

Conception d'un *SAR-ADC* de 10/13-bit à  $83\text{kS}\cdot\text{s}^{-1}$  avec fonction  
de *dithering*

par

Quentin SAUVÉ

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE  
AVEC MÉMOIRE EN GÉNIE ÉLECTRIQUE  
M.Sc.A.

MONTRÉAL, LE 8 JUIN 2017

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC



Quentin SAUVÉ, 2017



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

**PRÉSENTATION DU JURY**

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Nicolas Constantin, directeur de mémoire  
Département de génie électrique à l'École de Technologie Supérieure

M. Frédéric Nabki, co-directeur  
Département de génie électrique à l'École de Technologie Supérieure

Mme. Lyne Woodward, présidente du jury  
Département de génie électrique à l'École de Technologie Supérieure

M. Vahé Nerguizian, membre du jury  
Département de génie électrique à l'École de Technologie Supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 25 MAI 2017

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE



## **AVANT-PROPOS**

Ce document a pour objectif de présenter les travaux de recherche dans le cadre de ma maîtrise en génie électrique à l'École de Technologie Supérieure (ÉTS) à Montréal. Ce travail s'inscrit dans la continuité d'un projet réalisé par un étudiant français en stage à l'Université du Québec À Montréal (UQAM) en 2015. J'ai eu l'opportunité d'accéder à ce programme de maîtrise dans le cadre d'un accord de double diplôme entre l'ÉTS et mon école d'origine l'ENSEIRB MATMECA (Bordeaux, FRANCE).



## REMERCIEMENTS

Avant toute chose, je tiens à remercier mes directeurs de mémoire MM. Nicolas Constantin et Frédéric Nabki pour la confiance et le soutien qu'ils m'ont accordés sur ce projet. Je remercie également M. Mohammad Taherzadeh pour ses conseils sur les *SAR ADCs* et le "dithering" et M. Gabriel Morin-Laporte pour son aide dans la mise en place du flot de conception automatisé.

Je tiens également à remercier vivement M. Feng Q. Lui, ingénieur à la société canadienne de micro-électronique (en abrégé *CMC*), pour l'aide indispensable qu'il m'a apporté sur le plan technique et logiciel. Ces remerciements vont aussi à M. Himani Tyagi, ingénieur chez Cadence, pour l'aide qu'il m'a apporté sur le simulateur *AMS*.

Un grand merci à Damien Favre pour sa disponibilité dans la transmission des informations entre son projet et le mien. Je remercie M. Claude Chagnon, technicien informatique de l'UQAM pour son aide et sa disponibilité.

Je remercie également le corps professoral de mon école d'origine, l'ENSEIRB-MATMECA (Bordeaux, FRANCE), pour la confiance qu'ils m'ont accordé dans mon projet de double diplôme avec l'ÉTS Montréal.

Je remercie M. Florian Jacquet pour son aide dans la relecture de ce rapport.

Enfin, je remercie mes parents et ma sœur pour tout le soutien qu'il mon apporté durant mes études.





# CONCEPTION D'UN SAR-ADC DE 10/13-BIT À 83KS·S<sup>-1</sup> AVEC FONCTION DE DITHERING

Quentin SAUVÉ

## RÉSUMÉ

L'objectif de ce projet est d'étudier les possibilités d'implémentation d'un système de résolution variable sur les *ADC* (convertisseur analogique numérique) de type *SAR* (registre à approximations successives). Ce système utilise une technique dite de *dithering* pour porter la résolution native de 10 bits à 13 bits. Afin d'évaluer les performances de cette technique, un circuit a été conçu sur la technologie *GlobalFoundries CMOS 130nm*. Ce rapport revient sur les différentes étapes de conception, de simulation, de fabrication et de test d'un tel convertisseur. Les simulations informatiques ont montré des performances intéressantes en mode natif avec un *ENOB* moyen de 9.88 bits. En mode *dithering* la résolution est portée à 11.4 bits en simulation. Malheureusement, des problèmes sur les puces fabriquées n'ont pas permis de valider le concept expérimentalement.

La dernière partie de ce mémoire présente des considérations plus théoriques et s'attache à maximiser les performances du concept en proposant notamment le remplacement du *dithering* sous forme de moyennage par une recherche dichotomique. Dans notre cas, l'utilisation d'un tel algorithme réduirait le nombre de conversions nécessaires de 15 à 4 pour un résultat identique.

Enfin, un article intitulé "*A Calibration-Free 13-Bit 0.9 V Differential SAR-ADC with Hybrid DAC and Dithering*" a été accepté à la conférence *IEEE International Symposium on Circuits and Systems 2017* (en abrégé *ISCAS*).

**Mots clés:** *ADC*, *SAR*, *dithering*, reconfigurabilité, *DAC* hybride, alimentation multiple



## DESIGN OF A 83KS.S<sup>-1</sup> 10/13-BIT SAR-ADC WITH DITHERING FEATURE

Quentin SAUVÉ

### ABSTRACT

The main goal of this projet is to study the implementation possibilities of a variable resolution technique for *SAR-ADC* (successive approximation register - analog to digital converter). This system uses a *dithering* technique to increase the native resolution from 10 bits to 13 bits. In order to evaluate the performances of such a system, a circuit has been designed in *Global-Foundries CMOS* 130nm technology. This report describes the design, simulation, fabrication and test of such a converter. Computer simulations revealed interesting preformances in native mode with a medium *ENOB* of 9.88 bits. In simulation, the resolution reaches 11.4 bits with the *dithering* technique. Unfortunately, problems in the fabricated chips have prevented the experimental validation of this concept.

The last section of this report is based on theoritical considerations and offers the improvement of performance by replacing the averaging *dithering* by a binary search algorithm. In this case, such a modification would reduce the number of conversions from 15 to 4 to have the same result.

A paper called "*A Calibration-Free 13-Bit 0.9 V Differential SAR-ADC with Hybrid DAC and Dithering*" has been accepted at the 2017 2017 IEEE International Symposium on Circuits and Systems (ISCAS).

**Keywords:** *ADC*, *SAR*, dithering, configurability, hybrid *DAC*, multiple supply voltage



## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
CHAPITRE 1 ÉTAT DE L'ART DES SAR ADC .....	5
1.1 Principe de fonctionnement des SAR ADC .....	5
1.2 Revue de la littérature .....	8
CHAPITRE 2 TRAVAUX ANTÉRIEURS .....	13
2.1 Présentation succincte .....	13
2.2 Expérimentation .....	17
CHAPITRE 3 CAHIER DES CHARGES .....	19
CHAPITRE 4 NOUVELLE ARCHITECTURE PROPOSÉE .....	21
4.1 Le <i>Dithering</i> .....	21
4.2 Nouvelle Architecture .....	25
4.2.1 Architecture générale .....	26
4.2.2 Testabilité .....	30
CHAPITRE 5 CONCEPTION DE LA PUCE .....	31
5.1 Éléments analogiques .....	31
5.1.1 <i>Sample-and-Hold</i> .....	31
5.1.2 Convertisseur numérique/analogique ( <i>DAC</i> ) .....	34
5.1.3 Comparateur .....	36
5.2 <i>Level-Shifters</i> .....	37
5.3 Éléments numériques .....	39
5.3.1 Flot de conception des blocs numériques .....	39
5.3.2 Cœur logique .....	40
5.3.2.1 Le SAR .....	40
5.3.2.2 Les convertisseurs binaire-thermomètre .....	43
5.3.3 <i>DAC Drivers</i> .....	44
5.4 Éléments de mesure .....	44
5.4.1 Amplificateur Opérationnel .....	45
5.4.2 Points de test numérique .....	47
5.5 Distribution de l'alimentation .....	48
5.6 Présentation globale du <i>layout</i> et du flot de conception .....	49
CHAPITRE 6 TESTS ET SIMULATIONS .....	55
6.1 Tests sur les éléments analogiques .....	55
6.2 Tests sur les éléments numériques .....	56
6.2.1 Simulations pré-synthèse .....	57

6.2.2	Simulations post-synthèse .....	58
6.2.2.1	<i>Logic Core</i> .....	58
6.2.2.2	<i>DAC Driver</i> .....	60
6.3	Simulations Globales .....	60
6.3.1	Paramètres caractéristiques des <i>ADC</i> .....	60
6.3.2	Résultats .....	63
CHAPITRE 7 EXPÉRIMENTATIONS ET TESTS DE LA PUCE .....		71
7.1	Banc de test .....	71
7.2	Résultats expérimentaux .....	73
CHAPITRE 8 BILAN SUR LE FONCTIONNEMENT DU CONVERTISSEUR .....		77
8.1	Considérations sur les limites du <i>dithering</i> .....	77
8.2	Évolution du projet .....	81
CONCLUSION .....		85
ANNEXE I DESSINS DES MASQUES DE LA PUCE .....		87
ANNEXE II RÉSUMÉ DES SPÉCIFICATIONS DU CONVERTISSEUR .....		91
ANNEXE III AMPLIFICATEUR OPÉRATIONNEL .....		93
BIBLIOGRAPHIE .....		95

## LISTE DES TABLEAUX

		Page
Tableau 1.1	Peignes de condensateurs des <i>DAC</i> traditionnel ou avec capacité série .....	11
Tableau 5.1	Évolution des codes de sortie du registre <i>Shiftreg</i> .....	41
Tableau 5.2	Évolution des codes de sortie des convertisseurs binaire-thermomètre .....	43
Tableau 5.3	Numérotations et dimensions des sous-circuits constitutifs de la puce .....	51
Tableau 5.4	Numérotations et dimensions des sous-circuits constitutifs de la partie analogique du convertisseur.....	52
Tableau 6.1	Évaluation de la consommation des circuits logiques .....	67
Tableau 6.2	Résumé des performances simulées de l' <i>ADC</i> .....	70
Tableau 8.1	Lien entre la séquence de capacités utilisées et la résolution maximale accessible.....	78





## LISTE DES FIGURES

	Page
Figure 0.1	Synopsis des grandes étapes du projet ..... 2
Figure 1.1	Architecture d'un <i>ADC SAR</i> standard..... 5
Figure 1.2	Chronogramme théorique du déroulement d'un algorithme <i>SAR</i> sur 5 bits ..... 6
Figure 1.3	Schéma des connexions entre <i>S/H</i> , <i>DAC</i> et comparateur ..... 7
Figure 1.4	Architecture d'un <i>ADC SAR</i> avec <i>DAC</i> capacitif..... 8
Figure 1.5	Architecture d'un <i>DAC</i> à capacité série (ou <i>split capacitor</i> ) ..... 11
Figure 2.1	Architecture de la version 2015 de l' <i>ADC</i> ..... 14
Figure 2.2	Architecture interne des <i>DAC</i> hybrides ..... 15
Figure 2.3	Exemple d'utilisation du <i>dithering</i> pour augmenter la résolution ..... 16
Figure 2.4	Banc de test de l' <i>ADC</i> 2015 ..... 17
Figure 2.5	Résultats expérimentaux de l' <i>ADC</i> 2015 en mode natif ..... 18
Figure 4.1	Comparaison des fonctions de transfert de différents modes de fonctionnement envisagés pour l' <i>ADC</i> ..... 22
Figure 4.2	Mise en évidence de l'impossibilité de dépasser 13 bits de résolution en divisant la séquence de condensateurs ..... 23
Figure 4.3	Comparatif entre les codes de sortie d'un <i>ADC</i> 13 bits théorique et de la solution proposée ..... 25
Figure 4.4	Architecture générale de la nouvelle version de l' <i>ADC</i> ..... 26
Figure 4.5	Représentation de la répartition physique des sous-ensembles de la puce ..... 27
Figure 4.6	Représentation de la distribution des signaux sur les <i>DAC</i> via les <i>level-shifters</i> ..... 28
Figure 5.1	Schéma électrique du <i>sample-and-hold</i> ..... 32

