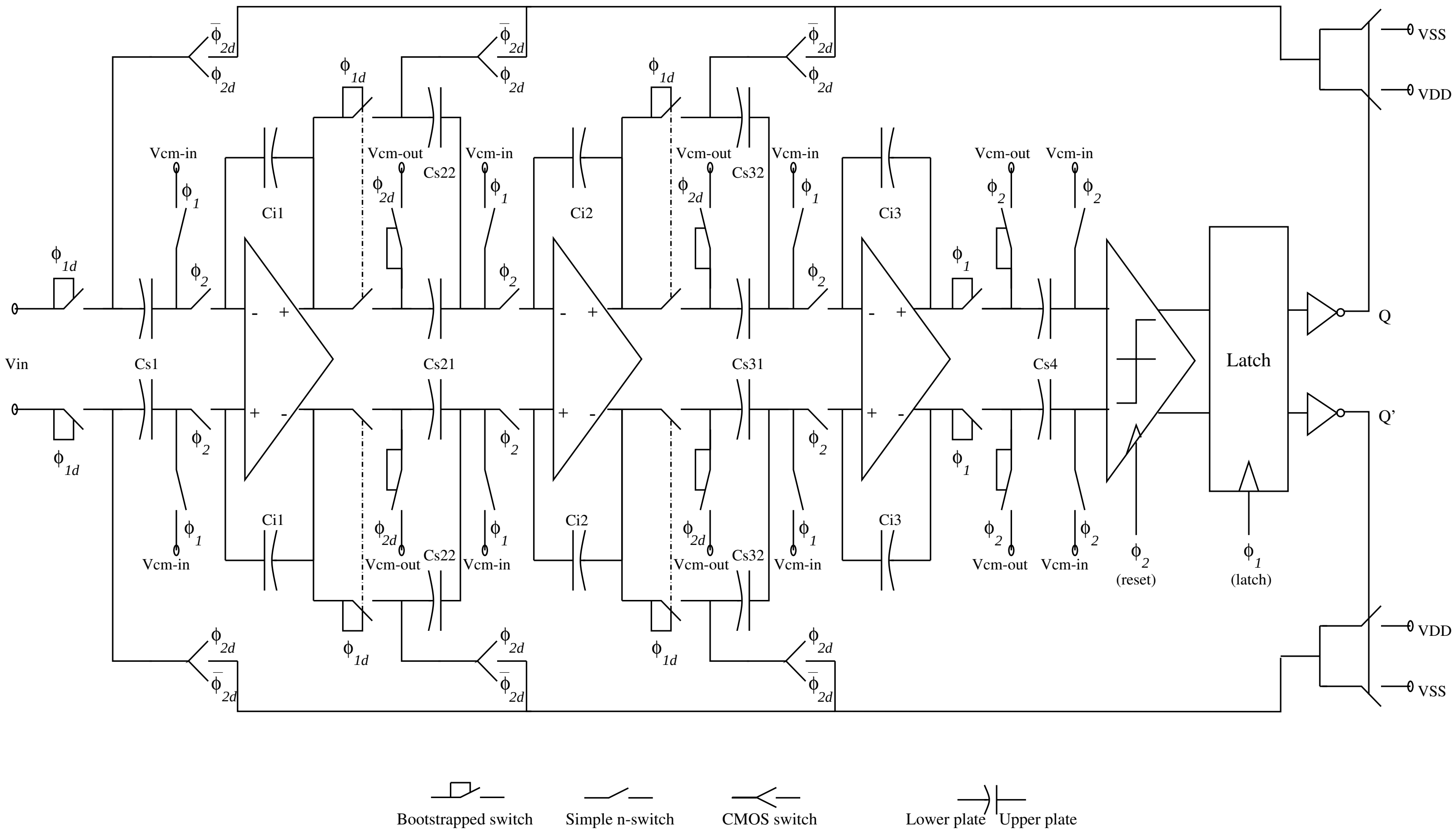


Figure 8.14: *Switched capacitor implementation of the modulator*.

	Integration Cap.	Sampling Cap.	Feedback Cap.
First Integrator	C_{I1} = 20.0	C_{S1} = 2.0	
Second Integrator	C_{I2} = 4.0	C_{S21} = 0.36	C_{S22} = 0.72
Third Integrator	C_{I3} = 2.5	C_{S31} = 0.35	C_{S32} = 0.425
Comparator		C_{S4} = 0.5	

Table 8.2: *Capacitor values (in pF), see section 8.5.2.3.*

place during ϕ_1 (the sampling phase). With this clocking arrangement, the time available for comparison is one-half a clock cycle.

8.5.2 Integrator Synthesis

In this section, equations describing the integrator main performance specifications are derived, this includes noise and dynamic range, settling, and CMFB circuit settling.

8.5.2.1 Noise and Dynamic Range Calculation

The dynamic range is defined as the ratio of the input power of a full scale sinusoidal input to the power of a small input for which the SNR is unity which represents the lowest detectable input signal. The dynamic range is thus given by:

$$DR = 10log(0.5V_{ref}^2) - 10log(N_T) \tag{8.28}$$

where V_{ref} is the modulator reference voltage and N_T is the total noise power in the signal BW which is equal to the power of the lowest detectable input signal.

The total noise power (N_T) contributing to the DR degradation can be divided into quantization noise (N_Q), switching noise or kT/C noise (N_{sw}) and opamp noise (N_{amp}):

$$N_T = N_Q + N_{sw} + N_{amp} \tag{8.29}$$

Quantization Noise: N_Q is determined by the modulator architecture. Fig. 8.2(b) shows the maximum achievable SQNR based on the quantization noise only. Circuit non-idealities cause quantization noise leakage that degrades the SNR. This effect has been previously studied in section 8.4.

Switch Noise: N_{sw} generated on the sampling capacitor C_S is given by kT/C_S. When sampled with a frequency f_s, this noise power is aliased into a band from 0 to $f_s/2$ (Assuming a single-sided frequency domain representation) [Gregorian86]. The total in-band noise due to switches is thus given by

$$N_{sw} = \left(\frac{kT}{C_S}\right) \cdot \left(\frac{1}{f_s/2}\right) .(f_m).(2).(2) = \frac{4kT}{OSR.C_S} \tag{8.30}$$

where OSR is the oversampling ratio given by equation (8.2). In the above equation, the first factor represents the total switching noise power which, when multiplied by the second factor, gives the power spectral density after aliasing. Both when multiplied by the maximum signal frequency f_m gives the in-band noise power. This power is then multiplied by 2 to take into account both the sampling and the integration phases. The final multiplication factor accounts for the used differential structure.

Opamp Referred Noise: N_{amp} consists of two components: thermal noise (N_{th}) and flicker or $1/f$ noise ($N_{1/f}$). This noise ($N_{th} + N_{1/f}$) is often calculated at the amplifier input. It must then be referred to the input sampling capacitor in order to be compared with the input signal and other noise sources. Thus,

$$N_{amp} = \left(N_{th} + N_{1/f}\right) .(F_{refer}) \tag{8.31}$$

where F_{refer} is the referring factor and is given by [Peluso98b]

$$F_{refer} = \left(\frac{1}{\beta_i}\right)^2 \cdot \left(\frac{1}{g_0}\right)^2 \tag{8.32}$$

where β_i is the feedback factor given by equation (8.14) and g_0 is the integrator gain. The first factor is the power gain of the amplifier (Assuming a high open-loop amplifier gain A_{d0}) referring the noise power to its output, while the second factor is the reciprocal of the integrator power gain thus referring the noise power to the integrator input. Note the strong influence of the integrator gain g_0 and the integration-phase feedback factor β_i on the input referred noise of the opamp. Since the first modulator coefficient a_1 corresponds to the first integrator gain

$$a_1 = g_0 = \frac{C_S}{C_I} \tag{8.33}$$

Combining this with equation (8.14) neglecting the opamp input capacitance, this referring coefficient F_{refer} renders to

$$F_{refer} \approx \left(1 + \frac{1}{a_1}\right)^2 \tag{8.34}$$

This coefficient F_{refer} is taken into consideration during modulator architecture optimization (see section 8.3.2).

The opamp in-band thermal noise in closed loop is given by

$$N_{th} = \left(\frac{8kT}{3g_{m1}}\right) .(\gamma_{th}). \left(f_{CL}\frac{\pi}{2}\right) \cdot \left(\frac{1}{f_s/2}\right) .(f_m) \tag{8.35}$$

where f_{CL} is the amplifier dominant closed-loop pole or simply the cut-off frequency. In the above equation, the first factor constitutes the input transistor thermal noise spectral density, γ_{th} represents the noise excess factor due to additional transistors given by equation (8.66). When multiplied by the noise bandwidth $f_{CL}\pi/2$ [Laker94] gives the total opamp thermal noise power. The last factor accounts for the aliasing effect. Assuming a one-pole opamp model, the closed-loop cut-off frequency f_{CL} is given by [Johns97]

$$f_{CL} = \beta_i f_t \tag{8.36}$$

where f_t is the GBW frequency of the open-loop amplifier given by $f_t \approx g_{m1}/(2\pi C_C)$, in which C_C is the opamp compensation capacitance. Thus equation (8.35) reduces to

$$N_{th} = \frac{2kT}{3}\frac{\gamma_{th}\beta_i}{OSR.C_C} \tag{8.37}$$

It should be noted that during calculations, the exact value of f_t is used.

On the other hand, flicker noise is added directly to the input signal without aliasing since it reduces to small values well below $f_s/2$ [Gregorian86]. Thus

$$N_f = \left(\frac{K_{Fp}}{C_{ox}W_1L_1}\right).(\gamma_f).ln\left(\frac{f_m}{f_l}\right) \tag{8.38}$$

where the first factor constitutes the input transistor flicker noise spectral density, γ_f represents the noise excess factor due to additional transistors given by equation (8.68). The last factor results from the integration of the noise density on the signal range from f_l to f_m.

Noise injected by the second and third integrators are suppressed by the transfer function of the preceding integrators. In fact they are shaped similar to the quantization noise [Peluso98a]. Thus only the first integrator noise is taken into account during the calculation of the total noise power.

Fig. 8.15 shows the different noise components. Circuit techniques such as *correlated double sampling* [Nagaraj87] or *chopper stabilization* [Hsieh81] are used to reduce the $1/f$ noise. The thermal noise is composed of two components: the switches noise N_{sw} and the amplifier thermal noise N_{th}. Since the switching noise is inversely proportional to the sampling capacitance (equation (8.30)), and the amplifier noise is inversely proportional to the amplifier compensation capacitance (equation (8.37)), to reduce the total thermal noise, both capacitance values must be increased. Both noise components are also inversely proportional to the OSR. This has a direct consequence on the amplifier power consumption as higher capacitances mean higher currents for the same GBW and SR and higher OSR means faster settling thus larger current. Therefore, in the noise budget, the white noise is usually the limiting noise contribution in the signal band as shown in Fig. 8.15.

Integrators after the first one are scaled down progressively. Decreasing sampling capacitors of subsequent stages reduces the capacitive load, power consumption, and layout area, while having negligible effect on the overall performance of the modulator. The scaling factors used are 1:0.5:0.4.

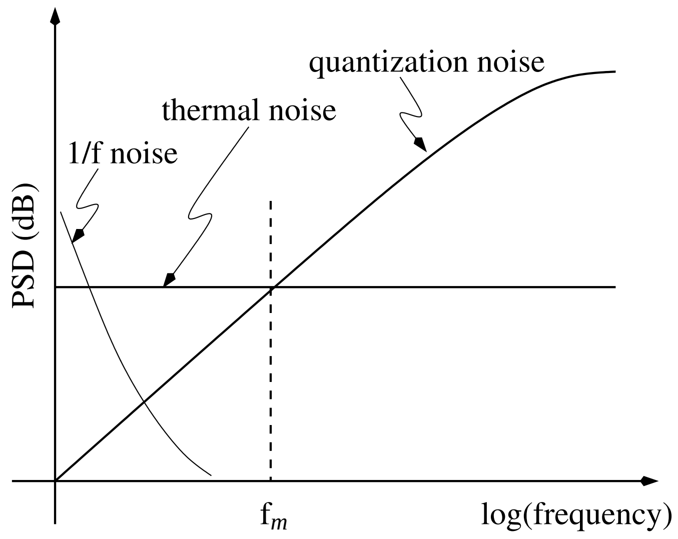


Figure 8.15: *Noise components.*

8.5.2.2 Opamp Synthesis

The low-voltage opamp has been presented in section 7.5. As stated in section 7.5.5, the Miller compensation has been retained and the complete opamp schematic is repeated in Fig. 8.16 for convenience.

In the rest of this section, the equations describing the main performance characteristics are derived. Since the differential gain of the completely symmetrical amplifier is defined as $A_d = (V_{out}^+ - V_{out}^-)/(V_{in}^+ - V_{in}^-)$, it is sufficient to analyze only one-half of the circuit. Fig. 8.17 shows the small signal model of only the left-hand side of the opamp. The output conductance at node $n1$ is given by $G_{15} = g_{ds1} + 2g_{ds51}$, while that at the circuit output is $G_o = g_{ds10} + g_{ds11}$. It should be noted that either one of the cascode transistors M31 or M32 is on at a time which is necessary to achieve chopper stabilization (see section 7.5.4). Since their gate bias is fixed, the effective cascode transistor transconductance is given by $G_{mc} = g_{m31} + g_{mb31}$. The capacitances C_L', C_{n1} and C_{n3} represent node capacitances given by

$$C_{n1} = C_{gs31} + C_{sb31} + C_{sb32} + C_{gd1} + C_{db1} + 2(C_{gd51} + C_{db51}) \tag{8.39}$$

$$C_{n3} = C_{gs11} + 2C_{gs51} + C_{db31} + C_{gd31} + C_{db32} + C_{db8} + C_{gd8} \tag{8.40}$$

$$C_L' = C_L + C_{LCM} + C_{gd11} + C_{db11} + C_{gd10} + C_{db10} \tag{8.41}$$

where C_L is the opamp load capacitance during the integration phase given by [Johns97]

$$C_L = \frac{(C_S + C_{ip})C_I}{C_S + C_{ip} + C_I} + C_{Ibp} \tag{8.42}$$

where C_{ip} is the opamp input capacitance, and C_{Ibp} is the bottom-plate capacitance of the integration capacitor C_I. C_{LCM} represents the loading of the dynamic CMFB circuit on the amplifier.

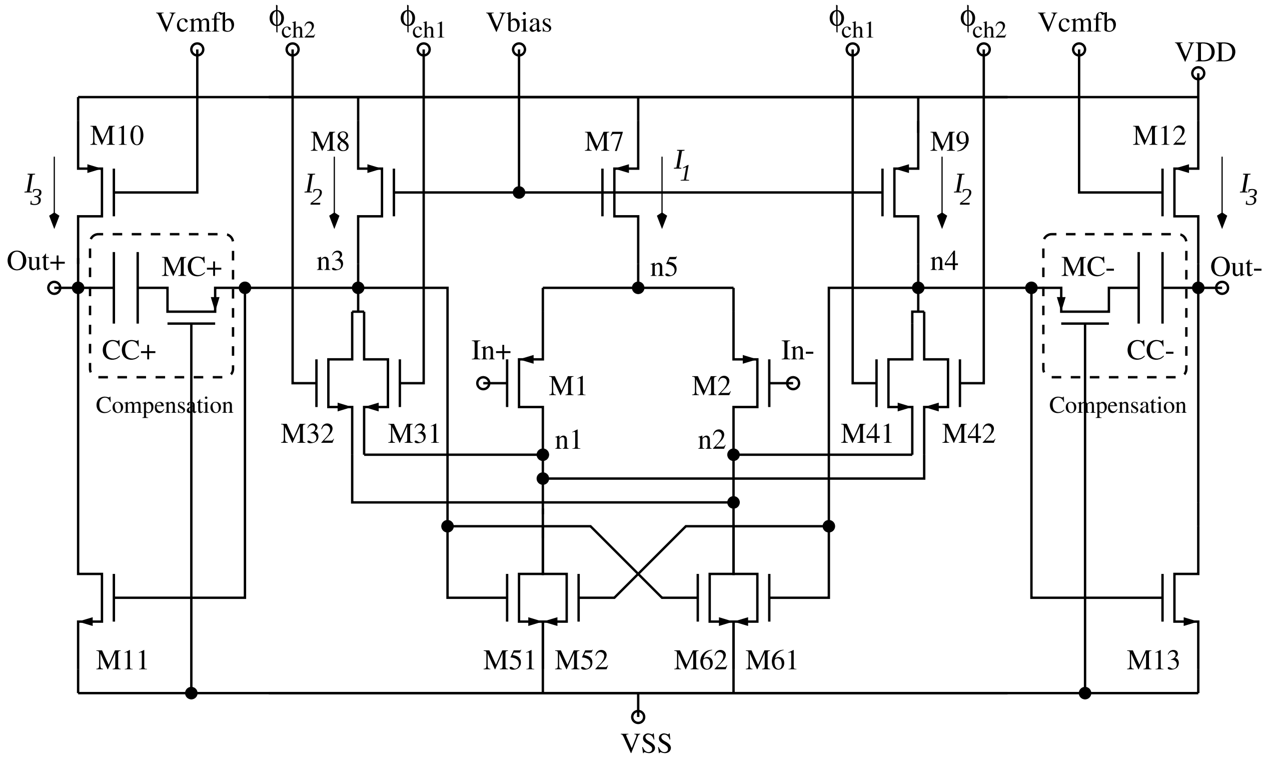


Figure 8.16: *Opamp schematic*.

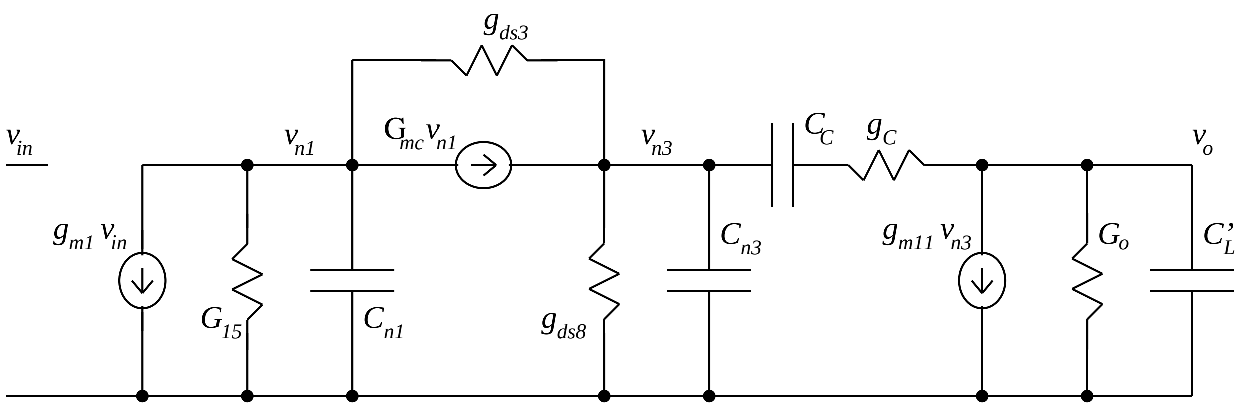


Figure 8.17: *Small signal model of the opamp shown in Fig. 8.16*.

C_{LCM} can be determined from Fig. 7.12 as follows: during ϕ_2 (integration phase) the two C1 capacitors are connected in series between the amplifier outputs. The point in between is however connected to the V_{cmfb} input of the amplifier which is charged by the gate capacitances of transistors M10 and M12. Assuming a differential output where $\Delta V_{out}^- = -\Delta V_{out}^+$, the voltage V_{cmfb} remains unchanged. Consequently, the gate capacitances draws no current and $C_{LCM} = C_1$. During the sampling phase, $\phi 1$, C2 is added in parallel to C1, but since during this phase the amplifier output hardly changes, this case is neglected.

Differential DC Gain: It can be calculated by direct inspection of Fig. 8.17 to be

$$A_{d0} = A_1 A_2 = \left(\frac{g_{mc1}}{g_{dsc1} + g_{ds8}}\right)\left(\frac{g_{m11}}{G_o}\right) \tag{8.43}$$

where g_{mc1} and g_{dsc1} are the equivalent folded cascode transconductance and output conductance respectively given by

$$g_{mc1} = \frac{g_{m1}}{1+\alpha} \tag{8.44}$$

$$g_{dsc1} = \frac{g_{ds31}}{1+1/\alpha} \tag{8.45}$$

where α is defined by

$$\alpha = \frac{g_{ds1} + 2g_{ds51}}{G_{mc} + g_{ds31}} \tag{8.46}$$

Frequency Analysis: Unfortunately, the small signal model shown in Fig 8.17 can not be further simplified and the resulting gain function is a complex one. Direct nodal analysis gives the following transfer function

$$A_d = \frac{v_o}{v_{in}} = \frac{g_{m1} Y W}{PQ - g_{ds3} Y Z} \tag{8.47}$$

where

$$X = y_C(g_{ds3} + g_{ds8} + sC_{n3}) + sC_C g_C \tag{8.48}$$

$$Y = y_C(G_{mc} + g_{ds3}) \tag{8.49}$$

$$Z = y_C(G_o + sC_L) + sC_C g_C \tag{8.50}$$

$$W = y_C g_{m11} - sC_C g_C \tag{8.51}$$

$$P = G_{15} + g_{ds3} + G_{mc} + sC_{n1} \tag{8.52}$$

$$Q = XZ + WsC_C g_C \tag{8.53}$$

$$y_C = g_C + sC_C \tag{8.54}$$

Besides the dominant pole at node $n3$ determined by the compensation capacitance C_C, the amplifier has four non-dominant poles and two zeros. The transition frequency f_t (at which $A_d =$

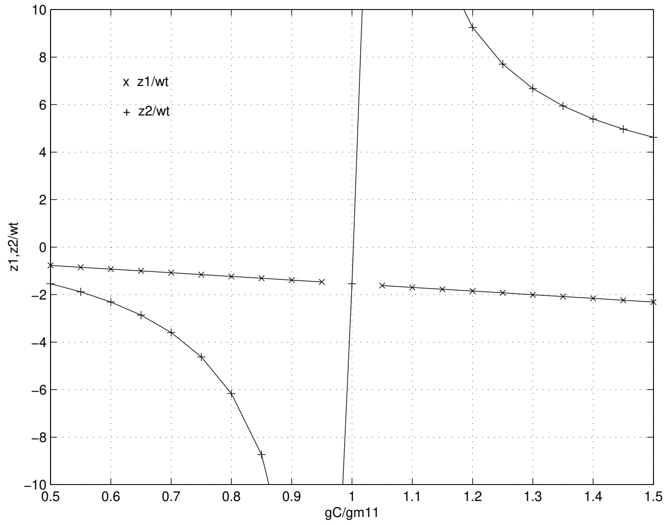


Figure 8.18: *Zeros plot*.

1, or approximately the GBW frequency) is calculated numerically from equation (8.47). Fig. 8.18 shows the variation of the position of the two zeros with the ratio $g_C/_{gm11}$. Around $g_C/g_{m11} = 1$, the first zero is always negative and vanishes completely at $g_C/g_{m11} = 1$. The designer has the choice of either eliminating this zero or using it to compensate one of the non-dominant poles in order to enhance the PM. This pole-zero doublet must be maintained at a frequency higher than the unity-gain frequency so as not to degrade the settling performance of the amplifier [Laker94]. The second zero is at high frequencies, it only changes sign as g_C/g_{m11} passes through unity.

Slew Rate: Let us assume a large differential voltage applied at the input such that $V_{in}^+ > V_{in}^-$. This causes M1 to be turned off and M2 to be turned heavily on. Since M1 is off, the difference of currents $(I_{D51} + I_{D52}) - I_{D8}$, that used to pass through M1, is obtained from M10 through the compensation capacitance C_C^+. Since the voltage at node n3 is *nearly* fixed and is equal to V_{GS11}, this current causes the output voltage V_{out}^+ to rise linearly with a SR given by

$$SR_{int+} = \frac{I_{D1}}{C_C^+} \tag{8.55}$$

The capacitance C_C is not the only capacitance that is going to be charged, the load capacitance also present at the output is charged by the available current from M10. This current is actually only $I_{D10} - I_{D1}$, since C_C^+ takes I_{D1} away. Consequently, for a large positive input at V_{in}^+, the voltage at node $n3$ decreases, decreasing the current through M11. Current $I_{D10} - I_{D1}$ then charges C_L', resulting in a positive voltage ramps with a slope given by

$$SR_{ext} = \frac{I_{D10} - I_{D1}}{C_L'} \tag{8.56}$$

where C'_L is given by equation (8.41).

On the other half-side of the opamp, since M1 is turned off, all the current I_{D7} of the current source transistor M7 is diverted through M2. Since this current is usually greater than $I_{D61}+I_{D62}$, both M2 and M7 will go into the triode region, causing I_{D7} to decrease until it is equal to $I_{D61}+I_{D62}$. The current I_{D9} passes then entirely through the compensation capacitor C_C^- and is sinked by transistor M13. Note that transistor M13 can sink large currents when overdriven discharging the load capacitance in the same time. This causes the output voltage V_{out}^- to decrease linearly by an internal SR given by

$$SR_{int-} = \frac{I_{D9}}{C_C^-} \tag{8.57}$$

The SR is thus limited by

$$SR = min(SR_{int+}, SR_{int-}, SR_{ext}) \tag{8.58}$$

Output Voltage Range: From Fig. 8.16 the output range is limited by the output transistors M10(M12) and M11(M13). The output CM voltage $VopCM$ is set to $V_{DD}/2$. Since these transistors must be kept in saturation under all conditions, and the output swing must be symmetrical around Vop_{CM}, thus the output range is given by

$$Vop_{swing} = V_{DD} - 2.max(V_{dsat10}, V_{dsat11}) \tag{8.59}$$

Since $V_{dsat} = V_{GS} - Vth$, in order to increase the output range, the effective gate-source voltage must be reduced. This, however, increases transistor sizes and consequently transistor parasitic capacitances increase which limit the achievable unity-gain frequency.

Input Capacitance: To calculate the input capacitance at V_{in}^+, the other input is shorted to ground. There are two possible paths for the input current (i) at the gate of M1: the first (i_1) passes through C_{gs1} and the second (i_2) passes through C_{gd1} such that $i = i_1 + i_2$.

Fig. 8.19(a) shows the small signal model for i_1 calculation. Since the gate of M2 is grounded, a conductance of g_{m2} appears at the source of M1 (the bulks of M1 and M2 are both connected to the source to enhance matching between the input differential pair transistors). It is to be noted that the output conductances of the input transistors g_{ds1} and g_{ds2} are eliminated from the model as they contribute with opposite and equal currents to node $n5$. Current i_1 is thus found to be

$$i_1 = v_{in}^+ . \frac{sC_{gs1}}{2} \tag{8.60}$$

In Fig. 8.19(b), node $n5$ is assumed to be a virtual ground. The conductance G_1 is the equivalent conductance at node $n1$ given by $G_1 = g_{m3} + g_{mb3} + g_{ds1} + g_{ds51} + g_{ds52}$. Knowing that the current through g_{ds3} is given by $(v_{n3} - v_{n1})g_{ds3} = (-A_1 v_{in}^+ - v_{n1})g_{ds3}$, i_2 is thus given by

$$i_2 = v_{in}^+ .sC_{gd1} \left(1 + \frac{g_{m1} + g_{ds3}A_1}{G_1 + g_{ds3}}\right) \tag{8.61}$$

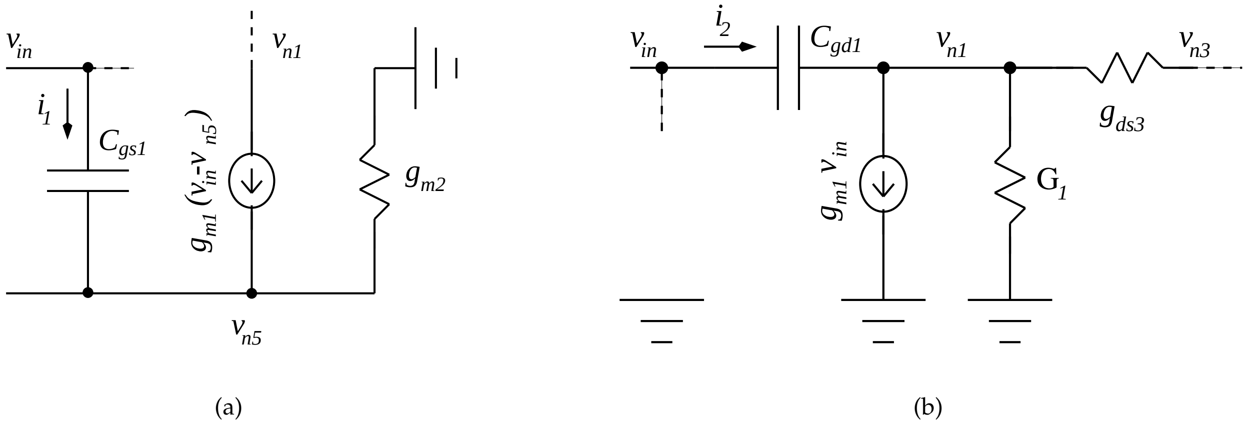


Figure 8.19: *Models for calculation of input currents: (a) i_1 and (b) i_2.*

From equations (8.60) and (8.61), the input capacitance is given by

$$C_{ip} = \frac{C_{gs1}}{2} + C_{gd1}\left(1 + \frac{g_{m1} + g_{ds3}A_1}{G_1 + g_{ds3}}\right) \tag{8.62}$$

The capacitance C_{gd1} is thus amplified by the Miller effect.

