

THÈSE DE DOCTORAT DE L'UNIVERSITÉ PARIS VI

Spécialité : **ARCHITECTURE DES SYSTÈMES INTÉGRÉS ET
MICRO-ÉLECTRONIQUE**

Présentée par : **Vincent BOURGUET**

Pour obtenir le titre de
DOCTEUR DE L'UNIVERSITÉ PARIS VI

**CONCEPTION D'UNE BIBLIOTHÈQUE DE COMPOSANTS
ANALOGIQUES POUR LA SYNTHÈSE ORIENTÉE LAYOUT.**

Soutenue le
14 Novembre 2007

Devant le jury composé de :

M.	Hervé BARTHELEMY	Rapporteur
M.	Jean TOMAS	Rapporteur
Mme.	Andreia CATHELIN	Examineur
M.	Mohamed DESSOUKY	Examineur
M.	Raimundo FREIRE	Examineur
M.	Patrick GARDA	Examineur
M.	Alain GREINER	Directeur de Thèse
Mme.	Marie-Minerve LOUËRAT	Co-directrice de Thèse

Remerciements

Je tiens à remercier chaleureusement Hervé Barthélémy et Jean Tomas, qui m'ont fait l'honneur d'être les rapporteurs de cette thèse. J'ai beaucoup apprécié leurs rapports détaillés qui témoignent de l'intérêt qu'ils ont porté à ce travail. Je remercie également Andreia Cathelin et Patrick Garda d'avoir accepté de participer au jury et dont les remarques ont été extrêmement intéressantes.

J'adresse ma reconnaissance toute particulière à Mohamed Dessouky pour ses précieux conseils, sa gentillesse et ses encouragements tout au long de ces longues années, à Raimundo Freire qui m'a accordé sa confiance pour la suite, ainsi qu'à Alain Greiner pour m'avoir donné l'opportunité d'effectuer ma thèse au sein du département SoC du LIP6.

Je remercie vivement Marie-Minerve Louërat qui a encadré cette thèse. Plutôt que d'énoncer l'ensemble de ses qualités, qu'elle sache que je mesure la chance que j'ai eu de travailler sous sa direction.

Je voudrais également préciser tout le plaisir que m'ont apporté les enrichissantes discussions avec Jacky Porte ou Franck Wajsbürt, que je remercie pour leur enthousiasme et leur gentillesse.

J'associe à ces remerciements les nombreuses personnes du laboratoire LIP6 auprès desquelles j'ai passé de très bons moments. Je pense notamment à Nicolas Beilleau et à ses blagues douteuses, à Laurent De Lamarre et à son inoubliable lancer de café dans l'écran, à Hassan Aboushady et à sa *dancefloor attitude*, à Matthieu Tuna et à sa vision super-sonique à 360° ou encore à Cécile Braunstein et à ses clopes exotiques. Je remercie tous les membres du "LIP6 Poker Tour" qui ne le savent pas encore, mais qui vont contribuer à financer mes recherches futures. Je pense également à Will, Ramy, Hayder, Zied, Christophe, Olivier, Patricia, Maxime, Timothée, Karine, Mounir et à tous les autres.

A titre plus personnel, j'adresse une énorme cumérisation à ceux qui la comprendront, et plus précisément à Bichon et Marie (sans lesquels ce manuscrit n'aurait jamais vu le jour), à Sherman (l'homme qui prend cher), à Jeannot (la femme qui ramène Sherman), à Fredo (qui dort sur place), à Lolo (qui ne dort pas), à Jay (qui vient pour dormir) ou à Pio (qui ne comprend pas pourquoi tout le monde dort). Mais aussi à Michel et Baba avec lesquels j'ai beaucoup travaillé sur la propagation des signaux entre la manette et la console, à Dédé la Biscotte qui doit essayer de moins tricoter, ou à Couette qui aura tenu le "Fonfons FC" à bout de bras.

Je remercie tous les habitants du 13 qui m'ont offert deux années exceptionnelles, et en particulier Jsouille et sa boîte à cochonneries qui auront largement contribué à mener cette thèse à son terme... en 6 ans.

Je remercie tous ceux qui m'ont soutenu de près ou de loin durant cette épreuve comme Gaëlle, Defré, Mel, Tom-Tom, Jax, Gamelle, Thanh, Song, Lorenz, Lise, Toto, Laspac, Nico, Dur, Mathilde, Beno, Steph, Han solo, Philou et Annie, ainsi que tous ceux qui m'ont fait l'immense honneur d'assister à ma soutenance : les beaux Dave et Rico, Rita (merci encore...), Cécile, Guigui, Elodie, Juju, Gwendo et n'oublions pas, **en effet**, Oré.

Je voudrais faire une dédicace très particulière à mes poteaux de toujours : Boss, Tifout et Vini. Ils étaient là lorsque j'ai démarré mes études à l'école maternelle de la croix verte, et ils le sont toujours, presque trente ans plus tard, lorsque je les conclue.

Avant de terminer, je veux remercier de tout mon coeur mes parents pour leur soutien, mon frère pour toute sa gentillesse (remercie également Hervé pour moi...), ma grande sœur pour ses cours d'orthographe, ainsi que mes neveux Baptiste et Antonin. En ce qui concerne mes grands parents, je veux qu'il soit écrit ici, noir sur blanc, à quel point je les aime.

Pour finir, il y en a une qui a vécu un enfer en partageant ma vie de thésard. Malgré cela, elle a conservé son éternel sourire et ne s'est jamais lassée d'écouter mes plaintes alors même que j'ignorais les siennes. Pour cela, et pour les dix années que je viens de passer à ses cotés, je remercie Adeline de tout mon amour.

GENAROOOOO!!!!!!!!!!

Résumé

Les technologies actuelles permettent d'intégrer des systèmes complexes sur une seule puce et on parle alors de SoC (System-on-Chip). La complexité de tels systèmes incite à baser leur conception sur des blocs préexistants et réutilisables, appelés IP (Intellectual Property). Pour les blocs numériques, la synthèse automatisée et donc la réutilisation, sont relativement bien maîtrisées. Cette synthèse s'appuie sur un IP logiciel qui est une description du fonctionnement dans un langage tel que VHDL, et sur une bibliothèque de cellules standard spécifiques à une technologie. En revanche, le grand nombre de performances qui caractérisent un bloc analogique rend la description formelle de son fonctionnement extrêmement difficile. C'est pourquoi un IP analogique est généralement un dessin des masques complètement caractérisé, spécifique à une technologie et dont la réutilisation n'est possible que dans des contextes très similaires. On parle d'IP matériel.

Le projet *CAIRO+* mené au LIP6 cherche à élever le niveau d'abstraction des IP analogiques afin de favoriser leur réutilisation. Il s'inspire pour cela du flot de synthèse utilisé en numérique qui s'appuie sur un langage de description et une bibliothèque de cellules standard. Le langage *CAIRO+* permet au concepteur d'exprimer son expertise et de formaliser ses compromis de conception, tandis que les cellules de la bibliothèque *CAIRO+* sont des générateurs capables de produire un dessin des masques pour différents jeux de spécifications et pour différents procédés de fabrication. Cette thèse s'est intéressée à la conception des générateurs de base de *CAIRO+* et a porté particulièrement sur les problèmes liés à la génération du dessin des masques, capitale en analogique. Elle a conduit à :

- La définition d'une méthodologie de conception d'un générateur de base.
- La réalisation d'une bibliothèque de générateurs de base s'intégrant dans une large gamme d'applications.

Les générateurs réalisés ont été validés sur différentes applications, et ont notamment permis la fabrication d'un circuit de mesure d'appariement de capacités chez STMicroelectronics à Crolles.

Mots Clefs

Circuits analogiques réutilisables, Génération automatique du dessin des masques, Migration technologique.

Abstract

Nowadays, semiconductor technologies allow to integrate complex systems on a single chip. Such systems are called SoC (System-On-Chip). To manage SoCs' complexity, design is based on pre-existing and reusable blocks called IPs (Intellectual Property). For digital blocks, automated synthesis and hence design reuse, is rather well tackled. It is based on a soft IP which is a description of the block's operation provided in a language such as VHDL and on a library of standard cells dedicated to a specific technology. On the other hand, the various performances that characterize an analog block make that until now, there is no systematic way for analog designers to provide a model of the block that can be used at the system level. This is why analog IPs are generally fully characterized layouts dedicated to a single technology. These hard IPs can only be reused in similar contexts.

The *CAIRO+* project developed in the LIP6 laboratory aims to raise the abstraction level of analog IPs in order to favour analog reuse. To achieve this, it takes up the basic ideas of the digital synthesis flow : a description language and a library of standard cells. Thanks to the *CAIRO+* language, the designer is able to express his knowledge in the form of design trade-offs. On the other hand, the cells of the *CAIRO+* library are generators able to produce a layout from different specification sets and for many technologies. This thesis presents the design of these basic generators and focus on the layout generation. It led to :

- The definition of a design methodology for *CAIRO+* basic generators.
- The effective design of a generator library allowing a wide application range.

Proposed generators have been validated with different kind of applications. In particular, they have been used in a circuit measuring capacitor mismatch which has been fabricated in the STMicroelectronics factory, in Crolles.

Keywords

Analog reuse, Analog design automation, Layout generation, Technology migration.

Table des matières

Remerciements	I
Résumé	III
Abstract	V
Table des matières	VII
Table des figures	XI
1 Introduction générale	1
1.1 Motivations	1
1.2 Contribution	2
1.3 Plan du manuscrit	3
2 Problématique	5
2.1 Introduction	5
2.2 Spécificités de la conception analogique	5
2.3 Le langage <i>CAIRO+</i> : cahier des charges	7
2.4 Bibliothèque de cellules génériques : cahier des charges	8
2.5 Conclusion	10
3 Etat de l'art	11
3.1 Introduction	11
3.2 Une conception à différents niveaux d'abstraction	11
3.3 Niveau système	13
3.4 Niveau bloc	13
3.5 Niveau circuit	13
3.6 Cellules de base	15
3.7 Conclusion	17
4 Les dispositifs élémentaires dans l'environnement <i>CAIRO+</i>	19
4.1 Introduction	19
4.2 Le flot de conception avec <i>CAIRO+</i>	19

4.3	Structure hiérarchique d'un générateur <i>CAIRO+</i>	21
4.4	Les étapes de génération d'un circuit	25
4.5	Les étapes de génération vues du dispositif élémentaire	26
4.6	Conclusion	28
5	Vers une bibliothèque de composants génériques	31
5.1	Introduction	31
5.2	Précision : dessin des masques optimisé	32
5.3	Appariement : techniques de dessin dédiées	39
5.4	Variations géométriques : dispositifs élémentaires déformables	49
5.5	Caractérisation du dessin des masques : documentation de la vue structurelle	53
5.6	Indépendance technologique : architecture logicielle adaptée	57
5.7	Procédure de validation	71
5.8	Conclusion	72
6	Méthodologie de conception d'un générateur de dispositifs élémentaires	73
6.1	Introduction	73
6.2	Une méthodologie en quatorze étapes	73
6.3	Conclusion	91
7	Bibliothèque actuelle	93
7.1	Introduction	93
7.2	Bibliothèque <i>libTRANSISTOR</i>	93
7.3	Bibliothèque <i>libMIMCAP</i>	115
7.4	Bibliothèque <i>libPOLYRES</i>	122
7.5	Conclusion	128
8	Applications	129
8.1	Introduction	129
8.2	Intégrateur actif RC différentiel	129
8.3	Amplificateur de transconductance	135
8.4	Circuit de mesure d'appariement de capacités	141
8.5	Conclusion	147
9	Conclusion et Perspectives	149
9.1	Contributions	149
9.2	Perspectives	151
A	Canevas de conception d'un générateur de dispositifs élémentaires	153
B	Structure de retour des générateurs de motifs	159
C	Exemple de fichier DTR	161

D	Syntaxe du fichier de réalisation	167
D.1	fichier de réalisation pour les générateurs MULTI_TR et MULTI_TR_SHD	167
D.2	fichier de réalisation pour le générateur MIM_MATRIX	171
	Liste des publications	173
	Bibliographie	175

Table des figures

2.1	L'inverseur CMOS et sa caractéristique de transfert pour différentes dimensions des transistors	6
2.2	Dessin d'un OTA cascode replié sur deux procédés de fabrication	9
3.1	Les différents niveaux d'abstraction de la conception analogique	12
4.1	Flot de conception avec <i>CAIRO+</i>	21
4.2	Hierarchie d'un convertisseur $\Sigma\Delta$ du second ordre sous <i>CAIRO+</i>	22
4.3	Les trois vues du convertisseur $\Sigma\Delta$ de la figure 4.2	23
4.4	Intégrateur différentiel cascodé mode courant	24
4.5	Dimensionnement électrique d'un transistor	27
4.6	Etapas de la génération avec <i>CAIRO+</i>	28
5.1	Grille de dessin imposée par le fondeur	32
5.2	Repliement d'un transistor et partage de la diffusion	33
5.3	Evolution de l'aire des diffusions en fonction de M	33
5.4	Evolution du périmètre des diffusions en fonction de M pour $W/E=10$	35
5.5	Périmètres moyens pour différents rapports W/E	35
5.6	D'autres effets du repliement : (a) Effet des transistors minces - (b) Réduction de la résistance de grille	36
5.7	Réduction de la résistance d'accès aux diffusions	37
5.8	(a) Dessin des masques d'une capacité MIM - (b) Champ électrique entre les armatures	37
5.9	Excès de gravure (over-etching)	38
5.10	Dessin des masques d'une résistance. (a) Résistance simple (b) Résistance en serpent (c) Serpentin à jonctions métalliques	39
5.11	Histogramme du <i>mismatch</i> pour un échantillon de 30 unités	40
5.12	Variation de l'épaisseur d'oxyde	41
5.13	Inter-digitation de transistors pour lesquels $M=3$	42
5.14	Dessin à géométrie centrée pour deux transistors à $M=4$	43
5.15	(a) Excès de gravure pour les doigts extrêmes - (b) Insertion de transistors fictifs . . .	43
5.16	Effet d'ombre	44
5.17	Transistors appariés vis-à-vis de l'effet d'ombre	44

5.18	Deux capacités de rapport $C1/C2 = 4/5$ appariées vis-à-vis de l'excès de gravure . . .	46
5.19	Capacité non-unitaire. (a) Rectangle - (b) Stub - (c) Trou	47
5.20	Dessin des masques de deux capacités appariées	48
5.21	Dessin des masques de deux résistances appariées : $R1 = R2$	48
5.22	Fonction de forme d'un transistor	49
5.23	Dessin d'un motif : un doigt de transistor	52
5.24	Placement et routage des motifs	52
5.25	Estimation des performances	54
5.26	Capacités parasites de routage dans (a) un transistor et (b) une capacité	55
5.27	Modèle utilisé pour un segment résistif	56
5.28	Les trois vues d'un dispositif élémentaire après création	58
5.29	DAPI : Calcul de W_{min}	60
5.30	Vues réelle et symbolique d'un motif	61
5.31	Jeu de règles CAIRO+	63
5.32	Type de calcul de la MAPI	64
5.33	Exemple de macro-rectangle	65
5.34	Fichier DTR : Syntaxe et accès	67
5.35	Interface entre le dispositif élémentaire et le procédé de fabrication	69
5.36	Procédure de validation d'un générateur de dispositifs élémentaires	72
6.1	Schéma électrique du dispositif BICAPA : $C2 = R.C1, R \in \mathbb{Q}$	74
6.2	Vue structurelle du dispositif BICAPA	74
6.3	Deux configurations de dessin des masques d'une capacité MIM	75
6.4	Plans de masse pour le dispositif BICAPA : (a) $R = 9/7$ dans une matrice 4×4 - (b) $R = 9/7$ dans une matrice 6×3	76
6.5	BICAPA : Règles de dessin intervenant dans le calcul de la largeur minimale de l'armature haute	81
6.6	Motifs du dispositif BICAPA	84
6.7	Connexion entre un via semi-symbolique et un fil de routage	84
6.8	Schéma de conception du motif <i>Celem_NE</i>	85
6.9	Graphe des dépendances horizontales pour le générateur <i>bicapa_elem</i>	87
6.10	<i>bicapa_elem</i> : Règles de dessin intervenant dans le calcul de la largeur de la boîte d'aboutement	89
6.11	Motifs obtenus	90
6.12	Dessin des masques final. $Celem = 50fF, n1 = 7, n2 = 9$	91
7.1	Générateurs de motifs pour <i>libTRANSISTOR</i>	94
7.2	Schéma électrique du dispositif <i>TR_MOS</i>	99

7.3	Différents dessins des masques réalisés par le dispositif TR_MOS : a- Type N avec anneau de garde, $W = 15\mu m$, $L = 350nm$, Technologie 120nm. b- Type N avec anneau de garde, $W = 15\mu m$, $L = 350nm$, Technologie 350nm. c- Type P, $W = 35\mu m$, $L = 3\mu m$, Technologie 180nm	101
7.4	Fonctions de forme du dispositif TR_MOS, Technologie 120nm ($\lambda = 120nm$), $L = L_{min}$: a- $W = 3\mu m$. b- $W = 50\mu m$. c- Distribution des points de la fonction de forme pour $W = 50\mu m$	103
7.5	Schéma électrique des générateurs DP_MOS_xx	104
7.6	Différents dessins des masques réalisés par les dispositifs DP_MOS_ID et DP_MOS_CC : a- DP_MOS_ID, type N, $W = 25\mu m$, $L = 340nm$, Technologie 120nm. b- DP_MOS_CC, type N avec anneau de garde, $W = 25\mu m$, $L = 340nm$, Technologie 120nm. c- DP_MOS_CC, type P avec anneau de garde, $W = 15\mu m$ et $L = 2\mu m$, Technologie 350nm	106
7.7	Fonctions de forme d'une paire différentielle en technologie 350nm pour $W = 25\mu m$ et $L = 350nm$ ($\lambda = 325nm$) : a- DP_MOS_CC b- DP_MOS_ID	107
7.8	Schéma électrique des dispositifs CM_MOS_xx	108
7.9	Différents dessins des masques réalisés par les dispositifs CM_MOS_ID et CM_MOS_CC : a- CM_MOS_ID, type N, $W = 25\mu m$, $L = 340nm$, Technologie 120nm. b- CM_MOS_CC, type N avec anneau de garde, $W = 25\mu m$, $L = 340nm$, Technologie 120nm. c- CM_MOS_CC, type P avec anneau de garde, $W = 15\mu m$ et $L = 2\mu m$, Technologie 350nm	109
7.10	Schéma électrique du dispositif MULTI_MOS	110
7.11	Dessin des masques du dispositif MULTI_MOS : $W_{SUB} = 9\mu m$, $L = 500nm$, $M_{SUB} = 3$, technologie 120nm	112
7.12	Dessin des masques du dispositif MULTI_MOS pour 2 transistors P identiques. $W_{SUB} = 6\mu m$, $L = 1.4\mu m$, $M_{SUB} = 2$, technologie 350nm	113
7.13	Fonction de forme du dispositif MULTI_MOS pour les conditions de la figure 7.12 ($\lambda = 325nm$)	113
7.14	Schéma électrique du dispositif MULTI_MOS_SHD	114
7.15	Dessin des masques du dispositif MULTI_MOS_SHD pour $W_{SUB} = 9\mu m$, $L = 500nm$, $M_{SUB} = 4$, technologie 120nm	116
7.16	Motif pour la réalisation de capacités uniques	117
7.17	Motifs utilisés pour les matrices de capacités : (a) Capacité fictive - (b) Capacité unitaire - (c)(d)(e) Capacités non-unitaires (resp. rectangle, stub et trou)	117
7.18	Schéma électrique du dispositif CMIM	119
7.19	Dessins des masques du dispositif CMIM. $C = 1pf$, technologie 350nm	120
7.20	Dessins des masques du dispositif CMIM. $C = 1pf$, technologie 120nm	120
7.21	Fonction de forme du dispositif CMIM ($\lambda = 120nm$)	121
7.22	Schéma électrique du dispositif MIM_MATRIX pour $N=3$	122
7.23	Dessins des masques du dispositif MIM_MATRIX. $C_{elem} = 0,6pf$, technologie 120nm	123
7.24	Motifs utilisés par les dispositifs résistifs	124
7.25	Schéma électrique du dispositif HIPORES	125

7.26	Dessins des masques du dispositif HIPORES. $R = 60k\Omega$, technologie $120nm$	126
7.27	Fonction de forme du dispositif HIPORES ($\lambda = 120nm$)	126
7.28	Schéma électrique du dispositif BIRES	127
7.29	Dessins des masques du dispositif BIRES. $R = 60k\Omega$, technologie $120nm$	127
7.30	Fonction de forme du dispositif BIRES ($\lambda = 120nm$)	128
8.1	Intégrateur actif RC différentiel	129
8.2	Schéma électrique interne de l'amplificateur différentiel	130
8.3	Arbre d'instanciation du générateur correspondant au circuit de la figure 8.2	131
8.4	Dessin des masques de l'intégrateur actif RC différentiel	133
8.5	Réponse en fréquence de l'intégrateur	134
8.6	Réponse en fréquence de l'AOP	134
8.7	OTA type N à 2 étages et à compensation Miller	135
8.8	Schéma d'instanciation du générateur d'OTA deux étages avec minimisation de la tension de décalage systématique	137
8.9	Dessin des masques d'un OTA deux étages pour les spécifications du tableau 8.3	139
8.10	Schéma d'instanciation du générateur d'OTA deux étages sans minimisation de la tension de décalage systématique	140
8.11	Dessin des masques d'un OTA deux étages sans minimisation de la tension de décalage systématique	141
8.12	Filtre passe-bas : Biquad à capacités commutées	142
8.13	Les deux matrices de capacités pour le circuit de mesure d'appariement	143
8.14	(a) Réponses en fréquence obtenues pour les 31 circuits fabriqués - (b) Ecart type	144
8.15	Circuit à grille pseudo-flottante	145
8.16	Erreurs d'appariement obtenues pour chaque matrice	146
8.17	Dessin des masques complet	146
D.1	Exemple : génération de 4 transistors avec MULTI_TR	168
D.2	Exemple de placement obtenu	168
D.3	Exemple de routage obtenu	169
D.4	Conflit de routage	170

Chapitre 1

Introduction générale

1.1 Motivations

Les circuits intégrés actuels visent la mise en œuvre de systèmes embarqués sur puce (System-On-Chip) constitués de blocs numériques et analogiques et contrôlés par un système d'exploitation logiciel. La complexité d'une telle démarche impose l'utilisation de blocs préexistants communément appelés IP (Intellectual Property).

Dans le domaine numérique, la nature discrète de l'information permet de définir un IP comme un automate d'état. Celui-ci est généralement décrit en langage VHDL et l'on parle d'"IP logiciel" (Soft IP). La synthèse automatisée d'un bloc numérique sur un procédé donné est alors possible et elle s'appuie sur une bibliothèque de cellules standard dont le dessin des masques est spécifique à la technologie cible. La portabilité des IP logiciels numériques permet de réduire considérablement le temps de conception des blocs numériques sur les SoC. Elle suppose cependant l'existence d'un langage de description formelle et d'une bibliothèque de cellules standard.

A l'inverse, les circuits analogiques traitent des signaux continus qui portent une information beaucoup plus complexe. Ainsi, le comportement de ces circuits est caractérisé par un ensemble de performances (gain statique, bande passante, marge de phase, excursion, etc...) qui dépend de la nature et de la complexité du circuit. De plus, pour une même fonctionnalité électrique, la ou les performances visées en priorité diffèrent d'une application à l'autre. Dès lors, un IP analogique ne peut se résumer à une description formelle du fonctionnement. Il faut en effet que toutes les performances du bloc soient caractérisées. C'est pourquoi un IP analogique est généralement un bloc purement matériel ("IP matériel" ou Hard IP) : il s'agit d'un dessin des masques complètement caractérisé et spécifique à une technologie [Hamour03]. Dans ces conditions, la portabilité de l'IP vers d'autres spécifications ou vers d'autres procédés technologiques devient très difficile, et le temps de conception des blocs analogiques est un problème majeur de la conception actuelle [MEDEA05].

Afin de rendre possible la portabilité des IP analogiques, il convient de concevoir des IP paramétrables, capables de prendre en compte les performances visées et d'effectuer les compromis

nécessaires pour les atteindre. Cette idée se heurte cependant à deux problèmes :

1. **La traduction formelle des spécifications au niveau circuit en dimensions électriques pour les composants élémentaires.** Celle-ci repose habituellement sur l'expertise du concepteur qui assume les conséquences des compromis effectués. Cette traduction nécessite donc la définition d'un langage de description permettant au concepteur d'exprimer ses connaissances.
2. **La traduction formelle des dimensions électriques des composants élémentaires en dimensions géométriques pour effectuer le dessin des masques.** La portabilité d'un IP analogique sur différents jeux de spécifications implique de grandes variations sur les dimensions électriques des composants élémentaires. Il faut ainsi disposer d'une bibliothèque de cellules génériques, capables de déterminer la géométrie des masques qui correspond aux dimensions électriques désirées.

Le projet *CAIRO+* [Dessouky01a, NGuyen06] mené au LIP6 fournit un langage adapté à la conception d'IP analogiques paramétrables. On propose qu'un IP analogique soit un programme en langage *CAIRO+*. Celui-ci est paramétré par les performances et la technologie visées, et délivre un schéma électrique dimensionné, une estimation des performances et un dessin des masques du circuit. Ce programme, que l'on appelle **générateur**, permet notamment au concepteur de formaliser la traduction des spécifications en dimensions électriques pour les composants élémentaires.

Un générateur *CAIRO+* est une structure hiérarchique qui fait appel à des générateurs de niveau fonctionnel inférieur. Il s'appuie donc sur une bibliothèque de générateurs de base qui représentent les composants électroniques élémentaires comme le transistor, la capacité ou la résistance. Ces générateurs de base sont capables de produire et de caractériser un dessin des masques qui correspond aux dimensions électriques visées pour le composant. C'est donc à leur niveau qu'est prise en charge la traduction des dimensions électriques en dimensions géométriques pour la réalisation physique.

Finalement, le projet *CAIRO+* cherche à répondre aux besoins de la conception analogique en fournissant :

- Un langage permettant de définir des IP logiciels paramétrables.
- Une bibliothèque de cellules standard génériques.

1.2 Contribution

Cette thèse s'inscrit dans le cadre du projet *CAIRO+* et vise la constitution d'une bibliothèque de cellules analogiques standard paramétrables et génériques vis-à-vis des spécifications et du procédé de fabrication. Elle se concentre particulièrement sur les problèmes liés à la réalisation physique de ces cellules.

Malheureusement, constituer une bibliothèque qui permette de couvrir un très large spectre d'applications n'est pas suffisant. En effet, un nouveau procédé de fabrication introduit parfois de nouvelles contraintes qu'il est difficile d'anticiper et qui inciteront à la conception de nouveaux composants. Par ailleurs, des applications très spécifiques pourront nécessiter l'utilisation de cellules *ad hoc* qui, là encore, devront enrichir cette bibliothèque. C'est pourquoi notre contribution portera sur deux points :

1. **La définition d'une méthodologie de conception des générateurs de base** : Comme nous allons le voir, la conception des générateurs de base est une tâche complexe. Chacun d'entre eux intègre une expertise spécifique à son dimensionnement électrique, ou à son dessin des masques. Notre méthodologie permet de systématiser la conception d'un tel générateur et facilite ainsi la réalisation, la maintenance et l'enrichissement éventuel de la bibliothèque.
2. **La réalisation d'une bibliothèque de générateurs de base** : Cette réalisation s'appuie sur la méthodologie définie. La bibliothèque réalisée permet la conception de circuits analogiques pour une large gamme d'applications.

1.3 Plan du manuscrit

Ce manuscrit est organisé en neuf chapitres :

- Après une brève introduction au premier chapitre, nous exposerons les motivations et les objectifs de notre travail au chapitre 2.
- Le troisième chapitre explore l'état de l'art dans le domaine des outils d'aide à la conception de circuits analogiques. Nous y verrons en quoi *CAIRO+* se démarque de chacun des outils existants.
- Le chapitre 4 présente brièvement l'environnement *CAIRO+* et décrit les étapes de la génération d'un circuit dans cet environnement. Nous y verrons notamment le rôle des générateurs de base dans le flot de conception.
- Le chapitre 5 recense l'ensemble des contraintes liées à la conception d'un générateur *CAIRO+* de base. Il montre comment ces contraintes ont été prises en compte.
- Les chapitres 6 et 7 exposent les résultats obtenus. Le chapitre 6 présente la méthodologie de conception que nous avons définie, tandis que les cellules constituant la bibliothèque actuelle sont présentées dans le chapitre 7.
- Le chapitre 8 illustre des applications de notre travail. Nous y présenterons notamment un circuit conçu sous *CAIRO+* et fondu par *STMicroelectronics*.
- Enfin, le chapitre 9 conclura le manuscrit et proposera d'éventuelles améliorations à apporter à ce travail.

Chapitre 2

Problématique

2.1 Introduction

Ce chapitre présente les principaux problèmes que nous devons résoudre. Nous verrons tout d'abord dans la section 2.2 les spécificités de la conception analogique, et en quoi elles nuisent à la synthèse automatique ou à la réutilisation. La section 2.3 montre qu'une partie des problèmes liés à ces spécificités peuvent être résolus par la mise en œuvre d'un flot de conception intégrant deux axes de communication. Dans la section 2.4, nous dégagerons les objectifs principaux de ce travail. Nous y énumérerons les contraintes que doivent satisfaire les générateurs *CAIRO+* de base pour répondre aux problèmes de plus bas niveau, qui concernent le dessin des masques notamment.

2.2 Spécificités de la conception analogique

Si la mise en œuvre d'un flot de conception automatisé est relativement bien maîtrisée dans le domaine numérique, elle reste très difficile pour les circuits analogiques. Les principales particularités de la conception analogique qui expliquent cette difficulté sont énumérées dans [Gielen91]. Voici celles qui nous paraissent fondamentales et qui ont guidé notre recherche [Dessouky01a] :

- 1. Il existe différentes topologies réalisant la même fonctionnalité.**

Ces topologies présentent des caractéristiques spécifiques et peuvent atteindre des performances différentes. Ce sont l'application visée et l'expérience du concepteur qui déterminent le choix de la topologie finale.

- 2. Les différents niveaux de la hiérarchie interagissent fortement.**

Tous les éléments d'un circuit analogique interagissent. Ces influences mutuelles rendent difficile la définition d'une hiérarchie, particulièrement utile dans la conception automatisée. Ainsi, la structure hiérarchique d'un circuit analogique n'est jamais globalement admise par tous les concepteurs. Une autre conséquence de ces influences est que lorsqu'une hiérarchie a été adoptée, le comportement d'un bloc est largement conditionné par celui des blocs de

niveau hiérarchique inférieur. Il est en particulier nécessaire de déterminer les tensions, les courants ou les impédances à tous les niveaux. Ceci introduit de fortes interactions entre les différents niveaux hiérarchiques.

3. Le dimensionnement des composants élémentaires est critique vis-à-vis des performances du circuit.

Bien que le nombre de transistors par bloc soit nettement inférieur à celui trouvé dans les circuits numériques, le dimensionnement des composants élémentaires est particulièrement critique en analogique. Dans ce domaine, les composants sont généralement placés dans des zones de fonctionnement sensibles. A titre d'exemple, considérons le montage de la figure 2.1. Il s'agit bien évidemment de l'inverseur CMOS bien connu des numériciens. Ce montage est également utilisé en analogique comme amplificateur de tension.

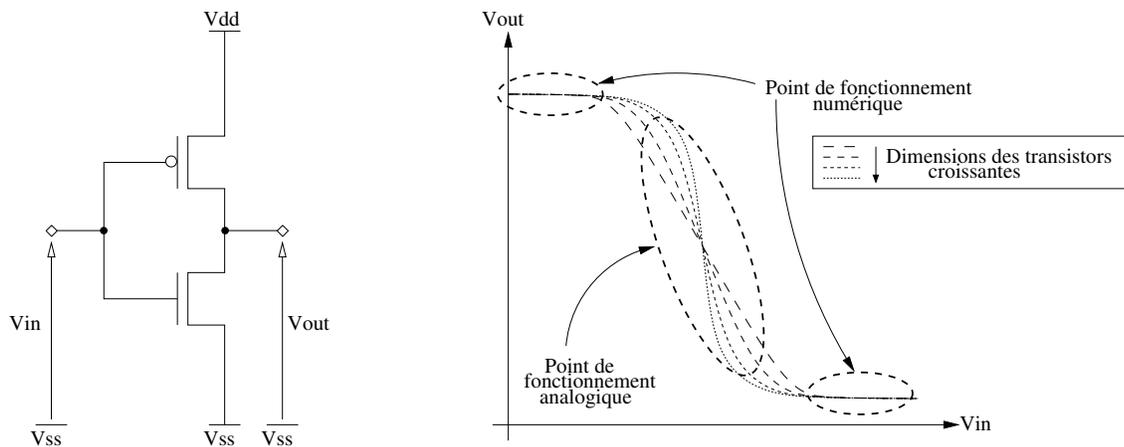


FIG. 2.1 : L'inverseur CMOS et sa caractéristique de transfert pour différentes dimensions des transistors.

Si l'on observe la caractéristique de transfert présentée sur cette même figure, on remarque que numériciens et analogiciens n'ont pas la même utilisation du montage. Pour les numériciens, la sortie n'a idéalement que deux valeurs possibles, tandis que les analogiciens se placent justement sur la zone linéaire pour obtenir le gain de l'amplificateur. Or, si les dimensions des transistors ont une grande influence sur la pente dans la zone linéaire, on peut constater qu'elles n'ont pas le même impact pour les numériciens.

4. Le dessin des masques a une grande influence sur les performances.

Puisque le dimensionnement des composants élémentaires est critique, il convient de les réaliser le plus fidèlement possible. Le dessin des masques d'un circuit analogique doit donc être extrêmement précis. Cette précision concernera les dimensions effectives des composants mais aussi leur appariement qui, comme nous le verrons, tient souvent une place importante dans le dessin. Par ailleurs, les défauts de fabrication ou les parasites altèrent les performances finales. Ces imperfections dépendent notamment de la qualité du dessin des masques et

doivent être prises en compte pendant le dimensionnement pour atteindre les performances désirées.

5. **Les dimensions des composants d'un même circuit peuvent varier de plusieurs ordres de grandeur selon les spécifications ou le procédé de fabrication visés.**

Pour permettre la réutilisation, *CAIRO+* devra être capable de s'adapter à ces grandes variations. Une des principales difficultés ici est la manipulation d'objets de tailles très hétérogènes.

6. **Les performances sont particulièrement sensibles aux paramètres technologiques.**

L'indépendance technologique visée par *CAIRO+* nécessite donc une caractérisation précise et systématique des technologies envisagées.

Le projet *CAIRO+* propose un langage permettant la description d'IP analogiques paramétrables et une bibliothèque de composants permettant la synthèse de ces IP. Ces deux aspects du projet sont nécessaires pour faire face à l'ensemble des difficultés que nous venons de citer.

2.3 Le langage *CAIRO+* : cahier des charges

Pour s'adapter aux spécificités de la conception analogique, le langage *CAIRO+* doit satisfaire un certain nombre de contraintes. Nous avons d'abord souligné que plusieurs topologies pouvaient atteindre une même fonctionnalité. Le choix d'une topologie est déterminé par l'application mais reste relativement subjectif et lié à l'expérience du concepteur. Lorsque l'on a l'ambition de proposer des IP analogiques réutilisables, ce choix est un problème majeur. Nous avons décidé de le contourner en nous plaçant au niveau circuit : un générateur *CAIRO+* est lié à une topologie donnée. Ces IP structurés (Firm IP [Saleh06]) se situent entre les IP purement logiciels, et les IP purement matériels : ils définissent le schéma d'interconnexion d'un circuit, et les informations nécessaires à son dimensionnement et à sa réalisation.

D'un autre côté, pour faire face aux difficultés désignées par les points 2, 3 et 4, et en particulier aux fortes interactions entre les niveaux hiérarchiques, le langage *CAIRO+* doit permettre d'instaurer deux types de communications :

- Une **communication verticale** entre les différents niveaux de la hiérarchie qui illustre les interactions soulignées au point 2. Elle permet à chaque niveau hiérarchique d'imposer des spécifications aux niveaux inférieurs ou de les questionner afin de déterminer leurs possibilités en termes de performances. On peut ainsi mettre en œuvre une traduction descendante des spécifications jusqu'à l'obtention des dimensions des composants de base et une estimation ascendante de la faisabilité et des performances associées. Evidemment,

pour que les informations transitant dans ce canal de communication vertical soient valides, elles doivent s'appuyer sur une modélisation précise des composants de base (point 3).

- A chaque niveau hiérarchique, une **communication transversale** entre le dimensionnement électrique et la génération des masques. Comme le souligne effectivement le point 4, le dimensionnement électrique d'un circuit analogique doit être fait en prenant en compte l'impact du dessin des masques. Dans *CAIRO+*, ces deux étapes sont synchronisées par une communication permanente.

Le langage *CAIRO+* [NGuyen06, NGuyen04] ne sera pas abordé dans cette thèse qui se concentre sur la conception de la bibliothèque de cellules de bases. En revanche, le chapitre 4 montre comment ces communications interviennent dans le flot de génération.

2.4 Bibliothèque de cellules génériques : cahier des charges

Dans la liste que nous avons établie, les points 4, 5 et 6 soulignent l'importance du dessin des masques dans la conception de circuits analogiques. Dès lors que l'on envisage une approche hiérarchique, le dessin des éléments actifs, c'est-à-dire celui des composants, est pris en charge par les cellules de base. Celles-ci se situent donc au cœur du problème de la conception analogique automatisée [ITRS06].

Dans notre approche, on ajoute une ambition supplémentaire : les cellules de base de *CAIRO+* doivent également permettre la génération du dessin sur différents procédés de fabrication. Ainsi, elles doivent finalement se soumettre aux contraintes suivantes :

- **Précision des masques** : Dans les circuits numériques, puisque les transistors ne sont utilisés que dans deux états (bloqué ou passant), la précision de la réalisation n'est pas capitale. Certains environnements de synthèse de circuits numériques effectuent d'ailleurs le dessin des masques sur une grille symbolique dont la précision est nettement inférieure à celle permise par la technologie [Greiner95]. De tels environnements peuvent également se permettre de ne proposer qu'une gamme discrète de puissances de sortie (*fanout*) pour chaque élément de la bibliothèque. En effet, surévaluer la puissance de sortie d'une porte logique ne conduit pas à un mauvais fonctionnement du circuit.
A l'inverse, le point 4 souligne l'influence du dessin sur les performances d'un circuit analogique. Il conviendra donc que les masques générés soient aussi précis que possible et on assurera une précision au pas de la grille de dessin imposée par le fondeur.
- **Appariement** : En analogique, il est fréquent que le rapport de dimensionnement entre différents composants soit très important. Le cas échéant, on visera une précision maximale quant à ce rapport. Pour cela, on aura recours à des techniques de dessin adaptées [Hastings01].

- **Caractérisation** : Les parasites introduits par la réalisation ont un impact significatif sur le comportement du circuit. Pour atteindre des performances données, il est donc nécessaire de prendre en compte ces parasites lors du dimensionnement. Nos composants devront ainsi être en mesure de caractériser les parasites qu'ils introduisent [Dessouky00].
- **Déformation** : Comme l'indique le point 5, la modification des spécifications ou du procédé cible modifie les dimensions des composants. Par ailleurs, ces dimensions n'évoluent que rarement dans le même sens et ceci entraîne la manipulation de blocs de tailles hétérogènes. La figure 2.2 illustre ce problème en présentant le dessin des masques d'un amplificateur de transconductance (OTA) réalisé sur deux technologies différentes. Ces deux dessins sont obtenus dans le cas de spécifications imposant un produit gain-bande de 50MHz. Si l'on s'intéresse à la paire différentielle présente dans ce circuit, on s'aperçoit qu'en passant du procédé $0.6\mu\text{m}$ au procédé $0.25\mu\text{m}$, les dimensions des deux transistors qui la composent passent de $W = 140\mu\text{m}$ et $L = 3.95\mu\text{m}$, à $W = 76,8\mu\text{m}$ et $L = 3,1\mu\text{m}$.

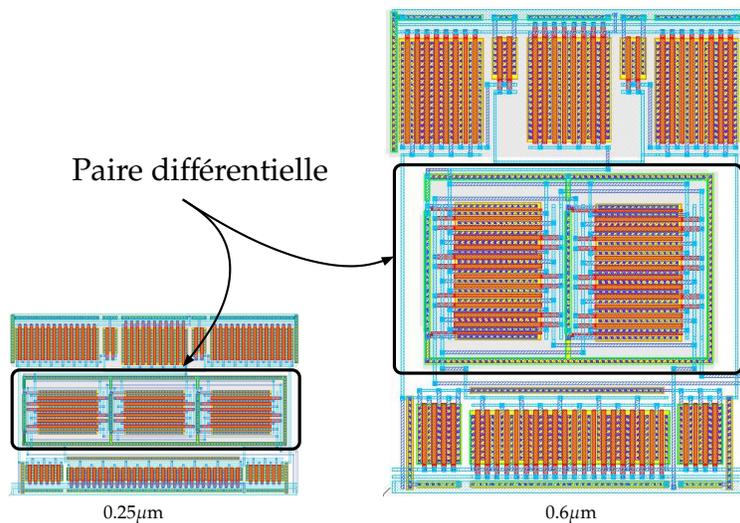


FIG. 2.2 : Dessin d'un OTA cascode replié sur deux procédés de fabrication.

Dans le but d'obtenir finalement un dessin des masques compact, nos générateurs devront proposer plusieurs réalisations physiques pour des spécifications électriques et un procédé de fabrication donnés. Chacune de ces réalisations présentera un facteur de forme différent, et les contraintes géométriques guideront le choix de la réalisation finale.

- **Migration technologique** : Pour finir, on désire que la description d'un circuit en CAIRO+ soit indépendante du procédé de fabrication visé. Ceci implique que les cellules de base soient génériques vis-à-vis de ce procédé. Elles seront donc en mesure de prendre en compte les paramètres et les règles de dessin imposées par la technologie. Ces paramètres et ces règles devront être décrits dans un format standard.

Du fait de ces contraintes, les concepteurs analogiciens sont habitués à faire un dessin des masques optimisé vis-à-vis de l'application et sont réticents à envisager une conception automatisée. Nous allons voir dans cette thèse qu'en identifiant les paramètres d'un dessin analogique, nous avons pu définir un niveau d'abstraction qui rend possible la génération d'un dessin optimisé sur différents procédés de fabrication et pour différents jeux de spécifications appliqués au circuit.

2.5 Conclusion

Nous avons énuméré dans ce chapitre les principales caractéristiques qui rendent la conception de circuits analogiques difficilement automatisable. Nous avons remarqué que pour rendre possible l'automatisation, la prise en charge des difficultés doit être partagée entre un langage de description et une bibliothèque de composants adaptés. Nous avons plus particulièrement défini les objectifs de ce travail en recensant les contraintes liées à la conception des cellules de base de *CAIRO+*.

Chapitre 3

Etat de l'art

3.1 Introduction

Ce chapitre présente l'état de l'art dans le domaine de la conception analogique automatisée. Les trois niveaux d'abstraction généralement considérés sont présentés dans la section 3.2. Les sections 3.3 et 3.4 présentent les méthodes et les outils qui ont respectivement été proposés pour la conception au niveau système et au niveau bloc. La section 3.5 aborde les outils qui, comme *CAIRO+*, permettent la synthèse au niveau circuit. Les outils travaillant à ce niveau sont hiérarchiques et se basent sur une bibliothèque de cellules qui posent des problèmes spécifiques. La section 3.6 montre les solutions que l'on peut trouver à ces problèmes dans la littérature.

3.2 Une conception à différents niveaux d'abstraction

La conception d'un système analogique se fait suivant plusieurs niveaux d'abstraction :

- **Niveau système** : A ce niveau, il s'agit d'abord de choisir une architecture efficace pour le système, puis de définir les blocs fonctionnels nécessaires ainsi que leurs spécifications.
- **Niveau bloc** : Un bloc est caractérisé par sa fonction et son interface. Le travail du concepteur consiste ici à choisir une topologie qui permette d'atteindre les spécifications issues du niveau système et de traduire ces spécifications pour les sous-blocs.
- **Niveau circuit** : A ce niveau, le schéma électrique est fixé. Le concepteur doit ici effectuer le dimensionnement des composants du circuit afin que celui-ci atteigne les performances désirées. Les imperfections du dessin des masques ou les parasites qu'il met en jeu doivent être pris en compte. Au niveau circuit, l'automatisation de la conception s'appuie sur une bibliothèque d'éléments de base. C'est à ce niveau qu'intervient *CAIRO+* qui s'appuie, lui, sur les cellules que nous allons présenter dans cette thèse.

Comme le montre la figure 3.1, l'approche généralement envisagée présente une phase descendante qui propage les spécifications et effectue le dimensionnement, et une phase ascendante qui prend en charge le dessin des masques [Gielen00]. Après chaque étape de la conception, une vérification est effectuée par simulation pour valider :

- le dimensionnement pendant la phase descendante.

- le dessin des masques pendant la phase ascendante.

A chaque niveau d'abstraction, des outils ont été proposés pour assister la conception analogique. Au niveau système, le concepteur utilise typiquement des outils tels que MATLAB [Matlab07], ou des langages de description comme VHDL-AMS [VHDL07] afin de modéliser le système. Il peut ainsi effectuer des vérifications haut niveau et valider ses choix de conception.

Au niveau bloc et au niveau circuit, plusieurs travaux se sont intéressés à la définition d'IP analogiques paramétrables. Associés à une méthode de synthèse, ces IP permettent d'obtenir un dessin des masques quels que soient les spécifications et le procédé de fabrication visés. Au niveau bloc cependant, le flot de synthèse doit intégrer une étape de sélection de la topologie. Il est donc spécifique au bloc considéré.

Les sections 3.3 et 3.4 présentent brièvement quelques outils proposés dans la littérature, qui facilitent la conception au niveau système et au niveau bloc, respectivement. Pour ces niveaux, un état de l'art plus détaillé est donné dans [Gielen00] et, plus récemment, dans [Saleh06].

Les sections 3.5 et 3.6 détaillent plus particulièrement les outils de définition et de synthèse d'IP analogiques au niveau circuit. C'est en effet à ce niveau que se proposent d'intervenir CAIRO+ et le travail présenté dans cette thèse.

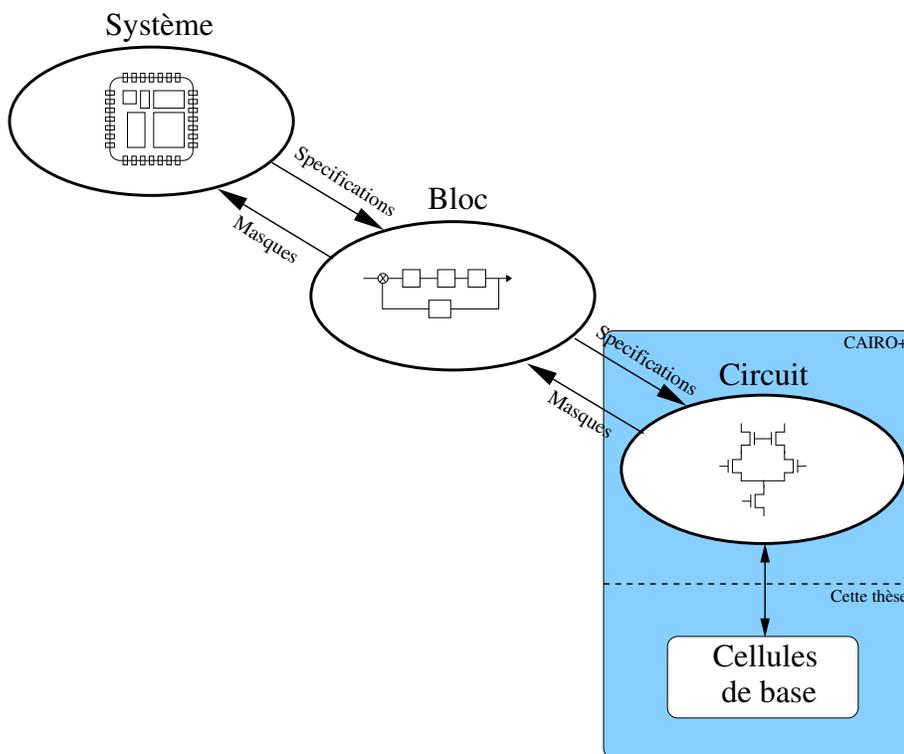


FIG. 3.1 : Les différents niveaux d'abstraction de la conception analogique.

3.3 Niveau système

Le choix de l'architecture d'un système analogique et des blocs fonctionnels le constituant est l'une des étapes la plus difficile à automatiser. En effet, il existe souvent un grand nombre de solutions et beaucoup de compromis doivent être effectués. De plus, ces compromis dépendent fortement de l'application et de l'environnement. L'expertise du concepteur est donc ici largement sollicitée. C'est pourquoi les outils de conception automatisée intervenant à ce niveau travaillent sur des architectures données. Par exemple, [Vassiliou96] a proposé une méthodologie de conception d'un système de pilotage vidéo constitué d'un synthétiseur de fréquence, de trois convertisseurs numérique-analogique, et d'un banc de registres. De même, dans [Donnay97], la synthèse d'un système d'interface pour un capteur de radiations a été décrite. L'architecture de ce système est invariante. Elle est basée sur une chaîne d'amplification, un échantillonneur-bloqueur, et un convertisseur analogique-numérique.

Pendant la phase ascendante de conception, le dessin des masques se fait par placement et routage des blocs fonctionnels instanciés.

3.4 Niveau bloc

Afin de réaliser la fonction spécifique à un bloc et d'atteindre les performances demandées par le niveau système, plusieurs topologies sont possibles. A ce niveau, il s'agit donc de choisir la topologie la mieux adaptée à l'application et de déterminer les spécifications qui doivent lui être imposées. Par exemple, dans [Van der Plas01], le choix d'une topologie a été effectué par élimination, en appliquant différents filtres à une bibliothèque de topologies. Les blocs adressés par cette méthode, et les topologies disponibles ne sont cependant pas clairement définis. Dans [Jusuf90], la synthèse d'un convertisseur analogique-numérique a été mise en oeuvre. Elle s'appuie sur une bibliothèque de trois *générateurs de netlist*.

Au niveau bloc, le dessin des masques correspond également au placement et au routage des sous-blocs.

3.5 Niveau circuit

Au niveau circuit, le schéma électrique à réaliser est clairement défini. Il reste à déterminer les dimensions de tous les composants et les tensions de polarisation afin d'obtenir les performances exigées par le niveau bloc, puis à dessiner les masques du circuit. Ces deux tâches dépendent l'une de l'autre dans la mesure où le dessin des masques doit être conforme aux résultats du dimensionnement, et que le dimensionnement doit se faire en prenant en compte les imperfections et les parasites introduits par le dessin des masques. Les itérations de conception entre le dimension-

nement et le dessin des masques entraînent un temps de conception long et l'automatisation au niveau circuit est un point clé de la conception analogique automatisée [Gielen00].

En ce qui concerne le dimensionnement électrique, on trouve deux types d'approche pour l'automatisation :

- **Celles basées sur des méthodes d'optimisation** (*Optimisation Based*) dans lesquelles le dimensionnement est vu comme un problème d'optimisation visant à minimiser une fonction de coût. Celle-ci représente les objectifs et les contraintes du dimensionnement. Le processus d'optimisation est itératif et s'appuie sur une estimation des performances obtenues après chaque itération. Cette estimation peut être basée sur des équations, comme dans GP-CAD [Hershenson98] ou ASTRX/OBLX [Ochotta96], ou sur des outils de simulation, comme dans ANACONDA [Phelps00] .

Du fait de la spécificité de chaque circuit analogique, l'une des principales difficultés ici est de déterminer une fonction de coût satisfaisante. Par ailleurs, les algorithmes utilisées entraînent généralement un temps de calcul long.

- **Celles basées sur l'expertise du concepteur** (*Knowledge Based*) qui cherchent à automatiser une méthodologie de dimensionnement systématique définie par le concepteur. Par exemple, [Gaubert05] a décrit une méthode de dimensionnement d'un amplificateur faible bruit. Ici, le concepteur est donc mis à contribution pour définir les calculs et les compromis à effectuer. Dans OCEANE [Porte07], l'expertise est intégrée à l'outil pour le dimensionnement d'une grande gamme de circuits. Dans [Castro02], un langage adapté est proposé au concepteur afin qu'il puisse exprimer un plan de conception. Bien qu'avec ce type d'outils, la synthèse soit relativement rapide, l'introduction d'un nouveau circuit nécessite un effort de conception important. C'est la raison pour laquelle les outils basés sur l'expertise du concepteur sont généralement mieux adaptés aux circuits usuels, qui seront souvent réutilisés.

Les méthodes de génération du dessin des masques peuvent également s'appuyer sur des algorithmes d'optimisation et/ou sur l'expertise du concepteur. Dans le cas où une optimisation est effectuée, elle cherche typiquement à obtenir un placement et un routage qui minimisent l'aire totale, la longueur des fils et les différents parasites. On trouvera par exemple dans cette catégorie [Lampaert95] ou encore [Malavasi96]. Lorsque l'expertise du concepteur est requise, elle peut s'exprimer sous forme de patrons (*template*) qui indiquent le placement relatif des éléments [Dessouky01a, Conway92], ou de procédures de dessin [Owen95b].

CAIRO+ permet d'automatiser à la fois le dimensionnement et le dessin des masques au niveau circuit. Il se base pour cela sur l'expertise du concepteur en lui fournissant un langage grâce auquel celui-ci peut formaliser ses plans de conception. Pour capturer l'expertise du concepteur, certains outils préfèrent mettre à sa disposition une interface graphique adaptée [Cadence07]. La souplesse d'un langage permet cependant le traitement d'une gamme de circuits nettement plus

importante [Castro06]. D'autres recherches fondées sur un langage ont fait le pari de s'appuyer sur des outils commerciaux. Par exemple, [Castro02] est basé sur le langage SKILL de Cadence et [Owen95b] sur le langage Lx de Mentor Graphics. A l'inverse, le langage CAIRO+ est une surcouche du langage C++. On envisage ainsi de l'intégrer à l'initiative System-C/AMS [SYSTEM07] qui vise la normalisation d'un langage de description de systèmes mixtes.

Quelle que soit l'approche proposée, les outils d'aide à la conception au niveau circuit sont hiérarchiques et s'appuient donc sur une bibliothèque de cellules de bases. La section suivante explore l'état de l'art concernant la conception de ces cellules.

3.6 Cellules de base

Dans la littérature, peu de travaux s'intéressent à une bibliothèque de cellules servant de base à la fois au dimensionnement électrique et au dessin des masques. Une telle bibliothèque est pourtant capitale pour la réutilisation d'IP analogiques. En effet, la synthèse de ces IP sur différents procédés de fabrication ou pour différents jeux de spécifications n'est envisageable qu'en disposant de cellules présentant des caractéristiques particulières. Pour chacune de ces caractéristiques, voyons quelles sont les propositions qui ont été formulées pour répondre aux problèmes introduits :

- **Prise en compte des contraintes de dimensionnement** : En analogique, il est fréquent que plusieurs éléments soient liés par des contraintes de dimensionnement ou de polarisation. Pour satisfaire ces contraintes, [Graeb01] introduit un niveau hiérarchique terminal "paire de transistors" dans lequel on trouve des montages tels que la paire différentielle ou le miroir de courant. Pour garantir le fonctionnement de ce type de montage, une série de règles de polarisation est définie. Ces règles peuvent être intégrées à un algorithme de dimensionnement automatique.
En ce qui concerne le dessin des masques, il faut régulièrement assurer l'appariement entre différents éléments. Là encore, la bibliothèque proposée dans [Rijmenants89] intègre des cellules non-minimales qui représentent plusieurs transistors, capacités ou résistances. Des générateurs spécifiques permettent alors d'effectuer un dessin spécifique pour ces cellules.
- **Dessin des masques optimisé** : En analogique, la qualité de la réalisation a une forte influence sur les performances du circuits. Le dessin des masques doit ainsi minimiser les effets parasites. Pour cela, [Malavasi95] et [Naiknaware99] ont proposé de représenter le circuit par un graphe à partir duquel les empilages de transistors sont générés de façon à minimiser la surface totale et les parasites liées aux diffusions ou au routage.
- **Indépendance vis-à-vis du procédé de fabrication** : La conception d'une bibliothèque de cellules indépendantes de la technologie a été proposée pour les circuits numériques dans [Pétrot94]. Les dimensions géométriques de tous les objets du dessin des masques y sont définies par rapport à une unité symbolique appelée λ dont la valeur réelle dépend du procédé visé. Une proposition de même type a été faite pour l'analogique par [Rijmenants89]

et [Castro02] qui utilisent des couches et des règles symboliques pour définir le dessin des masques. Une base de données extérieure y donne la nature des couches réelles et la valeur des règles réelles pour chaque procédé. Par ailleurs, [Owen95a] a indiqué qu'une représentation physique indépendante de la technologie nécessite une description relative du placement des objets.

- **Déformation** : Des IP paramétrables doivent être en mesure de produire des dessins présentant des facteurs de forme variables. Par exemple, une méthode de génération automatique est proposée dans [Mathias94]. En s'appuyant sur trois briques de base, cette méthode permet d'obtenir des transistors de formes variable, pas nécessairement rectangles. Pour caractériser ces déformations, [Conway92] recense les facteurs de forme possibles dans une fonction de forme. Celle-ci est construite de manière ascendante à partir de la fonction de forme des éléments de base. Cette méthode a également été utilisée dans [NGuyen06].
- **Caractérisation précise** : Au niveau circuit, le dimensionnement doit prendre en compte les imperfections du dessin pour assurer les performances. Dans [Dessouky01a], le dimensionnement se base sur une estimation des parasites. Cette estimation est obtenue en anticipant le dessin des masques. Dans [Agarwal04], les parasites sont préalablement mesurés par simulation pour différents résultats de dimensionnement. Ces mesures sont stockées dans des tables de recherche (*look-up tables*) à partir desquelles une interpolation linéaire est effectuée à chaque nouvelle génération.

Les cellules *CAIRO+* de base qui sont proposées dans cette thèse servent de base à la fois au dimensionnement électrique et au dessin des masques. Elles sont pour cela caractérisées par trois vues sur lesquelles nous reviendrons dans le chapitre suivant :

- Une vue structurelle qui permet notamment de définir des contraintes de polarisation ou d'appariement.
- Une vue comportementale qui modélise précisément chaque cellule.
- Une vue physique qui représente un dessin des masques optimisé et entièrement caractérisé.

Elles se différencient en cela des différentes propositions trouvées dans la littérature qui ne peuvent pas s'intégrer dans toutes les phases de la synthèse d'IP analogiques.

La réutilisation de circuits analogiques, et en particulier leur migration sur différents procédés de fabrication, n'est possible qu'en disposant d'outils d'automatisation qui permettent la synthèse de ces circuits. C'est la raison pour laquelle nous nous sommes concentrés dans ce chapitre sur les travaux permettant d'automatiser le flot de conception. Notons toutefois que certaines études ont porté exclusivement sur des méthodes de migration de blocs analogiques qui conservent les performances [Francken99, Galup02, Hammouda04, Levi06, Hammouda06].

3.7 Conclusion

Dans ce chapitre, nous avons présenté les recherches proposant des outils d'automatisation à tous les niveaux de la conception analogique. Nous avons vu en particulier les travaux qui, comme *CAIRO+*, ont porté sur le niveau circuit qui est le niveau clé de la définition d'IP analogiques. Enfin, nous avons vu les solutions proposées dans la littérature aux différents problèmes posés par la conception de cellules de base génériques sur lesquelles peuvent s'appuyer les processus hiérarchique de dimensionnement électrique et de dessin des masques.

Chapitre 4

Les dispositifs élémentaires dans l'environnement *CAIRO+*

4.1 Introduction

Ce chapitre montre le rôle des générateurs de base dans l'environnement *CAIRO+*. Nous y introduisons les notions nécessaires à la compréhension de cette thèse. La section 4.2 décrit le flot de conception d'un circuit analogique avec *CAIRO+*. Celui-ci s'appuie sur une représentation hiérarchique du circuit présentée dans la section 4.3. On y verra notamment où se situent les générateurs de base dans cette hiérarchie. La section 4.4 montre comment les informations transitent dans la hiérarchie lors de chacune des étapes de la génération d'un circuit. La section 4.5 revient plus particulièrement sur les générateurs de base. Elle indique comment ceux-ci traitent les informations qui leur parviennent des niveaux hiérarchiques supérieurs.

4.2 Le flot de conception avec *CAIRO+*

Le flot classique de conception d'un circuit analogique commence par le choix d'une topologie capable d'atteindre le comportement désiré. Lorsque le schéma électrique est défini, l'étape suivante consiste à attribuer des dimensions électriques aux éléments du circuit telles que celui-ci respecte les performances attendues. Il s'agit alors de générer le dessin des masques correspondant à ces dimensions, puis de vérifier la validité de ce dessin en procédant à une extraction et une simulation. La plupart du temps, il faut corriger le dimensionnement initial pour contrer l'effet des parasites introduits par la réalisation physique et effectuer ainsi plusieurs itérations de dimensionnement. Dans le meilleur des cas, chacune de ces étapes de conception est prise en charge par un outil spécifique. Cependant, il est fréquent que le dessin des masques, qui intègre une grande expertise, soit réalisé à la main, pénalisant ainsi considérablement le temps de conception. Quoiqu'il en soit, le nombre d'itérations nécessaires pour obtenir les performances visées entraîne un temps de conception trop important. De plus, un changement éventuel de procédé de fabrication réinitialise complètement le flot, et toutes les étapes doivent être reprises.