Figure 5.2	Vue schématique du <i>sample-and-hold</i> .....	32
Figure 5.3	Schéma bloc du générateur de signaux commandant les <i>sample-and-hold</i> .....	33
Figure 5.4	Chronologie des commutations des interrupteurs des <i>DAC</i> entre la fin de l'échantillonnage et le début de l'algorithme de recherche <i>SAR</i> .....	34
Figure 5.5	Structures des capacités <i>MIM</i> et <i>MOM</i> .....	35
Figure 5.6	Vue schématique du comparateur .....	37
Figure 5.7	Schéma des <i>level-shifters</i> .....	38
Figure 5.8	Représentation de la machine à états utilisée .....	40
Figure 5.9	Diagramme <i>RTL</i> du registre à approximations successives ( <i>SAR</i> ) .....	42
Figure 5.10	Diagramme <i>RTL</i> de l'étage de sortie du cœur logique .....	43
Figure 5.11	Vue schématique des sous-circuits de l'amplificateur opérationnel ( <i>AOP</i> ) .....	45
Figure 5.12	Schématisme de la caractéristique du circuit de polarisation .....	46
Figure 5.13	Vue schématique du point de mesure numérique .....	48
Figure 5.14	Vue schématique de la distribution de l'alimentation .....	49
Figure 5.15	Grandes étapes de conception de la puce .....	50
Figure 5.16	Dessin des masques de la puce sans les plots d'entrée/sortie .....	53
Figure 5.17	Dessin des masques de la partie analogique .....	54
Figure 6.1	Simulation temporelle réalisée sur le comparateur .....	56
Figure 6.2	Chronogramme pré-synthèse du cœur logique de la puce .....	57
Figure 6.3	Répartition de la consommation d'énergie dans le bloc numérique .....	59
Figure 6.4	Chronogramme typique d'une conversion (données : simulateur <i>AMS</i> ) .....	63
Figure 6.5	Résultats de simulation en mode natif avec $f_{CLK} = 100kHz$ et $f_{IN} = 200Hz$ .....	65

Figure 6.6	Résultats de simulation en mode natif avec $f_{CLK} = 1MHz$ et $f_{IN} = 1kHz$ .....	66
Figure 6.7	Portion de la fonction de transfert de l'ADC en résolution native et en mode <i>dithering</i> (données : simulateur AMS) .....	68
Figure 6.8	Résultats de simulation en mode <i>dithering</i> avec $f_{CLK} = 1MHz$ et $f_{IN} = 50Hz$ .....	69
Figure 7.1	Vue au microscope d'un <i>die</i> .....	71
Figure 7.2	Premier <i>PCB</i> de test de la nouvelle puce .....	72
Figure 7.3	Deuxième <i>PCB</i> de test de la nouvelle puce .....	73
Figure 7.4	Résultats expérimentaux en mode natif avec $f_{CLK} = 1MHz$ et $f_{IN} = 10Hz$ .....	74
Figure 7.5	Évolution des tensions générées par les <i>DAC</i> lors une conversion .....	75
Figure 8.1	Étapes des techniques $SAR^2$ et <i>dithering</i> .....	79
Figure 8.2	Évolutions du facteur de modification de la <i>FoM</i> en fonction de la résolution .....	80
Figure 8.3	Organisation possible d'une puce testant les performances du <i>dithering</i> .....	81



## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ADC	Analog to Digital Converter - <i>Convertisseur Analogique Numérique</i>
AOP	Amplificateur OPérationnel
AMS	Analog Mixed-Signal
BNC	Bayonet Neill–Concelman connector
CMOS	Complementary Metal Oxide Semiconductor
DAC	Digital to Analog Converter - <i>Convertisseur Numérique Analogique</i>
DNL	Differential NonLinearity - <i>Non Linéarité Différentielle</i>
ENOB	Effective Number Of Bits - <i>Nombre de Bits Effectifs</i>
ESD	ElectroStatic discharge - <i>Décharge électrostatique</i>
FoM	Figure of Merit - <i>Figure de Mérite</i>
FSM	Finite State Machine - <i>Automate à nombre fini d'états</i>
GND	Ground - <i>Masse du circuit</i>
IBM	International Business Machine
INL	Integral NonLinearity - <i>Non Linéarité Intégrale</i>
LSB	Least Significant Bit - <i>Bit de poids faible</i>
MEMS	MicroElectroMechanical Systems - <i>Microsystème électromécanique</i>
MIM	Metal Insulator Metal - <i>Capacité de type MIM</i>
MOM	Metal Oxide Metal - <i>Capacité de type MOM</i>
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor - <i>Transistor MOS à effet de champ</i>
MSB	Most Significant Bit - <i>Bit de poids fort</i>
MSV	Multiple Supply Voltages - <i>Alimentation multiple</i>
MUX	Multiplexer - <i>Multiplumeur</i>
NAND	No AND logic gate - <i>Porte logique Non Et</i>

NMOS	n-channel MOSFET - <i>Transistor MOS de type N</i>
PCB	Printed Circuit Board - <i>Circuit imprimé</i>
PISO	Parallel In - Serial Out - <i>Registre à entrée parallèle et sortie série</i>
PMOS	p-channel MOSFET - <i>Transistor MOS de type P</i>
QFN	Quad Flat No-leads package - <i>Boitier carré pour circuit intégré</i>
RFMR	Request for Manufacturing Resources Form
RTL	Register Transfer Level
SAR	Successive Approximation Register - <i>Registre à Approximations Successives</i>
S/H	Sample-and-Hold - <i>Échantillonneur / Bloqueur</i>
SINAD	Signal-to-Noise And Distortion ratio - <i>Ratio du signal sur le bruit plus la distorsion</i>
SNR	Signal Noise Ratio - <i>Rapport Signal sur Bruit</i>
VHDL	VHSIC Hardware Description Language - <i>Langage de description haut niveau</i>
VHSIC	Very High Speed Integrated Circuit - <i>Circuit intégré très haute vitesse</i>

## LISTE DES SYMBOLES ET UNITÉS DE MESURE

---

### UNITÉS GÉOMÉTRIQUES

#### Longueur

mm millimètre

$\mu m$  micromètre

nm nanomètre

#### Surface

$mm^2$  millimètre carré

---

### UNITÉS ÉLECTRIQUES

$\mu A$  microampère

$\mu V$  microvolt

mV millivolt

V volt

nW nanowatt

$\mu W$  microwatt

---

### UNITÉS DE FRÉQUENCE

kHz kilohertz

MHz megahertz

---

### UNITÉS DES COMPOSANTS ÉLECTRIQUES

pF picofarad

fF femtofarad

$k\Omega$  kiloohm

---

### AUTRES UNITÉS

dB décibel

$^{\circ}$  degré

K Kelvin

$\mu s$  microseconde





## INTRODUCTION

### 0.1 Contexte des convertisseurs analogique-numérique

Les convertisseurs analogique-numérique (abrégé *CAN* ou plus couramment *ADC* en anglais) sont une constituante essentielle des systèmes analogique-numérique. Ils sont utilisés dans tout système nécessitant d'interfacer des éléments analogiques et numériques entre eux. Un exemple classique est celui des capteurs. Ceux-ci produisent un signal analogique qui doit être converti pour être utilisé par un système numérique. Ils sont très présents dans le monde qui nous entoure et fournissent aux systèmes électroniques des données sur leur environnement comme la température ou la pression par exemple. Les applications sont très variées : automobile, aérospatial, téléphones intelligents...

Une grande diversité de capteurs implique une large gamme de convertisseurs pour les interfacer. Il existe ainsi de multiples types d'*ADC* en terme de résolution (nombre de bits sur lequel est codé le résultat numérique), de taux de conversion (fréquence maximale à laquelle il est en mesure d'actualiser le résultat) ou encore de consommation d'énergie. Les *ADC* à taux de conversion élevé sont traditionnellement utilisés pour des applications de traitement vidéo ou audio alors qu'on préférera des *ADC* plus lents mais moins gourmands en énergie pour des applications utilisées à l'échelle humaine. C'est ce deuxième type d'*ADC* qui nous intéresse ici avec la conception d'un module pouvant interfacer des capteurs *MEMS* (*i.e.* MicroElectro-Mechanical Systems) sur des applications embarquées (*e.g.* smartphones, objets connectés) où l'économie d'énergie est le facteur le plus important mais où la vitesse de conversion n'est pas nécessairement une priorité.

Le synopsis du projet est détaillé en figure 0.1 (page 2). Le projet est divisé en cinq grandes parties que nous allons étudier dans ce mémoire. Après une partie consacrée à l'état de l'art des *SAR-ADC* et à l'étude du projet antérieur (*cf.* chapitre 1 et 2 en page 5 et 13), nous détaillerons

les aspects de conception et de simulations du système correspondant aux étapes 2 et 3 du synopsis (cf. chapitre 3, 4, 5 et 6 en page 19, 21, 31 et 55). Nous poursuivrons par la présentation des deux dernières étapes du synopsis qui constituent la préparation et la réalisation des tests effectués sur la puce en laboratoire (cf. chapitre 7, page 71). L'ultime chapitre de ce rapport sera consacré aux perspectives d'évolution de cette étude (cf. chapitre 8, page 77).