Noise Performance: All transistor noise voltage sources can be added to one equivalent input noise voltage $\overline{dv_{ie}^2}$ using

$$\overline{dv_{ie}^2} = \sum_{i=1}^{n} \overline{dv_{ni}^2}\left(\frac{A_{vni}}{A_v}\right)^2 \tag{8.63}$$

where $\overline{dv_{ni}^2}$ is the equivalent input noise voltage of transistor Mi, A_{vni} is the gain from that noise source to the output and A_v is the amplifier output/input gain.

From Fig. 8.16, the noise contribution from the second stage is negligible assuming a high gain in the first stage. Also, the noise contribution from the cascode transistors M31, M32, M41 and M42 is negligible due to their small gain resulting from their high source resistance. The amplifier equivalent input noise voltage is then

$$\begin{aligned}\overline{dv_{ie}^2} &= 2\overline{dv_{n1}^2} + 4\overline{dv_{n51}^2}\left(\frac{g_{m51}}{g_{m1}}\right)^2 + 2\overline{dv_{n8}^2}\left(\frac{g_{m8}}{g_{m1}}\right)^2 \\ &= \overline{dv_{n1}^2}\cdot\left[2 + 4\frac{\overline{dv_{n51}^2}}{\overline{dv_{n1}^2}}\left(\frac{g_{m51}}{g_{m1}}\right)^2 + 2\frac{\overline{dv_{n8}^2}}{\overline{dv_{n1}^2}}\left(\frac{g_{m8}}{g_{m1}}\right)^2\right]\end{aligned} \tag{8.64}$$

The second term in the above equation is defined as the excess noise factor γ which gives the ratio of the equivalent input noise $\overline{dv_{ie}^2}$ to that of the input transistor $\overline{dv_{n1}^2}$ only.

The equivalent input noise voltage consists of both thermal noise and $1/f$ noise components. At intermediate frequencies the thermal noise is dominant. Substitution of the single transistor thermal noise source [Laker94]

$$\overline{dv_{nith}^2} = \frac{8kT}{3g_{mi}} df \tag{8.65}$$

the thermal excess noise factor can be calculated as follows

$$\gamma_{th} = 2\left[1 + 2\frac{g_{m51}}{g_{m1}} + \frac{g_{m8}}{g_{m1}}\right] \tag{8.66}$$

From equations (8.64), (8.65) and (8.66), it can be concluded that thermal noise can be reduced either by reducing the contribution of the input transistors by increasing their g_m, or by reducing γ_{th} by reducing the ratio of g_m's of the current source transistors M51 and M8 to that of the input transistors. At low frequencies the $1/f$ noise is dominant. Similarly using the single transistor $1/f$ noise source [Laker94]

$$\overline{dv_{nif}^2} = \frac{K_F}{C_{ox}W_iL_if} df \tag{8.67}$$

the $1/f$ excess noise factor is found to be

$$\gamma_f = 2\left[1 + 2\frac{K_{Fn}}{K_{Fp}}\frac{W_1L_1}{W_{51}L_{51}}\left(\frac{g_{m51}}{g_{m1}}\right)^2 + \frac{W_1L_1}{W_8L_8}\left(\frac{g_{m8}}{g_{m1}}\right)^2\right] \tag{8.68}$$

Again, it can be concluded that flicker noise can be reduced either by reducing the contribution of the input transistors by increasing their area, or by reducing γ_f. Since g_m is proportional to $sqrtI_DW/L$, reducing γ_f means making the lengths of the current source transistors M51 and M8 as long as possible.

Estimation of the Settling Error: The worst case settling error ϵ can be estimated by assuming that the integrator is a first-order system[1]. For such system the slewing-free response to a pulse input $u(kT_s + t)$ for $0 < t < t_i$ is given by

$$v(kT_s + t) = g_0u(kT_s)\left(1 - e^{-t/\tau}\right) + v(kT_s) \tag{8.69}$$

where g_0 is the integrator gain, T_s is the sampling period, t_i is the integration time, and τ is the linear settling time constant given by equation (8.22) in which the GBW (ω_t) is calculated taking into account the loading of the feedback network.

The peak rate of change in the pulse response given by equation (8.69) occurs at $t = 0$ and is given by

$$\frac{dv(kT_s + t)}{dt}|_{t=0} = g_0\frac{u(kT_s)}{\tau} \tag{8.70}$$

[1]This assumption becomes more true for a sufficiently high PM. During this design, the PM was kept above 70^o. More detailed analysis can be found in [Marques99] where settling of third-order systems is investigated.

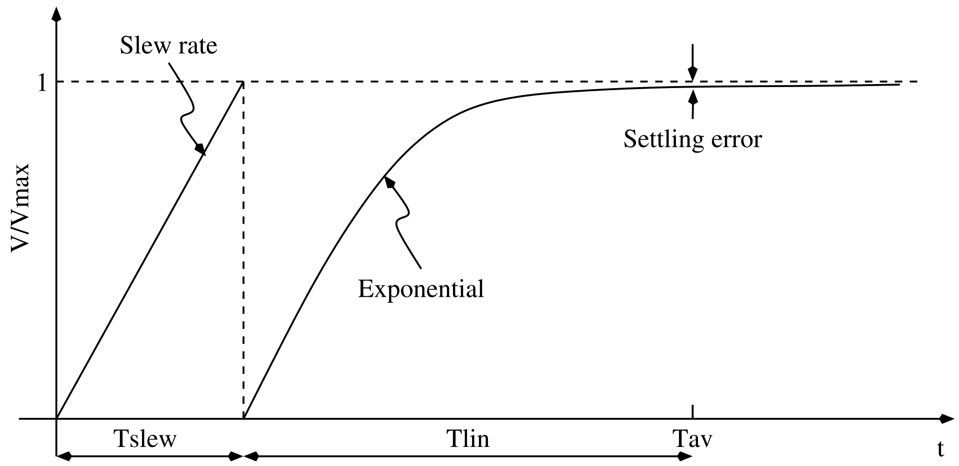


Figure 8.20: *Worst-case settling time.*

Setting $u(kT_s)$ at its maximum value of U_{max} which is the maximum stable input defined in section 8.3.2, this last equation then gives the minimum SR performance specification required for the opamp to avoid slewing distortion.

If slewing occurs, the output changes linearly with a slope determined by the SR for a certain time t_{slew}. Eventually, the output enters the linear region for a certain time t_{lin} where it continues exponentially such that

$$t_{av} = t_i = t_{slew} + t_{lin} \tag{8.71}$$

where t_{av} is the available time for settling which is the same as the integration time t_i.

In order to obtain a worst-case value for the settling error, both the slewing time and the linear settling time, corresponding to the maximum integrator input, are calculated separately and added as shown in Fig. 8.20 [Laker94]. First, the output is assumed to be only slew limited such that the slewing time corresponding to the maximum input (U_{max}) is given by

$$t_{slew} = \frac{g_0 U_{max}}{SR} \tag{8.72}$$

where $g_0 U_{max}$ represents the maximum step at the integrator output. Then, the integrator output is assumed to be only limited by the linear time constant (τ). Finally, from equations (8.69) and (8.71), the settling error (ϵ) is given by

$$\epsilon = e^{-t_{lin}/\tau} = e^{-(t_i - t_{slew})/\tau} \tag{8.73}$$

where t_{slew} is given by equation (8.72).

Common-mode Feedback Time Constant: As shown in section 7.5.2, the amplifier uses a SC CMFB circuit for the second stage. This circuit is shown in Fig. 7.12. In order to calculate the

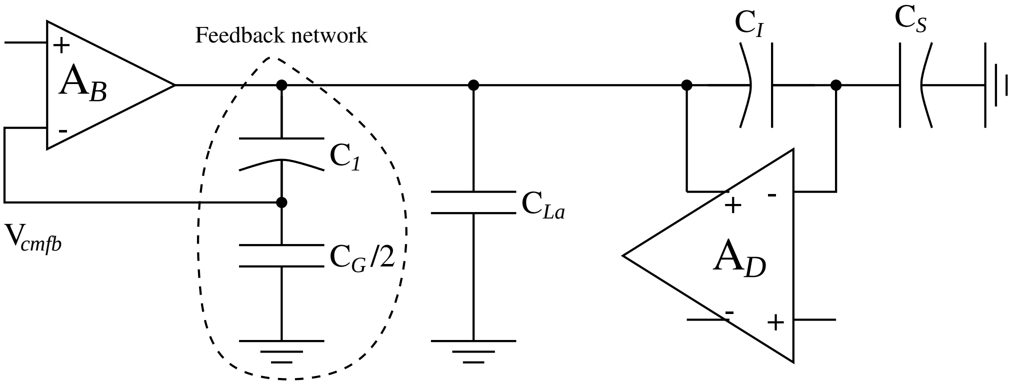


Figure 8.21: *The common feedback closed-loop load.*

settling time of the CMFB circuit, let us assume a first-order system characterized by a linear time constant (τ_{cmfb}) given by

$$\tau_{cmfb} = \frac{1}{\beta_{cmfb}\omega_{tcmfb}} \tag{8.74}$$

where β_{cmfb} is the CM feedback factor and ω_{tcmfb} is the GBW of the CM amplifier in closed loop.

The CMFB loop is composed of the common-source amplifier M10(M12), loaded with M11(M13) and shared with the differential amplifier shown in Fig. 7.13, in addition to the feedback capacitor C_1 shown in Fig. 7.12. The input capacitance (C_G) at the input of the CM amplifier (the gates of M10 and M12) is given by

$$C_G = C_{gs10} + C_{gs12} + C_{1bp} \tag{8.75}$$

in which C_{1bp} is the bottom plate parasitic capacitance associated with C_1.

The CM feedback factor β_{cmfb} during the integration phase (ϕ_2) is calculated considering a CM signal at the amplifier output, i.e. the two outputs are equal. The CM current in C_G is thus twice that in C_1, and the feedback factor is given by

$$\beta_{cmfb} = \frac{C_1}{C_1 + C_G/2} \tag{8.76}$$

Fig. 8.21 shows the CMFB system with all capacitive loading, for one output of the differential amplifier, in closed-loop [Peluso98a]. The amplifier A_B represents the CM amplifier M10(M12). The factor $1/2$ used for C_G is used to take into account the CM feedback factor given by equation (8.76). The effective load capacitance seen by A_B is found by analyzing this circuit from a series-shunt feedback perspective. Specifically, treating the feedback amplifier input V_{cmfb} as an open-circuit [Gray93], the load capacitance can thus be calculated

$$C_{Lcmfb} = \frac{C_1 C_G/2}{C_1 + C_G/2} + C_{La} + \frac{C_I(C_S + C_{ip})}{C_I + C_S + C_{ip}} \tag{8.77}$$

in which C_{La} is the CM amplifier load capacitance given by

$$C_{La} = C_{gd11} + C_{db11} + C_{gd10} + C_{db10} + C_{Ibp} + C_C \tag{8.78}$$

where C_{Ibp} is the bottom plate parasitic capacitance associated with C_I. Since the CM of the first stage is inherently regulated, the CM voltage of node $n3$ inside the amplifier (Fig. 7.13) is zero and the compensation capacitance C_C loads the CM amplifier of the second stage. Hence, the closed-loop GBW is given by

$$\omega_{tcmfb} = \frac{g_{m10}}{C_{Lcmfb}} \tag{8.79}$$

8.5.2.3 Integrator Sizing in COMDIAC

The above equations are incorporated in the sizing environment COMDIAC (see chapter 6) in order to estimate the integrator main performance characteristics. Fig. 8.22 shows a detailed design plan that describes the low-level synthesis step. Starting from the required SNR performance, quantization noise is calculated as shown in section 8.3. As explained in section 8.5.2.1, the in-band noise power must be dominated by the circuit noise rather than the quantization noise in order to minimize the total power consumption. Circuit noise depends on the circuit implementation, it is further decomposed to switch noise and amplifier noise (see section 8.5.2.1). Switch kT/C noise power is mainly determined by the value of the input sampling capacitor (C_{S1}). Based on equation (8.30), this capacitance is chosen to be 2pF. This leaves sufficient margin for opamp noise optimization. Starting from the modulator coefficients given in table 8.1 and taking into account integrator scaling, all capacitor values are determined and shown in table 8.2. Due to the very low coefficient of the first integrator (a_1) the amplifier noise becomes dominant, see equations (8.31) and (8.34). The amplifier thermal noise power depends directly on the compensation capacitor C_C, see equation (8.37). But since it also depends on the amplifier noise excess factor γ_{th}, which is not known before the complete amplifier design, fine tuning of the compensation capacitor value is thus needed during the amplifier sizing process.

Fig. 8.23 shows the integrator sizing plan. In our design procedure, there are two sets of input parameters. The first set is determined directly by the previous high-level analysis, and through performance parameter mapping (see section 8.4). This set of input parameters, shown horizontally in Fig. 8.23, includes:

- The maximum stable input signal amplitude (U_{max}).
- The sampling frequency (f_s) and the integration phase duty cycle.
- The opamp gain, GBW, and SR.
- Circuit noise.

Table 8.3 shows the values of these input parameters. The second set, shown vertically (shaded) on the right of Fig. 8.23, is used for opamp design optimization. It includes:

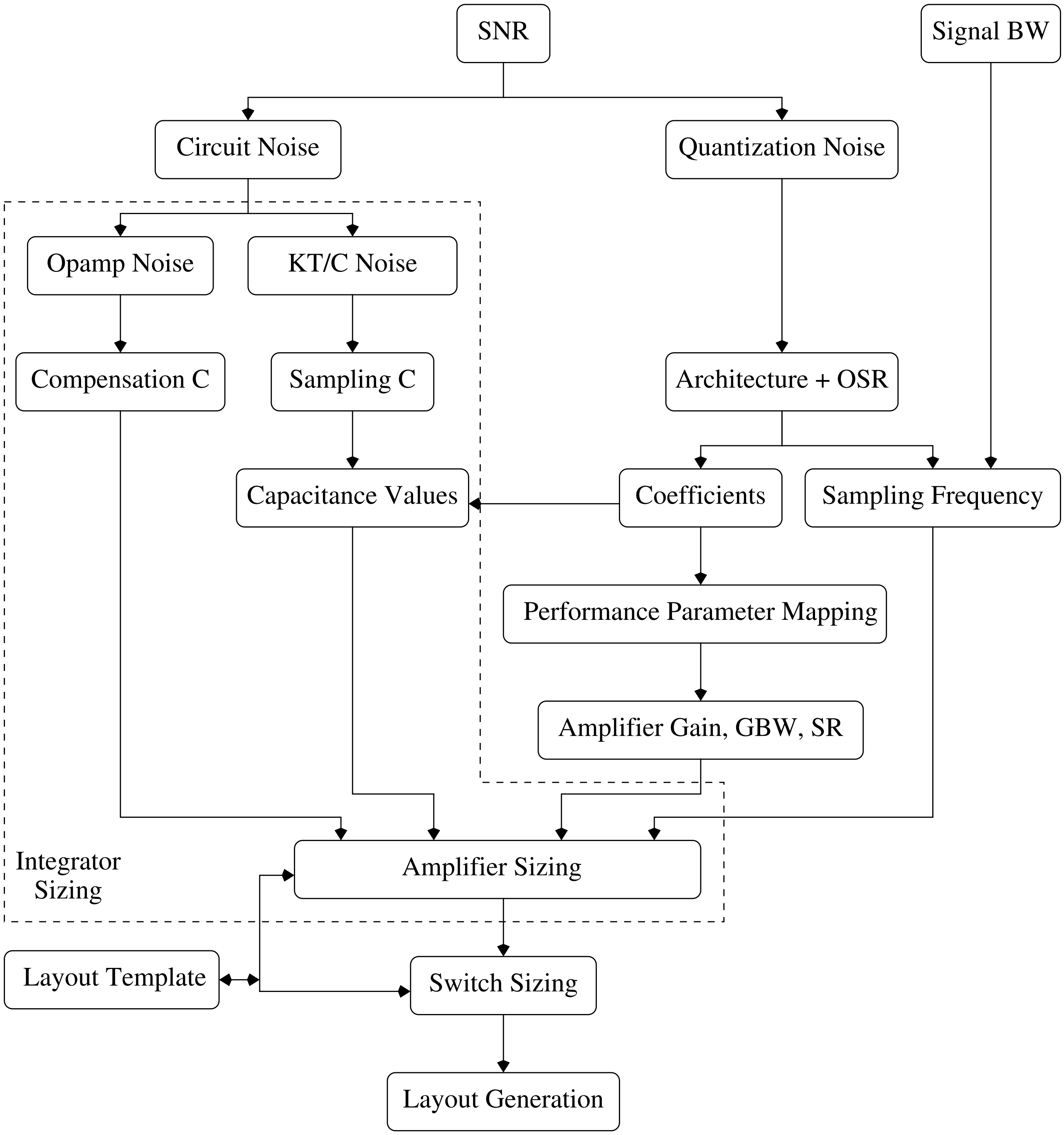


Figure 8.22: *Modulator design plan.*

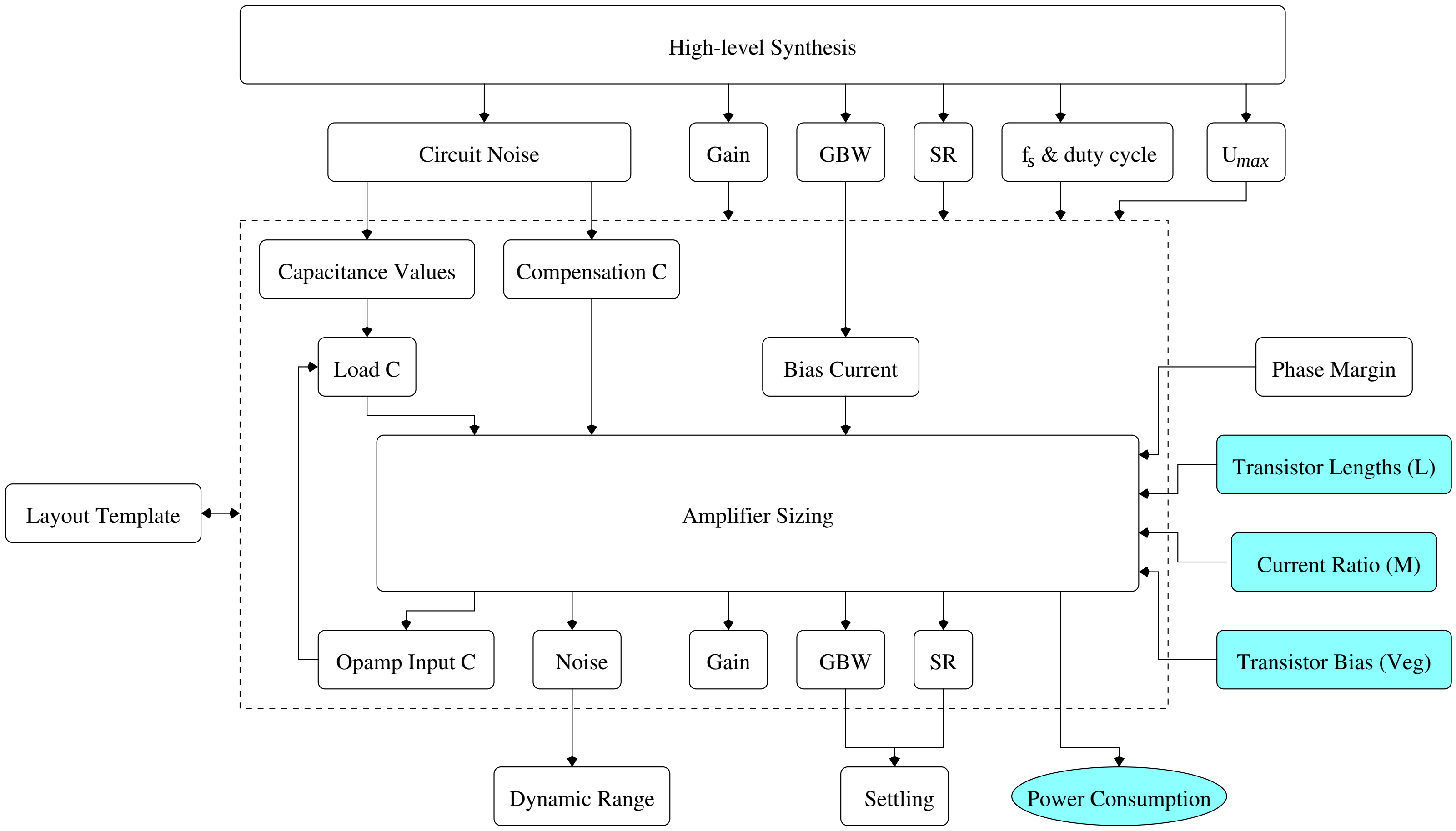


Figure 8.23: *Integrator design plan.*

Parameter	Definition	Value
V_{DD}	Supply voltage	1.0 V
U_{max}	Maximum stable input	0.78 V
OSR	Oversampling ratio	100
f_s	Sampling frequency	3.2 MHz
$duty_i$	Integration duty cycle	0.55
f_t	Gain-Bandwidth product	11 MHz
A_{d0}	DC gain	70 dB
SR	Slew rate	4 V/μs

Table 8.3: *First integrator COMDIAC input parameters*.

- The phase margin, PM.
- Selected transistor lengths.
- Transistor bias voltages V_{EG} and V_{DS}.
- Ratio between the bias-circuit current and the opamp branch currents.

The integrator sizing module uses the opamp module as a building block. The opamp load C_L is calculated taken into account the loading of the integrator feedback network as given by equation (8.42). Since C_L depends on the opamp input capacitance C_{ip} which is only calculated after sizing, two or more iterations are needed to find C_L. The equations derived in section 8.5.2.2, used to calculate the different opamp performance specifications, are implemented in the sizing tool COMDIAC as a separate sizing procedure. Referring to Fig. 8.16, the input parameters for the opamp sizing process are:

- the supply voltage, V_{DD},
- the load capacitance, C_L,
- the compensation capacitance, C_C,
- the phase margin, PM,
- the GBW (ω_t) or the bias circuit current (I_B) in the associated bias circuit (refer to section 9.2.1),
- the transistor element ratios M_7/M_B, M_8/M_B and M_{10}/M_B, where M_7, M_8 and M_9 are the number of parallel transistors constituting the current source transistors M7, M8 and M10 respectively, and M_B is that of the bias circuit transistor,
- the effective gate-source voltage, $V_{EG} = V_{GS} - V_{th}$, or the gate-source voltage, V_{GS} of each transistor, and

Transistor	V_{EG} (V)	V_{GS} (V)	W/L (μm)	M
M1, M2	-0.07		118.9/0.35	2
M3, M4		0.75	14.4/0.35	3
M51(2), M61(2)			11.8/0.35	5
M7		-0.8	804.8/3.0	16
M8, M9		-0.8	301.8/3.0	6
M10, M12		-0.74	91.2/0.35	16
M11, M13	0.135		24.4/0.35	2
MC+, MC-			85.0/0.35	3

Table 8.4: *Opamp transistor gate-source/effective gate-source voltages and the corresponding calculated sizes for the first integrator*.

- the drain-source voltage of each independent transistor, V_{DS}.

Since current mirrors are usually implemented using multiple parallel transistors of the same elementary transistor motif to enhance matching, the ratio between the number of parallel transistors is used as an input parameter to the sizing procedure rather than the current ratio. Slight differences in the mirrored current due to different drain-source voltages are then taken into account during sizing.

The optimization goal was to minimize the power consumption under a given settling and dynamic range performance. As explained in chapter 6, synthesis depends on *interactive* user feedback coupled with a fast sizing procedure. This means, that optimization is accomplished by calling the sizing procedure several times allowing design space exploration. The opamp sizing strategy is based on fixing the operating point of each transistor (see section 6.3), this means that sizing given the biasing current I_B is faster than that given the GBW as it avoids additional iterations to find the biasing current. During synthesis, first the opamp GBW is fixed according to high-level simulations, the corresponding current level is then determined and used afterwards in further sizing. For gain and frequency performance determining transistors M1(M2) and M11(M13), it is preferable to bias the transistor by fixing V_{EG} during sizing as it controls directly the transistor transconductance g_m. On the other hand, for biasing current source transistors, V_{GS} is used since it facilitates the design of the bias circuit. Due to the limited supply voltage, several transistors are biased in the moderate inversion region. Some transistors are also forced to have minimum lengths to reduce parasitics on the internal nodes thus reducing the power consumption. This, however, did not affect the noise performance since chopper stabilization is used for 1/f noise reduction. However, where matching is an issue, e.g. for biasing, a large L is forced where possible. Table 8.4 shows the chosen gate-source (V_{GS}) / effective gate-source (V_{EG}) voltages for each transistor. The gate-source voltage of transistors M51(2), M61(2), MC+ and MC-

Parameter	Definition	Value
C_S	Sampling capacitance	2.0 pF
C_I	Integration capacitance	20.0 pF
C_C	Compensation capacitance	14.0 pF
β_i	Equation (8.10)	0.8912
g_0	Equation (8.33)	0.1
C_L	Equation (8.42)	4.0 pF
PM@$\beta_i = 1$	Phase margin	72^o
I_B	Bias circuit current	10.0 μA
f_t	Gain-Bandwidth product	11.4 MHz
A_{d0}	Equation (8.43)	73.5 dB
τ	Equation (8.22)	15.7 ns
C_1	CMFB capacitor, Fig. 7.12	1.0 pF
τ_{cmfb}	Equation (8.74)	15.23 ns
SR	Equation (8.58)	4.15 V/μs
SR_{min}	Equation (8.70)	4.79 V/μs
ϵ	Equation (8.73)	-86.4 dB
Vop_{swing}	Equation (8.59)	0.74 V
N_{sw}	Equation (8.30)	-100 dB
$N_{th}.F_{refer}$	Equations (8.37) and (8.32)	-88.05 dB
$N_{1/f(8,9)}.F_{refer}$	Equation (8.38) and (8.32)	-99.42 dB
DR	Equation (8.28)	85 dB
P_{c1}	Power consumption	570 μW
P_{cT}	Total power consumption	950 μW

Table 8.5: *Modulator calculated parameters based on the first integrator, using COMDIAC.*

depend on that of M11(M13). The table also shows the obtained transistor sizes, together with the number of parallel transistor elements M which is determined by the layout tool based on the pre-defined layout template. Multiple synthesis runs have been tried, and the one with the minimum power consumption has been retained. Table 8.5 contains the calculated design parameters for the modulator based only on the first integrator. The flicker noise given by equation (8.38) is calculated only for transistors M8 and M9, since that of the other transistors is suppressed by chopper stabilization.

During sizing, layout parasitics are also taken into account according to the methodology presented in section 4.3. This includes exact transistor diffusion capacitance after the calculation of the parallel elements M, and the capacitors' bottom plate capacitance.

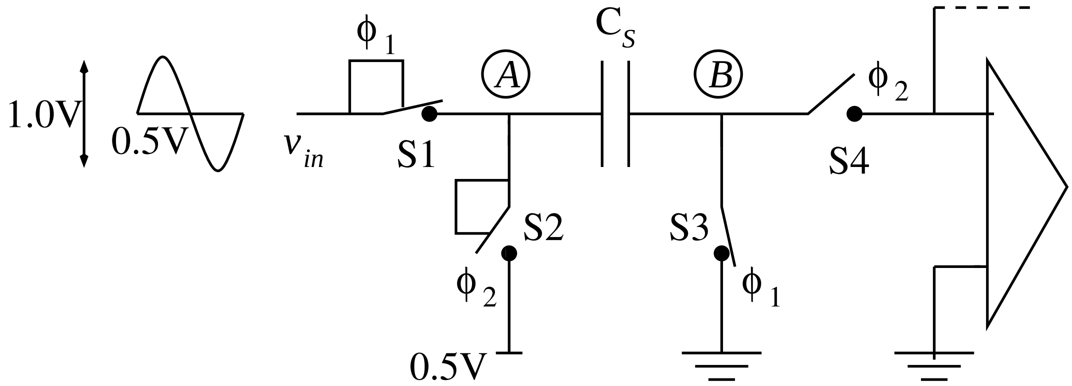


Figure 8.24: *Sampling (ϕ_1) and integration (ϕ_2) phases of the sampling capacitor in a low-voltage SC integrator.*

8.5.3 Switch Synthesis

In this section, the switch sizing procedure is presented. Special attention is given to low-voltage switch operation.

In very low voltage SC circuits, the switch overdrive V_{EG}, in spite of being held constant by the bootstrapping technique presented in section 7.4, it is limited to only a few hundreds of millivolts. Switches in their on-state are always considered as a small series resistance. However, the small switch overdrive does not guarantee its operation in the linear region. Specifically, if the switch has a high drain-source voltage in its off-state, the switch starts conducting in the saturation region [Peluso97] if $V_{DS} > V_{EG}$. The drain-source voltage then decreases due to charging/discharging the series capacitance and eventually the switch enters in the linear region.