Pour accélérer la démarche de conception, l'environnement CAIRO+ intègre :

- Un langage qui permet la description de procédures de dimensionnement indépendantes de la technologie cible et basées sur une modélisation précise des composants. Cette modélisation est identique à celle utilisée par les simulateurs.
- Un flot de génération automatisée du dessin des masques indépendant de la technologie cible. Grâce à la précision des modèles intégrés dans CAIRO+, le dessin obtenu est parfaitement caractérisé.
- Des structures de données communes permettant une communication privilégiée entre les processus de dimensionnement électrique et de génération des masques.

Ainsi, toutes les étapes impliquées dans les itérations de dimensionnement sont automatisées dans un environnement unique qui facilite par ailleurs leur communication. Le temps de conception d'un circuit est considérablement amélioré par cette approche. D'un autre côté, puisque le flot est indépendant de la technologie, un changement de procédé devient particulièrement aisé.

Avant d'étudier plus en détails le flot rendu possible par CAIRO+, voyons quelles sont ces structures de données sur lesquelles s'appuie chaque générateur. Elles correspondent en fait aux trois vues du bloc analogique considéré :

- **La vue structurelle** représente le schéma électrique visé. En début de flot, cette vue contient la topologie du circuit, c'est-à-dire le schéma d'interconnexion des composants pour lesquels les dimensions électriques ne sont pas encore définies. Ce n'est qu'au terme d'un premier dimensionnement qu'on leur affecte des dimensions. Ces dimensions évolueront ensuite au cours de la génération.
- **La vue comportementale** définit les aptitudes de la topologie en termes de performances. Tout se passe comme si l'on considérait l'ensemble des dimensionnements électriques envisageables pour la topologie afin d'établir l'espace des performances possibles.
- **La vue physique** contient le dessin des masques du circuit. Avant la génération effective de ces masques, elle est initialisée par une description du placement relatif des composants sur le silicium. Cette vue dépend des dimensions présentes dans la vue structurelle ainsi que du procédé de fabrication.

En s'appuyant sur ces trois vues, le flot CAIRO+ illustré sur la figure 4.1 peut être mis en œuvre. Si les spécifications visées pour le circuit rentrent dans le domaine de conception défini par la vue comportementale, le générateur procède au dimensionnement électrique. Celui-ci va traduire les spécifications en dimensions électriques pour les composants électroniques élémentaires. Ces dimensions sont attribuées aux composants de la vue structurelle. La lecture de la vue structurelle dimensionnée permet ainsi la génération des masques qui produit la vue physique du circuit. A l'issue de cette génération, les dimensions contenues dans la vue structurelle sont mises à jour avec les valeurs effectivement générées. De plus, des informations relatives aux parasites introduits lors de la réalisation peuvent également être associées à la vue structurelle. Ainsi, un nouveau dimensionnement peut, si nécessaire, être effectué en prenant en compte les informations

de la vue structurelle mise à jour. Au terme de ces itérations, le dessin des masques final du circuit est disponible. Pour valider ce dessin, il faut effectuer une extraction, et simuler le résultat de cette extraction. Si les performances obtenues ne correspondent pas aux spécifications initiales, le concepteur disposera des options suivantes :

- Affiner le dimensionnement électrique en revoyant le calcul effectué par son générateur.
- Se laisser une marge d'ajustement sur les spécifications. Cette solution, fréquemment utilisée, consiste à surévaluer les spécifications initiales pour corriger les effets non pris en compte par le dimensionnement électrique.
- Changer de procédé de fabrication.

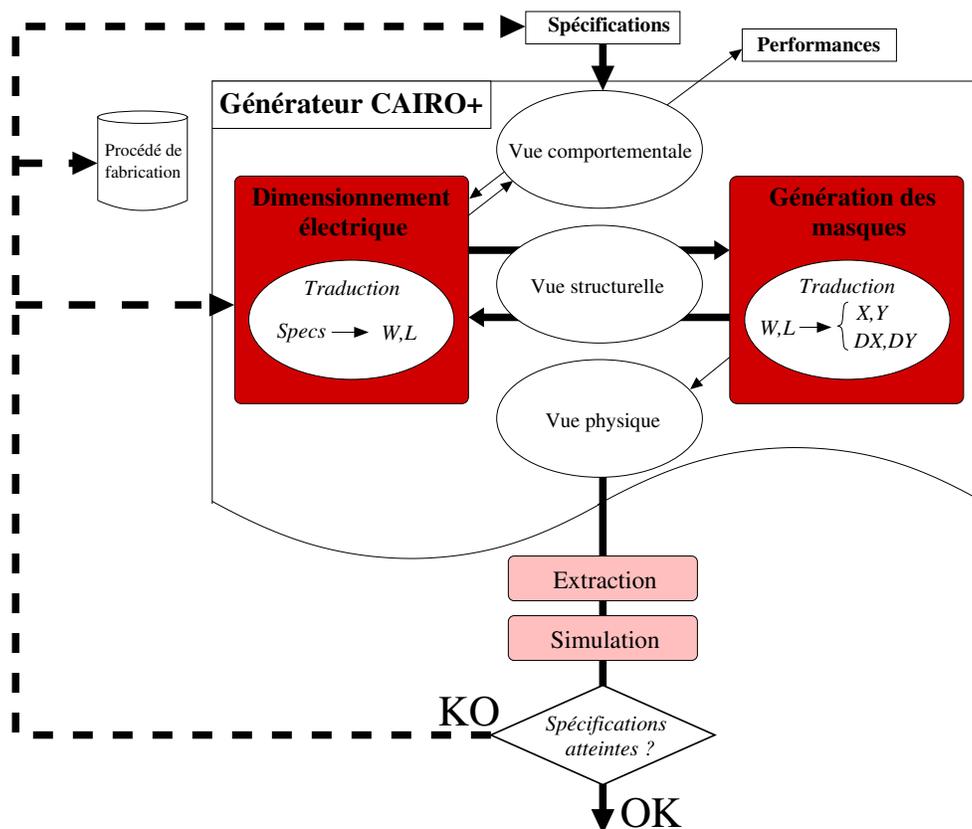


FIG. 4.1 : Flot de conception avec CAIRO+.

4.3 Structure hiérarchique d'un générateur CAIRO+

Pour mettre en œuvre le flot de la figure 4.1, CAIRO+ représente un circuit par une structure hiérarchique. Celle-ci peut être illustrée par un arbre d'instanciation dont les nœuds sont appelés **modules**. La figure 4.2 donne l'arbre d'instanciation d'un convertisseur $\Sigma\Delta$ du second ordre. Celui-ci instancie :

- deux intégrateurs différentiels (INT1 & INT2) constitués d'un amplificateur à transistors

(AMPLI) et de deux capacités (CAPAS).

- deux convertisseurs numérique vers analogique (DAC1 & DAC2).
- un comparateur (COMP) à base de transistors (TRANS).

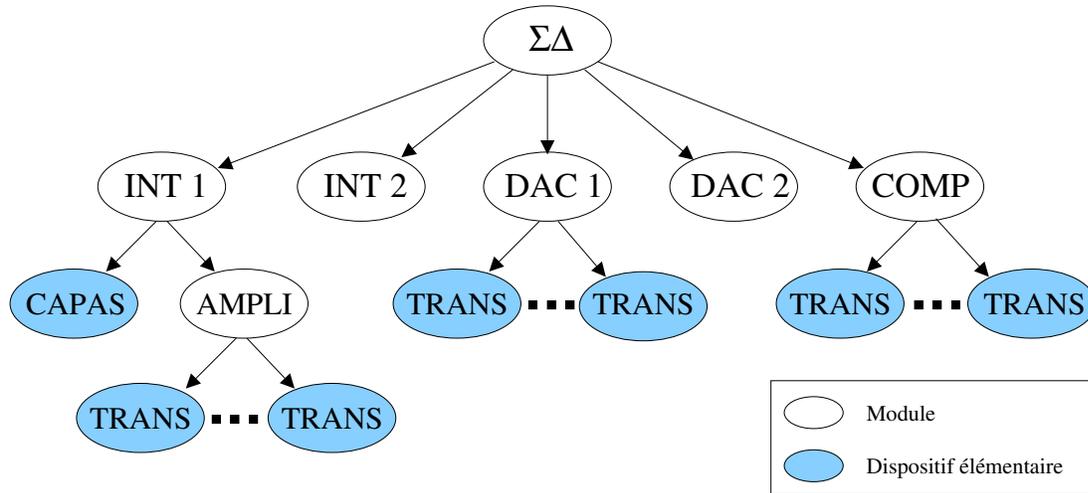


FIG. 4.2 : Hiérarchie d'un convertisseur $\Sigma\Delta$ du second ordre sous CAIRO+.

Chaque module est créé par un générateur CAIRO+ et contient ainsi les trois vues associées à ce générateur. Ces vues sont également des structures hiérarchiques qui respectent l'arbre d'instanciation :

- La vue structurelle instancie les vues structurelles des modules de niveau directement inférieur et décrit leurs interconnexions.
- La vue comportementale s'appuie également sur le comportement des modules directement inférieurs pour indiquer au niveau supérieur son propre comportement.
- La vue physique décrit simplement le placement relatif des modules de niveau directement inférieur sur le silicium.

La figure 4.3 illustre le contenu hiérarchique des trois vues du module $\Sigma\Delta$ de la figure 4.2.

Les modules présents au niveau élémentaire de la hiérarchie ne peuvent évidemment pas s'appuyer sur un niveau hiérarchique inférieur. La génération de ces éléments se fait de manière indépendante et son résultat conditionne la génération globale du circuit. Ces modules, appelés **dispositifs élémentaires**, sont issus des générateurs CAIRO+ de base qui font l'objet de cette thèse. Ils correspondent aux composants électroniques qui interviennent dans le circuit. Il s'agit en général de composants uniques tels qu'un transistor, une capacité ou encore une résistance.

Cependant, la hiérarchie implicitement manipulée par les concepteurs s'arrête souvent à un niveau plus complexe [Graeb01]. Il est fréquent, en effet, que plusieurs composants soient fortement liés électriquement et qu'ils soient considérés comme une unique primitive. A titre d'exemple, considérons le dimensionnement de l'intégrateur Gm-C de la figure 4.4. Le schéma de ce circuit étant parfaitement symétrique, on peut ne considérer que la partie droite de l'axe de symétrie. Les

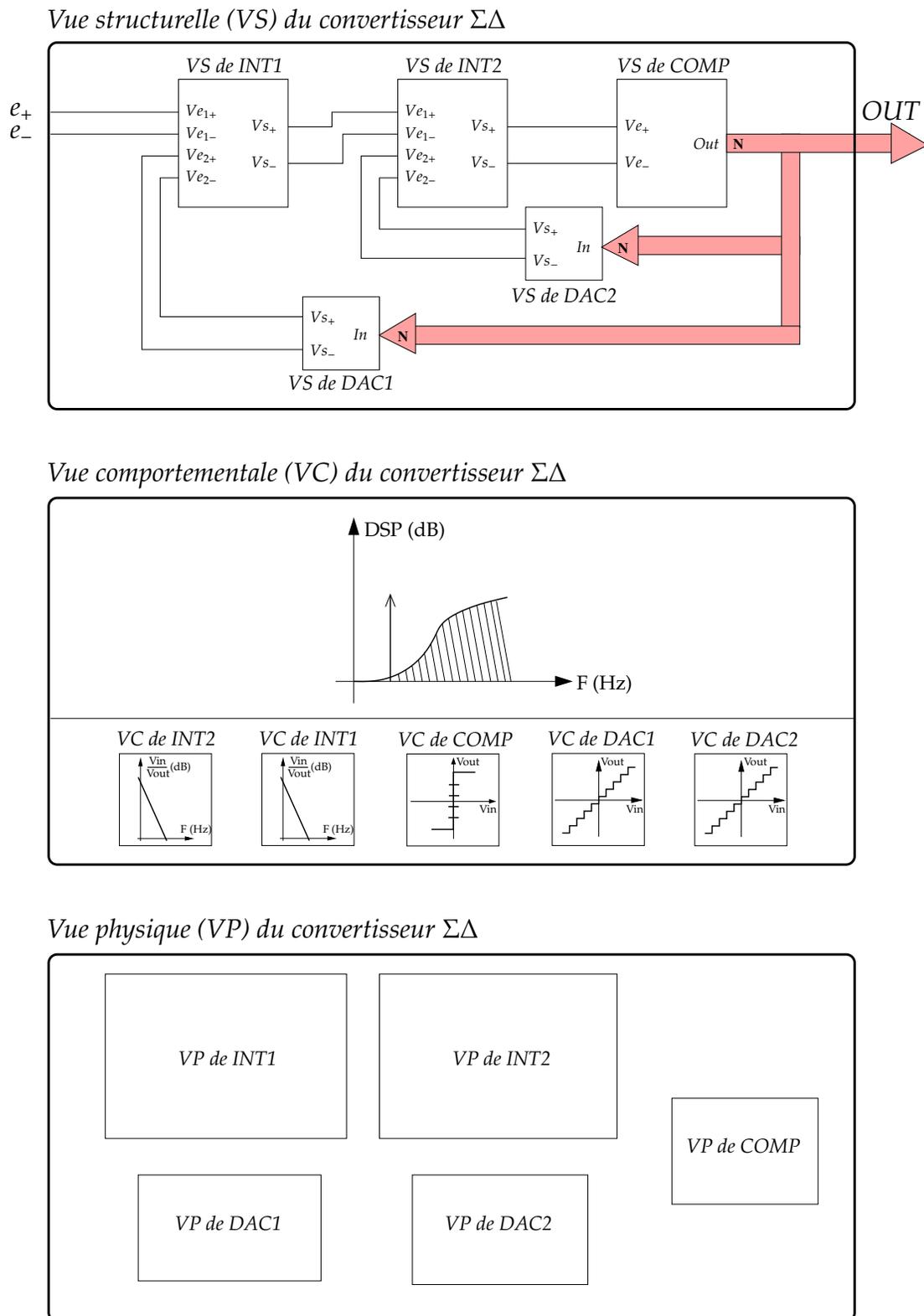


FIG. 4.3 : Les trois vues du convertisseur $\Sigma\Delta$ de la figure 4.2.

trois transistors de chaque ligne de ce schéma simplifié doivent alors être parfaitement identiques pour garantir le comportement de l'intégrateur [Aboushady02]. Finalement, sur le schéma complet, les six transistors de chaque ligne sont vus par le concepteur comme une unique primitive de dimensionnement.

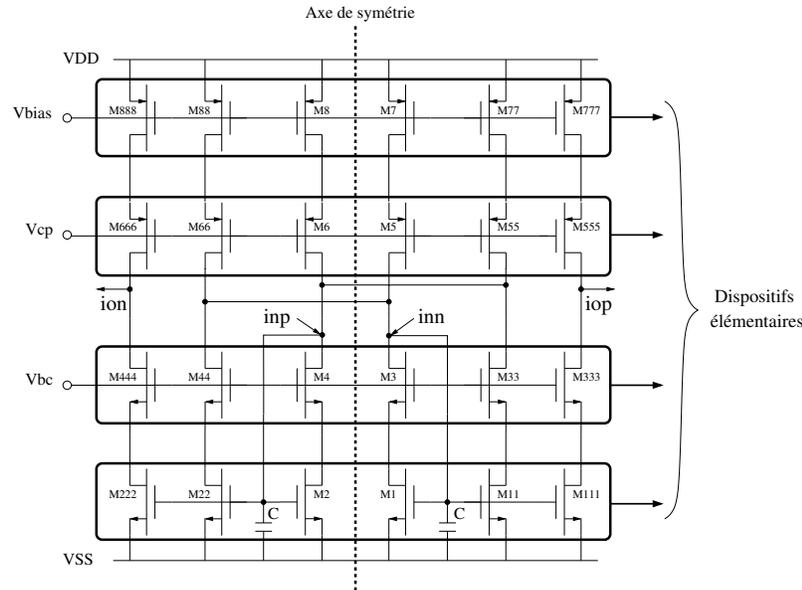


FIG. 4.4 : Intégrateur différentiel cascadé mode courant.

En conception analogique, on utilise régulièrement des dispositifs qui reposent sur une forte corrélation entre les éléments qui les composent. Le miroir de courant ou la paire différentielle sont deux montages extrêmement répandus constitués de deux transistors qui doivent être parfaitement identiques [Maloberti01, Johns97, Laker94, Razavi01, Porte07]. Dans CAIRO+, les procédures de dimensionnement traitent simultanément les deux transistors de ce genre de montage [deLamarre02, Iskander06]. Au niveau de leur réalisation, on vise une précision maximale sur le rapport des dimensions. Nous verrons au chapitre 5 que pour assurer cette précision, on doit utiliser des techniques de dessin dédiées. La plupart du temps, ces techniques consistent à mélanger les masques de tous les composants concernés afin de répartir équitablement les parasites et les imperfections de la fabrication. Tout cela incite à encapsuler ces composants dans un même dispositif élémentaire. Notre bibliothèque tient compte de ce type de contrainte en proposant des dispositifs élémentaires multi-composants. Par exemple, les intégrateurs de la figure 4.2 utilisent un dispositif élémentaire réalisant deux capacités appariées (CAPAS).

Remarquons enfin que l'un des intérêts de notre méthode est que le nombre de générateurs nécessaires à la conception d'un circuit est souvent très inférieur au nombre de modules impliqués dans la hiérarchie. En effet, des modules représentant la même fonction analogique sont en général issus d'un même générateur auquel on a adressé des paramètres éventuellement différents. Par exemple, tous les transistors de la hiérarchie présentée sur la figure 4.2 sont créés par un même générateur. De même, les deux intégrateurs ou les deux DACs sont respectivement créés

par un unique générateur. Deux modules issus du même générateur s'appuient sur le même arbre d'instanciation sous-jacent. En revanche, puisque l'on a adressé à ce générateur des paramètres différents, les vues caractéristiques des modules obtenus seront différentes à l'issue de la génération. Les dimensions des composants de leur vue structurelle seront différentes, et en conséquence, leur vue physique finale sera différente.

4.4 Les étapes de génération d'un circuit

Nous allons maintenant voir comment les différents modules de la hiérarchie interagissent pendant chacune des étapes de la génération d'un circuit. Sous *CAIRO+*, la génération se fait en quatre étapes :

- **Création** : L'arbre d'instanciation est créé en suivant l'idée que chaque module appelle la création de ses propres fils. Cette étape démarre donc par la création des dispositifs élémentaires et se fait selon un parcours ascendant de la hiérarchie. A la fin de l'étape de création, on dispose du schéma électrique hiérarchique non dimensionné du circuit dans la vue structurelle, et de la description du placement physique relatif de tous les modules dans la vue physique.
- **Dimensionnement** : Les spécifications appliquées au module de plus haut niveau se propagent dans la hiérarchie. Pour cela, chaque module dispose d'un ensemble de procédures de dimensionnement. Certaines lui permettent de traduire le jeu de spécifications qu'il reçoit en différents jeux de spécifications à appliquer à chacun de ses fils. D'autres lui permettent d'indiquer au module de niveau supérieur la faisabilité de la requête. Ces dernières constituent la vue comportementale du circuit. L'étape de dimensionnement est alors effectuée en deux phases. La première phase est l'exploration du domaine de conception (EDC) durant laquelle chaque module vérifie la faisabilité des spécifications qui lui sont adressées en interrogeant les aptitudes de ses fils. Cette phase correspond à un parcours globalement descendant de la hiérarchie qui va jusqu'aux dispositifs élémentaires. En cas d'échec au niveau élémentaire, ce parcours est cependant ponctué par une remontée du diagnostic jusqu'à un module capable de le traiter. La deuxième phase est la propagation hiérarchique effective des spécifications du module principal jusqu'aux dimensions électriques pour les dispositifs élémentaires. Puisque la validité de ces spécifications a été vérifiée, cette phase est sans échec. Il s'agit ici d'un parcours descendant de la hiérarchie.
- **Placement** : Chaque module effectue le placement physique de ses fils. Il s'appuie pour cela sur la description de leur placement relatif contenue dans la vue physique. Ceci n'est possible que dans la mesure où les masques des dispositifs élémentaires ont effectivement été générés. Cette étape est donc initiée par la génération des masques des dispositifs élémentaires et correspond à un parcours ascendant de la hiérarchie.
- **Routage** : Chaque module prend en charge le routage physique liant ses fils. Il s'agit également d'un parcours ascendant de la hiérarchie. A l'heure actuelle, ce routage est effectué de manière procédurale.

Les étapes de la génération d'un module CAIRO+ se traduisent donc par autant de parcours de l'arbre structurel. Il est évident alors que cette génération est fortement liée à celle des dispositifs élémentaires que l'on étudie ici.

4.5 Les étapes de génération vues du dispositif élémentaire

La génération des dispositifs élémentaires se distingue de celle des modules classiques dans la mesure où ceux-ci n'ont pas de module fils. Ils doivent par conséquent suivre une démarche autonome tout en proposant au niveau hiérarchique supérieur l'interface fonctionnelle d'un module classique. L'autre singularité à ce niveau est que l'on prend en charge le dimensionnement et la réalisation physique des éléments actifs du circuit. Il est donc nécessaire ici de disposer des informations électriques et physiques relatives à la technologie. Voyons comment ces contraintes s'intègrent dans les étapes de la génération au niveau élémentaire :

1. **Création** : Comme nous l'avons souligné, la création de l'arbre d'instanciation se fait de façon ascendante et est ainsi initiée par la création des dispositifs élémentaires. Ceux-ci se doivent donc de présenter une vue structurelle et une vue physique sur lesquelles pourront s'appuyer les modules de niveau supérieur pour effectuer leur propre création. Pour construire la vue structurelle d'un dispositif élémentaire, on utilise une bibliothèque d'**éléments structurels de base**. Celle-ci permet d'instancier des éléments uniques tels qu'un transistor, une capacité ou une résistance. Par exemple, la vue structurelle du dispositif "transistor simple" instanciera un unique transistor structurel, tandis que celle du dispositif "paire différentielle" instanciera deux transistors structurels et décrira leurs interconnexions. En revanche, aucune dimension électrique n'est pour le moment attribuée à ces éléments. En ce qui concerne la vue physique, le dispositif élémentaire se contente de créer une boîte d'aboutement vide. Celle-ci sert de base à la description du placement relatif et sera comblée après la génération effective des masques.
2. **Dimensionnement** : A l'instar d'un module classique, un dispositif élémentaire dispose d'un ensemble de procédures qui lui permet :
 - d'indiquer à son supérieur s'il est en mesure d'atteindre des spécifications demandées. Ces procédures constituent la vue comportementale du dispositif élémentaire.
 - de calculer les dimensions électriques des composants qu'il contient pour atteindre ces spécifications.

Il peut ainsi s'intégrer dans les deux phases du dimensionnement d'un circuit décrites à la section précédente. Les calculs effectués par ces procédures s'appuient sur une modélisation des composants semblable à celle que l'on trouve dans les simulateurs usuels. L'accès au modèle se fait par un ensemble de fonctions que nous avons appelé la **calcullette** [Iskander04, deLamarre02]. Chaque type de composant (transistor, capacité ou résistance) dispose de sa propre calcullette. En ce qui concerne le transistor, elle intègre le modèle Bsim3v3 [Cheng99], développé à l'université de Berkeley et très répandu chez les fondeurs. La figure 4.5 illustre le rôle de la calcullette dans le dimensionnement électrique d'un transistor. Pour ce dispositif,

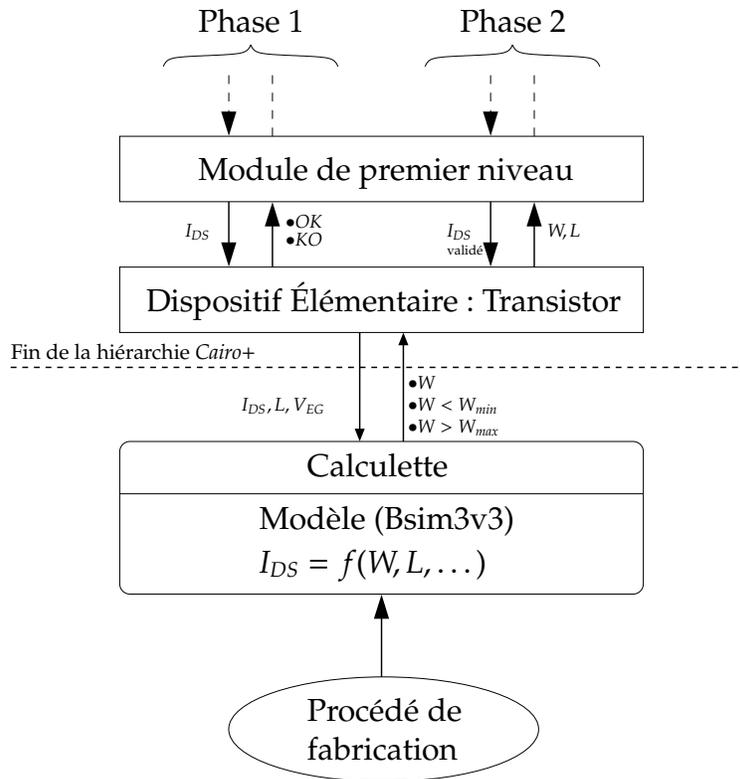


FIG. 4.5 : Dimensionnement électrique d'un transistor.

l'équation du modèle Bsim3v3, paramétrée par les informations relatives à la technologie, permet notamment le calcul du courant I_{DS} pour des dimensions W et L , et une polarisation données. En utilisant cette équation, la calculette peut indiquer la largeur W qui permet d'atteindre un I_{DS} désiré, pour une longueur L et une polarisation données. Cependant, elle peut également signaler que le calcul conduit à un W qui dépasse les bornes imposées par la technologie. En explorant des valeurs de L , le dispositif élémentaire peut ainsi indiquer au module de premier niveau :

- s'il est capable d'atteindre I_{DS} (Phase 1), et sinon, pourquoi.
- si oui, le couple $(W; L)$ qui conduit à I_{DS} (Phase 2).

Dans le cas du dimensionnement de composants passifs comme la capacité ou la résistance, la calculette permet notamment de s'assurer que les dimensions électriques C ou R sont physiquement réalisables.

3. Dessin des masques :

Le dimensionnement des dispositifs élémentaires conclut l'étape de dimensionnement global du circuit. Le placement ascendant des modules est alors initié par la génération des masques des dispositifs élémentaires. Il s'agit ici de générer la vue physique du composant en prenant en compte les règles de dessin imposées par la technologie. On procède pour cela à la traduction complète des dimensions électriques issues de l'étape précédente, en dimensions géométriques pour le dessin des masques. Cette traduction est qualifiée de complète pour

deux raisons :

- Elle aboutit à la géométrie de tous les rectangles en respectant les règles technologiques.
- Puisque l'on désire que, pour des dimensions électriques données, le dessin des masques puisse présenter plusieurs formes, cette traduction devra aboutir à plusieurs résultats. Une contrainte géométrique permettra finalement de choisir la forme qui s'intègre le mieux dans le dessin du circuit.

A la fin de cette étape, la vue structurelle du dispositif est mise à jour avec des informations concernant le dessin effectivement généré, notamment les dimensions effectives du ou des composants dessinés, ou encore les parasites introduits.

La génération du dessin des masques au niveau élémentaire est le point central de cette thèse et sera largement développée dans le chapitre 5.

La génération d'un dispositif élémentaire se fait par conséquent en trois étapes seulement. Elle se distingue ainsi de celle des modules qui se fait en quatre étapes. La figure 4.6 montre comment la communication est mise en œuvre entre les niveaux hiérarchiques pendant la génération d'un circuit.

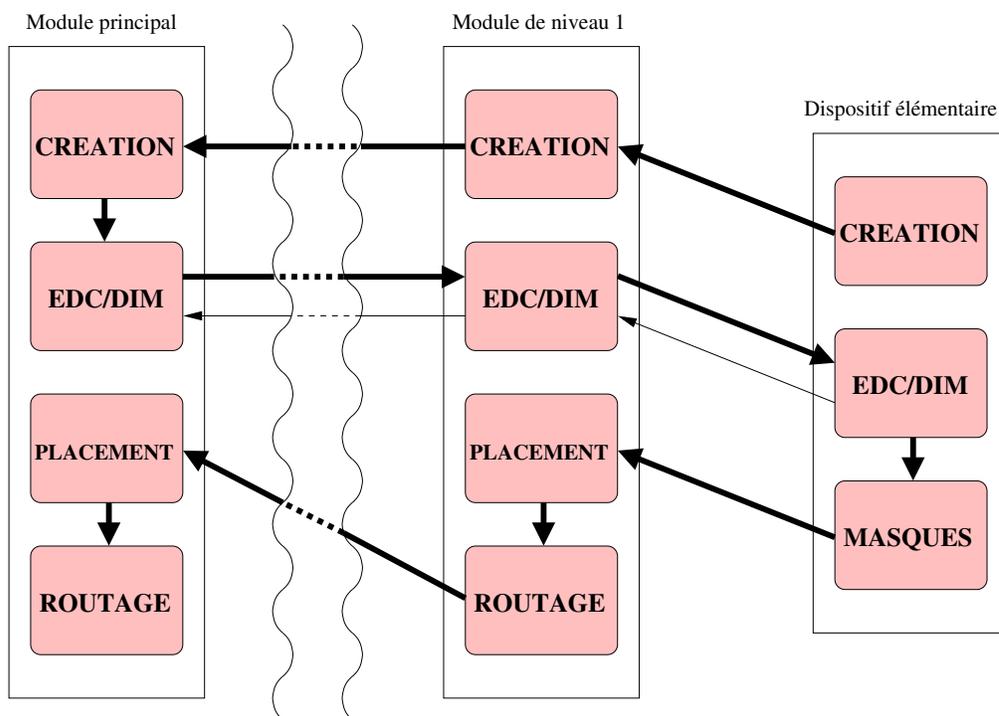


FIG. 4.6 : Etapes de la génération avec CAIRO+.

4.6 Conclusion

Nous avons présenté dans ce chapitre les différents acteurs de la génération d'un circuit avec CAIRO+ (dispositif élémentaire, module, calculette, etc...). Le rôle des dispositifs élémentaires

dans les différentes étapes de cette génération a été identifié. Dans le chapitre suivant, nous allons voir en particulier comment ces dispositifs élémentaires prennent en charge l'étape de dessin des masques.

Chapitre 5

Vers une bibliothèque de composants génériques

5.1 Introduction

Ce chapitre est le chapitre principal de cette thèse. Il détaille les solutions que nous proposons pour répondre aux contraintes liées à la réalisation des dispositifs élémentaires, énumérées au chapitre 2.

- Dans la section 5.2, nous verrons comment se fait un dessin des masques précis des composants analogiques que nous avons intégrés à *CAIRO+* : transistor MOS, capacité MIM et résistance polysilicium.
- Lorsque l'appariement entre différents composants est nécessaire, il faut utiliser des techniques de dessin particulières qui sont exposées dans la section 5.3.
- Par ailleurs, la généricité des dispositifs élémentaires de *CAIRO+* implique que les dimensions géométriques des dessins obtenus puissent connaître de grandes variations. Pour faire face à ces variations, la génération doit conduire à plusieurs dessins présentant des facteurs de forme différents : on parle alors de dispositifs élémentaires déformables. Nous verrons dans la section 5.4 comment les générateurs de dispositifs élémentaires articulent cette déformation avec la précision requise pour chaque dessin.
- Dans la section 5.5, nous verrons comment le dessin des masques obtenu est caractérisé. Cette caractérisation est essentielle pour que l'estimation des performances du circuit prenne en compte les parasites introduits par la réalisation.
- La section 5.6 présente l'architecture logicielle des générateurs de dispositifs élémentaires. En établissant une frontière forte avec le procédé de fabrication, cette architecture rend possible l'indépendance technologique.
- Enfin, la section 5.7 décrit la procédure de validation d'un générateur de dispositifs élémentaires.

5.2 Précision : dessin des masques optimisé

Dans le domaine analogique, il est capital d'obtenir une réalisation physique la plus précise possible. La précision est toutefois limitée par une grille sur laquelle doivent s'aligner les masques (fig. 5.1). Le pas δ de cette grille est fourni par le fondeur qui impose que les dimensions et les coordonnées de tous les rectangles constituant le dessin des masques soient multiples de δ . Les dispositifs élémentaires de CAIRO+ devront pouvoir générer un dessin des masques à une précision égale à δ .

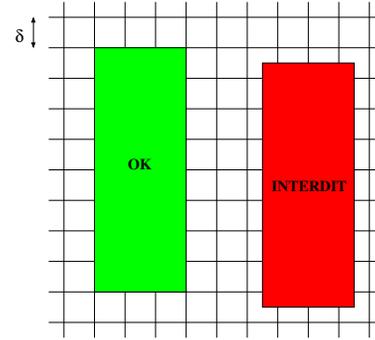


FIG. 5.1 : Grille de dessin imposée par le fondeur.

D'un autre côté, le style même du dessin a une forte influence sur le comportement électrique du circuit. En effet, selon le style de son dessin des masques, un composant sera plus ou moins sensible aux imperfections de fabrication, ou sera soumis à des parasites plus ou moins importants. Dans les sections suivantes, nous allons voir le style généralement adopté pour le dessin des composants analogiques classiques.

5.2.1 Dessin des masques d'un transistor MOS

En analogique, les dimensions des transistors d'un même circuit peuvent varier d'un facteur 100. Or, pour assurer des propriétés physiques homogènes à tous ces transistors, il est important que le dessin du circuit soit compact. Pour chaque transistor, il est donc nécessaire d'envisager plusieurs types de dessin, de formes différentes, afin de tous les représenter sur une surface raisonnable. Pour cela, on considère qu'un transistor de largeur W et de longueur L peut être représenté comme un ensemble de M sous-transistors, ou **doigts**, montés en parallèle. Les dimensions de ces sous-transistors sont alors $W_f = \frac{W}{M}$ et $L_f = L$, où W_f et L_f sont respectivement la largeur et la longueur d'un sous-transistor. Dans ce cas, on dit que l'on a replié le transistor, et le **nombre de repliements** M est le nombre de doigts formant le transistor. Comme le montre la figure 5.2, on peut obtenir différentes géométries en faisant varier M .

Cependant, les caractéristiques électriques du transistor varient légèrement avec la géométrie choisie. Par exemple, les capacités de jonction liées aux diffusions de source et de drain peuvent être exprimées ainsi [Razavi01] :

$$C = A \times C_j + P \times C_{jsw} \quad (5.1)$$

où A est la surface de la diffusion considérée, P son périmètre, C_j est la capacité par unité de surface, et C_{jsw} est la capacité par unité de longueur. Sur la figure 5.2, on remarque que le repliement d'un transistor entraîne un partage de diffusion entre deux doigts consécutifs. Essayons d'évaluer l'incidence de ce partage sur les capacités de jonction.

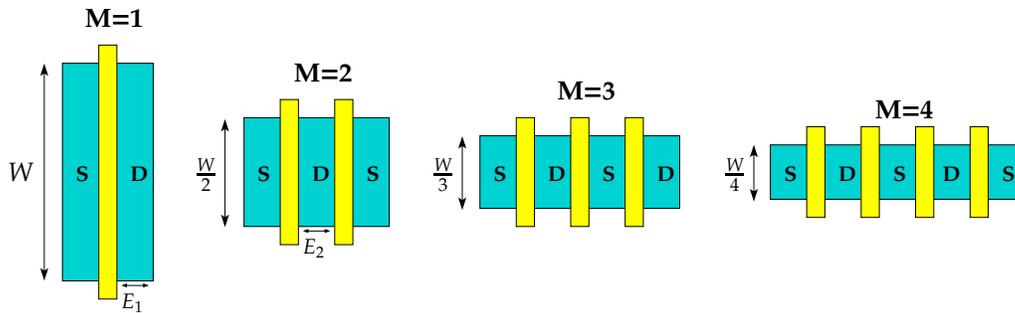


FIG. 5.2 : Repliement d'un transistor et partage de la diffusion.

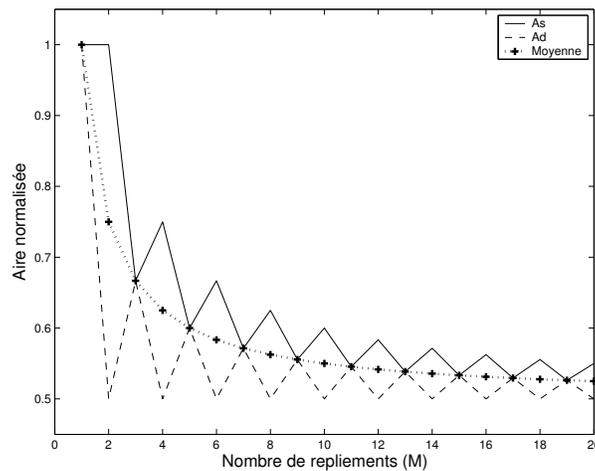
Sur la figure 5.2, on a choisi de commencer systématiquement le dessin du transistor par une diffusion de source. Il est bien sûr possible de commencer par un drain. Par ailleurs, nous avons appelé E_1 l'extension de diffusion autour de la grille, et E_2 la distance entre deux grilles consécutives. Nous considérerons ici que ces valeurs sont égales : $E_1 = E_2 = E$, même si en pratique, E_1 est généralement légèrement différent de E_2 . Sous ces conditions, on peut établir qu'en faisant varier le nombre de repliements M , on obtient :

$$As = \left\lfloor \frac{M+2}{2} \right\rfloor \times \left(\frac{W}{M} \cdot E \right) = \frac{\lfloor \frac{M+2}{2} \rfloor}{M} \times As_1 \quad (5.2)$$

$$Ad = \left\lfloor \frac{M+1}{2} \right\rfloor \times \left(\frac{W}{M} \cdot E \right) = \frac{\lfloor \frac{M+1}{2} \rfloor}{M} \times Ad_1$$

où $\lfloor x \rfloor$ est la partie entière de x .

Ici, As est l'aire de la source, et Ad celle du drain. As_1 et Ad_1 sont respectivement les aires de source et de drain pour $M=1$, et donc $As_1 = Ad_1 = W \times E$. La surface du drain ou de la source peut donc être exprimée comme un facteur de la surface initiale. La figure 5.3 montre l'évolution de ce facteur en fonction de M . On remarque qu'en commençant par une diffusion de source, l'aire de la source

FIG. 5.3 : Evolution de l'aire des diffusions en fonction de M .

est minimisée pour M impair, et celle du drain pour M pair. Cependant, la courbe moyenne montre que le repliement d'un transistor permet de réduire globalement l'aire des zones de diffusion tant que la largeur des doigts respecte la largeur minimale W_{min} imposée par la technologie. On peut également noter qu'en commençant par une source, on a deux propriétés importantes :

$$\begin{aligned} 1. \forall M, A_s &\geq A_d \\ 2. \forall M \text{ pair}, A_d &= A_{d_{min}} = \frac{1}{2}A_{d_1} \end{aligned} \quad (5.3)$$

Ces résultats sont particulièrement intéressants puisque dans de nombreux circuits, le nœud de sortie est connecté au drain d'un transistor. En commençant le repliement par une diffusion de source, on peut ainsi réduire les parasites sur le nœud sortie. De plus, la surface du drain, et donc la capacité de jonction sur cette diffusion, sera minimisée pour les nombres de repliements pairs.

Voyons maintenant comment évolue le périmètre des zones de diffusion lorsque M varie. Là encore, on peut exprimer les périmètres de source et de drain comme un facteur du périmètre initial P_1 avec $P_1 = P_{s_1} = P_{d_1} = 2.(W + E)$:

$$P_s = \left\lfloor \frac{M+2}{2} \right\rfloor \times 2. \left(\frac{W}{M} + E \right) = \left\lfloor \frac{M+2}{2} \right\rfloor \times \frac{\frac{1}{M} \cdot \frac{W}{E} + 1}{\frac{W}{E} + 1} \times P_{s_1} \quad (5.4)$$

$$P_d = \left\lfloor \frac{M+1}{2} \right\rfloor \times 2. \left(\frac{W}{M} + E \right) = \left\lfloor \frac{M+1}{2} \right\rfloor \times \frac{\frac{1}{M} \cdot \frac{W}{E} + 1}{\frac{W}{E} + 1} \times P_{d_1}$$

Il apparaît sur ces équations que ce facteur dépend du rapport W/E . Dans un premier temps, et dans le but d'observer la tendance de l'évolution du périmètre, considérons un cas typique pour lequel :

- $W/E = 10$.
- $W_{min} = E$, et donc $M_{max} = W/W_{min} = 10$.

Sous ces conditions, le facteur de périmètre évolue selon la courbe de la figure 5.4. Ici encore, selon que l'on considère la source ou le drain, la parité de M est importante. Mais surtout, on a :

$$\forall M, P_s \geq P_d \quad (5.5)$$

Par ailleurs, le périmètre moyen devient croissant à partir d'un M relativement petit. Effectivement, en dérivant les équations 5.4, on obtient que le périmètre minimal est obtenu pour :

$$M_{p_{min}} = \sqrt{\frac{W}{E}} \quad (5.6)$$

Ce résultat est confirmé par la figure 5.5 qui montre l'évolution du périmètre moyen pour différents rapports W/E . Heureusement, et dans la mesure où l'on a $W_{min} = E$, le périmètre moyen P_m est tel que :

$$\forall M \leq M_{max}, P_m \leq P_1 \quad (5.7)$$

Or, en pratique, on peut effectivement considérer que $W_{min} \simeq E$. Ainsi, le repliement d'un transistor ne peut que réduire le périmètre de la diffusion.

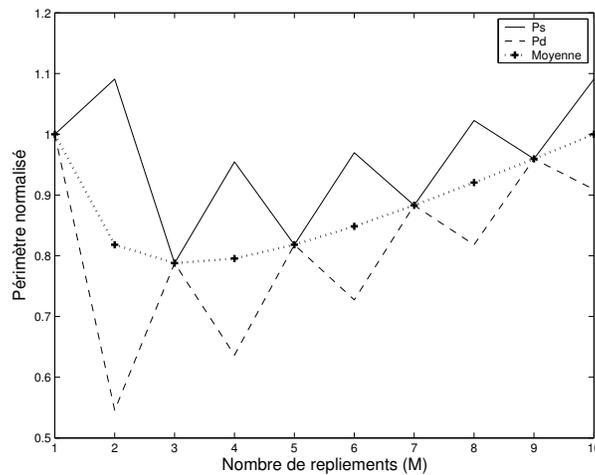


FIG. 5.4 : Evolution du périmètre des diffusions en fonction de M pour $W/E=10$.

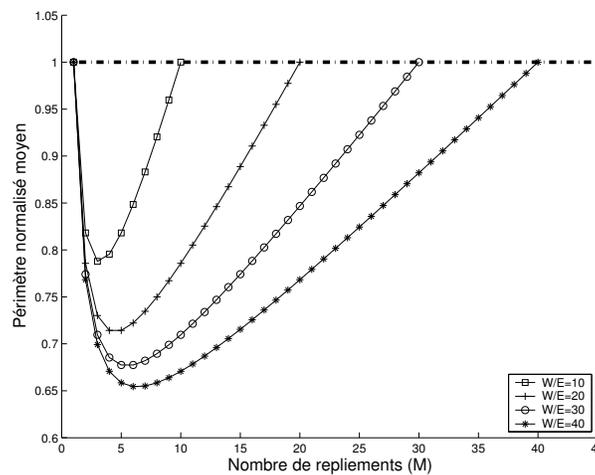


FIG. 5.5 : Périmètres moyens pour différents rapports W/E .

Pour analyser ces résultats, rappelons que dans le calcul de la capacité de jonction donné par l'équation 5.1, les paramètres C_j et C_{jsw} pondèrent respectivement l'influence de l'aire et du périmètre de la diffusion. Pour les technologies actuelles, ces paramètres sont tels que $C_j \gg C_{jsw}$. Dans ces conditions, la capacité de jonction moyenne est principalement liée à l'aire de la diffusion et sera donc minimisée par le repliement. Avec les technologies émergentes, la tendance est cependant à la réduction du rapport C_j/C_{jsw} . L'importance du périmètre des diffusions est donc grandissante. Malgré tout, nous avons vu que si le périmètre des diffusions ne diminue pas toujours avec le nombre de pliages, il reste inférieur au périmètre initial. Enfin, on a vu que la diffusion par laquelle on commence le dessin du transistor est pénalisée en termes d'aire et de périmètre, et donc de capacités parasites. Ainsi, un transistor replié n'est pas symétrique et il faut

choisir avec attention le nœud que l'on connecte à cette diffusion.

Un autre effet important du repliement d'un transistor est la sensibilité à l'effet de canal mince [Cheng99] : en observant la coupe d'un transistor MOS dans le sens de la largeur (fig. 5.6-a), on remarque qu'aux extrémités de la grille, l'oxyde devient plus épais, modifiant ainsi localement l'action du potentiel de grille sur la zone de déplétion. La tension de seuil V_{TH} du transistor est alors altérée. Ce phénomène est d'autant plus visible que la largeur du transistor est petite (canal mince). Or, lorsque l'on replie le transistor, et puisque $W_f = \frac{W}{M}$, on diminue la largeur des doigts, les rendant de plus en plus sensibles à cet effet. Ainsi, le courant de drain d'un transistor replié est légèrement différent de celui d'un transistor droit :

$$I_{DS}(W,L) \neq M \times I_{DS}\left(\frac{W}{M}, L\right) \quad (5.8)$$

En repliant un transistor, on réduit également la résistance de grille. En effet, puisque l'on met en parallèle des sous-transistors de largeur $W_f \leq W$, elle diminue lorsque M augmente (fig. 5.6-b).

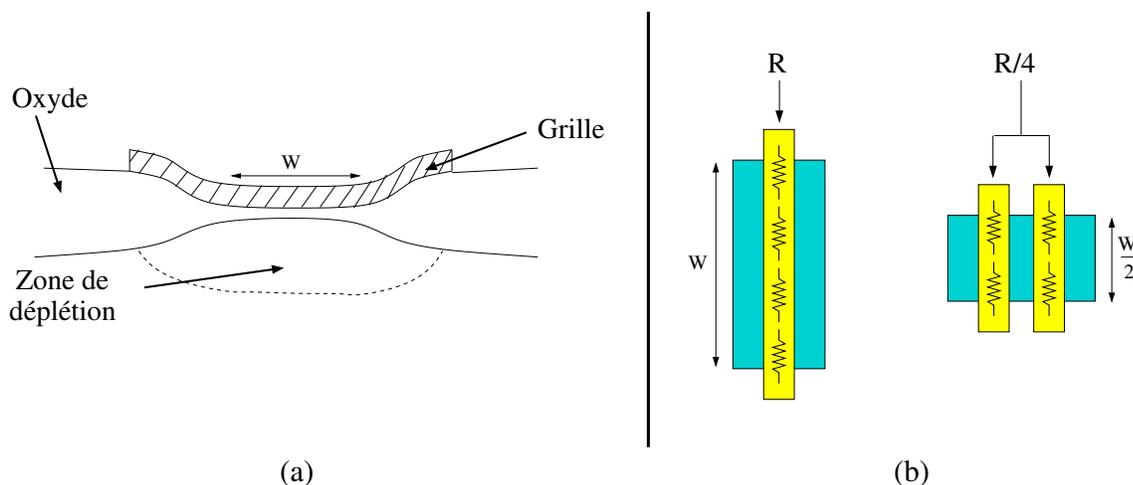


FIG. 5.6 : D'autres effets du repliement : (a) Effet des transistors minces - (b) Réduction de la résistance de grille.

Pour conclure sur le repliement, on voit qu'il est nécessaire de bien caractériser la géométrie du dessin effectué pour que la calculette puisse modéliser précisément le comportement d'un transistor. Nous verrons dans la section 5.5 comment la géométrie est caractérisée, et comment elle est prise en compte.

Pour finir, le dessin précis d'un transistor doit aussi prendre garde à multiplier les contacts assurant la connexion des zones de diffusion 5.7. La mise en parallèle de plusieurs contacts minimise en effet la résistance d'accès aux diffusions et rend le dessin plus robuste vis-à-vis des

éventuelles erreurs de fabrication qui pourraient intervenir au niveau des contacts. La correction est mieux assurée.

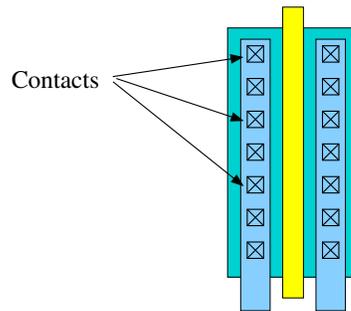


FIG. 5.7 : Réduction de la résistance d'accès aux diffusions.

5.2.2 Dessin des masques d'une capacité Métal-Isolant-Métal (MIM)

Nous avons intégré dans *CAIRO+* des générateurs de capacités MIM. Une capacité MIM est constituée de deux armatures métalliques séparées par une couche d'oxyde isolante. Si on peut lui reprocher une surface importante, elle a deux avantages :

- Elle présente une bonne linéarité : sa valeur dépend peu de la tension à ses bornes, ou de la température.
- Elle est facile à réaliser.

Aujourd'hui, la plupart des technologies proposent des métaux dédiés au dessin de capacités MIM. La figure 5.8-a représente le dessin des masques d'une capacité MIM. On remarque que l'armature

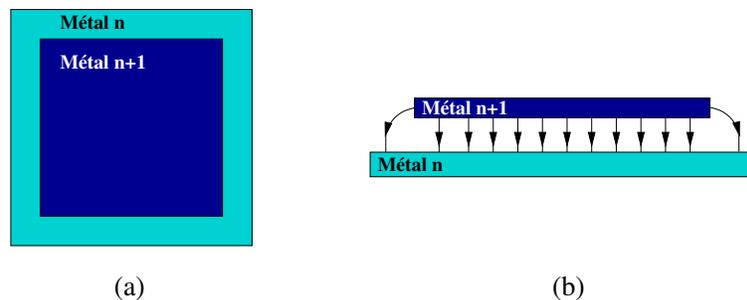


FIG. 5.8 : (a) Dessin des masques d'une capacité MIM - (b) Champ électrique entre les armatures.

basse est plus grande que l'armature haute. Ceci permet d'assurer la superposition des armatures malgré les erreurs de fabrication qui peuvent modifier leur position relative. Le champ électrique existant alors entre ces armatures est représenté sur la figure 5.8-b. Il est composé principalement de lignes de champ perpendiculaires aux armatures sur toute la surface de superposition. Mais il existe également des lignes de champ entre les bords de l'armature haute et l'armature basse. La valeur effective de la capacité fabriquée dépend donc de l'aire et du périmètre de l'armature

haute :

$$C = A \times C_a + P \times C_p \quad (5.9)$$

où A et P sont respectivement l'aire et le périmètre de l'armature haute, C_a et C_p sont les capacités surfacique et périmétrique données par le fondeur.

La précision des capacités MIM est dégradée par les erreurs de gravure lors de la fabrication. En effet, les armatures sont faites par photolithographie. Cette technique consiste à déposer, sur toute la surface de la tranche (*wafer*), une couche de métal sur laquelle on dépose à nouveau une couche photo-résistante. Après une exposition des armatures aux rayons ultra-violet, la zone photo-résistante qui les recouvre polymérise. Elle résiste alors localement au solvant organique utilisé pour retirer le reste de cette couche. On retire finalement le métal partout où il n'est plus recouvert par la couche photo-résistante (*etching*). Malheureusement, il est fréquent qu'une partie du métal présent sous la couche photo-résistante soit aussi retirée (*over-etching*, fig. 5.9).

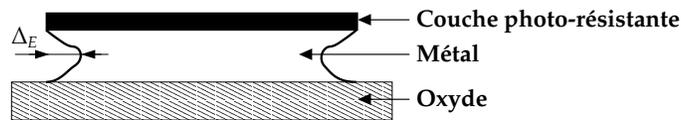


FIG. 5.9 : Excès de gravure (*over-etching*).

Ce phénomène réduit les dimensions effectives des armatures. Pour un excès de gravure Δ_E , la longueur et la largeur de l'armature haute sont réduites de $2\Delta_E$. Ceci conduit à une erreur relative pour la capacité donnée par [Johns97] :

$$\frac{\Delta C}{C} \simeq \frac{-2\Delta_E P}{A} \quad (5.10)$$

où P et A sont le périmètre et l'aire de l'armature haute. Elle est donc proportionnelle au rapport P/A . C'est pourquoi, dans la mesure du possible, on dessinera des capacités MIM carrées [Maloberti94]. C'est en effet la forme qui minimise le rapport P/A pour une valeur de capacité donnée.

Il est cependant rare d'utiliser une capacité unique et de nombreux circuits analogiques s'appuient plutôt sur des rapports capacitifs précis. Plusieurs contraintes doivent alors être appliquées au dessin pour assurer la précision nécessaire. Nous verrons à la section 5.3 quelles sont ces contraintes.

5.2.3 Dessin des masques d'une résistance polysilicium

Une résistance intégrée est faite d'une fine bande de matériau résistif. Les technologies classiques proposent notamment la résistance en polysilicium. Le polysilicium présente une bonne résistivité et est assez stable en température. Dans *CAIRO+*, nous avons donc conçu deux générateurs de résistances polysilicium.

Grossièrement, pour réaliser une telle résistance il suffit de placer deux contacts ohmiques aux extrémités de la bande résistive (fig.5.10-a). Si l'on néglige la résistance des contacts, et pour une

bande de largeur W , de longueur L , et d'épaisseur t , on obtient une résistance de valeur

$$R = \rho \frac{L}{S} = \rho \frac{L}{W.t} = R_{\square} \frac{L}{W} \quad (5.11)$$

où R_{\square} est la résistance par carré valant ρ/t , généralement donnée par le fondeur. Les bandes de polysilicium sont également fabriquées par photolithographie. Pour minimiser l'effet relatif de l'excès de gravure, le fondeur suggère souvent une largeur W assez grande pour la réalisation des résistances. En conséquence, et comme la résistivité des matériaux intégrés est relativement faible (typiquement, $R_{\square} = 1 \text{ k}\Omega/\square$), et il est souvent nécessaire de dessiner de longues résistances en serpentín (fig.5.10-b). On introduit alors un certain nombre d'angles droits dans le dessin. La

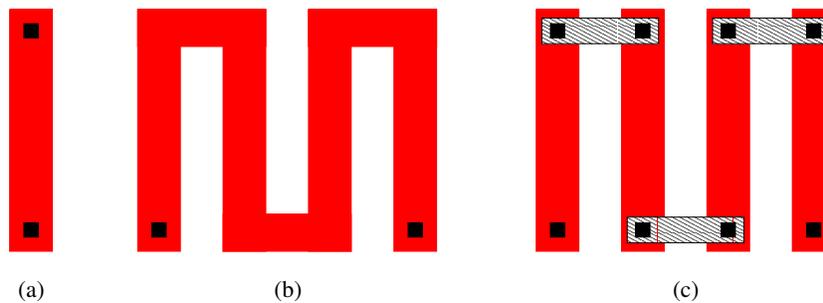


FIG. 5.10 : Dessin des masques d'une résistance. (a) Résistance simple (b) Résistance en serpentín (c) Serpentín à jonctions métalliques.

résistance des carrés au niveau de ces angles droits est mal définie. On peut l'estimer à celle d'un demi-carré. Pour une plus grande précision, on utilise en général des fils métalliques pour relier plusieurs bandes résistives (fig.5.10-c). On peut négliger la résistance de ces fils mais il faut en revanche prendre en compte la résistance des contacts supplémentaires.

A l'instar des capacités, les résistances sont très souvent utilisées comme des éléments appariés. Nous allons voir dans la section suivante qu'un dessin approprié permet d'assurer un bon appariement.

5.3 Appariement : techniques de dessin dédiées

L'une des principales caractéristiques de la conception de circuits analogiques est la fréquente nécessité d'appariement entre les composants [Pelgrom98]. Certains dispositifs extrêmement répandus comme la paire différentielle ou le miroir de courant se basent sur la précision du rapport entre les dimensions des transistors qui les composent. De même, le fonctionnement d'un intégrateur à capacités commutées dépendra fortement du rapport qui lie les capacités impliquées [Khalil05]. D'autres circuits peuvent également s'appuyer sur un rapport résistif précis. C'est le cas des convertisseurs analogique-numérique (CAN/ADC) basés sur une échelle de résistances [Pelgrom94]. Dans cette section, nous allons voir que les imperfections de fabrication nuisent à l'appariement des composants. En analysant les causes de mauvais appariement, nous verrons qu'un dessin des masques approprié permet de limiter leurs effets.

5.3.1 Deux classes d'erreurs d'appariement

Lors de la fabrication, plusieurs phénomènes nuisent à la qualité de l'appariement entre composants. On peut définir l'erreur relative d'appariement par :

$$\delta = \frac{R_m - R_v}{R_v} \quad (5.12)$$

où R_m est le rapport mesuré après fabrication, et R_v est le rapport initialement visé. Ce rapport peut concerner les tensions de seuil de deux transistors, les valeurs de deux capacités ou de deux résistances. Pour pleinement quantifier l'erreur d'appariement, il faut cependant effectuer les mesures sur un échantillon suffisamment représentatif de circuits. Si l'on effectue des mesures sur un échantillon de N unités, on observera les erreurs $\delta_1, \delta_2, \dots, \delta_N$. On peut alors déterminer la moyenne m_δ , et l'écart type σ_δ des résultats obtenus :

$$m_\delta = \frac{1}{N} \sum_{i=1}^N \delta_i \quad (5.13)$$

$$\sigma_\delta = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\delta_i - m_\delta)^2} \quad (5.14)$$

La figure 5.11, qui représente l'histogramme des erreurs d'appariement obtenues pour 30 mesures, illustre les grandeurs m_δ et σ_δ .

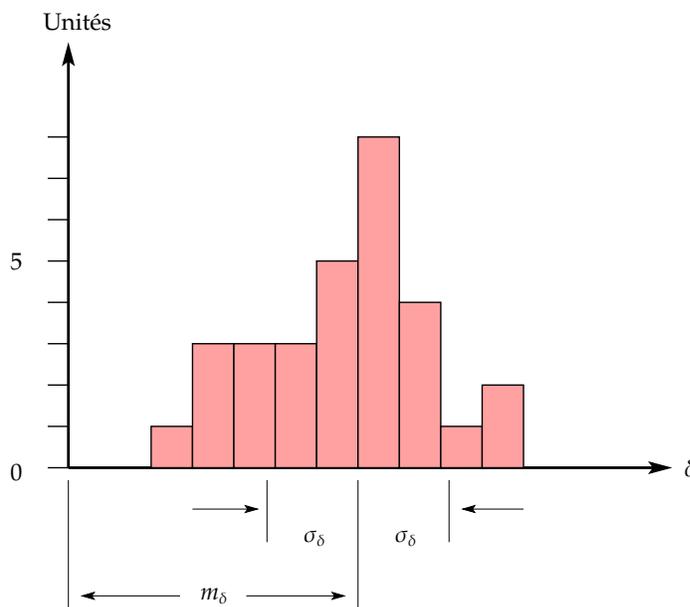


FIG. 5.11 : Histogramme du mismatch pour un échantillon de 30 unités.

On distingue alors deux types d'erreurs d'appariement [Hastings01] :

Erreurs systématiques : Elles sont mesurées par la grandeur m_δ et proviennent de phénomènes qui affectent toutes les unités de la même façon. On trouvera notamment dans cette catégorie

les erreurs d'appariement dues à l'excès de gravure. On peut également parler ici des erreurs issues d'une réalisation naïve. Par exemple, le routage qui dessert deux composants appariés doit répartir équitablement les parasites qu'il introduit. La plupart du temps, il est possible d'annuler les erreurs systématiques d'appariement en portant une attention particulière à la réalisation physique.

Erreurs aléatoires : Elles sont mesurées par la grandeur σ_δ et sont principalement dues à des variations statistiques sur les conditions de fabrication, ou sur les propriétés des matériaux. Bien qu'on ne puisse s'affranchir de ces variations lors de la fabrication, il est souvent possible d'en minimiser l'impact sur l'appariement.

5.3.2 Appariement de transistors

5.3.2.1 Gradient d'oxyde

L'une des principales erreurs systématiques d'appariement est due à la variation de l'épaisseur d'oxyde le long du wafer. Au premier ordre, on peut modéliser cette variation par une pente. Ainsi, si deux transistors de mêmes dimensions sont distants de d , on observera une erreur d'appariement proportionnelle à d . Comme le montre la figure 5.12-a, celle-ci provient de la différence d'épaisseur d'oxyde sous leur grille respective.