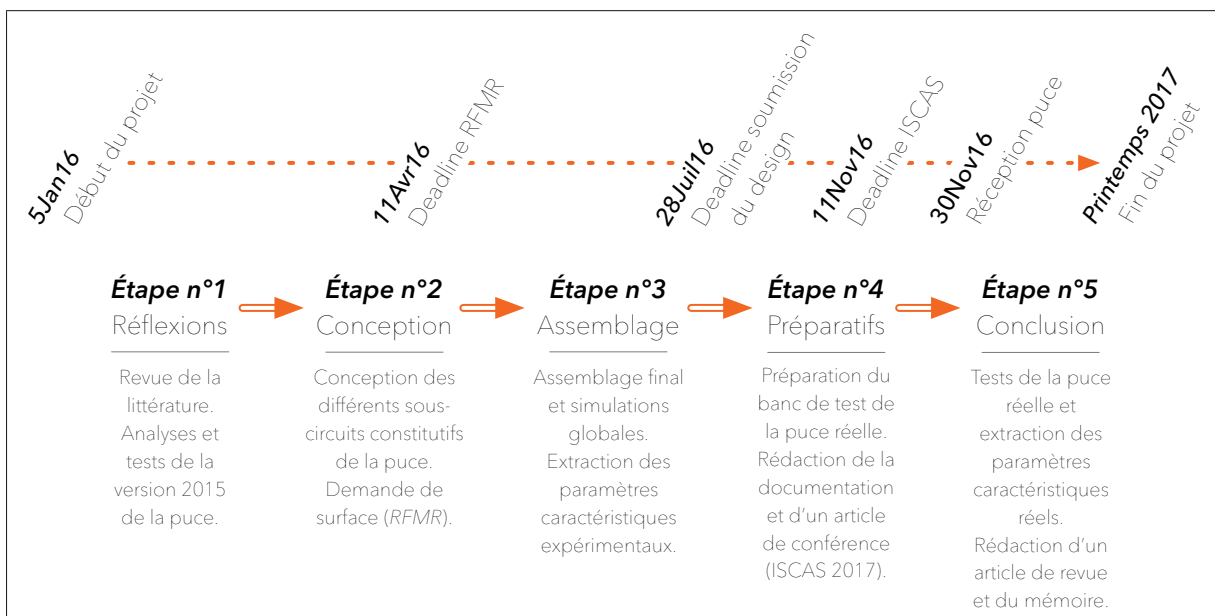


Figure 0.1 Synopsis des grandes étapes du projet

## 0.2 Contributions au projet

On peut dresser une liste (non exhaustive) des contributions majeures de ce projet dans le cadre de l'utilisation du *dithering* sur les *SAR-ADC* (notamment vis-à-vis de la version 2015 de l'*ADC*). On retiendra que :

- la consommation théorique d'énergie a été réduite de 75% par rapport à la puce précédente ;
- une architecture à alimentation multiple (dite *MSV*) a été utilisée pour réduire la consommation d'énergie ;

- des modèles théoriques ont été élaborés et ont permis de mieux comprendre l'impact des condensateurs de biais sur la résolution en mode *dithering* ;
- des simulations informatiques ont permis d'évaluer la valeur de l'*ENOB* en mode *dithering* à 11.4 bits (soit 15% de mieux qu'en mode natif) ;
- une architecture de tests analogique/numérique a été mise en place pour évaluer le fonctionnement interne de l'*ADC* ;
- un article, orienté sur le design général et les premiers résultats obtenus en simulation, a été accepté pour la conférence IEEE ISCAS 2017 ;
- une *IEEE Electronic Letter*, orientée sur les perspectives d'évolution du concept et apportant des résultats complémentaires à ceux proposés dans l'article ISCAS, a été soumise.



## CHAPITRE 1

### ÉTAT DE L'ART DES SAR ADC

#### 1.1 Principe de fonctionnement des SAR ADC

Avant de voir en détails la conception des SAR et les techniques couramment utilisées dans la littérature, il est important bien comprendre le fonctionnement de l'ADC afin de bien déterminer la mission et les spécificités de chacun des blocs le constituant. L'architecture conventionnelle d'un SAR est représentée ci-après (cf. Figure 1.1, page 5) et est la même que celle décrite dans (Maxim Integrated, 2001). On note que ce type d'ADC utilise un convertisseur numérique-analogique (CNA, ou DAC en anglais). Cet élément effectue l'opération inverse de l'ADC dans le sens où il convertit le signal numérique en une tension analogique. L'objectif de ce type d'ADC est de faire converger la tension produite par le DAC vers la tension à mesurer (qui aura, au préalable, été mémorisée grâce au *sample-and-hold* (abrégié S/H)).

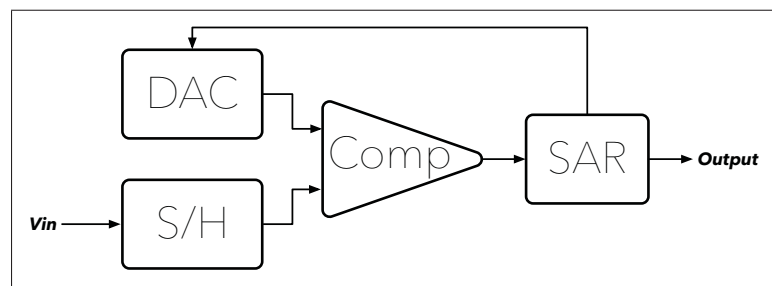


Figure 1.1 Architecture d'un ADC SAR standard

On peut distinguer deux grandes étapes dans le fonctionnement de l'ADC. Une étape d'acquisition où on travaille avec le *sample-and-hold* et une étape d'évaluation (*i.e.* l'étape lors de laquelle les approximations successives s'effectuent) où l'on travaille cette fois-ci avec l'ensemble DAC, comparateur et SAR.

Pour plus de clarté, considérons l'exemple suivant (cf. figure 1.2, page 6). Il s'agit du chronogramme d'un cycle de conversion sur un SAR ADC de 5 bits.

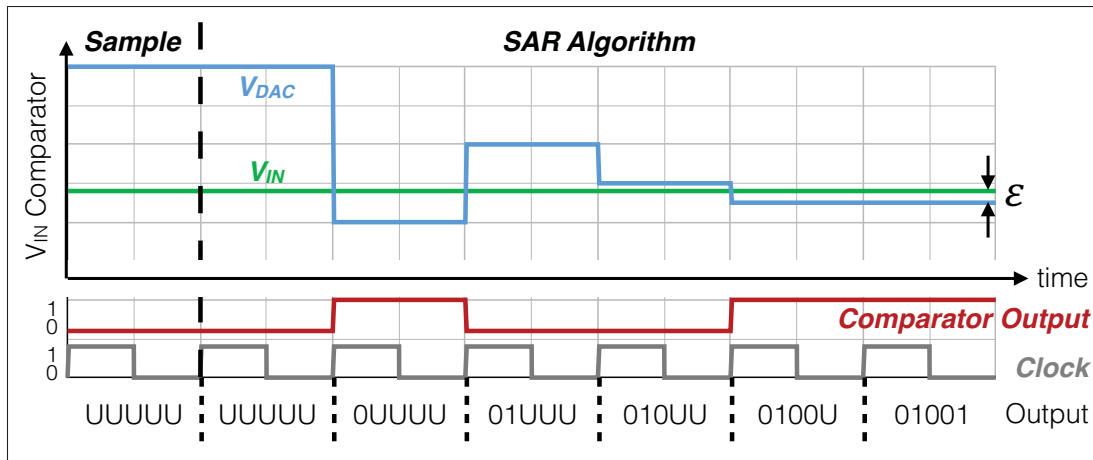


Figure 1.2 Chronogramme théorique du déroulement d'un algorithme SAR sur 5 bits

Le processus commence par l'échantillonnage (étape *Sample* sur la figure 1.2, page 6). Pendant cette période le *sample-and-hold* enregistre la valeur de  $V_{IN}$  pour ensuite la maintenir constante pendant le reste du cycle. Ainsi pendant tout le processus de convergence (étape *SAR Algorithm* sur la figure 1.2, page 6) le *sample-and-hold* maintient la tension  $V_{IN}$  qu'il a enregistré à l'entrée du comparateur.

À chaque cycle d'horloge le SAR commande un nouveau réglage au DAC en fonction des informations envoyées par le comparateur. Le premier cycle est celui de l'échantillonnage donc tous les bits de sortie sont indéfinis (cf. *UUUUU* sur la figure 1.2, page 6). Au second cycle, on commence la recherche par approximations successives. Le DAC est alors chargé à la moitié de la tension qu'il peut produire. Au troisième coup d'horloge, le SAR détermine l'action à effectuer en fonction de la valeur du comparateur lors du deuxième cycle. Dans notre cas, le comparateur donne zéro (niveau logique bas), il faut donc diminuer la tension du DAC. On place alors un zéro sur le premier bit du résultat (le *MSB*), synonyme que la première tension imposée au DAC (lors du cycle  $n^o2$ ) était plus grande que  $V_{IN}$ . Au quatrième cycle on positionne un 1 sur le deuxième bit du résultat et on augmente la tension du DAC en réponse au résultat du comparateur lors du cycle 3. À chaque nouveau cycle d'horloge l'amplitude de la transition sur la tension du DAC est divisée par deux. Il en va ainsi jusqu'à ce que tout les bits aient été complétés. On aboutit à la fin au résultat le plus proche de  $V_{IN}$  possible avec la résolution

dont on dispose. Pour avoir un résultat plus précis (*i.e.* réduire la valeur d' $\epsilon$ ) il faudrait une architecture d'une résolution supérieure (*i.e.* plus de 5 bits).

La majorité des *DAC* sont constitués d'un peigne de capacités (*capacitor array* en anglais). Une des deux bornes des capacités est connectée à la fois à la sortie du *S/H* et à l'entrée du comparateur (*cf.* figure 1.3, page 7). L'autre borne peut quant à elle être connectée à  $V_{DD}$ ,  $\frac{V_{DD}}{2}$  ou à la masse en fonction des signaux de commande.

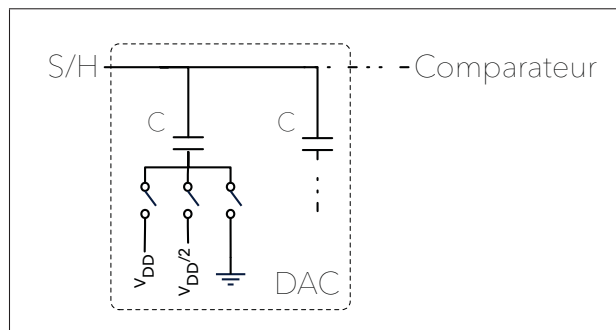


Figure 1.3 Schéma des connexions entre *S/H*, *DAC* et comparateur

Une architecture possible de l'*ADC* avec un tel *DAC* est représentée en figure 1.4 (page 8). Cet *ADC* fonctionne de la même façon que celui présenté précédemment, à la différence que la conversion se fait par rapport à une tension de référence (noté  $V_{ref}$  sur le schéma). Lors de l'échantillonnage le *sample-and-hold* charge le *DAC* à la tension  $V_{IN}$  puis se désactive. Dès lors, l'algorithme *SAR* se met en place. En chargeant et en déchargeant les capacités, on fait converger la tension de sortie du *DAC* de  $V_{IN}$  vers  $V_{ref}$ . On convertit ainsi l'écart relatif entre ces deux tensions.

Nous allons à présent aborder les idées novatrices et concepts intéressants présents dans la littérature scientifique.

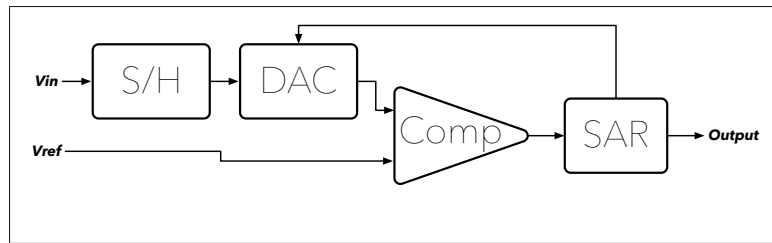


Figure 1.4 Architecture d'un ADC SAR avec DAC capacitif

## 1.2 Revue de la littérature

L'intérêt de disposer d'un convertisseur analogique-numérique reconfigurable est réel. On peut par exemple imaginer un système composé d'un cœur logique et de plusieurs capteurs, fonctionnant à des résolutions différentes, à interfacer. Si les fréquences d'échantillonnage ne sont pas trop élevées, il peut être intéressant d'utiliser un seul et unique ADC qui va convertir successivement chacune des tensions fournies par les capteurs. À chaque fois qu'on change de capteur, l'ADC est reconfiguré de telle sorte que le compromis entre fréquence d'utilisation et consommation d'énergie soit optimal pour cette utilisation.