8.5.3.1 Integrator Switch Synthesis

Fig. 8.24 shows the sampling and integration phases of the sampling capacitor in a SC integrator based on the double reference scheme presented in section 7.3.2. Switches S1 and S2 are bootstrapped switches while S3 and S4 are n-switches.
During the integration phase (ϕ_2), point B is connected to the virtual ground opamp input (at V_{SS}) while point A is connected to the reference potential at 0.5 V. First, consider the sampling phase (ϕ_1): Point B does not change its potential since it is also connected to V_{SS}. The drain-source voltage of S3 remains at zero potential which guarantees the operation of S3 in the linear region. This in not the case for S1 which connects point A to the input signal that is assumed to swing from $V_{DD} = 1V$ to $V_{SS} = 0V$. At both extremes the switch drain-source voltage is thus around $\pm 0.5V$ at the switching moment. If the overdrive of the bootstrapped switch V_{EG} is less than its drain-source voltage, S1 then *starts* conducting in the saturation region. It then enters the linear region as the charging goes on and the drain-source voltage moves towards zero. During the integration phase (ϕ_2), C_S is discharged through switches S2 and S4. Similar to S1, according

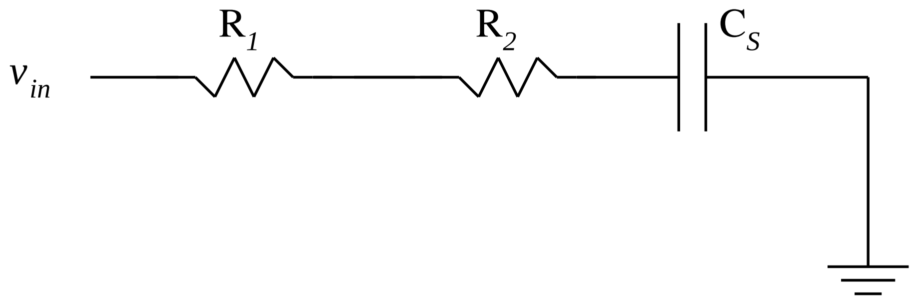


Figure 8.25: *Sampling/integration phases of Fig. 8.24 with both switches operating in the linear region.*

to the input voltage, S2 can also start conducting in the saturation region then moves to the linear region. On the other hand, S4 is always in the linear region while conducting.

During either the sampling or integration phases there exists two series switches charging/discharging the sampling capacitor C_S. One of these switches is always in the linear region while the other may occasionally starts conducting in the saturation region, according to the input signal, in which the switch's current is held constant at I_{Dsat} and the switch is said to slew at a rate given by

$$SR_{switch} = \frac{I_{Dsat}}{C_S} \tag{8.80}$$

for a certain time t_{slew}. The switch then enters the linear region where charging/discharging of C_S is continued with a time constant τ for a certain time $t_{lin} = t_{av} - t_{slew}$, where t_{av} is the available time. In the linear region the two series switches can be represented by linear resistances as shown in Fig. 8.25. Charging accuracy is measured using the settling error ϵ such that

$$\begin{aligned} t_{lin} &= \tau ln\left(\frac{1}{\epsilon}\right) \\ &= (R_1 + R_2)C_S ln\left(\frac{1}{\epsilon}\right) \\ &= t_{lin1} + t_{lin2} \end{aligned} \tag{8.81}$$

where

$$t_{lin1} = \tau_1 ln\left(\frac{1}{\epsilon}\right) = R_1 C_S ln\left(\frac{1}{\epsilon}\right) \tag{8.82}$$

and

$$t_{lin2} = \tau_2 ln\left(\frac{1}{\epsilon}\right) = R_2 C_S ln\left(\frac{1}{\epsilon}\right) \tag{8.83}$$

Switch sizes are chosen to yield a certain settling error (ϵ) in a given period of time t_{av}. During sizing, t_{av} is divided into slewing (t_{slew}), and linear (t_{lin}) times. From equation (8.81), the linear settling time t_{lin} is further divided into t_{lin1} and t_{lin2}, given by equations (8.82) and (8.83) respectively, each depending on one of the two switches, such that

$$t_{av} = t_{slew} + t_{lin} \tag{8.84}$$

$$= t_{slew} + t_{lin1} + t_{lin2} \tag{8.85}$$

This allows the two switches to be sized *separately* given only the sampling capacitance C_S. The slewing switch determines t_{slew} and t_{lin1} while the other switch determines t_{lin2}. The following section describes how sizes are calculated from these parameters.

8.5.3.2 Switch Sizing in COMDIAC

In this section, we discuss the method used for switch sizing implemented in COMDIAC, given the following input parameters:

- Transistor length.
- The load capacitance C_S.
- Gate-source voltage V_{GS} assumed to be constant during charging.
- Initial drain-source voltage V_{DSinit}.
- Bulk-source voltage V_{BS}.
- Available time for charging the load capacitance t_{av}.
- Settling error ϵ.

Worst-case settling is assumed by considering that the given t_{av} will be divided into a slewing time t_{slew} during which the switch slews throughout the whole given $V_{DS} = V_{DSinit}$ down to $V_{DS} = 0$ in addition to a linear time t_{lin} during which slewing is neglected and a linear settling to the required settling error is assumed as shown in Fig. 8.20. The sizing procedure is summarized in Fig. 8.26. It starts from the minimum transistor width W_{min}. If the transistor starts in the saturation region, it calculates the switch SR using equation (8.80). The slewing time is then calculated referring to Fig. 8.20 by

$$t_{slew} = \frac{V_{DSinit}}{SR_{switch}} \tag{8.86}$$

The linear time is then calculated using

$$t_{lin} = \tau ln\left(\frac{1}{\epsilon}\right) = \frac{C_S}{g_{ds}} ln\left(\frac{1}{\epsilon}\right) \tag{8.87}$$

Then if the total time $t_{slew}+t_{lin}$ is less then the given available time t_{av}, then sizing is accomplished, if not the process is repeated by incrementing the transistor width which in turn increases the

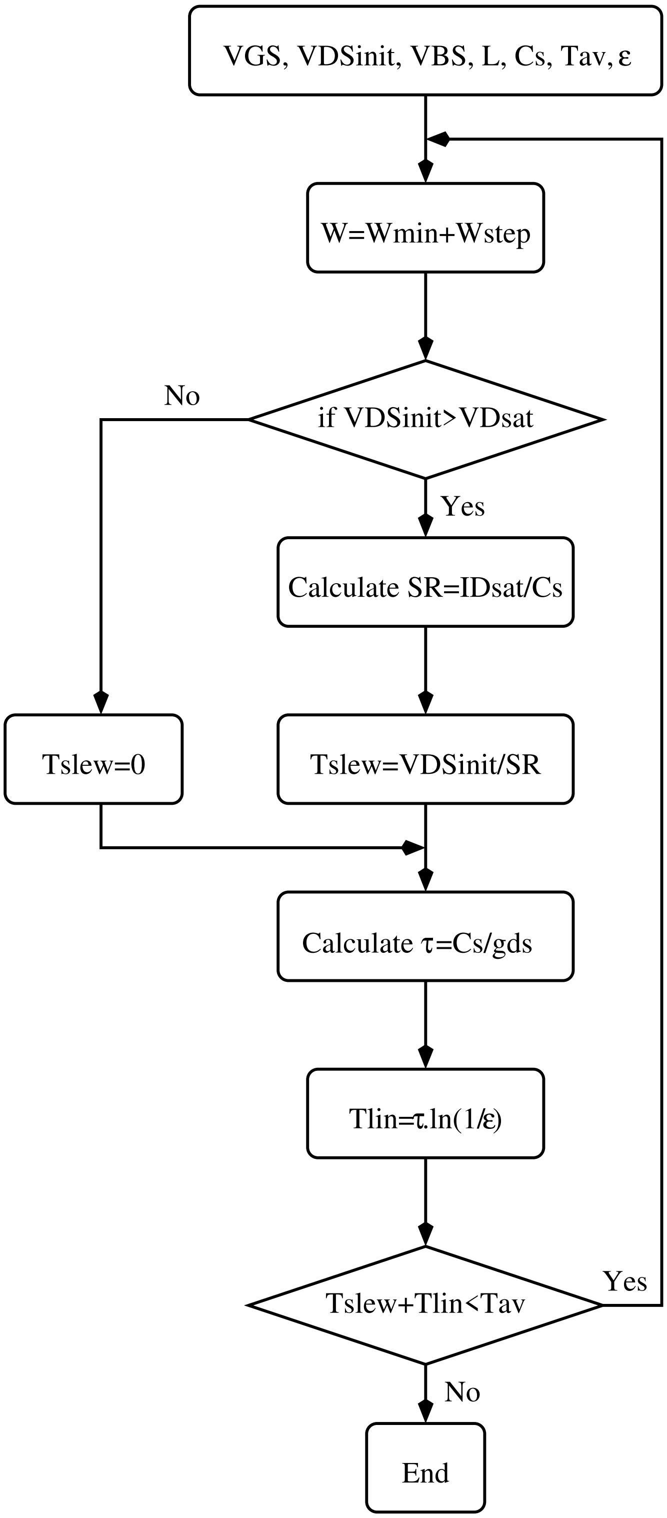


Figure 8.26: *Automatic switch sizing procedure.*

saturation current (and consequently the switch SR) and the drain-source conductance g_{ds} both leading to decrease t_{slew} and t_{lin} respectively.

These automating sizing procedure allow to size *separately* each switch in the modulator circuit shown in Fig. 8.14 based on its charge and the required settling error. This optimizes switch sizes for low-voltage operation which happens to be large compared to normal SC circuits due to the small switch overdrive in order to minimize as much as possible the clock feedthrough due to the large switch gate capacitances. In order to avoid any performance degradation due to switch settling, all switches are required to settle to the accuracy of the modulator.

8.5.3.3 Bootstrapped Switch Sizing

The bootstrapped switch described in section 7.4 as well as the accompanying bootstrapping circuit are both sized based on the basic switch sizing procedure described in section 8.5.3.2. Fig. 8.27 shows the bootstrapped switch circuit, repeated here for convenience. A special sizing procedure, presented in this section, is thus developed and incorporated in COMDIAC.

The main switch MNSW is first sized giving the same input parameters given in section 8.5.3.2. Since C_{offset} is first charged to V_{DD}, the gate voltage of MNSW at the end of ϕ_1 is given by (refer to appendix B)

$$v_G = \frac{C_{offset}}{C_{offset} + C_G}(v_{in} + V_{DD}) \tag{8.88}$$

while the voltage on the offset capacitance is given by

$$v_C = \frac{C_{offset}}{C_{offset} + C_G}V_{DD} - \frac{C_G}{C_{offset} + C_G}v_{in} \tag{8.89}$$

where C_G is the parasitic capacitance on the gate side of C_{offset} given by

$$C_G = C_{gsw} + C_{g1} + C_{g7} + C_{g6} + C_{g2} + C_{wellB} \tag{8.90}$$

The capacitance C_{offset} must be large enough to supply sufficient charge to the gate of MNSW when it is turned on. A significant voltage reduction across C_{offset}, due to capacitance division, might drive node B (and consequently the N-well of MP4) below V_{DD} causing latch-up. The capacitance C_{offset} is thus chosen to at least 10 times that of C_G. Given the size of MNSW, C_G is estimated to be $C_G = 5C_{gsw} + C_{wellB}$.

In order to determine the size of the remaining switches in the bootstrapping circuit using the same procedure described in section 8.5.3.2, in addition to the bias voltages which are easily determined from Fig. 8.27, the load capacitance seen by each switch, as well as the available time for charging t_{av} must be calculated. These two values are determined and shown in table 8.6 where

$$C_P = \frac{C_{offset}C_G}{C_{offset} + C_G} \tag{8.91}$$

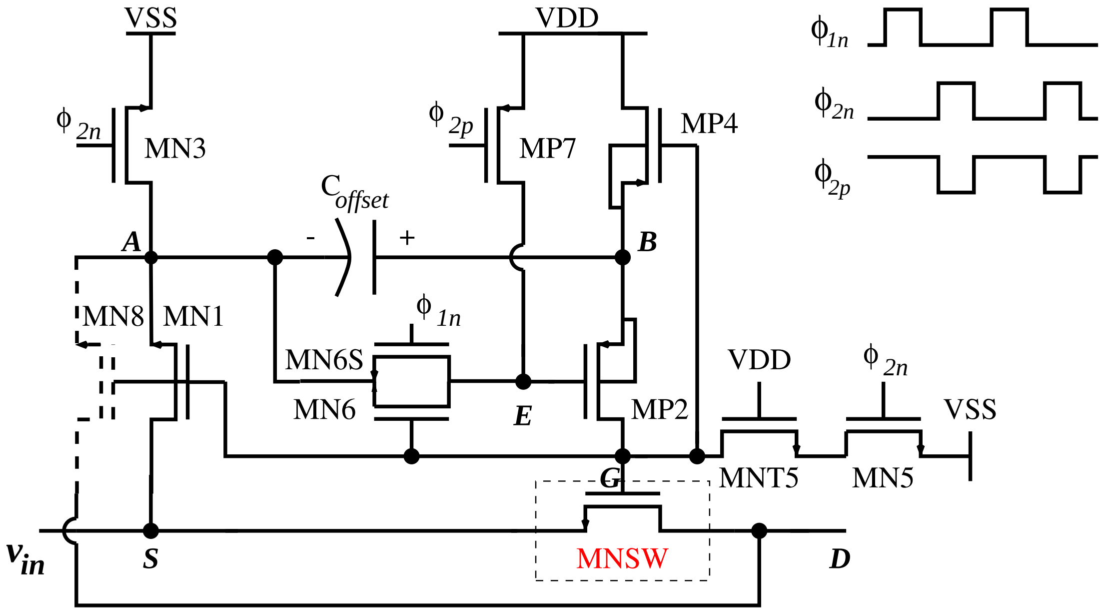


Figure 8.27: *Bootstrapped switch.*

Switch	Load Capacitance	t_{av}
MNSW	C_S	t_{avsw}
MN1/MN7	$C_P + C_{offsetbp}$	$0.1 \times t_{avsw}/2$
MP2	C_P	$0.1 \times t_{avsw}/2$
MN3	$C_{offset} + C_{offsetbp}$	$t_{avsw}/2$
MP4	C_{offset}	$t_{avsw}/2$
MN5/MNT5	C_G	$0.1 \times t_{avsw}/2$
MN6/MN6S	C_{g2}	$0.1 \times t_{avsw}/2$
MP7	C_{g2}	t_{avsw}

Table 8.6: *Load capacitance and charging available time for each switch in the bootstrapping circuit shown in Fig. 7.7.*

and $C_{offsetbp}$ is the bottom plate capacitance of C_{offset}. The available time for switches acting during the on-phase of MNSW, ϕ_1, is set to $0.1 \times t_{avsw}$, where t_{avsw} is the available time for MNSW. This value is then divided by two if two series switches are responsible for the charging/discharging process as explained in section 8.5.3.1. However, for MN3, MP4 and MP7 whose charging time is not critical to the operation of MNSW, the corresponding available time is set to be equal to t_{avsw}. In addition, the settling error ϵ for all switches in the bootstrapping circuit is not critical to circuit operation, so a value of 1% is usually sufficient.

As an example, consider the sizing of a bootstrapped switch under 1V operation with a clock

V_{DSinit}	1.0 V
V_{GS}	1.0 V
V_{BS}	-1.0 V
C_S	2.0 pF
t_{av}	156.25 ns
ϵ	1.0E-5

Table 8.7: *Example of the bootstrapping circuit: input parameters.*

Switch	W/L (μm)
MNSW	6.0/0.35
MN1/MN7	1.0/0.35
MP2	1.8/0.35
MN3	0.8/0.35
MP4	2.3/0.35
MN5/MNT5	0.5/0.35
MN6/MN6S	0.5/0.35
MP7	0.5/0.35

Table 8.8: *Example of the bootstrapping circuit: sizes.*

frequency of 3.2 MHz, in a standard CMOS technology with n/p-transistor threshold voltages of 580/600 mV. The source is discharged to zero volt before switching such that the initial V_{DS} may reach 1V. Since the bulk is tied to $V_{SS} = 0V$, the worst-case bulk-source voltage is equal to -1V. The load capacitance is equal to 2pF, and settling is required to a high accuracy corresponding to an ADC resolution of 14 bits. Input parameters are summarized in table 8.7. Following the above procedure, the offset capacitor was found to be 0.5pF. All transistor sizes are summarized in table 8.8 taking minimum transistor lengths.

8.5.4 Comparator-Latch Design

Since there is no critical design requirements on the comparators used in $\Delta\Sigma$ modulators, a simple low voltage comparator similar to that presented in [Peluso98b] is used. The comparator is shown in Fig. 8.28(a). It is composed of the input differential stage M1/M2 of p-type transistors with an input CM at V_{SS}. A SC level shifting circuit, shown in Fig. 8.14, is used to shift the CM level from that at the output of the last opamp at $V_{DD}/2$ to V_{SS}. The positive feedback connection of transistors M3 and M4 is used for the regeneration action. Resetting the comparator to the meta-stable state is done using a bootstrapped switch as shown in Fig. 8.28(a).

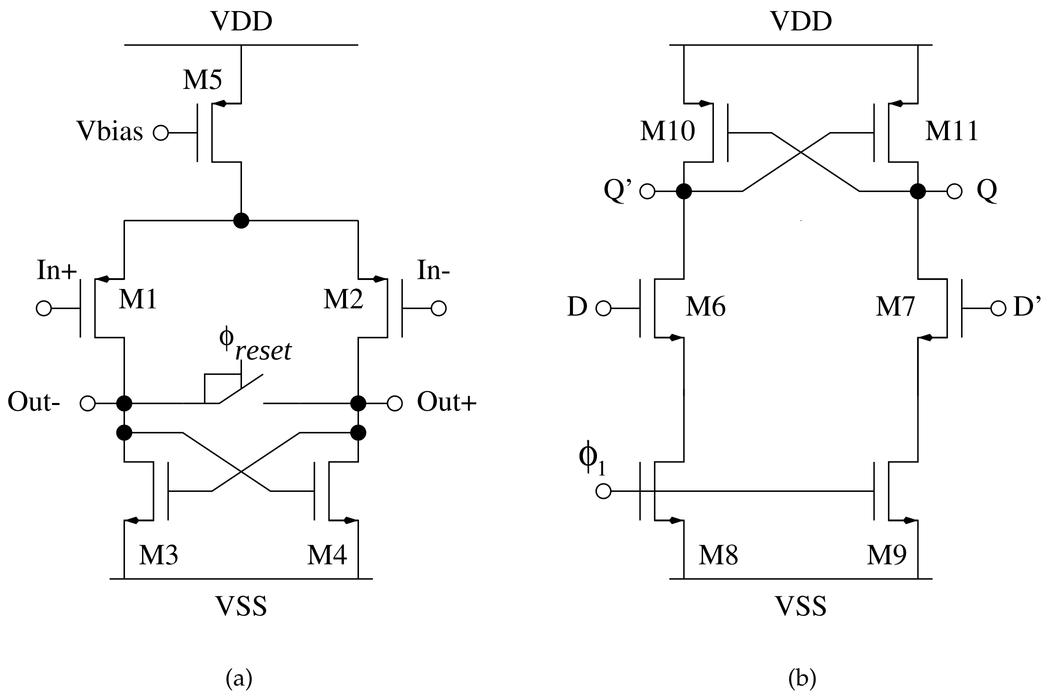


Figure 8.28: *Low voltage (a) comparator and (b) latch.*

The same latch, shown in Fig. 8.28(b), used in [Peluso98b] is also employed. The meta-stable point at the comparator output should be chosen below the threshold level of the latch V_{th6}, such that if the outputs of the comparator have not diverged enough in the available time, the latch should not trigger.

8.6 Conclusions

In this chapter, the detailed design of a very low-voltage $\Delta\Sigma$ modulator is presented. The design is performed to validate the design methodology, the synthesis tools, and the design techniques developed in the previous chapters.

A high performance 14 bit modulator for digital audio applications was fixed as a design goal. The major design steps were introduced together with the accompanying tools, starting from the high level specifications to the circuit level sizing.

On the system level, a third-order single-loop modulator architecture was shown to be suited for such low-voltage, low-power, high resolution converter. The first integrator coefficient was shown to have a great effect on the noise performance of the modulator. The coefficients were then determined based on modulator stability and noise considerations.

Block non-idealities were then modeled and simulated using discrete-time simulations by MATLAB. This includes the opamp finite gain, the opamp finite BW, the opamp speed and the comparator offset and hysteresis. These simulations have demonstrated the effect of each of these non-idealities on the signal-to-noise performance of the modulator. They also provide the needed performance specifications for these characteristics in order to limit the modulator noise leakage. This allow to map the high level specifications to the building blocks ones.

Each building block was then analyzed analytically on the transistor level and incorporated in the knowledge-based sizing tool COMDIAC presented in chapter 6.

Chapter 9

Prototype Implementation

9.1 Introduction

This chapter presents the prototype circuit implementation of the very low-voltage $\Delta\Sigma$ modulator[1] based on both the analysis given in the previous chapter, as well as the tools presented in chapters 5 and 6.

In section 9.2, some chip implementation choices concerning the bias circuit and clock generation are given.

In section 9.3, physical implementation of the sized modulator is described. Technology and layout issues are also discussed.

Section 9.4.2 describes measurement setup used during prototype test. Some measurement results are then given and compared to some recent low-voltage implementations.

In order to demonstrate eventual design reuse, section 9.5 introduces two other modulator designs using the same design procedure. The first is identical to the given modulator but in a different process while the other is a fourth-order one.

Finally, the chapter ends with some concluding remarks.

[1]Circuit design (using the tools described in chapters 5 and 6) and all measurements have been done at the Institut Supérieur d'Electronique du Nord (ISEN) under the supervision of prof. Andreas Kaiser.

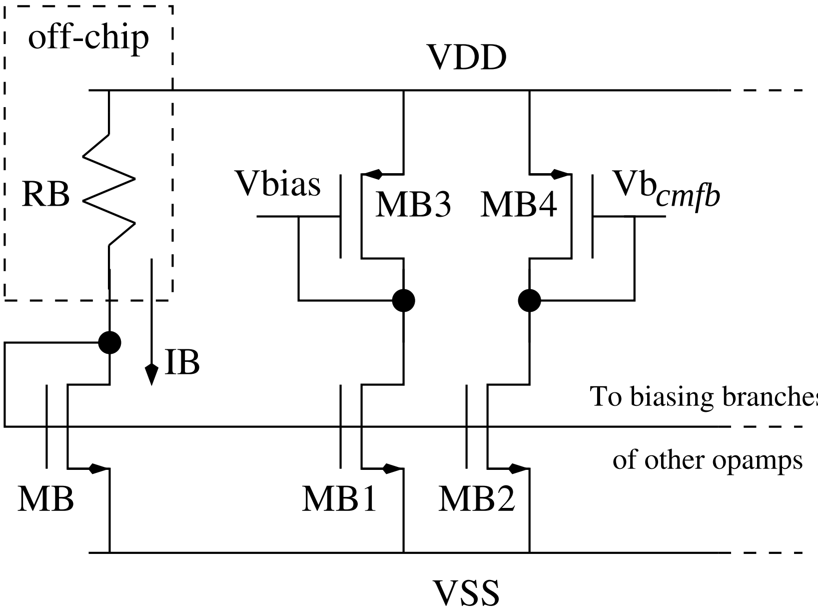


Figure 9.1: *Bias circuit for the first opamp.*

Transistor	V_{GS} (V)	W/L (μm)	M
MB, MB1, MB2	0.8	8.2/2.0	2
MB3	-0.8	50.3/3.0	1
MB4	-0.74	5.7/0.35	1

Table 9.1: *Bias network gate-source voltages and the corresponding calculated sizes for the first integrator.*

9.2 Electrical Design

This section describes electrical design chip implementation issues of the modulator.

9.2.1 Bias

Each integrator requires only two bias voltages. The opamp CM output voltage Vop_{CM} which is set to $V_{DD}/2$ and supplied off-chip. Its exact value is not critical to the circuit operation. The other bias voltage V_{bias} is needed in the opamp to bias transistors M7, M8, M9, M10 and M12 as shown in Fig. 8.16. V_{bias} is also used in the CMFB network as shown in Fig. 7.12. However, since considerable switching noise is introduced from the SC CMFB network, other nodes of the bias circuit can be disturbed if V_{bias} is used in the CMFB [Rezania95]. In order to avoid this, two separated bias voltages are generated using an on-chip bias circuit shown in Fig. 9.1. The diode-connected transistor MB generates the reference current I_B, determined by the value of R_B, which is then duplicated to generate the bias currents. For each opamp, two biasing branches are used to generate V_{bias} which is used directly in the opamp, and Vb_{cmfb} which is used in the CMFB SC network. Normally both voltages have the same value, however, in this design Vb_{cmfb} is set to a different value determined by transistor M10(M12) as follows: Decreasing V_{GS} of transistor M10 results in the increase of the opamp output swing given by equation (8.59). However, this also results in large transistor sizes for the same current which increases both the parasitic opamp load capacitance and the input capacitance C_G seen at the V_{cmfb} opamp input given by equation (8.75). Increasing C_G degrades the CMFB feedback factor β_{cmfb} given by equation (8.76) which in turn increases the CMFB settling. To overcome this problem, C_1 is chosen such that

$$C_1 = 5C_G \tag{9.1}$$

A compromise is thus made in choosing the value of Vb_{cmfb}.

It is important to use high output impedance current sources to reduce errors in the current values due to differences in drain voltages. Typically, cascoded current mirrors are used. However, under very low-voltage operation, this is not possible. Therefore, long channel transistors are used wherever possible. This enhances the output impedance as well as transistor matching.

Table 9.1 shows the gate-source voltages as well as the corresponding calculated transistor sizes of the bias network used for the first integrator. Same transistor lengths are used for current mirror transistors in the bias network and the opamp. Note that for MB4, a small transistor length is used to reduce parasitic capacitances of the mirror transistor M10(M12) in the opamp as discussed above.

Separate biasing voltages are used for each opamp by adding two current branches to produce both V_{bias} and Vb_{cmfb} per opamp. The transistor MB is physically placed close to the current branches of the first integrator to improve current matching of this critical stage.

9.2.2 Clock Generation

Fig. 9.2 shows the timing diagram of the required clock phases for the modulator. This includes two non-overlapping clock phases ϕ_1 and ϕ_2, together with their delayed versions ϕ_{1d} and ϕ_{2d}, and an inverted $\bar{\phi}_{2d}$ needed to drive the CMOS switch used in the feedback DAC as shown in Fig. 8.14. Two additional overlapping phases ϕ_{ch1} and ϕ_{ch2} are needed for chopper stabilization (see section 7.5.4). They are used to drive the input chopper switches of the first stage opamp as shown in Fig. 7.12 as well as the cascode transistors M31(M41) and M32(M42) inside the first stage opamp as shown in Fig. 8.16. The chopper clock phases must be stable during the integration and sampling phases. As shown from Fig. 8.14, the integration phase ends when ϕ_2 goes down while the sampling one begins when ϕ_{1d} goes up. ϕ_2 and ϕ_{1d} also control the access to the amplifier inputs and outputs respectively. Thus all ϕ_{ch1} and ϕ_{ch2} transitions must take place in the non-overlap time between ϕ_{1d} and ϕ_2 as shown in Fig. 9.2.

A standard non-overlapping clock generator is used to produce the required clock phases given an external clock signal having the required sampling period T_s. It consists of two cross-

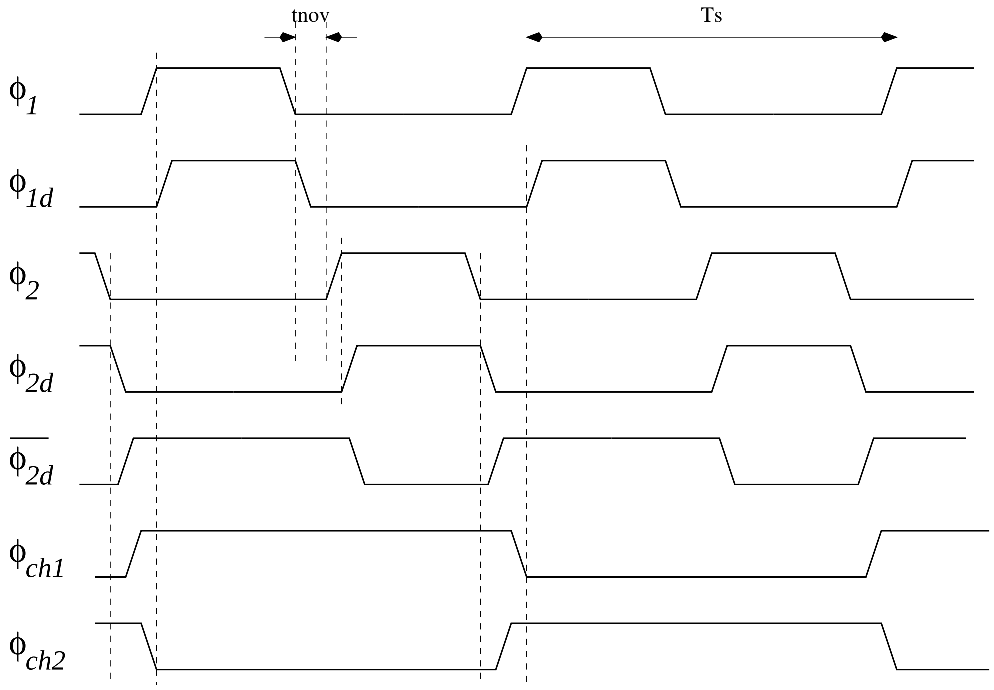


Figure 9.2: *Clock phases timing diagram*.