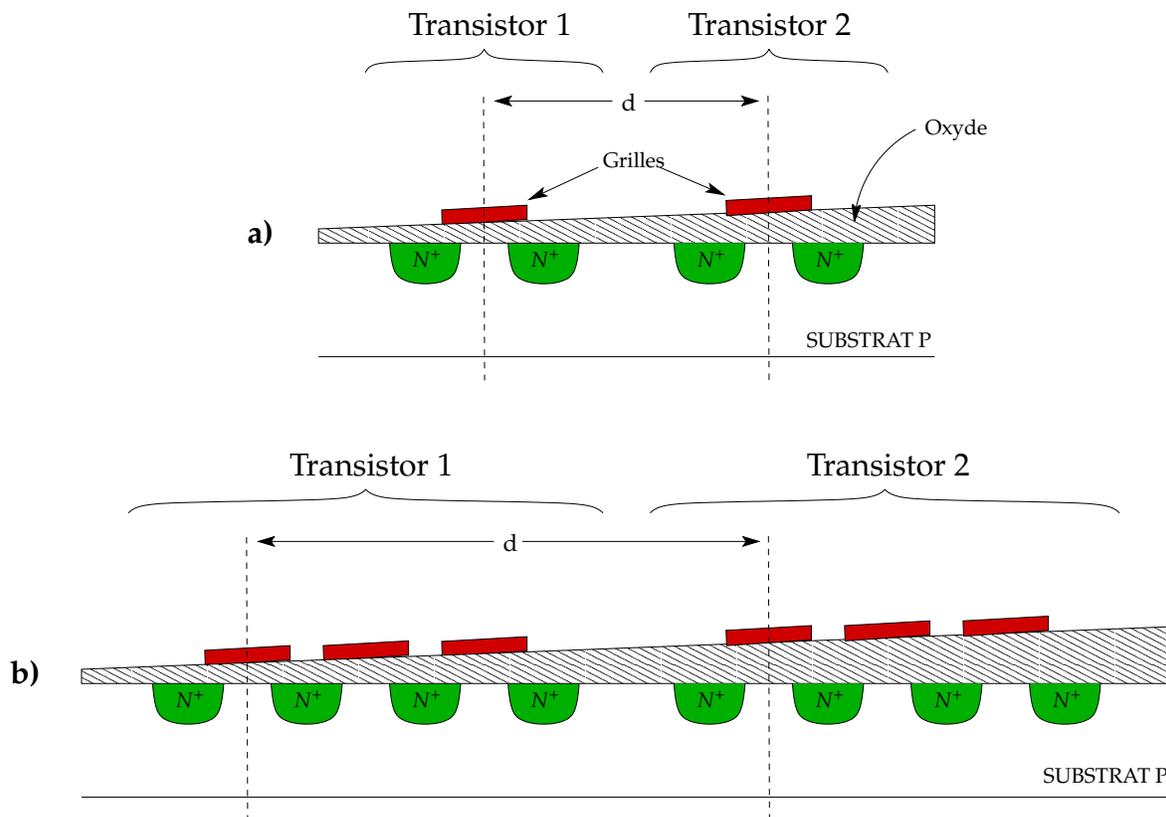


FIG. 5.12 : Variation de l'épaisseur d'oxyde.

Si l'on replie les deux transistors (fig. 5.12-b), d augmente, et l'erreur d'appariement devient plus importante. On peut cependant essayer de réduire d en mélangeant les transistors. Une première technique, appelée inter-digitation, consiste à alterner les doigts de chaque transistor dans un même empilage (fig. 5.13).

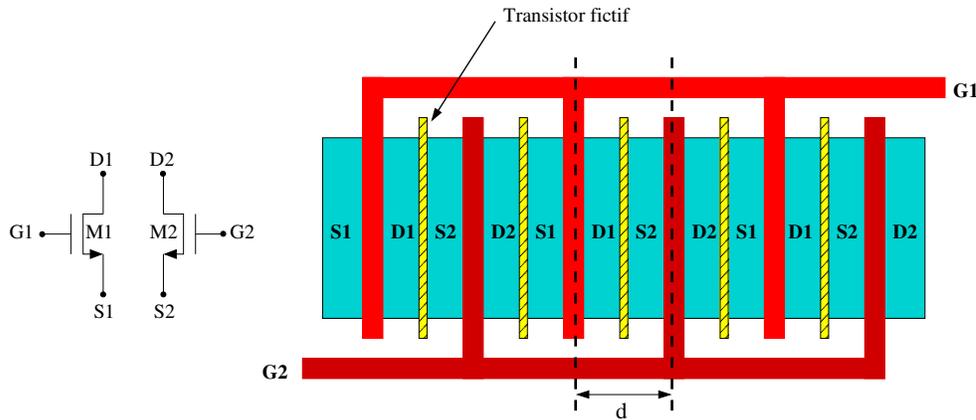


FIG. 5.13 : Inter-digitation de transistors pour lesquels $M=3$.

Lorsque, comme ici, les transistors inter-digités n'ont pas de diffusion commune, on insère des transistors fictifs (*dummy transistors*) qui assurent l'isolation électrique au niveau des diffusions. Il s'agit de transistors maintenus en régime bloqué souvent dessinés à longueur minimale. Cependant, il est très fréquent que des transistors appariés aient la source en commun. C'est le cas dans le miroir de courant ou la paire différentielle dont nous avons déjà parlé. On peut alors isoler le drain entre deux grilles d'un même transistor et inter-digiter des paires de grilles. Il n'est alors plus nécessaire d'introduire des transistors fictifs puisque la diffusion de source commune peut être partagée entre les paires de grilles des différents transistors (fig. 5.17).

Dans certains cas, on peut rendre le dessin insensible au gradient d'oxyde en adoptant une géométrie centrée [Bastos96]. Cette technique consiste à superposer les barycentres des transistors afin que le gradient ait sur eux un impact homogène. Pour réaliser cela, il faut faire en sorte que le dessin de chaque transistor présente un axe de symétrie, puis superposer les axes de symétrie des transistors à appairer. Par exemple, si l'on considère les deux transistors de la figure 5.14, on remarque que l'épaisseur moyenne de l'oxyde pour chacun de ces transistors est :

$$t_{ox1}^- = t_{ox2}^- = t_{ox} + \frac{7}{2} \Delta t_{ox} \quad (5.15)$$

Malheureusement, selon le nombre de transistors à appairer et les rapports qui les lient, il n'est pas toujours possible d'obtenir une géométrie parfaitement centrée. De plus, une telle géométrie est souvent plus facile à réaliser avec un nombre de repliements pair, ce qui ajoute une nouvelle contrainte de conception.

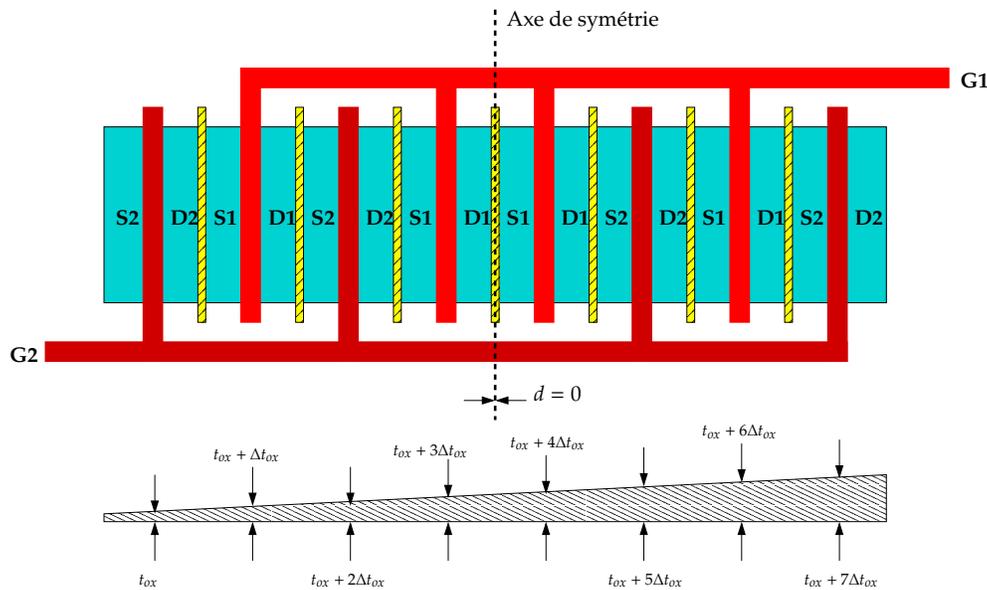


FIG. 5.14 : Dessin à géométrie centrée pour deux transistors à $M=4$.

5.3.2.2 Excès de gravure

Une autre importante erreur systématique d'appariement est due à l'excès de gravure que nous avons présenté à la section 5.2. Effectivement, la grille des transistors en polysilicium, fabriquée par photolithographie, est soumise à un excès de gravure qui dépend de son environnement. Ainsi, dans le cas d'un transistor replié, le premier et le dernier doigt n'ont pas exactement les mêmes propriétés électriques que les autres (fig. 5.15-a).

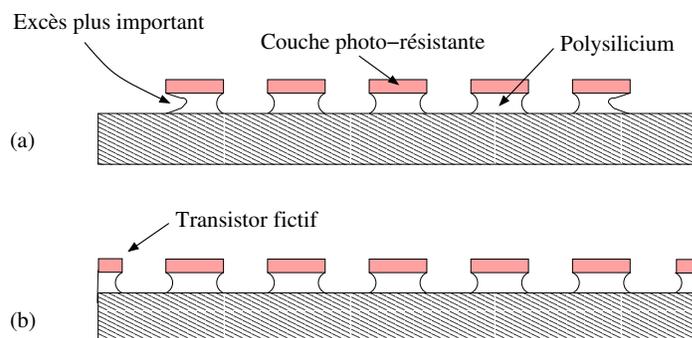


FIG. 5.15 : (a) Excès de gravure pour les doigts extrêmes - (b) Insertion de transistors fictifs.

Si l'on reprend l'exemple des deux transistors appariés de la figure 5.14, seul le transistor $M2$ est touché par ce problème. L'appariement est alors significativement réduit. Pour contrer cela, on va chercher à donner le même environnement à tous les doigts de tous les transistors en introduisant des transistors fictifs aux extrémités de l'empilage (fig. 5.15-b). Ainsi, tous les doigts sont dans un environnement identique et sont soumis à un excès de gravure comparable.

5.3.2.3 Effet d'ombre (*Shadowing*)

Le dopage des zones actives du transistor est effectué par implantation ionique. Il s'agit d'un rayonnement d'ions qui, pour diverses raisons, attaquent la surface selon un angle d'environ 7° . En dopant la zone active d'un transistor, on crée une asymétrie entre le drain et la source issue de la zone protégée du rayonnement par le relief de la grille. On appelle ce phénomène l'effet d'ombre (fig. 5.16). Il s'agit évidemment d'une source systématique de mauvais appariement.

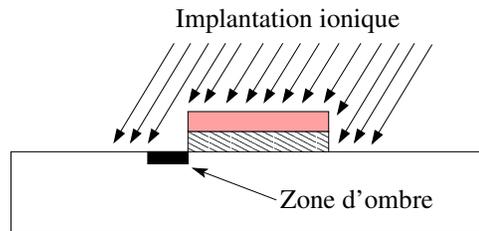


FIG. 5.16 : Effet d'ombre.

Pour obtenir un bon appariement, il faut faire en sorte que cette asymétrie soit la même pour tous les transistors concernés. La figure 5.17 propose différents dessins des masques d'une paire différentielle, composée de deux transistors à source commune exactement identiques. Le dessin

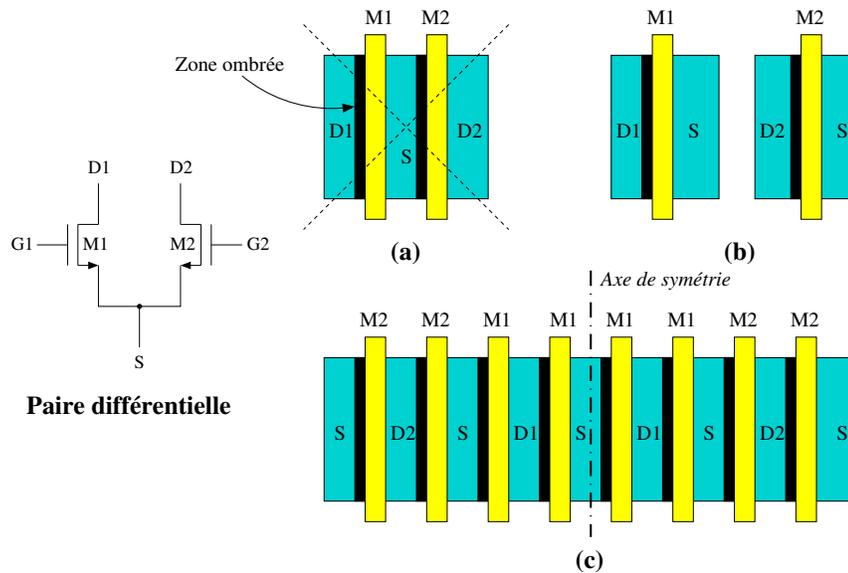


FIG. 5.17 : Transistors appariés vis-à-vis de l'effet d'ombre.

(a) est particulièrement naïf, notamment à cause de sa sensibilité à l'effet d'ombre. En effet, seul le drain du transistor M1 est soumis à ce problème. Le dessin (b) résout l'erreur d'appariement liées à l'effet d'ombre, mais augmente la distance entre les deux transistors, les rendant plus sensibles au gradient d'oxyde. Le dessin (c) illustre le regroupement par paire de grille dont nous avons déjà parlé. Ici, on a une géométrie centrée insensible au gradient d'oxyde. De plus, les drains étant

toujours enfermés entre deux grilles, ils sont soumis au même effet d'ombre, et celui-ci n'introduit donc plus d'erreur d'appariement. La source est commune et ne pose pas de problème en termes d'appariement.

5.3.2.4 Orientation du courant

Du fait de l'anisotropie du cristal de silicium, l'intensité du courant dans un transistor dépend de sa direction. En conséquence, les courants qui traversent des transistors appariés doivent être orientés dans la même direction [Pelgrom89]. Pour optimiser l'appariement, on a alors deux possibilités [Malavasi95] :

- Faire en sorte que tous les doigts de tous les transistors soient traversés par des courants allant dans la même direction. C'est le cas des transistors de la figure 5.14.
- Donner à tous les transistors un nombre de repliements pair, puis faire en sorte que pour chaque transistor, la moitié des doigts soit traversée par des courants allant dans une direction, et l'autre moitié soit traversée par des courants allant dans la direction opposée. Les transistors de la figure 5.17-c respectent cette contrainte.

Il est parfois nécessaire d'introduire des transistors fictifs pour assurer le respect d'une de ces règles.

D'autres caractéristiques influent sur la sensibilité du dessin aux erreurs d'appariement. En particulier, l'importance des erreurs aléatoires dépend beaucoup des dimensions géométriques des grilles de chaque transistor [Pelgrom89]. Ces dimensions sont définies par la largeur W , la longueur L , et le nombre de repliements M . Nous verrons que ces grandeurs sont les entrées des procédures de génération des masques des transistors. Un générateur de dispositif élémentaire ne peut donc pas influencer sur ces paramètres.

5.3.3 Appariement de capacités

L'appariement entre capacités est principalement dégradé par deux phénomènes : l'excès de gravure et le gradient d'oxyde [Johns97]. Comme l'indique l'équation 5.10, des capacités de géométrie identique, c'est-à-dire de rapport périmètre/surface identique, sont moins affectées par les erreurs d'appariement dues à l'excès de gravure. Cette équation n'est cependant qu'une approximation. Pour effectivement optimiser l'appariement, il faut que les capacités présentent exactement la même géométrie. Quel que soit le rapport visé, ceci est possible en divisant chaque capacité en plusieurs capacités unitaires carrées connectées en parallèle. Ainsi, toutes les capacités unitaires ont la même géométrie, et la forme carrée minimise l'effet de l'excès de gravure. La figure 5.18 montre le dessin des masques de deux capacités C_1 et C_2 liées par un rapport $C_1/C_2 = 4/5$. La présence de capacités fictives (notées D) et l'arrangement en matrice régulière permettent de maintenir un excès de gravure similaire pour toutes les capacités unitaires.

Ce type de dessin ne permet d'obtenir, en revanche, que des rapports rationnels. De plus, certains rapports, même rationnels, conduisent à des capacités unitaires trop petites, qui ne respectent

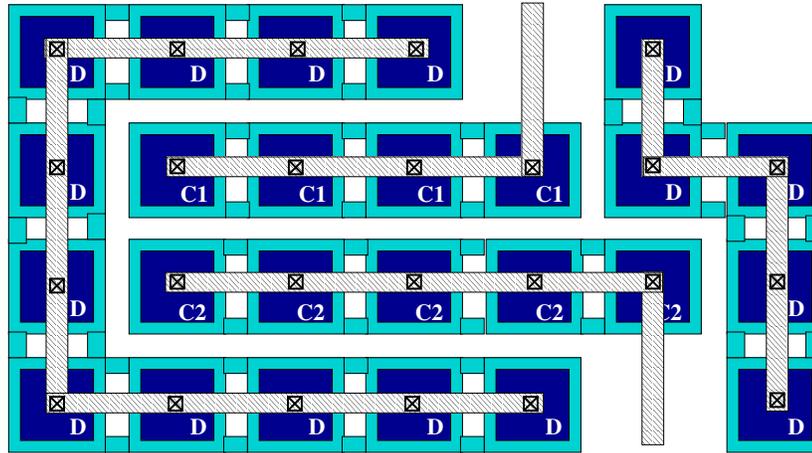


FIG. 5.18 : Deux capacités de rapport $C1/C2 = 4/5$ appariées vis-à-vis de l'excès de gravure.

pas les dimensions minimales de la technologie. Dans ce cas, ou pour obtenir des rapports non rationnels, on utilise une capacité non-unitaire. Il s'agit d'une capacité dont la valeur C_{NU} doit être telle que :

$$C_{NU} \in [C_U ; 2.C_U] \quad (5.16)$$

où C_U est la valeur de la capacité unitaire. Un rapport $C1/C2 = 5.4/3$ sera alors obtenu en représentant la capacité $C2$ par trois capacités unitaires, et la capacité $C1$ par quatre capacités unitaires, plus une capacité non-unitaire de valeur $C_{NU} = 1.4C_U$. Afin de maintenir la régularité de la matrice, il faut que la géométrie de la capacité non-unitaire s'inscrive dans celle de deux capacités unitaires adjacentes. Par ailleurs, et toujours dans le but de maximiser l'appariement, la capacité non-unitaire doit présenter un rapport périmètre/surface égal à celui des capacités unitaires. Dès lors, trois techniques sont généralement employées (fig. 5.19) :

- **Rectangle (fig. 5.19-a)** : La capacité non-unitaire est un rectangle dont la surface est donnée par :

$$A_{NU} = \frac{C_{NU}}{C_U} \times A_U. \quad (5.17)$$

si A_U est la surface d'une capacité unitaire. On peut déterminer ainsi des dimensions pour le rectangle qui conduisent à la surface désirée en respectant le rapport périmètre/surface nécessaire. L'avantage de cette technique est qu'elle n'introduit pas d'angle droit supplémentaire dans le dessin. Or, ces angles nuisent à l'appariement. Malheureusement, il n'est pas toujours possible d'inscrire ce rectangle dans des limites géométriques qui assurent la régularité de la matrice.

- **Stub (fig. 5.19-b)** : La capacité non-unitaire est dessinée comme une capacité unitaire à laquelle on ajoute un rectangle (*stub*) de largeur $W/2$, si W est le côté de la capacité unitaire. Cette technique ajoute deux angles droits au dessin et elle n'est valable que dans le cas où $C_{NU}/C_U \in [1 ; 1.5]$.
- **Trou (fig. 5.19-c)** : Lorsqu'aucune des deux techniques précédentes n'est possible, on

représente la capacité non-unitaire par un rectangle troué. La largeur de ce rectangle est le côté de la capacité unitaire. On détermine alors une longueur pour ce rectangle et des dimensions pour le trou qui maintiennent le rapport périmètre/surface désiré. Cette technique introduit cependant deux nouveaux angles droits dans le dessin.

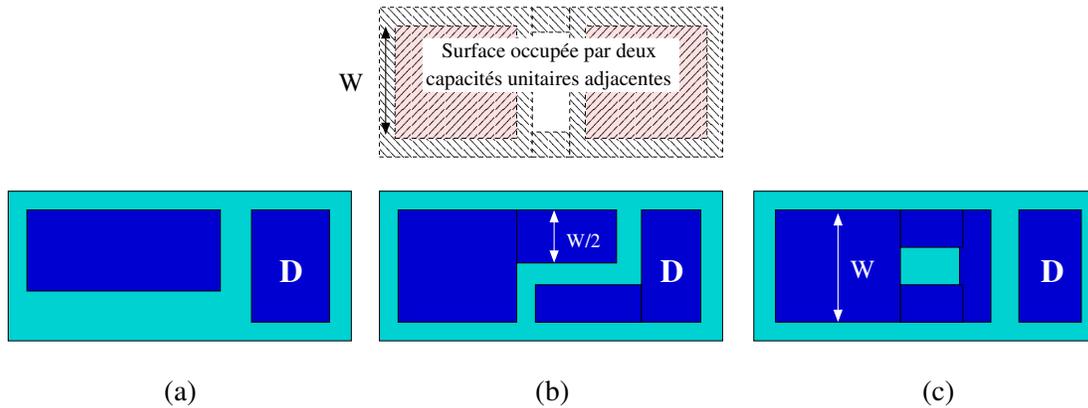


FIG. 5.19 : Capacité non-unitaire. (a) Rectangle - (b) Stub - (c) Trou.

D'un autre côté, le placement des capacités unitaires et non-unitaires dans la matrice a également une grande influence sur l'appariement. Pour minimiser l'impact du gradient d'oxyde sur l'appariement, nous savons qu'il faut aboutir à un dessin dont la géométrie est centrée. Comme nous l'avons vu, ceci est possible en rapprochant le plus possible les barycentres des éléments concernés. La figure 5.20 montre finalement le dessin de deux capacités appariées de rapport $C1/C2 = 5.4/3$. On remarque que, pour réaliser une géométrie centrée, il est souvent nécessaire d'avoir recours à des canaux de routage qui augmentent sensiblement la surface globale de la matrice. Le routage des capacités fictives n'est pas représenté sur la figure.

Pour finir, précisons que la valeur, et donc la taille, des capacités unitaires influe sur l'appariement. De grandes capacités unitaires minimisent l'effet relatif de l'excès de gravure. A l'inverse, en réduisant la taille des capacités unitaires, on augmente leur nombre, et il est plus facile d'obtenir une géométrie centrée. Il est en général possible de parvenir à un bon compromis [Shyu82].

5.3.4 Appariement de résistances

Pour appairer des résistances, on prend également en compte les excès de gravure et le gradient d'oxyde. On inter-digite donc des segments résistifs de même géométrie, et on utilise des résistances fictives. La figure 5.21 présente le dessin des masques de deux résistances égales.

Nous venons de recenser les techniques de dessin les plus utilisées pour obtenir une bonne précision et un bon appariement des composants analogiques. Nos générateurs sont en mesure de

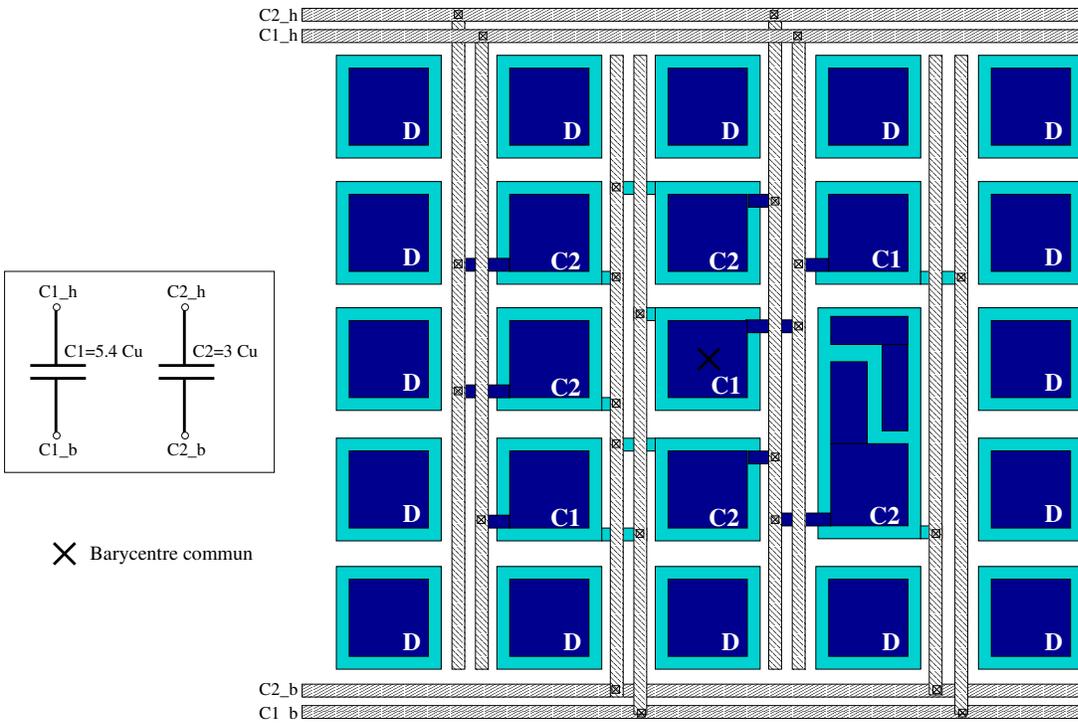


FIG. 5.20 : Dessin des masques de deux capacités appariées.

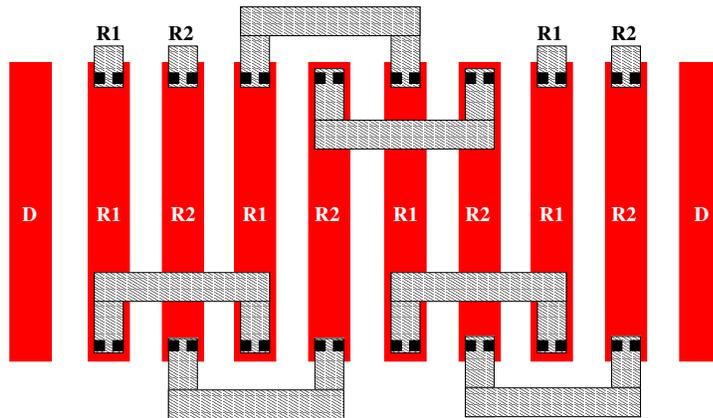


FIG. 5.21 : Dessin des masques de deux résistances appariées : $R1 = R2$.

produire des dessins issus de ces techniques. Le chapitre 7 montrera des exemples de réalisations effectivement obtenues avec CAIRO+.

5.4 Variations géométriques : dispositifs élémentaires déformables

5.4.1 Fonction de forme

Comme nous l'avons vu, l'un des principaux objectifs de *CAIRO+* est la mise en œuvre de circuits génériques. Ceci signifie que l'on doit pouvoir réutiliser nos générateurs sur différents procédés de fabrication, ou encore pour une large gamme de spécifications. En analogique, la conséquence d'une telle ambition est que l'on est amené à gérer des éléments de taille extrêmement variable. De plus, les changements observés sur ces tailles n'opèrent pas toujours dans la même direction pour tous les composants. On va donc chercher à laisser systématiquement un degré de liberté géométrique pour la réalisation physique de ces composants. On disposera alors de plusieurs facteurs de formes pour la vue physique de chaque dispositif élémentaire. Nous avons vu par exemple qu'un transistor pouvait être replié, et donc présenter différentes réalisations physiques pour des caractéristiques électriques similaires. Dans une matrice de capacités, on peut également faire varier la géométrie globale en modifiant le nombre de lignes et le nombre de colonnes de la matrice. On parle alors de dispositifs élémentaires déformables. Pour caractériser la déformabilité d'un dispositif élémentaire, on représente l'ensemble des facteurs de formes qu'il est capable de présenter dans une fonction de forme. Celle-ci donne l'ensemble des couples $(DX; DY)$ possibles pour le dispositif élémentaire, où DX est la largeur de la réalisation et DY sa hauteur. La figure 5.22 montre la fonction de forme d'un transistor. En repliant ce transistor, on réduit la hauteur globale du dessin, mais la largeur augmente. On peut ainsi obtenir plusieurs facteurs de formes pour le transistor en le repliant jusqu'à ce que la largeur des doigts ne respecte plus le W_{min} de la technologie. Le nombre de points de la fonction de forme dépend donc des dimensions du transistor vis-à-vis de ce W_{min} . Notons par ailleurs que la surface reste à peu près constante quelle que soit la forme de la réalisation.

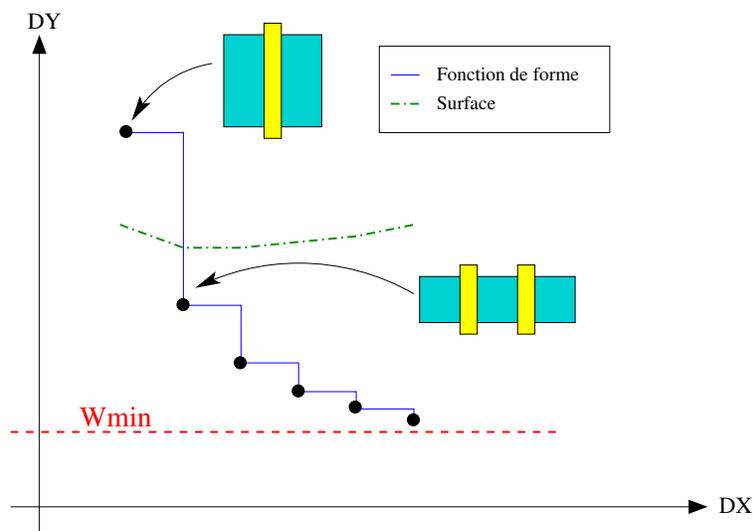


FIG. 5.22 : Fonction de forme d'un transistor.

5.4.2 Motifs sur grille mixte

La mise en œuvre de cette déformation est une tâche relativement complexe puisque les dispositifs élémentaires devront proposer plusieurs dessins des masques ayant les mêmes caractéristiques électriques. De plus, ces dessins devront être optimisés et prendre en compte les éventuelles contraintes d'appariement afin de respecter les spécificités de la conception analogique. Pour faciliter la gestion de ces multiples réalisations, nous avons choisi de définir le dessin comme un placement relatif d'objets plutôt que d'en donner une description absolue [Owen95a]. Ce choix permet notamment de systématiser la méthode de dessin des générateurs de dispositifs élémentaires. On aimerait alors utiliser le langage procédural de placement appelé GENLIB qui fait partie de la chaîne de CAO Alliance développée au LIP6.

Alliance est un environnement qui permet la conception de circuits numériques indépendants du procédé de fabrication [Pétrot94]. Pour mettre en œuvre cette indépendance, tous les outils de la chaîne Alliance travaillent sur une grille fixe, appelée grille symbolique. Le dessin sur grille symbolique est constitué de segments, auxquels on associe une largeur. Par exemple, un fil de métal habituellement dessiné comme un rectangle, sera vu sous Alliance comme un segment sur grille symbolique ayant une largeur donnée. Cela revient à dire que l'axe de ce fil se trouve sur la grille symbolique.

Par ailleurs, les masques dessinés sur la grille symbolique sont régis par un ensemble de règles de dessin symboliques invariantes, c'est-à-dire indépendantes du procédé de fabrication. Ces règles symboliques sont exprimées en λ qui est le pas de la grille symbolique. A partir du dessin symbolique d'un circuit, on peut alors obtenir le dessin des masques réels pour une technologie visée en attribuant une valeur en nanomètres à λ , qui permet de respecter toutes les règles de dessin effectivement imposées par cette technologie [Greiner95]. Le calcul de λ repose sur la correspondance entre le jeu de règles symboliques, et le jeu de règles réelles. Le tableau 5.1 illustre, de façon simplifiée, ce calcul pour deux technologies.

Règle considérée	Valeur minimale symbolique	Techno 1		Techno 2	
		Valeur minimale réelle (nm)	λ obtenu (nm)	Valeur minimale réelle (nm)	λ obtenu (nm)
W_TRANS	$2.\lambda$	450	225	200	100
L_TRANS	$1.\lambda$	350	350	180	180
W_MET1	$1.\lambda$	400	400	220	220
D_MET1	$3.\lambda$	500	175	240	080
W_CONT	$1.\lambda$	400	400	220	220
D_CONT	$3.\lambda$	450	150	240	080
Valeur du pas de grille		$\delta = 25nm$	$\lambda = 400nm$	$\delta = 10nm$	$\lambda = 220nm$

Tab. 5.1 : Calcul de λ pour deux technologies.

On considère ici six règles symboliques significatives : la largeur (W_TRANS) et la longueur (L_TRANS) minimales d'un transistor, la largeur minimale d'un fil de métal 1 (W_MET1), la distance minimale entre deux fils de métal 1 (D_MET1), la largeur minimale d'un contact (W_CONT) et la

distance minimale entre deux contacts (D_{CONT}). En pratique, il faut évidemment prendre en compte la liste exhaustive des règles symboliques. Pour chaque règle R_i de cette liste, on calcule le λ_i qui assure l'égalité entre la valeur symbolique R_{is} et la valeur réelle R_{ir} de la règle :

$$R_{is} \times \lambda_i = R_{ir} \Leftrightarrow \lambda_i = \frac{R_{ir}}{R_{is}} \quad (5.18)$$

Puisque le dessin d'un circuit numérique est composé majoritairement d'éléments minimaux, c'est-à-dire que les dimensions W et L d'un transistor ou la largeur d'un fil de routage sont souvent les plus petites permises par le procédé, l'ensemble des règles symboliques imposent des valeurs minimales. Ainsi, parmi les λ_i calculés, il suffit de conserver le plus grand pour assurer le respect de toutes les règles :

$$\lambda = \max(\lambda_i) \quad (5.19)$$

La dernière ligne du tableau 5.1 indique la valeur du λ finalement retenue en regard du pas δ de la grille physique pour les deux technologies envisagées. Il y apparaît qu'un transistor symbolique dont la longueur de la grille est $1.\lambda$ correspond à un transistor réel de longueur non minimale. La transformation du dessin symbolique vers le dessin réel permet en réalité de rattraper cette longueur réelle minimale.

Le résultat important de ce tableau est que le pas de la grille symbolique est généralement de l'ordre d'une vingtaine de pas de grille physique. Ceci ne pose pas de problème majeur dans le domaine numérique où la précision sur les dimensions des transistors n'est pas capitale. En revanche, l'idée d'utiliser un outil travaillant sur grille symbolique pour gérer les déformations des dispositifs élémentaires de CAIRO+ est incompatible avec la précision nécessaire en analogique. Pour contourner ce problème, les versions antérieures de CAIRO [Dessouky01b] utilisaient les outils d'Alliance en autorisant le dessin de rectangles dont les dimensions et les coordonnées étaient des fractions de λ . Ceci entraînait une conception relativement confuse du fait, notamment, de la manipulation parallèle des règles réelles et symboliques. Dans CAIRO+, nous proposons une approche qui s'appuie sur une manipulation distincte des grilles symbolique et réelle [Bourguet02]. Celle-ci permet ainsi de profiter de l'outil GENLIB tout en conservant une précision au δ près.

On constate en effet que la plupart des dessins de composants analogiques sont construits autour d'éléments répliqués. Par exemple, le dessin d'un transistor replié peut être vu comme un ensemble de doigts identiques. De même, une matrice de capacités est un arrangement de capacités unitaires identiques. Ainsi, on se propose de concevoir le dessin des masques d'un dispositif élémentaire comme un aboutement d'éléments que l'on appelle **des motifs**. Les motifs sont les éléments importants du dessin, c'est-à-dire qu'ils ont une très forte influence sur le comportement électrique final du dispositif élémentaire. En conséquence, les parties électriquement significatives de ces motifs (canal d'un transistor, armatures d'une capacité) seront dessinées sur la grille physique à la précision δ . En revanche, les parties moins significatives (diffusion du transistor, espace entre les capacités unitaire, etc...) seront étendues jusqu'à la grille symbolique. La figure 5.23 illustre ce mécanisme. Par souci de clarté, on a ici $\lambda = 4.\delta$. Par ailleurs, un seul contact de diffusion est représenté.

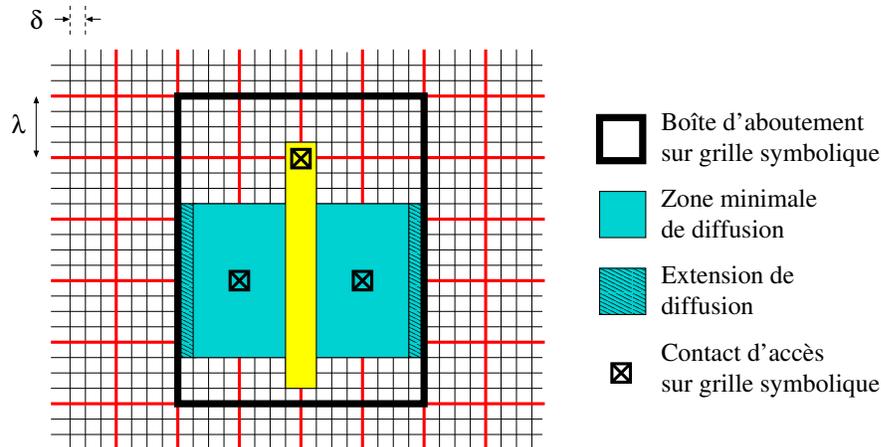


FIG. 5.23 : Dessin d'un motif : un doigt de transistor.

On dispose ainsi d'objets dont l'encombrement est aligné sur la grille symbolique mais dont les masques respectent la précision nécessaire. Ces objets peuvent être manipulés par notre langage procédural et il devient aisé d'effectuer différents placements des motifs pour constituer les différentes formes possibles de la réalisation physique. Par exemple, on va abouter un nombre variable de doigts pour constituer le transistor replié (fig 5.24), ou on va définir différents arrangements de capacités unitaires pour obtenir différentes matrices de capacités.

Comme le montre la figure 5.23, on imposera à ces motifs de présenter une interface physique sur la grille symbolique. Ceci permet que le routage interne du dispositif élémentaire devienne un routage entre motifs qui sera effectué de manière procédurale sur grille symbolique (fig 5.24).

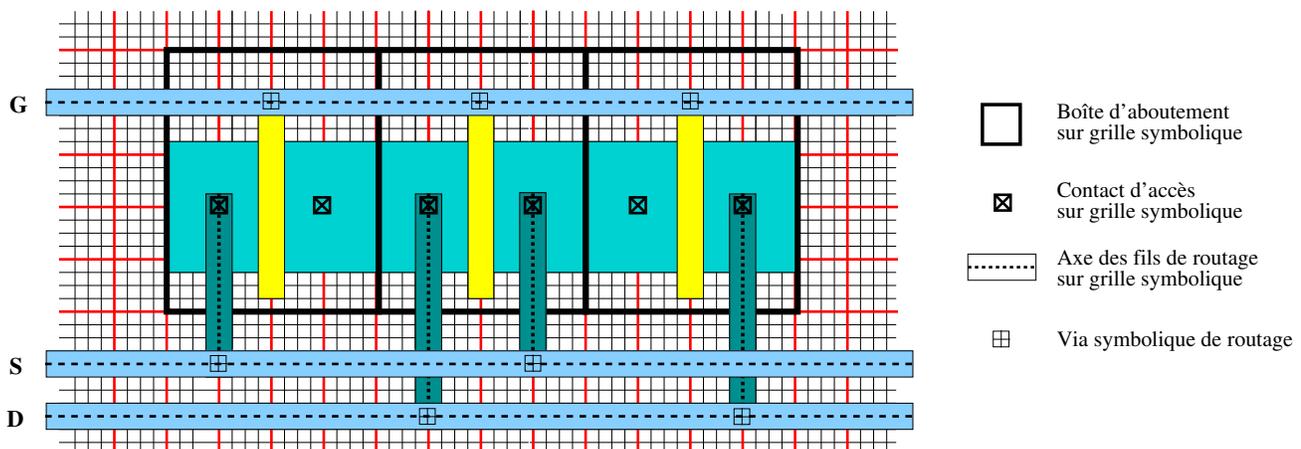


FIG. 5.24 : Placement et routage des motifs.

Dans notre méthode, le prix de la conception générique est donc une légère augmentation de la surface. En effet, la boîte d'aboutement de chaque motif devant se trouver sur la grille symbolique, sa hauteur et sa largeur peuvent connaître une augmentation allant jusqu'à $1.\lambda$. Si le dispositif

élémentaire est réalisé par un aboutement horizontal de h motifs, et un aboutement vertical de v motifs, la surface maximale qu'il pourra présenter sera :

$$A_{max} = A + hv [\lambda^2 + (dx + dy)\lambda] \quad (5.20)$$

où A , dx , et dy sont respectivement l'aire, la largeur et la hauteur de la réalisation minimale. Cette augmentation est acceptable puisque les blocs analogiques ne représentent qu'un pourcentage faible de la surface globale d'un SoC. En analogique, la surface est en conséquence moins critique que dans la conception numérique.

Pour obtenir une surface minimale, on pourrait envisager le développement d'un langage procédural de placement qui travaillerait directement sur grille physique. Dans ce cas, la méthode proposée garderait tout son intérêt. En effet, la séparation des contraintes de précision et de déformation introduite par notre approche permet de faciliter largement la conception des éléments complexes que sont les dispositifs élémentaires.

5.5 Caractérisation du dessin des masques : documentation de la vue structurelle

Comme nous l'avons précisé, le dessin des masques a une forte influence sur le comportement d'un circuit analogique. En effet, les dimensions effectives des composants réalisés ne sont pas toujours parfaitement égales à celles spécifiées. De plus, selon le style du dessin réalisé, les parasites introduits peuvent être différents. Les dimensions effectives et les parasites doivent être pris en compte afin d'obtenir une estimation aussi précise que possible des performances, et permettre à la boucle de synthèse *CAIRO+* (fig. 4.1) d'aboutir à un circuit qui, après extraction et simulation, atteint les performances initialement désirées. En ce qui concerne les parasites, on va distinguer deux catégories :

- Ceux issus des composants eux-mêmes. On trouvera par exemple dans cette catégorie les capacités intrinsèques et extrinsèques d'un transistor. Ces parasites varient avec la géométrie du dessin mais aussi avec les conditions de fonctionnement comme la polarisation ou la température. Pour caractériser le dessin vis-à-vis de ce type de parasites, on associera à l'élément structurel qui représente le composant dans la vue structurelle, des informations géométriques décrivant sa réalisation. La calculette pourra utiliser ces informations pour déterminer les parasites mis en jeu par le composant. Nous allons voir que cette pratique s'applique en particulier au transistor dont le modèle est suffisamment élaboré.
- Ceux issus du routage qui dépendent principalement de la géométrie des fils de routage. Ils seront introduits directement dans la vue structurelle sous forme d'éléments structurels parasites. A l'heure actuelle, nous n'avons pris en compte que les capacités parasites et il existe dans *CAIRO+* des capacités structurelles parasites. Celles-ci disposent de leur propre calculette qui permet de déterminer la valeur des capacités parasites en fonction de la géométrie des fils de routage.

Comme le montre la figure 5.25, le dispositif élémentaire peut alors effectuer une estimation des performances en s'appuyant sur les informations contenues dans la vue structurelle, et sur les différentes caulettes. D'un autre côté, nous allons voir que le *driver spice* intégré à *CAIRO+* utilise toutes ces informations pour générer une *netlist* proche de celle issue d'un extracteur classique.

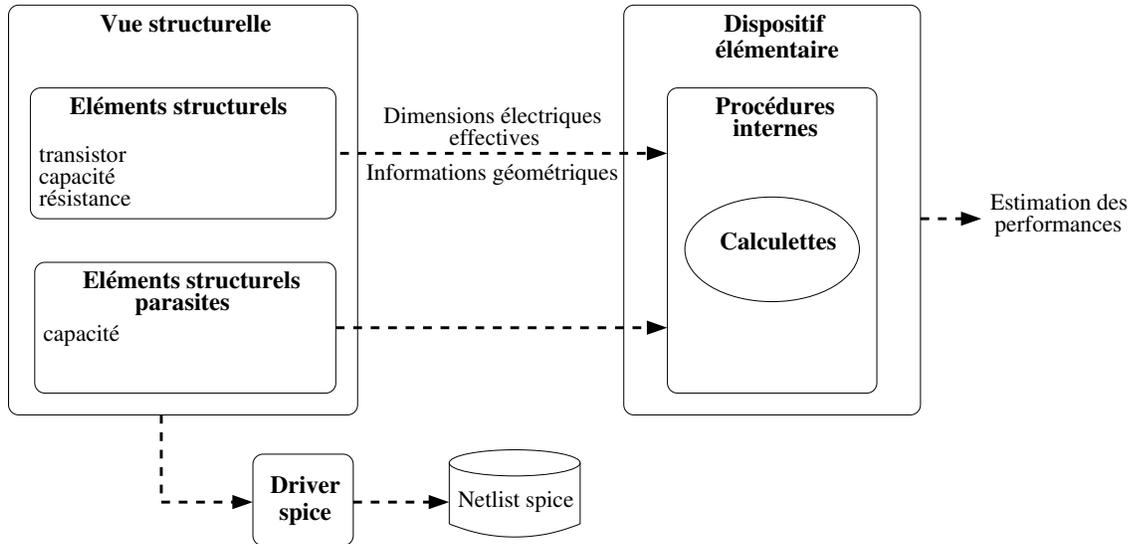


FIG. 5.25 : Estimation des performances.

5.5.1 Capacités parasites de routage

Les parasites issus du routage interne d'un dispositif élémentaire sont explicitement instanciés dans la vue structurelle de ce dispositif. Autrement dit, pendant l'étape de création, un générateur de dispositifs élémentaires crée une vue structurelle qui doit faire apparaître des éléments structurels parasites. A l'heure actuelle, seules les capacités parasites ont été prises en compte. Ainsi, le générateur de transistors crée la vue structurelle de la figure 5.26-a. Celle-ci instancie un transistor structurel et lui ajoute toutes les capacités parasites qui peuvent intervenir entre chaque couple de connecteurs. Grâce à la caulette de capacités parasites, on peut calculer la valeur de ces parasites. De même, on peut associer des capacités parasites par rapport au substrat à chacune des armatures d'une capacité MIM (fig. 5.26-b). On remarque que toutes les vues structurelles *CAIRO+* proposent un connecteur externe de substrat (*B*). On peut ainsi instancier des capacités parasites de routage par rapport au substrat à tous les niveaux d'une vue structurelle hiérarchique.

L'intérêt de cette représentation et qu'elle permet de faire intervenir des parasites sur tous les nœuds du schéma électrique. Elle permet également de générer une *netlist* au format *spice* qui tient compte des parasites de routage. En effet, dans la *netlist* générée, chaque dispositif élémentaire sera représenté par un sous-circuit contenant :

- Le ou les composants du dispositif.
- Les capacités parasites de routage.

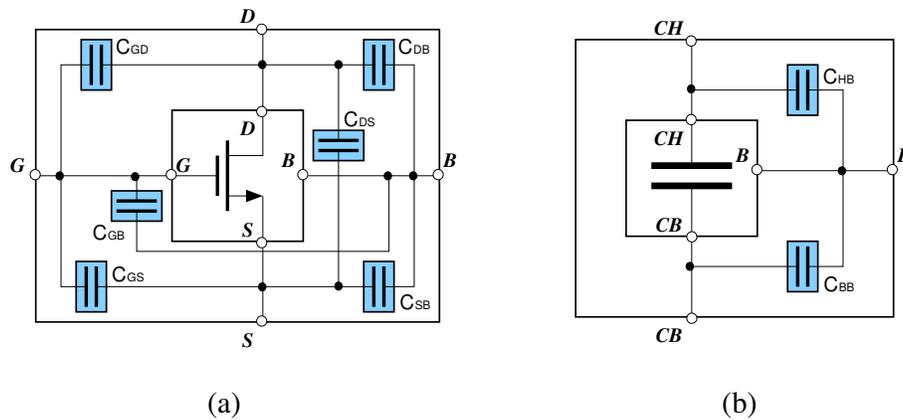


FIG. 5.26 : Capacités parasites de routage dans (a) un transistor et (b) une capacité.

5.5.2 Élément structurel transistor

Nous avons vu dans la section 5.2 que les parasites mis en jeu lors de la réalisation d'un transistor dépendent notamment de l'aire et du périmètre des diffusions. Ainsi, la géométrie du dessin des masques d'un transistor doit être entièrement caractérisée pour que la calculette, en s'appuyant sur les équations du modèle Bsim3v3, puisse déterminer ces parasites. Pour permettre à la calculette de disposer des informations géométriques relatives au dessin, celles-ci sont associées, à la fin de la génération, à l'élément structurel représentant le transistor considéré dans la vue structurale. La difficulté est que, dans le cas de transistors appariés, le dessin des masques est construit autour d'empilages qui font chacun intervenir plusieurs transistors électriquement indépendants (fig. 5.17-c). Il faut alors pouvoir caractériser le dessin de chacun de ces transistors à partir de la géométrie globale des empilages. Pour cela, chaque transistor structurel contient les informations suivantes :

- Wf et Lf : Dimensions effectives d'un doigt.
- M : Nombre de repliements.
- d_{gg} : Distance entre deux grilles consécutives
- d_e : Largeur de la diffusion en fin d'empilage
- ND_{in} et NS_{in} : Nombres de diffusions de drain et de source internes, c'est-à-dire étant enfermées entre deux grilles. Ces diffusions ont une largeur d_{gg} .
- ND_{ex} et NS_{ex} : Nombres de diffusions de drain et de source externes, c'est-à-dire se trouvant à l'extrémité de l'empilage et n'étant pas enfermées entre deux grilles. Ces diffusions ont une largeur d_e .
- $NTSD$: Nombre de transistors partageant la diffusion de drain.
- $NTSS$: Nombre de transistors partageant la diffusion de source.

Grâce à ces informations, la calculette peut notamment obtenir l'aire et le périmètre des zones de diffusion du transistor considéré. Lorsque ces zones sont partagées par plusieurs transistors, elle peut n'attribuer au transistor considéré qu'une fraction des parasites qui en sont issus.

Le *driver spice* utilise également ces informations pour préciser le nombre de repliements, l'aire et le périmètre des zones de diffusion, au moment de l'instanciation d'un transistor dans la netlist générée.

5.5.3 Élément structurel capacité MIM

Une capacité structurelle contient la valeur de la capacité. Ici, en effet, le principal type de parasites qui n'est pas directement issu du routage est la capacité parasite entre l'armature basse et le substrat. Celle-ci est uniquement liée à la géométrie de l'armature basse et il est relativement facile de la calculer. Pour simplifier notre représentation, nous l'avons assimilée à une capacité parasite de routage entre l'armature basse et le substrat. Ainsi, dans la vue structurelle du dispositif de capacité représentée sur la figure 5.26-b, elle sera ajoutée à C_{BB} , qui est la capacité parasite de routage entre l'armature basse et le substrat.

Par ailleurs, dans une matrice de capacité, le routage peut également créer une capacité parasite entre les deux armatures d'une capacité MIM. Dans ce cas, on ajoute directement cette capacité parasite à la valeur nominale de la capacité.

5.5.4 Élément structurel résistance

Une résistance est généralement constituée de segments résistifs montés en série et reliés par des fils métalliques. Une capacité parasite par rapport au substrat existe alors au niveau de chacun des segments résistifs ou des fils métalliques. Elle intervient cependant sur des noeuds qui ne sont pas représentés dans la vue structurelle et ne peut pas être considérée comme une capacité parasite de routage. C'est pourquoi on utilise le modèle de la figure 5.27 pour représenter chaque segment résistif. Pour prendre cette modélisation en compte, la résistance structurelle contient les

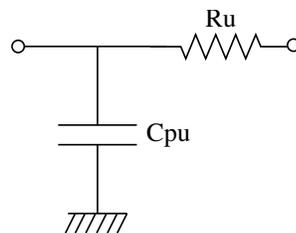


FIG. 5.27 : Modèle utilisé pour un segment résistif.

informations suivantes :

- R_u : Valeur de la résistance d'un segment.
- C_{pu} : Valeur de la capacité parasite associée à un segment. Elle correspond à la fois à la capacité entre le segment et le substrat, et à la capacité entre le fil métallique et le substrat.
- N_u : Nombre de segments.

Cette représentation permet aux procédures de la vue comportementale de disposer de toutes les informations nécessaires à l'estimation des performances, et au *driver spice* de générer un

sous-circuit composé de résistances élémentaires et de capacités parasites, fidèle au dessin des masques.

5.6 Indépendance technologique : architecture logicielle adaptée

La conception d'un générateur de dispositifs élémentaires qui prenne en compte l'ensemble des spécificités analogiques est une tâche complexe. Elle nécessite la mise en œuvre d'une architecture logicielle à la fois robuste et suffisamment souple pour être générique. Nous avons déjà précisé que le langage *CAIRO+* permet d'instaurer, au cours du flot, des communications grâce auxquelles chaque générateur est indépendant des spécifications. Dans cette section, nous allons voir que l'implémentation d'un générateur de dispositif élémentaire lui permet de s'adapter aux contraintes technologiques, quelles qu'elles soient [Bourguet04b]. Le chapitre 6 reprendra plus en détail les concepts exposés ici en reprenant une par une les étapes de la conception d'un générateur de dispositifs élémentaires.

5.6.1 Les quatre fonctions constructrices

Au chapitre 4, nous avons vu que la génération des dispositifs élémentaires se faisait en trois étapes. Pour concevoir un générateur de dispositifs élémentaires, il faut écrire les fonctions *CAIRO+* mettant en œuvre chacune de ces étapes. L'écriture de ces fonctions se fait en langage *CAIRO+*.

L'**étape de création** est prise en charge par la fonction `CAIRO_CREATE`. Il s'agit ici de créer les trois vues du dispositif élémentaire :

- La vue structurelle correspond au schéma électrique réalisé. Il faut y instancier le ou les éléments structurels qui interviennent dans le dispositif, et décrire leurs interconnexions. Les dimensions de ces composants sont initialisées aux valeurs minimales et seront régulièrement mises à jour au cours du flot. Il faut également y définir l'interface électrique du dispositif afin que cette vue structurelle puisse elle-même être instanciée dans celle du niveau hiérarchique supérieur.
- La création de la vue comportementale consiste à déclarer les différentes procédures qui la composeront, ainsi que les paramètres d'entrée et de sortie de ces procédures.
- En ce qui concerne la vue physique, on se contente de créer les structures de données qui la contiendront. Elle ne sera complètement décrite que lors de l'étape de dessin des masques.

La figure 5.28 montre les trois vues d'un dispositif élémentaire de transistor à l'issue de l'étape de création. On voit que les dimensions du transistor sont minimales et que les capacités parasites (notées C_{par}) sont nulles. La vue comportementale ne fait que recenser les procédures de dimensionnement ainsi que leurs paramètres d'entrée et de sortie. Enfin, la vue physique est vide.

Pour construire ces trois vues, la fonction `CAIRO_CREATE` reçoit des **paramètres structurels**. De manière générale, un générateur *CAIRO+* est dédié à une topologie invariante qui est

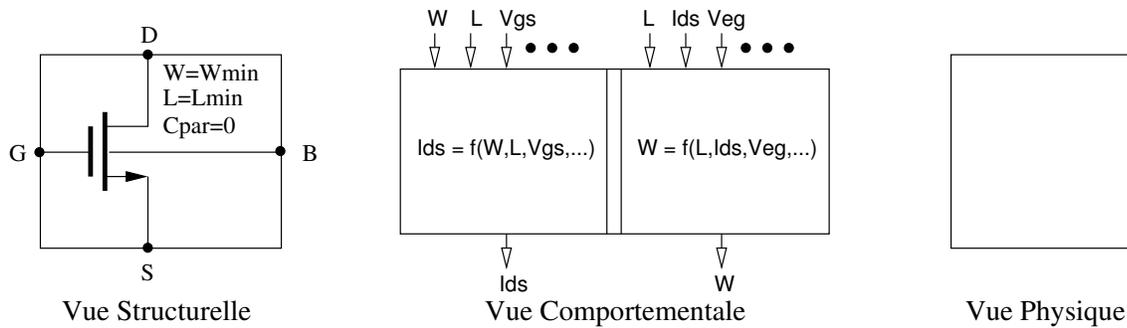


FIG. 5.28 : Les trois vues d'un dispositif élémentaire après création.

représentée par sa vue structurale. Cette contrainte apparaît trop restrictive lorsque l'on considère certaines topologies très semblables pour lesquelles on souhaiterait utiliser un même générateur. Par exemple, le dimensionnement d'un transistor se fait de la même façon qu'il s'agisse d'un transistor N ou P. Les dessins des masques de transistors de type différents sont également très semblables. De même, un miroir de courant est régulièrement utilisé comme dispositif de polarisation alimentant plusieurs branches d'un circuit. Le nombre de transistors qui constituent le miroir dépend dans ce cas du nombre de branches à alimenter. Puisqu'ils sont liés par des rapports précis, il suffit de dimensionner l'un des ces transistors pour obtenir les dimensions des autres et le dimensionnement se fait de la même façon quel que soit le nombre de transistors qui constituent le miroir.

Les paramètres structurels permettent de modifier la topologie issue d'un même générateur : le type d'un transistor ou le nombre de branches d'un miroir de courant seront des paramètres structurels. Pour conserver une cohérence électrique, comme dans les exemples que nous venons de citer, les paramètres structurels ne doivent avoir aucune influence sur le dimensionnement.

L'étape de dimensionnement électrique du dispositif élémentaire est effectuée par la fonction `CAIRO_DSES` (*Design Space Exploration and Sizing*). Celle-ci contient la description de toutes les procédures permettant :

- d'indiquer la faisabilité des spécifications d'entrée du dispositif.
- d'attribuer des dimensions électriques aux composants impliqués dans le dispositif.
- de déterminer le comportement qu'adoptera ce dispositif pour des dimensions données.

C'est donc à ce niveau que le concepteur analogicien exprimera toute sa connaissance. Pour cela, `CAIRO+` met à sa disposition un langage adapté grâce auquel il pourra accéder à une modélisation précise des composants (`Bsim3v3`) par l'intermédiaire de la calculatrice, mais également à des informations relatives au procédé de fabrication. Ces informations lui seront en particulier nécessaires pour vérifier que le dimensionnement aboutit à des dimensions électriques qui respectent les règles imposées par le fondeur. Par exemple, la largeur W d'un transistor issue d'un dimensionnement doit respecter la largeur minimale W_{min} du procédé.

L'ensemble de ces procédures s'appuie sur des paramètres issus du niveau hiérarchique supérieur, on les appelle les **paramètres électriques d'entrée**. A l'inverse, les informations renvoyées à ce niveau supérieur sont appelées **paramètres électriques de sortie**.

Les procédures qui aboutissent aux dimensions électriques de tous les composants du dispositif sont qualifiées de complètes. Une procédure complète doit attribuer les dimensions obtenues aux éléments structurels afin d'avoir une première description électrique du dispositif. Pour lancer la génération des masques, il est nécessaire d'avoir fait appel à une procédure complète.

Le dessin des masques des différents facteurs de formes est pris en charge par deux fonctions : `CAIRO_SHAPE` et `CAIRO_GENERATE`. C'est sur l'écriture de ces fonctions que s'est concentré ce travail. La fonction `CAIRO_SHAPE` est celle qui est effectivement appelée par `CAIRO+` pour lancer la phase de réalisation physique. Elle s'occupe de déterminer la fonction de forme du dispositif. Pour cela, elle commence par calculer le nombre de formes possibles avec les dimensions électriques issues de l'étape précédente, et pour le procédé de fabrication visé. Par exemple, pour déterminer le nombre de formes possible d'un transistor, il faut calculer le nombre maximal de repliements M_{max} donné par :

$$M_{max} = \left\lfloor \frac{W}{W_{min}} \right\rfloor \quad (5.21)$$

Celui-ci dépend de la largeur W à laquelle le dimensionnement électrique a abouti, et de la largeur minimale autorisée par la technologie. L'écriture de `CAIRO_SHAPE` nécessite donc un accès aux informations technologiques.

Lorsque l'on connaît le nombre de formes, il faut déterminer l'encombrement de chacune des réalisations afin de constituer la fonction de forme. Pour cela, on génère effectivement les dessins des masques correspondant à tous les facteurs de formes. Cette génération est effectuée par des appels successifs à la fonction `CAIRO_GENERATE` qui réalise une forme donnée. Comme nous l'avons vu dans la section 5.4, la réalisation se fait par aboutement de motifs. Ainsi, `CAIRO_GENERATE` fait appel à des générateurs de motifs, puis place et route les motifs obtenus pour construire le dessin des masques global du dispositif élémentaire. Dans la section suivante, nous allons voir que les générateurs de motifs sont des générateurs de modèles. A partir de ce modèle, il est donc possible de créer plusieurs instances d'un motif.

La génération du dessin est conditionnée par un ensemble de paramètres, appelés **paramètres de réalisation**, spécifiés dans un fichier de réalisation. L'utilisateur pourra par exemple y indiquer s'il désire la présence d'un anneau de garde, ou d'une diode de protection contre l'effet d'antenne. Les paramètres de réalisation sont optionnels, et, s'il ne sont pas spécifiés, un dessin par défaut sera généré. Par ailleurs, la fonction `CAIRO_GENERATE` est paramétrée par un **indice de forme** qui lui indique quelle est la réalisation physique qu'elle doit prendre en charge. Pour le transistor, cet indice sera le nombre de repliements M , pour une matrice de capacité, il pourra être le nombre de lignes de la matrice.

Pour effectuer la génération d'un dispositif élémentaire, on s'appuie donc sur un ensemble de générateurs de motifs et sur des informations technologiques. Ces dernières permettent de valider le dimensionnement et de déterminer le nombre de formes possibles. Un accès aux informations technologiques est fourni au générateur de dispositifs élémentaires par la DAPI, que nous allons présenter dans la section suivante.