Une solution d'ADC reconfigurable a été proposée dans (Gulati & Lee, 2001). Ce convertisseur utilise un système complexe pour faire varier la résolution de 6 à 16 bits et est basé sur une architecture de type delta-sigma. Ce projet est intéressant mais s'accorde mal avec l'objectif d'un convertisseur basse-fréquence et économe en énergie. En effet, on parle ici d'un système fonctionnant avec une horloge cadencée à plusieurs mégahertz de fréquence (2.62MHz et 10MHz en fonction du mode de fonctionnement utilisé) et consommant plusieurs dizaines de milliwatts de puissance (24.6mW et 17.7mW en fonction du mode). Or, dans cette étude nous visons une fréquence d'horloge de l'ordre du mégahertz pour une consommation de l'ordre du microwatt.

C'est dans cette optique de réduction de la consommation d'énergie, que nous nous sommes concentré sur les ADC de type SAR. En effet, ceux-ci montrent des résultats très intéressants en terme de consommation d'énergie comme dans (Chang, Wang, & Wang, 2007) et (Hong & Lee, 2007). Dans (Chang *et al.*, 2007), une méthode spécifique de commutation des condensateurs des DAC a permis d'obtenir un ADC 8 bits consommant uniquement 7.75μW.



Dans (Hong & Lee, 2007), les auteurs ont réussi à atteindre une consommation d'énergie de  $2.47\mu W$  sur un *ADC* 8 bits non différentiel.

L'utilisation d'une architecture *SAR* semble donc pertinente dans notre situation. La littérature scientifique consacrée aux *ADC* de type *SAR* est riche et on y trouve diverses techniques et sous-circuits différents. Nous allons ici nous intéresser à des travaux antérieurs effectués sur des systèmes fonctionnant dans des conditions similaires à celles que nous ciblons en terme de consommation, de fréquence ou encore de résolution. L'ensemble de ces travaux exploitent, sous diverses formes, l'algorithme *SAR* décrit précédemment (*cf.* section 1.1, page 5).

La majorité des *SAR* utilisent un *DAC* dit captatif. C'est le cas de l'architecture présentée dans (Zhang, Mawelo, Fayomi, & Nabki, 2013). Il s'agit d'un peigne de capacités mises en parallèle. Ces capacités sont chargées ou déchargées au fil de la recherche algorithmique ce qui a bien entendu pour effet de faire varier la tension aux bornes de celles-ci. On transforme donc un signal discret (donc numérique) correspondant au nombre de condensateurs qui sont chargés ou déchargés en signal continu (donc analogique) correspondant à la tension de sortie ainsi développée. Cet article propose une architecture non-différentielle (identique au modèle décrit en section 1.1, page 5) qui a la particularité d'être à résolution variable variant ici de 7 à 10 bits. Ce type d'architecture est intéressant dans le sens où l'*ADC* proposé est adaptable à une large gamme de capteurs. Tel que proposé, ce concept se résume à un *ADC* 10 bits que l'on va venir brider si nécessaire afin d'en réduire la consommation. On pourrait également envisager sur un tel système de réduire la résolution en vue d'accélérer la fréquence de travail (il faut moins de temps au *SAR* pour obtenir un résultat de résolution inférieure). En terme de performances, cet *ADC* affiche une consommation de  $0.4\mu W$  pour une fréquence d'échantillonnage de  $50kS \cdot s^{-1}$  lorsqu'il fonctionne en mode 10 bits.

D'autres architectures sont possibles, notamment au niveau du *DAC*. C'est le cas du système proposé dans (Przyborowski & Idzik, 2010) qui utilise un réseau de miroirs de courant en lieu et place du peigne de capacités. Sur ce *DAC*, le signal numérique commande les miroirs de courant. Chacun d'entre eux génère, s'il est activé, un courant qui est proportionnel à ses di-

mensions. À l'interconnexion des miroirs de courant existe alors un courant représentatif du code numérique présenté à l'entrée du convertisseur. Ce courant est ensuite traduit en tension via une résistance. Ce type d'architecture présente l'avantage d'être extrêmement compact et bien plus rapide par rapport à une structure capacitive plus conventionnelle. L'inconvénient majeur est la consommation d'énergie qui est bien plus importante que sur un *DAC* capacitif. Sur la même technologie que la notre (130nm), la consommation d'un *DAC* 10 bits sur ce type d'architecture est de l'ordre de  $500\mu W$  pour une fréquence d'échantillonnage de  $2MS \cdot s^{-1}$ . Compte tenu de ces données on peut s'attendre à une consommation s'établissant autour de  $12.5\mu W$  si la fréquence était de  $50kS \cdot s^{-1}$  soit un facteur supérieur à 10 par rapport au travail précédent alors qu'on ne mesure ici que la consommation du *DAC* et non celle de l'*ADC* au complet. Un autre désavantage est la consommation statique de ce système. S'agissant d'un système actif, celui-ci est alimenté et consomme en permanence que l'*ADC* soit sollicité pour une conversion ou non. Bien qu'intéressant en terme de surface occupée, ce type d'architecture ne correspond pas à nos attentes en terme de consommation. Ce type de système semble plus adapté à des environnements embarqués où l'énergie est moins problématique mais où la vitesse et la compacité du système peuvent être des facteurs importants (application embarqué type automobile ou aérospatial par exemple).

Outre les *ADC* à entrée commune, on peut également trouver des *ADC* à entrée dite différentielle. C'est le cas du système proposé dans (Taherzadeh-Sani, Lotfi, & Nabki, 2014). L'architecture proposée est de type *fully-differential* selon la définition proposée dans (Maxim Integrated, 2002). Ce type d'architecture est destinée à convertir la différence entre deux signaux en opposition de phase (le second signal étant la copie conforme du premier mais déphasé de  $180^\circ$ ). Comme on le verra par la suite (*cf.* section 2.1, page 13) ce type de structure permet de limiter le bruit mais impose de doubler certains éléments du design ce qui implique nécessairement une consommation de courant plus importante.

Les *DAC* implantés dans cet *ADC* utilisent une capacité en série, à l'image de la figure 1.5 (page 11), afin de réduire la taille des condensateurs utilisés. Ceci a pour effet de réduire la capacité totale du *DAC*. On le constate facilement sur le tableau 1.1 (page 11) où la plus grosse capacité

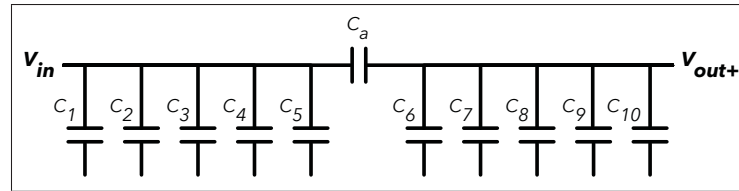


Figure 1.5 Architecture d'un DAC à capacité série (ou *split capacitor*)

est égale à 32 fois la capacité unitaire contre 128 fois avec la méthode traditionnelle. Ceci a pour principal avantage de réduire la surface utilisée ainsi que la consommation d'énergie. Un autre avantage sous-jacent est l'amélioration de la fréquence maximale à laquelle on peut utiliser le système des capacités plus petites impliquant des temps de charge et de décharge moins longs. Ici la capacité série est placée entre  $C_5$  et  $C_6$ .

Tableau 1.1 Peignes de condensateurs des DAC traditionnel ou avec capacité série  
Tiré de Taherzadeh-Sani *et al.* (2014)

Capacitors	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$	$C_7$	$C_8$	$C_9$	$C_{10}$	$C_a$
Without $C_a$	$\frac{C_u}{2}$	$\frac{C_u}{2}$	$C_u$	$2C_u$	$4C_u$	$8C_u$	$16C_u$	$32C_u$	$64C_u$	$128C_u$	N/A
With $C_a$	$\frac{C_u}{2}$	$\frac{C_u}{2}$	$C_u$	$2C_u$	$4C_u$	$2C_u$	$4C_u$	$8C_u$	$16C_u$	$32C_u$	$8C_u$

Le principal inconvénient de cette architecture dite à *split-capacitor* et qu'elle impose l'utilisation d'un système de calibration pour corriger les défauts apportés par la capacité série (*i.e.*  $C_a$  dans l'article). En effet, pour générer des tensions égales à celles produites par la séquence standard il faudrait utiliser une valeur de  $C_a$  qui serait une fraction précise de la capacité unitaire  $C_u$ . En réalité, il est impossible d'obtenir une telle capacité avec suffisamment de précision du fait des variations dans les procédés de fabrication. L'utilisation du circuit de calibration permet alors de corriger la tension de sortie pour la rapprocher au maximum de celle obtenue avec un DAC traditionnel. Le système de calibration est un élément complexe mais cette technique devient rapidement indispensable dès lors que l'on souhaite atteindre des résolutions supérieures à 10 bits en conservant une consommation de courant et une surface utilisée rai-

sonnable. En technologie 180nm et avec une tension d'alimentation de 1.5V la consommation est de  $1.16\mu W$  pour une résolution de 10 bits et une fréquence de conversion de  $110kS \cdot s^{-1}$ . Si on applique une règle en  $x^2$  (*i.e.* ici  $x = \frac{130}{180} = 0.72$ ), ceci correspondrait à une consommation de  $0.6\mu W$  en technologie 130nm.

Chacun de ces articles présente des solutions diverses plus ou moins adaptées à notre situation. Ces solutions constituent une base de réflexion pour notre étude. Celle-ci se base également sur un projet antérieur que nous allons décrire à présent.

## CHAPITRE 2

### TRAVAUX ANTÉRIEURS

Ce projet s'inscrit dans la continuité d'un projet de PFE effectué par un élève français de l'école CPE Lyon à l'UQAM pendant l'été 2015. Son objectif était de concevoir un *SAR ADC* reconfigurable d'une résolution allant de 10 à 12 bits, basses fréquences et à faible consommation de courant. Son rapport de PFE (*cf.* (Favre, 2015)) m'a permis d'apprivoiser plus rapidement ce projet.

#### 2.1 Présentation succincte

L'architecture proposée dans ce précédent projet est différentielle (*cf.* figure 2.1, page 14) et convertit donc la différence de tension entre ses deux entrées. Cette architecture impose de doubler certains éléments comme le *sample-and-hold* (notés *S/H<sub>p</sub>* et *S/H<sub>n</sub>* sur la figure 2.1) et les convertisseurs numérique/analogique (notés *DAC<sub>p</sub>* et *DAC<sub>n</sub>* sur la figure 2.1). Il en résulte une amélioration de la précision (on s'affranchit des bruits induits par les branches *S/H* et *DAC*) au prix d'une augmentation de surface sur silicium occupée et d'une consommation de courant accrue. Les signaux analogiques issus des deux *DAC* sont évalués par le comparateur. Ce comparateur est *latché* (verrouillé) et effectue la comparaison des deux signaux sur les fronts descendants du signal d'horloge. Le signal ainsi produit est acheminé à l'entrée du *SAR* qui peut dès lors ré-évaluer les commandes numériques à appliquer au *DAC* suivant l'algorithme d'approximations successives décrit précédemment (*cf.* partie 1.1, page 5).