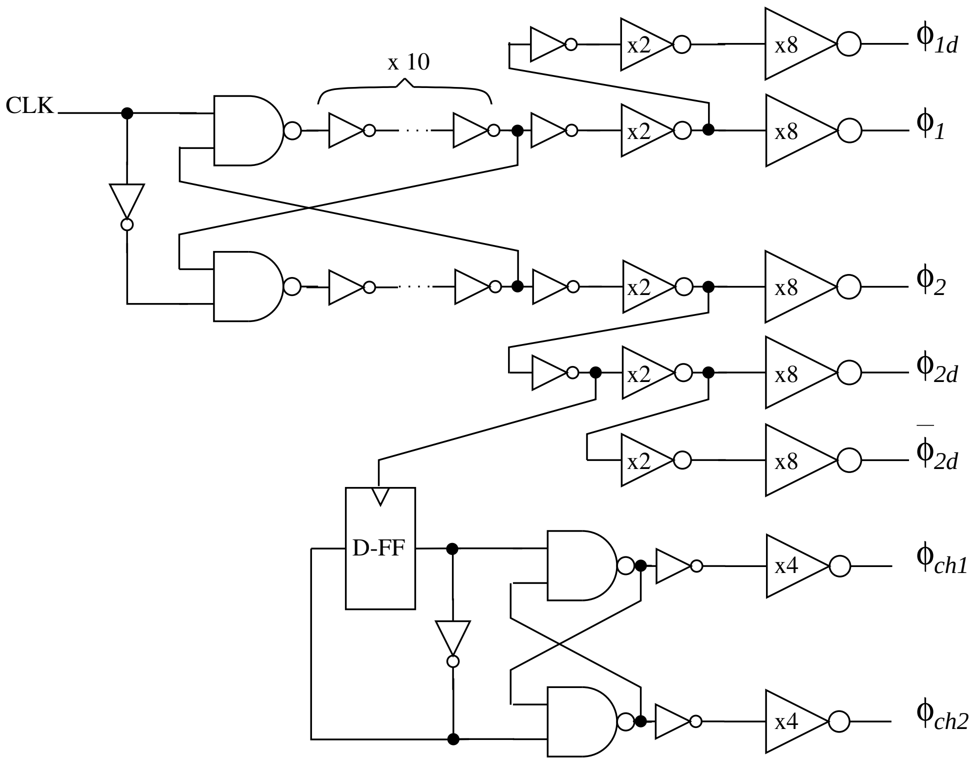


Figure 9.3: *Clock generator*.

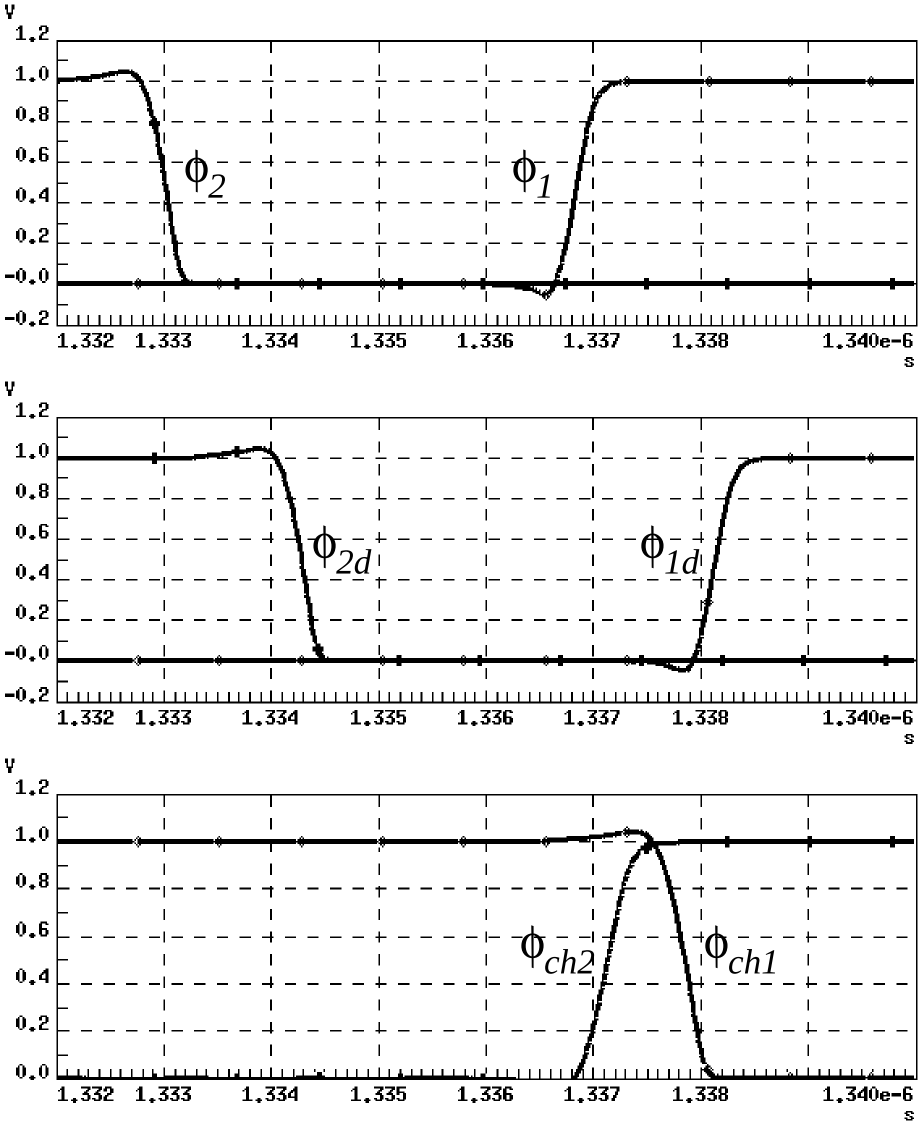


Figure 9.4: *Clock simulation results using worst-case transistor model.*

coupled NAND gates as shown in Fig. 9.3. Ten inverters are used to introduce the delay required to achieve t_{nov}. A frequency divider based on a D-FF is then used together with another two cross-coupled NAND gates without delay to generate the extra chopper phases. All clock phases are buffered with large inverters to drive the on-chip clock buses.

Fig. 9.4 shows the simulation results of the clock generator circuit shown in Fig. 9.3. Worst-case transistor models are used during the simulation to consider worst-case delays. The figure shows first ϕ_1 and ϕ_2. As can be seen the non-overlapping time t_{nov} is around 3.5ns. Their delayed versions ϕ_{1d} and ϕ_{2d} are also shown, the time delay is around 1ns. Finally, the chopper clock phases ϕ_{ch1} and ϕ_{ch2} are shown. Fig. 9.4 demonstrates that all the conditions on the clock phases

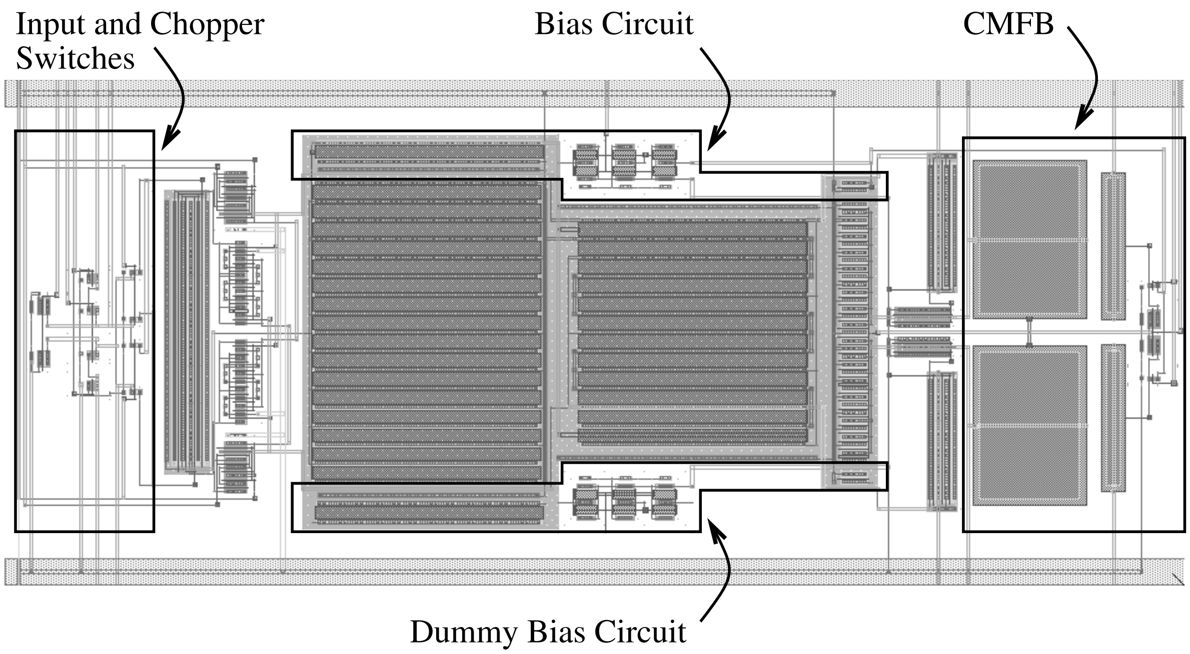


Figure 9.5: *Layout of the first stage amplifier.*

discussed above are well satisfied. This has also been verified using the typical and fast transistor models.

9.3 Physical Design

This section describes physical design implementation issues including the used process technology and layout.

9.3.1 Technology

The modulator has been implemented in a 0.35-μm standard CMOS process with two-level poly, five-level metal, and twin-tub. Only three levels of metal are used in this design. The fabrication process is developed for the design of analog circuits operating at a power supply voltage of 3.3 Volt. However, the implemented circuit was designed for 1-Volt reliable operation. The threshold voltages for wide/long n and p-transistors are 580 mV and 600 mV respectively. The process has highly linear poly-1/poly-2 capacitors with specific capacitance of 1.1 $\pm$ 0.15 fF/μm^2.

9.3.2 Layout

Layout has been generated hierarchically using the layout language CAIRO described in chapter 5. With the help of the internal device generators (transistors, differential pairs, capacitor

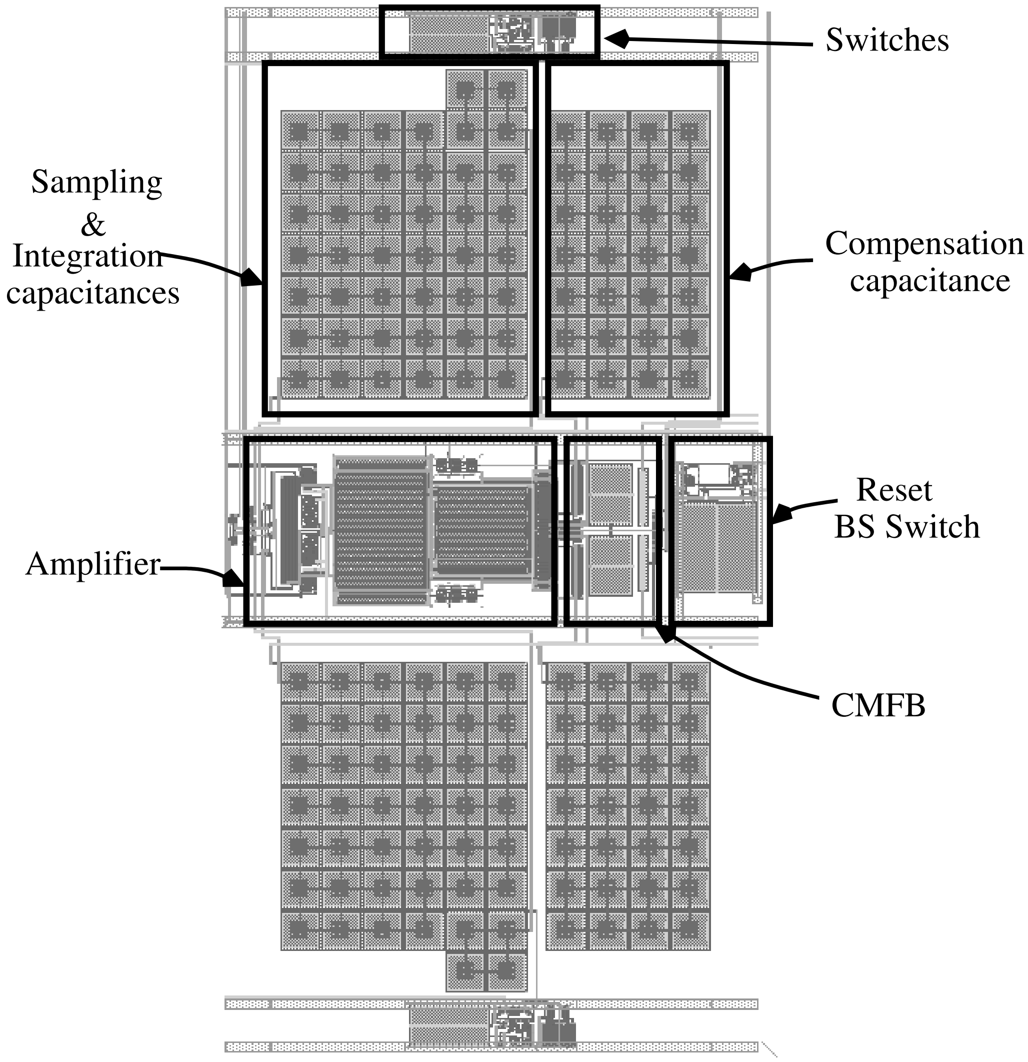


Figure 9.6: *Layout of the first integrator*.

arrays, ...) which take into consideration analog-specific constraints, the symmetrical relative placement functions based on *slicing structures*, the area optimization algorithm that automatically determines the number of folds for each transistor and the layout parasitic estimation mode, CAIRO has efficiently contributed to optimizing the generated layout.

The code corresponding to each block has been developed separately and then instantiated in higher blocks. The code corresponding to repeatedly used blocks like the bootstrapped switch is thus re-used several times each with a different set of transistor sizes. The fact that the layout is generated automatically starting from the code, and that it is independent of transistor sizes offers a great flexibility to size each switch separately in order to optimize switch dimensions and reduce charge injection effects.

Fig. 9.5 shows the layout of the first-stage amplifier shown in Fig. 8.16. The layout is shown to

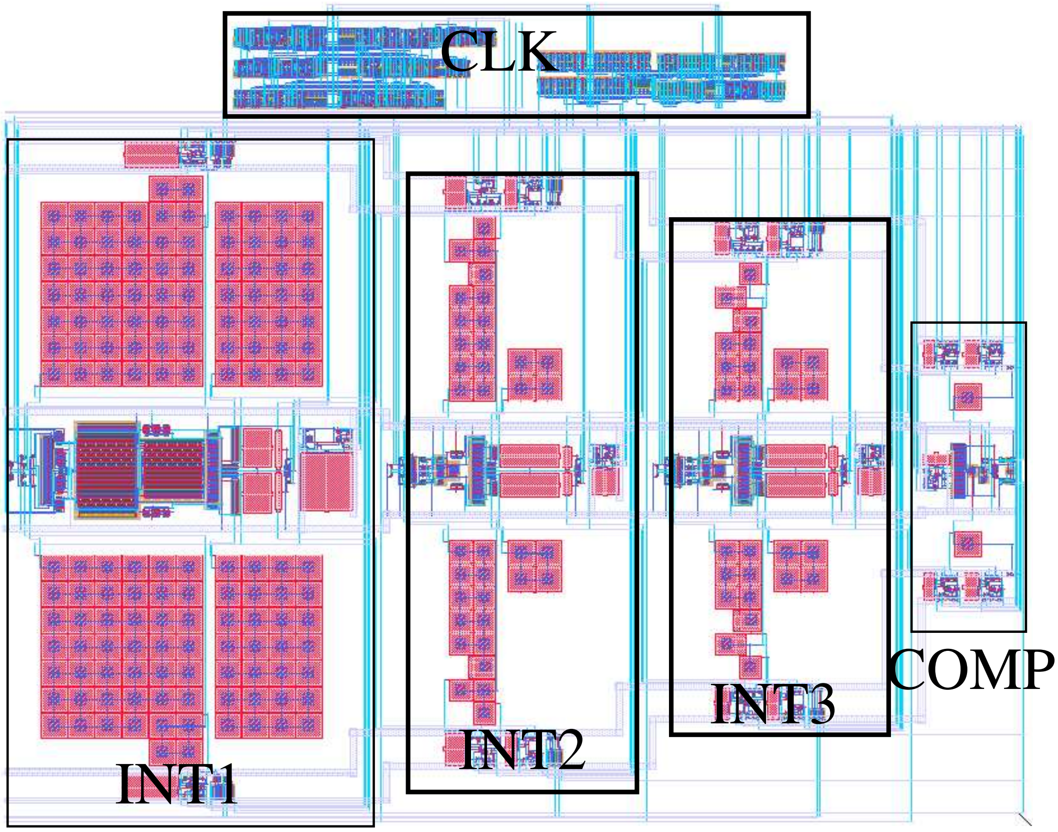


Figure 9.7: *Layout of the third-order modulator.*

be completely symmetrical which is an important consideration in fully differential circuits. The layout also shows the bias circuit shown in Fig. 9.1. Transistors MB3 and MB4 are placed physically besides the corresponding mirror transistors M7 and M10 respectively inside the opamp to enhance transistor matching. An additional dummy bias circuit is placed on the other side of the amplifier to preserve layout symmetry and avoid boundary dependent etching of polysilicon gates which leads to transistor mismatch. The input and chopper switches are also shown together with the CMFB SC circuit.

Fig. 9.6 shows the first integrator layout. Capacitor arrays has been generated by CAIRO built-in capacitor generator which is also capable of handling non-integer capacitor ratios. Signal carrying switches, including bootstrapped switches, have been placed on both ends away from the amplifier to avoid switching noise injection in the signal path. A reset bootstrapped switch has been also added at the output of each integrator which shorts the two differential outputs thus resetting the integrator in case of modulator instability.

The digital part has been synthesized automatically using the ALLIANCE CAD system [LIP] and incorporated as a black box in the language code. Both analog and digital parts use the symbolic layout approach described in section 5.9, such that the layout can be easily ported to another process with the minimum effort. This helps to re-use the *same* code in a future re-design of a similar circuit using a different process without much effort spent on the layout. Two different supplies are used for the analog (V_{DDA}/V_{SSA}) and the digital parts (V_{DD}/V_{SS}). This is used to prevent the digital switching noise from being injected in the signal path. Fig. 9.7 shows the complete modulator layout. Different integrator areas are due to integrator scaling.

After the complete design, and as a final verification step, transistor-level electrical simulation has been done on the extracted complete netlist. Long simulation time is, however, unavoidable. This is due to the presence of two frequencies with a two order of magnitude difference; the sampling clock frequency and the slowly varying input signal. As a result the circuit has to be simulated over tens of thousands of clock cycles in order to obtain the SNR. In addition, numerical algorithms used by the simulator contain inherent small amount of calculation error which adds a certain simulator noise to the output signal. In order to avoid this artificial noise component, the tolerance of the simulator must be limited below the modulator accuracy, which further increases the simulation time due to the increase of the number of iterations per time step. Simulation of the whole modulator takes around 3-4 days on a Sun Ultra-5 workstation. This makes this simulation an unpractical tool during circuit design. It should be also noted that, such simulations do not take into account circuit noise since transient analysis does not include the corresponding component models. This means that only quantization noise leakage information can be deduced, i.e. the SQNR.

9.4 Experimental Results

Fig. 9.8 shows the chip photograph. The core area excluding bonding pads is 0.9×0.7 mm^2. The chip has been packaged in a ceramic leadless chip carrier package (LCC 44). It has used 16 pins. This section presents the prototype test procedure as well as some obtained measurement results. Finally, some comparisons are given with recent $\Delta\Sigma$ implementations.

9.4.1 Test Setup

Fig. 9.9 shows a circuit diagram representing the test setup used to measure the prototype circuit performance.

A high linearity sinusoidal source (Bruel & Kjaer 1051) is used for the input signal. On the test board, this signal is converted from single-ended to a differential one using the SM-LP-5001 surface mount transformer from Bourns. The signal is then shifted to a CM level equal to $Vop_{CM} = V_{DD}/2$. The DC supply is drawn from a 1.5V battery followed by a potentiometer to adjust the circuit supply to 1V. A small series resistance is used to measure the drawn current. Two batterys are used, the first for the analog references: V_{DDA}, $Vref_p$, IB, and Vop_{CM}, while the second battery is used for the digital supply V_{DD}. 2.2 μF decoupling capacitors are connected between all bias voltages and V_{SS}.

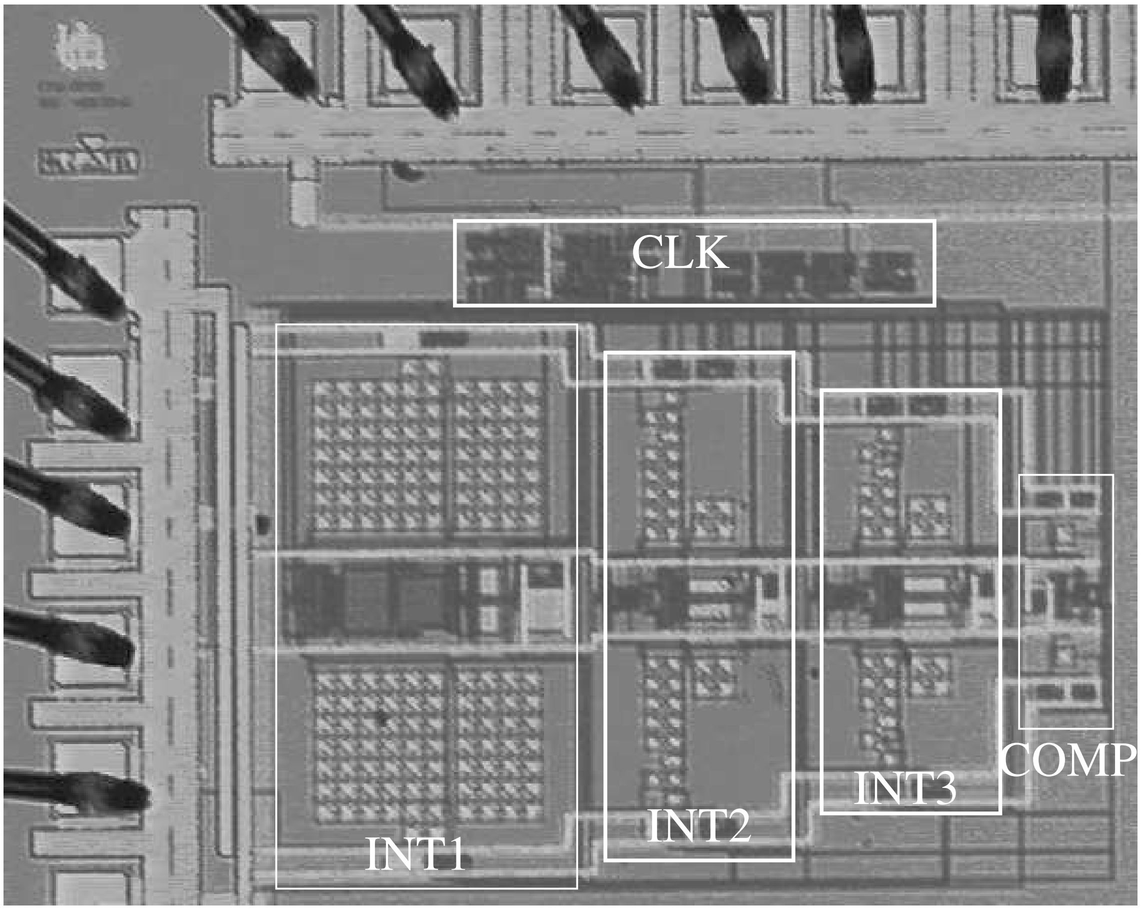


Figure 9.8: *Chip die photo.*

The external clock is supplied from the HP 33120A function/arbitrary waveform generator. The duty cycle of the clock is externally adjusted. A 50Ω resistance assures matching at the circuit clock input.

Two digital signals are transferred to the logic analysis system, the modulator output and the clock signal used for data acquisition. A special arrangement is used at the output to separate the ground noise of the digital measuring equipment including the PC from the circuit ground. This separation has been achieved through optocouplers. The 1-V digital output is first buffered using the CD74AC05 open-drain inverters from Harris Semiconductor. The inverters are powered using the 1.5-V battery. The photodiode of the HCPL-2630 optocoupler, from HP, loads the drain output of the inverter as shown in Fig. 9.9. The open-collector photo detector of the optocoupler is powered using a 5-V supply and has a separate ground on the test board. This signal is then transferred to the HP logic analysis system 16500B and then to the PC via the PCI-GPIB bus. The LabVIEW software then performs FFT, windowing and graphical manipulations.

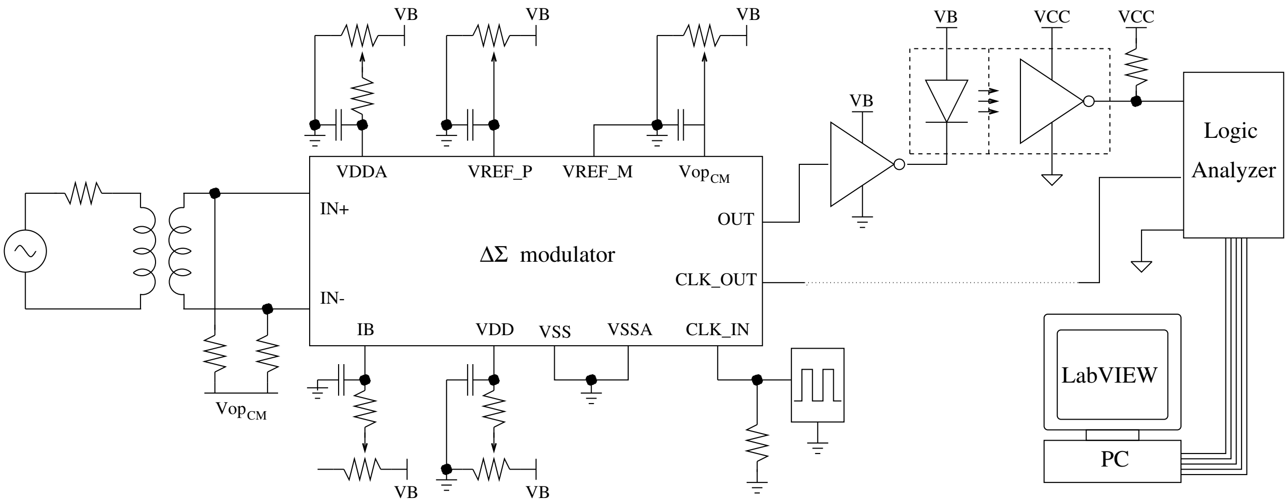
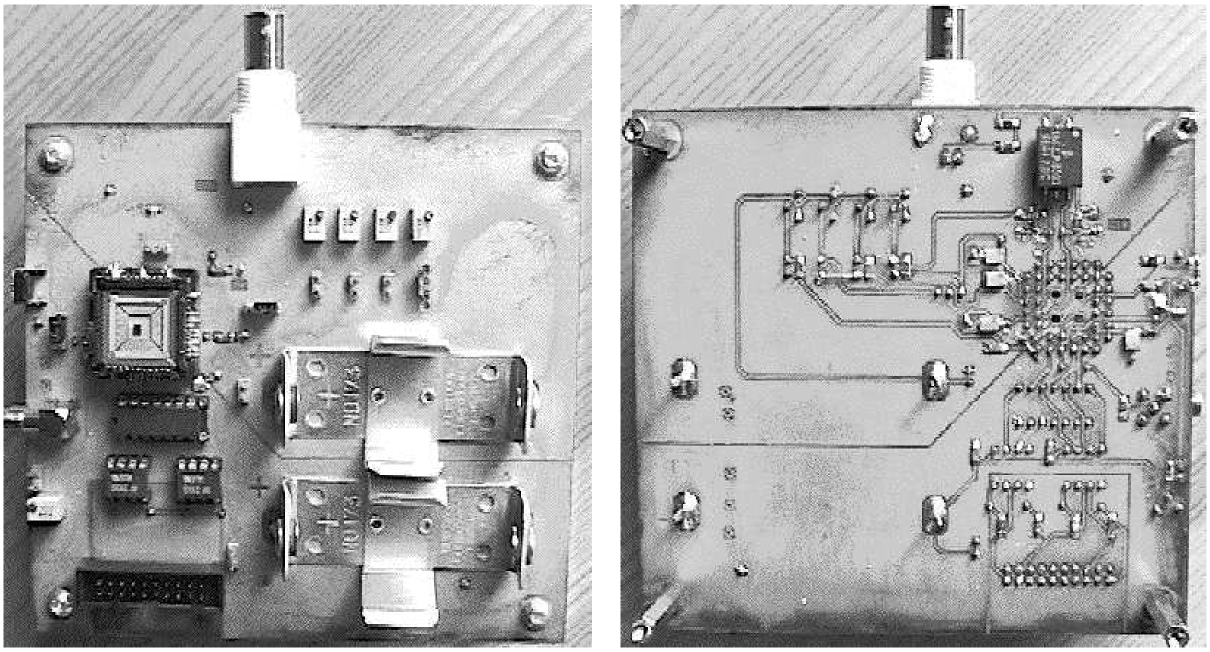


Figure 9.9: *Circuit diagram of the test setup. VB=1.5V and VCC=5V.*



(a) (b)

Figure 9.10: *PCB used for prototype test.*

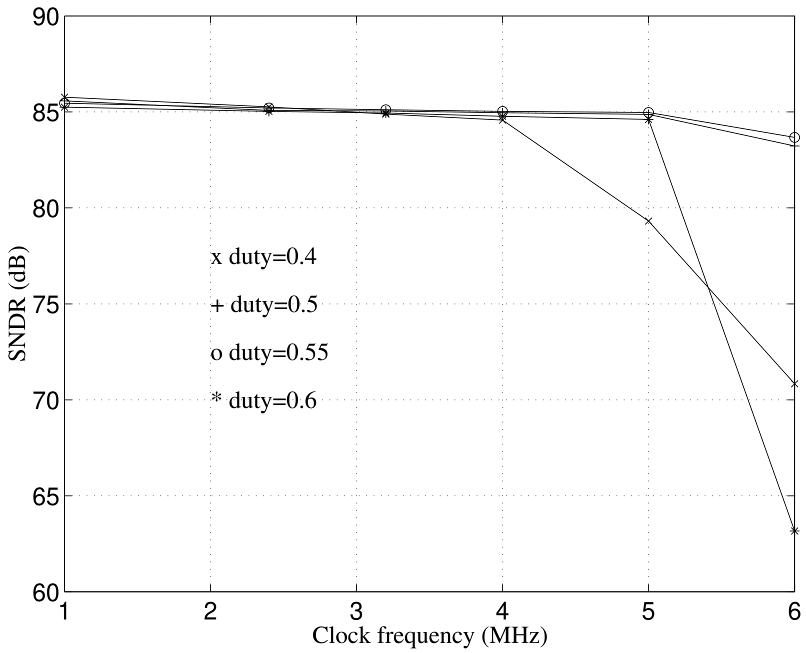


Figure 9.11: *SNDR versus clock frequency for different integration phase duty cycles for an input signal at -6dB.*

9.4.2 Measurement Results

Fig. 9.11 shows the SNDR ratio for an input signal at -6dB versus the clock frequency using different clock integration phase duty cycles. For the 0.4 duty cycle the SNDR roll-off takes place earlier due to incomplete settling in the integration phase. The same is true for the 0.6 duty cycle, but in this case, settling degradation happens in the sampling phase. For duty cycles of 0.5 and 0.55, the SNDR remains practically constant until a clock frequency of 5 MHz. Since the converter has been designed for a clock frequency of 3.2 MHz (refer to table 8.3), this means that the settling behavior of the integrator has been over-sized by choosing stringent opamp performance specifications during the high-level synthesis step. For the rest of measurements, a single clock at 5 MHz with a duty cycle of 0.55 is used to drive the converter.