5.6.2 Interface entre les dispositifs élémentaires et le procédé de fabrication : la DAPI

Pour permettre la migration technologique, nous avons essayé d'isoler au maximum le générateur de dispositifs élémentaires vis-à-vis des informations technologiques. Cet isolement est assuré par la DAPI (Dispositif API) qui joue le rôle d'interface entre le dispositif élémentaire et la technologie. Il s'agit d'un ensemble de fonctions spécifiques à un générateur de dispositifs élémentaires qui ont un accès direct aux informations technologiques. Ces fonctions effectuent les calculs nécessaires pour délivrer au générateur les informations formatées dont il a besoin. Par exemple, comme le montre la figure 5.29, la largeur minimale d'un transistor peut être sensiblement différente de celle directement imposée par la technologie lorsqu'un contact est placé sur la diffusion. Dans ce cas en effet, elle est définie comme le maximum entre W_{min} et la somme de la largeur L_C du contact et de deux fois l'extension E_{DC} de diffusion autour du contact. Ce type de calcul est effectué par les fonctions de la DAPI afin de fournir au dispositif élémentaire des informations géométriques traitées selon les contraintes du dessin.

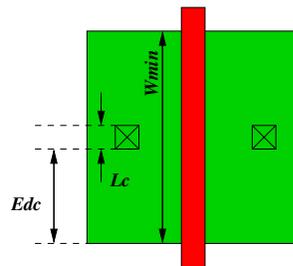


FIG. 5.29 : DAPI : Calcul de W_{min} .

D'un autre côté, nous avons vu que le dispositif élémentaire prenait en charge le routage de son propre dessin des masques. Il effectue ce routage de façon symbolique grâce aux fonctions du langage CAIRO+, dérivées de celle de GENLIB. Cependant, le routage de certains dispositifs s'effectue sur des couches qui dépendent de la technologie visée. Par exemple, une capacité MIM sera réalisée dans des métaux différents selon le procédé. En conséquence, le routage d'une matrice de capacité ne s'effectue pas en utilisant toujours les mêmes métaux. La couche des fils de routage sera déterminée par la DAPI. Celle-ci s'appuie pour cela sur une représentation adaptée des informations technologiques contenue dans le fichier DTR dont nous parlerons à la section 5.6.5.

5.6.3 Générateurs de motifs

La génération du dessin d'un dispositif élémentaire est faite par un placement et un routage de motifs. Les motifs contiennent les parties électriquement sensibles du dessin des masques. Ils correspondent aux capacités unitaires d'une matrice de capacités, ou aux doigts d'un transistor replié. Nous avons vu qu'ils étaient dessinés sur la grille physique proposée par le fondeur, mais qu'ils présentaient une interface physique sur grille symbolique. Bien que la chaîne de CAO Alliance serve à concevoir des masques symboliques, elle permet également de représenter un dessin des masques réels. Les structures de données permettant les représentations symbolique et réelle sont toutefois différentes. Nous nous servons de ces deux types de structures pour représenter le dessin d'un motif. Le dessin d'un motif est ainsi défini par une vue réelle et une vue symbolique :

La vue réelle contient le dessin de tous les rectangles du motif. Elle définit des informations directement exploitables par un fondeur en vue de la fabrication.

La vue symbolique du motif contient des rectangles virtuels, c'est-à-dire des informations uniquement exploitables par l'outil de placement et routage symbolique. En l'occurrence, il s'agira de l'interface physique du motif, c'est-à-dire de la boîte d'aboutement et des connecteurs. Un connecteur permet de référencer un point du dessin en lui attribuant un nom et une couche. Les connecteurs doivent être placés sur les points permettant de réaliser la connexion physique avec le motif, comme le contact de grille d'un transistor. Dès lors, le routage procédural se fait en précisant simplement le nom des connecteurs que relie un fil sans se soucier de leurs coordonnées.

La figure 5.30 illustre cette double représentation sur l'exemple d'un motif de doigt de transistor. Dans Alliance, la grille physique et la grille symbolique partagent la même origine. Ceci assure la correspondance géométrique entre les deux vues représentatives du motif. C'est uniquement

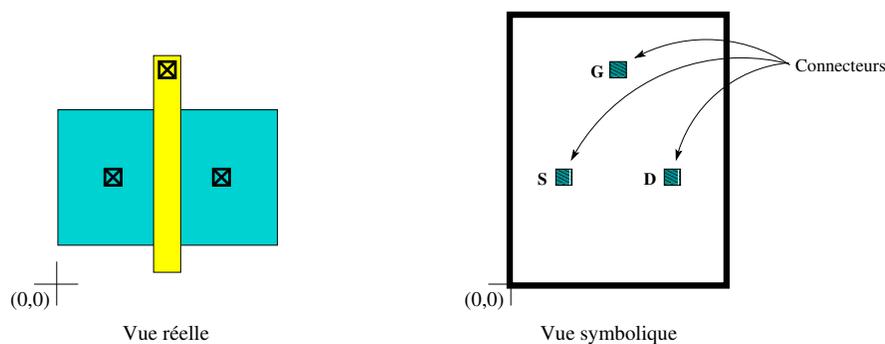


FIG. 5.30 : Vues réelle et symbolique d'un motif.

l'interface physique, donc la vue symbolique, de ces motifs qui sera manipulée par le langage de placement GENLIB pour constituer le dessin final du dispositif élémentaire. A la fin du flot de génération d'un circuit sous CAIRO+, le dessin des dispositifs élémentaires est un assemblage de vues symboliques. L'une des dernières étapes du flot est la substitution de ces vues symboliques

par les vues réelles correspondantes. Cette substitution n'est possible que si les deux vues d'un même motif portent le même nom.

Dans la mesure où l'on vise une génération des masques sur différents procédés, la difficulté au niveau de la conception des motifs réside dans la prise en compte des règles de dessin spécifiques à la technologie ciblée. Les générateurs de motifs encapsulent une grande partie de l'expertise nécessaire à la réalisation physique. En conséquence, on va chercher à séparer les difficultés liées à cette réalisation. Ainsi, les générateurs de motifs se contenteront de générer les rectangles constituant le dessin des masques. Tous les calculs permettant à ces rectangles de respecter les règles de dessin seront effectués par un ensemble de fonctions appelé MAPI (Motif API). Chaque générateur de motif dispose de sa propre MAPI qui lui sert d'interface avec la technologie. Les fonctions de la MAPI bénéficient d'un accès direct aux informations technologiques et peuvent prendre en compte les règles de dessin. La section 5.6.4 présente plus en détails le rôle de la MAPI.

L'écriture d'un générateur de motif est systématique. On commence par créer les vues symbolique et réelle du motif. Puis, on fait appel aux fonctions de la MAPI afin qu'elles déterminent les dimensions et les positions relatives de tous les rectangles physiques pour la vue réelle, et les dimensions de la boîte d'aboutement, la position et la couche des connecteurs pour la vue symbolique. Il suffit alors de générer effectivement le contenu de ces deux vues grâce au langage CAIRO+.

Les générateurs de motifs s'appuient sur un ensemble de paramètres qui peuvent être classés en deux catégories :

- **Paramètres standard** : Il s'agit du jeu de paramètres nécessaire et suffisant à la génération du motif. Le nom du modèle généré est un paramètre standard commun à tous les générateurs de motifs. Les autres dépendent du motif considéré. Par exemple, pour générer un doigt de transistor, il faut au moins disposer de ses dimensions W et L , et de son type N ou P . Un générateur de doigts de transistor propose donc quatre paramètres standard : le nom du modèle, la longueur L , la largeur W et le type.
- **Paramètres contextuels** : Puisque les générateurs de motifs et la MAPI qui leur est associée sont des éléments de grande complexité, on va chercher à les réutiliser dans différents contextes. Par exemple, tous les dispositifs élémentaires à transistors utiliseront les mêmes générateurs de motifs. Pour que chaque générateur puisse s'adapter à son contexte d'instanciation, on lui fournit un certain nombre de paramètres contextuels. Comme nous le verrons au chapitre 7, l'interface physique d'un doigt de transistor est par exemple définie par un ensemble de paramètres contextuels.

Enfin, tous les générateurs de motifs renvoient une structure de données identique contenant les informations géométriques relatives au dessin généré. Cette structure contient donc des champs très hétérogènes et chaque générateur remplira ceux qui lui correspondent. L'utilisation d'une structure unique allège le code du générateur de dispositifs élémentaires qui la lit. La structure de

retour des générateurs de motifs est donnée en annexe et le chapitre 7 reviendra sur les champs spécifiquement remplis par chaque générateur.

5.6.4 Interface entre les générateurs de motif et le procédé de fabrication : La MAPI

La MAPI effectue tous les calculs permettant au générateur de motif de réaliser un dessin conforme. Elle se situe ainsi au cœur du problème de la génération du dessin sur différents procédés de fabrication. La difficulté ici est liée à la grande diversité de règles qui peuvent être imposées par les différents fondeurs. Les jeux de règles à respecter sont à la fois complexes et variables. Pour résoudre ce problème, on va chercher à transformer ces multiples jeux en un unique jeu de règles *CAIRO+* simple. Le jeu de règles *CAIRO+* définit chaque rectangle du dessin par cinq variables [Bourguet04a] :

- L'abscisse x du coin inférieur gauche
- L'ordonnée y du coin inférieur gauche
- La largeur DX
- La hauteur DY
- La couche *LAYER*

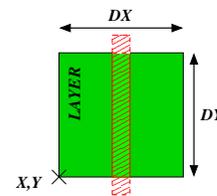


FIG. 5.31 : Jeu de règles *CAIRO+*.

La MAPI propose donc cinq fonctions par rectangle de la vue réelle du motif. D'un autre côté, la MAPI permet de construire la vue symbolique du motif. Pour cela, elle propose également deux fonctions calculant les dimensions en λ de la boîte d'aboutement. En revanche, les coordonnées et la couche des connecteurs de la vue symbolique sont directement déduites de celles des rectangles réels sur lesquels ils seront superposés. Finalement, la MAPI d'un motif contenant n rectangles propose $5.n + 2$ fonctions.

Dès lors, pour effectuer la transformation du jeu de règles fondeur vers le jeu de règles *CAIRO+*, la MAPI possède un atout majeur : elle est spécifique à un générateur de motif. Elle connaît ainsi l'environnement de chaque rectangle et sait quelles sont les règles qu'il doit respecter. De plus, tous les dispositifs élémentaires effectuent un aboutement de motifs d'un même type. Par exemple, les générateurs de transistors et de paires différentielles utiliseront le même ensemble de motifs. En conséquence, la MAPI connaît le type de motif qui peut être abouté au dessin dont elle a la charge et peut faire respecter les règles qui interviennent entre rectangles appartenant à des motifs adjacents.

Les dimensions et la position relative de la plupart des rectangles d'un motif sont les plus petites possibles afin d'économiser la surface de silicium. Seuls les rectangles directement liés aux dimensions **électriques** du motif peuvent être non minimaux. Par exemple, seule la grille d'un doigt de transistor ou l'armature haute d'une capacité unitaire sera non minimale. Or, la validité de ces dimensions a été assurée pendant l'étape de dimensionnement et lors du calcul du nombre de formes. On est donc sûr, pendant la génération du dessin, que ce type de rectangles ne posera pas

de problème. En conséquence, la MAPI ne se confronte qu'à des règles qui imposent des valeurs minimales. Pour les respecter, elle effectue des calculs de maximum afin de déterminer la valeur la plus contraignante, selon la technologie visée. Par exemple, l'extension minimale de diffusion autour de la grille E_{DG} peut être déterminée par deux combinaisons de règles (fig 5.32) :

1. L'extension directement imposée par la technologie : $E1$
2. La somme de la distance entre la grille et un contact de diffusion ($E21$), de la largeur d'un contact ($E22$) et de l'extension de diffusion autour du contact ($E23$) :

$$E2 = E21 + E22 + E23$$

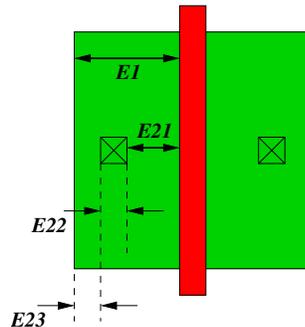


FIG. 5.32 : Type de calcul de la MAPI.

Selon la valeur de chacune des règles impliquées, c'est-à-dire selon le procédé visé, l'une ou l'autre de ces deux combinaisons peut être la plus contraignante. La plus contraignante étant celle qui correspond à la valeur la plus grande, la MAPI effectuera le maximum de ces deux combinaisons afin de déterminer la valeur à respecter finalement :

$$E_{DG} = \max(E1, E2) \quad (5.22)$$

Lorsque le dessin minimal du motif est disponible, certains rectangles moins significatifs doivent cependant être étendus jusqu'à la boîte d'aboutement. Par exemple, et comme l'illustrent les figures 5.23 et 5.24, les zones de diffusion entre deux doigts de transistor consécutifs doivent être étendues jusqu'à la boîte d'aboutement afin d'assurer la continuité de cette diffusion. A ce stade, il est donc nécessaire de disposer des dimensions de cette boîte d'aboutement. Pour cela, à partir du dessin minimal du motif, on calcule les dimensions symboliques de la boîte d'aboutement. Elles correspondent à l'arrondi par excès au λ de la hauteur et de la largeur du dessin minimal.

La couche de tous les rectangles du motif est directement dictée par la technologie. En effet, si certains rectangles, comme la grille d'un transistor *low-leakage* (transistor usuel à faible courant de fuite), sont toujours dessinés dans la même couche, d'autres sont dessinés dans une couche qui dépend de la technologie. Par exemple, le poly2 peut être recommandé pour dessiner l'armature haute d'une capacité MIM dans une technologie, et le metal4 peut l'être dans une autre.

Cette méthode s'appuie sur le fait que le jeu de règles définissant un motif, c'est-à-dire le nombre de fonctions de la MAPI, est invariant. Ceci suppose que le nombre de rectangles qui constituent le dessin du motif soit constant. Ce n'est malheureusement pas le cas. D'une technologie à une autre, il est par exemple possible que le dessin d'une capacité MIM soit défini par un nombre différent de rectangles. Certaines proposent en effet des métaux dédiés au dessin des armatures, d'autres utilisent des métaux qui peuvent également être utilisés pour le routage. Dans ce dernier cas, il faut généralement préciser la localisation de la capacité pour y traiter spécifiquement l'oxyde. Cette localisation est effectuée par un rectangle supplémentaire, appelé marqueur, qui indique la zone de traitement de l'oxyde pour la réalisation de capacités MIM. Pour répondre au nombre variable de rectangles d'un motif, nous avons introduit une couche "NO_LAYER" qui permet d'inhiber la génération d'un rectangle (cf. 5.6.5). Ainsi, il faut concevoir le motif de manière à ce qu'il contienne tous les rectangles qui peuvent potentiellement apparaître dans toutes les technologies envisagées. La MAPI peut alors inhiber la génération des rectangles qui ne sont pas concernés par le procédé visé. Nous verrons à la section 5.6.6 ce qu'il se passe lorsqu'une nouvelle technologie introduit des rectangles qui n'avaient pas été prévus.

D'un autre côté, le nombre de rectangles formant le dessin d'un motif peut également varier avec les dimensions de l'élément dessiné. Nous avons vu par exemple qu'afin de minimiser la résistance d'accès à la diffusion d'un transistor, on y place généralement autant de contacts que possible (fig 5.33-a). Le nombre de ces contacts dépend donc de la largeur W du transistor. On ne peut pas, dans ce cas, utiliser la méthode par inhibition puisqu'elle suppose que l'on ait défini un nombre maximal de contacts pouvant intervenir. Or, selon les règles de la technologie et la taille du transistor, le nombre de contacts nécessaires sera extrêmement variable. Cependant, tous les rectangles impliqués ici étant de même nature, ils sont tous régis par les mêmes règles de dessin. Dans ce cas, on introduit un **macro-rectangle** qui définit la zone dans laquelle doit être inséré un maximum de contacts. Les coordonnées et les dimensions de ce macro-rectangle doivent être calculées de manière à ce qu'un contact placé n'importe où à l'intérieur respecte les règles de dessin avec son environnement (fig 5.33-b). La MAPI propose donc aussi des fonctions de génération des

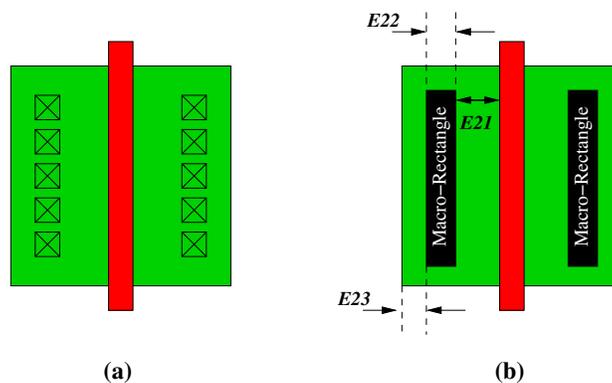


FIG. 5.33 : Exemple de macro-rectangle.

macro-rectangles qui s'ajoutent aux $5.n + 2$ fonctions initiales. Pour cet exemple, la fonction placera le maximum de contacts dans le macro-rectangle en fonction de la dimension d'un contact et de la distance minimale séparant deux contacts.

Pour finir, signalons que la position et la taille des rectangles d'un motif sont souvent relatives, c'est-à-dire qu'elle dépendent de la position ou de la taille d'autres rectangles du motif. Les différentes fonctions de la MAPI peuvent donc s'appeler entre elles. Ces dépendances relatives se font entre les fonctions calculant des dimensions verticales (Y ou DY) d'une part, et celles calculant des dimensions horizontales (X ou DX) d'autre part. Evidemment, des dépendances cycliques empêcheraient le bon fonctionnement et l'ensemble des dépendances de la MAPI peuvent ainsi être décrites par deux graphes orientés acycliques (DAG), appelés graphes de dépendance. Plus les dépendances sont nombreuses entre les fonctions de la MAPI, plus la maintenance et la mise à jour seront aisées.

Une propriété importante des graphes de dépendance est que les fonctions qui calculent la géométrie des rectangles représentant les dimensions électriques du motif ne dépendent d'aucune autre. En effet, les dimensions de l'armature haute d'une capacité, ou de la grille d'un transistor sont directement issues des paramètres standard du générateur de motifs.

5.6.5 Représentation adaptée des informations relatives au procédé de fabrication : Les fichiers RDS et DTR

La méthode envisagée repose sur l'existence d'une description adaptée des règles et informations technologiques. Les fichiers RDS et DTR contiennent cette description.

Le fichier RDS (*Rectangle Data Structure*) est utilisé par les outils d'Alliance pour effectuer la transformation d'un dessin symbolique en dessin réel. Ce fichier définit les règles de cette transformation pour une technologie donnée. On y trouve notamment la valeur du pas de la grille physique δ et du pas de la grille symbolique λ en nanomètres, qui sont utiles dans notre approche. C'est pourquoi une lecture du fichier RDS est effectuée automatiquement en début de flot par CAIRO+. Les variables globales `RDS_LAMBDA` et `RDS_PHYSICAL_GRID` qui représentent respectivement les valeurs de λ et de δ sont alors disponibles pour la conception. Le fichier RDS est également lu en fin de flot afin de transformer des fils de routage symboliques du dessin en rectangles réels. Cependant, les règles de dessin telles que le fondeur les impose n'y apparaissent pas et ce fichier n'est pas suffisant pour réaliser le dessin des masques à la précision nécessaire aux circuits analogiques.

Une représentation des règles fondeur qui puisse être exploitée par la DAPI et la MAPI est assez difficile à mettre en œuvre. Nous avons remarqué en effet qu'au delà de la valeur de ces règles, c'est leur nature qui peut largement varier d'un procédé de fabrication à un autre. Pour pouvoir gérer l'hétérogénéité des jeux de règles fournis par les fabricants, nous nous sommes orientés vers une représentation dont la syntaxe est volontairement libre. Celle-ci est incarnée par le fichier DTR (*Device Technological Rules*) qui représente les informations sous forme de tableaux. Comme

le montre la figure 5.34, ceux-ci contiennent un ensemble d'entrées auxquelles sont associées une ou plusieurs valeurs. Dans notre cas, une entrée représentera une règle de dessin. Pour une entrée

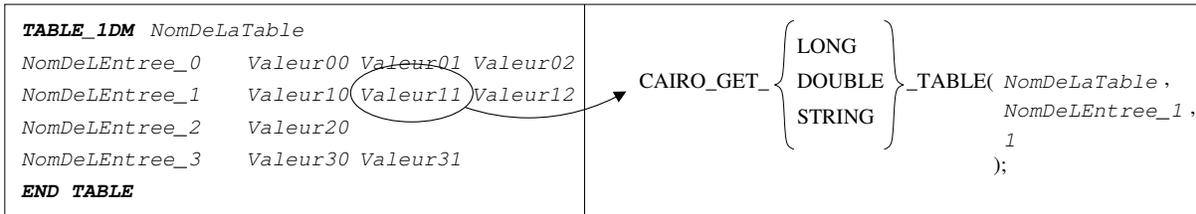


FIG. 5.34 : Fichier DTR : Syntaxe et accès.

donnée, les différentes valeurs sont indexées de façon à ce que la première valeur exprimée reçoive l'index 0, la suivante l'index 1 et ainsi de suite. Chacune de ces valeurs peut correspondre à un type d'information quelconque : textuel, entier ou flottant. On dispose de trois fonctions d'accès qui permettent de récupérer une valeur de chacun des types possibles. Ces fonctions prennent en paramètre le nom de la table, le nom de l'entrée, et l'index de la valeur désirée. Bien sûr, il faut impérativement utiliser la fonction qui correspond au type de l'information récupérée. Pour assurer que cette contrainte soit respectée, et pour favoriser la plus grande liberté, il faudrait que la même personne écrive le fichier DTR, et le lise. C'est pourquoi une première idée a été de faire apparaître dans le fichier DTR une table par dispositif élémentaire existant. Ainsi, le concepteur d'un nouveau dispositif élémentaire enrichit le fichier DTR avec une table contenant toutes les règles dont il a besoin. Il connaît parfaitement la teneur de cette table et peut accéder aux valeurs sans risquer un conflit de type. En pratique, cette organisation impliquait une répétition excessive des règles les plus usuelles telles que les dimensions minimales d'un transistor, les dimensions d'un via, etc. Nous avons donc modifié l'organisation du fichier DTR de façon à factoriser ce type de règles. Désormais, le fichier DTR contient au minimum trois tables :

1. Une table **RULES** : Elle contient les règles de dessin les plus utilisées. Nous avons rangé dans cette catégorie les règles qui concernent le dessin des transistors et celles qui régissent les fils de routage. En effet, toutes les technologies utilisent les mêmes couches pour le dessin d'un transistor et imposent à peu près le même ensemble de règles en ce qui le concerne. De même, les fils de routage sont soumis à des règles de même nature dans toutes les technologies.

Puisque les règles de la table **RULES** pourront être utilisées par tous les générateurs de motifs ou de dispositifs élémentaires, il est nécessaire de définir une nomenclature pour les entrées de cette table. Cette nomenclature s'applique à la nature des règles, et aux couches concernées. L'environnement Alliance avait déjà défini une nomenclature pour les couches, que nous avons reprise. Quant à la nature des règles, nous avons défini, pour les cinq types de règles qui peuvent appartenir à la table **RULES**, la nomenclature suivante :

- **Dimensions minimales et maximales d'un transistor :**

Bien que la grille d'un transistor soit en polysilicium, les règles qui la régissent sont différentes de celles qui régissent le polysilicium lorsqu'il est utilisé dans un autre contexte.

Nous avons donc défini les règles suivantes :

`RW_TRANS` et `XW_TRANS` sont les largeurs minimale et maximale d'un transistor.

`RL_TRANS` et `XL_TRANS` sont les longueurs minimale et maximale d'un transistor.

- **Les règles définissant la largeur minimale d'un rectangle d'une couche :** `RW_couche`.
 - **Les règles définissant la distance minimale entre deux rectangles d'une même couche :** `RD_couche`.
 - **Les règles définissant la distance minimale entre deux rectangles de couches différentes :** `RD_couche1_couche2`. Par convention, le nom de ce type de règles fera intervenir les couches 1 et 2 dans l'ordre alphabétique.
 - **Les règles définissant l'extension minimale d'une couche 1 autour d'une couche 2 :** `RE_couche1_couche2`.
2. Une table `LAYERS` : Elle définit la couche de tous les rectangles de tous les motifs dans la technologie concernée. Elle définit également les couches de routage utilisées par tous les dispositifs élémentaires. Nous verrons à la section 6.2.10 que les rectangles constituant un motif sont référencés par un numéro. Les entrées de la table `LAYERS` qui correspondent à ces rectangles sont alors la concaténation du nom du motif auquel appartient le rectangle, et du numéro du rectangle dans ce motif (par exemple : `MOTIF1_12`). Pour les couches de routage, les entrées sont la concaténation du nom du dispositif élémentaire concerné, et d'un nom choisi par le concepteur de ce dispositif élémentaire (par exemple : `DISPOSITIF1_ROUTING`). Lorsqu'un concepteur crée un nouveau dispositif élémentaire, il ajoute donc un ensemble d'entrées dans cette table correspondant aux rectangles des motifs qu'il introduit et aux couches de routage dont il a besoin. Les valeurs de la table `LAYERS` sont, quant à elles, des couches Alliance. Le fichier RDS contient alors la correspondance entre les couches Alliance et les couches aux formats gds et ci.f. Comme nous l'avons vu, il est également possible de donner la valeur "NO_LAYER" à certaines entrées de la table `LAYERS`. On inhibe dans ce cas la génération des rectangles correspondant dans la technologie.
 3. Une table `ELECTRICAL` : Elle contient des caractéristiques électriques du procédé, et notamment celles liées aux composants passifs. On trouvera ici des informations telles que les capacités surfacique et périmétrique des métaux par rapport au substrat ou la résistance par carré d'une résistance polysilicium.

D'un autre côté, pour conserver la grande liberté associée au fichier DTR, chaque concepteur peut ajouter une table spécifique au dispositif élémentaire qu'il réalise. Dans ce cas, la syntaxe est complètement libre puisqu'il sera le seul à lire les données présentes dans cette table. Pour illustrer l'intérêt de ces tables spécifiques, considérons le dessin de capacités MIM. Nous avons déjà précisé que les couches utilisées pour les dessiner dépendaient de la technologie. En conséquence, le type de via permettant de connecter l'armature haute dépend également de la technologie. Ainsi, pour représenter la règle qui définit la largeur de ces vias, il faut utiliser un nom de règle qui fait référence à une couche générique : `RW_MIMTOPVIA` par exemple. Ici, `MIMTOPVIA` est une couche générique correspondant à un via de connexion de l'armature haute et la valeur de cette règle dépendra du type de via effectivement utilisé dans la technologie décrite par le fichier DTR. En

conclusion, les tables spécifiques permettent de représenter tout type de règles dont le concepteur peut avoir besoin pendant la conception d'un générateur de dispositifs élémentaires, et qui ne sont pas qualifiées pour apparaître dans la table `RULES`.

Les différentes MAPI et DAPI font constamment appel aux règles présentes dans le fichier DTR. Ces appels sont contenus dans le code de ces API et ne peuvent être annulés. Puisque la nature des règles peut varier d'une technologie à une autre, il est possible que la MAPI ou la DAPI fasse appel à une règle inexistante dans le procédé de fabrication visé. Pour résoudre ce problème, nous proposons de faire apparaître dans tous les fichiers DTR toutes les règles pouvant intervenir dans toutes les technologies. On peut alors inhiber l'influence d'une règle définissant une valeur minimale en lui attribuant une valeur nulle, ou inhiber celle d'une règle définissant une valeur maximale en lui attribuant une valeur surréaliste.

Finalement, la migration technologique est permise dans *CAIRO+* grâce à une représentation adaptée de la technologie : le fichier DTR, et à la définition d'une interface entre cette technologie et le dispositif élémentaire : la DAPI et la MAPI. La figure 5.35 illustre cette organisation.

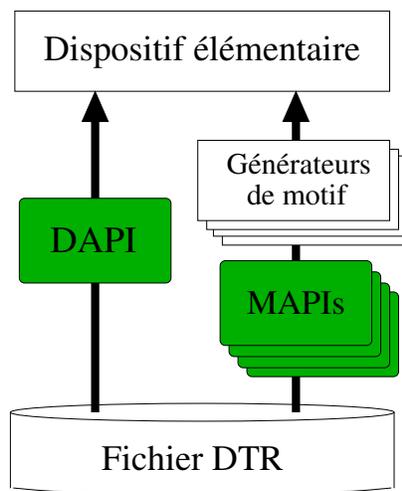


FIG. 5.35 : Interface entre le dispositif élémentaire et le procédé de fabrication.

5.6.6 Introduction d'un nouveau procédé de fabrication

Lorsque l'on souhaite effectuer la génération d'un circuit sur un nouveau procédé de fabrication, il est nécessaire de disposer des caractéristiques de ce procédé. Sous *CAIRO+*, trois fichiers permettent de représenter ces caractéristiques :

- Un fichier au format `Spice`, directement fourni par le fondeur, donne les paramètres du modèle `Bsim3v3`.
- Le fichier `RDS` qui donne le pas de la grille physique et la valeur de λ . Ce fichier est généré automatiquement à partir des règles symboliques et d'une représentation partielle des règles

fondeur. Cette génération automatique prend notamment en charge le calcul de λ .

- Le fichier DTR qu'il est nécessaire d'écrire pour ce nouveau procédé.

L'écriture d'un nouveau fichier DTR est une tâche relativement aisée puisque, comme nous l'avons souligné, tous les fichiers DTR représentant tous les procédés de fabrication doivent présenter les mêmes entrées. Dans le cas idéal, il suffit donc, pour introduire une nouvelle technologie, de recopier un fichier DTR existant et de mettre à jour la valeur de toutes les règles en inhibant celles qui n'interviennent pas dans cette nouvelle technologie.

Cependant, il est très fréquent qu'un nouveau procédé de fabrication introduise de nouvelles règles. Dans ce cas, ces nouvelles règles doivent être ajoutées à tous les fichiers DTR. Dans les fichiers DTR qui existaient déjà, elles n'intervenaient pas jusqu'ici et devront donc être inhibées. Par ailleurs, la MAPI de tous les motifs qui sont affectés par ces nouvelles règles doit être mise à jour afin qu'elle puisse les prendre en compte. La DAPI des dispositifs élémentaires affectés est également à mettre à jour.

D'un autre côté, les motifs sont spécifiés de façon à prendre en compte tous les rectangles pouvant éventuellement intervenir dans leur dessin des masques quelle que soit la technologie. Lorsque que l'on introduit une nouvelle technologie, il est possible que celle-ci implique l'ajout d'un rectangle qui n'avait pas été prévu dans le dessin de certains motifs. Dans ce cas, il faut tout d'abord ajouter les cinq fonctions permettant la génération de ce rectangle dans la MAPI de tous les motifs concernés. Il faut également mettre à jour la table `LAYERS` de tous les fichiers DTR, en y incorporant la définition de la couche du rectangle pour tous les motifs qui le dessinent. Dans les fichiers DTR déjà existants, la couche du nouveau rectangle sera définie à "`NO_LAYER`" pour qu'il ne soit pas généré.

Ces opérations quelque peu fastidieuses sont le coût de la migration technologique ; un coût finalement peu élevé dans la mesure où l'organisation DTR/MAPI/DAPI localise précisément les changements.

5.6.7 Conception d'un nouveau dispositif élémentaire

La conception d'un nouveau dispositif élémentaire peut également avoir un impact sur la représentation du procédé de fabrication, et plus précisément, sur le fichier DTR. Deux cas sont possibles en effet :

- Le nouveau dispositif utilise des générateurs de motifs déjà existants. Dans ce cas, les MAPI correspondantes existent déjà et ne sont pas à réécrire. En revanche, la DAPI étant spécifique à un dispositif élémentaire, il est nécessaire de l'écrire pour le nouveau dispositif. Celle-ci peut éventuellement s'appuyer sur des règles qui n'intervenaient pas jusqu'ici. On introduit alors ces règles en respectant la méthode décrite à la section précédente.
- Le nouveau dispositif introduit de nouveaux motifs : il faut notamment écrire une MAPI pour chacun des nouveaux générateurs de motifs. Celle-ci peut bien sûr faire intervenir de nouvelles règles, qu'il faut introduire dans tous les fichiers DTR. De plus, la définition des couches de tous les rectangles constituant tous les nouveaux motifs doit être ajoutée à tous les fichiers DTR. Enfin, il faut bien évidemment écrire la DAPI du nouveau dispositif.

Bien que ces opérations de mise à jour soient relativement lourdes, on s'en affranchit la plupart du temps si les générateurs de motifs ont été soigneusement conçus. Nous verrons par exemple au chapitre 7 que les générateurs de motifs prenant en charge les doigts de transistor ont été conçus de façon à pouvoir être réutilisés dans n'importe quel dispositif élémentaire à transistor : miroir de courant, paire différentielle, matrice de transistors, etc...

5.7 Procédure de validation

Pour vérifier le bon fonctionnement d'un générateur de dispositifs élémentaires, il faut, dans toutes les technologies supportées par CAIRO+ :

- Vérifier que les procédures de dimensionnement internes sont valides et conduisent à des dimensions électriques qui permettent d'atteindre les spécifications.
- Vérifier que le dessin des masques obtenu respecte les règles de dessin imposées par la technologie et qu'il réalise effectivement la vue structurelle dimensionnée du dispositif étudié.

Au niveau des générateurs de dispositifs élémentaires, la validation des procédures de dimensionnement est particulièrement nécessaire pour les générateurs de transistors. Pour ce type de générateur, ces procédures sont en effet chargées de transformer des spécifications telles qu'un courant de drain I_{DS} ou une transconductance G_m en dimensions W et L pour le transistor. À l'inverse, les paramètres électriques d'entrée des générateurs de composants passifs comme la capacité ou la résistance sont directement les dimensions électriques de ces composants : la valeur C d'une capacité ou R d'une résistance.

Pour valider le résultat des procédures de dimensionnement, on dispose d'une *netlist* au format *spice* que l'on peut soumettre à un simulateur. Si les résultats de la simulation correspondent aux spécifications qui avaient été imposées aux procédures de dimensionnement, on peut considérer qu'elles sont valides. Il faut cependant effectuer cette simulation pour plusieurs jeux de spécifications.

Lorsque l'on est sûr que les procédures de dimensionnement aboutissent à des dimensions correctes, il faut vérifier que le dessin des masques réalise bien les composants voulus, avec les dimensions souhaitées. Cette vérification se fait dans un environnement externe de vérification physique paramétré par les informations technologiques (on utilise typiquement l'environnement Cadence paramétré par le *design kit* du fondeur).

Pour toutes les technologies supportées et pour différents jeux de spécifications, on commence par vérifier que le dessin généré ne viole aucune règle de dessin. On le soumet pour cela à un outil de vérification des règles de dessin (*DRC*) qui s'appuie sur la liste exhaustive des règles imposées par le fondeur. Si certaines règles sont violées, c'est qu'elles ont été mal prises en compte dans les motifs qui doivent les respecter. En conséquence, il faut corriger les calculs effectués par la MAPI de ces motifs et éventuellement ajouter des règles qui avaient été omises dans les fichiers DTR.

Lorsque les règles de dessin sont respectées, il faut également s'assurer que le dessin correspond effectivement aux composants que l'on cherche à obtenir, et que les dimensions électriques de ces composants sont effectivement réalisées. On procède pour cela à une extraction, et à une comparaison de *netlists* (*LVS*). L'extraction révèle le schéma électrique correspondant au dessin des masques. En comparant ce schéma avec la *netlist* issue de *CAIRO+*, on peut finalement vérifier que l'on a bien réalisé les composants voulus aux bonnes dimensions. Si l'extracteur ne reconnaît pas certains composants, il est probable que des rectangles nécessaires à leur réalisation aient été oubliés. Si les dimensions obtenues ne correspondent pas à celles désirées, il faut revoir :

- La fonction `CAIRO_GENERATE` et en particulier le calcul des dimensions des motifs en fonction de l'indice de forme. Par exemple, le calcul de la largeur W_f d'un doigt de transistor en fonction du nombre de repliements M peut être mauvais.
- La MAPI de chacun des motifs qui peut rendre de mauvaises dimensions géométriques pour les rectangles significatifs comme la grille d'un transistor ou l'armature haute d'une capacité.

Finalement, la procédure de validation d'un générateur de dispositifs élémentaires est illustrée sur la figure 5.36.

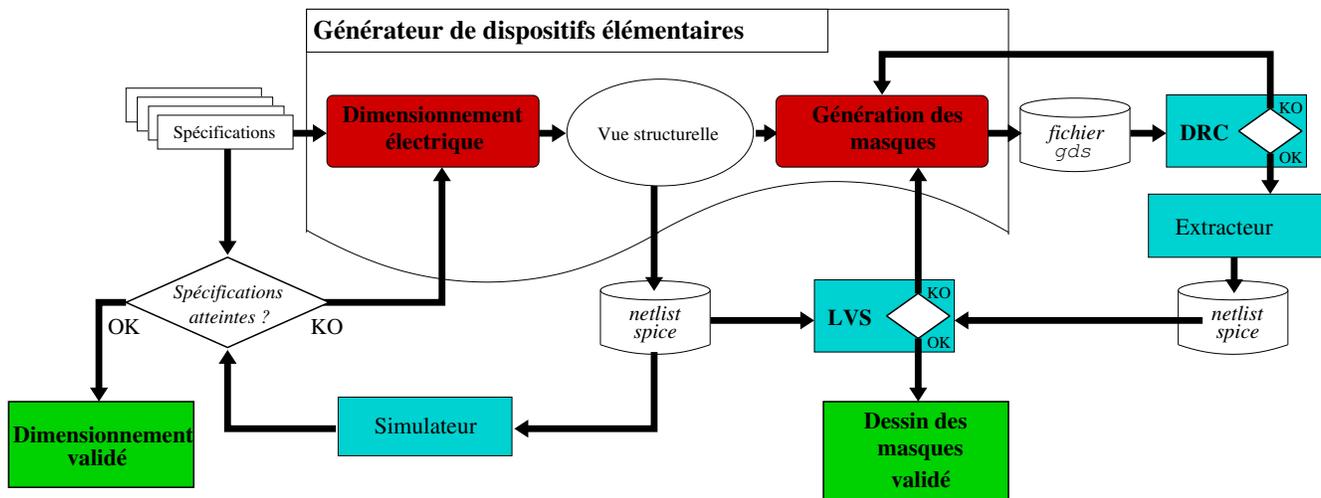


FIG. 5.36 : Procédure de validation d'un générateur de dispositifs élémentaires.

5.8 Conclusion

Dans ce chapitre, nous avons vu que la prise en compte de l'ensemble des contraintes liées à la conception de circuits analogiques faisait des générateurs de dispositifs élémentaires des entités très complexes. L'organisation de ces générateurs autour de motifs, de MAPIs et d'une DAPI permet cependant de localiser précisément chaque niveau d'expertise. Elle a également permis la définition d'une méthodologie systématique de conception d'un générateur de dispositifs élémentaires que nous allons présenter dans le chapitre suivant.

Chapitre 6

Méthodologie de conception d'un générateur de dispositifs élémentaires

6.1 Introduction

Notre travail a abouti à deux résultats :

- L'élaboration d'une méthodologie de conception de générateurs de dispositifs élémentaires pour *Cairo+*. Celle-ci s'appuie sur les concepts que nous avons présentés dans le chapitre précédent.
- Le développement d'une bibliothèque de générateurs de dispositifs élémentaires permettant de couvrir un large spectre de circuits analogiques.

Ce chapitre présente la méthodologie définie et détaille l'ensemble des étapes de la conception d'un générateur de dispositifs élémentaires. Grâce à cette méthodologie, nous avons pu développer plusieurs générateurs présentés dans le chapitre 7. Ceux-ci permettent la conception automatisée de circuits pour différents types d'applications. Quelques exemples d'applications seront donnés dans le chapitre 8. Pour élargir la gamme d'applications que peuvent atteindre les circuits conçus sous *CAIRO+*, la méthodologie permettra à chacun de réaliser de nouveaux générateurs de dispositifs élémentaires.

6.2 Une méthodologie en quatorze étapes

Pour concevoir un générateur de dispositifs élémentaires, nous proposons une démarche en quatorze étapes. En nous appuyant sur l'exemple du dispositif *BICAPA* qui réalise deux capacités MIM appariées, nous allons, dans cette section, détailler chacune de ces étapes.

Dans notre méthodologie, le premier objectif est d'identifier quels seront les paramètres du générateur. Ceci n'est cependant possible que si l'on a précisément défini l'architecture de la vue structurelle interne et celle du dessin des masques. Les deux premières étapes cherchent donc à identifier ces deux architectures.

6.2.1 Première étape : identification de la vue structurelle

Pour définir l'architecture de la vue structurelle interne, il faut commencer par clairement identifier le schéma électrique que l'on cherche à réaliser. Ce schéma se résume souvent à un unique composant électronique, mais nous avons vu qu'il peut faire intervenir plusieurs composants liés par une contrainte d'appariement. Avec le générateur *BICAPA*, on cherche à réaliser deux capacités MIM qui sont liées par un rapport rationnel R . Il s'agit d'un dispositif élémentaire multi-composants typique et les deux capacités apparaissent dans le schéma électrique. Celui-ci est représenté sur la figure 6.1.

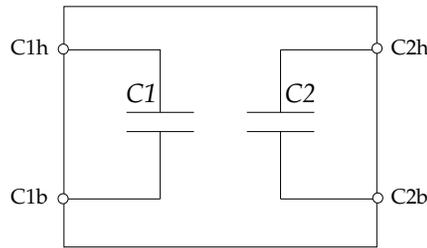


FIG. 6.1 : Schéma électrique du dispositif *BICAPA* : $C2 = R.C1, R \in \mathbb{Q}$.

En conséquence, la vue structurelle de notre générateurinstanciera deux capacités structurelles et décrira leurs interconnexions. Elle devra également instancier les capacités parasites de routage que l'on souhaite prendre en compte. En général, on suppose que le routage pourra introduire des parasites entre tous les nœuds du schéma électrique. Cependant, conformément à ce que nous avons vu au chapitre 5, les capacités parasites de routage qui interviennent entre les deux armatures d'une capacité MIM sont directement ajoutées à la valeur nominale et n'apparaissent pas dans la vue structurelle. Ainsi, la vue structurelle interne du générateur *BICAPA* sera celle illustrée sur la figure 6.2.

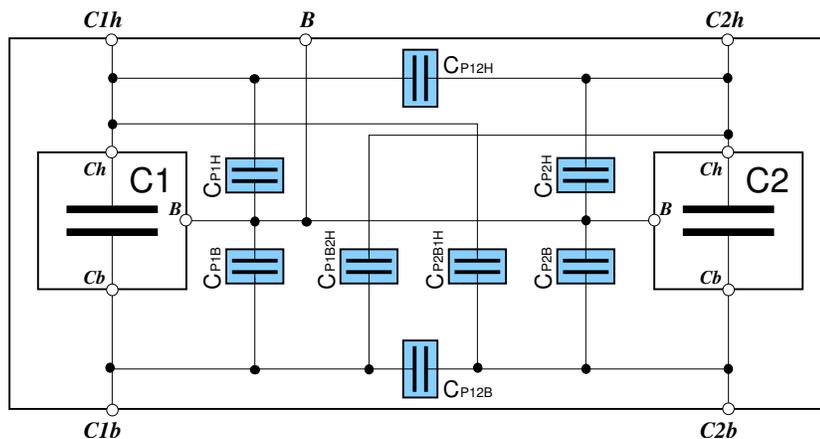


FIG. 6.2 : Vue structurelle du dispositif *BICAPA*.

6.2.2 Etape 2 : choix d'un plan de masse pour le dessin des masques

Lorsque la vue structurelle a été définie, il faut déterminer l'architecture du dessin des masques, c'est-à-dire établir un plan de masse. Le choix d'un plan de masse doit être fait de façon particulièrement réfléchi et nécessite une exploration approfondie de la littérature. Il faut en effet que celui-ci présente un dessin optimisé qui minimise notamment les erreurs d'appariement. De plus, nous avons vu qu'un dispositif élémentaire doit être déformable et il faut envisager un plan de masse capable de présenter différents facteurs de formes. Enfin, et surtout, il est impératif que le plan de masse adopté soit valable pour tous les procédés de fabrication envisagés. En ce qui concerne le générateur BICAPA, la réalisation de capacités MIM peut être soumise à des contraintes différentes selon le procédé. Certaines technologies interdisent, par exemple, la configuration de la figure 6.3-a où la connexion de l'armature basse est assurée par un via situé sous cette armature. Dans ces technologies, seule la configuration de la figure 6.3-b est autorisée. Il faut donc porter

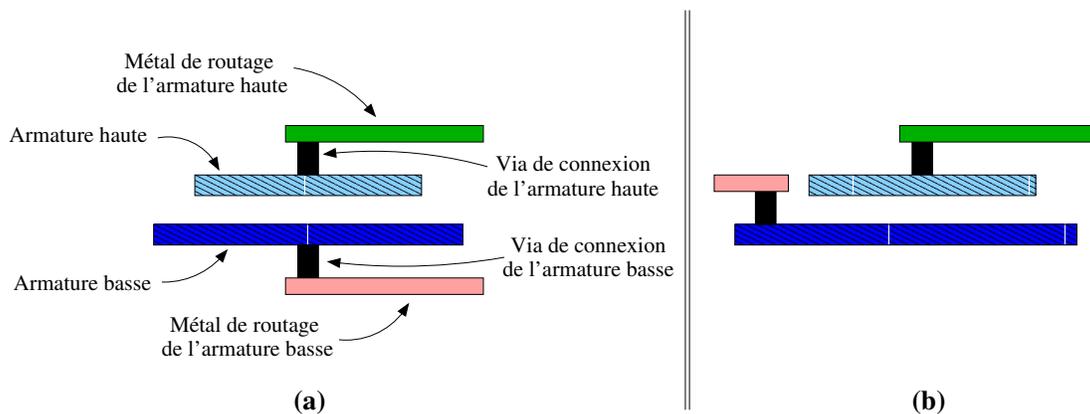


FIG. 6.3 : Deux configurations de dessin des masques d'une capacité MIM.

une attention particulière à la portabilité du plan de masse que l'on choisit.

En ce qui nous concerne, nous souhaitons optimiser l'appariement des deux capacités. Dans le chapitre 5, nous avons vu qu'un bon appariement pouvait être obtenu en réalisant chaque capacité sous forme de capacités élémentaires, que l'on mélange dans une matrice afin d'obtenir une géométrie centrée. Le plan de masse que l'on va adopter représentera donc une matrice de capacités élémentaires. La déformation de cette matrice sera possible en modifiant son nombre de lignes et son nombre de colonnes.

Finalement, nous choisirons ici le plan de masse présenté sur la figure 6.4 qui illustre deux capacités liées par un rapport R égal à $9/7$ et qui est compatible avec l'ensemble des technologies étudiées. Pour favoriser l'appariement, des capacités fictives sont introduites autour de la matrice et les capacités élémentaires sont mélangées en suivant un algorithme simple :

- Sur chaque ligne de la matrice, on alterne les capacités élémentaires de $C1$ et de $C2$.

- On construit la matrice en alternant des lignes qui commencent respectivement par une capacité élémentaire de C1 et par une capacité élémentaire de C2.

Le routage des capacités élémentaires, quant à lui, est effectué dans des canaux existants entre chaque ligne et chaque colonne de la matrice. Les canaux horizontaux sont dédiés au routage des armatures basses, les canaux verticaux à celui des armature hautes. Enfin, un métal sera adopté pour les fils de routage horizontaux, un autre pour les fils de routage verticaux.

Il est très important de toujours représenter le plan de masse choisi pour différentes formes. On révèle alors en général des contraintes nouvelles. Sur la figure 6.4-b, on voit par exemple qu'en modifiant la configuration, on peut aboutir à une matrice qui n'est pas pleine. En effet, pour placer seize capacités élémentaires sur trois lignes, il faut une matrice de six colonnes. Il y a alors deux cellules de la matrice qui ne sont pas comblées. Afin de ne pas limiter le nombre de formes du dispositif BICAPA, nous autoriserons ce genre de configuration. Par contre, pour conserver une forme globale rectangulaire, ces cellules seront comblées par des capacités fictives.

Dans ce genre de situation, l'algorithme de placement que nous avons défini ne conduit pas à une géométrie parfaitement centrée. Le mélange des capacités élémentaires reste cependant convenable et rapproche suffisamment les deux barycentres.

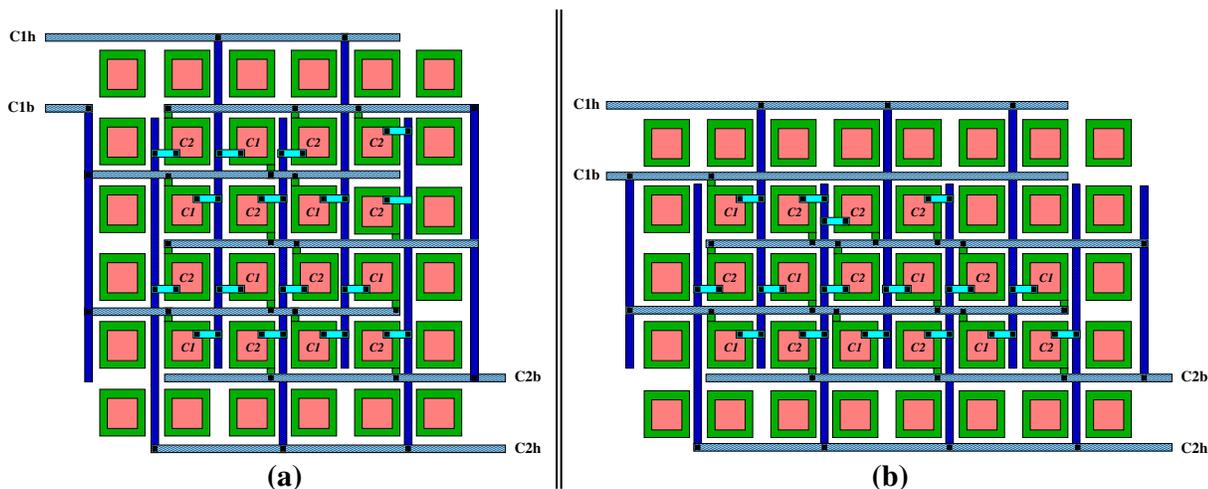


FIG. 6.4 : Plans de masse pour le dispositif BICAPA : (a) $R = 9/7$ dans une matrice 4×4 - (b) $R = 9/7$ dans une matrice 6×3 .

Dès que l'on dispose du plan de masse, il faut définir quel sera l'indice de forme. Il s'agit du paramètre sur lequel on agira pour déformer le dispositif. Ici, puisque le nombre de capacités élémentaires est donné, on peut choisir la forme de la matrice en modifiant son nombre de lignes. Le nombre de lignes $nb1$ de la matrice sera l'indice de forme du générateur BICAPA. Il est alors possible qu'une modification du nombre de ligne ne modifie pas la largeur de la matrice. Par exemple, s'il l'on considère une matrice de 19 capacités élémentaires et de 5 lignes, il faut 4 colonnes pour pouvoir placer toutes les capacités élémentaires. Si cette même matrice présente 6 lignes, il lui

faut également 4 colonnes. Dans de tels cas, *CAIRO+* ne retient que les formes d'encombrement minimal et, dans notre exemple, la configuration 4×6 n'est pas prise en compte dans la fonction de forme.

6.2.3 Etape 3 : identification des paramètres du dispositif

A partir du schéma électrique et du plan de masse, on va pouvoir définir les paramètres du générateur. Nous avons vu à la section 5.6.1 qu'il y a quatre types de paramètres pour un générateur de dispositifs élémentaires :

- Les paramètres structurels qui modifient la vue structurelle interne. Ils sont passés à la fonction `CAIRO_CREATE`.
- Les paramètres électriques d'entrée qui servent aux procédures de dimensionnement décrites dans la fonction `CAIRO_DSES`.
- Les paramètres électriques de sortie qui sont calculés par les procédures de la vue comportementale. Ces procédures sont également décrites dans la fonction `CAIRO_DSES`.
- Les paramètres de réalisation qui sont spécifiés dans un fichier de réalisation, et qui permettent de modifier le dessin des masques généré par la fonction `CAIRO_GENERATE`.

Le générateur `BICAPA` ne proposera pas de paramètre structurel. Il réalise exclusivement le schéma électrique de la figure 6.1. Il ne sera donc pas possible, par exemple, de modifier le nombre de capacités réalisées.

Pour les composants passifs, les paramètres électriques d'entrée sont généralement les dimensions électriques des composants réalisés. Ici, on pourrait ainsi envisager deux paramètres électriques d'entrée : la valeur C_1 de la capacité 1 et la valeur C_2 de la capacité 2. Cependant, puisque l'on réalise une matrice de capacités élémentaires et que seuls les rapports fractionnels sont autorisés, on préférera proposer les paramètres électriques d'entrée suivants :

- C_{e1em} : Valeur de la capacité élémentaire.
- $n1$: Nombre de capacités élémentaires qui composent la capacité 1.
- $n2$: Nombre de capacités élémentaires qui composent la capacité 2.

Cet ensemble de paramètres assure le respect d'un rapport fractionnel. Il permet par ailleurs de laisser au concepteur du module appelant, le choix de la valeur de la capacité élémentaire et du rapport $n1/n2$ pour un rapport R visé. Il fera ce choix en fonction des contraintes de surface ou d'appariement qu'il souhaite respecter. En effet, rappelons qu'un plus grand nombre de capacités élémentaires permet d'obtenir plus facilement une géométrie centrée, mais augmente la surface globale de la matrice. D'un autre côté, de grandes capacités élémentaires minimisent les erreurs aléatoires d'appariement, mais impliquent un nombre moins important de capacités unitaires pour une valeur de capacité totale donnée.

Les paramètres électriques de sortie sont renvoyés au module de niveau supérieur pendant la phase d'estimation des performances. Pour un composant passif, il faut pouvoir renvoyer

les dimensions électriques effectives des composants réalisés, ainsi que les capacités parasites de routage que le dessin des masques fait apparaître. Le générateur BICAPA proposera ainsi les paramètres électriques de sortie suivants :

- C1 : Valeur effective de la capacité C1.
- C2 : Valeur effective de la capacité C2.
- CP1H : Capacité parasite de routage entre l'armature haute de C1 et le substrat.
- CP1B : Capacité parasite de routage entre l'armature basse de C1 et le substrat.
- CP2H : Capacité parasite de routage entre l'armature haute de C2 et le substrat.
- CP2B : Capacité parasite de routage entre l'armature basse de C2 et le substrat.
- CP12H : Capacité parasite de routage entre les armatures hautes de C1 et de C2.
- CP12B : Capacité parasite de routage entre les armatures basses de C1 et de C2.
- CP1B2H : Capacité parasite de routage entre l'armature basse de C1 et l'armature haute de C2.
- CP2B1H : Capacité parasite de routage entre l'armature basse de C2 et l'armature haute de C1.

Grâce à ces informations, le module de niveau supérieur pourra déterminer précisément son propre comportement.

Enfin, le concepteur du générateur de dispositifs élémentaires est libre de proposer ou pas des paramètres de réalisation. S'il décide d'en proposer, il doit définir une valeur par défaut pour chacun d'entre eux. En effet, les paramètres de réalisation sont optionnels. Pour le générateur BICAPA, nous proposerons par exemple la présence de capacités fictives aux bords de la matrice comme paramètre de réalisation. En effet, si l'appariement n'est pas capital, on peut réduire la surface de la matrice en enlevant ces capacités fictives. Par défaut, elles seront cependant présentes.

Il faut alors choisir une syntaxe qui permette de représenter la valeur de ce paramètre dans le fichier de réalisation. On propose que le paramètre soit nommé `dummys` et accepte deux valeurs Y et N, indiquant respectivement la présence et l'absence des capacités fictives. Le fichier de réalisation ne contiendra donc qu'une ligne du type :

```
1  dummys N
```

En cas de valeur incorrecte, différente de Y ou N, la réalisation par défaut sera effectuée et des capacités fictives seront placées autour de la matrice.

6.2.4 Etape 4 : identification des procédures de dimensionnement

Dès lors que l'on dispose de l'ensemble des paramètres du générateur, on va s'attaquer à l'écriture des quatre fonctions constructrices. La première d'entre elles, `CAIRO_CREATE`, doit construire la vue structurelle, initialiser une vue physique vide, et déclarer les paramètres électriques et les procédures qui apparaîtront dans la fonction `CAIRO_DSES`. Pour l'écrire, il nous faut donc d'abord identifier les procédures de dimensionnement et celles d'estimation des

performances proposées par le générateur. Les procédures de dimensionnement prennent en charge la traduction des spécifications en dimensions électriques pour les composants, et vérifient la validité des dimensions obtenues. Les procédures d'estimation des performances déterminent le comportement du dispositif pour des dimensions données.

Le cas des composants passifs est particulier car les spécifications sont les dimensions électriques des composants. Pour ce type de composants, les procédures de dimensionnement se contentent de vérifier la faisabilité des dimensions spécifiées. Le générateur `BICAPA` ne proposera donc qu'une unique procédure de dimensionnement, appelée `direct_size` qui vérifiera la faisabilité de la capacité élémentaire `C_elem`. Elle transmettra ensuite ses paramètres d'entrée `C_elem`, `n1` et `n2` à la fonction `CAIRO_SHAPE` qui pourra effectuer la génération du dessin.

Pour les générateurs de composants passifs, les procédures d'estimation des performances permettent simplement de renvoyer au module de niveau supérieur les paramètres électriques de sortie. Ainsi, le générateur `BICAPA` proposera les procédures suivantes :

- `get_C1`
- `get_C2`
- `get_CP1H`
- `get_CP1B`
- `get_CP2H`
- `get_CP2B`
- `get_CP12H`
- `get_CP12B`
- `get_CP1B2H`
- `get_CP2B1H`

6.2.5 Etape 5 : écriture de la fonction `CAIRO_CREATE`

A ce stade de notre méthodologie, nous disposons de toutes les informations nécessaires à l'écriture de la fonction `CAIRO_CREATE`. Il s'agit d'y décrire la vue structurelle en tenant compte des paramètres structurels, de créer une vue physique vide et enfin de déclarer les paramètres électriques ainsi que les procédures de dimensionnement et d'estimation des performances. Lorsqu'il y en a, les paramètres structurels doivent également être transmis à la fonction `CAIRO_SHAPE` afin qu'ils soient pris en compte pendant la génération du dessin des masques. Le langage `CAIRO+` permet d'effectuer aisément ces opérations.

6.2.6 Etape 6 : écriture de la `DAPI` et mise à jour du fichier `DTR`

Comme nous l'avons vu au chapitre précédent, les trois fonctions constructrices qu'il reste à écrire s'appuient sur des informations technologiques :

- Les procédures de `CAIRO_DSES` doivent aboutir à des dimensions qui sont physiquement réalisables.
- Le nombre de formes déterminé par `CAIRO_SHAPE` dépend en général de règles technologiques.
- Le routage pris en charge par `CAIRO_GENERATE` se fait dans des couches précisées dans le fichier DTR.

C'est la DAPI qui fournit les informations technologiques nécessaires à ces trois fonctions. Il va maintenant falloir spécifier et écrire les fonctions de la DAPI. Ceci commence par bien identifier les règles technologiques dont elles ont besoin et les mettre dans le fichier DTR si elles n'y apparaissent pas déjà.

6.2.6.1 Fonctions de la DAPI requises par `CAIRO_DSES`

Lors de la génération de composants passifs, afin de vérifier que les dimensions obtenues sont valides, on fait appel à la calculette pour déterminer les dimensions électriques minimales et maximales. Par exemple, en donnant à la calculette de capacités MIM la largeur minimale de l'armature haute et les capacités surfacique et périmétrique, elle peut calculer la capacité réalisable la plus petite. Dans les procédures de `CAIRO_DSES`, il faut donc pouvoir disposer de certaines informations technologiques, afin de les transmettre à la calculette. Ces informations sont fournies par la DAPI qui, dans le cas du dispositif `BICAPA`, proposera les fonctions suivantes :

- `double get_rw_mim()` : renvoie la largeur minimale de l'armature haute en micron.
- `double get_xw_mim()` : renvoie la largeur maximale de l'armature haute en micron.
- `double get_cs_mim()` : renvoie la valeur de la capacité surfacique des capacités MIM.
- `double get_cp_mim()` : renvoie la valeur de la capacité périmétrique des capacités MIM.

Ces fonctions ne se résument pas forcément à la simple lecture d'informations dans le fichier DTR. En effet, lorsque l'on observe le plan de masse plus en détails (fig. 6.4), on s'aperçoit que le respect d'un grand nombre de règles de dessin dépend de la largeur des armatures. Par exemple, les rectangles qui permettent d'effectuer la liaison entre les armatures et les pistes de routage, que nous avons appelés les "prolongements" des armatures, doivent respecter une distance minimale entre eux. L'armature haute doit être suffisamment large pour que cette distance soit respectée. Il en va de même pour les vias aux extrémités des prolongements. Finalement, pour déterminer la largeur minimale de l'armature haute, il faut prendre en compte l'ensemble des règles présentées sur la figure 6.5 :

- ① `RW_TOPMIM` : Largeur minimale de l'armature haute.
- ② `RW_BOTMIM` : Largeur minimale de l'armature basse.
- ③ `RW_BOTPROLMIM` : Largeur minimale du prolongement de l'armature basse. Bien que ce prolongement soit réalisé dans la même couche que l'armature elle-même, il est possible que, dans la mesure on ne lui superpose pas d'armature haute, il soit contraint par une largeur minimale différente de celle de l'armature. C'est pourquoi on envisage une règle différente.
- ④ `RE_BOTMIM_TOPMIM` : Extension minimale de l'armature basse autour de l'armature haute.