Cette puce a deux particularités majeures par rapport à un *SAR* différentiel traditionnel sur lesquelles il convient de s'arrêter. Tout d'abord, ce circuit présente des *DAC* d'architecture hybride. En effet, ceux-ci sont scindés en une partie binaire et une partie unaire (*cf.* figure 2.2, page 15). La partie binaire correspond à l'architecture classique qui se résume en une série de capacités en parallèle dont les poids sont ordonnés de manière binaire. Le *DAC* unaire est quant à lui composé de capacités identiques montées en parallèle. Les 5 *LSBs* sont attribués au *DAC* binaire alors que les 5 *MSBs* sont attribués au *DAC* unaire. Cette architecture permet d'éliminer

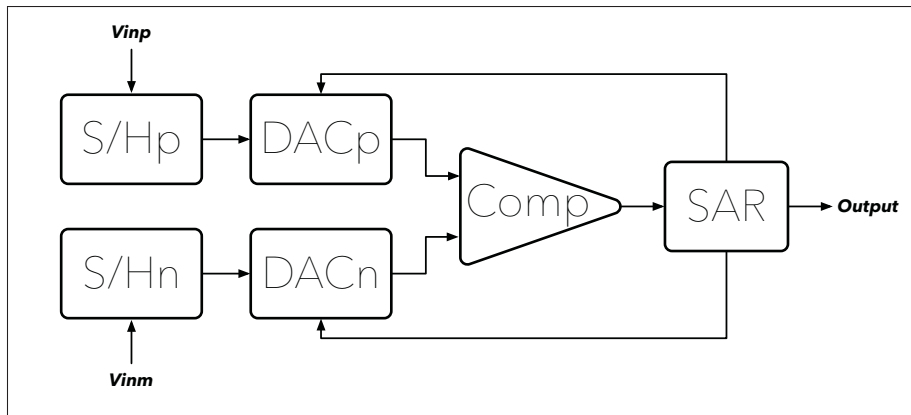


Figure 2.1 Architecture de la version 2015 de l'ADC

la distorsion de croisement qui peut s'avérer critique sur certains gros DAC. En effet, sur une architecture binaire, les capacités sont chargées et déchargées de manière simultanée ce qui peut entraîner des problèmes. Prenons l'exemple suivant : on est dans la configuration où les capacités  $C$  et  $2C$  sont chargées et on souhaite augmenter la capacité totale de  $3C$  à  $4C$ . On va alors décharger  $C$  et  $2C$  pour charger  $4C$ . Supposons maintenant que  $C$  et  $2C$  soient un peu plus grosses que leurs valeurs théoriques et que  $4C$  soit un peu plus petite (du fait des fluctuations possibles de la valeur des capacités lors du procédé de fabrication). La différence de capacité entre les deux configurations est alors loin de la valeur de  $1C$  souhaitée et on peut même se retrouver dans la configuration extrême où la nouvelle capacité (*i.e*  $4C$ ) est plus petite que la précédente (*i.e*  $C+2C$ ).

En utilisant un DAC unaire, on s'affranchit de ce problème puisque on effectue les charges et décharges de façons dissociées. Cette architecture présente néanmoins un inconvénient en terme de complexité. En effet, l'utilisation d'un DAC unaire impose l'implémentation de convertisseur binaire-thermomètre afin de convertir les commandes binaires en provenance du SAR en signaux unaires à destination des multiplexeurs commandant les interrupteurs de chacun des condensateurs.

La deuxième innovation de cette puce est l'implémentation d'un système de *dithering* visant à augmenter la résolution de l'ADC de 10 à 12 bits. Le principe est de réaliser plusieurs me-

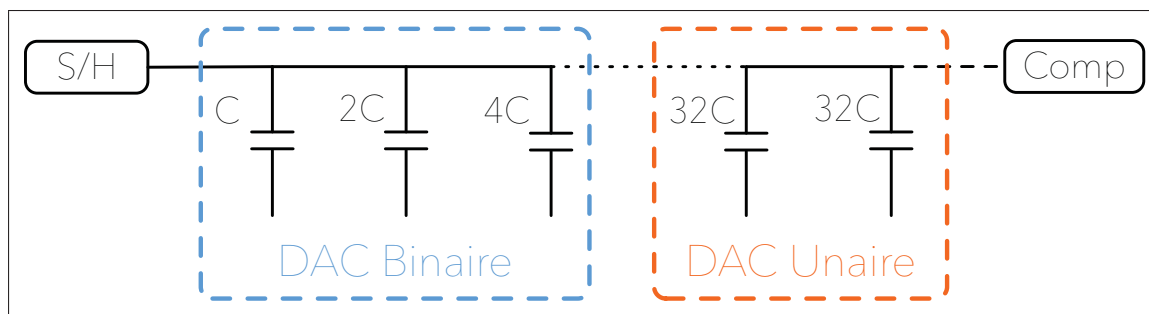


Figure 2.2 Architecture interne des DAC hybrides

sure de la même tension d'entrée puis de les moyenner afin d'obtenir un résultat de résolution supérieur. Cependant, il est inutile de se contenter de mesurer successivement une même tension, celle-ci donnant invariablement le même code de sortie. L'idée est d'ajouter un bruit sur le signal pendant l'échantillonnage. Ce bruit est modélisé grâce aux petites capacités du DAC qui seront connectées, pendant l'échantillonnage, à  $V_{rep}$  ou  $V_{refm}$  au lieu de  $V_{cm}$ . En utilisant les trois premiers condensateurs, on obtient 14 combinaisons possibles, également réparties de part et d'autre de la valeur initiale. On a donc 15 résultats distincts à moyenner pour obtenir une résolution supérieure.

Le *dithering* permet de s'affranchir du bruit de quantification du DAC en évaluant si la perturbation ajoutée lors de l'échantillonnage permet de passer au code supérieur ou inférieur (cf. figure 2.3, page 16). Cette figure explicite le principe de base du *dithering* sur un ADC non-différentiel de 5 bits. Une conversion native (en haut de la figure 2.3) donne ici le résultat "01001". Dans une deuxième conversion (en bas de la figure 2.3), l'échantillonnage est biaisé d'une tension égale à  $\frac{V_{LSB}}{2}$  et le résultat devient alors "01010". Ceci nous permet d'affirmer que la tension d'entrée se trouve dans la portion haute de l'espace entre "01001" et "01010" (voir les bulles en figure 2.3). Ainsi le résultat amélioré est "010011"

Ce convertisseur reprend donc ce principe en l'appliquant au DAC positif et en utilisant 14 tensions de biais différentes. La moyenne des 15 échantillons a ainsi pour but d'évaluer à quel point la tension mesurée est proche du code initial, cela revient à trouver un nouveau palier intermédiaire correspondant à la tension d'entrée, donc à augmenter la résolution de l'ADC.

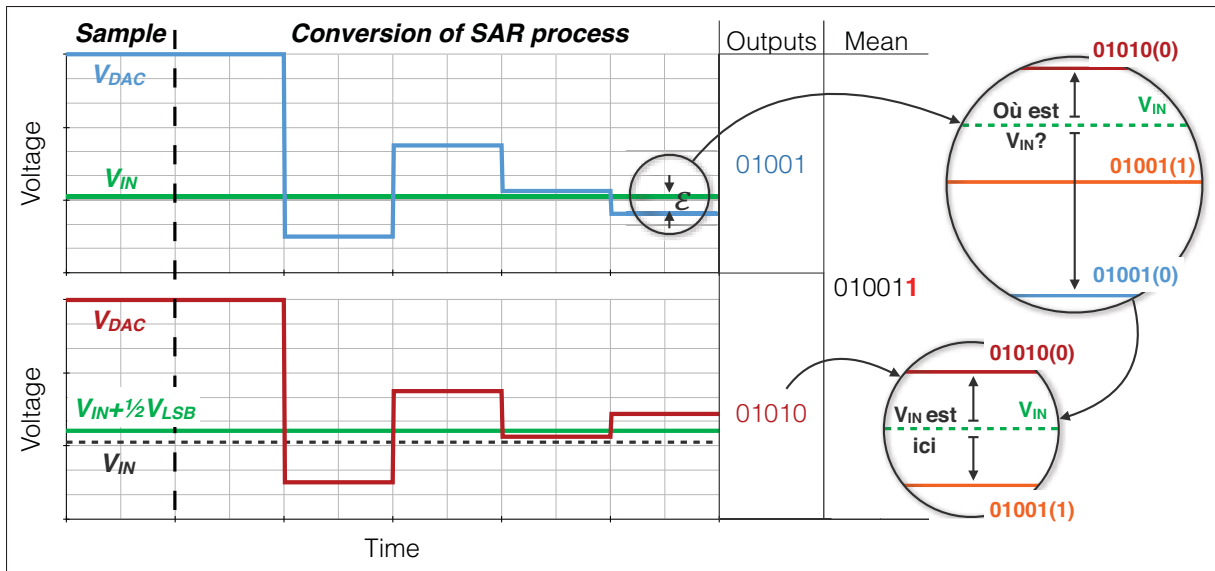


Figure 2.3 Exemple d'utilisation du *dithering* pour augmenter la résolution

Cette augmentation de la résolution a un prix puisqu'il faut alors 15 cycles de conversion au lieu d'un pour déterminer la valeur du code de sortie. Bien entendu, les deux conséquences majeures sont alors une multiplication par 15 de la consommation de courant pour une conversion et une division par 15 de la fréquence maximale de conversion.

La totalité du travail effectué sur cette puce l'a été de façon non automatisée. Ainsi toutes les parties logiques (*i.e.* le SAR, les multiplexeurs des DAC et les convertisseurs binaire/thermomètre) ont été dessinées à la main à partir de l'outil de *layout* de *Cadence Virtuoso*. L'architecture *full-custom* ainsi créée a l'inconvénient d'être synonyme de temps de simulations extrêmement longs ce qui a empêché l'obtention des paramètres caractéristiques de l'ADC (*i.e.* *ENOB*, *DNL*, *INL*, *FoM*). Le fonctionnement de l'ADC était validé sur une entrée fixe mais celui n'a pas pu être simulé sur une sinusoïde complète comme il est d'usage. Passons à présent à une brève analyse des résultats expérimentaux de cette puce.



## 2.2 Expérimentation

Les délais de fabrication étant relativement longs, dans notre cas environ quatre mois, l'étudiant qui a conçu cette première puce n'a pas eu la possibilité de la tester lui-même. J'ai donc eu la responsabilité de tester cette puce moi-même en me basant sur son rapport de stage. Pour cela, j'ai conçu deux *PCBs* (cf. figure 2.4, page 17) : le *PCB* de test à proprement parler et un *PCB* de support visant à recevoir le boîtier *QFN* (*Quad Flat No-leads package*) contenant le circuit.