The modulator operates at a V_{DD} of 1V and dissipates 950 μW. 60% of the total power is consumed by the first integrator. The two reference voltages $Vref_p$ and $Vref_m$ are set to 1V and 0V respectively. With an OSR of 100, the signal bandwidth is 25 kHz. Fig. 9.12 shows the measured SNR and SNDR vs. the relative input amplitude (V_{in}/V_{ref}). An input sinusoidal signal at 3.2 kHz was used to produce these plots. It is apparent that the modulator achieves a dynamic range of 88 dB, a peak SNR and a peak SNDR of 87 dB and 85 dB respectively.

Fig. 9.13 shows the measured output spectrum for a 3.2 kHz input signal at -6 dB relative input level, while Fig. 9.14 shows the baseband measured output spectrum for the same input

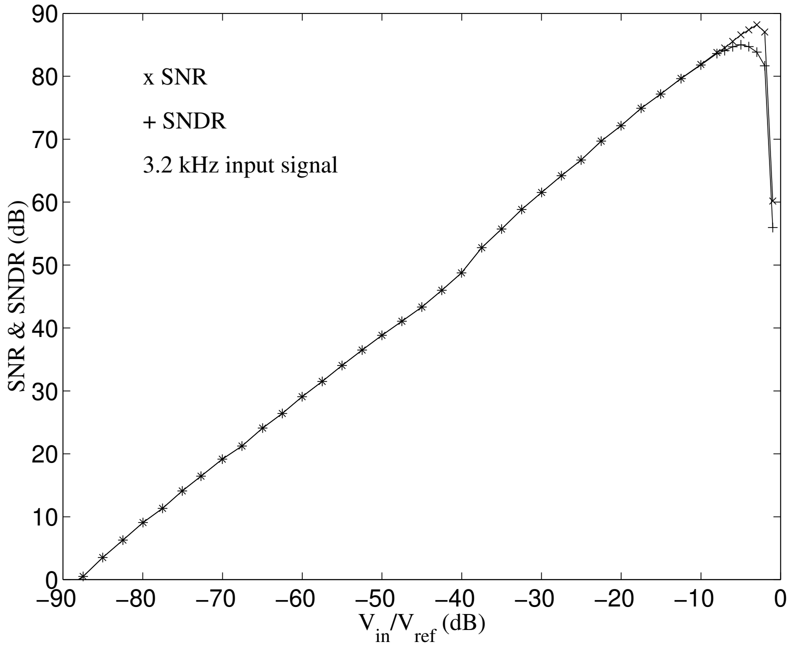


Figure 9.12: *Measured SNR & SNDR.*

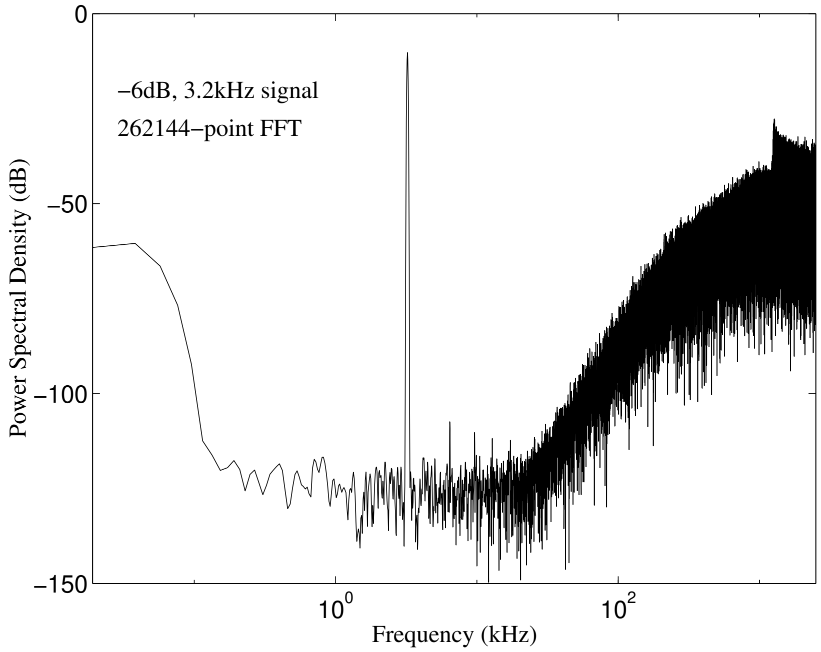


Figure 9.13: *Measured output spectrum.*

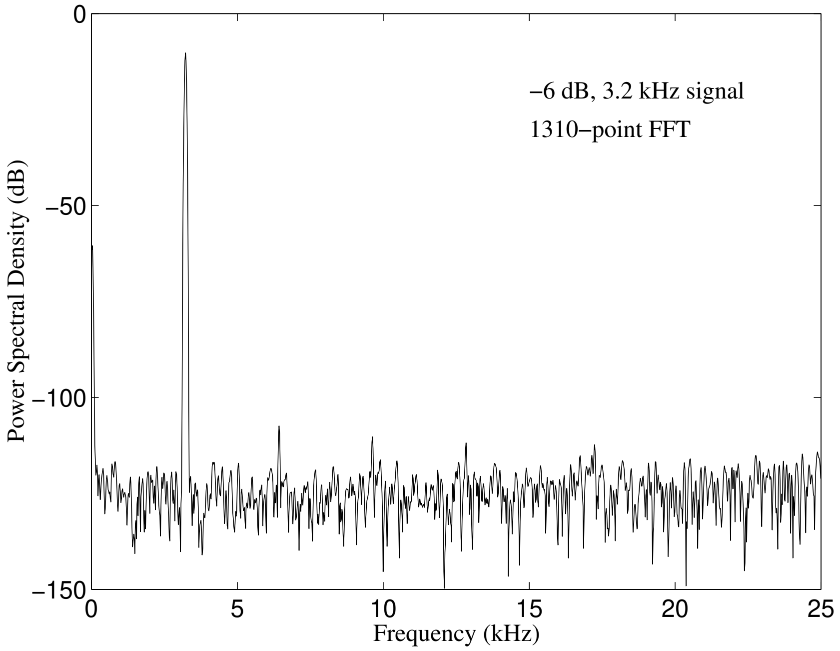


Figure 9.14: *Measured output baseband spectrum for a large amplitude signal.*

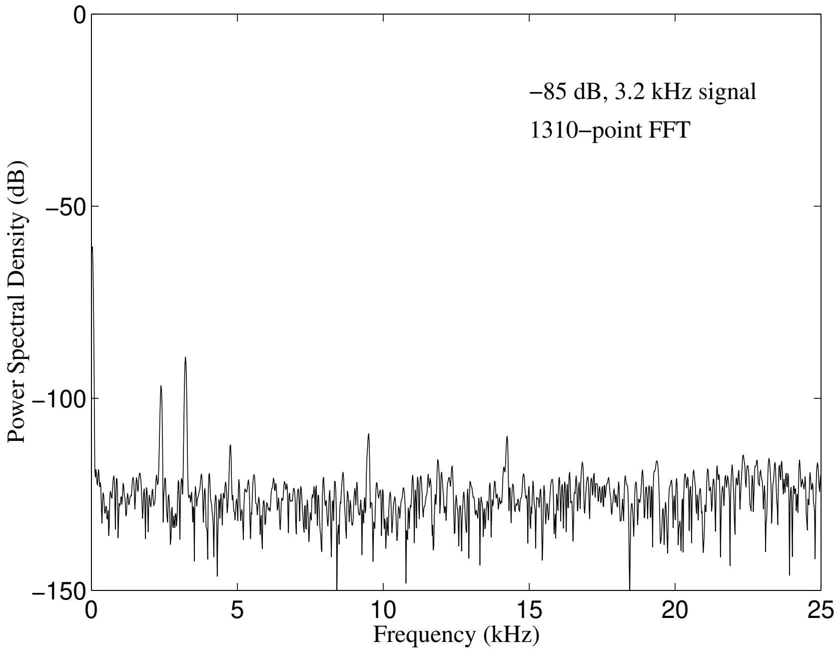


Figure 9.15: *Measured output baseband spectrum for a small amplitude signal.*

Supply Voltage	1V
Reference Voltage	1V
Dynamic Range	88dB
Peak SNR / SNDR	87dB / 85 dB
Number of bits	14
Oversampling Ratio	100
Sampling Rate	5MHz
Signal Bandwidth	25kHz
Power Consumption	950 μW
Figure of Merit $\times 10^6$	275
Die Area	0.9mm $\times$ 0.7mm
Technology	0.35-μm CMOS TMDP

Table 9.2: *Measured converter performance summary, refer to tables 8.3 and 8.5 for comparison.*

	C1	C2	C3	C4	C5
SNR (dB)	87	87	87	87	83
SNDR (dB)	85	85	83	83	80
I_B (μA)	8.0	10.0	10.0	9.0	10.0
V_{DD} (V)	1.0	1.0	1.0	1.0	1.1

Table 9.3: *Measured circuits statistics.*

signal. The frequency independence of the noise floor indicates that the modulator's performance is thermal-noise limited rather than quantization-noise limited as depicted in section 8.5.2.1. Fig. 9.15 shows the baseband measured output spectrum for a small amplitude 3.2 kHz input signal at -85 dB relative input level. Some tones begin to appear at very small amplitudes, this can be attributed to the use of chopper stabilization for 1/f noise reduction (see section 7.5.4). When the chopper clock at half the sampling frequency is present, intermodulation products are almost inevitable, due to reasons such as capacitive coupling between the modulator output and the chopper clock [Wang00]. This causes high-energy high-frequency tones to fold back to signal band when a small DC signal is applied to the modulator. The measured performance is summarized in table 9.2 (compare to the calculated performance values in table 8.5).

Table 9.3 shows the measured maximum SNR and SNDR of the fabricated circuits. Also shown are the bias conditions. A slight modification of the bias current is needed to obtain maximum performance. Circuit C5 didn't work at 1.0V, however, it gave less than expected results by raising the supply voltage to 1.1V. It can be deduced that further improvements can be obtained by optimizing the opamp design with respect to the variation of process parameters, in order to obtain

	[Rabii97]	[Peluso98b]	[Coban99]	This work
Type	SC-VM	SO	SC-VM	SC-BS
V_{DD}(V)	1.8	0.9	1.5	1.0
DR(dB)	99	77	98	88
SNDR(dB)	95	62	88	85
BW(kHz)	25	16	20	25
Pc(mW)	2500	40	1000	950
$4kT.DR.BW/P_c$	1316	330	2091	275
$4kT.SNDR.BW/P_c$	533	11	209	138

Table 9.4: *Performance comparison*.

a more robust design. In addition, a timing problem was observed in some circuits (C3-C5), the output clock signal used for signal acquisition (see Fig. 9.9) was found not to be in perfect synchronization with the output data stream. This can lead to a degradation of the measured SNR. However, no further investigations could be done due to the absence of appropriate output signal nodes, for example other clock phases, the other complementary output, ...

9.4.3 Performance Comparison

Table 9.4 compares recent low-voltage $\Delta\Sigma$ implementations. The first [Rabii97] and the third [Coban99] ones use a switched-capacitor implementation with a special voltage-multiplication circuit for the switches (SC-VM), while the second [Peluso98b] one is a switched-opamp (SO) implementation. The last column represents the results obtained in this work using switched-capacitor with the switch-bootstrapping technique (SC-BS). The SO and BS implementations have the lowest supply voltage around 1V. The BS implementation has a higher speed, thus a higher BW. In addition, the BS technique enhances the circuit linearity which leads to a considerable improvement in the SNDR. Thus, it also has a higher dynamic range. To be noted is the very low power consumption of the SO technique with respect to the other implementations which all lie in the same order of magnitude.

The power efficiency of various A/D converters are often compared using the following figure of merit [Rabii97]:

$$FM = \frac{4kT \times DR \times BW}{P_c} \tag{9.2}$$

where DR is the dynamic range expressed as a ratio, BW is the signal bandwidth and P_c is the total power dissipation of the converter. The figure of merit of the VM implementations are in the same order of magnitude as they have a relatively higher supply voltage, thus more design flexibility. Based on the dynamic range, the figure of merit of the actual design is slightly less than that of the

SO. However, if we consider the distortion performance in the figure of merit as shown in the last row of table 9.4, the BS technique is shown to have a considerably higher figure of merit.

9.5 Design Reuse

In this section, two modulator designs in a different process are described. Both designs are based on the knowledge database and heuristics developed during the first modulator design. They have been designed from high-level specifications to the layout. Complete circuit-level simulations of the extracted netlist have been run to verify and evaluate the performance characteristics in presence of parasitics. However, they have not been fabricated due to time limits.

9.5.1 Another Process Technology

Since the modulator is targeted for 1-Volt reliable operation of future very low-voltage technologies, a real low-voltage process has to be used for circuit implementation. Another available low-voltage process at the time of circuit implementation was an another 0.35-μm process from a *different* foundry. The process is a standard 3.3-V p-substrate CMOS process with double level poly and triple metal. The threshold voltages for wide/long n and p-transistors are 500 mV and 650 mV respectively. The fabrication process has a highly linear poly-1/poly-2 capacitors with specific capacitance of 0.86 ± 0.1 fF/μm^2.

9.5.2 Process Migration of the Same Modulator

Using the same design procedure described in section 8.2, the same modulator shown in Fig. 8.14 has been migrated to the new process with the same performance specifications as follows:

1. The same high-level synthesis results (section 8.3) and performance parameter mapping specification values (section 8.4) have been applied.

2. The same integrator sizing procedure is used to size the integrators. Since this sizing is done interactively (section 8.5.2.3), this is the most time-consuming step. However, COMDIAC allows rapid design-space exploration in the presence of parasitics.

3. The same switch sizing procedure (section 8.5.3.2) is used to size separately all switches in the circuit.

4. The same layout templates have been reused both for parasitics estimation and layout generation.

Since COMDIAC allows to set all bias voltages and currents, during re-sizing all these values are kept the same starting from the *initial* design as a starting point. Further design optimization was then done to enhance the circuit performance in the new process. It should be also noted that,

	Technology 1	Technology 2
Technology	twin-tub 2P5M	p-sub 2P3M
Min. gate length	0.35μ	0.3μ
n/p threshold	580/600mV	500/650mV
Poly capacitor	1.1 ± 0.15fF/μm^2	0.86 ± 0.1fF/μm^2
1^{st} integrator P_c	570μW	610μW
Total P_c	950μW	820μW
Modulator Area	0.9×0.7mm^2	1.0×0.9mm^2

Table 9.5: *Modulator in two technologies.*

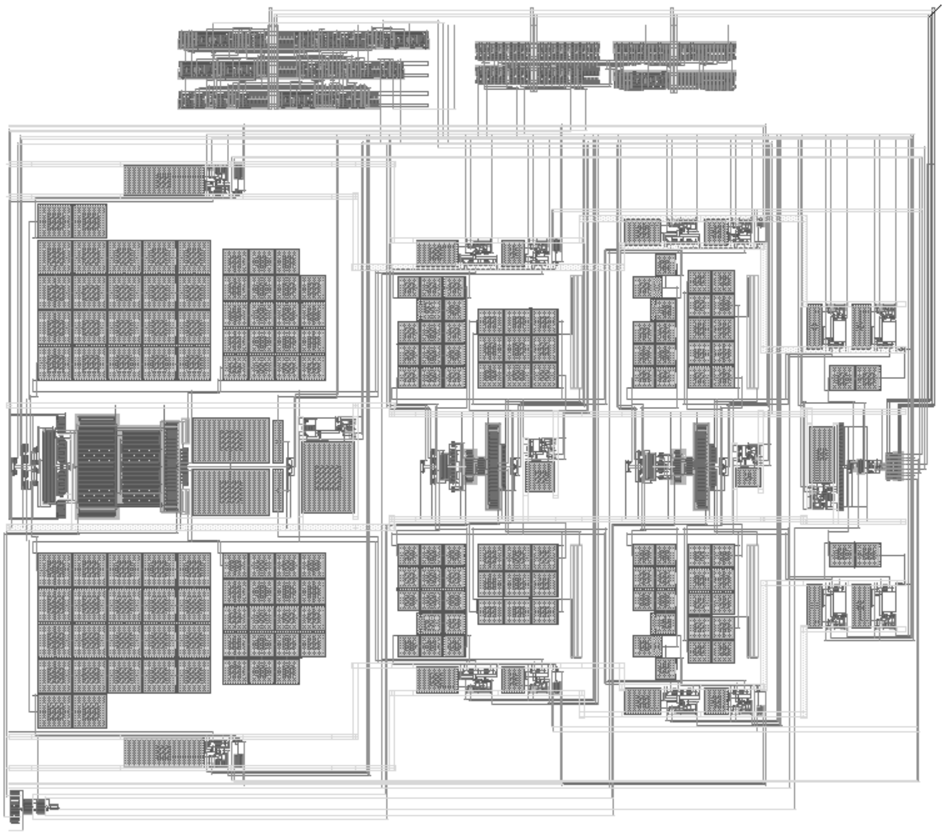


Figure 9.16: *Layout of the third-order modulator in another 0.35-μ process.*

bias voltages are all set relatively, i.e. we use $V_{EG} = V_{GS} - V_{th}$ rather V_{GS}, and $V_{ED} = V_{DS} - V_{dsat}$ rather than V_{DS}, such that the bias point is kept independent of process values.

From the schematic point of view, the only difference was the use of a real compensation poly resistance in series with the compensation capacitance of the amplifier instead of implementing it with a transistor as shown in Fig. 8.16. The reason behind this was the high worst-case value of the p-transistor threshold voltage in this process (around 0.8 V). Table 9.5 shows a comparison of the

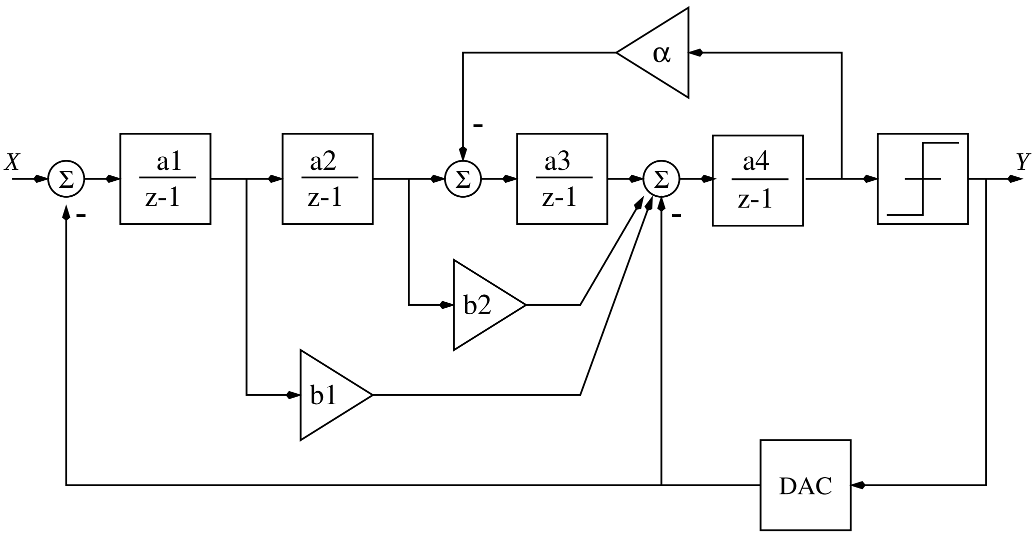


Figure 9.17: *Fourth-order modulator topology.* $a_1 = 1/2$, $a_2 = 2/25$, $a_3 = 1/10$, $a_4 = 4/5$, $b_1 = 6/5$, $b_2 = 1$, $\alpha = 1/6$.

two implementations. The lower specific capacitance of the second process has resulted in larger capacitance area and consequently larger parasitic capacitance values. This increased the load on the first amplifier and so is its power consumption. While for the first design the same opamp has been used for the second and third integrators, in this design two different amplifiers, optimized for each stage, have been used which enabled a reduction of the total power consumption.

A variation in the layout was to group the n-transistor current sources (transistors MB, MB1, MB2, ...) of the bias circuit of all opamps, shown in Fig. 9.1, in a single block while placing the p-transistors (MB3, MB4, ...) inside the corresponding opamp. This corresponds to routing bias currents from the bias circuit to different opamps instead of routing bias voltages as shown in Fig 9.1. Routing bias currents has the advantage of being independent of the IR drops due to routing which could create current errors.

The complete layout of the modulator is shown in Fig. 9.16. The complete re-design and layout generation was completed in only one week due to the re-use of sizing plans and layout code. Due to the use of the same layout templates, the new modulator has the same floorplan as the previous one. The obtained overall area is 1.0 mm $\times$ 0.9 mm. Slight increase in the area with respect to the original modulator is due to the lower capacitance per unit area of the available poly capacitor, which leads to larger capacitor areas.

9.5.3 Fourth-Order Modulator

In order to further investigate another dimension of design reuse, a completely different modulator topology has been used to design another modulator using the same low-voltage building blocks as the previous ones thus the same *design knowledge*. A single-loop fourth-order topol-

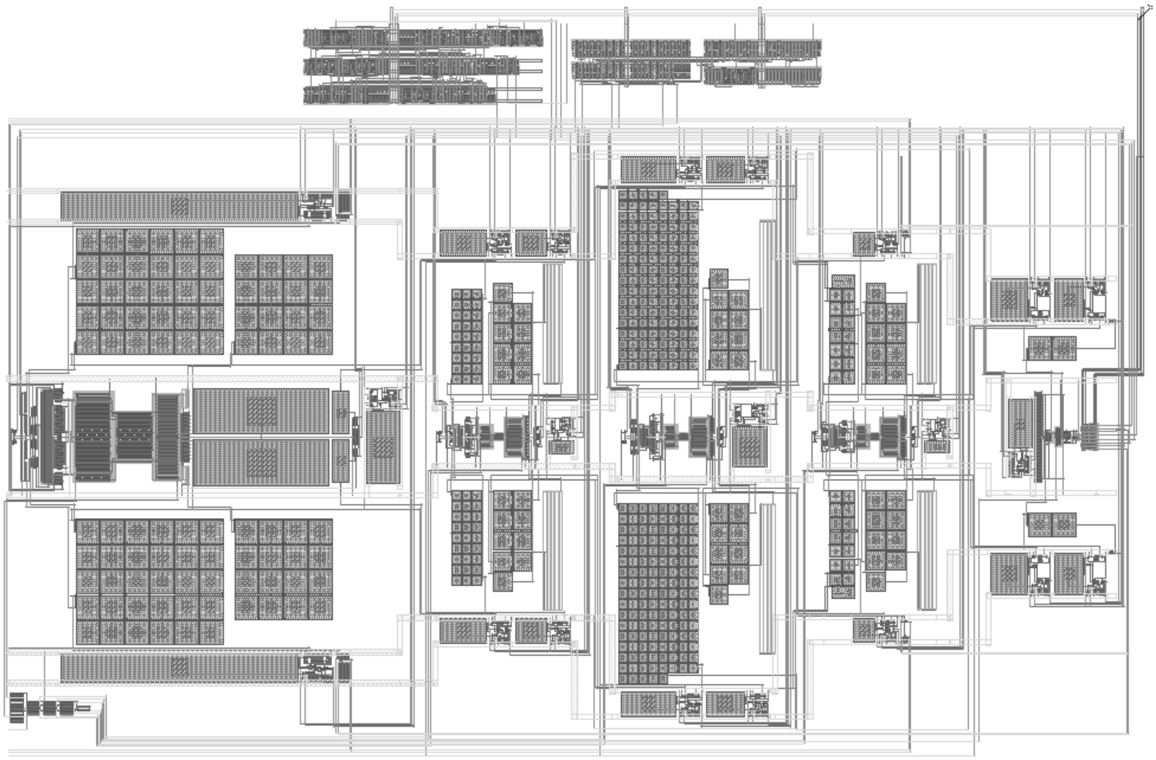


Figure 9.18: *Layout of the fourth-order modulator in 0.35-μ process.*

ogy [Coban99] has been chosen. The modulator employs a mixed loop topology having both feedforward and feedback paths as shown in Fig. 9.17. The same design procedure, described in section 8.2, has been employed during the design. This modulator topology has a lower integrator output swings, making it suitable for low-voltage implementation. An additional benefit of reduced output swing (especially that of the first stage) is that the first integrator gain can be chosen with less concern about saturating the first integrator output [Coban99]. Here, it is selected to optimize the op-amp referred noise (see section 8.3.2) thus minimizing the power dissipation of the first integrator. Performance parameter mapping has been performed using the same block behavioral models described in section 8.4. The later 0.35-μ process has been used. Simulations show that for a supply voltage of 1V, an OSR of 70 and a signal BW of 20kHz, the modulator achieves a DR of 90dB.

Fig. 9.18 shows the complete layout of the modulator. The overall area is 1.3 mm $\times$ 0.85 mm. Due to the re-use of the same layout templates, the modulator floorplan is very close to the previous ones except for the additional integrator stage. The complete design and layout generation was completed in two weeks.

9.6 Conclusions

In this chapter, the complete design flow of a 1V 1mW $\Delta\Sigma$ modulator with 88dB DR in a 25kHz BW has been presented. Very low-voltage SC design circuit techniques introduced in chapter 7 have allowed the 1V circuit operation. This includes the special low-voltage bootstrapped switch and the low-voltage differential amplifier.

Bootstrapped switches have allowed very low voltage SC in a standard CMOS process. Since they keep a signal-independent gate-source switch drive, they also improve charge injection and linearity performance of the SC circuit. This has contributed to the suppression of distortion sources and the relatively high realized SNDR.

The modified low-voltage two-stage folded cascode opamp has allowed to use a standard passive CMFB SC network thus reducing the overall power consumption. It has also allowed to use chopper stabilization to reduce the 1/f noise.

Obtained results show the feasibility of very low-voltage high performance circuits using standard SC techniques.

Sizing has been done using the knowledge-based sizing tool COMDIAC. A sizing plan and the corresponding performance evaluation equations have been developed for each block and implemented in the tool. The corresponding layout templates have also been built using the layout language CAIRO and have been used concurrently during sizing to calculate the associated circuit parasitics. The final layout has also been generated using the same templates.

The design knowledge and heuristics captured in the above tools during the design have allowed to efficiently re-design another two modulators in a relatively small amount of time. The layout-oriented design methodology, besides its efficiency in parasitics calculation during the first design, has also contributed to easily migrate the design to another process taking into account the new process physical parameters.