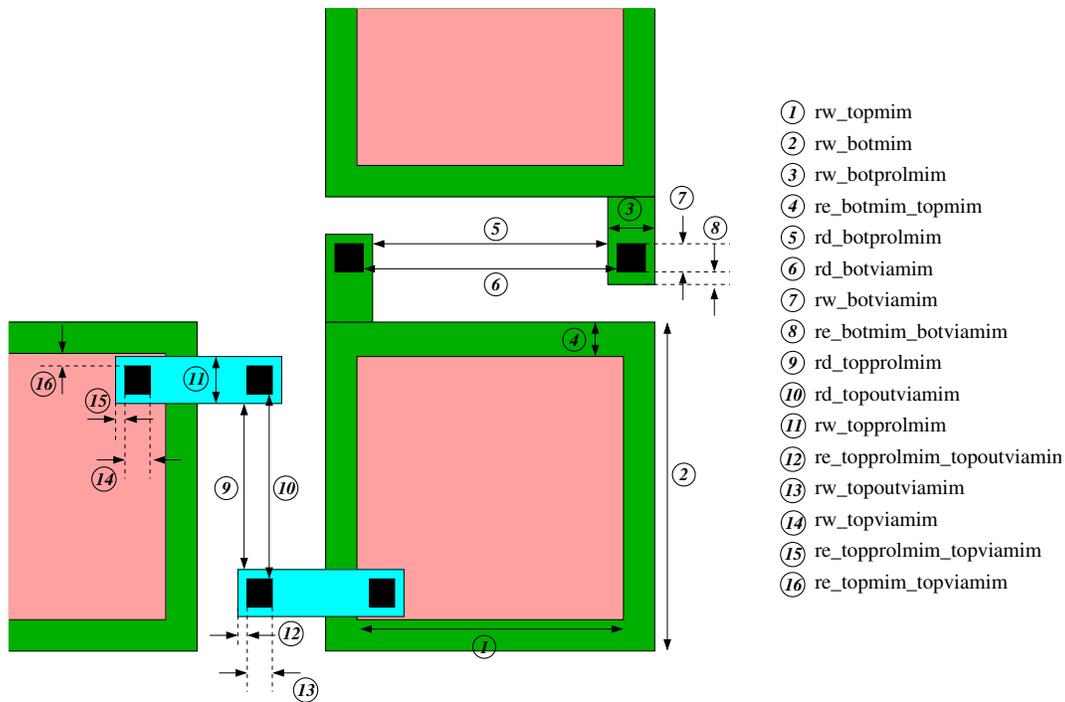


FIG. 6.5 : BICAPA : Règles de dessin intervenant dans le calcul de la largeur minimale de l'armature haute.

- ⑤ RD_BOTPROLMIM : Distance minimale entre deux prolongements d'armature basse.
- ⑥ RD_BOTVIAMIM : Distance minimale entre deux vias de l'armature basse.
- ⑦ RW_BOTVIAMIM : Largeur minimale d'un via de l'armature basse.
- ⑧ RE_BOTMIM_BOTVIAMIM : Extension minimale de l'armature basse autour d'un via de l'armature basse.
- ⑨ RD_TOPPROLMIM : Distance minimale entre deux prolongements d'armature haute.
- ⑩ RD_TOPOUTVIAMIM : Distance minimale entre deux vias à l'extrémité d'un prolongement d'armature haute.
- ⑪ RW_TOPPROLMIM : Largeur minimale du prolongement de l'armature haute.
- ⑫ RE_TOPPROLMIM_TOPOUTVIAMIM : Extension minimale du prolongement d'armature haute autour d'un via à l'extrémité de ce prolongement.
- ⑬ RW_TOPOUTVIAMIM : Largeur minimale du via à l'extrémité d'un prolongement d'armature haute.
- ⑭ RW_TOPVIAMIM : Largeur minimale d'un via sur l'armature haute.
- ⑮ RE_TOPPROLMIM_TOPVIAMIM : Extension minimale du prolongement d'armature haute autour d'un via sur l'armature haute.
- ⑯ RE_TOPMIM_TOPVIAMIM : Extension minimale de l'armature haute autour d'un via sur l'armature haute.

Avec l'ensemble de ces règles, on peut définir dix combinaisons linéaires $Wmin_i$ pour calculer la largeur minimale de l'armature haute :

$$\begin{aligned}
 Wmin_1 &= \textcircled{1} & Wmin_2 &= \textcircled{2} - 2 \times \textcircled{4} \\
 Wmin_3 &= \textcircled{5} + 2 \times \textcircled{3} - 2 \times \textcircled{4} & Wmin_4 &= \textcircled{5} + 2 \times \textcircled{7} + 4 \times \textcircled{8} - 2 \times \textcircled{4} \\
 Wmin_5 &= \textcircled{6} + 2 \times \textcircled{7} + 2 \times \textcircled{8} - 2 \times \textcircled{4} & Wmin_6 &= \textcircled{6} - 2 \times \textcircled{8} + 2 \times \textcircled{3} - 2 \times \textcircled{4} \\
 Wmin_7 &= \textcircled{9} + 2 \times \textcircled{11} - 2 \times \textcircled{15} + 2 \times \textcircled{16} & Wmin_8 &= \textcircled{9} + 2 \times \textcircled{15} + 2 \times \textcircled{14} + 2 \times \textcircled{16} \\
 Wmin_9 &= \textcircled{10} + 2 \times \textcircled{14} + 2 \times \textcircled{16} & Wmin_{10} &= \textcircled{10} - 4 \times \textcircled{15} + 2 \times \textcircled{11} + 2 \times \textcircled{16}
 \end{aligned} \tag{6.1}$$

Afin que toutes les règles de dessin soient respectées, la combinaison menant à la largeur minimale la plus grande sera retenue :

$$Wmin = \max(Wmin_i) \tag{6.2}$$

La fonction `get_rw_mim` se doit finalement d'effectuer le calcul de toutes les combinaisons linéaires, et de renvoyer la plus grande. En revanche, les fonctions `get_xw_mim`, `get_cs_mim` et `get_cp_mim` renverront directement les valeurs données dans le fichier DTR.

Pour écrire ces fonctions, il faut vérifier que toutes les règles impliquées dans les calculs qu'elles effectuent apparaissent dans le fichier DTR. Les règles `CS_MIM` et `CP_MIM` appartiennent à la table `ELECTRICAL`. Toutes les autres seront indiquées dans une table `BICAPA` spécifique au générateur en cours de conception et dans laquelle on utilisera la nomenclature de la figure 6.5.

6.2.6.2 Fonctions de la DAPI requises par `CAIRO_SHAPE`

Nous avons défini l'indice de forme du générateur `BICAPA` comme étant le nombre de lignes de la matrice. Les formes extrêmes sont donc la matrice ligne et la matrice colonne, qui présente $n1 + n2$ lignes. Le calcul du nombre de formes ne nécessite donc ici aucune information technologique.

6.2.6.3 Fonctions de la DAPI requises par `CAIRO_GENERATE`

Enfin, nous avons proposé deux métaux pour le routage selon la direction horizontale ou verticale des fils. La DAPI doit donc mettre à disposition de `CAIRO_GENERATE` les deux fonctions suivantes :

- `double get_alu_rout_H()` : renvoie la couche de routage des fils horizontaux.
- `double get_alu_rout_V()` : renvoie la couche de routage des fils verticaux.

Les couches physiques utilisées pour le routage seront définies dans la table `LAYERS` du fichier DTR par les règles suivantes :

- `bicapa_alu_rout_H` : couche de routage des fils horizontaux.
- `bicapa_alu_rout_V` : couche de routage des fils verticaux.

6.2.7 Etape 7 : écriture des procédures de dimensionnement : CAIRO_DSES

On dispose désormais de tous les éléments permettant d'écrire les différentes procédures de synthèse (i.e. dimensionnement) et d'analyse (i.e. évaluation des performances) au sein de la fonction CAIRO_DSES. Pour cela, le concepteur doit s'appuyer sur les fonctions de la calculette de capacités pour accéder au modèle et sur celles de la DAPI pour accéder à la technologie.

Nous ne détaillerons pas plus cette partie qui ne constitue pas l'objet principal de cette thèse.

6.2.8 Etape 8 : écriture de CAIRO_SHAPE

Il reste à prendre en charge la génération du dessin des masques. Celle-ci est prise en charge par les fonctions CAIRO_SHAPE, qui calcule le nombre de formes que pourra prendre le dispositif, et CAIRO_GENERATE, qui génère effectivement chacune des formes.

Pour le générateur BICAPA, le nombre maximal de lignes de la matrice est $nbl = n1 + n2$. L'indice de forme nbl prendra ainsi des valeurs comprises entre 1 et $n1 + n2$. Lorsqu'elle a effectué ce calcul, CAIRO_SHAPE fait appel à la fonction CAIRO_GENERATE pour toutes les valeurs de l'indice de forme. Rappelons qu'il est possible que certaines valeurs de l'indice de forme conduisent à des réalisations qui ne seront pas retenues par CAIRO+.

6.2.9 Etape 9 : identification des motifs

La dernière fonction restant à écrire est CAIRO_GENERATE. Celle-ci prend en charge la génération effective du dessin des masques en s'appuyant sur des générateurs de motifs. Il faut donc commencer par identifier précisément ces motifs d'après le plan de masse que l'on a choisi. Il s'agit de mettre en évidence les éléments répliqués qui, par aboutement, construisent le dessin. Dans le cas général, une dizaine de générateurs de motifs sont au maximum nécessaires à la réalisation d'un dispositif élémentaire. Pour le dispositif BICAPA, on définit cinq motifs qui permettent de construire le plan de masse désiré. Ceux-ci sont illustrés sur la figure 6.6. Les quatre premiers permettent de réaliser une capacité élémentaire et sont caractérisés par l'orientation des prolongements des armatures. Le prolongement de l'armature haute se situe toujours soit à l'est, soit à l'ouest du motif, tandis que le prolongement de l'armature basse se situe soit au nord, soit au sud. Le motif *Cfict* correspond à une capacité élémentaire fictive.

On remarque que les prolongements des armatures, assimilables à des fils de routage, font partie des motifs, et sont donc dessinés directement sur la grille réelle. En effet, si l'on avait choisi de dessiner ces prolongements sur la grille symbolique avec GENLIB, on aurait ajouté une contrainte supplémentaire sur la géométrie minimale des armatures. La valeur de la capacité élémentaire minimale aurait été augmentée.

Le problème est que les vias à l'extrémité de ces prolongements assurent la connexion électrique entre la capacité élémentaire et les fils de routage. Ils devraient donc se trouver sur la grille symbolique pour pouvoir assurer un contact avec les fils de routage qui eux sont symboliques.

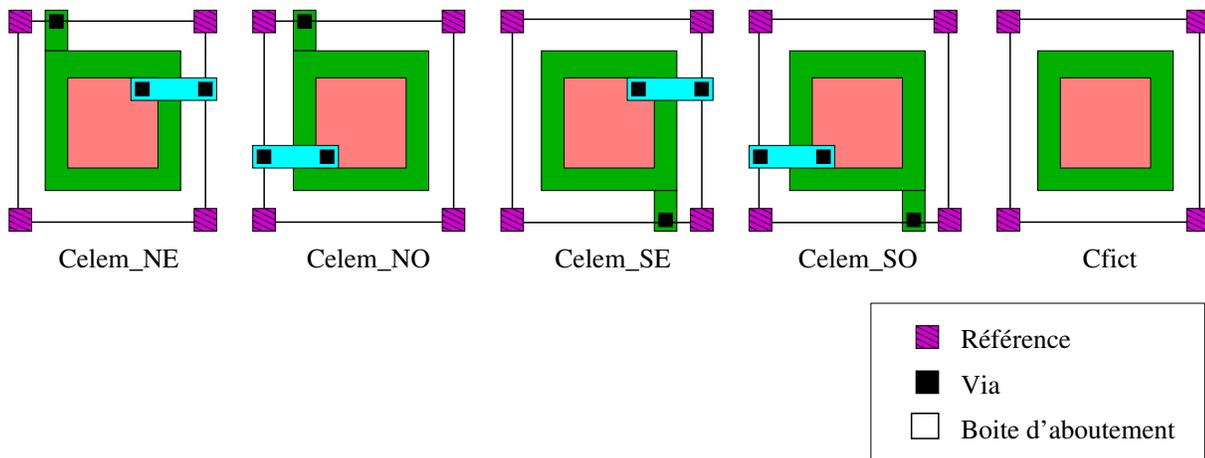


FIG. 6.6 : Motifs du dispositif BICAPA.

Ce problème peut être résolu si ces vias ont au moins une coordonnée sur la grille symbolique, on parle de vias semi-symboliques. Comme le montre la figure 6.7, un fil de routage symbolique peut couvrir un via semi-symbolique et la connexion est assurée. Sur la figure 6.6, on voit que les



FIG. 6.7 : Connexion entre un via semi-symbolique et un fil de routage.

vias à l'extrémité des prolongements se trouvent sur la boîte d'aboutement. Ils présentent donc effectivement une coordonnée sur la grille symbolique.

Malheureusement, on ne peut plus alors placer de connecteur symbolique sur ces vias comme il faut normalement le faire dans la vue symbolique d'un motif (cf. 5.6.3). En conséquence on placera plutôt des références aux quatre coins du motif. Une référence est, à l'instar d'un connecteur, un objet virtuel qui permet simplement de référencer un point du plan par un nom. Cependant, elle n'est pas associée à une sémantique de connexion. Grâce à ces références, le routage pourra se faire d'un coin de motif à un autre coin de motif, couvrant ainsi les vias qui assurent la connexion avec les motifs.

6.2.10 Etape 10 : schéma de conception des motifs

Nous avons vu que la conception des motifs repose sur la MAPI qui permet de calculer la géométrie de chacun des rectangles constituant le motif. Pour faciliter l'écriture et la maintenance de la MAPI, on se base sur un **schéma de conception** du motif. Celui-ci recense l'ensemble des rectangles qui constituent la vue réelle du motif et leur attribue un nom. Ce nom est généralement un numéro qui, par convention, est associé à l'importance électrique du rectangle. Les numéros les plus petits correspondent ainsi aux rectangles directement liés au comportement électrique du motif. Par exemple, la grille d'un transistor sera appelée 00. La figure 6.8 montre le schéma de conception du motif *Celem_NE*. Les autres motifs de capacités élémentaires se différencient du motif *Celem_NE* seulement par la position de quelques rectangles. Dès lors, leur conception se basera sur ce même schéma de conception, c'est-à-dire que la numérotation des rectangles restera la même. Le motif *Cfict* ne présentera que les rectangles 00 et 10.

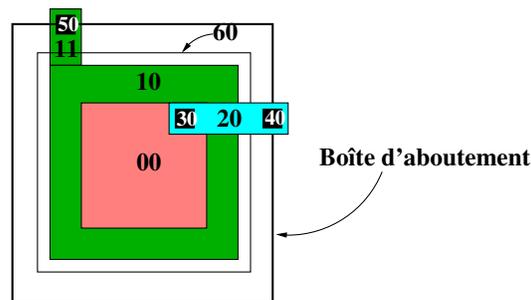


FIG. 6.8 : Schéma de conception du motif *Celem_NE*.

La boîte d'aboutement constitue le référentiel du motif. Aussi, et bien qu'elle appartienne à la vue symbolique, elle est représentée sur le schéma de conception. Par ailleurs, on dénombre huit rectangles dans le motif :

- L'armature haute, qui détermine directement la valeur de la capacité, est appelée 00.
- L'armature basse qui constitue l'autre rectangle significatif est appelée 10.
- Pour connecter l'armature basse, on utilise un prolongement dans le même métal appelé 11.
- Pour connecter l'armature haute, on utilise un prolongement de métal différent appelé 20.
- Pour connecter l'armature haute à son prolongement, on utilise le via 30.
- La connexion entre l'armature haute et la piste de routage sera établie par le via 40.
- Pour connecter l'armature basse à la piste de routage, on place le via 50.
- Dans certaines technologies, les capacités sont repérées par un marqueur. Il s'agit d'un rectangle qui permet de situer les zones dans lesquelles l'oxyde doit être plus mince. Pour ce type de procédé, on envisage le rectangle 60 qui permettra de placer un marqueur.

Pour finir, remarquons qu'aucun macro-rectangle n'a été identifié dans ce schéma.

6.2.11 Etape 11 : Conception des générateurs de motifs

Avec le schéma de conception des motifs, on peut d'ores et déjà concevoir les générateurs de motifs. En effet, bien que l'on ne dispose pas encore de la MAPI, toutes les fonctions qu'elles proposera sont définies : cinq par rectangle du schéma de conception et deux pour les dimensions de la boîte d'aboutement.

Pour écrire ces générateurs de motifs, il faut commencer par déterminer quels seront leurs paramètres. Pour mémoire, on distingue les paramètres standard qui sont incontournables, et les paramètres contextuels qui permettent d'intégrer le motif dans son contexte d'instanciation. Pour les générateurs que nous proposons, les paramètres standard pourront être :

- `modelName` : Nom du modèle généré. Ce paramètre est obligatoire pour tous les générateurs de motifs.
- `Celem` : Valeur de la capacité élémentaire.

D'un autre côté, dans la mesure où les cinq motifs que nous proposons sont très ressemblants et sont décrits par le même schéma de conception, on ne va écrire qu'un unique générateur. Un paramètre contextuel `orientation` permettra de choisir le motif à générer. Ce paramètre pourra prendre les valeurs suivantes :

- `NE` pour générer le motif *Celem_NE*.
- `NO` pour générer le motif *Celem_NO*.
- `SE` pour générer le motif *Celem_SE*.
- `SO` pour générer le motif *Celem_SO*.
- `FI` pour générer le motif *Cfict*.

A partir de cette spécification, l'écriture d'un générateur de motif suit toujours les mêmes étapes :

- Création des structures de données qui contiennent la vue symbolique et la vue réelle du motif.
- Appels aux fonctions de la MAPI afin de déterminer la géométrie de tous les rectangles du motif.
- Génération effective des rectangles symboliques dans la vue symbolique et les rectangles réels dans la vue réelle.
- Evaluation des parasites introduits. Pour cela, on utilise éventuellement la calculette.
- Caractérisation du motif dans la structure de données de retour.

Ici, un seul générateur, `bi_capa_elem`, prend en charge la réalisation des cinq motifs nécessaires.

6.2.12 Etape 12 : graphes de dépendance

Il faut maintenant s'attaquer aux fonctions des différentes MAPI correspondant à chaque générateur de motifs. Nous avons vu que les fonctions d'une MAPI calculent la géométrie des rectangles de façon relative. Autrement dit, elles s'appellent entre elles. Ces dépendances peuvent être représentées par deux graphes orientés acycliques (DAG), que l'on appelle les graphes de dépendance. L'un représente les dépendances régissant les calculs de dimensions horizontales

telles que des abscisses X ou des largeurs DX , l'autre représente les dépendances régissant les calculs de dimensions verticales telles que des ordonnées Y ou des hauteurs DY . Avant d'entamer l'écriture des fonctions de toutes les MAPI, il faut dessiner ces graphes de dépendance.

En observant le schéma de conception d'un motif, on peut identifier les rectangles dépendants. Par exemple, sur le schéma de conception de la figure 6.8, la géométrie de l'armature basse sera directement déduite de celle de l'armature haute. L'idée est d'identifier le plus de dépendances possibles afin de faciliter la réalisation et la maintenance de la MAPI. On peut alors dessiner les graphes de dépendance et vérifier qu'ils sont bien acycliques. La figure 6.9 montre le graphe des dépendances horizontales pour le schéma de conception du générateur de motifs bicapa_elem. Sur ce graphe DX_{AB} désigne la largeur de la boîte d'aboutement, tandis que X_{zz} et DX_{zz} désignent

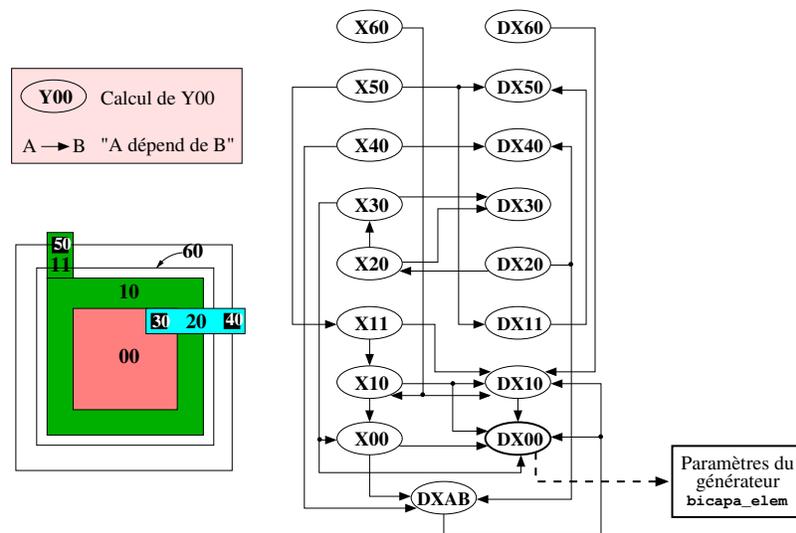


FIG. 6.9 : Graphe des dépendances horizontales pour le générateur bicapa_elem.

respectivement l'abscisse et la largeur du rectangle zz . Rappelons d'une part que la boîte d'aboutement se trouve à l'origine et que son abscisse est nulle, et d'autre part que la largeur de l'armature haute DX_{00} ne dépend que du paramètre standard C_{elem} du générateur bicapa_elem.

Dans la mesure où l'on a décidé qu'un unique générateur permettrait de réaliser les cinq motifs de notre dispositif, les graphes de dépendance vont varier selon l'orientation du motif. Cependant, seules les dépendances liées aux rectangles 11, 20, 30, 40 et 50 vont être modifiées. Par exemple, selon que l'on oriente le prolongement de l'armature haute à l'ouest ou à l'est, x_{30} dépendra dans un cas uniquement de x_{00} , et dans l'autre cas, de x_{00} , de dx_{00} et de dx_{30} . Pour s'assurer qu'il n'existe pas de cycle dans les graphes de dépendance, il faut représenter toutes les dépendances quelles que soient leurs conditions d'apparition. Ainsi, la figure 6.9 montre les dépendances horizontales qui peuvent exister pour le générateur bicapa_elem, quelle que soit la valeur du paramètre contextuel orientation.

6.2.13 Etape 13 : écriture des MAPI et mise à jour du fichier DTR

A ce stade de notre méthodologie, on peut écrire les fonctions de toutes les MAPI. Chaque MAPI propose trois types de fonctions :

- Les fonctions calculant les dimensions symboliques de la boîte d'aboutement.
- Les fonctions calculant les coordonnées, les dimensions et la couche de tous les rectangles de la vue réelle du motif.
- Les fonctions de génération des macro-rectangles.

De manière générale, on commence par écrire les fonctions qui ne dépendent que des paramètres standard du générateur de motifs. Puis, on écrit les fonctions qui en dépendent, et ainsi de suite. Enfin, on écrit les fonctions de génération des macro-rectangles. Ces dernières sont particulières puisqu'elles ne sont pas chargées de déterminer une géométrie, mais de générer un certain nombre de rectangles dans une zone géométrique donnée. Elles prendront ainsi en paramètres les cinq variables qui caractérisent le macro-rectangle.

Pour illustrer l'écriture d'une MAPI, nous nous proposons de détailler les premières fonctions que nous écrirons pour la MAPI du générateur `bi_capa_elem` : celles qui calculent `DX00`, `DX10` et `DXAB`.

La valeur de `DX00` est déduite de celle de `Celem`. La fonction permettant de la calculer va utiliser la calculette de capacités MIM pour déterminer la largeur du carré qui réalise `Celem`. Il lui reste uniquement à arrondir le résultat au pas de grille δ . Aucune règle de dessin n'est nécessaire ici puisque la faisabilité de `Celem` a déjà été vérifiée (cf. section 6.2.4).

Le calcul de `DX10` est légèrement plus complexe. En effet, la largeur de l'armature basse doit respecter la largeur minimale `RW_BOTMIM` imposée par la technologie, mais également l'extension minimale autour de l'armature haute `RE_BOTMIM_TOPMIM`. Ainsi, en reprenant la numérotation de la figure 6.5, on la détermine par le calcul suivant :

$$DX10 = \max(DX00 + 2 \times \textcircled{4} ; \textcircled{2}) \quad (6.3)$$

Les règles `RW_BOTMIM` et `RE_BOTMIM_TOPMIM` sont déjà présentes dans le fichier DTR car elles avaient été utilisées par la DAPI.

Le calcul des dimensions de la boîte d'aboutement est en général assez complexe. Effectivement, c'est ici que l'on prend en compte les règles de dessin, et en particulier les distances minimales, qu'il faut respecter avec les motifs adjacents. Dans le cas du motif que nous étudions, on veut de plus que la boîte d'aboutement soit carrée. Ainsi, la hauteur sera égale à la largeur et le calcul de la largeur doit permettre de respecter les règles avec les motifs qui seront aboutés tout autour. La figure 6.10 recense ainsi l'ensemble des règles de dessin qui interviennent dans le calcul de `DXAB` (attention, sur cette figure, nous utilisons une nouvelle numérotation des règles, différente de celle de la figure 6.5) :

- ① `RD_BOTMIM` : Distance minimale entre deux armatures basses.
- ② `RD_TOPMIM` : Distance minimale entre deux armatures hautes.
- ③ `RD_TOPMIM_BOTVIA` : Distance minimale entre l'armature haute et un via de l'armature basse.

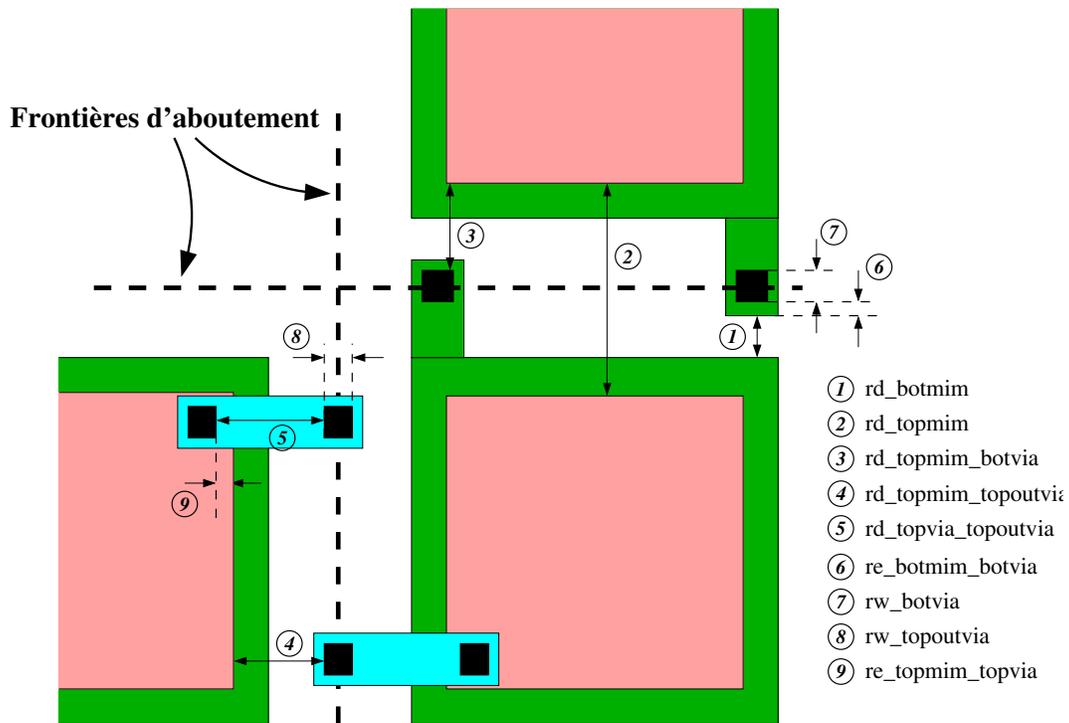


FIG. 6.10 : bicapa_elem : Règles de dessin intervenant dans le calcul de la largeur de la boîte d'aboutement.

- ④ RD_TOPMIM_TOPOUTVIA : Distance minimale entre l'armature haute et un via à l'extrémité du prolongement de l'armature haute.
- ⑤ RD_TOPVIA_TOPOUTVIA : Distance minimale entre un via de l'armature haute et un via à l'extrémité du prolongement de l'armature haute.
- ⑥ RE_BOTMIM_BOTVIA : Extension minimale de l'armature basse autour d'un via de l'armature basse.
- ⑦ RW_BOTVIA : Largeur d'un via de l'armature basse.
- ⑧ RW_TOPOUTVIA : Largeur d'un via à l'extrémité du prolongement de l'armature haute.
- ⑨ RE_TOPMIM_TOPVIA : Extension de l'armature haute autour d'un via de l'armature haute.

Toutes ces règles seront placées dans la table BICAPA du fichier DTR. On peut trouver cinq combi-

naisons linéaires $dxab_i$ qui aboutissent à la largeur de la boîte d'aboutement $DXAB$:

$$\begin{aligned} dxab_1 &= DX10 + 2 \times \left(\textcircled{1} + \textcircled{6} + \frac{\textcircled{7}}{2} \right) \\ dxab_2 &= DX00 + 2 \times \frac{\textcircled{2}}{2} \\ dxab_3 &= DX00 + 2 \times \left(\textcircled{3} + \frac{\textcircled{7}}{2} \right) \\ dxab_4 &= DX00 + 2 \times \left(\textcircled{4} + \frac{\textcircled{8}}{2} \right) \\ dxab_5 &= DX00 + 2 \times \left(\textcircled{5} + \frac{\textcircled{8}}{2} - \textcircled{9} \right) \end{aligned} \quad (6.4)$$

La plus grande d'entre elles permettra de respecter toutes les distances avec les motifs voisins et la valeur symbolique de $DXAB$ sera finalement donnée par :

$$DXAB = \frac{\lceil \max(dxab_i) \rceil^\lambda}{\lambda} \quad (6.5)$$

où $\lceil x \rceil^\lambda$ est l'arrondi par excès au lambda de x .

Les autres fonctions de la MAPI sont écrites en suivant le même raisonnement, à savoir :

- Identifier les combinaisons linéaires de règles qui régissent le calcul
- Conserver la valeur la plus contraignante, c'est-à-dire la plus grande.

Finalement, le générateur de motif et sa MAPI étant disponible, on peut vérifier le résultat de la génération des motifs. La figure 6.11 montre ainsi les motifs obtenus pour une capacité élémentaire de 50fF dans une technologie 180 nm.

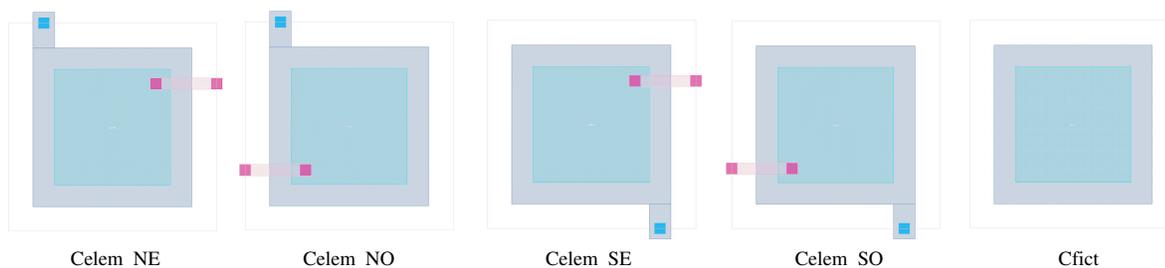


FIG. 6.11 : Motifs obtenus.

6.2.14 Etape 14 : écriture de CAIRO_GENERATE

La dernière fonction constructrice, à savoir `CAIRO_GENERATE`, peut maintenant être écrite. Il faut d'abord y générer tous les motifs nécessaires en faisant appel aux différents générateurs de motifs. On peut alors effectuer le placement et le routage des différents motifs selon les algorithmes définis

à l'étape 2. Au fur et à mesure du placement et du routage, on calculera les parasites grâce aux données rendues par les générateurs de motifs, et grâce à la calculette de capacités parasites.

La figure 6.12 montre un exemple de dessin finalement obtenu pour le générateur BICAPA en technologie $180nm$. Celui-ci correspond bien au plan de masse visé.

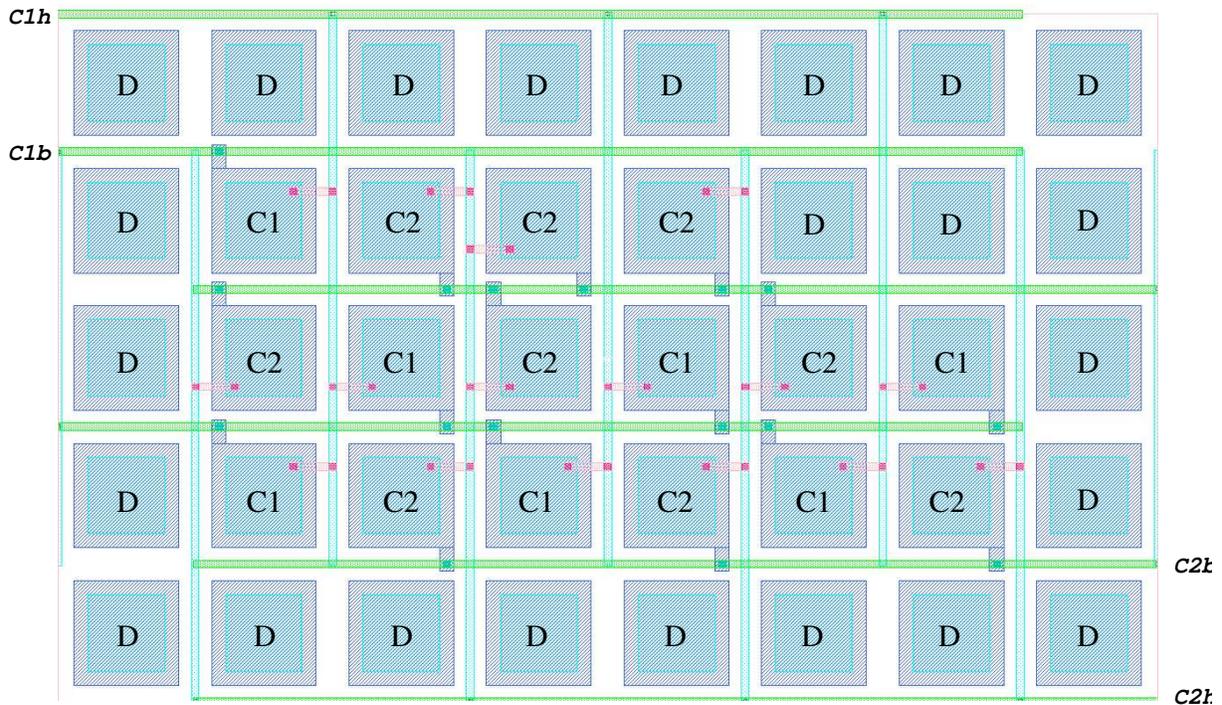


FIG. 6.12 : Dessin des masques final. $C_{elem} = 50fF$, $n1 = 7$, $n2 = 9$.

6.3 Conclusion

Nous avons détaillé dans ce chapitre toutes les étapes de la conception d'un générateur de dispositifs élémentaires. En respectant cette méthodologie, nous avons pu développer une bibliothèque de générateurs de dispositifs élémentaires permettant la conception de circuits analogiques pour une large gamme d'applications. Dans le chapitre suivant, nous allons présenter chacun des générateurs de cette bibliothèque.

Chapitre 7

Bibliothèque actuelle

7.1 Introduction

Ce chapitre présente l'ensemble des générateurs de dispositifs élémentaires actuellement disponibles en donnant les informations nécessaires à leur utilisation. Ces générateurs sont en fait classés en trois bibliothèques :

- La bibliothèque `libTRANSISTOR` contient les générateurs de dispositifs à transistors. Les générateurs de cette bibliothèque sont présentés dans la section 7.2.
- La section 7.3 présente la bibliothèque `libMIMCAP` qui contient les générateurs de capacités MIM.
- Enfin, la bibliothèque `libPOLYRES` qui contient les générateurs de dispositifs résistifs est présentée dans la section 7.4.

A chacune de ces trois bibliothèques est associé un ensemble de générateurs de motifs dédiés. Nous commencerons donc la présentation des bibliothèques `libTRANSISTOR`, `libMIMCAP` et `libPOLYRES` par une présentation des générateurs de motifs qui leur sont respectivement associés.

7.2 Bibliothèque `libTRANSISTOR`

Dans *CAIRO+*, le dessin des masques de tous les dispositifs à transistors est construit de la même façon : par aboutement de doigts de transistors. Puisque les générateurs de transistors sont très largement utilisés, on a cherché à faciliter leur conception en mutualisant complètement les générateurs de motifs qu'ils utilisent. Les générateurs de motifs de la bibliothèque `libTRANSISTOR` proposent ainsi un certain nombre de paramètres contextuels qui permettent cette réutilisation.

7.2.1 Générateurs de motifs pour les dispositifs à transistors

Tous les générateurs de la bibliothèque `libTRANSISTOR` s'appuient sur six générateurs de motifs. La figure 7.1 illustre un exemple de motif issu de chacun de ces générateurs.

- Le générateur `MOS_M1` dessine un transistor non replié, c'est-à-dire dont le nombre de repliements M vaut 1. Dans ce cas particulier, le dessin du transistor est représenté par un motif

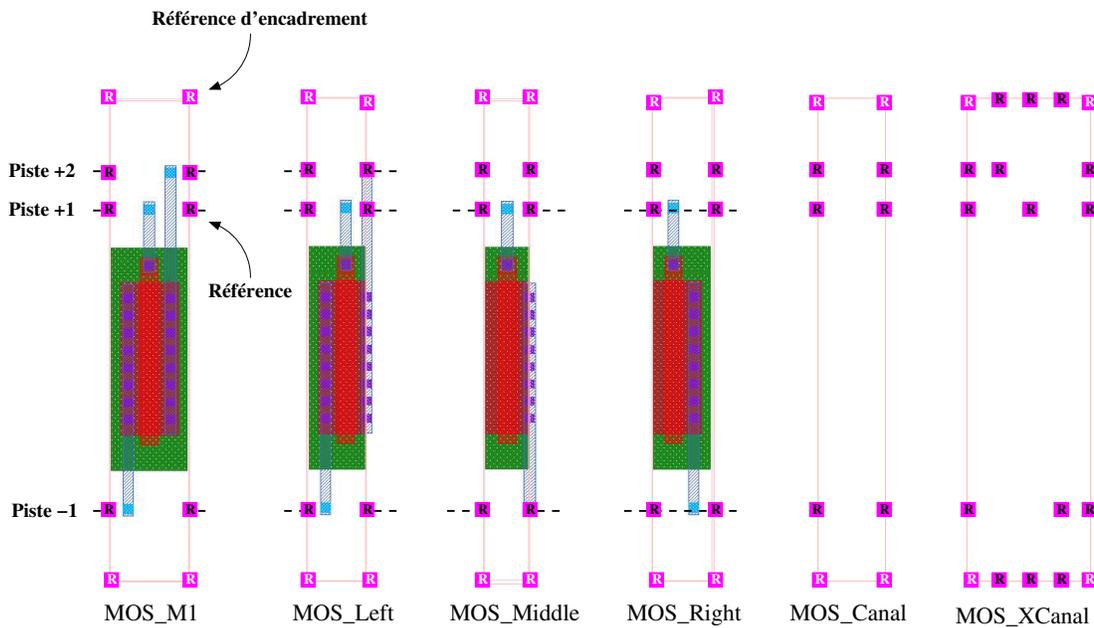


FIG. 7.1 : Générateurs de motifs pour libTRANSISTOR.

unique. Aucune diffusion n'est ainsi étendue et on peut minimiser la surface du transistor.

- Le générateur `MOS_Left` prend en charge le dessin du premier doigt d'un transistor replié. La particularité ici est que l'aboutement se fera uniquement sur la face droite du motif. Ainsi, la diffusion de gauche peut être minimale, alors que celle de droite devra être étendue jusqu'à la boîte d'aboutement.
- Le générateur `MOS_Middle` dessine un doigt intérieur d'un transistor replié. Pour ce motif, l'aboutement se fera sur les faces gauche et droite. Les deux diffusions seront étendues de façon identique de part et d'autre de la grille.
- Le générateur `MOS_Right` prend en charge le dessin du dernier doigt d'un transistor replié. L'aboutement se fera ici sur la face gauche du motif. Ainsi, la diffusion de droite peut être minimale.
- Le générateur `MOS_Canal` permet d'insérer des espaces dans le dessin d'un empilage. Il permettra notamment de respecter la distance nécessaire entre une zone active et son anneau de garde.
- Le générateur `MOS_XCanal` est utilisé pour le routage de plusieurs empilages. Il définit des références au niveau des intersections entre fils verticaux et horizontaux. Des vias symboliques pourront ainsi être placés sur ces références.

Les quatre premiers générateurs prennent en charge le dessin des doigts des transistors. On remarque que les motifs qu'ils génèrent contiennent des fils métalliques verticaux assurant la connexion électrique avec les diffusions et la grille. Bien que ces fils auraient pu être considérés comme des fils de routage et, à ce titre, être dessinés sur la grille symbolique, nous avons choisi de les intégrer aux motifs. Ceci permet de ne pas imposer aux contacts de diffusion de se situer sur la

grille symbolique. Dans le cas contraire, on ajouterait une contrainte supplémentaire qui pourrait conduire à une augmentation de la taille des zones de diffusion. Des vias semi-symboliques à l'extrémité de ces fils verticaux permettent la connexion avec des pistes de routage horizontales. Ces pistes sont indiquées sur la figure 7.1. Par souci d'homogénéité, la même technique est employée pour le routage de la grille.

Les zones de diffusion internes d'un empilage (celles enfermées entre deux doigts) sont créées par l'aboutement de deux motifs. Par convention, c'est le motif de gauche qui prend en charge le routage de ce type de diffusion. C'est pourquoi les motifs `MOS_Middle` et `MOS_Right` ne présentent pas de fil métallique vertical connectant leur diffusion de gauche. En revanche, le fil vertical connectant la diffusion de droite des motifs `MOS_Left` et `MOS_Middle` se trouve sur la boîte d'aboutement, afin que celui-ci soit centré sur la diffusion après aboutement.

Dans un premier temps, nous allons présenter les quatre générateurs de doigts de transistors et notamment leurs différents paramètres. Nous verrons ensuite les paramètres des deux autres générateurs qui sont dédiés au routage.

7.2.1.1 Paramètres standard des générateurs de doigts de transistor

Nous avons déjà vu que tous les générateurs de motifs doivent proposer un paramètre standard `modelName` qui détermine le nom du modèle généré. Les générateurs `MOS_M1`, `MOS_Left`, `MOS_Middle` et `MOS_Right` proposent donc ce paramètre. Par ailleurs, un doigt de transistor ne peut être dessiné sans connaître sa largeur `Wf`, sa longueur `L` et son type. Il s'agit donc de trois paramètres standard pour les générateurs de doigts de transistor. Nous verrons à la section 7.2.2 quels sont les types qui peuvent être souhaités pour un transistor.

7.2.1.2 Paramètres contextuels des générateurs de doigts de transistor

Pour que nos générateurs de motifs, et en particulier ceux de doigts de transistors, puissent servir à la conception de tous les dispositifs élémentaires à transistors, ils doivent proposer plusieurs paramètres contextuels.

Tout d'abord, selon la position d'un doigt dans un empilage, la diffusion de gauche pourra correspondre à une source ou à un drain. Il en va de même pour la diffusion de droite. D'un autre côté, dans les dispositifs multi-transistors, chaque doigt peut appartenir à un transistor électrique différent. Dans ce cas, la grille de chaque doigt sera par exemple connectée à une piste de routage différente. Dès lors, il est nécessaire de pouvoir spécifier, pour chaque doigt, les pistes de routage horizontales sur lesquelles devront se connecter respectivement la diffusion de gauche, la grille et la diffusion de droite. Pour cela, on propose trois paramètres contextuels `posLeft`, `posGate` et `posRight`. Ces trois paramètres prennent des valeurs entières positives ou négatives. Une valeur positive indique une piste au dessus du doigt, une valeur négative indique une piste en dessous. Ainsi, sur la figure 7.1, on a donné au générateur `MOS_Left` les paramètres contextuels suivants :

- `posLeft = -1`
- `posGate = 1`

- `posRight = 2`

Pour effectuer le routage procédural inter-motifs, nous avons vu qu'il faut s'appuyer sur les références mises à disposition par la vue symbolique des motifs. Dans le cas des motifs de doigts de transistor, ces références ne peuvent pas apparaître sur les vias à l'extrémité des fils internes verticaux, qui correspondent pourtant à l'interface électrique des motifs. En effet, ces vias ne sont que semi-symboliques et seule leur ordonnée est multiple de λ . On se propose alors de placer des références à la même hauteur que ces vias, sur les faces est et ouest de chacun des motifs. Après l'aboutement, les pistes de routage horizontales pourront être tracées entre les références à l'ouest du motif `MOS_Left` et celles à l'est du motif `MOS_Right`.

A priori, il s'agit donc de placer deux références (une sur la face est, l'autre sur la face ouest) par via de connexion et donc par fil métallique vertical interne au motif. Ceci n'est malheureusement pas suffisant. Comme on peut le voir sur la figure 7.1, le motif `MOS_Right` doit présenter des références permettant de router sa diffusion de gauche, bien qu'il ne contienne pas de fil vertical interne la connectant. Dans le cas de dispositifs multi-transistors, il est également possible que certains transistors ne soient physiquement représentés que par des doigts internes. Pour que le routage prennent en compte ces transistors, il faut pouvoir présenter des références qui leur sont spécifiques, notamment sur les motifs extrêmes `MOS_Left` et `MOS_Right`.

Un paramètre contextuel `signals` permet alors d'insérer des références qui ne sont pas directement liées au motif courant. Il définit en effet une liste des références que le motif doit proposer sur sa boîte d'aboutement. Ce paramètre est de type `Ref_List` qui est une liste chaînée de références caractérisées par un nom et une position :

```

1 typedef struct reflight{
2     string refName; // Nom de la reference
3     int position; // Position de la piste de routage
4     reflight *next;
5 } Ref_List;
```

Pour chaque élément de cette liste, on placera deux références nommées `refName_O` et `refName_E` respectivement sur les faces ouest et est de la boîte d'aboutement, à une hauteur donnée par `position`. Le placement des références se fait donc sur la seule lecture de cette liste et indépendamment des valeurs attribuées aux paramètres `posLeft`, `posGate` et `posRight`. Il est à la charge du concepteur de s'assurer que les pistes spécifiées dans ces paramètres disposent d'une référence correspondante dans la liste de références.

Finalement, la hauteur de la boîte d'aboutement d'un motif de doigt de transistor est calculée de façon à ce que les références les plus hautes se trouvent sur les coins supérieurs de cette boîte d'aboutement, et les références les plus basses se trouvent sur les coins inférieurs. C'est la raison pour laquelle on ajoute en général dans la liste deux **références d'encadrement** : l'une au dessus de la référence la plus haute et l'autre en dessous de la référence la plus basse. Ces références d'encadrement ne correspondent à aucune piste de routage. Elles permettent d'introduire un espace entre les deux pistes de routage extrêmes et la boîte d'aboutement (fig 7.1).

7.2.1.3 Paramètres de sortie des générateurs de doigts de transistor

Dans la structure rendue par tous les générateurs de motifs, les générateurs de doigts de transistor renseignent les champs suivants :

- `DXAB` et `DYAB` : Dimensions symboliques de la boîte d'aboutement du motif.
- `DXAB_mu` et `DYAB_mu` : Dimensions réelles de la boîte d'aboutement du motif.
- `Wf` et `L` : Largeur et longueur effectives du doigt.
- `C_Left`, `C_Gate` et `C_Right` : Capacités parasites entre les fils internes verticaux et le substrat.
- `half_dgg` : largeur d'une diffusion interne. Après l'aboutement avec un autre motif, la distance `dgg` entre deux grilles consécutives vaut le double de ce paramètre.
- `de` : largeur d'une diffusion externe.

Evidemment, le motif `MOS_Middle` ne renseigne ni `C_Left`, ni `de`. Le motif `MOS_Right` ne renseigne pas `C_Left` non plus.

Grâce à ces informations, le générateur de dispositifs élémentaires peut déterminer parfaitement la géométrie globale du dessin, et prendre en compte toutes les capacités parasites de routage.

7.2.1.4 Paramètres standard des générateurs dédiés au routage

Les générateurs `MOS_Canal` et `MOS_XCanal` doivent proposer un certain nombre de références pour le routage. On ne peut ainsi les générer qu'en connaissant les dimensions de leur boîte d'aboutement et la liste des références à placer. Le générateur `MOS_Canal` prend donc comme paramètres standard, les dimensions `width` et `height` de sa boîte d'aboutement, et la liste `signals` des références qu'il doit placer. En revanche, la largeur de la boîte d'aboutement du motif `MOS_XCanal` dépend du nombre de références de la liste. Les paramètres standard de ce générateur ne sont donc que `height` et `signals`.

Ces deux générateurs ne proposent pas de paramètre contextuel.

7.2.1.5 Paramètres de sortie des générateurs dédiés au routage

Dans la structure rendue, les deux générateurs de motifs dédiés au routage ne remplissent que les champs indiquant les dimensions de leur boîte d'aboutement :

- `DXAB` et `DYAB`.
- `DXAB_mu` et `DYAB_mu`.

Le générateur de dispositifs élémentaires peut alors prendre en compte ces dimensions pour déterminer les dimensions globales du dessin.

7.2.1.6 Tableau récapitulatif

Le tableau 7.1 résume finalement les paramètres de chaque générateur de motifs pour les dispositifs à transistors.

Nom du générateur	Paramètres standard	Paramètres contextuels	Informations rendues
MOS_M1	string modelName double Wf double L int type	int posLeft int posGate	DXAB DYAB DXAB_mu DYAB_mu C_left (MOS_M1 et MOS_Left)
MOS_Left		int posRight Ref_List *signals	C_Gate C_Right Wf L
MOS_Middle		int posGate int PosRight	half_dgg de (sauf MOS_Middle)
MOS_Right		Ref_List *signals	
MOS_Canal	string modelName int width (MOS_Canal)		DXAB DYAB
MOS_XCanal	int height Ref_List *signals		DXAB_mu DYAB_mu

TAB. 7.1 : *libTRANSISTOR* : Paramètres des générateurs de motifs.

Avec ces six générateurs de motifs, nous avons conçu les générateurs de dispositifs élémentaires à transistors que nous allons présenter dans les sections suivantes.

7.2.2 Générateur de transistors simples : TR_MOS

7.2.2.1 Schéma électrique et paramètres structurels

Le générateur TR_MOS permet de réaliser un transistor unique. Il propose trois paramètres structurels présentés dans le tableau 7.2.

Nom du paramètre	Description	Valeurs possibles
string tr_name	Nom du transistor dans la vue structurelle	Chaîne de caractères
int type	Type du transistor	TRANSN : Transistor N à faible courant de fuite TRANSP : Transistor P à faible courant de fuite TRANSN_FAST : Transistor N rapide (faible V_T) TRANSP_FAST : Transistor P rapide (faible V_T)
bool BS	Substrat polarisé à la source	True / False

TAB. 7.2 : TR_MOS : Paramètres structurels.

La figure 7.2 donne deux exemples du schéma électrique réalisé pour deux jeux de paramètres structurels différents.

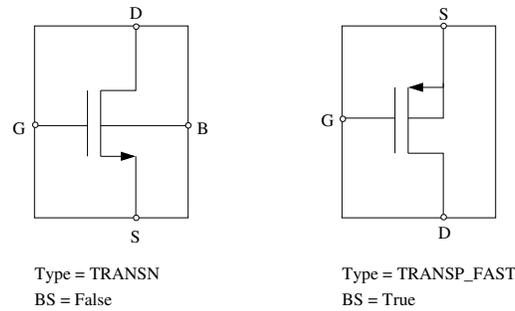


FIG. 7.2 : Schéma électrique du dispositif TR_MOS.

7.2.2.2 Procédures de dimensionnement et paramètres électriques

L'ensemble des procédures de dimensionnement électrique et d'estimation des performances que propose le dispositif TR_MOS est organisé en 5 familles :

- Famille Z : Procédures de dimensionnement direct.
- Famille A : Procédures d'estimation des performances basées sur le modèle Bsim3v3 direct.
- Famille B : Procédures de dimensionnement basées sur une inversion du modèle Bsim3v3 et sur un V_{GS} donné.
- Famille C : Procédures de dimensionnement basées sur une inversion du modèle Bsim3v3 et sur un V_{EG} donné (où $V_{EG} = V_{GS} - V_{TH}$).
- Famille T : Procédures d'exploration du domaine de conception.

Toutes les procédures proposées par le générateur TR_MOS ainsi que leurs paramètres d'entrée et de sortie sont présentées dans le tableau 7.3.

7.2.2.3 Dessin des masques et paramètres de réalisation

Le dessin réalisé par le générateur TR_MOS est soumis aux paramètres de réalisation présentés dans le tableau 7.4.

Nom du paramètre	Description	Valeurs possibles	Valeur par défaut
sourceFirst	L'empilage commence par une diffusion de source	True / False	True
ring	Présence d'un anneau de garde autour du transistor	True / False	True

TAB. 7.4 : TR_MOS : Paramètres de réalisation.

La syntaxe d'un fichier de réalisation est donnée en annexe D.

Finalement, la figure 7.3 montre différents dessins des masques obtenus. La figure 7.3-a montre le résultat de la génération d'un transistor N de dimensions $W = 15\mu\text{m}$ et $L = 350\text{nm}$ sur une

Famille	Nom de la procédure	Paramètres d'entrée	Paramètres de sortie	Description
Z	direct_size	W, L, M*		Permet de spécifier directement les dimensions électriques du transistor
A	VTH(VGS, W, L)	W, L, VGS, VBS, VDS, Temp	VTH	Tension de seuil (V)
	VDSAT(VGS, W, L)		VDSAT	Tension de saturation (V)
	IDS(VGS, W, L)		IDS	Courant drain-source (A)
	REGION(VGS, W, L)		REGION	Régime de fonctionnement
	GM(VGS, W, L)		GM	Transconductance (A/V)
	GDS(VGS, W, L)		GDS	Conductance de sortie (A/V)
	GMB(VGS, W, L)		GMB	Conductance d'entrée (A/V)
	CAPA(VGS, W, L)		CGS, CGD, CGB, etc...	Toutes les capacités parasites (F)
B	W(VGS, L, IDS)	VGS, L, IDS, VBS, VDS, Temp	W	Largeur du transistor (m)
	L(VGS, W, IDS)	VGS, W, IDS, VBS, VDS, Temp	L	Longueur du transistor (m)
	W(VGS, L, GM)	VGS, L, GM, VBS, VDS, Temp	W	Largeur du transistor (m)
	L(VGS, W, GDS)	VGS, W, GDS, VBS, VDS, Temp	L	Longueur du transistor (m)
C	IDS(VEG, W, L)	VEG, W, L, VBS, VDS, Temp	IDS	Courant drain-source (A)
	W(VEG, L, IDS)	VEG, L, IDS, VBS, VDS, Temp	W	Largeur du transistor (m)
	L(VEG, W, IDS)	VEG, W, IDS, VBS, VDS, Temp	L	Longueur du transistor (m)
T	IRANGE	VDS, VBS, VGS, Temp	I_MIN, I_MAX	Courant drain-source min et max (A)

* Paramètre optionnel

TAB. 7.3 : TR_MOS : Procédures de dimensionnement et paramètres électriques.

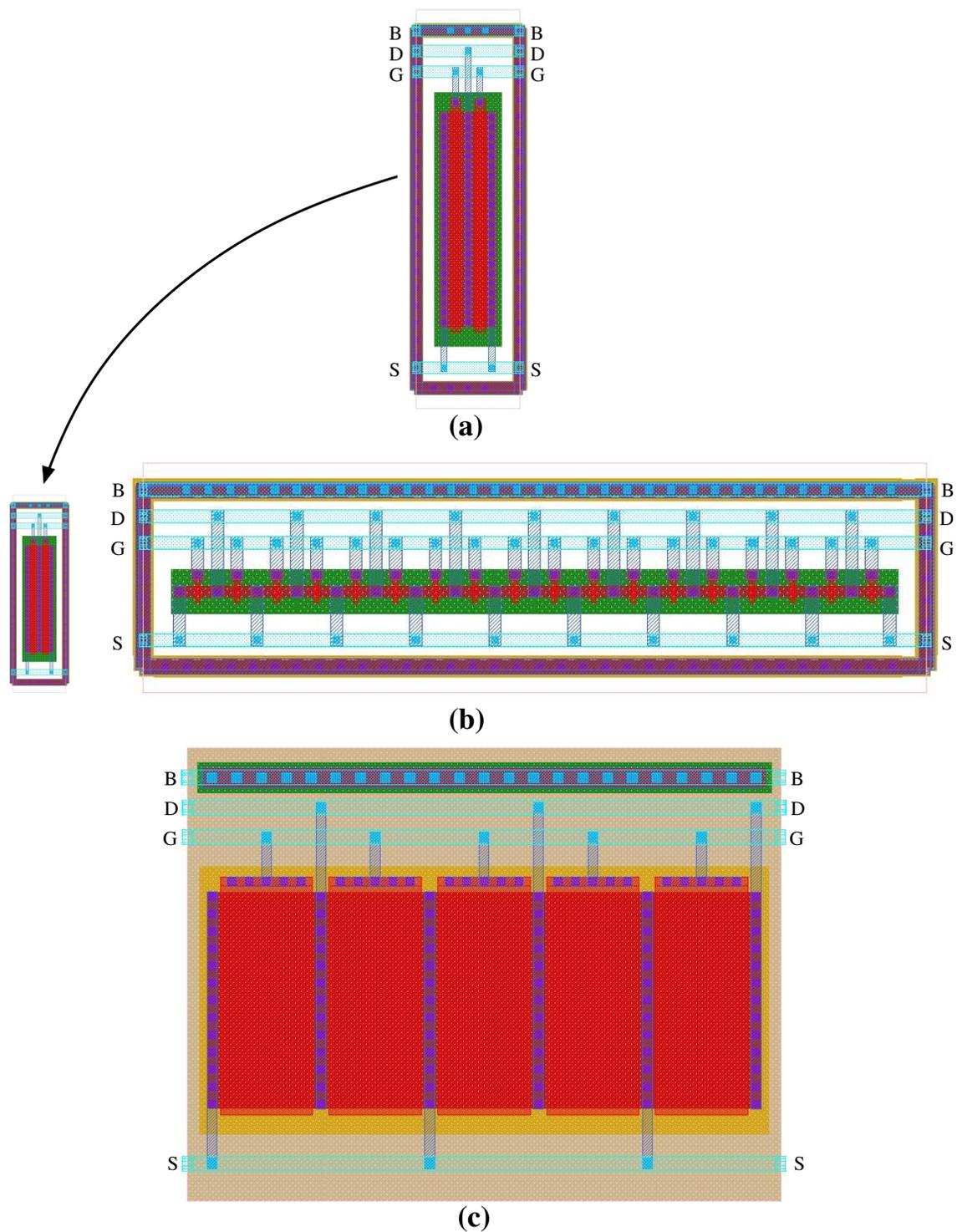


FIG. 7.3 : Différents dessins des masques réalisés par le dispositif TR.MOS : **a-** Type N avec anneau de garde, $W = 15\mu\text{m}$, $L = 350\text{nm}$, Technologie 120nm. **b-** Type N avec anneau de garde, $W = 15\mu\text{m}$, $L = 350\text{nm}$, Technologie 350nm. **c-** Type P, $W = 35\mu\text{m}$, $L = 3\mu\text{m}$, Technologie 180nm.

technologie $120nm$. La figure 7.3-b reprend ce dessin et le met à l'échelle du résultat obtenu sur une technologie $350nm$ pour les mêmes dimensions et la même contrainte de hauteur. La différence de surface flagrante entre ces deux figures est due principalement aux règles de dessin qui s'appliquent aux fils de routage. Ceux-ci sont plus larges et plus espacés. La distance entre les grilles est également plus grande du fait de règles, là aussi, plus contraignantes.

La figure 7.3-c illustre le résultat de la génération d'un transistor P de dimensions $W = 35\mu m$ et $L = 3.0\mu m$ sur une troisième technologie de $180nm$. Sur cette dernière figure, le nombre de repliements M a été fixé à 5 par l'utilisateur.

7.2.2.4 Fonction de forme

Pour les dispositifs issus du générateur TR_MOS, le nombre de formes est le nombre maximal de repliements W/W_{min} . Pour une technologie $120nm$, pour laquelle $\lambda = 120nm$, et avec $W = 3\mu m$, la fonction de forme obtenue est donnée par la figure 7.4-a. Sur cette figure, on a $L = L_{min}$. Dans ces conditions, on remarque que, contrairement à ce que nous avons annoncé au chapitre 5, la surface n'est pas constante, mais qu'elle augmente linéairement avec le nombre de repliements. En effet, au chapitre 5, nous n'avions pas pris en compte le routage du transistor. Ici, la fonction de forme est basée sur les dimensions de la boîte d'aboutement globale du transistor, à l'intérieur de laquelle on va trouver les pistes de routage. Celles-ci ajoutent une hauteur forfaitaire à chaque forme qui se traduit par une augmentation linéaire de la surface. Afin de mieux observer cette tendance, nous avons tracé la fonction de forme obtenue avec $W = 50\mu m$ sur la figure 7.4-b. La fonction de forme présente dans ce cas un nombre de points beaucoup plus important. L'augmentation de la surface peut alors devenir critique. Cependant, si l'on observe la figure 7.4-c qui montre la distribution du nombre de points de cette fonction de forme dans des intervalles de largeur, on remarque que 85% des points correspondent au premier tiers des largeurs possibles. Les formes présentant ces largeurs couvrent plus de 98% des hauteurs possibles. Ainsi, les chances de présenter une forme de largeur supérieure sont infimes. On peut donc conclure que la surface du transistor évolue peu pour les formes probables.