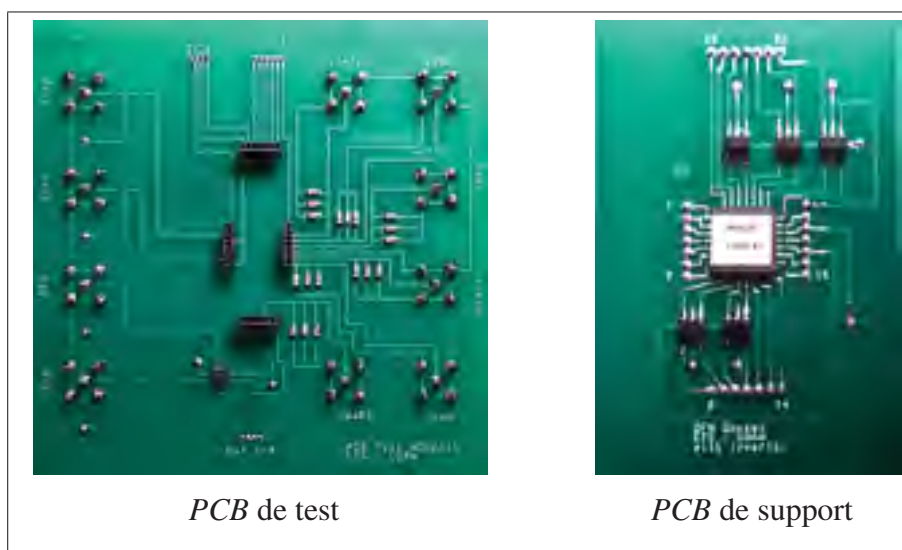


Figure 2.4 Banc de test de l'ADC 2015

Ceci permet une plus grande flexibilité lors des tests dans la mesure où l'on peut tester facilement une série de puce. Les deux *PCBs* sont interconnectés grâce à des *headers*. La puce ne possédant pas de circuit d'inverseurs sur les entrées et sorties, j'ai dû placer ceux-ci à l'extérieur de la puce sur le *PCB*. Ces derniers permettent de reformer les signaux d'entrée sortie de la puce et dans le cas des sorties contribuent à *driver* les capacités des pins et autres connecteurs. Outre les prises *BNC*, on trouve sur le *PCB* les capacités de découplage placées sur chacune des alimentations.

Les tests ont été réalisés au laboratoire de l'UQAM sur un oscilloscope mixte (numérique/analogue). Malheureusement, ceux-ci ont révélés que la puce n'était pas fonctionnelle. En effet,

un phénomène de saturation est présent sur la sortie, et ce, même pour des tensions d'entrée très faibles (*cf.* figure 2.5, page 18). Le signal de sortie est ainsi noyé dans le bruit comme en témoigne la valeur du SNR (*i.e.* le rapport signal sur bruit) du signal de sortie qui s'établit à 7.82dB. Sur le spectre, on constate bien que l'amplitude (entre 15dB et 20dB) et la fréquence du bruit sont très importantes. Dans ces conditions, il est difficile d'évaluer les performances de cet *ADC* dans la mesure où aucune entrée/sortie de testabilité n'a été prévue. J'ai testé quatre puces différentes pour des résultats similaires. Néanmoins, ces tests se sont tout de même avérés bénéfiques pour la conception de ma propre puce comme nous le verrons par la suite. Suite à ces tests, j'ai en effet jugé opportun d'ajouter des pins de testabilité en des points importants du circuit afin d'évaluer le fonctionnement des différents sous-ensembles le constituant.

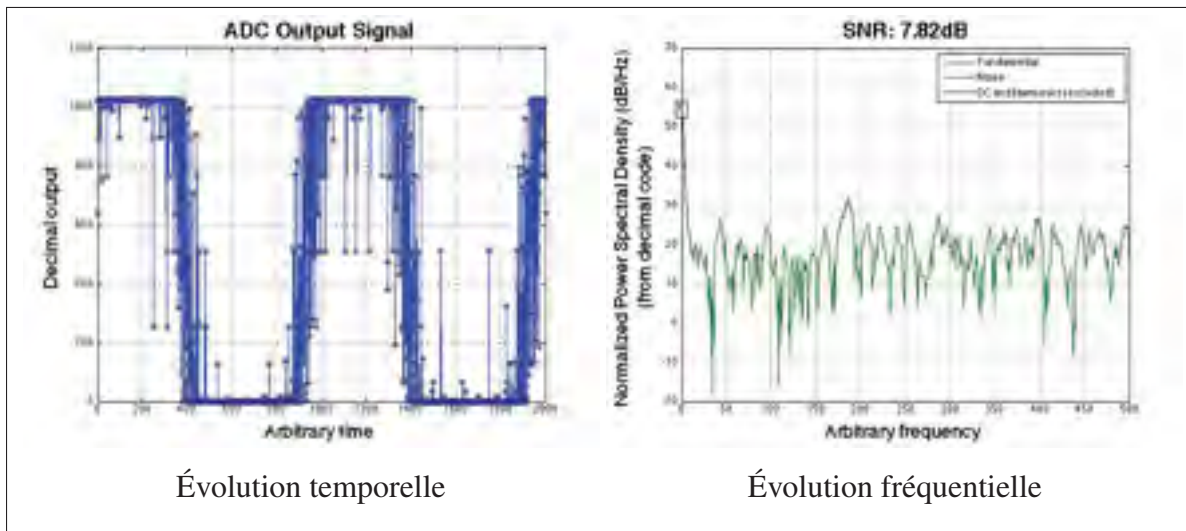


Figure 2.5 Résultats expérimentaux de l'ADC 2015 en mode natif

## CHAPITRE 3

### CAHIER DES CHARGES

La technologie choisie pour cette conception est le *CMOS* 130nm initialement produite par *IBM* et maintenant proposée par *GlobalFoundries*. Elle présente l'avantage d'être économique, assez complète et éprouvée du point de vue des composants proposés. La tension de référence de cette technologie est de 1.2V mais certains composants sont disponibles pour des tensions allant jusqu'à 3.3V. Dans un souci de diminution de la consommation de puissance, nous ne travaillerons pas au-delà d'une tension de 0.9V. Ce choix d'alimentation a bien entendu un impact sur la rapidité du système mais celui-ci est acceptable dans la mesure où nous opérons ici sur un système à basse fréquence. En effet, la fréquence d'horloge est de 1MHz ce qui nous permet, pour une résolution de 10 bits, d'atteindre un taux de conversion de  $83kS \cdot s^{-1}$ . La résolution native de notre *ADC* sera de 10 bits mais un système de *dithering* doit permettre une extension de cette résolution à 13 bits moyennant une diminution importante du taux de conversion. Afin de limiter les effets parasites liés au bruit inhérent à chacun des éléments du circuit, l'architecture proposée sera différentielle. L'*ADC* aura donc pour mission de convertir la différence de tension entre les deux entrées de la puce.

Nous allons à présent considérer l'architecture qui nous permettra d'atteindre ces spécifications.



## CHAPITRE 4

### NOUVELLE ARCHITECTURE PROPOSÉE

#### 4.1 Le *Dithering*

Le *dithering* constitue le cœur de ce projet dont l'objectif principal est de tester la viabilité et les performances de ce type de reconfigurabilité. Comme on l'a vu précédemment (*cf.* 2.1, page 13), la version 2015 de la puce embarquait déjà cette fonctionnalité mais les problèmes rencontrés lors des tests en laboratoire ne m'ont pas permis de la valider. Le principal obstacle dans la mise au point de cette méthode reste la difficulté à la simuler. En effet les temps de simulation, déjà long pour le mode de fonctionnement standard de l'ADC, sont ici multipliés par 15 rendant la vérification informatique de cette méthode compliquée à mettre en œuvre. Néanmoins, des tests purement algorithmiques sous *Matlab* restent possible. Initialement, j'ai mené ces simulations dans le seul but de justifier et de comprendre le principe d'un *dithering*. J'y ai néanmoins trouvé une perspective d'amélioration de cette fonctionnalité par rapport à la version 2015 de l'ADC.

Sous *Matlab*, il est aisé de tracer les fonctions de transfert théoriques de l'ADC. Il s'agit là de faire correspondre sur un même graphique la tension différentielle imposée à l'entrée de l'ADC (*cf.* axe des abscisses) et le code obtenu à sa sortie (*cf.* axe des ordonnées). Le fonctionnement typique d'un ADC est une structure en marche d'escalier (*cf.* figure 4.1, page 22).

Considérons quelques instants la figure 4.1. Les cinq courbes proposées sont toutes ramenées sur une échelle de 12 bits. Les valeurs initialement sur 10 bits sont ainsi multipliées par quatre et celles sur 13 bits divisées par deux. En bleu, on trouve la courbe théorique de l'ADC 10 bits (*i.e.* *Native (10 bits)* sur la figure 4.1, 22). Cette courbe représente le fonctionnement natif de l'ADC donc lorsque le *dithering* est désactivé. La deuxième courbe (*i.e.* *Dithering ADC2015 (12 bits)* sur la figure 4.1) représente le fonctionnement en mode *dithering* de la version 2015 de l'ADC. On constate sur cette courbe que le nombre de paliers a été multiplié par quatre par rapport à la résolution native de 10 bits. On a ainsi atteint une résolution de 12 bits. À

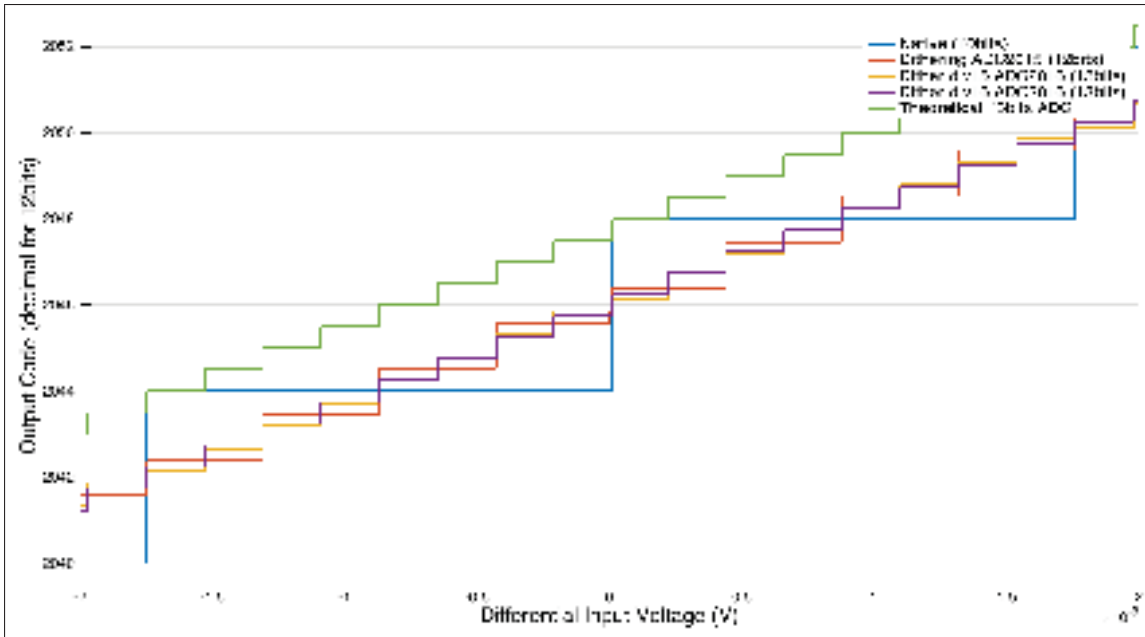


Figure 4.1 Comparaison des fonctions de transfert de différents modes de fonctionnement envisagés pour l'ADC

titre de rappel, ce mode *dithering* associe les capacités  $C$ ,  $2C$  et  $4C$  comme parasites lors de l'échantillonnage. Une amélioration du système est possible à ce stade. En effet, si au lieu d'utiliser les capacités  $C$ ,  $2C$  et  $4C$  lors de l'échantillonnage, on utilise les valeurs  $\frac{C}{2}$ ,  $C$  et  $2C$ , on aboutit à la courbe jaune (*i.e. Dither div15 ADC2016 (13 bits)* sur la figure 4.1). On constate que le simple fait de changer la séquence des capacités en divisant celles-ci par deux nous a fait gagner un bit de résolution.