Chapter 10

Conclusion and Future Directions

10.1 Conclusion

It is clear that as mixed-signal SoC's are getting more and more dense, thanks to the continuous shrink of device dimensions allowed by new technologies, the only way to efficiently design such heavily packed chips is by embedding re-usable IP cores. IP blocks also increase the probability of first pass silicon and drastically reduces time-to-market. However, design reuse of an analog block is far from being a push-button operation. Some CAD tools need to be used in order to *adapt* analog cores for any new application, or simply to migrate it to other processes. These tools must be able to handle and transfer both design experience and heuristics from the original design to subsequent IP core versions.

In this work, a design methodology for analog circuit design for reuse, based on the integration of both electrical and physical design, has been presented. The methodology is based on capturing design experiences in *design plans* that contain relevant design steps, using two knowledge-based CAD tools. On one side, circuit sizing procedures, coded in the knowledge-based sizing environment COMDIAC, focus on the most significant performance characteristics and design heuristics while leaving the possibility to the designer to control design details. This is coupled to a fast, yet very accurate, performance evaluation based on pre-derived equations and built-in detailed SPICE-like transistor model equations. On the other side, the methodology relies on a technology- and size-independent *layout templates* that contain physical layout information related to the circuit, written in the layout language CAIRO. CAIRO incorporates procedural device generators where several analog layout constraints are studied and taken into consideration using efficient algorithms, in addition to simple and fast area optimization. Physical layout constraints including layout parasitics, global aspect ratio and circuit reliability design rules are taken into account *during* circuit sizing. The methodology and the associated tools thus contribute during the design phase by avoiding laborious sizing-layout iterations. Being fast, the tools allow to explore the design space in presence of layout parasitics. Knowledge capture in design plans and layout templates also lead to first-pass silicon realizations of technology migrated versions of the same circuit with guaranteed performance. The same knowledge can also be reused in similar designs leading to reduced design time due to the accumulation of stored design experience. The methodology is a compromise between two possible candidates for analog design reuse. The first is through specific block generators that store design experiences, but require a considerable effort and time spent in the associated preparatory work including both electrical and physical design formulation, besides a huge dedicated software design and maintenance work. Being inflexible, non-interactive, with only limited parameter boundaries, restricts the practical utility of the generated blocks. The second way is resizing based *only* on a working design where usually most of the original circuit and layout design considerations are lost while re-sizing due to the lack of such information.

The methodology has been applied in a design context. A low-voltage low-power $\Delta\Sigma$ modulator has been selected as a design application. Higher integration densities in future technologies and smaller channel lengths lead to lower supply voltages. This study has led to new circuit architectures and building blocks that allow very low-voltage, robust, switched-capacitor circuit operation in standard CMOS technologies. Very low-voltage switch operation is made possible through a special bootstrapped switch which allows rail-to-rail signal switching while limiting all gate-source and gate-drain voltages to V_{DD} thus preventing gate-oxide overstress. A modified opamp architecture was used. The proposed fully differential opamp allows very low supply voltage operation and minimizes the additional CMFB circuitry thus reducing overall power consumption. A 1-V, 1-mW $\Delta\Sigma$ modulator has been designed using the above methodology and CAD tools. Measurements show that for an OSR of 100 the modulator achieves a DR of 88 dB, a peak SNR of 87 dB and a peak SNDR of 85 dB in a signal bandwidth of 25 kHz. Obtained results show the feasibility of very low-voltage high performance circuits using standard switched-capacitor techniques.

Design reuse has been, then, investigated in two different ways: First, by re-designing (from specifications to layout) the same modulator in a different technology, and secondly, by re-designing a fourth-order one with more demanding specifications, and having a different topology, but using the same building blocks. The time needed to design these modulators is shown to be greatly reduced. Besides being of a particular interest from the design point of view, the circuits also demonstrate the utility of the proposed methodology.

This work shows that design experience acquired during analog design can be efficiently stored for eventual similar designs. This allows rapid IP design reuse. While analog design automation methodologies are not yet widely accepted by the analog designers community, design reuse requirements will soon be a huge driving force behind the inevitable future demand on such tools. Design plans and layout templates are believed to be the most efficient way for design reuse. This may be the first step towards more flexible, yet predictable, analog IP cores than existing hard ones. In addition, with the actual trend towards mobile and RF-designs where electrical and physical design can not be handled separately anymore, the presented layout-oriented design methodology and the accompanying tools are hoped to offer suitable solutions both to accelerate the design cycle and to facilitate future design reuse.

10.2 Future Work

This work has tackled different aspects of both analog design automation and circuit design. While it has not been easy to deal with both of these domains in the same time, it has been really an advantage that a CAD tool builder be at the same time its direct user. In this kind of research where different kind of problems are faced, several research paths also result. In particular, the following points seem interesting to investigate:

- The layout-oriented design methodology has been used in this work only on the cell level. In the $\Delta\Sigma$ modulator, each integrator is composed of three main parts, namely switches, the capacitor matrix and the amplifier. Each of these elements were sized separately using the corresponding layout template to account for the associated parasitics. The amplifiers were sized last to be able to calculate its load capacitance. As mentioned in section 4.5, the methodology is a top-down one to allow top-down shape constraint and parasitics propagation from one hierarchical level to the other. This property is not yet fully exploited and further investigations are thus encouraged.
- Concerning the procedural layout tool CAIRO:
  - While the relative procedural description of module placement is a direct task, routing description is still a time-consuming one. Relative routing must be coded in such a way that ensures its correctness with respect to design rules for all possible aspect ratios and device sizes of a given module. An automatic router could help to alleviate a lot of the burden. Critical nets could still be described procedurally while others could be left to an automatic routing phase.
  - Parasitic capacitance extraction needs some refinement. The approach used in the tool is a simple non-hierarchical one that relies on the procedural description of each module. Further enhancement may include considering coupling capacitors between complicated metal structures, intra-module parasitic capacitance extraction, and investigating shielding structures for critical nets.
  - Parasitic resistance extraction should also be incorporated. While the operation of analog circuits is not very sensitive to the absolute values of such resistances, resistive mismatch could play an important role in performance deterioration.
  - Integrated inductors are currently heavily used in RF front-ends. Inductance extraction accompanied with the associated parasitics in the layout-oriented methodology could then help the designers to accurately determine inductance parameters.

  - Other simple and compound devices can also be added to enrich the device generator library. This may include MOS cascode transistors, bipolar transistors, diodes, integrated inductors, . . .

- With respect to the sizing tool COMDIAC:

  - As mentioned in section 8.5.2.3, design space exploration was done manually. Analog design is characterized by its numerous design parameters. Only very experienced designers could reach nearly optimum solutions with respect to power consumption and layout area. An optimization CAD tool could be investigated in this context. Simplified sizing plans focus on the main characteristics and design heuristics in order to enhance design yield and reliability while the optimization tool works to fine-tune second-order effects and minimize/maximize certain design objectives.
  - As mentioned in section 3.2, behavioral models are important in system-level design and verification. The sizing tool must then be able to generate the corresponding behavioral model of each circuit after sizing. These models must also be able to account for parasitics effects so as to be used in the proposed methodology.
  - Adding new designs, efficiently, to COMDIAC requires further improvements from the software point of view. Experience show that the stored SPICE-like models and basic building blocks sizing procedures helps to capture sizing plans of new designs, basically based on existing designs in a follow-the-example way. However, a comprehensive procedure with some pre-defined functions need to be clearly defined to facilitate this process.

- On the low-voltage low-power circuit design level:

  - The proposed techniques were tested for an audio application. Investigation of higher speed applications would show the possibility of high-speed very low-voltage operation.
  - However, higher speed means also higher power consumption. In addition, as mentioned earlier, power consumption of analog circuits increases as the supply voltage decreases. New circuit techniques are thus needed to limit the drain of power. Class AB amplifiers could be investigated in this context.

Appendix A

CAIRO Layout Language Syntax

This appendix describes the layout language CAIRO. Together with chapter 5, they represent the user manual.

A.1 Module Definition

Any module is defined by describing both *placement* and *routing* of its child devices and submodules as follows:

```
#include <cairo.h>
main(argc, argv)
int argc;
char **argv;
{
/***** (1) Open a SPICE file  ******************************/
CAIRO_OPEN_SPICE_FILE("netlist.cir");
/***** (2) Sizing-Parasitics optimization loop *************/
CAIRO_OPTIMIZE(TRUE);
/***** (3) Placement  **************************************/
CAIRO_OPEN_MODULE("module_name");
    <Device Declaration>
    <Hierarchy Definition>
CAIRO_CLOSE_MODULE("module_name");
/***** (4) Area optimization *******************************/
CAIRO_RESHAPE("module_name", ...);
CAIRO_PLACE("module_name");
/***** (5) Routing  ****************************************/
CAIRO_BEGIN_ROUTE("module_name","module_name");
```

```
    <Routing functions>
CAIRO_END_ROUTE("module_name");
/***** (6) Verification  **********************************/
CAIRO_DRC("module_name");
}
```

As can be seen, module definition include six main parts:

1. Opening a netlist file that contains the device physical information in addition to some layout options for the implementation.
2. Mode of operation determination. CAIRO has two mode of operations, either parasitics calculation with no layout generation, i.e. CAIRO_OPTIMIZE(TRUE), or layout generaton, i.e. CAIRO_OPTIMIZE(FALSE).
3. Device placement, refer to sections A.2 and A.3.
4. Area optimization, refer to section A.4.
5. Routing, refer to section A.5.
6. Verification, refer to section A.7.

In the following sections a detailed description of the functions allowing device declaration, hierarchy definition, area optimization and routing is presented.

A.2 Available Devices

In this section, functions that allow the use of built-in module generators are described.

A.2.1 TRANSISTOR

The TRANSISTOR device adds a MOS transistor to the current module as follows:

SYNOPSIS:

```
void CAIRO_TRANSISTOR(name, type, W, L, bulk, <layout options>, C_END);
void CAIRO_TRANSISTOR_SPI(name, type, spice_name, bulk, <layout
options>, C_END);
char type, *name, *spice_name;
float W, L;
char bulk;
```

PARAMETERS:

type: Transistor type NTRANS/PTRANS.

name: Model name to be assigned to the device.
spice_name: Name of the transistor in the SPICE netlist.
W: Gate width of the transistor in μm.
L: Gate Length of the transistor in μm.
bulk: Bulk connection associated with the device. It can be:
B_S: the bulk is internally connected to the source, i.e. there is no *Bulk* connection.
B_O: the bulk is a separate connector, *Bulk*.
B_R: the bulk is a separate connector in the form of a ring surrounding the device.
B_N: the bulk is read from the netlist file.
<layout options>: See section A.2.1.1.

DESCRIPTION:
The first function declares a transistor given its W and L, while the second captures this information from a Spice file.

EXAMPLE:
`CAIRO_TRANSISTOR("MN1",NTRANS,50.0,2.0,B_O,"DUMMY",C_END);`
Declares an n-type transistor of name *MN1*, of gate width 50.0 μm and length 2.0 μm, and with a separate Bulk connector. Dummy transistors are added at both ends.

A.2.1.1 Transistor Layout Options

The layout style of the transistor is under control using the following style parameters:

I: Transistor drain current.
M: Number of transistor folds.
STACKS: Number of transistor stacks.
DUMMY: Places dummy transistors at both ends.
DIFF_CAP: Diffusion capacitance minimization. `MIN_D` minimizes drain capacitance, `MIN_S` minimizes source capacitance and `MIN_S` equalizes drain and source capacitances.
GATE_Y, DRAIN_Y, SOURCE_Y, BULK_Y: Controls the vertical order of the corresponding terminal.
GATE_X, DRAIN_X, SOURCE_X, BULK_X: Controls the horizontal order of the corresponding terminal for more than one stack.
GATE_Wx, DRAIN_Wx, SOURCE_Wx, BULK_Wx: Controls the width of different routing wires of the corresponding terminal (x=horizontal/vertical wire order).
GATE_TYPE_x, DRAIN_TYPE_x, SOURCE_TYPE_x, BULK_TYPE_x: Controls the layer of different routing wires of the corresponding terminal (x=horizontal/vertical wire order).

The default values of these parameters are found in a special *.sty* file, see section A.8.

A.2.2 DIFFPAIR

The DIFFPAIR device adds a differential pair MOS transistors, i.e. two transistors having the same gate W/L and a common *source* connector, to the current module as follows:

SYNOPSIS:

```
void CAIRO_DIFFPAIR(name, type, W, L, bulk, <layout options>, C_END);
void CAIRO_DIFFPAIR_SPI(name, type, spice_name1, spice_name2, bulk,
<layout options>, C_END);
char type, *name, *spice_name1, *spice_name2;
float W, L;
char bulk;
```

PARAMETERS:

type**:** Transistor type NTRANS/PTRANS.
name**:** Model name to be assigned to the device.
spice_name1**:** Name of the first transistor of the differential pair in the SPICE netlist.
spice_name2**:** Name of the second transistor of the differential pair in the SPICE netlist.
W**:** Gate width of the transistor in μm.
L**:** Gate Length of the transistor in μm.
bulk**:** Defined as in section A.2.1.
<layout options>**:** See section A.2.1.1.

DESCRIPTION:

The first function declares a differential pair of MOS transistors given its W and L, while the second captures this information from a Spice file.

EXAMPLE:

```
CAIRO_DIFFPAIR("MN1",NTRANS,50.0,2.0,B_S,"DIFF_CAP",MIN_D, C_END);
```

Declares an n-type differential pair transistor of name *MN1,* of gate width 50.0 μm and length 2.0 μm. Its bulk connected to its source connector. The drain diffusion capacitance is minimized.

A.2.3 BIASPAIR

The BIASPAIR module adds a pair of current mirror MOS transistors of identical length and width used for biasing, i.e. two transistors having a common gate and source connectors, to the current module as follows:

SYNOPSIS:

```
void CAIRO_BIASPAIR(name, type, W, L, bulk, <layout options>, C_END);
void CAIRO_BIASPAIR_SPI(name, type, spice_name1, spice_name2, bulk,
<layout options>, C_END);
```

```
char type, *name, *spice_name1, *spice_name2;
float W, L;
char bulk;
```

PARAMETERS:

type: Transistor type NTRANS/PTRANS.
name: Model name to be assigned to the device.
spice_name1: Name of the first transistor of the transistor pair in the SPICE netlist.
spice_name2: Name of the second transistor of the transistor pair in the SPICE netlist.
W: Gate width of the transistor in μm.
L: Gate Length of the transistor in μm.
bulk: Defined as in section A.2.1.
<***layout options***>: See section A.2.1.1.

DESCRIPTION:

The first function declares a current mirror pair of MOS transistors given its W and L, while the second captures this information from a Spice file.

EXAMPLE:

```
CAIRO_BIASPAIR("MN1",NTRANS,50.0,2.0,B_O,"GATE_TYPE_H",ALU2, C_END);
```

Declares an n-type pair with common gate and source transistors of name *MN1*, of gate width 50.0 μm and length 2.0 μm, and with a separate Bulk connector. The gate of all parallel transistors are joined using metal level 2.

A.2.4 CURRENT_MIRROR

The CURRENT_MIRROR device adds a multiple MOS transistors device, whose widths are an integer ratio, to the current module as follows:

SYNOPSIS:

```
void CAIRO_CURRENT_MIRROR(name, type, W, L, bulk, R1, R2, ... , W_END,
<layout options>, C_END);
void CAIRO_CURRENT_MIRROR_SPI(name,type, spice_name1, spice_name2, ... ,
W_END, bulk, <layout options>, C_END);
char type, *name, *spice_name1, *spice_name2;
float W, L;
int R1, R2;
char bulk;
```

PARAMETERS:

type: Transistor type NTRANS/PTRANS.
name: Model name to be assigned to the device.
spice_name1, spice_name2, ... : Name of the transistors in the SPICE netlist.

R1, R2, . . . **:** Ratio of transistor widths.
***W*:** Gate width of the transistor in μm.
***L*:** Gate Length of the transistor in μm.
***bulk*:** Defined as in section A.2.1.
<***layout options***>**:** See section A.2.1.1.

DESCRIPTION:
The first function declares any number of transistors (according to the number of ratio elements Rx) given their L and W of the first transistor (corresponding to R1), while the second captures this information from a Spice file.

EXAMPLE:
```
CAIRO_CURRENT_MIRROR("MN1",NTRANS,50.0,2.0,B_O,1,3,7,W_END,"DUMMY",C_END);
```

Declares a device composed of three transistors grouped in a stack of n-type of name *MN1*. The first transistor has a gate width and length of 50.0 μm 2.0 μm respectively, while the ratio of widths are $1 : 3 : 7$. It has a separate Bulk connector. Dummy transistors are added at both ends.

A.2.5 CAPACITOR

The CAPACITOR device adds a single capacitor to the current module as follows:

SYNOPSIS:
```
void CAIRO_CAPACITOR(name, type1, type2, value);
void CAIRO_CAPACITOR_SPI(name, type1, type2, spice_name);
char *name, *spice_name, type1, type2;
float value;
```

PARAMETERS:
***name*:** Model name to be assigned to the device.
***spice_name*:** Name of the capacitor in the SPICE netlist.
***type1*:** Type of the **lower** plate of the capacitor.
***type2*:** Type of the **upper** plate of the capacitor.
***value*:** Capacitor value in picofarads.

DESCRIPTION:
The first function declares a capacitor given its value, while the second captures this information from a Spice file.

EXAMPLE:
```
CAIRO_CAPACITOR("C1",POLY,POLY2,2.0);
```
Declares a POLY-POLY capacitor of name *C1* of 2pF.

A.2.6 CAPACITOR MATRIX

The MULTIPLE_CAPACITOR device adds a capacitor matrix composed of a maximum of five capacitors based on unit capacitances to the current module as follows:

SYNOPSIS:
```
void CAIRO_MULTIPLE_CAPACITOR(name, type1, type2, value, c1, c2, c3, c4,
c5, dummy);
void CAIRO_MULTIPLE_CAPACITOR_SPI(name, type1, type2, no, spice_name);
char *name, *spice_name, type1, type2;
float value, c1, c2, c3, c4, c5;
int no; BOOLEAN dummy;
```
PARAMETERS:
name: Model name to be assigned to the device.
spice_name: Name of the capacitor in the SPICE netlist.
type1: Type of the **lower** plate of the capacitor.
type2: Type of the **upper** plate of the capacitor.
value: Unit capacitor value.
no: Number of capacitors.
c1, c2, c3, c4, c5: Capacitor ratios.
dummy: Places dummy capacitors around the matrix.
DESCRIPTION:
The first function declares a capacitor matrix given its value and the capacitor ratios, while the second captures this information from a Spice file.
EXAMPLE:
```
CAIRO_MULTIPLE_CAPACITOR("C1",POLY,POLY2,1.0,4,6,2,1.5,5,YES);
```
Declares a POLY-POLY capacitor matrix of name *C1* composed of five capacitances. The unit capacitance is of 1pF and the capacitance ratio is 4:6:2:1.5:5. Dummy capacitances are placed around the matrix.

A.2.7 RESISTOR

The *RESISTOR* device adds a single poly resistance to the current module as follows:

SYNOPSIS:
```
void CAIRO_RESISTOR(name, value);
void CAIRO_RESISTOR_SPI(name, spice_name);
char *name, *spice_name;
float value;
```

PARAMETERS:

name: Model name to be assigned to the device.

spice_name: Name of the resistor in the SPICE netlist.

value: Resistor value in ohms.

DESCRIPTION:

The first function declares a resistor given its value, while the second captures this information from a Spice file.

EXAMPLE:

```
CAIRO_RESISTOR("R1",2.0e3);
```

Declares a resistor of name *R1* of 2 Kohms.

A.2.8 BLACK BOX

The *BLACK_BOX* device adds the layout (following the same symbolic approach) of a previously generated module to the current module. The added module does not follow any area optimization. It is treated as a *fixed* black box. The declaration is as follows:

SYNOPSIS:

```
void CAIRO_BLACK_BOX(name);
char *name;
```

PARAMETERS:

name: Name of the symbolic layout file which contains the module.

DESCRIPTION:

This function declares a module which already exists as a symbolic layout file to be used in the current module.

EXAMPLE:

```
CAIRO_BLACK_BOX("Pad1");
```

Declares a module whose layout already exists in the symbolic format in the "Pad1" file.

A.3 The Hierarchy

The hierarchy has been defined in section 5.5. A module is defined using the following functions: First adding the declared devices to horizontal *groups* using the CAIRO_ADD_DEVICE() function.

SYNOPSIS:

```
void CAIRO_ADD_DEVICE(device_name, group_name, ins_name, symmetry,<surround
paramters>,C_END);
char *device_name, *group_name, *ins_name, symmetry;
```

PARAMETERS:

device_name: Name of a previously declared device.

***group_name*:** Name of the group to contain the device.
***ins_name*:** Name of the instance of the device in the group.
***symmetry*:** Orientation of the device. It can be one of the following:
NOSYM: no geometrical operation is performed.
SYM_Y: Y becomes -Y, i.e. symmetry around the X-axis.
SYM_X: X becomes -X, i.e. symmetry around the Y-axis.
SYMXY: X becomes -X, Y becomes -Y.
ROT_P: a positive 90 degrees rotation takes place.
ROT_M: a negative 90 degrees rotation takes place.
SY_RP: Y becomes -Y, followed by a positive 90 degrees rotation.
SY_RM: Y becomes -Y, followed by a negative 90 degrees rotation.
***surround parameters*:** which can be (refer to section 5.3):
LE: Sets the left surround distance.
RI: Sets the right surround distance.
TO: Sets the top surround distance.
BO: Sets the bottom surround distance.

DESCRIPTION:
Instantiates a declared *device*, or a previously built *module*, of name *device_name* and adds it to the group *group_name*, with an orientation *symmetry*, and an instance name *ins_name*.
If it is the first time to use the group *group_name* a new group is created. If multiple devices are to be included in the group, they are placed physically according to the order of their CAIRO_ADD_DEVICE() statement from left to right.
Note that in case of using previously built modules as devices, no rotation of the instance is allowed, i.e. the allowed *symmetry* is only: NOSYM, SYM_Y, SYM_X, SYMXY. This is due to the reshaping algorithm used in CAIRO.

EXAMPLE:
```
CAIRO_ADD_DEVICE("MN5" ,"group1","I5",SYM_X,"LE",PITCH,C_END);
```
Instantiates the declared device "MN5" to "group1" with the instance name "I5" and a symmetry around its *X*-axis. The left surround is fixed to a value of *PITCH*.

Then by adding these groups to vertical *slices* using the CAIRO_ADD_GROUP() function.

SYNOPSIS:
```
void CAIRO_ADD_GROUP(group_name, slice_name, <surround paramters>,
C_END);
char *group_name, *slice_name;
```
PARAMETERS:
***group_name*:** Name of a previously filled group, see CAIRO_ADD_DEVICE().
***slice_name*:** Name of the slice.

surround parameters: Sets the surrounding space as in the CAIRO_ADD_DEVICE() function.
DESCRIPTION:
Adds the group *group_name* to the slice *slice_name*. If it is the first time to use the slice *slice_name* a new slice is created.
If multiple groups are to be included in the slice, they are placed physically according to the order of their CAIRO_ADD_GROUP() statement from bottom to top.
EXAMPLE:
```
CAIRO_ADD_GROUP("group1","slice2","TO",PITCH,C_END);
```
Adds "group1" to "slice2" with a top separation of one PITCH.

And finally, by adding slices to the current module using the CAIRO_ADD_SLICE() function.
SYNOPSIS:
```
void CAIRO_ADD_SLICE(slice_name, <surround paramters>, C_END);
char *slice_name;
```
PARAMETERS:
slice_name: Name of a previously filled slice, see CAIRO_ADD_GROUP().
surround parameters: Sets the surrounding space as in the CAIRO_ADD_DEVICE() function. Only "LE" and "RI" are allowed.
DESCRIPTION:
Adds the slice *slice_name* to the current module.
If multiple slices are to be included in the module, they are placed physically according to the order of their CAIRO_ADD_SLICE() statement from left to right.
EXAMPLE:
```
CAIRO_OPEN_MODULE("M1");
...
CAIRO_ADD_SLICE("slice3", "LE", PITCH, C_END);
```
Adds "slice3" to module "M1" with a separation of one PITCH from the previous slice.

A.4 Area Optimization

Area optimization is done using the CAIRO_RESHAPE function as follows:
SYNOPSIS:
```
void CAIRO_RESHAPE(main_module_name, opt_mode, shape_factor, force_opt);
char *main_module_name, *shape_factor;
OPTIM opt_mode;
BOOLEAN force_opt;
```
PARAMETERS:
main_module_name: Name of the main module to be optimized.

opt_mode: H: optimization given the layout height, A: optimization given the layout aspect ratio.

shape_factor: This is the layout height if the *opt_mode*=H, or the layout aspect ratio if the *opt_mode*=A.

force_opt: If this parameter is TRUE, optimization is always done. While if it is FALSE, optimization is no repeated if all devices already exist. This option is useful during routing where by setting it to FALSE, repeated reshaping is avoided.

DESCRIPTION:

Performs area optimization according to the option used (either specifying the layout height or the layout aspect ratio). This function is called only *once* in the code, it is used for the main module of the layout. It creates physically on the hard-disk all the built-in devices instantiated in the main module, after having computed their convenient shape.

EXAMPLE:

```
CAIRO_RESHAPE("M1",A,"2.0",TRUE);
```

Performs a forced area optimization for module *M1* using an aspect ratio of 2.0, i.e. the width of the layout is approximately equal to twice its height.

Followed by physical placement of all *shaped* devices using the CAIRO_PLACE function:

SYNOPSIS:

```
void CAIRO_PLACE(instance_name);
char *instance_name;
```

PARAMETERS:

instance_name: Name of the module instance to be placed.

DESCRIPTION:

Builds the given instance on the hard-disc by placing all its instances. This function is called for each module instance. Usually it follows the call of the CAIRO_RESHAPE() function. If it is the main module that is to be routed, the *ins_name* is the same as the *module_name*.

EXAMPLE:

```
CAIRO_PLACE("M1ins");
```

Places the previously reshaped module instance *M1ins*.

It should be noted that in a hierarchical description, modules can instansiate other modules. Area optimization and module physical placement should be done only once at the highest level of hierarchy, i.e. in the main module. Other sub-modules are only defined be assigning their relative placement, refer to section A.2 and A.3, and routing, refer to section A.5.

A.5 Routing

Routing is done explicitly by the user, i.e. the user has to describe , using the language, how each terminal is physically connected to other terminals. A description of the routing functions then follows:

SYNOPSIS:

```
void CAIRO_WIRE1(layer, width, ins1, con1, index1, ins2, con2, index2,
hv);
char layer;
long width;
char *ins1, *con1, *ins2, *con2;
long index1,index2;
ORIENT hv;
```

PARAMETERS:

layer:
Layout layer of the segment. ***width***: Width of the segment.
ins1: Name of the instance in which the connector *con1* is to be looked for.
ins2: Name of the instance in which the connector *con2* is to be looked for.
con1: Name of the connector, used as first endpoint of the segment.
index1: Index of con1.
con2: Name of a connector, used as last endpoint of the segment.
index2: Index of con2.
hv: Orientation of the segment, either horizontal (HOR) or vertical (VER).

DESCRIPTION:
Connects two terminals with a one segment horizontal/vertical wire. Note that if the two terminals are not on the same horizontal/vertical straight line, an error message occurs.

EXAMPLE:

```
CAIRO_WIRE1(ALU1,SW_ALU1,"MN1","CON1",0,"MN2","CON2",1,HOR);
```

Connects the two connectors *CON1.0* and *CON2.1,* belonging to the instances *MN1* and *MN2* respectively with a horizontal wire of *ALU1* of width *SW_ALU1*. See fig. A.1-a.

SYNOPSIS:

```
void CAIRO_WIRE2(layer1, layer2, width1, width2, ins1, con1, index1,
ins2, con2, index2, hv);
char layer1, layer2;
long width1, width2;
char *ins1, *con1, *ins2, *con2;
long index1,index2;
```

```
ORIENT hv;
```

PARAMETERS:

layer1, layer2: Layout layers of the two segments.

width1, width2: Width of the two segments.

ins1: Name of the instance in which the connector *con1* is to be looked for.

ins2: Name of the instance in which the connector *con2* is to be looked for.

con1: Name of the connector, used as first endpoint of the segment.

index1: Index of con1.

con2: Name of a connector, used as last endpoint of the segment.

index2: Index of con2.

hv: Orientation of the first segment, either horizontal (HOR) or vertical (VER).

DESCRIPTION:

Connects two terminals with a two segment L-shaped wire.

EXAMPLE:

```
CAIRO_WIRE2(ALU1,ALU2,SW_ALU1,SW_ALU2,"MN1","CON1",0,"MN2","CON2",1,HOR);
```

Connects the two connectors *CON1.0* and *CON2.1*, belonging to the instances *MN1* and *MN2* respectively with two wires; the first is in *ALU1*, of width *SW_ALU1*, while the second is in *ALU2* and of width *SW_ALU2*. The first wire is horizontal, while the second is vertical. See fig. A.1-b.