7.2.3 Générateurs de paires différentielles : DP_MOS_ID et DP_MOS_CC

Les générateurs DP_MOS_ID et DP_MOS_CC permettent de réaliser des paires différentielles. DP_MOS_ID inter-digite deux transistors dans un empilage, tandis que DP_MOS_CC génère un dessin à géométrie centrée.

7.2.3.1 Schéma électrique et paramètres structurels

Le schéma électrique réalisé par ces deux générateurs est le même. Il est soumis aux paramètres structurels du tableau 7.5.

La figure 7.5 donne deux exemples du schéma électrique réalisé pour deux jeux de paramètres structurels différents.

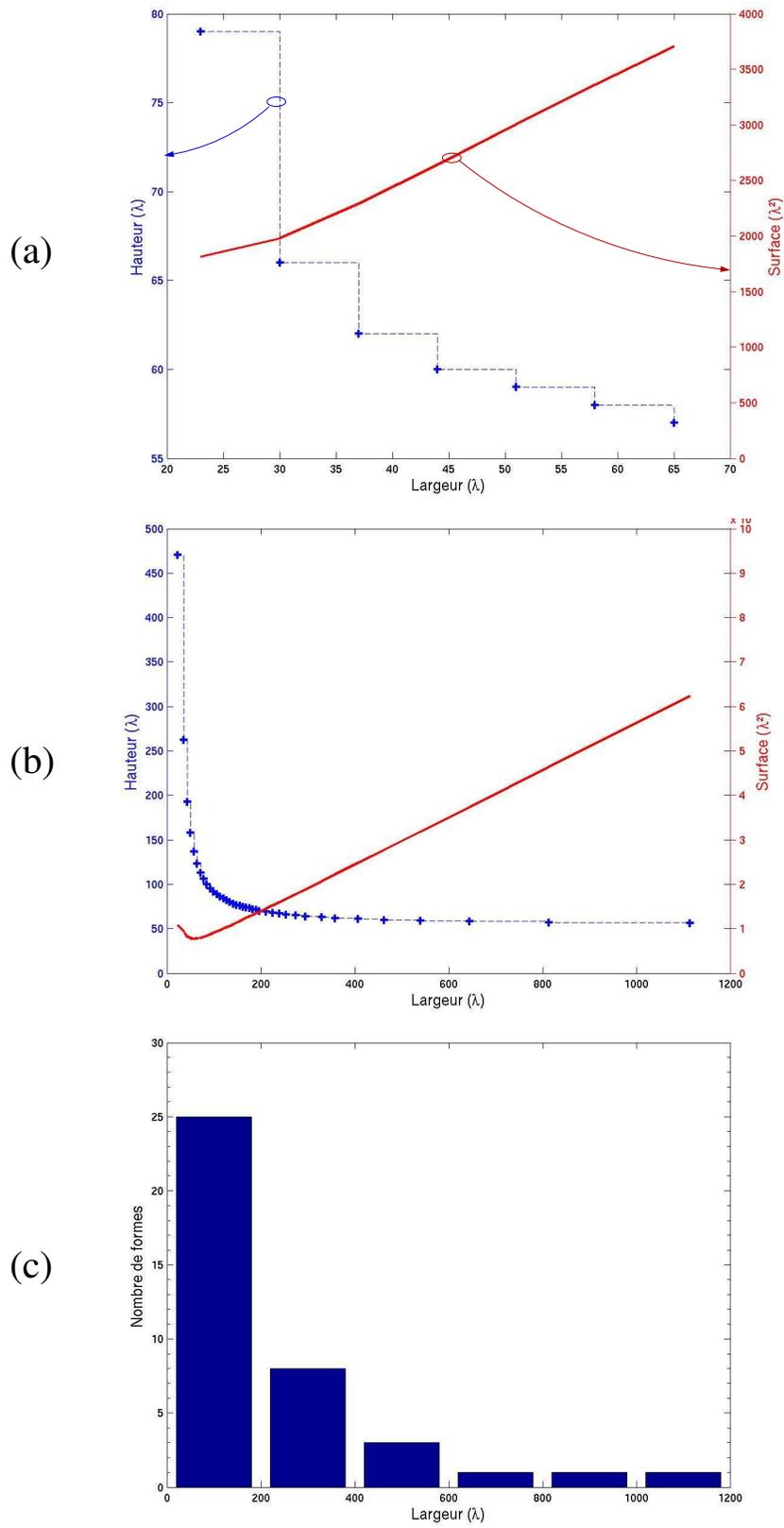


FIG. 7.4 : Fonctions de forme du dispositif TR_MOS, Technologie 120nm ($\lambda = 120\text{nm}$), $L = L_{min}$: **a-** $W = 3\mu\text{m}$. **b-** $W = 50\mu\text{m}$. **c-** Distribution des points de la fonction de forme pour $W = 50\mu\text{m}$.

Nom du paramètre	Description	Valeurs possibles
string tr1_name	Nom des transistors dans la vue structurale	Chaîne de caractères
string tr2_name		
int type	Type du transistor	TRANSN : Transistor N à faible courant de fuite TRANSP : Transistor P à faible courant de fuite TRANSN_FAST : Transistor N rapide (faible V_T) TRANSP_FAST : Transistor P rapide (faible V_T)
bool BS	Substrat polarisé à la source	True / False

TAB. 7.5 : DP_MOS_ID et DP_MOS_CC : Paramètres structurels.

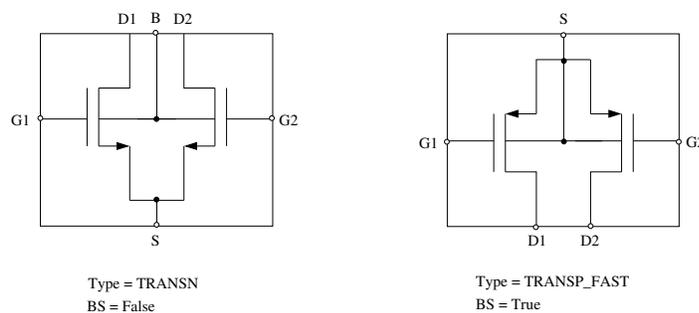


FIG. 7.5 : Schéma électrique des générateurs DP_MOS_xx.

7.2.3.2 Procédures de dimensionnement et paramètres électriques

Dans la mesure où les deux transistors d'une paire différentielle doivent être identiques, les procédures de dimensionnement sont les mêmes que celles proposées par le dispositif TR_MOS. Cependant, les techniques de dessin employées imposent des contraintes spécifiques sur le nombre de repliements M . En effet, puisque la source est partagée par les deux transistors, on va représenter physiquement les transistors par des paires de grilles (cf. chap. 5), en isolant ainsi les drains. Selon le générateur utilisé, on effectuera alors une inter-digitation de ces paires de grilles, ou une géométrie centrée. La technique d'inter-digitation est réalisable quel que soit le nombre de repliements. Si celui-ci est impair, il suffit de rajouter un doigt unique du transistor 1 à une extrémité l'empilage, et un doigt unique du transistor 2 à l'autre extrémité. En revanche, la géométrie centrée ne peut s'obtenir qu'en disposant d'un nombre pair de paires de grilles pour chaque transistor. Dès lors, le nombre de repliements dans ce cas doit être multiple de 4. Ceci est vérifié dans la procédure "direct_size" de la famille Z.

7.2.3.3 Dessin des masques et paramètres de réalisation

Les paramètres de réalisation des générateurs DP_MOS_ID et DP_MOS_CC sont donnés dans le tableau 7.6.

Nom du paramètre	Description	Valeurs possibles	Valeur par défaut
dummys	Présence de transistors fictifs aux extrémités de l'empilage	True / False	True
ring	Présence d'un anneau de garde autour du transistor	True / False	True

TAB. 7.6 : DP_MOS_ID et DP_MOS_CC : Paramètres de réalisation.

Quelques exemples de dessins des masques obtenus sont montrés sur la figure 7.6. La figure 7.6-a montre le résultat de la génération d'une paire différentielle de type N par le dispositif DP_MOS_ID en technologie 120nm. Les dimensions de chaque transistor sont $W = 25\mu m$ et $L = 340nm$. La figure 7.6-b montre le résultat obtenu pour les mêmes paramètres avec le dispositif DP_MOS_CC et avec un anneau de garde. Enfin, la figure 7.6-c illustre la génération d'une paire différentielle de type P en technologie 350nm avec le dispositif DP_MOS_CC. Les transistors représentés sur cette figure ont pour dimensions $W = 15\mu m$ et $L = 2\mu m$.

7.2.3.4 Fonction de forme

La figure 7.7 donne les fonctions de forme réalisées par les dispositifs DP_MOS_ID et DP_MOS_CC pour $W = 25\mu m$ et $L = 350nm$ sur une technologie 350nm pour laquelle $\lambda = 325nm$. En observant cette figure, on s'aperçoit que la surface augmente avec le nombre de repliements de façon beaucoup plus importante que pour le transistor simple. Cela s'explique par le forfait de surface plus important introduit ici par le routage. En effet, puisque chaque dispositif contient deux transistors, le routage est plus complexe et prend donc plus d'espace que pour un transistor unique. Ainsi, bien que le dispositif DP_MOS_CC, qui propose un dessin à géométrie centrée, soit meilleur en termes d'appariement, il souffre d'un nombre de formes moins important que le dispositif DP_MOS_ID dans la mesure où il se limite aux nombres de repliements multiples de 4. Pour ce dispositif, il est beaucoup plus probable d'obtenir une forme dont la surface est pénalisante. C'est pourquoi le dispositif DP_MOS_ID, caractérisé par appariement moins optimal, peut se révéler suffisant très souvent. Ce choix est sous la responsabilité du concepteur de module.

7.2.4 Générateurs de miroirs de courant : CM_MOS_ID et CM_MOS_CC

Les générateurs CM_MOS_ID et CM_MOS_CC réalisent des miroirs de courant simples, c'est-à-dire composés de deux transistors identiques. Ils génèrent respectivement une inter-digitation des transistors, et une géométrie centrée.

Le miroir de courant et la paire différentielle étant des montages basés tous deux sur une paire de transistors identiques, les générateurs de ces deux dispositifs proposent les mêmes paramètres.

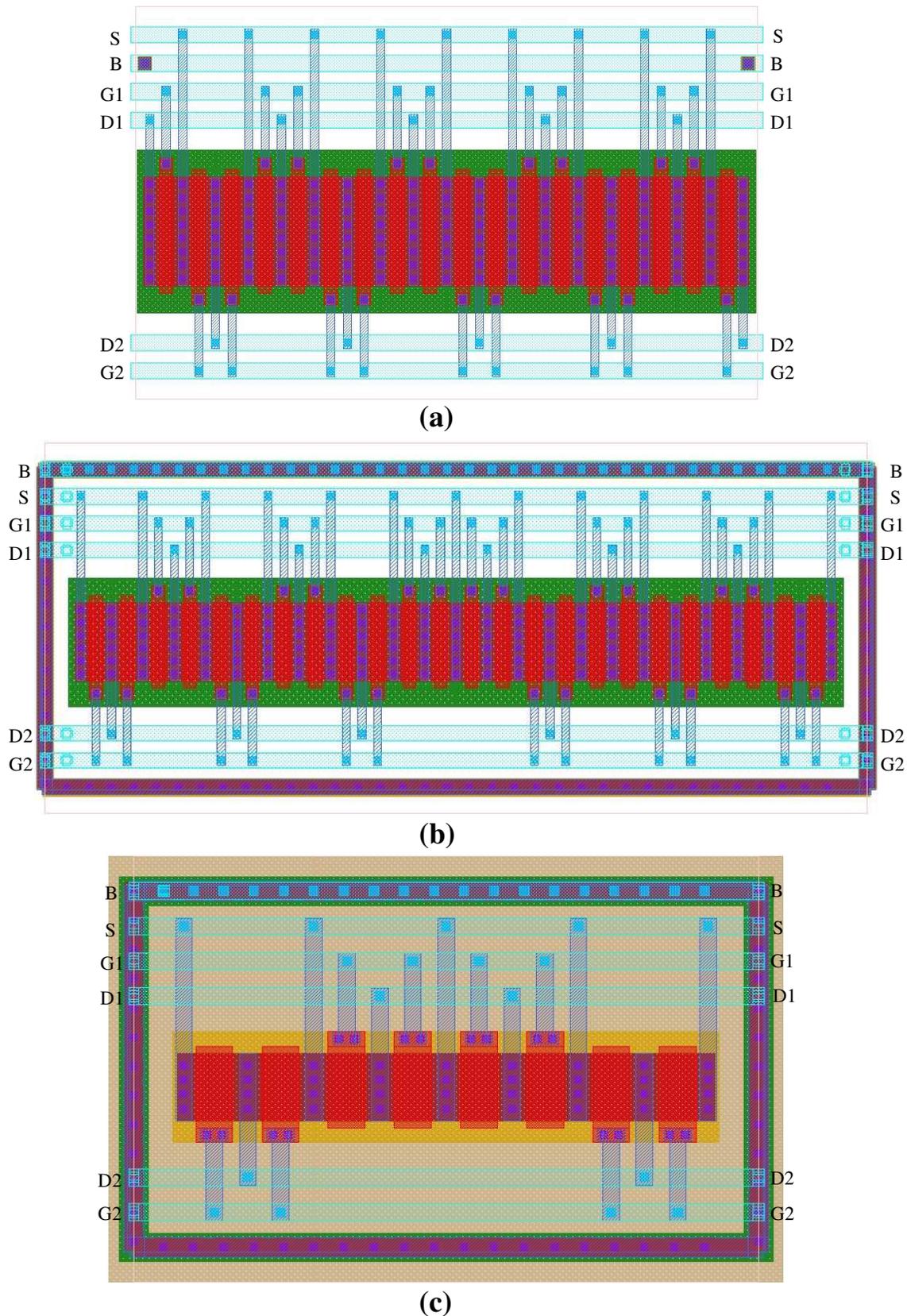


FIG. 7.6 : Différents dessins des masques réalisés par les dispositifs DP_MOS_ID et DP_MOS_CC : **a-** DP_MOS_ID, type N, $W = 25\mu\text{m}$, $L = 340\text{nm}$, Technologie 120nm. **b-** DP_MOS_CC, type N avec anneau de garde, $W = 25\mu\text{m}$, $L = 340\text{nm}$, Technologie 120nm. **c-** DP_MOS_CC, type P avec anneau de garde, $W = 15\mu\text{m}$ et $L = 2\mu\text{m}$, Technologie 350nm .

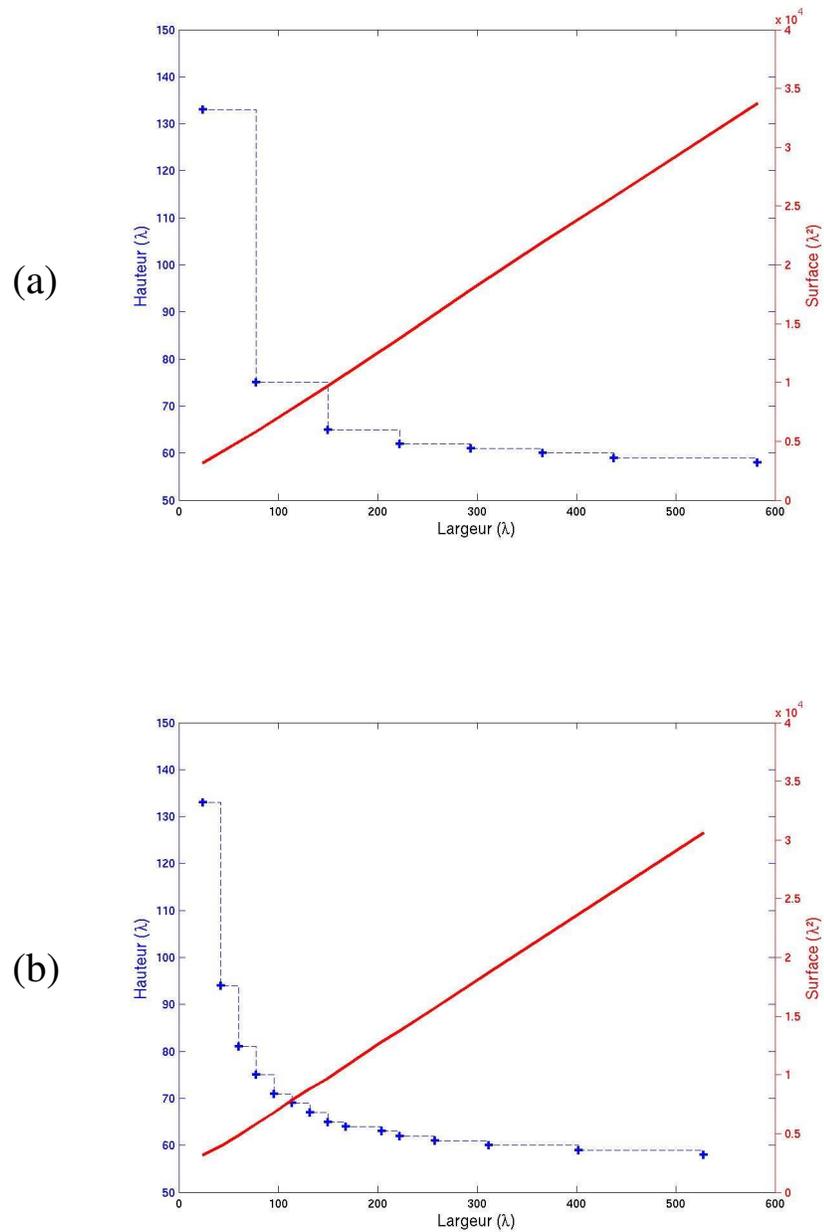


FIG. 7.7 : Fonctions de forme d'une paire différentielle en technologie 350nm pour $W = 25\mu\text{m}$ et $L = 350\text{nm}$ ($\lambda = 325\text{nm}$) : **a-** DP_MOS_CC **b-** DP_MOS_ID .

7.2.4.1 Schéma électrique et paramètres structurels

Les paramètres structurels sont ici les mêmes que ceux des générateurs de paires différentielles (cf. section 7.2.3.1). Deux exemples de schéma électrique réalisé par les générateurs `CM_MOS_CC` et `CM_MOS_ID` sont illustrés sur la figure 7.8.

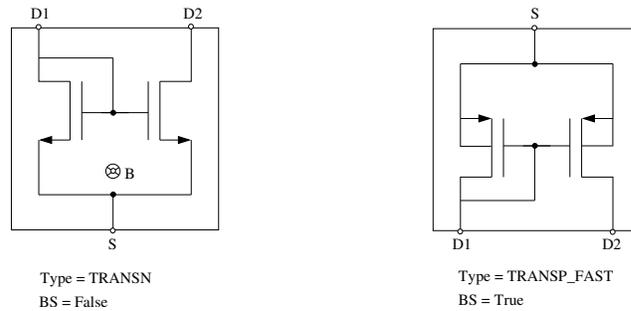


FIG. 7.8 : Schéma électrique des dispositifs `CM_MOS_xx`.

7.2.4.2 Procédures de dimensionnement et paramètres électriques

Les deux transistors du miroir de courant simple sont identiques et il suffit d'en dimensionner un pour avoir les dimensions électriques de l'autre. Les générateurs `CM_MOS_xx` proposent des procédures similaires à celles du générateur `TR_MOS` à ceci près qu'ici $V_{GD} = 0$ et donc $V_{DS} = V_{GS}$. Le générateur `CM_MOS_CC`, qui aboutit à une géométrie centrée, impose un nombre de repliements M limité aux multiples de 4.

7.2.4.3 Dessin des masques et paramètres de réalisation

Les paramètres de réalisation conditionnant le dessin des miroirs de courant sont les mêmes que ceux intervenant dans la génération d'une paire différentielle.

La figure 7.9 montre des dessins de masques produits par les générateurs de miroirs de courant. Les conditions de génération correspondent à celles qui avaient été utilisées pour la génération des paires différentielles de la figure 7.6.

La fonction de forme obtenue pour les miroirs de courant est également très semblable à celle obtenue pour les paires différentielles. En effet, seul le routage est légèrement modifié entre ces deux dispositifs.

7.2.5 Générateurs de matrices de transistors : `MULTI_MOS`

Le générateur `MULTI_MOS` réalise un ensemble de plusieurs transistors qui ne partagent que le connecteur de substrat, et éventuellement la grille. Il permet d'optimiser leur appariement en les arrangeant dans une matrice dont la géométrie peut être centrée. Pour cela, chaque transistor est

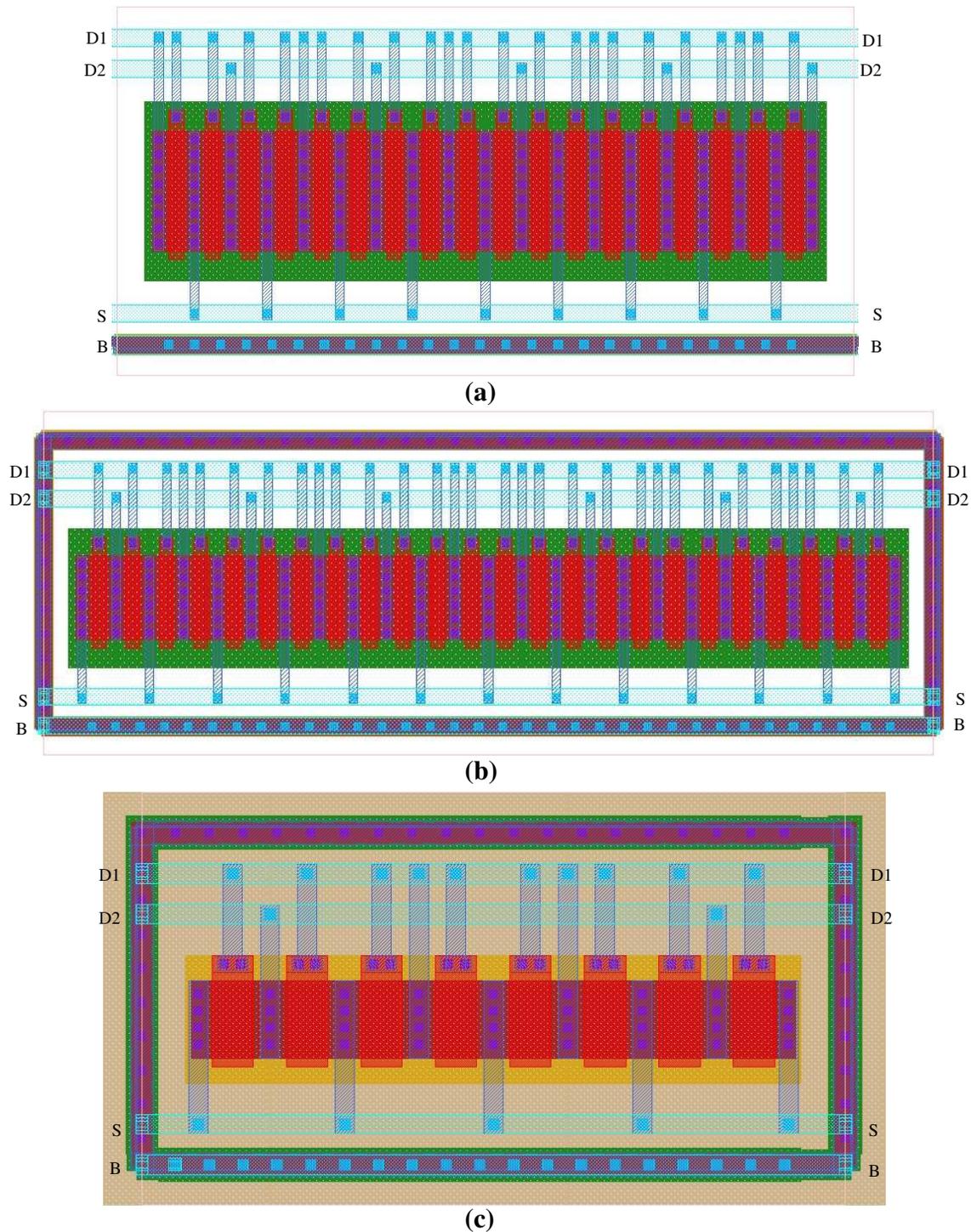


FIG. 7.9 : Différents dessins des masques réalisés par les dispositifs CM_MOS_ID et CM_MOS_CC : **a-** CM_MOS_ID, type N, $W = 25\mu\text{m}$, $L = 340\text{nm}$, Technologie 120nm. **b-** CM_MOS_CC, type N avec anneau de garde, $W = 25\mu\text{m}$, $L = 340\text{nm}$, Technologie 120nm. **c-** CM_MOS_CC, type P avec anneau de garde, $W = 15\mu\text{m}$ et $L = 2\mu\text{m}$, Technologie 350nm .

divisé en sous-transistors. Les nombres respectifs de sous-transistors déterminent les rapports de largeur que l'on souhaite obtenir entre les transistor.

A l'heure actuelle, l'utilisation de ce générateur est particulière puisque le nombre de sous-transistors de chaque transistor, ainsi que leur placement dans la matrice, sont des paramètres de réalisation **obligatoires**. Ceci offre la plus grande liberté possible au concepteur qui peut alors fixer :

- les rapports qu'il désire entre les différents transistors du schéma électrique.
- le placement des sous-transistors dans la matrice et assurer ainsi l'appariement nécessaire.

A terme, on envisage d'intégrer à ce générateur l'algorithme de placement présenté dans [Sayed02]. La déformation de ce dispositif se fait en repliant les sous-transistors.

7.2.5.1 Schéma électrique et paramètres structurels

Le schéma électrique visé ici représente N transistors. Les paramètres structurels du générateur MULTI_MOS sont présentés dans le tableau 7.7. La figure 7.10 montre deux schémas électriques

Nom du paramètre	Description	Valeurs possibles
int N	Nombre de transistors dans le schéma électrique	[1-9]
bool sameGrid	Grille partagée	True / False
int type	Type du transistor	TRANSN : Transistor N à faible courant de fuite TRANSP : Transistor P à faible courant de fuite TRANSN_FAST : Transistor N rapide (faible V_T) TRANSP_FAST : Transistor P rapide (faible V_T)

TAB. 7.7 : MULTI_MOS : Paramètres structurels.

réalisés par le générateur MULTI_MOS pour différents jeux de paramètres structurels.

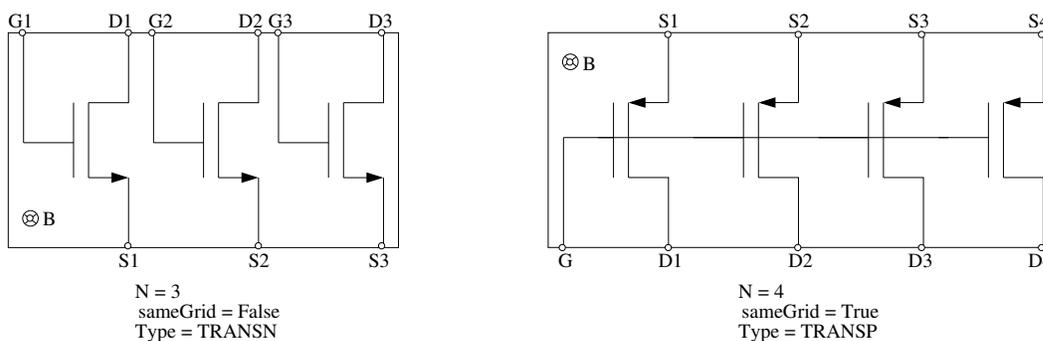


FIG. 7.10 : Schéma électrique du dispositif MULTI_MOS.

7.2.5.2 Procédures de dimensionnement et paramètres électriques

Les rapports entre les transistors du schéma électrique sont donnés dans le fichier de réalisation dans lequel le concepteur spécifie les différents nombres de sous-transistors. Le dimensionnement de ce dispositif correspond au dimensionnement d'un sous-transistor. Il propose pour cela les mêmes procédures que le générateur de transistors simples.

Lorsqu'un algorithme de placement sera intégré à ce générateur, on pourra envisager l'écriture de procédures plus complexes permettant de dimensionner les transistors pour des rapports de courant donnés. La difficulté sera de déduire les rapports entre les dimensions des transistors à partir des rapports de courants visés. En effet, ici, tous les transistors ont des connecteurs de drain a priori distincts et peuvent avoir différentes tensions V_{DS} .

7.2.5.3 Dessin des masques et paramètres de réalisation

Le dessin des masques généré ici est donc largement conditionné par le fichier de réalisation. La syntaxe de ce fichier est décrite dans l'annexe D. Le tableau 7.8 résume les paramètres qui doivent y être spécifiés.

Nom du paramètre	Description
placement	Placement relatif des sous-transistors dans la matrice
routing	Placement relatif des fils de routage dans les canaux

TAB. 7.8 : MULTI_MOS : Paramètres de réalisation (cf. annexe D).

La figure 7.11 présente un dessin des masques obtenu pour 4 transistors de type N, tels que $W_1 = 2.W_2 = 4.W_3 = 4.W_4$. Les dimensions des sous-transistors sont $W = 9\mu m$ et $L = 500nm$. Leur placement respectif est indiqué sur la figure. Le nombre de repliements des sous-transistors a été fixé à $M = 3$. La technologie cible est une technologie $120nm$.

La figure 7.12 montre le dessin des masques obtenu pour 2 transistors P identiques de dimensions $W = 24\mu m$ et $L = 1.4\mu m$ en technologie $350nm$. Dans cet exemple, la grille est commune aux deux transistors. Chaque transistor est donc construit avec 4 sous-transistors, on a alors $W_{SUB} = 6\mu m$. Par ailleurs, le nombre de repliements des sous-transistors a été fixé à $M = 2$.

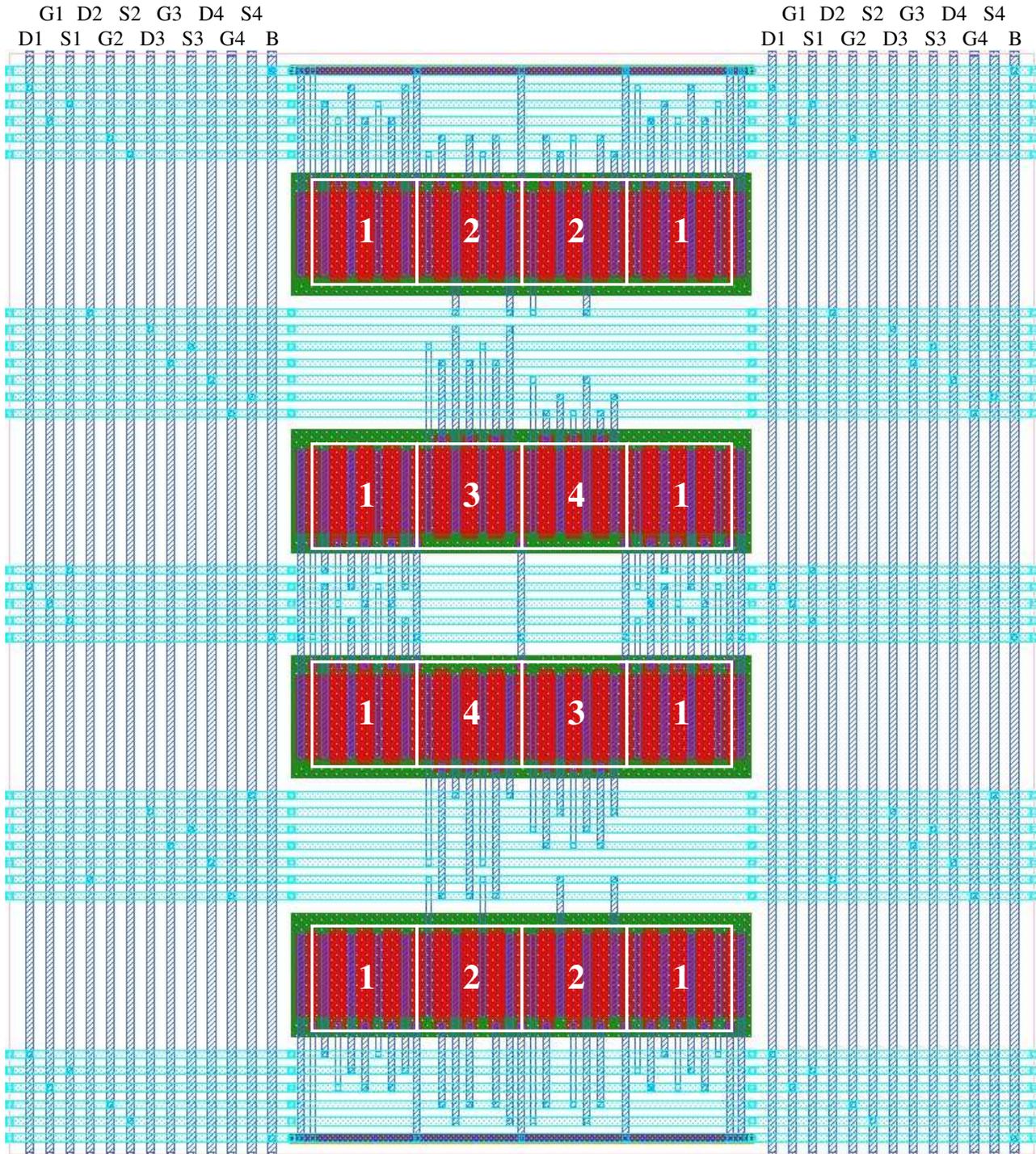


FIG. 7.11 : Dessin des masques du dispositif MULTI_MOS : $W_{SUB} = 9\mu m$, $L = 500nm$, $M_{SUB} = 3$, technologie 120nm.

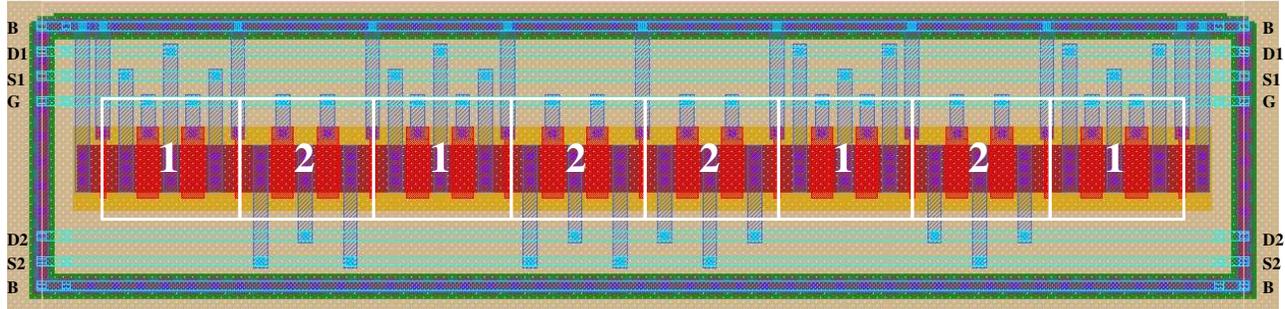


FIG. 7.12 : Dessin des masques du dispositif MULTI_MOS pour 2 transistors P identiques. $W_{SUB} = 6\mu m$, $L = 1.4\mu m$, $M_{SUB} = 2$, technologie 350nm.

7.2.5.4 Fonction de forme

La déformation se fait ici en repliant les sous-transistors. Pour une réalisation donnée, tous les sous-transistors ont le même nombre de repliements. La figure 7.13 donne la fonction de forme caractérisant le dispositif dans les conditions de la figure 7.12 ($\lambda = 325nm$).

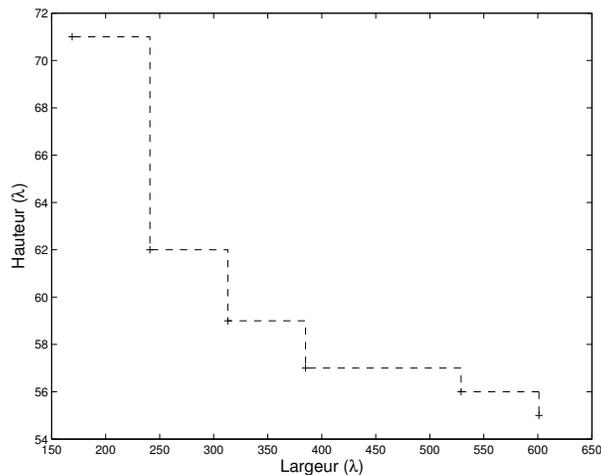


FIG. 7.13 : Fonction de forme du dispositif MULTI_MOS pour les conditions de la figure 7.12 ($\lambda = 325nm$).

7.2.6 Générateur de matrices de transistors à diffusion commune : MULTI_MOS_SHD

Le générateur MULTI_MOS_SHD réalise un ensemble de plusieurs transistors qui partagent soit la source, soit le drain. Il s'utilise comme le générateur MULTI_MOS. Cependant, le partage d'une diffusion permet un dessin des masques nettement plus compact et optimisé vis-à-vis des capacités parasites.

7.2.6.1 Schéma électrique et paramètres structurels

Comme le montre le tableau 7.9, on retrouve naturellement pour ce générateur les paramètres structurels de MULTI_MOS. Il propose également un paramètre `sharedDrain` qui permet de partager le drain, plutôt que la source.

Nom du paramètre	Description	Valeurs possibles
int N	Nombre de transistors dans le schéma électrique	[1-9]
bool sameGrid	Grille partagée	True / False
bool sharedDrain	Partage du drain	True / False
int type	Type du transistor	TRANSN : Transistor N à faible courant de fuite TRANSP : Transistor P à faible courant de fuite TRANSN_FAST : Transistor N rapide (faible V_T) TRANSP_FAST : Transistor P rapide (faible V_T)

TAB. 7.9 : MULTI_MOS_SHD : Paramètres structurels.

Deux exemples de schéma électrique réalisé par le générateur MULTI_MOS_SHD sont illustrés sur la figure 7.14.

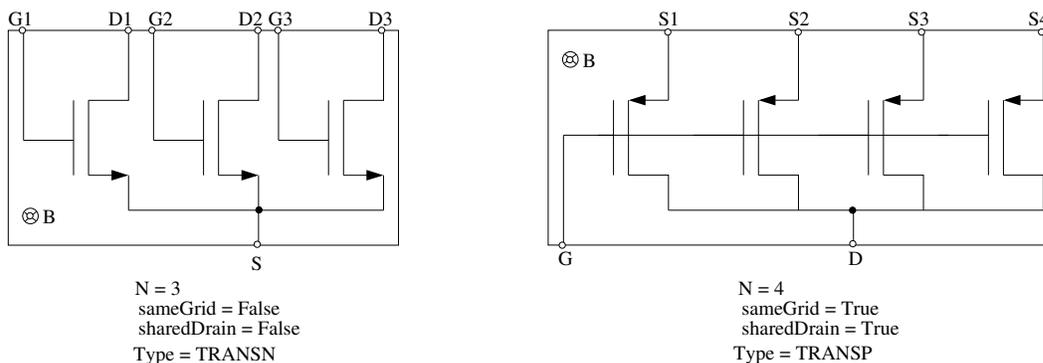


FIG. 7.14 : Schéma électrique du dispositif MULTI_MOS_SHD.

7.2.6.2 Procédures de dimensionnement et paramètres électriques

La méthode de dimensionnement est la même que celle du générateur MULTI_MOS. En revanche, le partage d'une diffusion implique que le dessin des masques soit construit autour de paires de grilles. Le nombre de repliement est donc limité aux multiples de 4.

7.2.6.3 Dessin des masques et paramètres de réalisation

Ce générateur s'appuie sur les mêmes paramètres de réalisation que le générateur MULTI.MOS. La figure 7.15 montre le dessin obtenu pour 4 transistors partageant la source dans les conditions de la figure 7.11.

7.3 Bibliothèque libMIMCAP

La bibliothèque libMIMCAP contient les générateurs de capacités MIM. Nous avons détaillé dans le chapitre précédent la conception du dispositif BICAPA qui prend en charge la réalisation de deux capacités appariées. Ce dispositif ne sera pas davantage présenté ici. On trouvera également dans cette bibliothèque un générateur de capacité unique et un générateur de matrice de capacités appariées.

7.3.1 Générateurs de motifs pour les dispositifs capacitifs

Il a été plus difficile de mutualiser les générateurs de motifs nécessaires au dessin des différents dispositifs de cette bibliothèque. En effet, le dessin d'une capacité unique est constitué du seul motif illustré sur la figure 7.16. On va chercher ici à optimiser le dessin de ce motif unique. En revanche, le dessin d'une matrice de capacités s'appuie sur un ensemble de motifs dédiés qui permettent de représenter :

- Une capacité fictive (fig. 7.17-a).
- Une capacité unitaire (fig. 7.17-b).
- Une capacité non unitaire obtenue par la technique du rectangle (fig. 7.17-c).
- Une capacité non unitaire obtenue par la technique du stub (fig. 7.17-d).
- Une capacité non unitaire obtenue par la technique du trou (fig. 7.17-e).

Trois générateurs de motifs sont alors nécessaires :

- Le générateur `mim_capa` réalise le motif d'une capacité unique.
- Le générateur `mim_cu` permet de réaliser une capacité unitaire, ou une capacité fictive.
- Le générateur `mim_ncu` permet de réaliser les capacités non-unitaires. Selon les conditions, il utilisera par ordre de priorité : la technique du rectangle, celle du stub, ou celle du trou.

7.3.1.1 Paramètres standard

Les trois générateurs de motifs capacitifs proposent évidemment le paramètre standard `modelName`. Par ailleurs, ils se basent tous sur la valeur de la capacité élémentaires `Celem`. Le générateur `mim_ncu` s'appuie également sur le rapport `R` qu'il doit réaliser, dont la valeur est comprise entre 1 et 2.

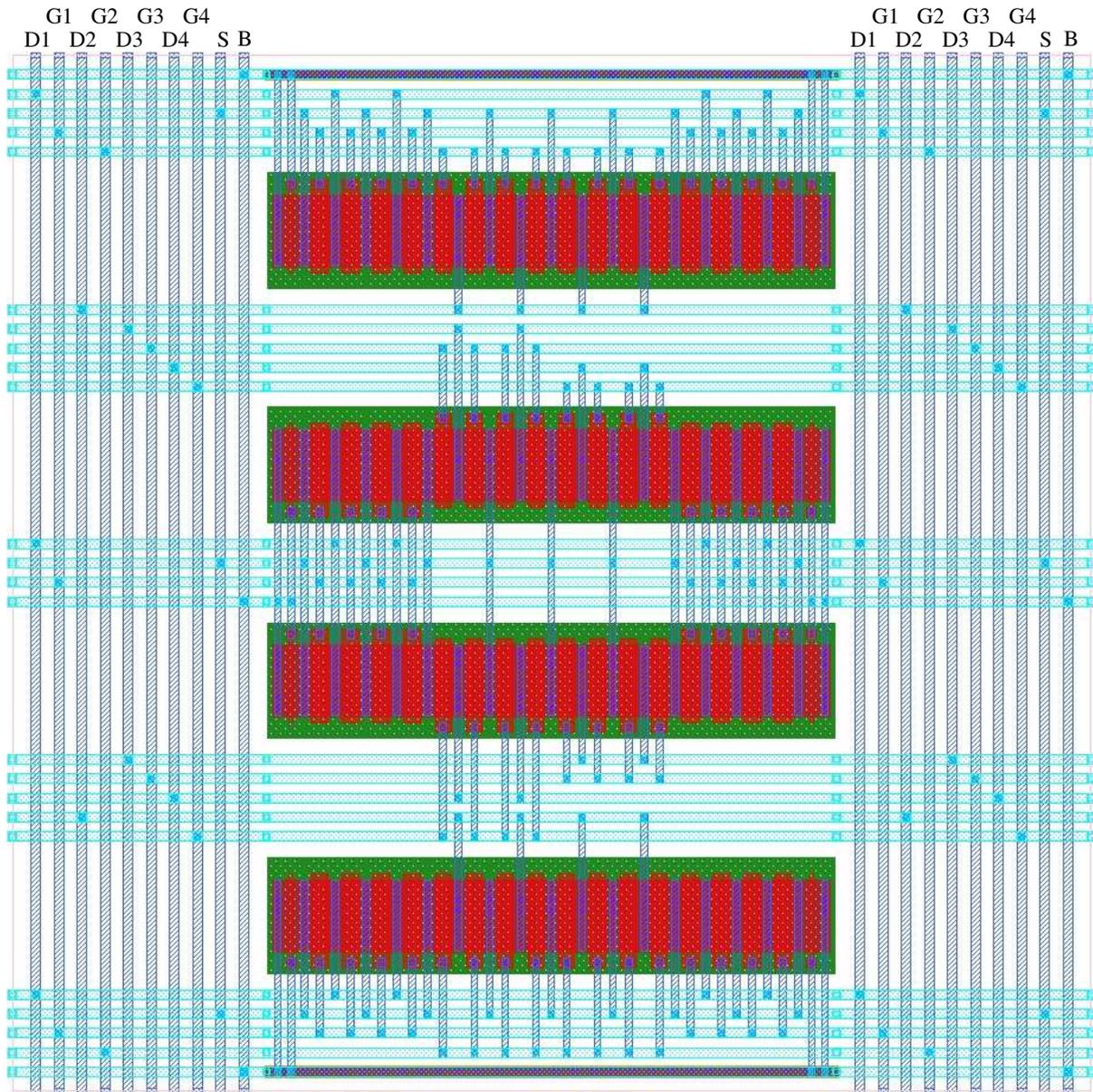


FIG. 7.15 : Dessin des masques du dispositif `MULTI_MOS_SHD` pour $W_{SUB} = 9\mu m$, $L = 500nm$, $M_{SUB} = 4$, technologie 120nm.

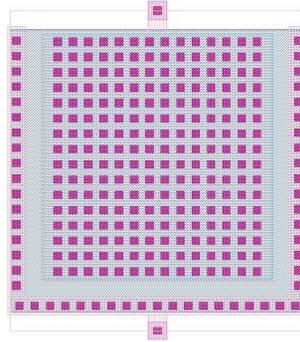


FIG. 7.16 : Motif pour la réalisation de capacités uniques.

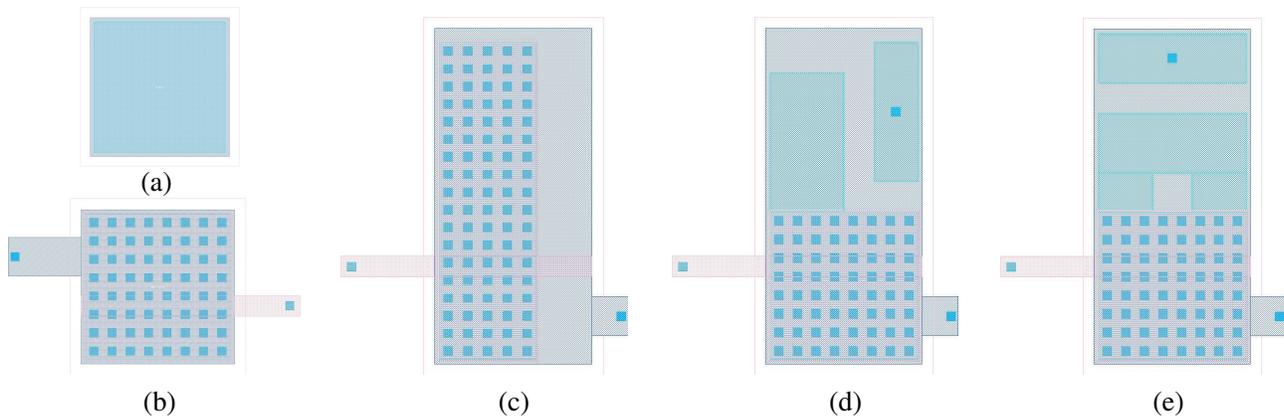


FIG. 7.17 : Motifs utilisés pour les matrices de capacités : (a) Capacité fictive - (b) Capacité unitaire - (c)(d)(e) Capacités non-unitaires (resp. rectangle, stub et trou) .

7.3.1.2 Paramètres contextuels

Puisque le motif généré par `mim_capa` n'est pas destiné à un aboutement, il ne dépend pas du contexte. En revanche, les générateurs `mim_cu` et `mim_ncu` proposent des paramètres contextuels qui permettent aux motifs générés de s'intégrer dans différents types de matrices. Les matrices obtenues diffèrent principalement par l'orientation des canaux de routage, et le nombre de pistes traversant chacun de ces canaux. Les prolongements d'armatures doivent pouvoir s'adapter à la configuration des canaux pour assurer la connexion électrique. Ils sont donc conditionnés par les paramètres contextuels suivants :

- `orientation` : Ce paramètre est un caractère et indique la face sur laquelle apparaît le prolongement de l'armature haute. Celui de l'armature basse se situe toujours sur la face opposée. Les valeurs possibles sont : 'N', 'S', 'E' et 'O'.
- `track_top` : Ce paramètre entier indique la piste dans le canal sur laquelle doit se connecter le prolongement d'armature haute.
- `track_bottom` : Ce paramètre entier indique la piste dans le canal sur laquelle doit se connecter le prolongement d'armature basse.

7.3.1.3 Paramètres de sortie

Tous les générateurs de motifs que nous présentons ici rempliront les champs suivants de la structure de retour :

- `DXAB` et `DYAB` : Dimensions symboliques de la boîte d'aboutement du motif.
- `DXAB_mu` et `DYAB_mu` : Dimensions réelles de la boîte d'aboutement du motif.
- `C` : Valeur de la capacité effectivement réalisée.
- `C_top`, `C_bottom` : Capacités parasites entre les armatures et le substrat.

7.3.1.4 Tableau récapitulatif

Le tableau 7.10 résume finalement les paramètres des générateurs de motifs pour les dispositifs capacitifs.

Nom du générateur	Paramètres standard	Paramètres contextuels	Informations rendues
<code>mim_capa</code>	<code>string modelName</code>		<code>DXAB DYAB DXAB_mu DYAB_mu</code>
<code>mim_cu</code>	<code>double Celem</code>	<code>char orientation</code>	<code>C_top C_bottom</code>
<code>mim_ncu</code>	<code>idem + double R</code>	<code>int track_top</code> <code>int track_bottom</code>	<code>C</code>

TAB. 7.10 : *libMIMCAP* : Paramètres des générateurs de motifs.

7.3.2 Générateur de capacités MIM : CMIM

Le générateur CMIM réalise une capacité MIM unique. Comme nous l'avons vu, il est basé sur un seul motif produit par le générateur `mim_capa`.

7.3.2.1 Schéma électrique et paramètres structurels

Le schéma électrique réalisé est donné par la figure 7.18. Le seul paramètre structurel est le nom `Cname` de la capacité dans la vue structurelle.

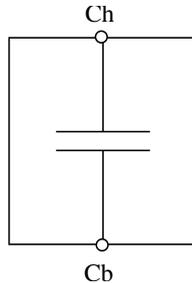


FIG. 7.18 : Schéma électrique du dispositif CMIM.

7.3.2.2 Procédures de dimensionnement et paramètres électriques

Le tableau 7.11 recense les procédures et les paramètres électriques du générateur CMIM.

Nom de la procédure	Paramètres d'entrée	Paramètres de sortie	Description
<code>direct_size</code>	C		Valeur de la capacité (F)
<code>get_C</code>		C	Valeur de la capacité réalisée (F)
<code>get_Ch</code>		CH	Valeur de la capacité parasite entre l'armature haute et le substrat (F)
<code>get_Cb</code>		CB	Valeur de la capacité parasite entre l'armature basse et le substrat (F)

TAB. 7.11 : CMIM : Procédures de dimensionnement et paramètres électriques.

7.3.2.3 Dessin des masques et paramètres de réalisation

Le dispositif CMIM ne propose pas de paramètre de réalisation. La figure 7.19 montre deux formes obtenues pour une capacité de 1pf en technologie 350nm . La figure 7.20 montre un dessin obtenu pour la même capacité en technologie 120nm . Le nombre important de vias sur les armatures permet de réduire la résistance d'accès à ces armatures.

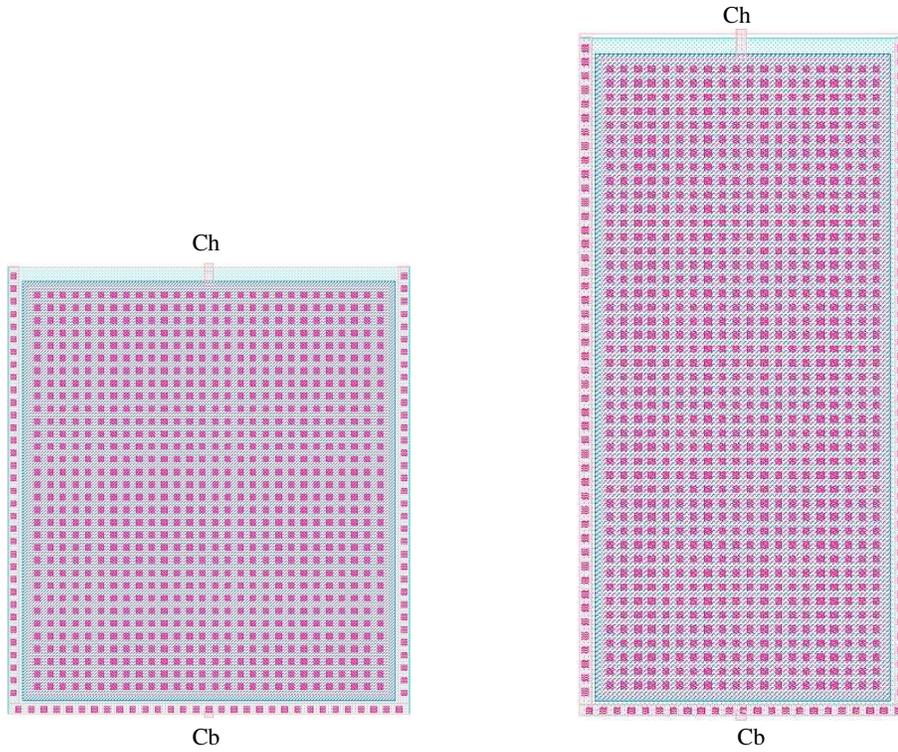


FIG. 7.19 : Dessins des masques du dispositif CMIM. $C = 1\text{pf}$, technologie 350nm.

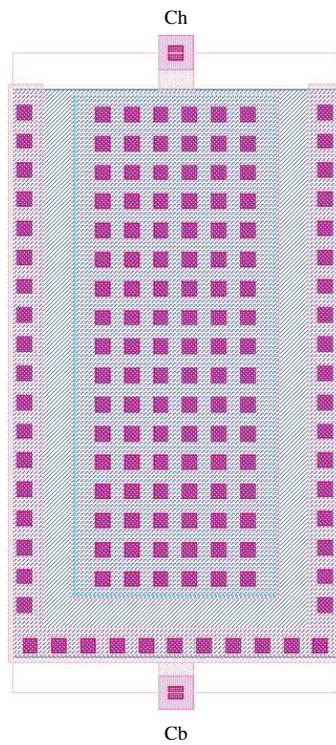


FIG. 7.20 : Dessins des masques du dispositif CMIM. $C = 1\text{pf}$, technologie 120nm.

7.3.2.4 Fonction de forme

La déformation se fait ici en envisageant différents facteurs de forme pour les armatures rectangulaires de manière à obtenir la capacité désirée. On aurait pu imaginer incrémenter successivement la largeur de l'armature haute d'un pas de grille, et de calculer la hauteur correspondante. Cette technique a été rejetée car elle aurait conduit à un nombre de forme trop grand, et ainsi à un temps de calcul pénalisant. On a donc fixé arbitrairement le nombre de forme à 50. Ainsi, on commence par calculer les deux formes extrêmes, c'est-à-dire celle qui correspond à la largeur minimale et celle qui correspond à la hauteur minimale. Puis on réalise 50 formes uniformément réparties entre ces deux extrêmes. Dans ces conditions, la fonction de forme obtenue dans les conditions de la figure 7.20 est présentée sur la figure 7.21. Pour mémoire, en technologie 120nm, on a $\lambda = 120nm$.

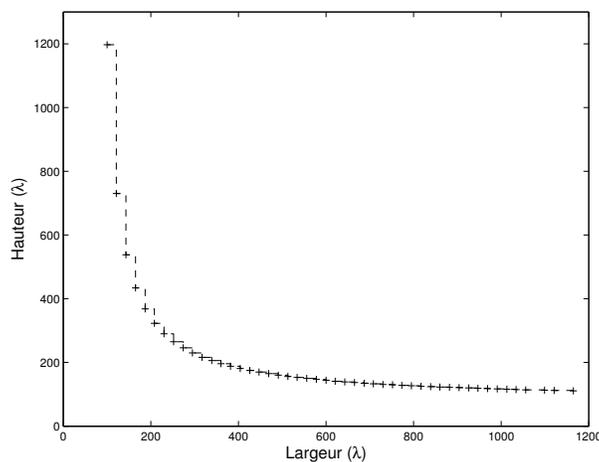


FIG. 7.21 : Fonction de forme du dispositif CMIM ($\lambda = 120nm$).

7.3.3 Générateurs de matrices de capacités MIM : MIM_MATRIX

Le générateur MIM_MATRIX réalise plusieurs capacités MIM dont l'appariement peut être optimisé. Pour cela, chaque capacité est physiquement représentée par un ensemble de capacités unitaires et non-unitaires, mélangées dans une matrice. A l'heure actuelle, les rapports capacitifs visés ainsi que le placement relatif des capacités élémentaires dans la matrice sont des paramètres de réalisation obligatoires.

7.3.3.1 Schéma électrique et paramètres structurels

Le schéma électrique réalisé contient N capacités. N est le seul paramètre structurel proposé par le dispositif MIM_MATRIX. La figure 7.22 donne le schéma électrique réalisé pour $N = 3$.

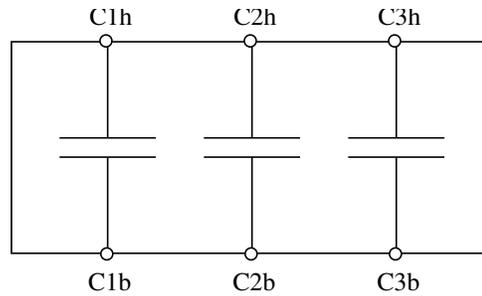


FIG. 7.22 : Schéma électrique du dispositif MIM_MATRIX pour $N=3$.

7.3.3.2 Procédures de dimensionnement et paramètres électriques

A l'heure actuelle, ce générateur ne propose qu'une seule procédure de dimensionnement `direct_size` qui permet de spécifier la valeur de la capacité élémentaire.

7.3.3.3 Dessin des masques et paramètres de réalisation

Le tableau 7.12 donne les paramètres de réalisation du générateur MIM_MATRIX.

Nom du paramètre	Description	Valeurs possibles	Valeur par défaut
placement	Placement relatif des capacités élémentaires dans la matrice	cf. annexe D	NEANT
ratios	Rapports capacitifs visés		

TAB. 7.12 : MIM_MATRIX : Paramètres de réalisation.

Du fait de la présence éventuelle de capacités non-unitaires dans la matrice, les rapports ne peuvent pas être ici directement déduits du placement. Ils doivent donc être explicitement précisés grâce au paramètre `ratios`. La cohérence entre le placement et les rapports spécifiés est cependant vérifiée.

La figure 7.23 montre un exemple de dessin obtenu pour trois capacités telles que $C1 = 1.47.Celem$, $C2 = 4.Celem$, $C3 = 6.Celem$. La génération a eu lieu pour $Celem = 0,6pf$ et en technologie $120nm$.

7.4 Bibliothèque libPOLYRES

La bibliothèque libPOLYRES contient deux générateurs de dispositifs résistifs. Le premier réalise une résistance polysilicium simple, le second deux résistances polysilicium appariées.

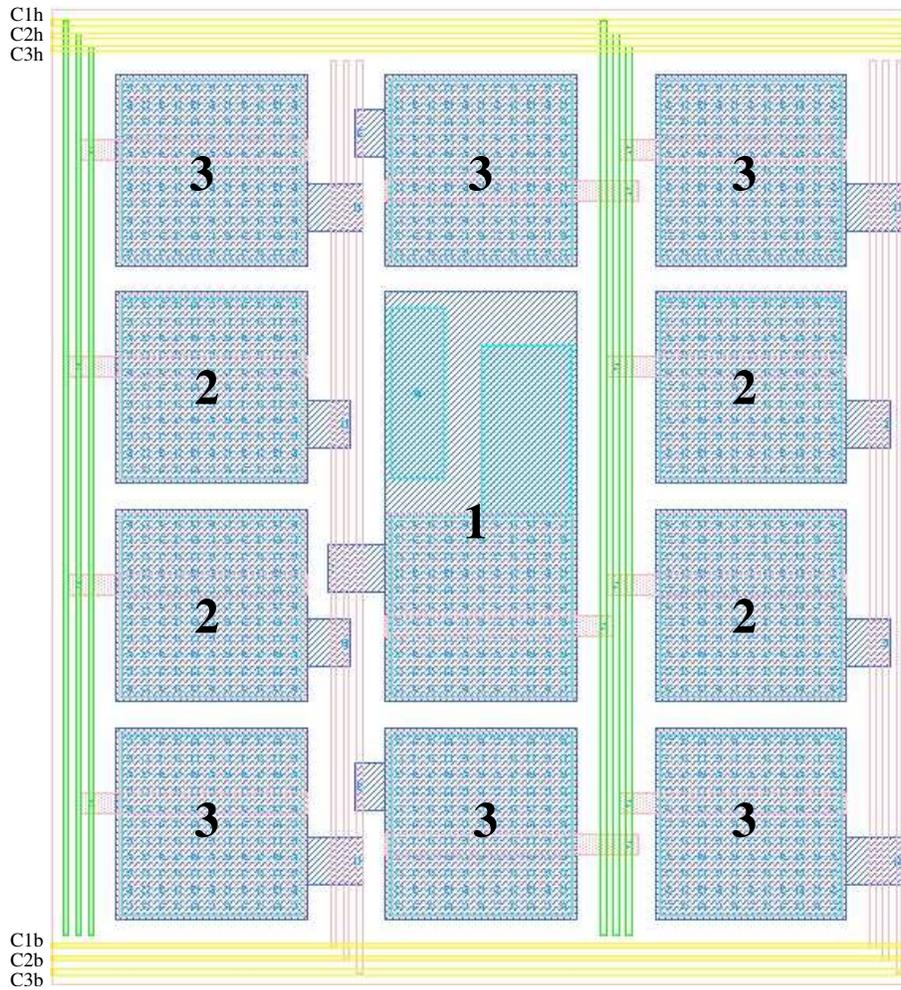


FIG. 7.23 : Dessins des masques du dispositif MIM_MATRIX. $C_{elem} = 0,6pf$, technologie 120nm.