Cette amélioration est due à la tension générée par les capacités lors de l'échantillonnage. Par définition la capacité unitaire correspond à un  $LSB$ . Lorsque celle-ci n'est pas connectée à  $V_{cm}$  (*i.e.  $\frac{V_{dd}}{2}$* ) lors de l'échantillonnage, la perturbation engendrée sur la tension de sortie du DAC est alors  $\frac{V_{LSB}}{2}$ . L'ADC étant différentiel et le *dithering* n'opérant que sur une seule de ses deux branches, la perturbation sur le système complet est ainsi  $\frac{V_{LSB}}{4}$ . Ainsi, avec les capacités  $C$ ,  $2C$  et  $4C$ , la version 2015 de l'ADC est en mesure de générer les perturbations suivantes sur le système complet :  $\pm \frac{V_{LSB}}{4}$ ,  $\pm \frac{V_{LSB}}{2}$ ,  $\pm \frac{3V_{LSB}}{4}$ ,  $\pm V_{LSB}$ ,  $\pm \frac{5V_{LSB}}{4}$ ,  $\pm \frac{3V_{LSB}}{2}$ ,  $\pm \frac{7V_{LSB}}{4}$ . En divisant la séquence par deux, on divise également les perturbations possibles par deux, soit :  $\pm \frac{V_{LSB}}{8}$ ,

$\pm \frac{V_{LSB}}{4}$ ,  $\pm \frac{3V_{LSB}}{8}$ ,  $\pm \frac{V_{LSB}}{2}$ ,  $\pm \frac{5V_{LSB}}{8}$ ,  $\pm \frac{3V_{LSB}}{4}$ ,  $\pm \frac{7V_{LSB}}{8}$ . Le pas de la séquence détermine le gain en résolution, ainsi un pas de  $\frac{V_{LSB}}{2}$  nous donne un bit supplémentaire, un pas de  $\frac{V_{LSB}}{4}$  deux bits supplémentaires et un pas de  $\frac{V_{LSB}}{8}$  trois bits supplémentaires. On ne peut cependant pas aller au delà, en effet à  $\frac{V_{LSB}}{16}$  les paliers ne sont plus réguliers comme en témoigne la figure 4.2 en page 23. La courbe proposée correspond à une division de la séquence de condensateur par huit ce qui devrait donner une résolution de 15 bits (le phénomène apparait dès 14 bits mais il y est moins flagrant). Ce phénomène peut s'expliquer relativement simplement. En effet à partir de 14 bits, la plus grosse tension parasite accessible sur le DAC est inférieure à  $2 \cdot V_{LSB}$  soit  $V_{LSB}$  à l'échelle du système différentiel. Par conséquent, les parasites ne sont pas en mesure d'augmenter ou de réduire la tension d'entrée pour permettre la modification du code de sortie si la tension d'entrée est trop proche d'un code en résolution native. Ainsi pour les tensions d'entrée trop proches des paliers 10 bits, toutes les conversions donnent le même résultat et ce, quelque soit le biais d'échantillonnage ajouté. On moyenne ainsi 15 valeurs identiques égales au résultat natif ce qui explique la stagnation du code 13 bits autour des codes 10 bits.

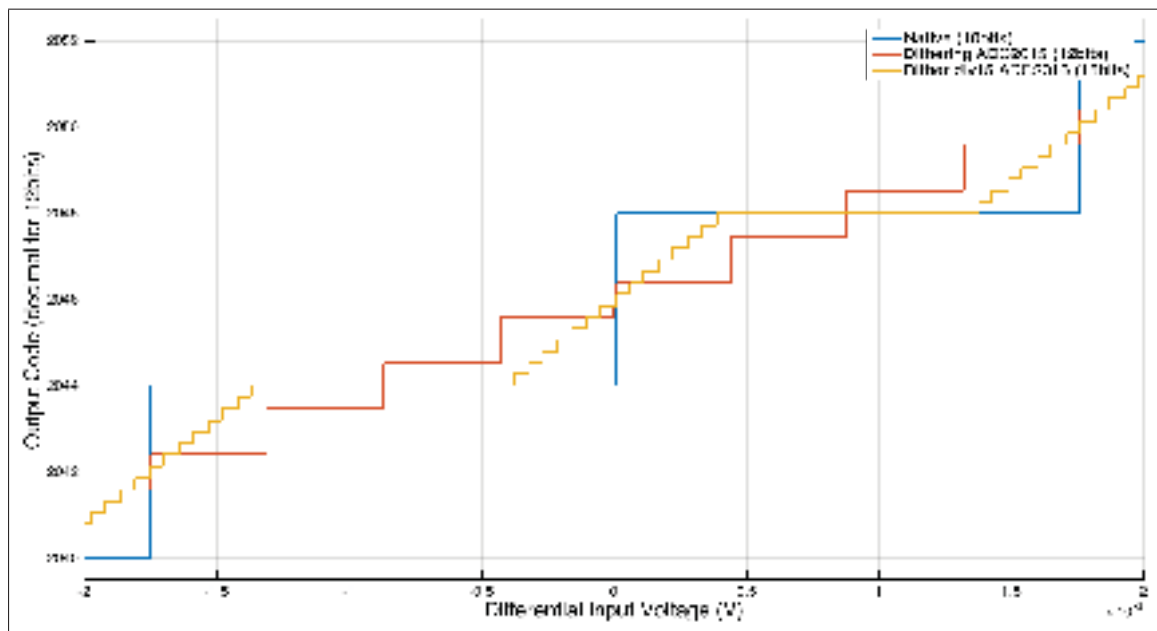


Figure 4.2 Mise en évidence de l'impossibilité de dépasser 13 bits de résolution en divisant la séquence de condensateurs

Nous allons donc tenter d'atteindre une résolution de 13 bits en divisant la séquence de *dithering* par deux. Pour cela, il conviendra d'ajouter au réseau de capacités deux condensateurs élémentaires montés en série afin de créer la capacité  $\frac{C}{2}$ .

L'utilisation d'un flot de conception automatisé pour la création des parties numériques nous permet de complexifier ces dernières. Ceci permet notamment le traitement des données du *dithering* en interne. Concrètement, sur la version 2015, les données étaient récupérées en sortie du système et on devait moyennner les 15 valeurs sous *Matlab*. L'automatisation nous permet de gérer ceci en interne. Néanmoins la mise en place d'un moyennage sur 15 valeurs reste complexe et gourmande en énergie, ce qui va à l'encontre de la philosophie de cette puce (*i.e.* basse fréquence et faible consommation). La solution proposée ici, consistant à moyennner 16 valeurs, est bien plus simple et économe d'un point de vue algorithmique et énergétique. En effet, la somme de 16 mots de 10 bits donne naturellement un mot de 14 bits, il ne reste alors qu'à supprimer le dernier bit pour ne conserver que le résultat sur 13 bits.

La question est donc où trouver cette 16<sup>ème</sup> valeur ? On va utiliser la valeur initiale trouvée par l'ADC (*i.e.* celle sans perturbation). Le choix de cette valeur n'est pas anodin. Étant centrale du point de vue des tensions ajoutées ou retranchées au signal échantillonné lors de la séquence de *dithering*, c'est en effet elle qui sera le moins susceptible de perturber le résultat final. Pour en être sûr, il convient de comparer ce mode de calcul (*cf.* courbe *Dither div16 ADC2016 (13 bits)* en figure 4.1, page 22) avec le calcul idéal sur 15 valeurs (*cf.* courbe *Dither div15 ADC2016 (13 bits)* de la même figure). On constate en figure 4.1 que l'écart entre les deux méthodes est toujours très inférieur à un *LSB*. Pour valider définitivement notre méthode, on trace l'évolution de l'écart entre un ADC 13 bits et notre version de *dithering* moyennée sur 16 valeurs (*cf.* figure 4.3 , page 25).

On constate que l'écart avec un ADC parfait de 13 bits est constant si l'on exclue les pics en  $\pm 0.9V$ . Ces pics sont sans importance car il est fort probable que le fonctionnement de l'ADC pour ces tensions d'entrées soit impossible à cause du comportement de certains éléments ana-



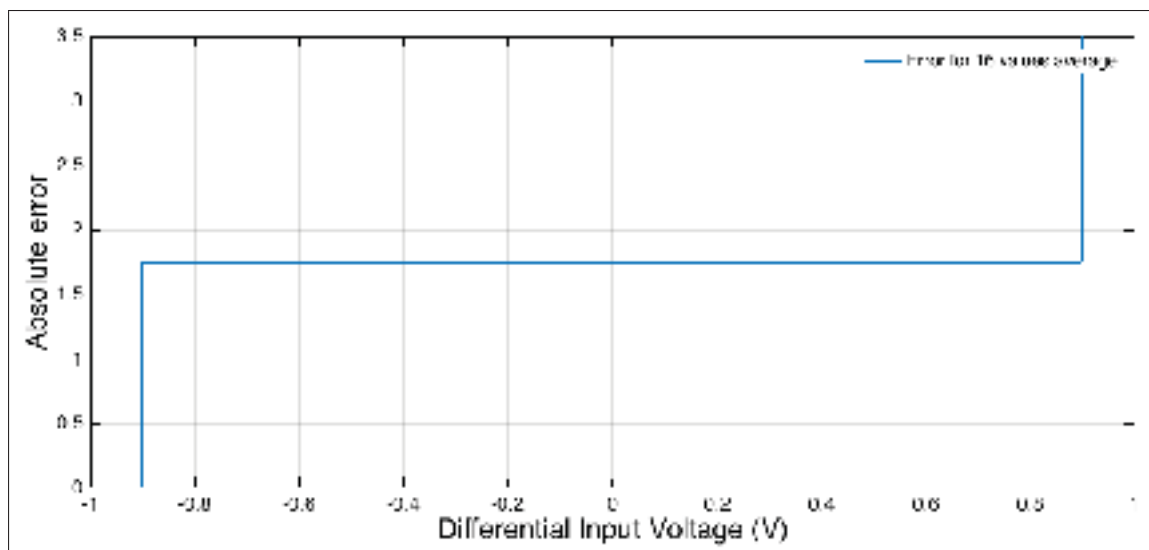


Figure 4.3 Comparatif entre les codes de sortie d'un *ADC* 13 bits théorique et de la solution proposée

logiques comme le *sample-and-hold* ou le comparateur qui ne sont pas conçus pour travailler avec des tensions d'entrées égales à la tension d'alimentation.