SYNOPSIS:

```
void CAIRO_WIRE3(layer1, layer2, layer3, width1, width2, width3, ins1,
con1, index1 , ins2, con2, index2, xy, hv);
char layer1, layer2, layer3;
long width1, width2, width3;
char *ins1, *con1, *ins2, *con2;
long index1,index2;
long xy;
ORIENT hv;
```

PARAMETERS:

layer1, layer2, layer3: Layout layers of the three segments.

width1, width2, width3: Width of the three segments.

ins1: Name of the instance in which the connector *con1* is to be looked for.

ins2: Name of the instance in which the connector *con2* is to be looked for.

con1: Name of the connector, used as first endpoint of the segment.

index1: Index of con1.

con2: Name of a connector, used as last endpoint of the segment.

index2: Index of con2.

xy: X-coordinate of the first segment's end point if it is horizontal, or its y-coordinate if it is

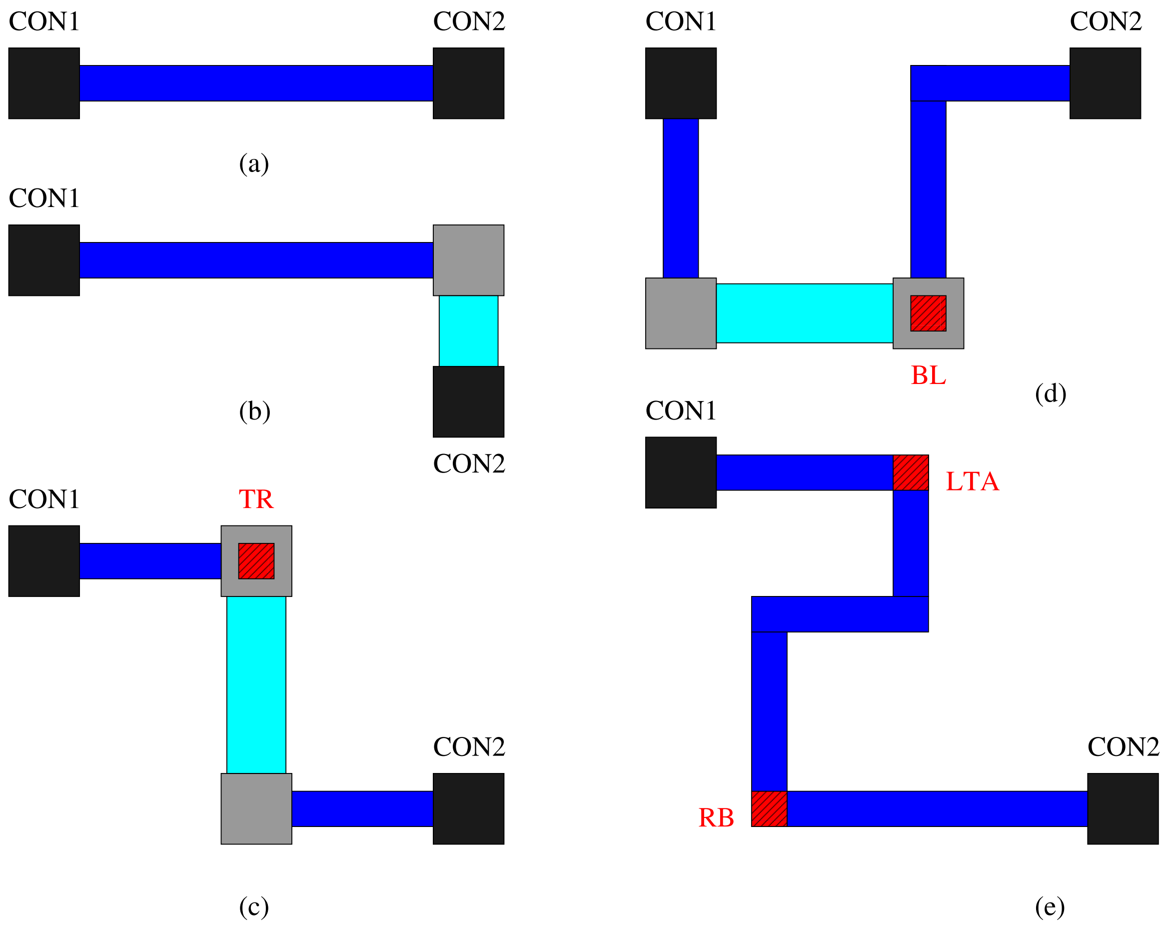


Figure A.1: *Routing functions.*

vertical.

hv: Orientation of the first segment, either horizontal (HOR) or vertical (VER).

DESCRIPTION:

Connects two terminals with a three segment wire.

EXAMPLE:

```
CAIRO_WIRE3(ALU1, ALU2, ALU1, SW_ALU1, SW_ALU2, SW_ALU1, "MN1", "CON1",
0, "MN2", "CON2", 1, CAIRO_GET_X("MN1",TR), HOR);
```

Connects the two connectors *CON1.0* and *CON2.1*, belonging to the instances *MN1* and *MN2* respectively with three wires; the first is in *ALU1*, and of width *SW_ALU1*, the second is in *ALU2* and of width *SW_ALU2*, and the third is in *ALU1*, and of width *SW_ALU1*. The first wire is horizontal. See fig. A.1-c. Note the use of the function *CAIRO_GET_X()* for capturing the x-coordinate of the reference *TR*. This assures relative routing, see section A.5.2.

SYNOPSIS:

```
void CAIRO_WIRE4(layer1, layer2, layer3, layer4, width1, width2,
width3, width4, ins1, con1, index1 , ins2, con2, index1, xm, ym, hv);
char layer1, layer2, layer3, layer4;
long width1, width2, width3, width4;
char *ins1, *con1, *ins2, *con2;
long index1,index2;
long xm, ym;
ORIENT hv;
```

PARAMETERS:

***layer1, layer2, layer3, layer4*:** Layout layers of the four segments.
***width1, width2, width3, width4*:** Width of the four segments.
***ins1*:** Name of the instance in which the connector *con1* is to be looked for.
***ins2*:** Name of the instance in which the connector *con2* is to be looked for.
***con1*:** Name of the connector, used as first endpoint of the segment.
***index1*:** Index of con1.
***con2*:** Name of a connector, used as last endpoint of the segment.
***index2*:** Index of con2.
***xm, ym*:** Coordinates of the second segment's end points in the current module.
***hv*:** Orientation of the first segment, either horizontal (HOR) or vertical (VER).

DESCRIPTION:

Connects two terminals with a four segment wire.

EXAMPLE:

```
CAIRO_WIRE4(ALU1, ALU2, ALU1, ALU1, SW_ALU1, SW_ALU2, SW_ALU1,
SW_ALU1, "MN1", "CON1", 0, "MN2", "CON2", 1, CAIRO_GET_X("MN1",BL),
CAIRO_GET_Y("MN1",BL), VER);
```

Connects the two connectors *CON1.0* and *CON2.1,* belonging to the instances *MN1* and *MN2* respectively with four wires; the first is in *ALU1,* and of width *SW_ALU1,* the second is in *ALU2* and of width *SW_ALU2,* the third and the fourth are in *ALU1,* and of width *SW_ALU1.* The first wire is vertical. See fig. A.1-d. Note the use of the x and y coordinate capturing functions *CAIRO_GET_X()* and *CAIRO_GET_Y()* respectively, This assures relative routing, see section A.5.2.

SYNOPSIS:

```
void CAIRO_WIRE5(layer1, layer2, layer3, layer4, layer5, width1,
width2, width3, width4, width5, ins1, con1, index1, ins2, con2, index1,
xm, ym, xy, hv);
char layer1, layer2, layer3, layer4, layer5;
long width1, width2, width3, width4, width5;
char *ins1, *con1, *ins2, *con2;
```

```
long index1,index2;
long xm, ym, xy;
ORIENT hv;
```

PARAMETERS:

layer1, layer2, layer3, layer4, layer5**:** Layout layers of the five segments.

width1, width2, width3, width4, width5**:** Width of the five segments.

ins1**:** Name of the instance in which the connector *con1* is to be looked for.

ins2**:** Name of the instance in which the connector *con2* is to be looked for.

con1**:** Name of the connector, used as first endpoint of the segment.

index1**:** Index of con1.

con2**:** Name of a connector, used as last endpoint of the segment.

index2**:** Index of con2.

xm, ym**:** Coordinates of the third segment's end points in the current module.

xy**:** X-coordinate of the first segment's end point if it is horizontal, or its y-coordinate if it is vertical.

hv**:** Orientation of the first segment, either horizontal (HOR) or vertical (VER).

DESCRIPTION:

Connects 2 connectors of instances with a 5 segment wire. All segments may belong to different layers.

EXAMPLE:

```
CAIRO_WIRE5(ALU1, ALU1, ALU1 , ALU1 ,ALU1, SW_ALU1, SW_ALU1,
SW_ALU1, SW_ALU1, SW_ALU1, "MN1", "CON1", 0, "MN2", "CON2", 1,
CAIRO_GET_X("MN2",RB),
CAIRO_GET_Y("MN2",RB), CAIRO_GET_X("MN1",LTA), HOR);
```

Connects the two connectors *CON1.0* and *CON2.1*, belonging to the instances *MN1* and *MN2* respectively with five wires; all are in *ALU1*, and of width *SW_ALU1*. The first wire is horizontal. See fig. A.1-e. Note the use of the x and y coordinate capturing functions *CAIRO_GET_X()* and *CAIRO_GET_Y()* respectively, This assures relative routing, see section A.5.2.

A.5.1 Definition of Module Connectors

After routing a module, its external *connectors* must be defined. Those are the terminals of the created module (sub-circuit), that are used for connecting the module with other blocks. In order to add the connector on the abutment box, we use one of the following functions:

SYNOPSIS:

```
void CAIRO_ADD_SUPPLY_H(layer, top, down);
void CAIRO_ADD_SUPPLY_V(layer, top, down);
char layer,*top,*down;
```

PARAMETERS:

layer: Physical layer for the supply connector, and the wire between the two connectors.

top, down: Name of the two supply connectors.

DESCRIPTION:

Adds a horizontal/vertical two supply segments that passes through the top/left and bottom/right of the current module. Two connectors are then added on both ends on the abutment box.

EXAMPLE:

```
CAIRO_ADD_SUPPLY_H(ALU1, "VSS", "VDD");
```

Adds two horizontal ALU1 segments at the top and bottom of the current module with connectors on both sides called "VSS" and "VDD" respectively.

SYNOPSIS:

```
void CAIRO_PLACE_CON_H(insname, conname, index, newname, layer, width);
void CAIRO_PLACE_CON_V(insname, conname, index, newname, layer, width);
char *insname, *conname;
int index;
char *newname,layer;
long width;
```

PARAMETERS:

insname: Name of the instance in which the connector is to be searched for.

conname: Name of the connector of the instance to be used for the module connector placement.

index: Index of *conname*.

conname: Name of the new connector to be placed.

layer: Physical layer for the new connector, and the wire between the two connectors.

width: Width of the new connector.

DESCRIPTION:

Adds a horizontal/vertical segment that passes through an internal connector of an instance in the current module. The segment is drawn through the module, from one side of the abutment box to the other side. Two connectors are then added on both ends on the abutment box.

EXAMPLE:

```
CAIRO_PLACE_CON_H("MN1","Gate",0,"Bias",POLY,SW_POLY);
```

Adds a horizontal POLY segment of width SW_POLY that passes through the "Gate" connector of the "MN1" instance.

SYNOPSIS:

```
void CAIRO_PLACE_CON1(insname, refname, index, conname, layer, width,
```

```
face);
char *insname, *refname, *conname;
int index;
char layer, face;
long width;
```

PARAMETERS:

insname: Name of the instance in which the connector is to be searched for.

refname: Name of the connector of the instance to be used for the module connector placement.

index: Index of *refname*.

conname: Name of the new connector to be placed.

layer: Physical layer for the new connector, and the wire between the two connectors.

width: Width of the new connector.

face: Face of the figure on which the new connector is to be placed. It can take any of the following values:

NORTH: for a connector placed on the top of the module.
SOUTH: for a connector placed on the bottom of the module.
EAST: for a connector placed on the right side of the module.
WEST: for a connector placed on the left side of the module.

DESCRIPTION:

Places a connector on a given side of the abutment box on the same straight line from a defined connector of an internal instance. A segment is then drawn between the two connectors.

EXAMPLE:

```
CAIRO_PLACE_CON1("OPAMP", "OUTPUT", 2, "OUT1", ALU1, SW_ALU1,EAST);
```

Places a connector of name *OUT1* on the *EAST* side of the current module to the right of the connector *OUTPUT.2,* at the same y-coordinate. Then it joins the two connectors with a one segment wire of *ALU1* of width *SW_ALU1*.

SYNOPSIS:

```
void CAIRO_PLACE_CON2(insname, refname, index, conname, xy, layer1,
layer2, width1, width2, face);
char *insname, *refname, *conname;
long index;
char layer1, layer2, face;
long width1, width2, xy;
```

PARAMETERS:

insname: Name of the instance in which the connector is to be searched for.

refname: Name of the connector of the instance to be used for the module connector placement.

index: Index of *refname*.

***conname*:** Name of the new connector to be placed.

***xy*:** Position of the new connector w.r.t. the connector of the instance. If *face* is EAST or WEST, *xy* gives the relative y-coordinate of the new connector w.r.t. the connector of the instance, while if *face* is NORTH or SOUTH it gives its relative x-coordinate. ***layer1, layer2*:** Physical layers of the two segments joining the two connectors. *layer2* also gives the layer of the new connector.

***width1, width2* :** Width of the two segments joining the two connectors. *width2* also gives the width of the new connector. ***face*:** Face of the figure on which the new connector is to be placed. It can take any of the following values:

NORTH: for a connector placed on the top of the module.
SOUTH: for a connector placed on the bottom of the module.
EAST: for a connector placed on the right side of the module.
WEST: for a connector placed on the left side of the module.

DESCRIPTION:
Places a connector on a given side of the abutment box with respect to a defined connector of an internal instance. Two segments are then drawn between the two connectors.

EXAMPLE:
```
CAIRO_PLACE_CON2("OPAMP", "OUTPUT", 1, "OUT1", -SAD_ALU1, ALU1, ALU2,
SW_ALU1, SW_ALU2,NORTH);
```
Places a connector of name *OUT1* on the *NORTH* side of the current module to the top of the connector *OUTPUT.1* (of type *CON*), at an y-coordinate which is equal to that of *OUTPUT* minus *SAD_ALU1*. Then it joins the two connectors with a two segment wire of *ALU1* and *ALU2* of width *SW_ALU1* and *SW_ALU1* respectively.

SYNOPSIS:
```
void CAIRO_PLACE_CON3(insname, refname, index, conname, x, y, layer1,
layer2, layer3, width1, width2, width3, face, hv);
char *insname, *refname, *conname;
long index;
char layer1, layer2, layer3, face;
long width1, width2, width3, xy;
ORIENT hv;
```
PARAMETERS:

***insname*:** Name of the instance in which the connector is to be searched for.

***refname*:** Name of the connector of the instance to be used for the module connector placement.

***index*:** Index of *refname*.

***conname*:** Name of the new connector to be placed.

***x, y*:** Co-ordinates of a point inside the current module on the same straight line as the new connector.

***layer1, layer2, layer3*:** Physical layers of the two segments joining the two connectors. *layer3* also gives the layer of the new connector.

width1, width2, width3 **:** Widths of the three segments joining the two connectors. *width3* also gives the width of the new connector.

***face*:** Face of the figure on which the new connector is to be placed. It can take any of the following values:

NORTH: for a connector placed on the top of the module.
SOUTH: for a connector placed on the bottom of the module.
EAST: for a connector placed on the right side of the module.
WEST: for a connector placed on the left side of the module.

***hv*:** Orientation of the first segment, either horizontal (HOR) or vertical (VER).

DESCRIPTION:

Places a connector on a given side of the abutment box with respect to a defined connector of an internal instance. Three segments are then drawn between the two connectors.

EXAMPLE:

```
CAIRO_PLACE_CON3("OPAMP", "OUTPUT", 1, "OUT1", CAIRO_GET_X("MN1","Gate",0),
CAIRO_GET_Y("MN1","Gate",0), ALU1, ALU2, ALU1, SW_ALU1, SW_ALU2, SW_ALU1,
NORTH, HOR);
```

Places a connector of name *OUT1* on the *NORTH* side of the current module on the same straight line as the *Gate* connector of the *MN1* instance. Then the two connectors are joined with a one segment wire. The *Gate* connector is then joined to the *OUTPUT* connector of the *OPAMP* instance with a two segment wire similar to CAIRO_WIRE2().

A.5.2 Capturing Relative Coordinates

Pre-defined reference points (see section 5.3) and connectors are used to describe relative routing. Their coordinates are captured by the following functions:

SYNOPSIS:

```
long CAIRO_GET_X(insname, refname, index);
long CAIRO_GET_Y(insname, refname, index);
char *insname, *refname;
long index;
```

PARAMETERS:

***insname*:** Name of the instance in which the reference/connector exists.

***refname*:** Name of the reference/connector.

***index*:** Index of connector, or =REF if *refname* is a reference point.

DESCRIPTION:

Returns the x/y position of the reference *refname* contained in the instance *insname* with respect to the current module.

EXAMPLE:

```
CAIRO_GET_X("MN1",RB,REF);
```
Returns the x-coordinate of the reference *RB* in the instance *MN1*.

A.6 Technology Variables

They are used to screen the layout rules. They correspond to the symbolic layout layers used by CAIRO in order to render the produced layout technology independent.

PITCH The minimum distance between two contacts of type ALU1-ALU2. This is an important variable during routing. It is usually used as a separation between different routing wires.

SW_LAYER The minimum symbolic width of the corresponding layer. Defined variables are: SW_NWELL, SW_NTIE, SW_PTIE, SW_NDIF, SW_PDIF, SW_GATE, SW_POLY, SW_POLY2, SW_ALUx, SW_CONT, SW_VIAx.

SW_LAYER1_LAYER2 The minimum symbolic width of layer1 over layer2. Defined variables are: SW_POLY_CONT, SW_POLY2_CONT, SW_ALU1_CONT, SW_ALUx_VIAx.

SD_LAYER The minimum symbolic distance between two edges of the same layer. Defined variables are: SD_NWELL, SD_NTIE, SD_PTIE, SD_NDIF, SD_PDIF, SD_GATE, SD_POLY, SD_POLY2, SD_ALUx, SD_CONT, SD_VIAx.

SD_LAYER1_LAYER2 The minimum symbolic distance between the edges of layer1 and layer2. Defined variables are: SD_NWELL_PTIE, SD_NWELL_NDIF, SD_NTIE_PTIE, SD_NTIE_NDIF, SD_NTIE_PDIF, SD_NTIE_GATE, SD_NTIE_POLY, SD_PTIE_NDIF, SD_PTIE_PDIF, SD_PTIE_POLY, SD_PTIE_GATE, SD_NDIF_PDIF, SD_NDIF_GATE, SD_NDIF_POLY, SD_PDIF_GATE, SD_PDIF_POLY, SD_GATE_POLY, SD_GATE_VIA, SD_POLY_VIA, SD_POLY2_VIA, SD_CONT_VIA.

SAD_LAYER The minimum symbolic distance between two axes of the same layer. Defined variables are: SAD_NWELL, SAD_NTIE, SAD_PTIE, SAD_NDIF, SAD_PDIF, SAD_GATE, SAD_POLY, SAD_POLY2, SAD_ALUx, SAD_CONT, SAD_VIA.

CURRENT_W Used for routing wire widths. The width is set according to the current passing in the connector to which the wire is connected. If no current information is available, SW_LAYER is used instead.

A.7 Design Rule Check and Layout Statistics

To be sure of the correctness of the layout each time the layout is generated, it is recommended to include the verification of the layout design rules in the program. This is done using the following function:

SYNOPSIS:

```
void CAIRO_DRC(instance_name);
char *instance_name;
```

PARAMETERS:

***instance_name*:** Name of the instance of the module to be verified.

DESCRIPTION:

Runs the symbolic Design Rule Checker *DRuC* on the symbolic layout *instance_name.ap*.

EXAMPLE:

```
CAIRO_DRC("M2");
```

Performs a DRC on the instance *M2*.

Information concerning the generated layout can also be obtained using the following function:

SYNOPSIS:

```
void CAIRO_STATISTICS(instance_name);
char *instance_name;
```

PARAMETERS:

***instance_name*:** Name of the instance of the module to be operated.

DESCRIPTION:

Prints on the screen statistical information concerning the layout of the circuit/sub-circuit *name*, for example its technology, width, height, area, ... A file *instance_name.inf* containing the same information is also created.

EXAMPLE:

```
CAIRO_STATISTICS("M2");
```

Prints the corresponding layout statistics of the instance *M2*.

A.8 Related Files

module_name.c Input CAIRO C file.

module_name.ap Output symbolic layout file.

module_name.inf Output file containing layout information about the generated module.

module_name.cap Output file containing all signal capacitances of the user-constructed module *module_name*.

module_name.cur Output file containing all connector currents of the user-constructed module *module_name*.

module_name.sty Input file containing all device styles in the module *module_name*. File of interaction between COMDIAC and CAIRO.

transistor.sty Input file containing default *transistor* style parameters. If not found in the current directory, the one in $(CAIRO)/etc is used.

diffpair.sty Input file containing default *differential pair* style parameters. If not found in the current directory, the one in $(CAIRO)/etc is used.

biaspair.sty Input file containing default *bias pair* style parameters. If not found in the current directory, the one in $(CAIRO)/etc is used.

Appendix B

Voltage Transients in R-C Networks

B.1 R-C-C Network

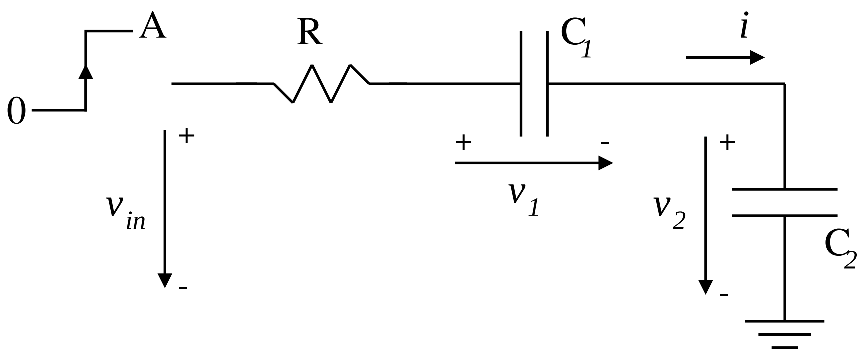


Figure B.1: *R-C-C network.*

This section analyzes the transient voltages and currents on an R-C-C network subjected to a voltage step input and having initial capacitance charges. The voltages on each capacitor are calculated separately. Consider the network shown in Fig. B.1 with a step input given by

$$v_{in}(t) = Au(t) \tag{B.1}$$

The current i can be calculated by

$$i = C_1\frac{dv_1}{dt} = C_2\frac{dv_2}{dt} = \frac{v_{in} - v_1 - v_2}{R} \tag{B.2}$$

Taking Laplace transform and considering the period $t > 0$, we get

$$RC_1[sV_1(s) - v_1(0)] = V_{in}(s) - V_1(s) - V_2(s) \tag{B.3}$$

$$C_1[sV_1(s) - v_1(0)] = C_2(sV_2(s) - v_2(0)) \tag{B.4}$$

Solving for V_1 we get

$$V_1(s) = \frac{1}{s + 1/\tau}\left[\frac{1}{s}\left(\frac{A}{\tau_1} + \frac{v_1(0)}{\tau_2} - \frac{v_2(0)}{\tau_1}\right) + v_1(0)\right] \tag{B.5}$$

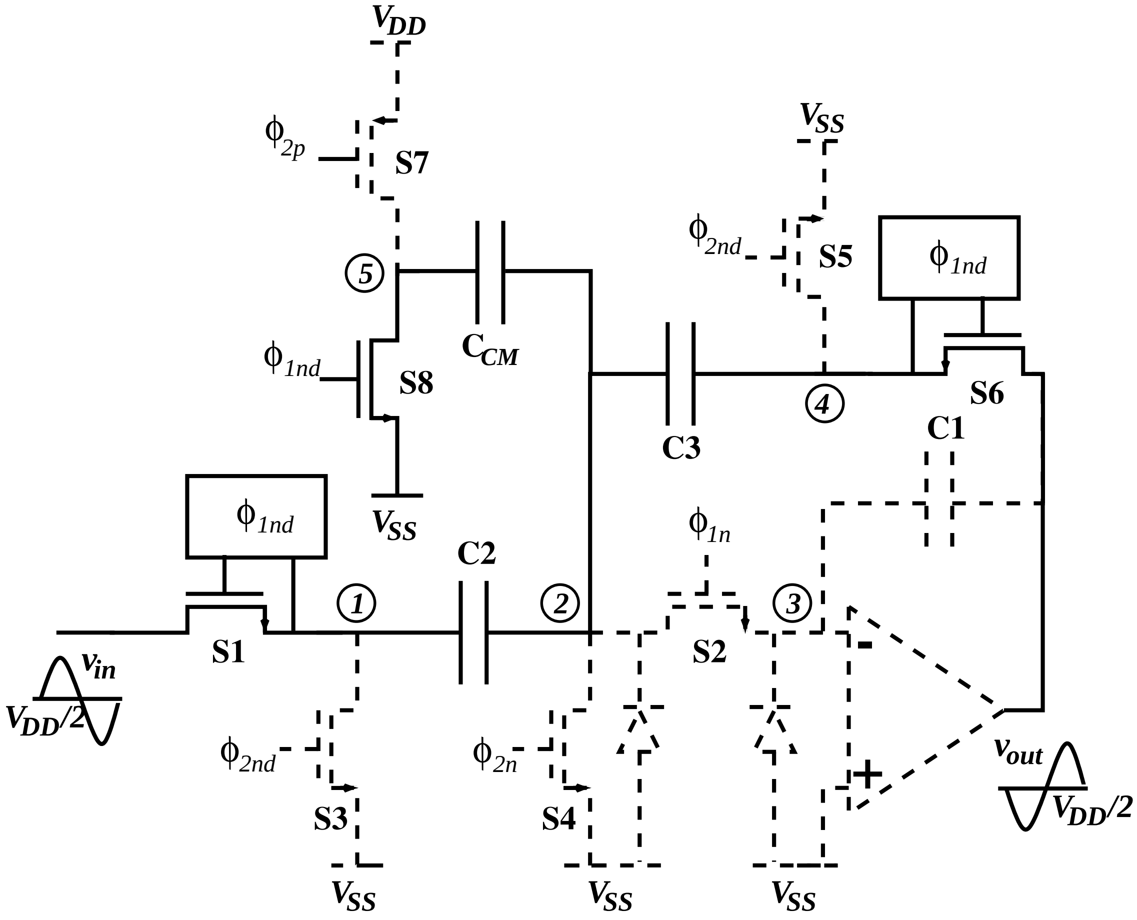


Figure B.2: *SC section using the charge cancellation scheme.*

where $\tau = RC_1C_2/(C_1 + C_2)$, $\tau_1 = RC_1$ and $\tau_2 = RC_2$. Then by taking the inverse Laplace transform, we get

$$v_1(t) = \frac{C_2A + C_1v_1(0) - C_2v_2(0)}{C_1 + C_2}\left(1 - e^{-t/\tau}\right) + v_1(0)e^{-t/\tau} \tag{B.6}$$

and similarly

$$v_2(t) = \frac{C_1A + C_2v_2(0) - C_1v_1(0)}{C_1 + C_2}\left(1 - e^{-t/\tau}\right) + v_2(0)e^{-t/\tau} \tag{B.7}$$

This means that the final voltage on each capacitor also depend on the distribution of charges at $t \leq 0$.

B.2 Analysis of Voltage Transients in the Charge Cancellation Scheme

As mentioned in section 7.3.1, one of the disadvantages of using V_{SS} as an analog reference is the bulk leakage due to transient negative spikes at certain nodes. However, often voltage-mode

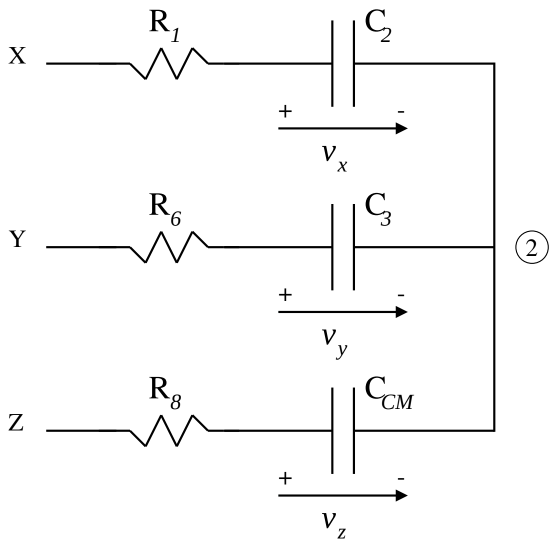


Figure B.3: *SC branches connected to node 2 in Fig. B.2 during ϕ_1.*

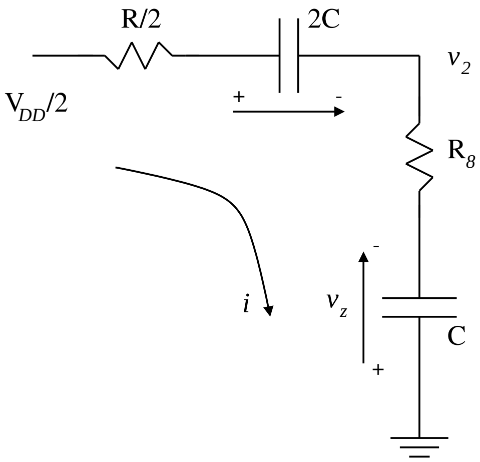


Figure B.4: *Equivalent circuit at node 2 in Fig. B.3, with $X = v_{in} = V_{DD}/2 + v$, $Y = v_{out} = V_{DD}/2 - (C_2/C_3)v$, $Z = V_{SS}$, $v_x(0) = 0$, $v_y(0) = 0$, $v_z(0) = V_{DD}$, $C_2 = C_3 = C_{CM} = C$, and $R_1 = R_6 = R$.*

cancellation is possible at these nodes to prevent the occurrence of such negative spikes. Consider for instance Fig. B.2 which shows the same SC section as in Fig. 7.3 with emphasis on the switched branches connected to node 2 during ϕ_1. Let us assume that the opamp has no effect on this node, the resulting circuit can be represented as shown in Fig. B.3, where 3 different voltage steps X, Y and Z are applied to capacitors C_2, C_3 and C_{CM} respectively through resistances R_1, R_6 and R_8 which represent the series resistances of switches S1, S6 and S8 respectively. The problem is, then, reduced to adjust the resistance values to have a null or a positive value transient spike at node 2. Comparing both figures, the capacitor initial voltages (at the end of ϕ_2) $v_x(0) = 0$, $v_y(0) = 0$ and $v_z(0) = V_{DD}$ respectively, while the input voltages are given by $X = v_{in} = V_{DD}/2 + v$,

$Y = v_{out} = V_{DD}/2 - (C_2/C_3)v$ and $Z = V_{SS}$. For a unity gain low-pass filter and from the charge cancellation condition (7.3), we have $C_2 = C_3 = C_{CM} = C$. The resistances are taken $R_1 = R_6 = R$ to have equal time constants. The input small signal components $+/-v$ cancel such that the X and Y branches become identical and are added in parallel so that the equivalent circuit reduces to that shown in Fig. B.4. The voltage V_2 at $t = 0$ is given by

$$V_2(0) = i(0)R_8 - V_{DD} = \frac{R_8 - R}{2R_8 + R} V_{DD} \tag{B.8}$$

As a result of the charge cancellation condition (7.3), the voltage at node 2 finally settles to V_{SS}. Thus, according to the results of section B.1, V_2 changes exponentially such that

$$v_2(t) = \frac{R_8 - R}{2R_8 + R} V_{DD} e^{-3t/(2R_8+R)C} \tag{B.9}$$

It can be kept positive by choosing $R_8 \geq R$.