7.4.1 Générateur de motifs pour les dispositifs résistifs

La figure 7.24 montre les motifs utilisés dans la construction des dessins des masques de résistances. Les motifs *a* et *b* sont des segments résistifs dont l'aboutement permet la réalisation de résistances en serpentins. Le motif *c* est un segment résistif fictif. Ces motifs sont issus d'un même générateur `res_segment` auquel on a adressé des paramètres contextuels différents.

7.4.1.1 Paramètres standard

Les deux paramètres standard du générateur `res_segment` sont le nom du modèle `modelName` et la valeur du segment résistif `Rseg`.

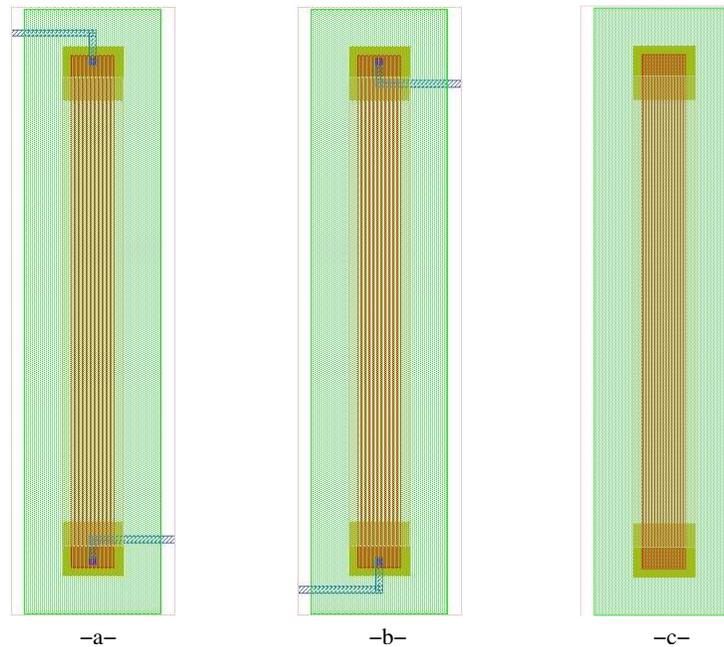


FIG. 7.24 : Motifs utilisés par les dispositifs résistifs.

7.4.1.2 Paramètres contextuels

Les différents motifs générés par `res_segment` se distinguent par la position des fils métalliques internes assurant la connexion aux bornes du segment. On peut spécifier cinq positions pour chacun de ces fils :

- Face est et interne (EI).
- Face ouest et interne (OI).
- Face est et externe (EE).
- Face ouest et externe (OE).
- Pas de fil interne (XX - Segment fictif).

Un fil interne passe au dessus du segment, tandis qu'un fil externe le contourne. Par exemple, sur la figure 7.24-a, le fil desservant le connecteur du haut est externe. Ainsi, le générateur `res_segment` propose deux paramètres contextuels `posCon1` et `posCon2` permettant de spécifier la position des fils desservant les deux connecteurs du segment. Ces paramètres peuvent prendre les valeurs EI, OI, EE ou OE selon la position désirée, ou la valeur XX pour obtenir un segment fictif.

7.4.1.3 Paramètres de sortie

Le générateur `res_segment` remplit les champs suivants de la structure de retour.

- `DXAB` et `DYAB` : Dimensions symboliques de la boîte d'aboutement du motif.
- `DXAB_mu` et `DYAB_mu` : Dimensions réelles de la boîte d'aboutement du motif.
- `R` : Valeur de la résistance effectivement réalisée.
- `C` : Capacité parasite du segment par rapport au substrat.

7.4.1.4 Tableau récapitulatif

Le tableau 7.13 résume finalement les paramètres du générateur `res_segment`.

Nom du générateur	Paramètres standard	Paramètres contextuels	Informations rendues
<code>res_segment</code>	<code>string modelName</code> <code>double Rseg</code>	<code>int posCon1</code> <code>int posCon2</code>	<code>DXAB DYAB DXAB_mu DYAB_mu</code> <code>R C</code>

TAB. 7.13 : *libPOLYRES* : Paramètres des générateurs de motifs.

7.4.2 Générateur de résistances : HIPORES

Le générateur HIPORES réalise des résistances polysilicium uniques.

7.4.2.1 Schéma électrique et paramètres structurels

Le schéma électrique réalisé est donné par la figure 7.25. Le seul paramètre structurel est le nom `Rname` de la résistance dans la vue structurelle.

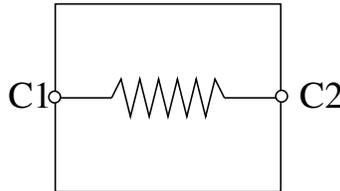


FIG. 7.25 : Schéma électrique du dispositif HIPORES.

7.4.2.2 Procédures de dimensionnement et paramètres électriques

Le tableau 7.14 recense les procédures et les paramètres électriques proposés par ce générateur.

Nom de la procédure	Paramètres d'entrée	Paramètres de sortie	Description
<code>direct_size</code>	<code>R</code>		Valeur de la résistance (Ω)
<code>get_Nseg</code>		<code>Nseg</code>	Nombre de segments
<code>get_Rseg</code>		<code>Rseg</code>	Résistance effective d'un segment (Ω)
<code>get_Cseg</code>		<code>Cseg</code>	Capacité parasite entre un segment et le substrat (F)

TAB. 7.14 : HIPORES : Procédures de dimensionnement et paramètres électriques.

7.4.2.3 Dessin des masques et paramètres de réalisation

Le dispositif HIPORES ne propose pas de paramètre de réalisation. La figure 7.26 montre un exemple de dessin obtenu pour une résistance de $60k\Omega$ en technologie $120nm$.

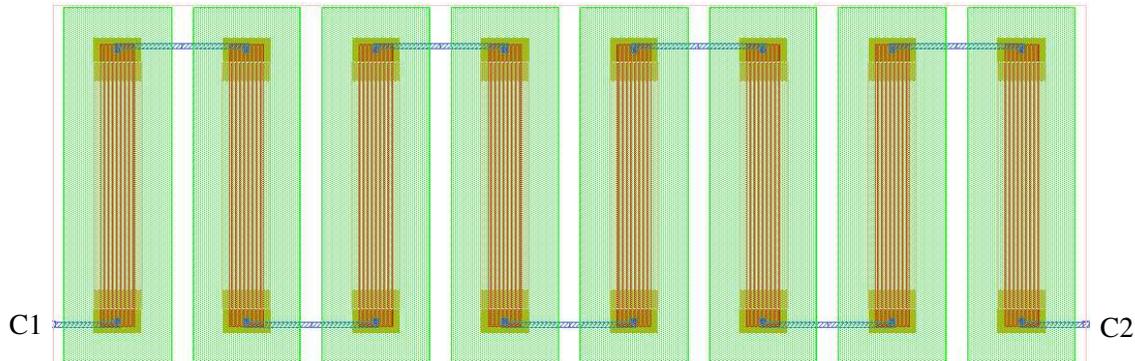


FIG. 7.26 : Dessins des masques du dispositif HIPORES. $R = 60k\Omega$, technologie $120nm$.

7.4.2.4 Fonction de forme

Les différentes formes du dispositif HIPORES sont obtenues en faisant varier le nombre de segments résistifs. Dans les conditions de la figure 7.26, on obtient la fonction de forme illustrée sur la figure 7.27.

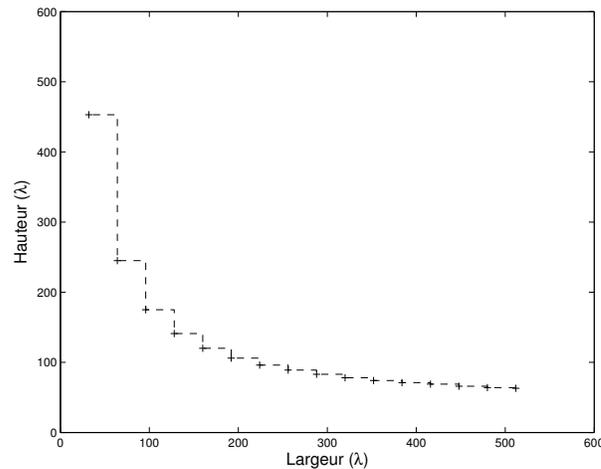


FIG. 7.27 : Fonction de forme du dispositif HIPORES ($\lambda = 120nm$).

7.4.3 Générateur de paires de résistances appariées : BIRES

Le générateur BIRES réalise deux résistances appariées de même valeur.

7.4.3.1 Schéma électrique et paramètres structurels

Les seuls paramètres structurels de ce générateur sont les noms des résistances dans la vue structurelle : R1name et R2name.

Le schéma électrique réalisé est donné par la figure 7.28.

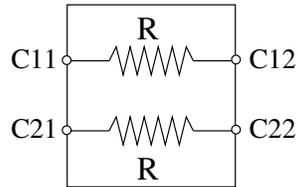


FIG. 7.28 : Schéma électrique du dispositif BIRES.

7.4.3.2 Procédures de dimensionnement et paramètres électriques

Les deux résistances réalisées ici sont identiques. Ainsi, le générateur BIRES propose les mêmes procédures et les mêmes paramètres électriques que le générateur HIPORES.

7.4.3.3 Dessin des masques et paramètres de réalisation

L'unique paramètre de réalisation mis à disposition du concepteur par le générateur BIRES est le paramètre `dummy` qui indique la présence de segments résistifs fictifs aux deux extrémités du dessin.

La figure 7.29 montre un exemple de dessin obtenu pour deux résistances de $60k\Omega$ en technologie $120nm$, sans résistances fictives.

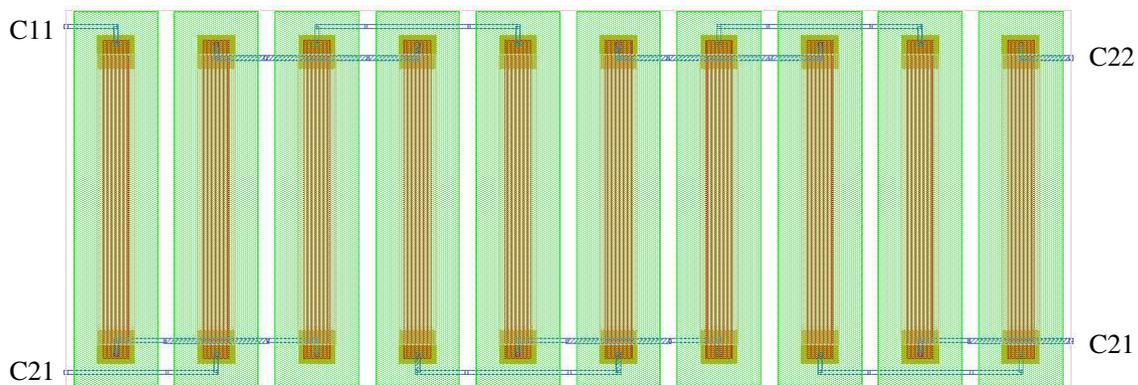


FIG. 7.29 : Dessins des masques du dispositif BIRES. $R = 60k\Omega$, technologie $120nm$.

7.4.3.4 Fonction de forme

Ici encore, on peut obtenir différentes formes en faisant varier le nombre de segments résistifs. Dans les conditions de la figure 7.29, on obtient la fonction de forme illustrée sur la figure 7.30.

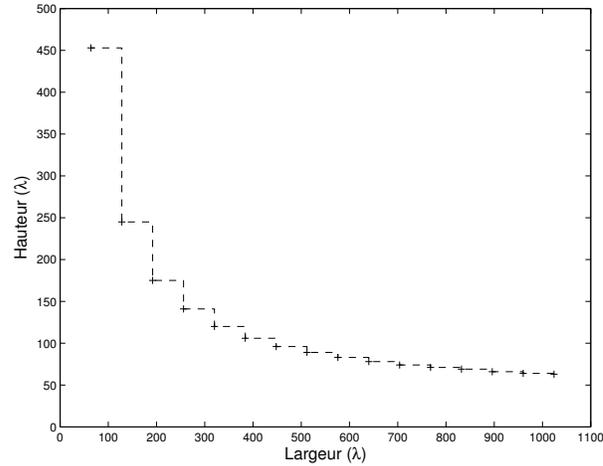


FIG. 7.30 : Fonction de forme du dispositif BIRES ($\lambda = 120\text{nm}$).

7.5 Conclusion

Nous avons présenté dans ce chapitre les générateurs de dispositifs élémentaires qui composent la bibliothèque actuelle. Pour chacun d'entre eux, nous avons énuméré les paramètres, et montré le dessin des masques obtenu. Les générateurs de motifs utilisés ont également été abordés. Dans le chapitre suivant, nous allons voir quelques exemples de circuits qui ont pu être réalisés sous CAIRO+ grâce à cette bibliothèque de dispositifs élémentaires. que nous venons de voir.

Chapitre 8

Applications

8.1 Introduction

Ce chapitre présente quelques circuits analogiques CMOS conçus à l'aide de la bibliothèque que nous avons réalisée. La section 8.2 présente la synthèse d'un intégrateur actif RC sous *CAIRO+*. Dans la section 8.3, nous verrons que *CAIRO+* a pu être interfacé avec un outil de dimensionnement électrique, et ainsi être utilisé comme générateur de dessin des masques. Nous y détaillerons en particulier la synthèse d'un amplificateur de transconductance à deux étages. Enfin, un circuit de mesure de l'appariement de capacités qui a été fabriqué par *STMicroelectronics* est présenté dans la section 8.4.

8.2 Intégrateur actif RC différentiel

A titre de premier exemple d'application de notre travail, nous allons présenter la synthèse sous *CAIRO+* d'un circuit relativement complexe. Nous allons nous intéresser pour cela à la conception d'un intégrateur actif RC différentiel CMOS réutilisable, dont le schéma est donné sur la figure 8.1.

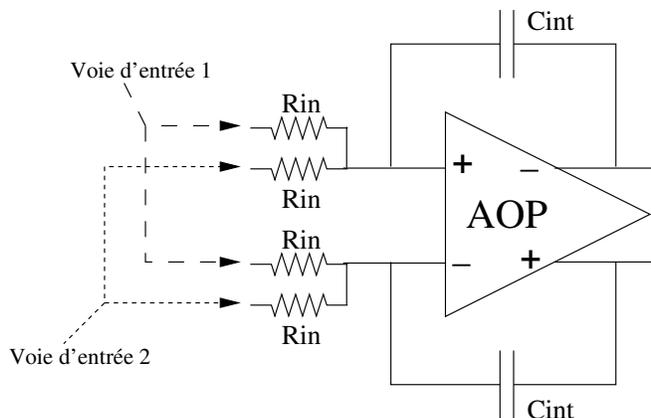


FIG. 8.1 : Intégrateur actif RC différentiel.

Nous ne détaillerons pas ici profondément le circuit puisqu'il s'agit d'un travail qui a été effectué par Laurent de Lamarre durant sa thèse [deLamarre07]. La bonne linéarité de cet intégrateur lui a permis de l'intégrer dans un modulateur $\Sigma\Delta$ du 5^{ème} ordre. Le signal de contre réaction de ce modulateur est réinjecté à la deuxième voie d'entrée. En lui passant différents jeux de spécifications, le générateur *CAIRO+* que nous présentons dans cette section a permis de synthétiser les 5 intégrateurs du modulateur $\Sigma\Delta$.

Dans ce circuit, l'amplificateur opérationnel est basé sur un amplificateur de transconductance (OTA) différentiel à deux étages et à compensation Miller. Nous reviendrons sur ce type d'OTA dans la section suivante. La figure 8.2 détaille plus précisément le schéma électrique de l'AOP. Le premier étage est un OTA différentiel simple. Chacune des deux sorties de ce premier étage est amplifiée par un étage de sortie qui est un amplificateur à source commune chargée par une source de courant. Les capacités de compensation C_c placées entre les sorties du premier étage et celles du second permettent d'assurer la stabilité (cf. 8.3). Finalement, un circuit de contre-réaction de mode commun (CCRMC) permet de stabiliser le système en réduisant le gain de mode commun. Le signal de mode commun est prélevé par un diviseur de tension composé de deux résistances R_{mc} et est stabilisé par les capacités C_{mc} . Notons que tous ces étages sont polarisés par un miroir de courant alimenté par un courant de référence I_{ref} .

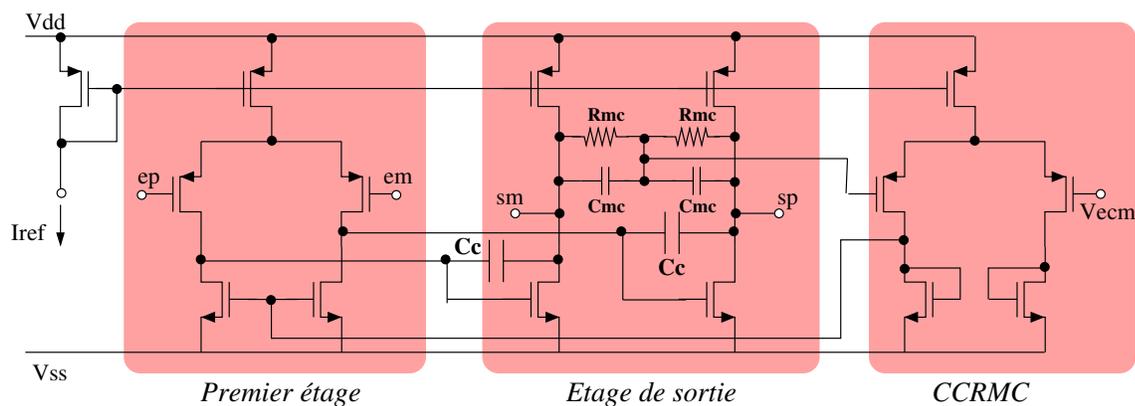


FIG. 8.2 : Schéma électrique interne de l'amplificateur différentiel.

Pour réaliser l'intégrateur étudié ici sous *CAIRO+*, des générateurs ont été développés pour chaque module de l'arbre d'instanciation illustré sur la figure 8.3. Au plus haut niveau hiérarchique (Niveau 3), l'intégrateur s'appuie sur un module AOP et sur une série de dispositifs élémentaires qui réalisent les résistances d'entrée et les capacités d'intégration. Les résistances R_{in} sont générées par paires appariées grâce au générateur *BIREs*. En revanche, le concepteur a choisi de séparer les capacités d'intégration dans deux dispositifs élémentaires différents, C_{int1} et C_{int2} , obtenus par le générateur *CMIM*. Il se laisse ainsi plus de liberté pour le placement physique de ces capacités.

Au niveau hiérarchique 2, le module AOP est basé sur :

- deux modules représentant chacun des deux étages de l'OTA.

- un module représentant le circuit de contre-réaction de mode commun.
- différents dispositifs élémentaires. Il s'agit des capacités de compensation qui sont générées indépendamment par le générateur CMIM, et des composants du pont diviseur permettant de prélever le signal de mode commun. Les résistances R_{mc} sont appariées et générées par le générateur BIRES, tandis que les capacités C_{mc} sont deux instances du générateur CMIM. Dans la hiérarchie choisie ici par le concepteur, les éléments de ce pont n'ont donc pas été intégrés au circuit de contre-réaction lui-même.

Au niveau hiérarchique 1, on trouve les modules qui s'appuient exclusivement sur des dispositifs élémentaires (Niveau 0). Le module représentant le premier étage de l'OTA instancie un miroir de courant à deux transistors dont l'appariement est obtenu grâce au générateur CM_CC. De même, la paire différentielle est issue du générateur DP_CC. Enfin, pour maximiser l'appariement entre les transistors de charge dont la source est partagée, le concepteur a utilisé le générateur MULTI_TR_SHD.

Les deux branches de l'étage de sortie doivent également être appariées. Pour cela, les deux transistors N montés en source commune sont regroupés dans même dispositif élémentaire, ainsi que les deux transistors P qui les chargent. Ces deux dispositifs élémentaires sont obtenus par le générateur MULTI_TR_SHD.

Enfin, la hiérarchie interne du circuit de contre-réaction de mode commun ressemble à celle du premier étage. Cependant, la polarisation est assurée par un transistor unique généré par TR_MOS. Ce transistor sera électriquement connecté au miroir de courant global.

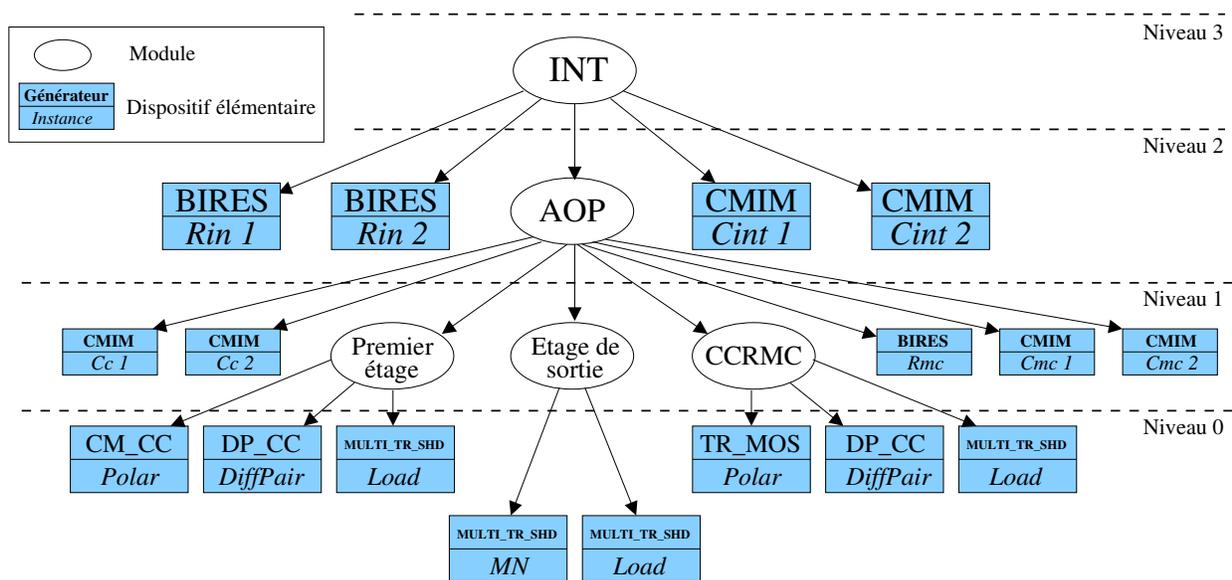


FIG. 8.3 : Arbre d'instanciation du générateur correspondant au circuit de la figure 8.2.

Dans un modulateur $\Sigma\Delta$, le premier étage de filtrage est soumis aux spécifications les plus contraignantes. Par exemple, l'étude au niveau système a donné les spécifications du tableau 8.1

pour l'intégrateur au premier étage.

Spécification	Description	Valeur
f_T	Fréquence d'intégration	2 MHz
SNR	Rapport Signal/Bruit	12 bits (72 dB)

TAB. 8.1 : Jeu de spécifications pour la synthèse de l'intégrateur actif RC différentiel.

Lors de cette étude système, les imperfections de l'AOP sont également prises en compte. Ceci implique que les spécifications de l'AOP sont déjà déterminées au niveau système. Le concepteur a ainsi pu écrire en langage *CAIRO+* une procédure de dimensionnement pour l'intégrateur qui propage directement les spécifications de l'AOP au générateur d'AOP d'une part, et qui calcule d'autre part la résistance R_{in} et la capacité C_{int} en fonction des spécifications f_T et SNR [deLamarre06]. Le tableau 8.2 résume les spécifications qui sont appliquées aux modules du niveau hiérarchique 2. La résistance de charge de l'AOP est la résistance d'entrée de l'intégrateur suivant dans le filtre de boucle du modulateur $\Sigma\Delta$.

Spécification	Description	Valeur
Ad_0	Gain statique minimal de l'AOP	40 dB
f_T	Fréquence de transition de l'AOP	115.4 MHz
R_L	Résistance de charge de l'AOP	5.3 k Ω
C_L	Capacité de charge de l'AOP	C_{int}
R_{in}	Résistance d'entrée	2.95 k Ω
C_{int}	Capacité d'intégration	0.27 pF

TAB. 8.2 : Spécifications appliquées aux modules de niveau 2.

La procédure de dimensionnement de l'AOP transforme les spécifications du niveau 2 en dimensions électriques des composants de niveaux 1 et 0, et en tensions et courants de polarisation [deLamarre07].

A l'issue d'une génération sur une technologie 120nm, on obtient le dessin des masques de la figure 8.4. On remarque que le concepteur a cherché à obtenir une symétrie maximale dans le dessin des masques afin d'assurer l'appariement entre les deux branches différentielles du circuit. De plus, bien que le circuit ne soit pas routé à l'heure actuelle, ce dessin des masques ne génère pas d'erreur de DRC.

En termes de performances, la figure 8.5 compare la réponse en fréquence idéale de l'intégrateur avec celle obtenue en simulant la *netlist* générée par *CAIRO+*. On y voit que la fréquence de transition désirée est atteinte.

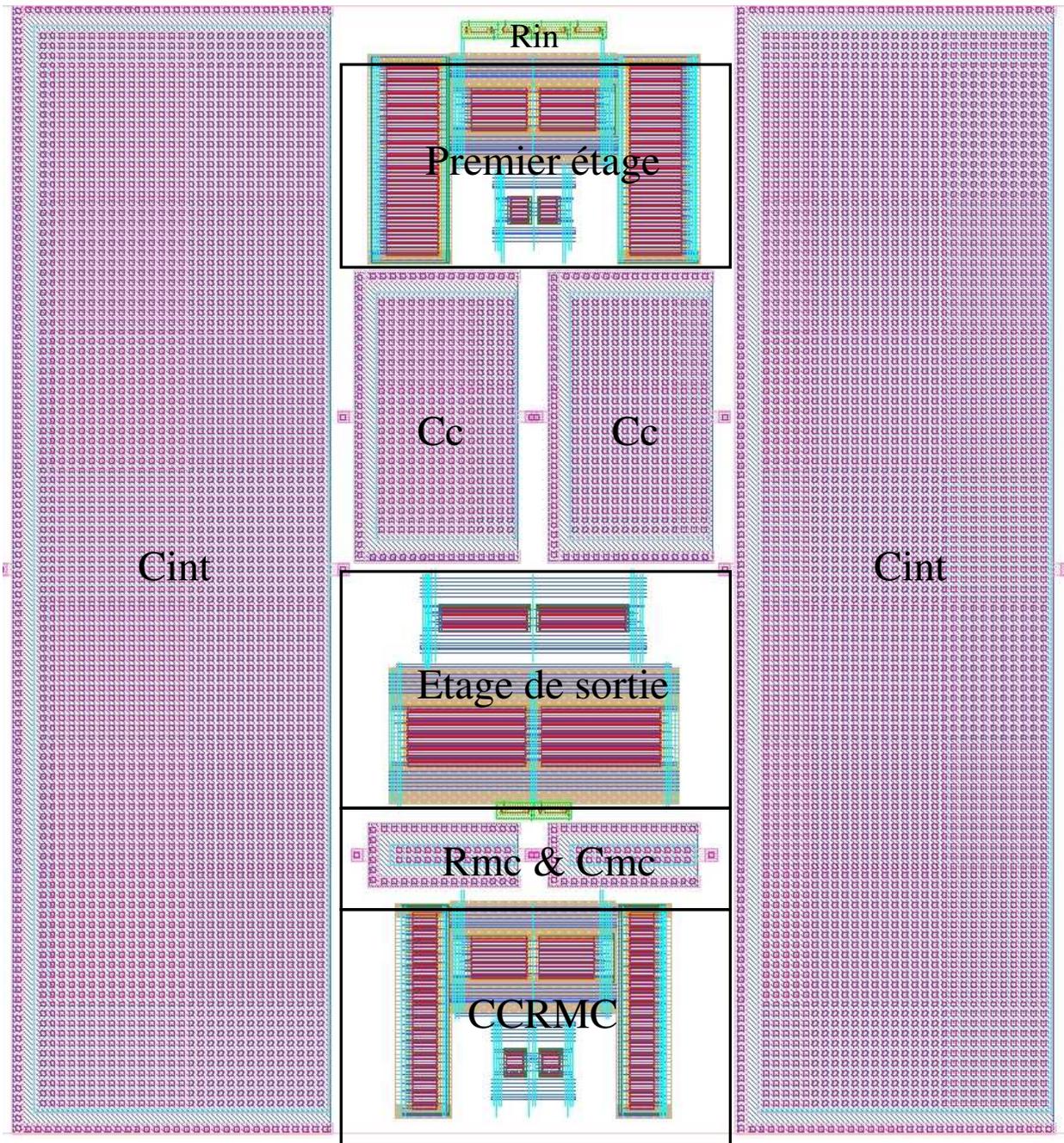


FIG. 8.4 : Dessin des masques de l'intégrateur actif RC différentiel.

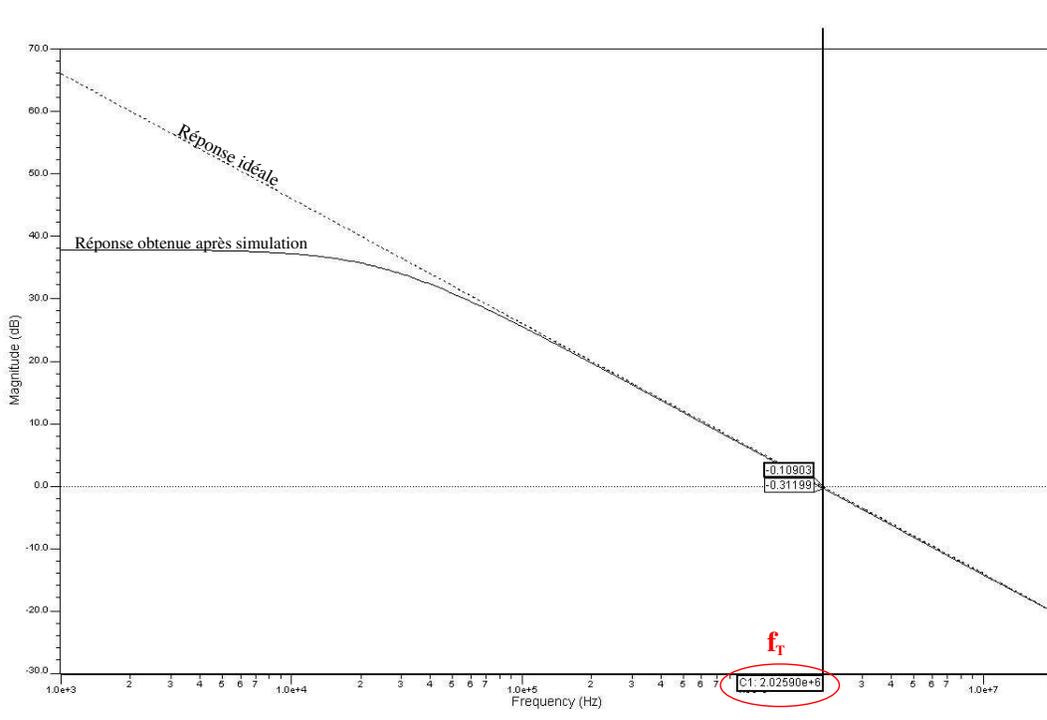


FIG. 8.5 : Réponse en fréquence de l'intégrateur.

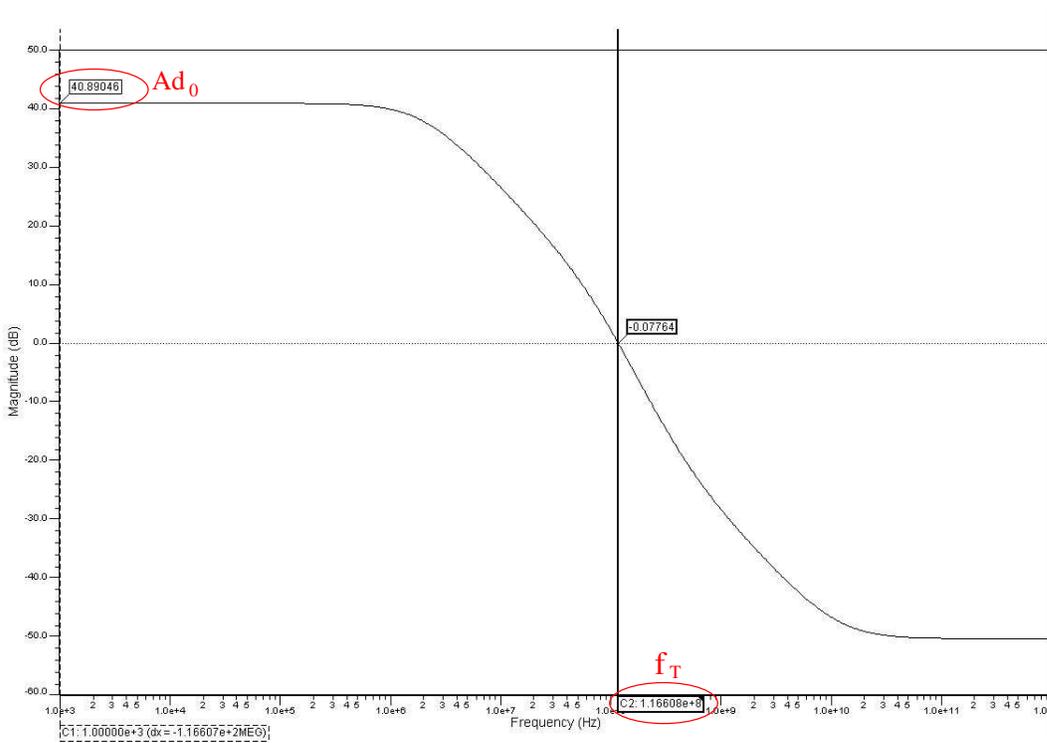


FIG. 8.6 : Réponse en fréquence de l'AOP.

La figure 8.6 montre quant à elle la réponse en fréquence de l'AOP. Le gain statique et la fréquence de transition qui lui avaient été imposés sont respectés.

8.3 Amplificateur de transconductance

Dans ce deuxième exemple, nous allons voir que *CAIRO+* a pu être interfacé avec l'outil de dimensionnement *OCEANE* [Porte07] qui a été développé par Jacky Porte. Dans cette configuration, *CAIRO+* est utilisé principalement comme générateur de dessin des masques et on illustre particulièrement le travail réalisé dans cette thèse.

Ici, le générateur *CAIRO+* utilise donc les dimensions issues d'une exécution d'*OCEANE*. La communication entre *CAIRO+* et *OCEANE* se fait par fichier : à l'issue de son calcul, *OCEANE* documente un fichier contenant les dimensions électriques de tous les composants du circuit. La procédure de dimensionnement du générateur *CAIRO+* transmet ces valeurs aux procédures "direct_size" de tous les dispositifs figurant dans le circuit. Le générateur *CAIRO+* produit alors le dessin des masques et documente à son tour un fichier contenant toutes les informations électriques et géométriques relatives aux composants effectivement réalisés, ainsi que les parasites de routage introduits. Sur la base de ces nouvelles informations, *OCEANE* peut alors affiner son dimensionnement et, au bout de quelques itérations, on obtient le dessin final du circuit respectant les performances désirées.

OCEANE permet le dimensionnement d'une très grande gamme de circuits analogiques concernant aussi bien la polarisation, l'amplification ou le filtrage. Nous allons nous intéresser ici la synthèse d'un amplificateur opérationnel de transconductance (OTA) à deux étages et à compensation de type Miller. La figure 8.7 montre le schéma électrique d'un tel OTA de type N.

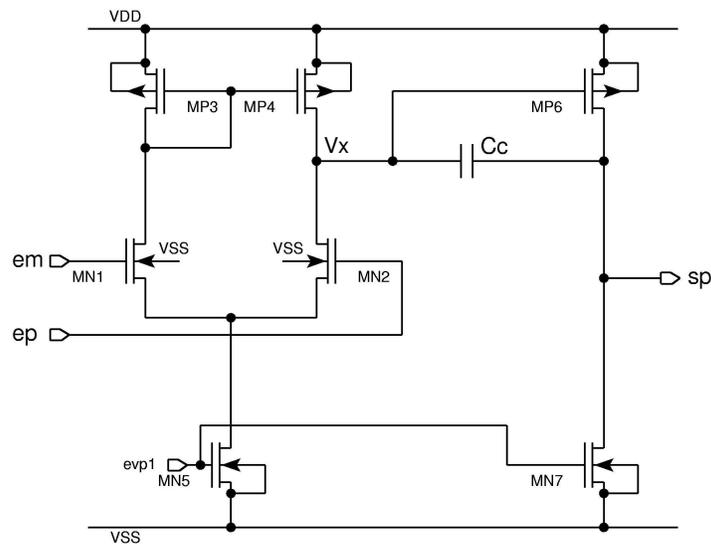


FIG. 8.7 : OTA type N à 2 étages et à compensation Miller.

Ce circuit est constitué d'un premier étage qui est un OTA simple, auquel on a ajouté un second étage qui est un amplificateur à source commune. Ce deuxième étage permet principalement d'améliorer l'excursion en sortie tout en conservant une résistance de sortie raisonnable. En revanche, il introduit un pôle dans la fonction de transfert qui nuit à la stabilité. Pour assurer la stabilité, on effectue une compensation de type Miller en plaçant la capacité de compensation C_c entre la sortie du premier étage et celle du second.

Le dimensionnement du premier étage est basé sur l'équilibre des branches dans l'OTA simple. Ainsi, il se fait en considérant une tension de sortie V_X du premier étage telle que $V_{ds_3} = V_{ds_4} (= V_{gs_3})$.

D'un autre côté, le rapport des courants de polarisation des deux étages, qui influe notamment sur le *slewrate* et sur les caractéristiques fréquentielles du circuit, est imposé par les spécifications au niveau système. La tension de sortie V_{sp} étant également fixée afin d'assurer la plus grande excursion en sortie (en général $V_{sp} = V_{dd}/2$), les dimensions du transistor MP6 fixent la tension V_{gs_6} . Ainsi, selon le rapport W_6/L_6 , il est possible d'observer $V_{gs_6} \neq V_{gs_3}$, c'est-à-dire $V_{ds_3} \neq V_{ds_4}$. Si le gain du premier étage est Ad_1 , cette différence se traduit par une tension de décalage systématique qui, ramenée à l'entrée, vaut :

$$Ed = \frac{V_{gs_3} - V_{gs_6}}{Ad_1} \quad (8.1)$$

Pour s'affranchir de cette tension parasite, il faut dimensionner le transistor MP6 de façon à assurer l'égalité $V_{gs_6} = V_{gs_3}$. Pour cela, on considère les courants de polarisation qui traversent les différents transistors du montage. Tout d'abord, on remarque que les transistors MN5 et MN7 servent de sources de courant et seront donc intégrés à un miroir de courant. Dès lors, si l'on appelle I_0 le courant de drain du transistor MN5 et I_7 le courant traversant les transistors MP6 et MN7, on a :

$$\frac{I_0}{I_7} = \frac{(W/L)_5}{(W/L)_7} \quad (8.2)$$

Par ailleurs, puisque MN1 et MN2 forment une paire différentielle, on a :

$$I_3 = \frac{I_0}{2} \quad (8.3)$$

où I_3 est le courant traversant le transistor MP3. On en déduit :

$$\frac{I_3}{I_7} = \frac{(W/L)_3}{(W/L)_6} = 0.5 \times \frac{(W/L)_5}{(W/L)_7} \quad (8.4)$$

On voit ainsi que pour minimiser la tension de décalage systématique, il faut apparier les transistors MP3, MP4 et MP6 (MP3 et MP4 sont identiques). Ceci introduit une contrainte supplémentaire de conception qui limite le comportement fréquentiel de l'amplificateur. C'est la raison pour laquelle nous avons envisagé deux directives de synthèse pour l'OTA : avec ou sans minimisation de la tension de décalage systématique.

Spécification	Description	Valeur
Ad_0	Gain statique	65 dB
I_0	Courant de polarisation du 1 ^{er} étage	16 μ A
MP	Marge de phase	76°
I_7/I_5	Rapport des courants de polarisation	5
C_L	Capacité de charge	3 pF
Vdd	Tension d'alimentation positive	3.3 V
Vss	Tension d'alimentation négative	0 V
Vicm	Tension de mode commun d'entrée	1.65 V
Vocm	Tension de mode commun de sortie	1.65 V
$V_{eg3}/V_{eg4}/V_{eg6}$	Tension effective de grille	0.2V
V_{eg1}/V_{eg2}	Tension effective de grille	0.2V
V_{eg5}/V_{eg7}	Tension effective de grille	0.2V
H	Hauteur maximale du dessin des masques	Libre
DP_STYLE	Méthode d'appariement de la paire différentielle	Géométrie centrée
CAPA_STYLE	Style de la capacité	Capacité unique

TAB. 8.3 : Jeu de spécification pour l'OTA deux étages avec minimisation de la tension de décalage systématique.

Après quatre itérations entre CAIRO+ et OCEANE, on obtient les dimensions et la tension de polarisation V_{p1} données dans le tableau 8.4. Le dessin des masques obtenu est visible sur la figure 8.9.

Composant	Dimensions	
MN1 & MN2	W = 1.40 μ m	L = 0.55 μ m
MP3 & MP4	W = 5.1 μ m	L = 0.55 μ m
MN5	W = 3.2 μ m	L = 0.55 μ m
MP6	W = 51.0 μ m	L = 0.55 μ m
MN7	W = 16.0 μ m	L = 0.55 μ m
Cc	2.29 pF	
V_{p1}	0.741 V	

TAB. 8.4 : Dimensions obtenues pour l'OTA deux étages avec minimisation de la tension de décalage systématique.

Les performances obtenues après simulation sont résumées dans le tableau 8.5. On constate que la fréquence de transition obtenue, liée principalement au courant de polarisation I_0 , se situe autour de 5 Mhz. Par ailleurs, le circuit affiche une consommation de 0.32 mW. Enfin, précisons

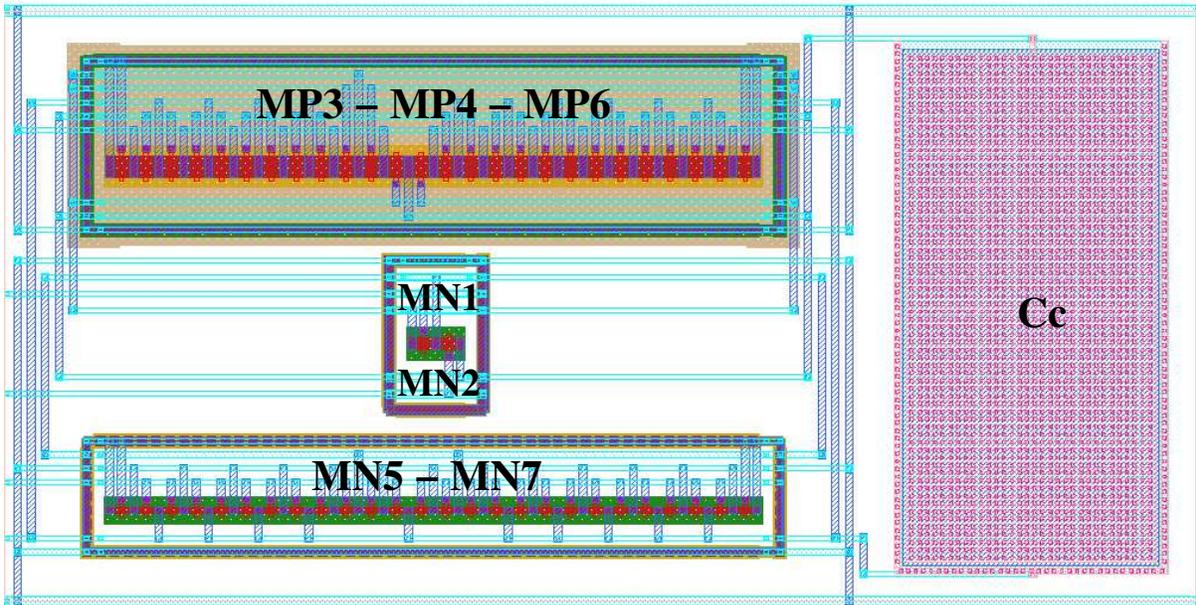


FIG. 8.9 : Dessin des masques d'un OTA deux étages pour les spécifications du tableau 8.3.

Performance	Description	Valeur
f_T	Fréquence de transition	4.98 MHz
Ad_0	Gain statique	66.64 dB
MP	Marge de phase	75.96°
	Consommation	0.32 mW
	Surface totale	$9.67 \times 10^{-3} \text{mm}^2$

TAB. 8.5 : Performances obtenues après simulation.

que la synthèse avec dessins des masques a été effectuée en 27 secondes.

8.3.2 Synthèse sans minimisation de la tension de décalage systématique

Si l'on tolère une tension de décalage systématique en entrée, la tension effective de grille du transistor MP6 peut être ajustée. On peut ainsi atteindre des performances fréquentielles plus ambitieuses. Dans ce cas, les dimensions du transistor MP6 sont indépendantes de celles des transistors MP3 et MP4 et celui-ci sera réalisé par un dispositif élémentaire dédié. Le schéma d'instanciation du générateur d'OTA devient celui de la figure 8.10. Désormais, les transistors MP3 et MP4 sont réalisés par un générateur de miroirs de courant CM_ID ou CM_CC. Là encore, c'est l'utilisateur qui spécifie la méthode d'appariement à utiliser. Le transistor MP6, quant à lui, est pris en charge par le générateur de transistors TR.MOS. Pour ces trois transistors, on indique que la source et le substrat sont connectés.

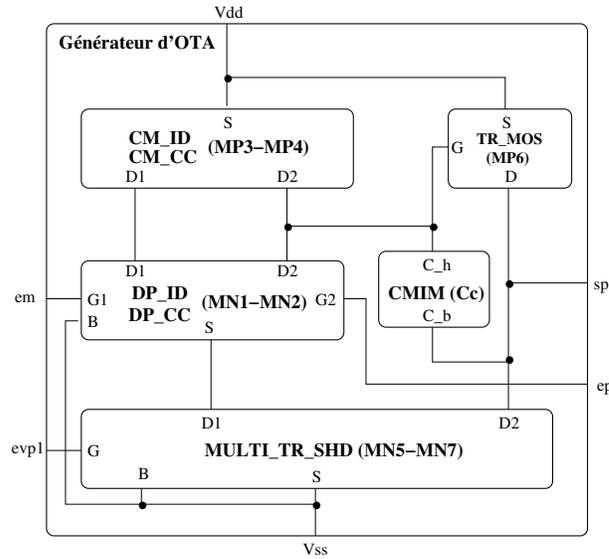


FIG. 8.10 : Schéma d'instanciation du générateur d'OTA deux étages sans minimisation de la tension de décalage systématique.

On peut alors imposer le même jeu de spécification que celui présenté dans le tableau 8.3. Cependant, la tension effective de grille du transistor MP6 peut maintenant être fixée à une valeur inférieure : $0.1V$ par exemple. Dans ce cas, et après, là encore, quatre itérations, on obtient les dimensions présentées dans le tableau 8.6. et le dessin des masques de la figure 8.11.

Composant	Dimensions	
MN1 & MN2	$W = 1.25\mu\text{m}$	$L = 0.5\mu\text{m}$
MP3 & MP4	$W = 4.45\mu\text{m}$	$L = 0.5\mu\text{m}$
MN5	$W = 2.9\mu\text{m}$	$L = 0.5\mu\text{m}$
MP6	$W = 134.4\mu\text{m}$	$L = 0.5\mu\text{m}$
MN7	$W = 14.5\mu\text{m}$	$L = 0.5\mu\text{m}$
C_c	1.28 pF	
V_{ep1}	0.752 V	

TAB. 8.6 : Dimensions obtenues pour l'OTA deux étages sans minimisation de la tension de décalage systématique.

Le tableau 8.7 montre les résultats obtenus par simulation. A consommation identique, on obtient une fréquence de transition environ deux fois et demi plus grande. On remarque également que la surface du circuit a été réduite par rapport au cas précédent. Enfin, le temps d'exécution de la synthèse est comparable à celui de l'OTA avec minimisation de la tension de décalage systématique.

En conclusion, nous voyons les générateurs CAIRO+ peuvent être utilisés pour fournir au

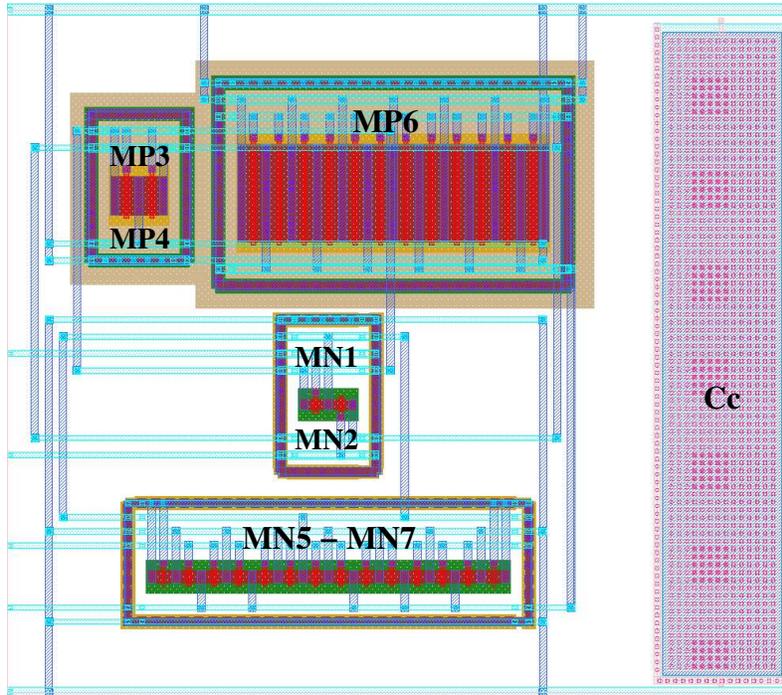


FIG. 8.11 : Dessin des masques d'un OTA deux étages sans minimisation de la tension de décalage systématique.

Performance	Description	Valeur
f_T	Fréquence de transition	8.58 MHz
Ad_0	Gain statique	65.17 dB
MP	Marge de phase	75.96°
	Consommation	0.33 mW
	Surface totale	$7.1 \times 10^{-3} mm^2$

TAB. 8.7 : Performances obtenues après simulation.

concepteur un dessin des masques précis et documenté. Les informations relatives à ce dessin peuvent être intégrées à des procédures de dimensionnement extérieures à l'environnement CAIRO+.

8.4 Circuit de mesure d'appariement de capacités

La dernière application de notre travail que nous allons présenter est un circuit permettant la mesure de l'appariement entre différentes capacités MIM. Ce circuit a permis de quantifier l'importance du placement relatif des capacités élémentaires dans une matrice vis-à-vis de l'appariement. Il s'agissait de comparer l'appariement obtenu par un placement naïf, et celui obtenu par

un placement issu de l'algorithme présenté dans [Sayed02]. Cet algorithme permet d'optimiser l'appariement entre différentes capacités liées par des rapports arbitraires qui peuvent ne pas être entiers. Pour cela, il détermine un placement relatif des capacités élémentaires qui se rapproche le plus possible d'une géométrie centrée.

Le circuit intègre deux blocs analogiques distincts permettant les mesures :

- Un filtre passe-bas à capacités commutées dont les performances dépendent fortement des rapports capacitifs.
- Un circuit à grille pseudo-flottante qui facilite la mesure de l'appariement entre deux capacités [Buisson98].

Les mesures présentées dans cette section ont été effectuées sur un échantillon de 31 circuits fabriqués par *STMicroelectronics* en technologie $0.18\mu\text{m}$ et ont donné lieu à deux publications [Khalil04, Khalil05].

8.4.1 Filtre passe-bas à capacités commutées

Le filtre passe-bas réalisé est un biquad à capacités commutées dont le schéma est donné sur la figure 8.12. Les performances de ce circuit dépendent fortement des rapports entre les différentes

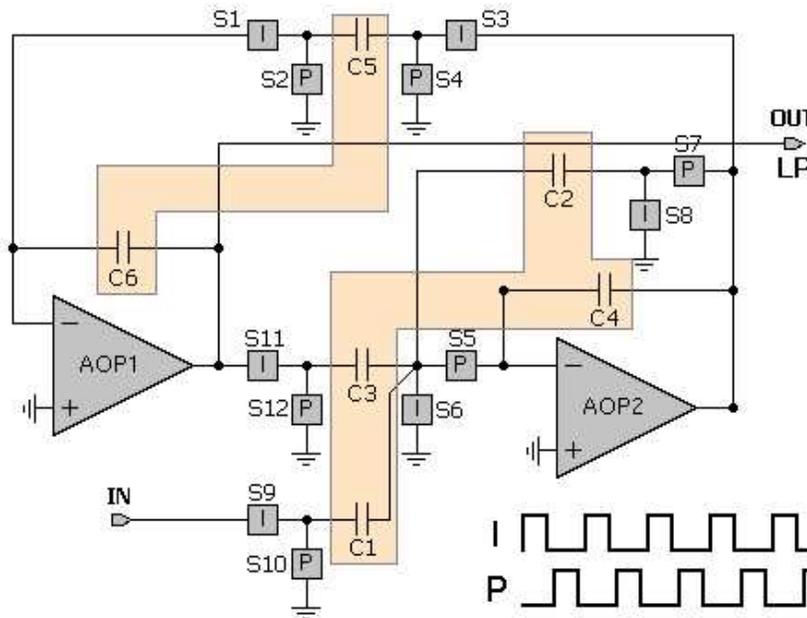


FIG. 8.12 : Filtre passe-bas : Biquad à capacités commutées.

capacités. En effet, la fréquence de coupure f_c , le gain statique A_{DC} et le facteur de qualité Q sont donnés par :

$$f_c = \frac{f_s}{2\pi} \cdot \cos^{-1} \left(1 - \frac{C5 \cdot C3}{C6 \cdot (C2 + 2 \cdot C4)} \right) \quad (8.5)$$

où f_s est la fréquence d'échantillonnage.

$$A_{DC} = 20. \log \left(\frac{C1}{C3} \right) \quad (8.6)$$

$$Q = \sqrt{\frac{C3.C5}{2.C2.C6} \left(1 + \frac{2.C4}{C2} - \frac{C3.C5}{2.C2.C6} \right)} \quad (8.7)$$

Selon la qualité de l'appariement réalisé entre les différentes capacités, on va ainsi observer des performances plus ou moins proches des spécifications.

Les spécifications qui ont été choisies sont les suivantes : $f_c = 5.5kHz$, $f_s = 100kHz$, $A_{DC} = -23.86dB$, et $Q = 13.96$. Le choix d'un facteur de qualité relativement grand permet de mieux observer les résultats dans la zone de résonance. Ces spécifications ont conduit aux valeurs de capacités données dans le tableau 8.8. La valeur de la capacité élémentaire a été fixée à 0.2 pF.

Capacité	C1	C2	C3	C4	C5	C6	Celem
Valeur (pF)	0.2	0.22	3.12	8.96	0.2	0.582	0.2

Tab. 8.8 : Valeurs des capacités du biquad.

Dès lors, afin de mettre en évidence l'influence du placement relatif des capacités élémentaires sur l'appariement, les capacités C1, C2, C3 et C4 ont été réalisées par deux matrices différentes.

- Une matrice dessinée "à la main" dans laquelle le placement des capacités élémentaires est naïf (fig. 8.13-a).
- Une matrice réalisée par le générateur CAIRO+ MIM_MATRIX dans laquelle le placement des capacités élémentaires est donné par l'algorithme étudié (fig. 8.13-b).

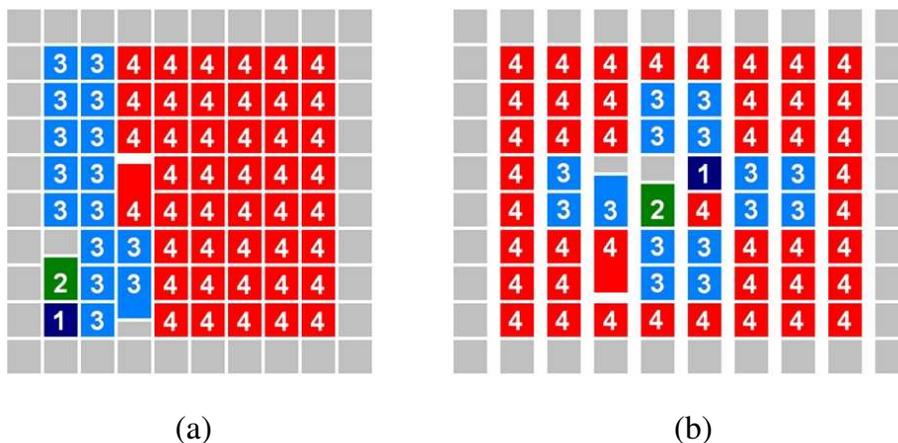


FIG. 8.13 : Les deux matrices de capacités pour le circuit de mesure d'appariement.

Des commutateurs permettent d'utiliser au choix la matrice naïve, ou la matrice optimisée. Le reste du circuit étant commun, on peut ainsi conclure que les différences observées sont dues au placement relatif des capacités élémentaires. La figure 8.14 montre les résultats obtenus. Les

réponses fréquentielles des 31 circuits fabriqués utilisant chacune des deux matrices sont tracées sur la figure 8.14-a. Les variations observées sur les performances sont minimisées lorsque l'on utilise la matrice optimisée. Ceci se confirme sur la figure 8.14-b qui montre l'écart type observés pour ces réponses fréquentielles selon la matrice utilisée.

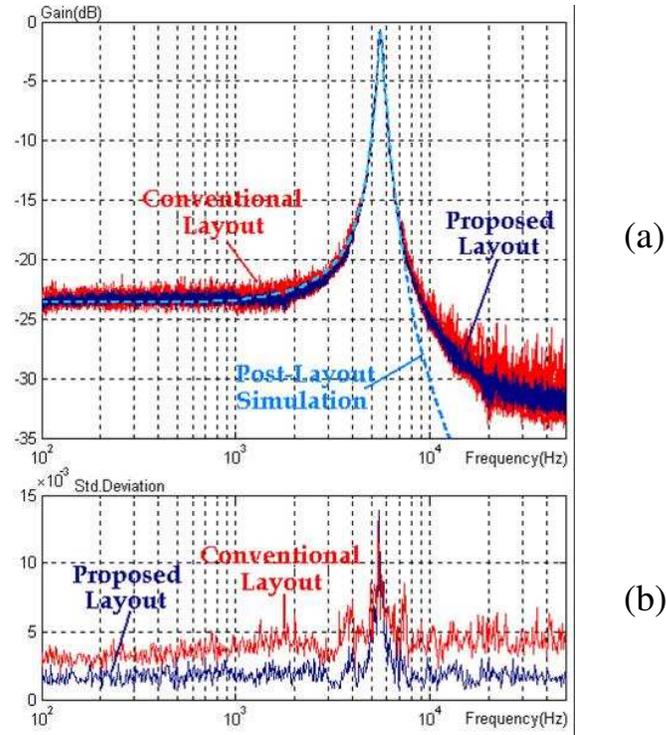


FIG. 8.14 : (a) Réponses en fréquence obtenues pour les 31 circuits fabriqués - (b) Ecart type.

Le tableau 8.9 montre, en fonction de la matrice utilisée, les valeurs moyennes et les écarts types obtenus pour les performances auxquelles nous nous sommes intéressés. Les variations observées

		Matrice naïve	Matrice optimisée
A_{DC} (dB)	Moyenne	-23.675	-23.688
	Ecart type	0.4543	0.1898
f_c (Hz)	Moyenne	5562.7	5559.7
	Ecart type	14.947	8.233
Q	Moyenne	13.409	13.512
	Ecart type	0.2474	0.0389

TAB. 8.9 : Valeur moyenne et écart type des paramètres fréquentiels.

sur la fréquence de coupure f_c et le gain statique A_{DC} sont réduites d'un facteur avoisinant 2 avec l'utilisation de la matrice optimisée. Les variations sur le facteur de qualité Q sont réduites quant

à elles d'un facteur supérieur à 6.

Ce premier circuit tend à confirmer l'importance du placement des capacités élémentaires vis-à-vis de l'appariement. Voyons maintenant les résultats obtenus avec l'autre technique de mesure envisagée.

8.4.2 Circuit à grille pseudo-flottante

Le deuxième bloc présent sur notre circuit est un montage à grille pseudo-flottante dont le schéma est donné sur la figure 8.15. Sur ce schéma, la diode permet de protéger le circuit contre l'effet d'antenne, et C_{par} représente les capacités parasites intervenant au niveau de la grille.

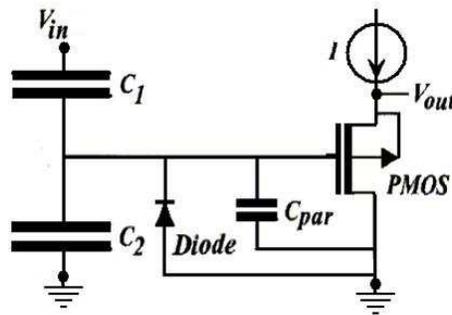


FIG. 8.15 : Circuit à grille pseudo-flottante.

Il s'agit d'un montage drain commun qui, à courant de polarisation I constant, est caractérisé par la relation suivante :

$$V_{out} = \left(\frac{C_1}{C_1 + C_2 + C_{par}} \right) \times V_{in} + V_{TSAT} \quad (8.8)$$

où V_{TSAT} est une constante qui dépend notamment du courant de polarisation, et de la tension de seuil du transistor.

La méthode de mesure consiste alors à observer la pente de la sortie V_{out} lorsque :

- Plusieurs niveaux de tension d'entrée V_{in} sont appliqués sur C_1 , C_2 étant connectée à la masse. On observe dans ce cas la pente P_1 .
- Plusieurs niveaux de tension d'entrée V_{in} sont appliqués sur C_2 , C_1 étant connectée à la masse. On observe dans ce cas la pente P_2 .

On a alors :

$$\frac{P_1}{P_2} = \frac{C_1}{C_1 + C_2 + C_{par}} \div \frac{C_2}{C_1 + C_2 + C_{par}} = \frac{C_1}{C_2} \quad (8.9)$$

En ce qui nous concerne, en plus des commutateurs déjà utilisés pour le biquad et qui permettent de choisir la matrice à mesurer, on dispose de commutateurs permettant de choisir différents couples de capacités dans chacune des matrices. On peut alors mesurer l'appariement de chaque couple. La figure 8.16 illustre les résultats obtenus en indiquant l'erreur mesurée par rapport à la valeur théorique de chaque rapport capacitif. On y constate très clairement que la matrice optimisée permet de réduire globalement les erreurs d'appariement de moitié.

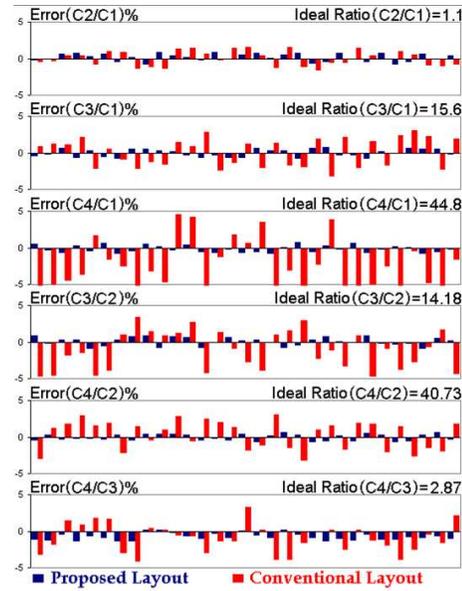


FIG. 8.16 : Erreurs d'appariement obtenues pour chaque matrice.

Au delà des résultats très intéressants obtenus concernant l'appariement des capacités, ce circuit aura permis de valider les dessins produits par CAIRO+. En effet, la matrice présentant les meilleurs résultats a été conçue sous CAIRO+ et on peut donc supposer que l'automatisation ne nuit pas ici à la qualité du dessin. De plus, en ayant été soumis aux contraintes de la fabrication, les dessins générés ont prouvé leur validité. La figure 8.17 montre finalement le dessin des masques du circuit complet qui a été envoyé en fabrication.

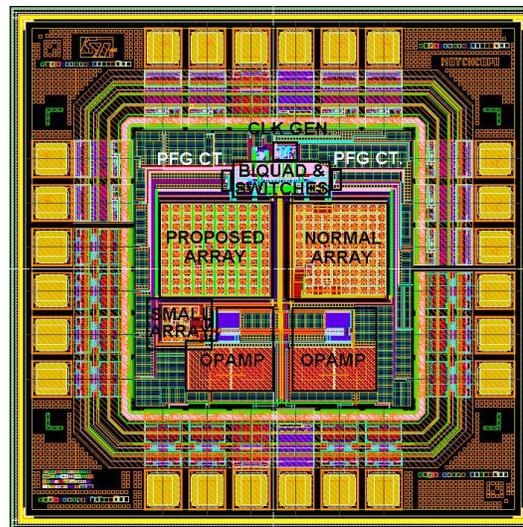


FIG. 8.17 : Dessin des masques complet.

8.5 Conclusion

Les différentes applications présentées dans ce chapitre ont permis de valider le travail effectué durant cette thèse. La synthèse classique d'un circuit sous *CAIRO+* a été présentée. Nous avons également montré que nos générateurs pouvaient être utilisés avec un outil extérieur de dimensionnement. Enfin, en confrontant les dessins produits par *CAIRO+* au processus de fabrication, leur validité a été prouvée.

Chapitre 9

Conclusion et Perspectives

La complexité croissante des systèmes intégrés sur puce fait émerger le besoin de blocs fonctionnels réutilisables, appelés IP. Bien que l'on maîtrise relativement bien la définition d'IP numériques, celle d'IP analogiques est particulièrement difficile en raison de l'expertise sollicitée lors de la conception des blocs analogiques. Dans ce domaine en effet, le fonctionnement d'un circuit est caractérisé par un ensemble de performances interdépendantes et de perpétuels compromis sont ainsi nécessaires durant la conception. Par ailleurs, les dimensions des composants de base et la qualité de leur réalisation physique influent largement sur le comportement final du circuit et soumettent le concepteur à des contraintes supplémentaires.

Le projet *CAIRO+* propose un environnement de conception et de synthèse d'IP analogiques. Il permet la réutilisation de ces IP sur différents procédés de fabrication, et pour différents jeux de spécifications. Pour cela, *CAIRO+* s'inspire du flot de synthèse utilisé en numérique qui s'appuie sur :

- Un langage de description d'IP.
- Une bibliothèque de cellules de base permettant la synthèse de ces IP.

Afin de s'adapter aux spécificités de la conception analogique, le langage *CAIRO+* permet d'exprimer les compromis de conception, tandis que les cellules de base proposées s'appuient sur une modélisation précise des composants pour produire et caractériser un dessin des masques optimisé.

Cette thèse a porté sur la conception des cellules de base pour *CAIRO+*, et particulièrement sur les problèmes liés à leur réalisation physique, capitale en analogique.

9.1 Contributions

Notre travail a engendré un résultat sur deux plans :

- La définition d'une méthodologie de conception de générateurs de base pour *CAIRO+*.
- La réalisation d'une bibliothèque de générateurs de base permettant la conception d'une large gamme de circuits analogiques.

La méthodologie que nous avons proposée permet aux générateurs réalisés de s'intégrer dans une démarche de réutilisation. Elle assure en effet le respect des contraintes liées au dessin des masques analogiques, quel que soit le contexte de la synthèse, c'est-à-dire quel que soient le procédé cible et les spécifications imposées. En effet, nos générateurs de base présentent les caractéristiques suivantes :

Précision du dessin des masques

Les dimensions électriques des composants de base sont capitales en analogiques. Le dessin de ces composants doit donc être fidèle aux dimensions visées. Les générateurs que nous avons conçus assurent une précision maximale sur les rectangles significatifs.

Caractérisation du dessin des masques

Puisque les performances sont particulièrement sensibles aux dimensions des composants de base, les imperfections de fabrication ou les parasites influent fortement sur le comportement. Nos générateurs caractérisent précisément le dessin qu'ils produisent. Une boucle de synthèse peut alors être mise en œuvre entre le dimensionnement électrique et la génération des masques afin d'obtenir un dessin final satisfaisant.

Prise en compte des contraintes d'appariement

De nombreux dispositifs analogiques reposent sur la précision des rapports entre les dimensions de différents composants. Pour répondre à ce type de contraintes, nous avons proposé des générateurs dédiés capables de produire un dessin optimisé vis-à-vis de l'appariement. Ces générateurs s'appuient sur les techniques de dessin spécifiques aux composants réalisés, et assurant une précision maximale sur les rapports de dimensions après fabrication.

Migration technologique

Nous avons défini une représentation simple des règles technologiques capable d'exprimer la grande diversité de règles imposées par chaque fondeur. Grâce à cette représentation, et à une architecture logicielle adaptée, la génération d'un circuit sous *CAIRO+* peut se faire sur différents procédés de fabrication. Le résultat de cette génération respecte les règles de dessin spécifiques au procédé cible et, grâce à la boucle de synthèse, peut satisfaire les spécifications.

Gestion des déformations

En analogique, les dimensions électriques des composants peuvent connaître de grandes variations lors du changement des spécifications ou du procédé cible. Nos générateurs de base sont capables de produire plusieurs dessins de facteurs de forme différents pour un même composant. Ces dessins présentent des caractéristiques électriques similaires et permettent d'obtenir finalement un dessin global compact. On réduit alors la surface du circuit tout en assurant des propriétés physiques homogènes à tous les composants.

La méthode que nous avons proposée dans cette thèse a été validée sur plusieurs applications. A ce titre, nous avons présenté la synthèse d'un intégrateur actif RC différentiel utilisé dans un convertisseur $\Sigma\Delta$. Nous avons également vu que le dessin produit par nos générateurs a pu être exploité par un outil extérieur de dimensionnement électrique. Enfin, nous avons confronté avec succès un générateur de capacités aux contraintes de la fabrication, en l'intégrant dans un circuit fondu par *STMicronics*.

9.2 Perspectives

Plusieurs améliorations peuvent être apportées à ce travail. Voici celles vers lesquelles il nous paraît intéressant de se tourner en priorité :

Utilisation exclusive de la grille physique

Dans notre approche, la génération du dessin des masques s'appuie sur deux grilles : la grille physique dont le pas vaut δ et la grille symbolique de pas λ . Comme nous l'avons vu, on a typiquement $\lambda \approx 10.\delta$. L'utilisation de la grille symbolique nuit donc à la précision du dessin et peut introduire des parasites supplémentaires. Nous proposons de conserver l'approche qui consiste à construire le dessin comme un aboutement de motifs, mais de s'affranchir de l'utilisation de la grille symbolique. La gestion des déformations resterait dans ce cas relativement aisée, tandis qu'on assurerait une précision maximale sur le dessin généré. Cela suppose de développer un langage de placement et de routage travaillant sur grille physique.

Prise en compte de l'électro-migration

A l'heure actuelle, la largeur des fils de routage reliant les différents motifs est fixe. Cependant, le courant traversant un transistor dépend de ses dimensions électriques. Afin de prévenir l'électro-migration, il serait intéressant de déterminer la largeur des fils de routage internes à un dispositif élémentaire en fonction du courant qui les traverse.

Algorithme de placement et de routage pour les dispositifs élémentaires complexes

Les trois générateurs matriciels que nous avons conçus pour le moment (*MULTI_TR*, *MULTI_TR_SHD* et *MIM_MATRIX*) se basent sur un placement et un routage définis par l'utilisateur dans le fichier de réalisation. Si cette approche laisse la plus grande liberté possible, elle rend l'utilisation de ces générateurs un peu fastidieuse. Nous aimerions implanter dans ce type de générateurs des algorithmes de placement et de routage qui donneraient une solution favorisant l'appariement.

Caractérisation de l'appariement

Pour des dimensions électriques données, nous sommes capables d'optimiser l'appariement entre composants. Cependant, la qualité l'appariement dépend des dimensions électriques.

Nous suggérons donc de proposer au concepteur un ensemble de procédures qui lui permettraient d'ajuster le dimensionnement en vue d'un appariement optimal. Avec de telles procédures, des paramètres comme le nombre de repliements d'un transistor ou la valeur de la capacité élémentaire pourraient être déterminés afin de réduire plus efficacement les erreurs aléatoires et systématiques d'appariement.

Mise à jour du modèle de transistor

Comme nous l'avons vu, le modèle Bsim3v3 est actuellement intégré à CAIRO+. Celui-ci ne prend cependant pas en compte certains effets qui apparaissent dans les technologies inférieures à 130nm. Par exemple, l'utilisation de la STI (*Shallow Trench Isolation*) engendre des effets de contraintes mécaniques qui impliquent que les différents doigts d'un empilage ont un comportement qui dépend de la surface totale de la zone active et de leur emplacement au sein de cette zone [Dunga07]. Les modèles de nouvelle génération tels que le modèle Bsim4 prennent en compte ce genre de défaut. Bien que ce point sorte du cadre de cette thèse, il faudra mettre à jour le modèle de transistor dans CAIRO+.

Amélioration des modèles de parasites pour les composants passifs

Pour chaque type de composants passifs, nous avons vu qu'un modèle de parasites a été adopté. Cependant, dans certaines applications, des modèles plus précis peuvent s'avérer nécessaires. Nous pensons qu'il serait intéressant de mettre à disposition du concepteur différents modèles, plus ou moins précis, qu'il pourrait choisir en fonction de l'application.

Dispositifs élémentaires inductifs

Les inductances intégrées sont des éléments particulièrement dépendants de leur réalisation physique. Il est ainsi relativement difficile de conserver des spécifications données sur différents procédés. De même, le facteur de qualité d'une inductance dépend fortement de sa géométrie. La déformation, nécessaire dans notre approche, est alors difficile. Enfin, selon l'application, il existe une multitude de modèles représentant les inductances et il s'avère délicat de n'en adopter qu'un. Les inductances sont des éléments primordiaux de la conception hautes fréquences et doivent être prises en charge par CAIRO+

Dispositifs élémentaires pour les micro-systèmes

Actuellement, les technologies CMOS sont exploitées pour concevoir des structures électromécaniques intégrées (MEMS). Dès lors, différents types de capteurs peuvent par exemple être directement intégrés sur la puce [Lira05]. On peut envisager le développement de générateurs de dispositifs élémentaires pour les MEMS.

Certaines de ces perspectives, notamment les trois premières, sont déjà abordées au sein du laboratoire LIP6 par Yifei Wu.

Annexe A

Canevas de conception d'un générateur de dispositifs élémentaires

Cette annexe présente brièvement l'implémentation d'un générateur de dispositifs élémentaires. On y retrouve les quatre fonctions constructrices présentées à la section 5.6 (page 57), ainsi que les principales fonctions *CAIRO+* qu'elles appellent. On ne présente cependant ici qu'un canevas de conception. Pour plus de détails sur le langage *CAIRO+*, le lecteur se réfèrera à [NGuyen06].

```
1
2 /*****/
3 /* Creation */
4 /*****/
5 CAIRO_BEGIN_CREATE (nomGénérateur ,
6                     char *nomInstance
7                     [, liste des parametres structurels]
8                     )
9
10
11
12 // Declaration des variables locales a CREATE
13 long var1=0;
14 double var2=0.0;
15
16 // Appels aux fonctions de la DAPI
17 var1 = nomGénérateur_DAPI_GET_var1 ();
18 var2 = nomGénérateur_DAPI_GET_var2 ();
19
20 // Mise a disposition des parametres structurels
21 // et des informations issues de la DAPI aux
22 // autres fonctions constructrices.
23 CAIRO_SET_LOCAL_VARIABLE ("paramStruct1", valParamStruct1);
24 CAIRO_SET_LOCAL_VARIABLE ("VAR1", var1);
```

```

25 CAIRO_SET_LOCAL_VARIABLE("VAR2", var2);
26
27
28
29 /* ***** */
30 /*      Creation de la vue structurelle      */
31 /* (La declaration des signaux est implicite) */
32 /* ***** */
33
34 //Connecteurs externes
35 CAIRO_IO("connecteur1", CP_WEST, CP_EAST);
36 CAIRO_IO("connecteur2", CP_NORTH, CP_SOUTH);
37
38 //Instanciation des elements structurels (cf. [NGuyen06])
39 CAIRO_LOTRS(...); // Transistor
40 CAIRO_LORES(...); // Resistance
41 CAIRO_LOCAP(...); // Capacite
42 CAIRO_PARCAP(...); // Capacite parasite
43
44
45
46 /* ***** */
47 /* Creation de la vue physique */
48 /* ***** */
49 CAIRO_HORIZONTAL_CONTAINER(nomInstance);
50
51
52
53 /* ***** */
54 /* Creation de la vue comportementale */
55 /* ***** */
56
57 // Declaration des parametres electriques
58 CAIRO_DECLARE_PARAM("paramElec1");
59 CAIRO_DECLARE_PARAM("paramElec2");
60 CAIRO_DECLARE_PARAM("paramElec3");
61
62 //Declaration des procedures de la vue comportementale
63 CAIRO_DECLARE_PROCEDURE("procedure1", CP_UNCOMPLETE,
64     "paramElec1", CP_IN,
65     "paramElec2", CP_OUT
66     );
67 CAIRO_DECLARE_PROCEDURE("direct_size", CP_COMPLETE,
68     "paramElec2", CP_IN,
69     "paramElec3", CP_INOUT
70     );
71
72
73

```

```
74 CAIRO_END_CREATE (nomGenerateur)
75
76
77
78 /* *****/
79 /* Dimensionnement */
80 /* *****/
81 CAIRO_BEGIN_DSES (nomGenerateur, char *nomInstance)
82
83
84
85 //Debut d'une procedure
86 CAIRO_BEGIN_PROCEDURE ("procedure1")
87
88 //Declaration des variables locales a la procedure
89 double varParamElec1; //variable recevant le parametre electrique d'entree
90 double varParamElec2; //variable contenant la valeur de sortie
91 double vth; //variable recevant une valeur rendue par la calculette
92 //ici par exemple : Vth d'un transistor
93
94 long var1=0; // variables communes aux
95 double var2=0.0; // 4 fonctions constructrices
96
97 //Recuperation des variables communes
98 //aux 4 fonctions constructrices et
99 //mises a disposition lors de la creation
100 CAIRO_GET_LOCAL_VARIABLE ("VAR1", var1);
101 CAIRO_GET_LOCAL_VARIABLE ("VAR2", var2);
102
103 //Tentative de lecture des parametres
104 //electriques d'entree
105 CAIRO_TRY_GET_VALUE ("paramElec1", varParamElec1)
106 IF_NO_VALUE //en cas de probleme
107 // Traitement...
108 ENDIF_NO_VALUE
109
110 //Appels a la calculette (ici vth d'un transistor)
111 vth = CAIRO_MOS_VTH (...); //cf. [deLamarre07]
112
113 //Calculs divers...
114 varParamElec2 = varParamElec1 * var1 - vth;
115
116 //Retour du/des resultat(s)
117 // --> vers autres fonctions constructrices...
118 CAIRO_SET_LOCAL_VARIABLE ("paramElec2", varParamElec2);
119 // --> vers niveau superieur...
120 CAIRO_SET_VALUE ("paramElec2", varParamElec2, CP_VALID);
121
122
```

```

123 // fin de la procedure
124 END_PROCEDURE
125
126 // Autre procedure
127 CAIRO_BEGIN_PROCEDURE("direct_size")
128
129 ...
130
131 END_PROCEDURE
132
133
134 CAIRO_END_DSES(nomGenerateur)
135
136
137
138 /*****/
139 /* Dessin des masques */
140 /*****/
141 CAIRO_BEGIN_SHAPE(nomGenerateur, char *nomInstance)
142
143 //Declaration des variables locales a SHAPE
144 long nbForme=0;
145 double var2=0.0;
146 double varParamElec2=0.0;
147
148 //Recuperation des variables communes
149 //aux 4 fonctions constructrices et
150 //mises a disposition lors de la creation
151 //ou du dimensionnement
152 CAIRO_GET_LOCAL_VARIABLE("VAR2", var2)
153 CAIRO_GET_LOCAL_VARIABLE("paramElec2", varParamElec2)
154
155 //Calcul du nombre de formes
156 nbForme = long(varParamElec2 / var2);
157
158 //Appels a GENERATE pour toutes les formes
159 //ici, i est l'indice de forme
160 for(int i=1; i<=nbForme; i++){
161     CAIRO_GENERATE(nomGenerateur, char *nomInstance, i)
162 }
163
164 CAIRO_END_SHAPE(nomGenerateur)
165
166 CAIRO_BEGIN_GENERATE(nomGenerateur, char *nomInstance, long i)
167
168 //Declaration des variables locales a GENERATE
169 double capaPar=0.0; // Capacite parasite quelconque
170 long dx=0, dy=0; // dimensions finales
171 motif_info *infoMotif1; // info sur le motif1 (cf. annexe B)

```

```
172 motif_info *infoMotif2;      // info sur le motif2 (cf. annexe B)
173 double varParamElec2;
174 double varParamStruct1;
175
176 //Recuperation des variables communes
177 //aux 4 fonctions constructrices et
178 //mises a disposition lors de la creation
179 //ou du dimensionnement
180 CAIRO_GET_LOCAL_VARIABLE("paramElec2", varParamElec2);
181 CAIRO_GET_LOCAL_VARIABLE("paramStruct1", varParamStruct1);
182
183 //Generation des motifs
184 infoMotif1 = motif1("modele1", paramStd [, paramContext ]);
185 infoMotif2 = motif2("modele2", paramStd [, paramContext ]);
186
187 //Creation de la boite d'aboutement
188 CAIRO_DEF_AB();
189
190 //Placement de motifs,
191 //avec calcul des dimensions et des parasites
192 CAIRO_PLACE("modele1","instance1",0,0)
193 capaPar = capaPar + infoMotif1 -> C_left;
194 dx = dx + infoMotif1 -> DXAB;
195
196 CAIRO_PLACE_TOP("modele2", "instance2");
197 capaPar = capaPar + infoMotif2 -> C_left;
198 dy = dy + infoMotif1 -> DYAB;
199
200 //routage de motifs et documentation implicite
201 //des elements structurels parasites
202 CAIRO_WIRE(...);
203 CAIRO_WIRE(...);
204
205 //Mise a jour de la vue structurelle
206 CAIRO_SET_LOTRS_INFO(...); //transistor
207 CAIRO_SET_LOCAP_INFO(...); //capacite
208 CAIRO_SET_LORES_INFO(...); //resistance
209
210 //Ajout des dimensions dans la fonction de forme
211 CAIRO_SET_DEVICE_DIMENSIONS(dx,dy)
212
213
214 CAIRO_END_GENERATE(nomGenerateur)
```


Annexe B

Structure de retour des générateurs de motifs

Cette annexe présente la structure de retour commune à tous les générateurs de motifs.

```
1 typedef struct MOTIF_INFO{
2
3     // Informations communes
4     long DXAB, DYAB;           //Dimensions symboliques de la boite d'aboutement
5     double DXAB_mu, DYAB_mu;  //Dimensions relles de la boite d'aboutement en microns
6
7     //Informations pour le transistor
8     double Wf, L;             //Dimensions d'un doigt.
9     double C_left;           //Capa parasites
10    double C_gate;           // des fils internes
11    double C_right;          // par rapport au substrat.
12    double half_dgg;         //demi largeur de diffusion interne
13    double de;               //largeur de diffusion externe
14
15    //Informations pour la capacite
16    double C;                 //Valeur de la capa.
17    double C_top;            //Capas parasites des armatures
18    double C_bottom;        //par rapport au substrat
19
20    //Informations pour la resistance
21    double R;                 //resistance d'un segment resistif
22    /* la capacite parasite d'un segment est donnee dans le champ C */
23
24 }motif_info;
```


Annexe C

Exemple de fichier DTR

Cette annexe détaille la syntaxe du fichier DTR en donnant le code d'un fichier simplifié pour une technologie imaginaire. Chaque ligne de ce fichier suit la syntaxe suivante :

```
1 NomDeLaRegle valeur0 valeur1 ... valeurN ## Reference dans le DRM
```

Par ailleurs, le caractère '#' précède un commentaire. Enfin, toutes les dimensions données ici sont en microns.

imaginaire.dtr

```
1 #####
2 ## Table Rules ##
3 #####
4 TABLE_1DM RULES
5
6 # Canal d'un transistor
7     RW_TRANS 0.15          ## 1-1
8     XW_TRANS 5000         ## Regle inhibee
9
10    RL_TRANS 0.15          ## 2-1
11    XL_TRANS 5000         ## Regle inhibee
12
13    RD_GATE 0.20           ## 3-1
14    RE_GATE_ACTI 0.20     ## 4-1
15
16
17 #Couches basses
18    RW_PWELL 0.0           ## Regle inhibee
19    RD_PWELL 0.0           ## Regle inhibee
20    RE_PWELL_ACTI 0.0     ## Regle inhibee
21
22    RW_NWELL 0.6           ## 5-1
23    RD_NWELL 0.6           ## 5-2
24    RE_NWELL_ACTI 0.35    ## 5-3
25
26    RW_NIMP 0.35           ## 6-1
```

```

27      RD_NIMP 0.35          ## 6-2
28      RE_NIMP_GATE 0.4      ## 6-3
29      RE_NIMP_POLY 0.2     ## 6-4
30      RE_NIMP_ACTI 0.20    ## 6-5
31      RE_NIMP_CONT 0.10    ## 6-6
32
33      RW_PIMP 0.35          ## 7-1
34      RD_PIMP 0.35          ## 7-2
35      RE_PIMP_GATE 0.4     ## 7-3
36      RE_PIMP_POLY 0.2     ## 7-4
37      RE_PIMP_ACTI 0.20    ## 7-5
38      RE_PIMP_CONT 0.10    ## 7-6
39
40      RD_ACTI 0.23          ## 8-1
41      RE_ACTI_CONT 0.09    ## 8-2
42      RE_ACTI_GATE 0.23    ## 8-3
43      RD_ACTI_CONT 0.16    ## 8-4
44      RD_ACTI_POLY 0.09    ## 8-5
45
46 #Polysilicium (sauf grille d'un transistor)
47      RW_POLY 0.15          ## 9-1
48      RD_POLY 0.48          ## 9-2
49      RE_POLY_CONT 0.09    ## 9-3
50
51 #Contact
52      RW_CONT 0.16          ## 10-1
53      RD_CONT 0.20          ## 10-2
54      RD_CONT_GATE 0.11    ## 10-3
55
56 #Metaux et vias
57      RW_ALU1 0.16          ## 20-1
58      RD_ALU1 0.20          ## 20-2
59      RE_ALU1_CONT 0.0 0.5  ## 20-3 (valeur1 si en fin de piste)
60      RE_ALU1_VIA1 0.01 0.5 ## 20-5 (valeur1 si en fin de piste)
61
62      RW_VIA1 0.19          ## 25-1
63      RD_VIA1 0.22          ## 25-2
64      #etc...
65
66 END_TABLE_1DM
67
68
69 #####
70 ## Table specifique au dispositif HIPORES ##
71 #####
72 TABLE_1DM HIPORES
73      RL_RES 5.0            ## R1-1
74      RW_RES 2.0            ## R1-2
75      RD_DOPRES 0.82        ## R2-1

```

```
76      RD_SILPROT 0.82      ## R3-1
77      RD_CONT_SILPROT 0.38  ## R4-1
78      RE_DOPRES_POLY 1.3    ## R5-1
79      RE_SILPROT_POLY 0.24  ## R6-1
80      RI_PIMP_POLY 1.00     ## R7-1 (RI = intersection)
81      RI_SILPROT_PIMP 0.34  ## R8-1 (RI = intersection)
82 END_TABLE_1DM
83
84 #####
85 ## Table specifique au dispositif MIM_MATRIX ##
86 #####
87 TABLE_1DM MIM_MATRIX
88
89 #Couches generiques
90      RW_TOPPLAT 3.5        ## C1-1
91      RW_BOTPLAT 4.5        ## C1-2
92      RD_TOPPLAT 3.5        ## C2-1
93      RD_BOTPLAT 2.5        ## C2-2
94      RE_TOPPLAT_TOPVIA 0.5 ## C3-1
95      RE_TOPALU_TOPVIA 0.7  ## C3-2
96      RE_TOPALU_TOPOUTVIA 0.7 ## C3-3
97      RE_BOTPLAT_TOPPLAT 0.5 ## C4-1
98      RE_BOTPLAT_BOTOUTVIA 0.3 ## C4-2
99      RW_TOPVIA 1.0         ## C5-1
100     RD_TOPVIA 1.0         ## C5-2
101     RW_BOTOUTVIA 1.0      ## C6-1
102     RD_BOTOUTVIA 1.0      ## C6-2
103     RW_TOPOUTVIA 1.0     ## C6-3
104     RW_TOPALU 1.0        ## C7-1
105     RD_TOPALU 2.0        ## C7-2
106
107 # Regles symboliques pour le routage
108 # definies par le concepteur
109     DIST_DOWN 20
110     DIST_TOP 8
111     DIST_OUT 4
112     LARG_DOWN 18
113     LARG_TOP 18
114     LARG_OUT 4
115
116 END_TABLE_1DM
117
118
119 TABLE_1DM LAYERS
120
121 ## La correspondance en les couches RDS et
122 ## les couches physiques est édonne dans
123 ## le fichier RDS (cf. page 68).
124
```

```

125
126 ## Motifs pour le transistor N
127     M1TRANS_N_00 RDS_POLY
128     M1TRANS_N_01 RDS_POLY
129     M1TRANS_N_02 RDS_POLY
130     M1TRANS_N_10 RDS_ACTIV
131     M1TRANS_N_11 RDS_NIMP
132     #...
133
134     LEFTFING_N_00 RDS_POLY
135     LEFTFING_N_01 RDS_POLY
136     LEFTFING_N_02 RDS_POLY
137     LEFTFING_N_10 RDS_ACTIV
138     LEFTFING_N_11 RDS_NIMP
139     #...
140
141 ## Motifs pour le transistor P
142     M1TRANS_P_00 RDS_POLY
143     M1TRANS_P_01 RDS_POLY
144     M1TRANS_P_02 RDS_POLY
145     M1TRANS_P_10 RDS_ACTIV
146     M1TRANS_P_11 RDS_PIMP
147     #...
148
149     LEFTFING_P_00 RDS_POLY
150     LEFTFING_P_01 RDS_POLY
151     LEFTFING_P_02 RDS_POLY
152     LEFTFING_P_10 RDS_ACTIV
153     LEFTFING_P_11 RDS_PIMP
154     #...
155
156 ## Routage du transistor
157     ALU_ROUT_H ALU2
158
159
160 ## Motifs resistifs
161     RES_SEGMENT_00 RDS_POLY
162     RES_SEGMENT_10 RDS_USER4      ## RDS_USER4 ==> SILPROT
163     RES_SEGMENT_20 RDS_USERS5    ## RDS_USERS5 ==> DOPRES
164     RES_SEGMENT_30 RDS_PIMP
165     RES_SEGMENT_31 RDS_PIMP
166
167
168 ## Motif Celem
169     CELEM_00 RDS_USER1      ## RDS_USER1 ==> BOTMIM
170     CELEM_01 RDS_USER1      ## RDS_USER2 ==> TOPMIM
171     CELEM_02 RDS_VIA6       ## RDS_USER3 ==> ALUCAP
172     CELEM_10 RDS_USER2
173     CELEM_11 RDS_USER2

```

```
174      CELEM_12 RDS_USER2
175      CELEM_13 RDS_USER2
176      CELEM_14 RDS_USER2
177      CELEM_20 RDS_VIA6
178      CELEM_21 RDS_VIA6
179      CELEM_30 RDS_USER3
180      CELEM_31 RDS_USER3
181      CELEM_32 RDS_VIA6
182      CELEM_33 RDS_NOLAYER      ## generation inhibee
183      CELEM_34 RDS_NOLAYER      ## generation inhibee
184
185 ## Couches de routage de MIM_MATRIX
186      MIM_MATRIX_ROUT_H ALU6
187      MIM_MATRIX_ROUT_B ALU7
188      MIM_MATRIX_ROUT_OUT ALU5
189      MIM_MATRIX_ROUTVIA_TOP CONT_VIA5
190      MIM_MATRIX_ROUTVIA_BOT CONT_VIA5
191
192 END_TABLE_1DM
193
194
195 TABLE_1DM ELECTRICAL      ## Process Electrical Characteristics
196      ALU1_PAR 0.053105 0.04331 0.1      ## Layer Cs(pF) Cp(pF) Rsq(Ohms/sq)
197      ALU2_PAR 0.03093 0.0336 0.1
198      ALU3_PAR 0.017 0.042 0.06
199      ALU4_PAR 0.0 0.0 00.0
200
201      CAPAMIM 2.060079754 0.1254525371      ## Capacitance Cs and Cp (fF/um2())
202      RPOLY 1200.0      ## RPOLY Element Rsq (ohm/sq)
203 END_TABLE_1DM
```


Annexe D

Syntaxe du fichier de réalisation

Dans cette annexe, nous allons approfondir la syntaxe du fichier de réalisation. Pour les paramètres de réalisation usuels, chaque ligne de ce fichier contient le nom d'un paramètre et sa valeur. Par exemple, pour le générateur `DP_CC`, le fichier de réalisation pourra contenir les lignes suivantes :

```
1 dummies True
2 ring False
```

Dans le cas où la valeur indiquée n'est pas une valeur possible, ou si un paramètre de réalisation n'est pas spécifié dans le fichier de réalisation, la valeur par défaut est prise en compte.

En ce qui concerne les dispositifs élémentaires qui réalisent un dessin des masques matriciel, le fichier de réalisation doit également contenir des informations relatives au placement des éléments de la matrice, et à leur routage. La section D.1 explique comment spécifier le placement et le routage pour les matrices de transistors issues des générateurs `MULTI_TR` et `MULTI_TR_SHD`. Dans la section D.2, nous verrons qu'en ce qui concerne le générateur de matrices de capacités `MIM_MATRIX`, les rapports désirés, ainsi que le placement des capacités élémentaires doivent également être spécifiés dans le fichier de réalisation.

D.1 fichier de réalisation pour les générateurs `MULTI_TR` et `MULTI_TR_SHD`

D.1.1 Placement

Pour ces deux générateurs, chacun des transistors de la vue structurelle est physiquement représenté par un certain nombre de sous-transistors reliés en parallèle. Un sous-transistor occupe une case (ou cellule) de la matrice finale et peut éventuellement être replié. Finalement, les nombres de sous-transistors réalisant chaque transistor de la vue structurelle définissent les rapports de largeur désirés :

$$\frac{W_i}{W_j} = \frac{Nst_i}{Nst_j} \quad (D.1)$$

où W_i et W_j sont les largeurs de deux transistors i et j , et Nst_i et Nst_j sont les nombres de sous-transistors les constituant respectivement.

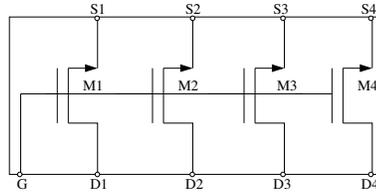


FIG. D.1 : Exemple : génération de 4 transistors avec MULTI_TR.

Dans le fichier de réalisation, il s'agit de donner une image du placement relatif des sous-transistors dans la matrice. Selon le nombre de sous-transistors que l'on placera pour chaque transistor de la vue structurelle, on obtiendra différents rapports de largeur. Lorsque que l'on veut placer un sous-transistor, on indique le numéro du transistor auquel il appartient (de 1 à N , N étant le paramètre structurel précisant le nombre de transistors à réaliser). On peut alors écrire, sur k lignes de j caractères, l'image d'une matrice $k \times j$. Par exemple, considérons le schéma électrique de la figure D.1 et supposons que l'on souhaite :

$$W_1 = 2 \times W_2 = 3 \times W_3 = 6 \times W_4 \quad (\text{D.2})$$

Pour obtenir ces rapports de largeur, on peut envisager les nombres de sous-transistors suivants :

- $Nst_1 = 6$
- $Nst_2 = 3$
- $Nst_3 = 2$
- $Nst_4 = 1$

Il s'agit alors de donner le placement relatif de ces 12 sous-transistors dans le fichier de réalisation. Par exemple, les lignes suivantes donneront le placement de la figure D.2.

```

1 begin placement
2 1111
3 1122
4 2334
5 end placement

```

1	1	1	1
1	1	2	2
2	3	3	4

FIG. D.2 : Exemple de placement obtenu.

Comme on peut le voir, la description du placement relatif des sous-transistors doit se trouver entre les lignes "begin placement" et "end placement". Par ailleurs, seules sont autorisées les matrices rectangles. Dans le cas où le nombre de sous-transistors ne permet pas d'obtenir une matrice rectangle, il est possible d'insérer des sous-transistors fictifs dans la matrice image grâce à la lettre 'D'. Tous les connecteurs de ces transistors fictifs seront physiquement reliés à la masse afin de les maintenir en régime bloqué.

Par défaut, les sous-transistors sont dessinés en commençant par une diffusion de source. Pour optimiser l'appariement (cf. chap. 5), il est possible de les faire commencer par une diffusion de

drain en faisant précéder leur numéro du signe '-' dans la matrice image.

Notons enfin que la cohérence entre le placement spécifié et le paramètre structurel N est vérifiée.

D.1.2 Routage

Physiquement, une matrice de k lignes et de j colonnes est réalisée par k empilages de j sous-transistors. Pour la router, on fait passer des nappes de fils :

- entre chaque empilage,
- au dessus de l'empilage le plus haut,
- en dessous de l'empilage le plus bas.

Ainsi, $(k + 1)$ nappes de fils permettent de router la matrice.

Dans le fichier de réalisation, il faut indiquer quelles sont les pistes qui passent dans chaque nappe. Pour cela, on fait référence à une piste en indiquant le nom du signal électrique qu'elle porte : D_i , G_i , S_i ou B (i est le numero d'un transistor). On donne alors une description du routage sur $(k + 1)$ lignes, chaque ligne donnant les pistes qui passent dans une même nappe. Sur une même ligne, l'ordre de spécification des pistes correspond à leur placement relatif dans la nappe, du haut vers le bas.

Si l'on reprend l'exemple de la figure D.2, un fichier de réalisation contenant les lignes suivantes conduira à la configuration de la figure D.3.

```

1 begin routing
2 B D1 S1 * *
3 G1 S1 D1 G2 *
4 G2 D2 S2 D3 S3
5 G3 D4 G4 S4 B
6 end routing

```

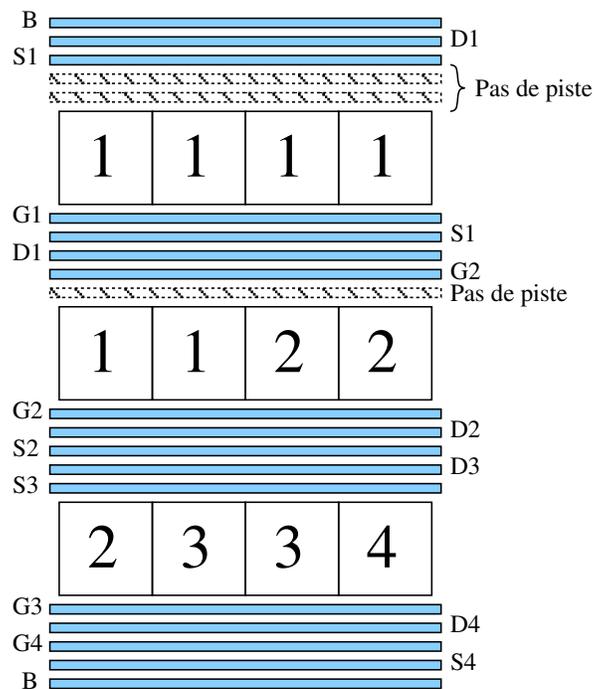


FIG. D.3 : Exemple de routage obtenu.

La description du routage doit se trouver entre les lignes "begin routing" et "end routing".

La première ligne décrit la nappe la plus haute, la deuxième celle en dessous du premier empilage, et ainsi de suite. En insérant le caractère '*' dans la description, on peut laisser une piste vide dans la nappe. Ceci peut notamment permettre d'améliorer l'appariement en maintenant un espace constant entre les différents empilages. Par ailleurs, la description doit obligatoirement contenir $(k + 1)$ lignes. Il est cependant possible de laisser une ligne vide (c'est-à-dire une ligne contenant uniquement un retour chariot) afin de spécifier une nappe vide. Enfin, précisons que physiquement, deux pistes adjacentes sont distantes de 5λ .

Dans le cas où les grilles des transistors sont reliées (paramètre structurel `sameGrid` valant `True`), on fera apparaître, pour le routage de la grille commune, des pistes portant simplement le nom G. Dans le cas du générateur `MULTI_TR_SHD`, on fera de même apparaître des pistes S pour la source commune, ou des pistes D pour le drain commun, selon la valeur du paramètre `sharedDrain`.

Avec la syntaxe que nous proposons, il est possible que le concepteur spécifie un placement et un routage qui conduisent à un conflit de routage. Pour mieux comprendre ce que nous appelons un conflit de routage, étudions la configuration de la figure D.4.

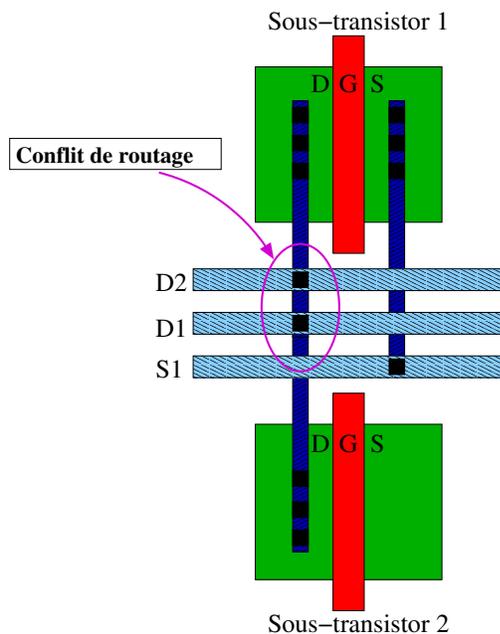


FIG. D.4 : Conflit de routage.

Sur cette figure, le drain du sous-transistor 1 est routé par une piste qui se trouve en dessous de celle qui route le drain du sous-transistor 2. Dans ce cas, les fils métalliques internes des motifs de doigts de transistors se recouvrent et il y a un court-circuit entre les drains. Si le placement et le routage donnés impliquent des conflits de routage, les générateurs `MULTI_TR` et `MULTI_TR_SHD` génèrent un message d'erreur indiquant les sous-transistors et les pistes concernés.

D.2 fichier de réalisation pour le générateur MIM_MATRIX

D.2.1 Rapports

Pour le générateur MIM_MATRIX, il est nécessaire de donner les rapports de capacités désirés dans le fichier de réalisation. En effet, ces rapports donnent la valeur de chaque capacité par rapport à la capacité élémentaire qui est spécifiée en tant que paramètre électrique d'entrée. Dans la mesure où ceux-ci ne sont pas nécessairement entiers, il ne peuvent pas être directement déduits du placement. Pour quatre capacités, on pourra par exemple trouver dans le fichier de réalisation les lignes suivantes :

```
1 begin ratios
2 3 2.5 4 1
3 end ratios
```

On voit dans cet exemple que les rapports sont donnés sur une même ligne et entre les lignes "begin ratios" et "end ratios". Le générateur vérifie que le nombre de rapports spécifiés correspond au paramètre structurel N.

D.2.2 Placement et routage

Pour le générateur de matrices de capacités MIM_MATRIX, le fichier de réalisation doit contenir une description du placement relatif des capacités unitaires et non-unitaires. A l'instar de ce que nous avons vu pour les générateurs de matrices de transistors, celles-ci y sont référencées par le numéro de la capacité de la vue structurelle à laquelle elles appartiennent. Ici encore, la matrice doit être rectangle et le caractère 'D' permet d'insérer des capacités fictives.

Une capacité non-unitaire occupe deux cellules adjacentes de la matrice. Dans la matrice image du fichier de la réalisation, on indique l'emplacement d'une capacité non-unitaire en faisant précéder, sur deux cellules adjacentes (adjacence verticale ou horizontale), le numéro de la capacité par un signe '-'. Par exemple, la description suivante conduit au placement de la figure 5.20 (page 48).

```
1 begin placement
2 DDDDD
3 D221D
4 D21-2D
5 D12-2D
6 DDDDD
7 end placement
```

Le routage de la matrice de capacités se fait automatiquement par des canaux de routage. Ces canaux sont soit verticaux, soit horizontaux. Leur orientation dépend notamment de celle

des capacités non-unitaires. Dans l'exemple précédent, la capacité non-unitaire étant orientée verticalement, seuls des canaux verticaux sont envisageables. Dès lors, si plusieurs capacités non-unitaires sont nécessaires, elles doivent toutes avoir la même orientation. Dans le cas contraire, le générateur `MIM_MATRIX` produit un message d'erreur. Si aucune capacité non-unitaire n'est présente dans la matrice, les canaux suivent une orientation qui minimise la surface globale de la matrice.

Pour finir, signalons que la cohérence entre le placement décrit et les rapports spécifiés est vérifiée par le générateur `MIM_MATRIX`. En effet, une capacité de rapport R réel doit être physiquement représentée par :

- R capacités unitaires si R est entier.
- $\lfloor R - 1 \rfloor$ capacités unitaires, et une capacité non-unitaire de rapport $(R + 1) - \lfloor R \rfloor$ (où $\lfloor x \rfloor$ est la partie entière de x) si R est non-entier.

Liste des publications

- DiaaEldin Khalil, Mohamed Dessouky, Vincent Bourguet, Marie-Minerve Louërat, Andreia Cathelin et Hani Ragai, "Evaluation of Capacitor Ratios in Automated Accurate Common-Centroid Capacitor Arrays", *International Symposium on Quality Electronic Design (ISQED'05)*, pp. 143-147, San Jose, USA, Mars 2005.
- Vincent Bourguet, Laurent De Lamarre et Marie-Minerve Louërat, "Analog IC Design with a Library of Parameterized Device Generators", *International Conference on Design of Circuits and Integrated Systems (DCIS'2004)*, pp. 739-744, Bordeaux, France, Novembre 2004.
- Pierre Nguyen Tuong, Vincent Bourguet, Laurent De Lamarre, Marie-Minerve Louërat et Alain Greiner, "A Language to Design Generators of Analog Functions", *Forum on Specification & Design Languages (FDL'2004)*, pp. 30-31, Lille, France, Septembre 2004.
- DiaaEldin Khalil, Mohamed Dessouky, Bourguet Vincent, Marie-Minerve Louërat, Andreia Cathelin et Hani Ragai, "Compensated Layout for Automated Accurate Common-Centroid Capacitor Arrays", *International Conference on Electrical Electronic and Computer Engineering (ICEEC'2004)*, pp. 481-484, Le Caire, Egypte, Septembre 2004.
- Vincent Bourguet, Laurent De Lamarre et Marie-Minerve Louërat, "A Layout-Educated Analog Design Flow", *Midwest Symposium on Circuits and Systems (MWSCAS'2004)*, pp. 482-488, Hiroshima, Japon, Juillet 2004.
- Vincent Bourguet, Marie-Minerve Louërat et Alain Greiner, "Composants analogiques déformables pour CAIRO+", *Colloque du GDR CAO de circuits et systèmes intégrés*, pp. 25-28, Paris, France, Mai 2002.

Bibliographie

- [Aboushady02] Hassan Aboushady. *Design for Reuse of Current-Mode Continuous-Time $\Sigma\Delta$ Analog-to-Digital Converters*. PhD thesis, Université Pierre et Marie Curie, Lab. LIP6, Paris, 2002.
- [Agarwal04] Anuradha Agarwal, Hemanth Sampath, Veena Yelamanchili, and Ranga Vemuri. Fast and Accurate Parasitic Capacitance Models for Layout-Aware Synthesis of Analog Circuits. In *Proc. Design Automation Conf.*, pages 145–150, June 2004.
- [Bastos96] J. Bastos, M. Steyaert, B. Graindourze, and W. Sansen. Matching of MOS Transistors with Different Layout Styles. In *Proc. IEEE Int. Conf. on Microelectronic Test Structures*, pages 17–18, March 1996.
- [Bourguet02] Vincent Bourguet, Marie-Minerve Rosset-Louërat, and Alain Greiner. Composants analogique déformables pour CAIRO+. In *Troisième Colloque du GDR CAO de circuits intégrés et systèmes*, pages 25–28, Paris, May 2002. Laboratoire LIP6.
- [Bourguet04a] Vincent Bourguet, Laurent de Lamarre, and Marie-Minerve Louërat. A Layout-Educated Analog Design Flow. In *Proc. IEEE Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 482–488, Hiroshima, Japan, July 2004.
- [Bourguet04b] Vincent Bourguet, Laurent de Lamarre, and Marie-Minerve Louërat. Analog IC Design with a library of Parameterized device Generators. In *Proc. Int. Conf. on Design of Circuits and Integrated Systems*, Bordeaux, November 2004.
- [Buisson98] O. Buisson, G. Morin, F. Paillardet, and E. Mazaleyra. A New Characterization Method for Accurate Capacitor Matching Measurements Using Pseudo-Floating Gate Test Structures in Submicron CMOS and BICMOS Technologies. In *Proc. IEEE Int. Conf. on Microelectronic Test Structures*, volume 11, pages 223–227, March 1998.
- [Cadence07] Cadence, http://www.cadence.com/products/custom_ic/neocell. *Virtuoso Neo-Cell Analog Physical Synthesis*, 2007.
- [Castro02] R. Castro-Lopez, F.V. Fernandez, F. Medeiro, and A. Rodríguez-Vázquez. Generation of Technology-Independent Retargetable Analog Blocks. *Analog In-*

- egrated Circuits and Signal Processing, Kluwer Academic Publishers, 33 :157–170, 2002.*
- [Castro06] R. Castro-Lopez, F.V. Fernandez, and A.R. Vazquez. *Reuse-Based Methodologies and Tools in the Design of Analog and Mixed-Signal Integrated Circuits*. Springer, 2006.
- [Cheng99] Y. Cheng and C. Hu. *MOSFET Modeling and BSIM3 User's Guide*. Kluwer Academic Publishers, first edition, 1999.
- [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.
- [deLamarre02] Laurent de Lamarre. *CAIRO+ : Intégration du modèle Bsim3v3, stage de DEA*. Université Pierre et Marie Curie, Lab. LIP6, dept. ASIM, Paris, 2002.
- [deLamarre06] Laurent de Lamarre, Marie-Minerve Louërat, and Andreas Kaiser. Optimizing Resistances and Capacitances of a Continuous-Time Sigma-Delta ADC . In *Proc. IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 419–422, Nice, December 2006.
- [deLamarre07] Laurent de Lamarre. *Conception d'un circuit de conversion analogique-numérique de type $\Sigma\Delta$ sous forme d'IP analogique*. PhD thesis, Université Pierre et Marie Curie, Lab. LIP6, Paris, 2007.
- [Dessouky00] Mohamed Dessouky, Marie-Minerve Louërat, and Jacky Porte. Layout-Oriented Synthesis of High Performance Analog Circuits. In *Proc. Design Automation and Test in Europe Conf.*, pages 53–57, March 2000.
- [Dessouky01a] Mohamed Dessouky. *Design for Reuse of Analog Circuits. Case Study : Very Low-Voltage $\Delta\Sigma$ Modulator* . PhD thesis, Université Pierre et Marie Curie, Lab. LIP6, Paris, 2001.
- [Dessouky01b] Mohamed Dessouky and Andreas Kaiser. Very Low-voltage $\Delta\Sigma$ Modulator with 88dB Dynamic Range using Local Switch Bootstrapping. *IEEE J. of Solid-State Circuits*, 36(3), March 2001.
- [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.
- [Dunga07] Mohan V. Dunga, Wenwei Yang, Xuemei Xi, Jin He, Weidong Liu, Xiaodong Gin, Jeff J. Ou, Mansun Chan, Ali N. Niknejad, and Chenming Hu. *BSIM4.6.1 MOSFET Model - User's Manual*. <http://www-device.eecs.berkeley.edu/~bsim3/bsim4.get.html>, 2007.
- [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.

- [Galup02] Carlos Galup-Montoro, Marcio Cherem Schneider, and Rafael Matos Coitinho. Resizing rules for mos analog-design reuse. *IEEE Des. Test*, 19(2) :50–58, 2002.
- [Gaubert05] Jean Gaubert, Sylvain Bourdel, Philippe Pannier, Hervé Barthélémy, Marc Battista, and Mathieu Egels. Méthodes de conception pour amplificateurs faible bruit pour systèmes intégrés uwb 3.1-10.6 ghz. In *Proc. Traitement Analogique de l'Information du Signal et ses Applications (TAISA)*, pages 17–20, October 2005.
- [Gielen91] Georges Gielen and Willy Sansen. *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Kluwer Academic Publishers, first edition, 1991.
- [Gielen00] Georges G. E. Gielen and Rob A. Rutenbar. Computer-Aided Design of Analog and Mixed-signal Integrated Circuits. *IEEE Proceedings*, 88(12) :1825–1852, December 2000.
- [Graeb01] Helmut E. Graeb, Stephan Zizala, Josef Eckmueller, and Kurt Antreich. The sizing rules method for analog integrated circuit design. In *Proc. IEEE Int. Conf. on Computer-Aided-Design*, pages 343–349, November 2001.
- [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.
- [Hammouda04] Sherif Hammouda, Mohamed Dessouky, Mohamed Tawfik, and Wael M. Badawy. Analog ip migration using design knowledge extraction. In *Proc. IEEE Custom Integrated Circuits Conf.*, pages 333–336, October 2004.
- [Hammouda06] Sherif Hammouda, Hazem Said, Mohamed Dessouky, Mohamed Tawfik, Quang Nguyen, Wael M. Badawy, Hazem M. Abbas, and Hussein I. Shahein. Chameleon art : a non-optimization based analog design migration framework. In *Proc. Design Automation Conf.*, pages 885–888, 2006.
- [Hamour03] M. Hamour, R. Saleh, S. Mirabbasi, and A. Ivanov. Analog IP Design Flow for SoC Applications. In *Proc. IEEE Int. Symposium on Circuits and Systems*, volume 4, pages 676–679, May 2003.
- [Hastings01] Alan Hastings. *The Art of Analog Layout*. Prentice Hall, 2001.
- [Hershenson98] Hershenson, Boyd, and Lee. GPCAD : A Tool for CMOS Op-Amp Synthesis. In *Proc. IEEE Int. Conf. on Computer-Aided-Design*, pages 296–303, 1998.
- [Iskander04] Ramy Iskander, Laurent de Lamarre, Andreas Kaiser, and Marie-Minerve Louërat. Design Space Exploration for Analog IPs using CAIRO+ . In *The International Conference on Electrical Electronic and Computer Engineering*, pages 481–484, Cairo, Egypt, September 2004.

- [Iskander06] Ramy Iskander, Marie-Minerve Louërat, and Andreas Kaiser. Dimensionnement Automatique d'un Circuit Analogique à l'aide des Transistors de Référence . In *Proc. Traitement Analogique de l'Information du Signal et ses Applications (TAISA)*, pages 57–60, Strasbourg, October 2006.
- [ITRS06] Semiconductor Industry Association. International technology roadmap for semiconductors. <http://www.itrs.net/reports.html>, 2006.
- [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.
- [Khalil04] DiaoEldin Khalil, Mohamed Dessouky, Vincent Bourguet, Marie-Minerve Louërat, Andreia Cathelin, and Hani Ragai. Compensated Layout for Automated Accurate Common-Centroid Capacitor Arrays . In *The International Conference on Electrical Electronic and Computer Engineering*, pages 481–484, Cairo, Egypt, September 2004.
- [Khalil05] DiaoEldin Khalil, Mohamed Dessouky, Vincent Bourguet, Marie-Minerve Louërat, Andreia Cathelin, and Hani Ragai. Evaluation of Capacitor Ratios in Automated Accurate Common-Centroid Capacitor Array . In *International Symposium on Quality Electronic Design*, page 143, March 2005.
- [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.
- [Levi06] T. Levi, N. Lewis, J.Tomas, and P. Fouillat. Scaling Rules For Mos Analog Design Reuse. In *Proc. Int. Conf. Mixed Design of Integrated Circuits and Systems (MIXDES)*, volume 1, pages 378–382, June 2006.
- [Lira05] José G.A. Lira, Adeilton C. Oliveira Júnior, Raimundo C.S. Freire, Ioshiaki Doi, Benedito A. Luciano, and Jacobus W. Swart. Dynamic characterization of thermo-resistive micro-sensor. In *Proc. IEEE Instrumentation and Measurement Technology Conf. (IMTC)*, pages 1647–1651, May 2005.
- [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.

- [Maloberti94] Franco Maloberti. Layout of analog and mixed analog-digital circuits. In Jose E. Franca and Yannis Tsividis, editors, *Design of Analog-Digital VLSI Circuits for Telecommunication and Signal Processing*, chapter 11. Prentice Hall, 1994.
- [Maloberti01] Fanco Maloberti. *Analog Design for CMOS VLSI Systems*. Kluwer Academic Publishers, first edition, 2001.
- [Matlab07] MathWorks, Inc. *MATLAB and SIMULINK User's Guides*, 2007.
- [Mathias94] H. Mathias, L.Hébrard, J.Berger-Toussan, G.Jacquemod, F.Gaffiot, and M. Le Helley. Automatic Layout Generation for CMOS Analog Transistors. In *Proc. European Design Automation Conf.*, pages 54–58, 1994.
- [MEDEA05] MEDEA+, <http://www.medeaplus.org>. *Design Automation Roadmap*, July 2005.
- [Naiknaware99] Ravindranath Naiknaware and Terri S. Fiez. Automated hierarchical CMOS analog circuit stack generation with intramodule connectivity and matching considerations. *IEEE J. of Solid-State Circuits*, 34(3) :304–317, March 1999.
- [NGuyen04] Pierre Nguyen Tuong, Vincent Bourguet, Laurent de Lamarre, Marie-Minerve Louërat, and Alain Greiner. A Language to Design Generators of Analog Functions. In *Forum on Specifications and Design Languages (FDL)*, pages 30–31, Lille, September 2004.
- [NGuyen06] Pierre Nguyen Tuong. *Définition et Implémentation d'un Langage de Conception de Composants Analogiques Réutilisables*. PhD thesis, Université Pierre et Marie Curie, Lab. LIP6, Paris, 2006.
- [Ochotta96] Emil S. Ochotta, Rob A. Rutenbar, and L. Richard Carley. Synthesis of high-performance analog circuits in ASTRX/OBLX. *IEEE Trans. Computer-Aided Design*, 15(3) :273–294, March 1996.
- [Owen95a] Bryn R. Owen. *BALLISTIC: An Analog Layout Language*. University of Toronto, Toronto, Canada, beta+ edition, September 1995.
- [Owen95b] 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.
- [Pelgrom89] Marcel J.M. Pelgrom, AADC.J. Duinmaijer, and Anton P.G. Welbers. Matching Properties of MOS Transistors. *IEEE J. of Solid-State Circuits*, pages 1433–1440, October 1989.
- [Pelgrom94] M. Pelgrom, J. Rens, M. Vertregt, and M. Dijkstra. A 25-Ms/s 8-bit CMOS A/D Converter for Embedded Application. *IEEE J. of Solid-State Circuits*, 29 :879–886, 1994.
- [Pelgrom98] Marcel J.M. Pelgrom, AAD C.J. Duinmaijer, and Anton P.G. Welbers. Transistor Matching in Analog CMOS Applications. *Proc. IEEE Int. Electron Devices Meeting*, pages 915–918, 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://www-asim.lip6.fr/publications/1994/index.gb.html>, July 1994.
- [Phelps00] R. Phelps, M. Krasnicki, R.A. Rutenbar, L.R. Carley, and J.R Hellums. Anaconda : simulation-based synthesis of analog circuits via stochastic pattern search. In *IEEE Trans. Computer-Aided Design*, pages 703–717, June 2000.
- [Porte07] Jacky Porte. *OCEANE : Outils pour la Conception et l'Enseignement des circuits intégrés ANalogiquEs – reference manual*. Université Pierre et Marie Curie, Lab. LIP6, Paris, January 2007.
- [Razavi01] Behzad Razavi. *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2001.
- [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.
- [Saleh06] Resve Saleh, Steve Wilton, Shahriar Mirabbasi, Alan Hu, Mark Greenstreet, Guy Lemieux, Partha Pratim Pande, Christian Grecu, and Andre Ivanov. System-on-Chip : reuse and Integration . *IEEE Proceedings*, 94(6) :1050–1069, June 2006.
- [Sayed02] DiaaEldin Sayed and Mohamed Dessouky. Automatic Generation of Common-Centroid Capacitor Arrays with Arbitrary Capacitor Ratio. In *Proc. Design Automation and Test in Europe Conf.*, March 2002.
- [Shyu82] Jyn B. Shyu, Gabor C. Temes, and Kung Yao. Random errors in mos capacitors. *IEEE J. of Solid-State Circuits*, 17(6) :1070–1076, December 1982.
- [SYSTEM07] *System-C AMS Initiative's home page*. <http://www.systemc-ams.org>, 2007.
- [Van der Plas01] Geert Van der Plas, Geert Debyser, Francky Leyn, Koan Lampaert, Jan Vandebussche, Georges G. E. Gielen, Willy Sansen, Petar Veselinovic, and Domine Leenaerts. AMGIE– A Synthesis Environment for CMOS Analog Integrated Circuits. *IEEE Trans. Computer-Aided Design*, 20(9) :1037–1058, September 2001.
- [Vassiliou96] I. Vassiliou, H. Chang, A. Demir, E. Charbon, P. Miliozzi, and A. Sangiovanni-Vincentelli. A video driver system designed using a top-down, constraint-driven methodology. In *Proc. IEEE Int. Conf. on Computer-Aided-Design*, pages 463–468, November 1996.
- [VHDL07] IEEE, <http://www.vhdl.org/analog/>. *IEEE 1076.1 (VHDL-AMS) Working Group's home page*, 2007.