List of Publications

The following publications can be downloaded from: http://www-asim.lip6.fr/publications/.

Methodology and CAD Tools

- Mohamed Dessouky, Andreas Kaiser, Marie-Minerve Louërat and Alain Greiner, "Analog Design for Reuse - Case Study: Very Low-voltage $\Delta\Sigma$ Modulator", *Design Automation and Test in Europe Conference (DATE'01)*, Munich, Germany, March 2001.

- Mohamed Dessouky, Marie-Minerve Louërat and Jacky Porte, "Layout-Oriented Synthesis of High Performance Analog Circuits", *Design Automation and Test in Europe Conference (DATE'00)*, pp. 53-57, Paris, France, March 2000.

- Mohamed Dessouky and Marie-Minerve Louërat, "A Layout Approach for Electrical and Physical Design Integration of High-Performance Analog Circuits", *1st International Symposium on Quality Electronic Design (ISQED'00)*, pp. 291-298, San Jose, USA, March 2000.

- Mohamed Dessouky, Alain Greiner and Marie-Minerve Rosset-Louërat, "CAIRO : A Hierarchical Layout Language for Analog Circuits", *Mixed Design of Integrated Circuits and Systems (MIXDES'99)*, pp. 105-110, Krakow, Poland, June 1999.

- Mohamed Dessouky, Jacky Porte and Marie-Minerve Rosset-Louërat, "TANIS : Un Outil pour la Synthèse de Circuits CMOS Analogiques", *Colloque CAO de Circuits Intégrés et Systèmes*, GDR 732, pp 186-189, France, Aix-en-Provence, May 1999.

- Mohamed Dessouky, Jacky Porte and Marie-Minerve Rosset-Louërat, "Synthèse de Circuits Faible Tension CMOS Analogiques", *2ème Journées Francophones d'études Faible Tension Faible Consommation (FTFC'99)*, pp. 126-130, Paris, France, May 1999.

- Mohamed Dessouky, Jacky Porte, Alain Greiner and Marie-Minerve Rosset-Louërat, "Synthèse de Circuits Analogiques CMOS", *1ère Journée Nationale Réseau Doctoral Microélectronique*, Toulouse, France, April 1998.

- Mohamed Dessouky, Alain Greiner and Marie-Minerve Rosset-Louërat, "CAIRO : Un Langage pour le Layout Analogique Symbolique", *Actes du 1er Colloque CAO de Circuits Intégrés et Systèmes*, pp. 14-17, Grenoble France, January 1997.

Low-voltage Circuit Design

- Mohamed Dessouky and Andreas Kaiser "Very Low-voltage Digital-Audio $\Delta\Sigma$ Modulator with 88-dB Dynamic Range using Local Switch Bootstrapping", to be published in *IEEE J. of Solid-State Circuits*, March 2001.

- Mohamed Dessouky and Andreas Kaiser "A 1V 1mW Digital-Audio Delta-Sigma Modulator with 88dB Dynamic Range using Local Switch Bootstrapping", *Custom Integrated Circuits Conference (CICC'00)*, pp. 13-16, Orlando FL, US, May 2000.

- Mohamed Dessouky and Andreas Kaiser, "Very Low-Voltage Fully-Differential Amplifier for Switched-Capacitor Applications", *IEEE International Symposium on Circuits and Systems (ISCAS'00)*, Vol. V, pp. 441-444, Geneva, Switzerland, May 2000.

- Mohamed Dessouky and Andreas Kaiser, "Rail-to-Rail Operation of Very Low Voltage CMOS Switched-Capacitor Circuits", *International Symposium on Circuits and Systems (ISCAS'99)*, Vol. II, pp. 144-147, Orlando FL, USA, May 1999.

- Mohamed Dessouky and Andreas Kaiser, "Circuits a Capacités Commutées Fonctionnant en mode "Rail-to-Rail" à Très Basse Tension", *2ème Journées Francophones d'études Faible Tension Faible Consommation (FTFC'99)*, pp. 25-28, Paris, France, May 1999.

- Mohamed Dessouky and Andreas Kaiser, "Circuits a Capacités Commutées Fonctionnant en mode "Rail-to-Rail" à Trés Basse Tension", *Journées Nationales du Réseau Doctoral de Micro-électronique*, Bordeaux, France, May 1999.

- Mohamed Dessouky and Andreas Kaiser, "Input Switch Configuration Suitable for Rail-to-Rail Operation of Switched-Opamp Circuits", *IEE Electronics Letters*, Vol. 35, No 1, pp. 8-10, January 1999.

Additional Publications

- Abdelhakim Khouas, Mohamed Dessouky and Anne Derieux, "Optimized Statistical Analog Fault Simulation", *IEEE Asian Test Symposium (ATS'99)*, Shanghai, China, November 1999.

- Hassan Aboushady, Mohamed Dessouky, Elizabeth de Lira Mendes and Patrick Loumeau, "A Third-Order Current-Mode Continuous-Time Sigma-Delta Modulator", *IEEE International Conference on Electronic Circuits and Systems (ICECS'99)*, Paphos, Cyprus, September 1999.

- Hassan Aboushady, Elizabeth de Lira Mendes, Mohamed Dessouky and Patrick Loumeau, "A Current-Mode Continuous-Time Sigma-Delta Modulator with Delayed Return-to-Zero Feedback", *International Symposium on Circuits and Systems (ISCAS'99)*, Orlando FL, USA, May 1999.

Bibliography

[Abo99a] Andrew M. Abo and Paul R. Gray. A 1.5-V, 10-bit, 14MS/s CMOS pipeline analog-to-digital converter. *IEEE J. of Solid-State Circuits*, 34(5):599–606, May 1999.

[Abo99b] Andrew Masami Abo. *Design for Reliability of Low-voltage, Switched-capacitor Circuits*. PhD thesis, University of California, Berkeley, http://kabuki.eecs.berkeley.edu/∼abo, 1999.

[Adams96] Robert W. Adams and Richard Schreier. Stability theory for $\Delta\Sigma$ modulators. In Steven R. Norsworthy, Richard Schreier, and Gabor C. Temes, editors, *Delta-Sigma Data Converters. Theory, Design, and Simulation*, chapter 4. IEEE Press, 1996.

[Arora96] N. D. Arora, K. V. Raol, and M. Richardson. Modelling and extracion of interconnect capacitances for multilayer VLSI circuits. *IEEE Trans. Computer-Aided Design*, 15(1), January 1996.

[Association99] Semiconductor Industry Association. Internaltional technology roadmap for semiconductors. http://www.itrs.net/1999_SIA_Roadmap/Home.htm, 1999.

[Baschirotto97a] Andrea Baschirotto and Rinaldo Castello. A 1V 1.8-MHz CMOS switched-opamp SC filter with rail-to-rail output swing. *IEEE J. of Solid-State Circuits*, 32(12):1979–1986, December 1997.

[Baschirotto97b] Andrea Baschirotto and Rinaldo Castello. A 1V 1.8MHz CMOS switched-opamp SC filter with rail-to-rail output swing. In *Proc. IEEE Int. Solid State Circuit Conf.*, pages 58–59, 1997.

[Bastos96] J. Bastos, M. Steyart, B. Graindourze, and W. Sansen. Matching of MOS Transistors with Different Layout Styles. In *Proc. IEEE Int. Conf. on Mecroelectronic Test Structures*, pages 17–18, March 1996.

[Bexten93] Volker Meyer zu Bexten, Claudio Moraga, Roland Klinke, Werner Brockherde, and Klaus-Gunther. ALSYN: Flexible rule-based layout synthesis for analog IC's. *IEEE J. of Solid-State Circuits*, 28(3):261–267, March 1993.

[Boser88] Bernhard E. Boser and Bruce A. Wooley. The design of sigma-delta modulation analog-to-digital converters. *IEEE J. of Solid-State Circuits*, 23:1298–1308, December 1988.

[Brandt96] Brian Brandt, Paul F. Ferguson, and Mike Rebeschini. Analog circuit design for $\Delta\Sigma$ ADCs. In Steven R. Norsworthy, Richard Schreier, and Gabor C. Temes, editors, *Delta-Sigma Data Converters. Theory, Design, and Simulation*, chapter 11. IEEE Press, 1996.

[Bruce96] J. D. Bruce, H. W. Li, M. J. Dallabetta, and R. J. Baker. Analog layout using ALAS! *IEEE J. of Solid-State Circuits*, 31(2):271–274, February 1996.

[Castello85] Rinaldo Castello and Paul R. Gray. A High-Performance Micropower Switched-Capacitor Filter. *IEEE J. of Solid-State Circuits*, 20(6):1122–1132, December 1985.

[Chang97] Henry Chang, Edoardo Charbon, Umakanta Choudhury, Alper Demir, Eric Felt, Edward Liu, Enrico Malavasi, Alberto Sangiovanni-Vincentelli, and Iassan Vassiliou. *A Top-Down Constraint-Driven Design Methodology for Analog Integrated Circuits*. Kluwer Academic Publishers, first edition, 1997.

[Charbon92] Edoardo Charbon, Enrico Malavasi, Umakanta Choudhury, Andrea Casotto, and Alberto Sangiovanni-Vincentelli. A constraint-driven placement methodology for analog integrated circuits. In *Proc. IEEE Custom Integrated Circuits Conf.*, pages 2821–2824, May 1992.

[Charbon93] Edoardo Charbon, Enrico Malavasi, and Alberto Sangiovanni-Vincentelli. Generalized constraint generation for analog circuit design. In *Proc. IEEE Int. Conf. on Computer-Aided-Design*, pages 408–414, November 1993.

[Chesneau98] Alphonse Chesneau and Matthieu Lintz. *Réalisation d'un Générateur Automatique de Filtres à capacités commutées, stage de DEA*. Univetsié Pierre et Marie Curie, Lab. LIP6, dept. ASIM, Paris, 1998.

[Choudhury90a] Umakanta Choudhury and Alberto Sangiovanni-Vincentelli. Constraint-based channel routing for analog and mixed analog/digital circuits. In *Proc. IEEE Int. Conf. on Computer-Aided-Design*, pages 198–201, November 1990.

[Choudhury90b] Umakanta Choudhury and Alberto Sangiovanni-Vincentelli. Constraint generation for routing analog circuits. In *Proc. Design Automation Conf.*, pages 561–566, June 1990.

[Choudhury91] Umakanta Choudhury and Alberto Sangiovanni-Vincentelli. An analytical-model generator for interconnect capacitances. In *Proc. IEEE Custom Integrated Circuits Conf.*, pages 861–864, May 1991.

[Coban99] Abdulkerim L. Coban and Phillip E. Allen. A 1.5V 1.0mW Audio $\Delta\Sigma$ Modulator with 98dB Dynamic Range. In *Proc. IEEE Int. Solid State Circuit Conf.*, pages 50–51, 1999.

[Cohn91] J. M. Cohn, R. A. Rutenbar, and L. R. Carley. KOAN/ANAGRAM II: New Tools for Device-Level Analog Placement and Routing. *IEEE J. of Solid-State Circuits*, 26(3):330–342, March 1991.

[Conway92] J. D. Conway and G. G. Schrooten. An Automatic Layout Generator for Analog Circuits. In *Proc. European Design Automation Conf.*, pages 513–519, 1992.

[Crols94] Jan Crols and Michel S. J. Steyaert. Switched-opamp: An approach to realize full CMOS switched-capacitor circuits at very low power supply voltages. *IEEE J. of Solid-State Circuits*, 29(8):936–942, August 1994.

[Donnay97] S. Donnay, G. Gielen, W. Sansen, W. Kruiskamp, D. Leenaerts, and W. van Bokhoven. High-level synthesis of analog sensor interface front-end. In *Proc. European Design and Test Conf.*, pages 56–60, March 1997.

[El-Turky89] Fatehy El-Turky and Elizabeth E. Perry. BLADES: An artificial intelligence approach to analog circuit design. *IEEE Trans. Computer-Aided Design*, 8(6):680–692, June 1989.

[Francken99] Kenneth Francken and Georges Gielen. Methodology for Analog Technology Porting Including Performance Tuning. In *Proc. IEEE Int. Symposium on Circuits and Systems*, volume 1, pages 415–418, May 1999.

[Geerts99] Yves Geerts, Augusto Manuel Marques, Michel S. J. Steyaert, and Willy Sansen. A 3.3-V, 15-bit, Delta-Sigma ADC with a Signal Bandwidth of 1.1 MHz for ADSL Applications. *IEEE J. of Solid-State Circuits*, 34(7):927–936, July 1999.

[Gielen91] Georges Gielen and Willy Sansen. *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Kluwer Academic Publishers, first edition, 1991.

[Gielen93] George Gielen, Koen Swings, and Willy Sansen. Open analog synthesis system based on declarative methods. In Johan H. Huijsing, Rudy J. van der Plassche, and Willy Sansen, editors, *Analog Circuit Design: Operational Amplifiers, Analog to Digital Converters and Analog Computer Aided Design*, pages 421–445. Kluwer Academic Publishers, 1993.

[Gray93] Paul R. Gray and Robert Gray Meyer. *Analysis and Design of Analog Integrated Circuits*. John Wiley and Sons, Inc., third edition, 1993.

[Gregorian86] Roubik Gregorian and Gabor C. Temes. *Analog MOS Integrated Circuits for Signal Processing*. John Wiley and Sons, Inc., first edition, 1986.

[Greiner90] Alain Greiner and Jean-Pierre Leroy. A symbolic layout view in edif for process independant design. In *Fourth European Edif Forum*, Daresbury, Cheshire, UK, 1990.

[Greiner92] Alain Greiner and François Pêcheux. Alliance: A Complete Set of CAD Tools for Teaching Digital VLSI design. In *Proc. of the 3rd Eurochip Workshop on VLSI Design Training*, pages 230–237, Toledo, Spain, September 1992.

[Greiner94] Alain Greiner and Frédéric Pétrot. Using C to write portable CMOS VLSI module generators. In *Proc. European Design Automation Conf.*, pages 676–681, September 1994.

[Greiner95] Alain Greiner, Frédéric Pétrot, and Franck Wajsburt. Fixed grid symbolic layout translation into mask rectangles. In *2nd Advanced Training Course: Mixed Design of VLSI Circuits - Education of Computer Aided Design of Modern VLSI Circuits*, pages 281–286, Poland, May 1995.

[Harjani89] Ramesh Harjani, Rob A. Rutenbar, and L. Richard Carley. OASYS: A framework for analog circuit synthesis. *IEEE Trans. Computer-Aided Design*, 8(12):1247–1266, December 1989.

[Hogervorst96] Ron Hogervorst and Johan H. Huijsing. *Design of Low-Voltage, Low-Power Operational Amplifier Cells*. Kluwer Academic Publishers, first edition, 1996.

[Hsieh81] K.C. Hsieh, P. R. Gray, D. Senderowicz, and D. G. Messerschmidt. A low-noise chopper-stabilized differential switched-capacitor filtering technique. *IEEE J. of Solid-State Circuits*, 16:708–715, December 1981.

[Johns97] David A. Johns and Ken Martin. *Analog Integrated Circuit Design*. John Wiley and Sons, Inc., first edition, 1997.

[Jusuf90] Gani Jusuf, Paul R. Gray, and Alberto Sangiovanni-Vincentelli. CADICS - Cyclic Analog-To-Digital Converter Synthesis. In *Proc. IEEE Int. Conf. on Computer-Aided-Design*, pages 286–289, November 1990.

[Koh90] Han Young Koh, Carlo H. Sequin, and Paul R. Gray. OPASYN: A Compiler for CMOS Operational Amplifiers. *IEEE Trans. Computer-Aided Design*, 9(2):113–125, February 1990.

[Laarhoven87] P. J. M. Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. Kluwer Academic Publishers, 1987.

[Laker94] Kenneth R. Laker and Willy M. C. Sansen. *Design of Analog Integrated Circuits and Systems*. McGraw-Hill, first edition, 1994.

[Lampaert95] Koen Lampaert, Georges Gielen, and Willy M. Sansen. A Performance-Driven Placement Tool for Analog Integrated Circuits. *IEEE J. of Solid-State Circuits*, 30(7):773–780, July 1995.

[Lampaert99] Koen Lampaert, Georges Gielen, and Willy Sansen. *Analog Layout Generation for Performance and Manufacturability*. Kluwer Academic Publishers, first edition, 1999.

[LIP] LIP6 Laboratory, ASIM Dept., http://www-asim.lip6.fr/alliance/. *ALLIANCE: A free VLSI CAD System*.

[Lipman98] Jim Lipman. Multicore chips challenge system-on-chip designers. *EDN Magazine. http://www.ednmag.com/reg/1998/061898/13cs.htm*, June 1998.

[Liu99] Weidong Liu, Xiaodong Jin, James Chen, Min-Chie Jeng, Zhihong Liu, Yuhua Cheng, Kai Chen, Mansun Chan, Kelvin Hui, Jianhui Huang, Robert Tu, Ping K. Ko, and Chenming Hu. *BSIM3v3.2.2 MOSFET Model, Users' Manual*. Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, http://www-device.EECS.Berkeley.EDU/~bsim3, 1999.

[Malavasi95] Enrico Malavasi and Davide Pandini. Optimum CMOS Stack Generation with Analog Constraints. *IEEE Trans. Computer-Aided Design*, 14(1):107–122, January 1995.

[Malavasi96] Enrico Malavasi, Edoardo Charbon, Eric Felt, and Alberto Sangiovanni-Vincentelli. Automation of ICL Layout with Analog Constraints. *IEEE Trans. Computer-Aided Design*, 15(8):923–942, August 1996.

[Marques99] Augusto Marques. *High Speed CMOS Data Converters*. PhD thesis, Katholieke Universiteit Leuven, 1999.

[Mat97] MathWorks, Inc. *MATLAB and SIMULINK User's Guides*, 1997.

[Mead80] Carver Mead and Lin Conway. *Introduction to VLSI systems*. Addison Welsey, 1980.

[Medeiro94] F. Medeiro, B. Pérez-Verdú, A. Rodríguez-Vázquez, and J. L. Huertas. Global design of analog cells using statistical optimization techniques. *Analog Integrated Circuits and Signal Processing, Kluwer Academic Publishers*, 6:179–195, 1994.

[Medeiro95] F. Medeiro, B. Pérez-Verdú, A. Rodríguez-Vázquez, and J. L. Huertas. A Vertically Integrated Tool for Automated Design of $\Sigma\Delta$ Modulators. *IEEE J. of Solid-State Circuits*, 30(7):762–772, July 1995.

[Medeiro99] Fernardo Medeiro, Belen Pérez-Verdú, and Angel Rodríguez-Vázquez. *Top-Down Design of High-Performance Sigma-Delta Modulators*. Kluwer Academic Publishers, 1999.

[Met96] Meta-Software Inc. *HSPICE User's Manual*, 1996.

[Mitra92] S. Mitra, S. K. Nag, R. Rutenbar, and L. Carley. System-level routing of mixed-signal ASICs in WREN. In *Proc. IEEE Int. Conf. on Computer-Aided-Design*, pages 394–399, 1992.

[Mitra96] S. Mitra, R. Rutenbar, L. Carley, and D. J. Allstot. Substrate-aware mixed-signal macro-cell placement in WRIGHT. *IEEE J. of Solid-State Circuits*, 30(3):269–278, March 1996.

[Moser97] Vincent Moser, Hans Peter Amaann, and Fausto Pellandini. Behavioural modelling of analogue systems with absynth. In A. Vachoux, J.-M. Bergé, O. Levia, and J. Rouillard, editors, *Analog and Mixed-Signal Hardware Description Languages*, pages 103–130. Kluwer Academic Publishers, 1997.

[Mukherjee94] Tamal Mukherjee, L. R. Carley, and R. A. Rutenbar. Synthesis of manufacturable analog circuits. In *Proc. IEEE Int. Conf. on Computer-Aided-Design*, pages 586–593, November 1994.

[Nagaraj87] K. Nagaraj, T. R. Viswanathan, K. Singhal, and J. Vlach. Swicthed-capacitor circuits with reduces sensitivity to amplifier gain. *IEEE Trans. on Circuits and Systems*, 34(5):571–574, May 1987.

[Naiiknaware99] Ravindranath Naiiknaware and Terri S. Fiez. Automated hierarchical CMOS analog circuit stack generation woth intramodule connectivity and matching considerations. *IEEE J. of Solid-State Circuits*, 34(3):304–317, March 1999.

[Neff95] Robert Neff. *Automatic Synthesis of CMOS Digital/Analog Converters*. PhD thesis, University of California at Berkeley, http://kabuki.eecs.berkeley.edu/~neff, 1995.

[Nye88] W. Nye, D.C. Riley, A. Sangiovanni-Vincentelli, and A. L. Tits. DELIGHT.SPICE: An Optimization-Based System for Design of Integrated Circuits. *IEEE Trans. Computer-Aided Design*, 7(4):501–518, April 1988.

[Ochotta98] Emil S. Ochotta, Tamal Mukherjee, Rob A. Rutenbar, and L. Richard Carley. *Practical Synthesis of High-Performace Analog Circuits*. Kluwer Academic Publishers, first edition, 1998.

[Onodera90] Hidetoshi Onodera, Hiroyuki Kanbara, and Keikichi Tamaru. Operational-Amplifier Compilation with Performance Optimization. *IEEE J. of Solid-State Circuits*, 25(2):466–473, April 1990.

[Ousterhout94] John K. Ousterhout, editor. *Tcl and the Tk tool kit*. Addison-Wesley Publishing Company, 1994.

[Owen95] Bryn R. Owen, R. Duncan, S Jantzi, C. Ouslis, S. Rezania, and K. Martin. BALLISTIC: An Analog Layout Language. In *Proc. IEEE Custom Integrated Circuits Conf.*, 1995.

[Peluso97] Vincenzo Peluso, Michel S. J. Steyaert, and Willy Sansen. A 1.5V-100μW $\Sigma\Delta$ Modulator with 12-b Dynamic Range Using the Switched-Opamp Technique. *IEEE J. of Solid-State Circuits*, 32(7):943–952, July 1997.

[Peluso98a] Vincenzo Peluso. *Low-voltage, Low-power Design of CMOS $\Delta\Sigma$ A/D Converters*. PhD thesis, Katholieke Universiteit Leuven, 1998.

[Peluso98b] Vincenzo Peluso, Peter Vancorenland, Augusto Marques, Michiel Steyaert, and Willy Sansen. A 900-mV low-power $\Delta\Sigma$ A/D converter with 77-dB dynamic range. *IEEE J. of Solid-State Circuits*, 33(12):1887–1897, December 1998.

[Pétrot94] Frédéric Pétrot. *Outils d'aide au développement de Bibliothèques VLSI portables*. PhD thesis, Université Pierre et Marie Curie, Laboratoire MASI, Paris, http:http://www-asim.lip6.fr/publications/1994/index.gb.html, July 1994.

[Phelps00] Rodney Phelps, Michael J. Krasnicki, Rob A. Rutenbar, L. Richard Carley, and James R. Hellums. A Case Study of Synthesis for Industrial-Scale Analog IP:

Redesign of the Equalizer/Filter Frontend for an ADSL CODEC. In *Proc. Design Automation Conf.*, http://www.dac.com/profrTOC.html, June 2000.

[Porte97] Jacky Porte. *COMDIAC: Compilateur de Dispositifs Actifs, reference manual*. Ecole Nationale Supérieure des Télécommunications, Paris, September 1997.

[Rabii97] Shahriar Rabii and Bruce A. Wooley. A 1.8-V digital-audio $\Sigma\Delta$ modulator in 0.8-μm CMOS. *IEEE J. of Solid-State Circuits*, 32(6):783–796, June 1997.

[Rezania95] Sepideh Rezania. A module generator for general-purpose amplifiers. Master's thesis, University of Toronto, Canada, http://www.eecg.toronto.edu/~analog/HTML/PubsMartin.html, 1995.

[Ribner84] David B. Ribner and Miles A. Copeland. Design Techniques for Cascoded CMOS Op Amps with Improved PSRR and Common-Mode Input Range. *IEEE J. of Solid-State Circuits*, 19(6):919–925, December 1984.

[Rijmenants89] Jef Rijmenants, James Litsios, Thomas R. Schwarz, and Marc G. R. Degrauwe. ILAC: An automated layout tool for analog CMOS circuits. *IEEE J. of Solid-State Circuits*, 22(2):417–425, April 1989.

[Sakurai83] T. Sakurai and K. Tamaru. Simple formulas for two- and three-dimensional capacitances. *IEEE Trans. on Electron Devices*, 30(2):183–185, February 1983.

[Sansen98] W. Sansen, M. Steyaert, V. Peluso, and E. Peeters. Toward sub 1V analog integrated circuits in submicron standard CMOS technologies. In *Proc. IEEE Int. Solid State Circuit Conf.*, pages 186–187, 1998.

[Sauer96] Donald J. Sauer. Constant impedance sampling switch for an analog to digital converter. In *US Patent 5,500,612*, March 1996.

[Schreier] Richard Schreier. *The Delta-Sigma toolbox for MATLAB*. Oregon State University, ftp://next242.ece.orst.edu/pub/delsig.tar.Z.

[Stanisic94] B. R. Stanisic, N. K. Verghese, D. J. Allstot, R. A. Rutenbar, and L. R. Carley. Addressing substrate coupling in mixed-mode ICs: Simulation and power distribution synthesis. *IEEE J. of Solid-State Circuits*, 29(3):226–237, March 1994.

[Steensgaard99] Jesper Steensgaard. Bootstrapped low-voltage analog switches. In *Proc. IEEE Int. Symposium on Circuits and Systems*, volume 2, pages 29–32, May 1999.

[Vandenbussche98] J. Vandenbussche, S. Donnay, G. Gielen, F. Leyn, and W. Sansen. Hierarchical top-down design of analog sensor interfaces: from system-level specifications down to silicon. In *Proc. Design Automation and Test in Europe Conf.*, pages 716–720, March 1998.

[Vanderplaats84] G. V. Vanderplaats. *Numerical Optimization Techniques for Engineering Design with Applications*. McGraw-Hill, 1984.

[Velghe93] R. M. D. A. Velghe, D. B. M. Klaassen, and F. M. Klaassen. Compact MOS modelling for analog circuit simulation. In *Proc. IEEE Int. Electron Devices Meeting*, pages 485–488, 1993.

[VSIA97] VSIA. VSI Alliance Architecture Document (under license), Version 1.0. Technical report, Virtual Socket Interface Alliance, http://www.vsi.org/, 1997.

[Waltari98] Mikko Waltari and K. Halonen. Fully differential switched opamp with enhanced common mode feedback. *IEE Electron. Lett.*, 34(23):2181–2182, November 1998.

[Wang00] Chris Binan Wang. A 20 bit 25khz Delta Sigma A/D converter utilizing frequency-shaped chopper stabilization scheme. In *Proc. IEEE Custom Integrated Circuits Conf.*, pages 9–12, May 2000.

[Wolf99] M. Wolf and U. Kleine. Reliability Driven Module Generation for Analog Layouts. In *Proc. IEEE Int. Symposium on Circuits and Systems*, pages 412–415, June 1999.

[Zwan97] Eric J. van der Zwan. A 2.3 mW CMOS $\Delta\Sigma$ modulator for audio applications. In *Proc. IEEE Int. Solid State Circuit Conf.*, pages 220–221, 1997.