## 4.2 Nouvelle Architecture

L'architecture générale est proche de celle de la version 2015 de l'*ADC*. La différence majeure réside dans la modification de la séquence de *dithering* mais surtout l'utilisation d'outils automatisés pour la création des blocs logiques. En utilisant ces outils, on vise une forte baisse de la consommation d'énergie mais aussi le développement de la testabilité et de la facilité d'utilisation du circuit comme nous le verrons par la suite. La réduction de la consommation d'énergie peut également se traduire par une réduction des tensions d'alimentation. Dans cette optique, la partie numérique est sous-alimentée en région *subthreshold*. La tension d'alimentation de la partie numérique sera déterminée de façon expérimentale sur la puce réelle. Il est néanmoins indispensable d'utiliser des *level shifters* (non représentés sur le schéma 4.4) sur tous les signaux situés entre les parties basse et haute tension de la puce. Nous allons ici développer chaque sous-ensemble de la puce afin d'en expliciter l'architecture.

### 4.2.1 Architecture générale

L'architecture générale du système est représentée en figure 4.4 (page 26) et se décompose en sept sous-blocs. Notons à ce stade que le découpage de l'architecture proposée ici ne correspond pas à celui de la puce réelle mais présente l'avantage d'être plus compact. En effet, des parties numériques sont également présentes dans les *DAC* comme nous le verrons par la suite.

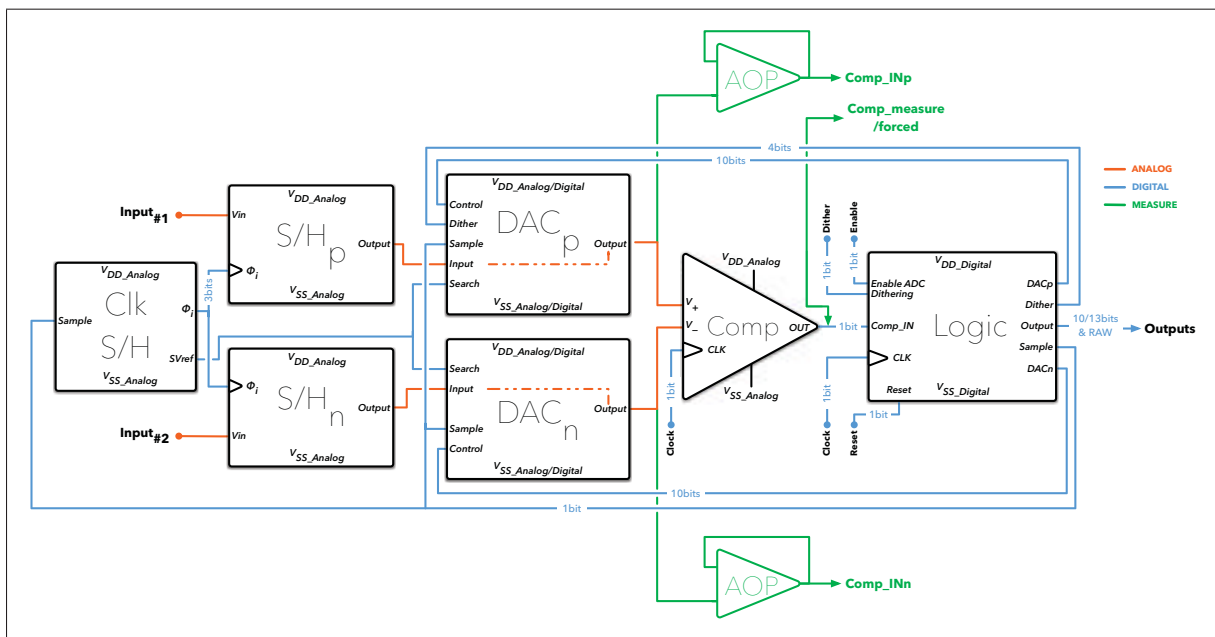


Figure 4.4 Architecture générale de la nouvelle version de l'ADC

L'entrée analogique du système se fait par les deux *sample-and-hold*. Ceux-ci sont pilotés par un module nommé *Clk S/H* et dont nous expliquerons le fonctionnement plus tard (*cf.* partie 5.1.1, page 31). Une fois les *DAC* chargés, le bloc logique pilote les jeux de condensateurs suivant l'algorithme *SAR* en fonction des données fournies par le comparateur.

Le choix d'une architecture à alimentation multiple (en anglais *MSV* pour *Multiple Supply Voltages*) et l'utilisation d'un *DAC* hybride complexifie l'architecture. La figure 4.5 (page 27) donne une vision de l'architecture plus proche de l'implémentation réelle. On y trouve le bloc logique (qui est le seul élément fonctionnant à basse tension d'alimentation), l'ensemble des

*level shifters* regroupés dans un même bloc (cf. *Level-Shifter Bank*), les deux *drivers* des DAC et enfin la partie analogique qui va contenir tous les autres éléments ainsi que les deux amplificateurs opérationnels (non représentés sur le schéma). Cette décomposition de la puce en cinq sous-parties est exactement celle que l'on trouvera sur la puce réelle.

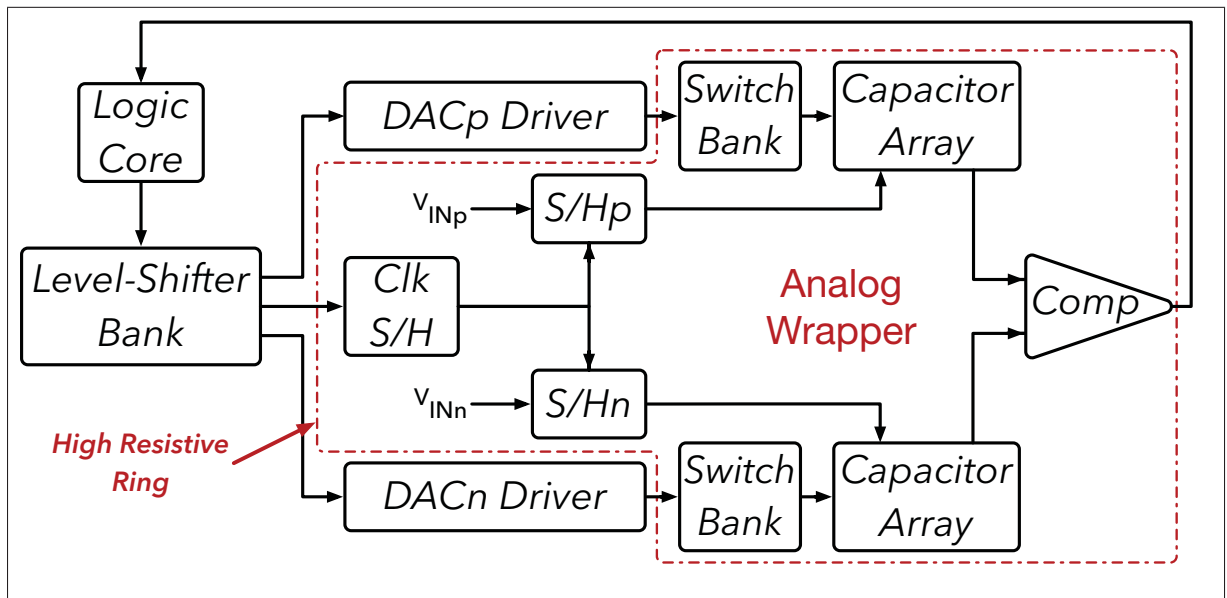


Figure 4.5 Représentation de la répartition physique des sous-ensembles de la puce

On peut décomposer les DAC en trois grandes structures : les *drivers*, les interrupteurs et le réseau de capacités.

Les *drivers* sont essentiellement composés de multiplexeurs qui ont pour mission de définir quels interrupteurs doivent être commutés en fonction des signaux de contrôle en provenance du SAR, des signaux de *dithering* et des signaux d'échantillonnage. Les interrupteurs sont au nombre de trois par capacités, chacun d'entre eux permettant de connecter la base du condensateur à  $V_{DD}$ ,  $V_{cm}$  ou  $V_{SS}$ .

L'association d'un DAC hybride à une architecture MSV impose de déterminer où placer les *level shifters* c'est à dire de déterminer quels sont les blocs qui fonctionnent en basse tension et quels sont les autres. Comme sur la version 2015 de l'ADC, l'architecture hybride rend indis-

pensable l'implémentation de convertisseurs binaire-thermomètre (cf. *Binary/Unary Converter* sur le schéma 4.6, page 28) pour *driver* le DAC unaire. En effet, les signaux de contrôle en provenance du SAR sont sur 10 bits dont 5 à destination de la partie unaire (les 5 *MSBs*) mais il faut ici *driver* 31 condensateurs au lieu de 5. Le convertisseur transforme donc ces 5 bits en 31 bits pilotant le DAC unaire.

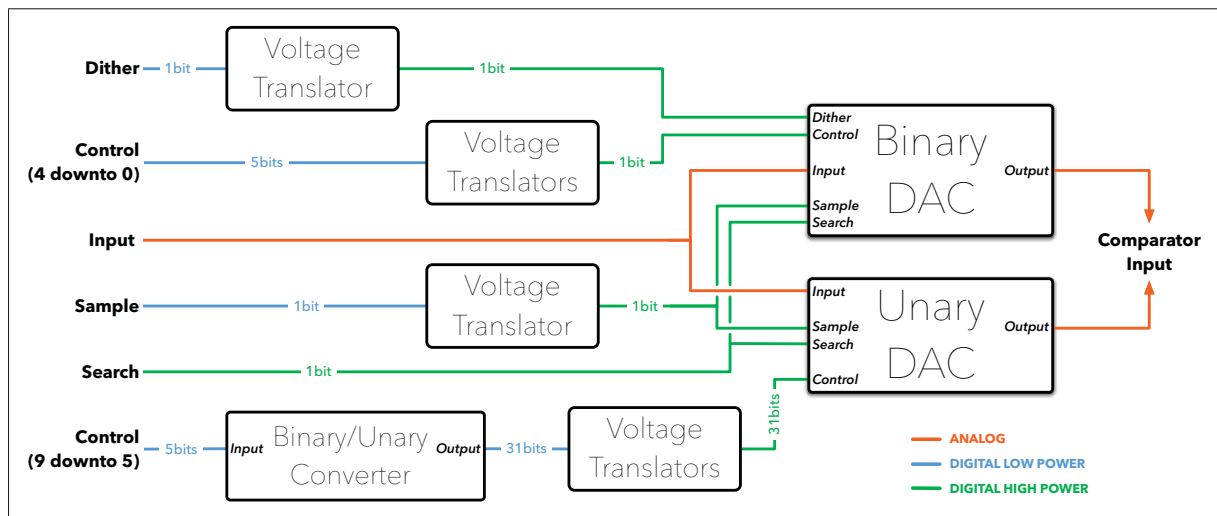


Figure 4.6 Représentation de la distribution des signaux sur les DAC via les *level-shifters*

Pour résumer la situation, nous avons donc trois éléments numériques susceptibles de fonctionner à basse tension d'alimentation : le cœur logique, les *DAC driver* et les convertisseurs binaire-thermomètre. Trois solutions peuvent être envisagées :

1. Faire fonctionner les trois éléments en basse tension,
2. Ne faire fonctionner que le cœur logique en basse tension,
3. Faire fonctionner le cœur logique et les convertisseurs en basse tension.

La première solution est exclue. En effet, en sortie des *DAC driver*, le nombre de signaux est multiplié par trois (un par interrupteur à piloter pour chaque condensateur) ce qui impliquerait un nombre de *level shifters* très importants. La place disponible sur la puce ne nous permet pas d'envisager cette option. De plus, les *DAC driver* étant combinatoires, leur consommation dans la situation présente est minime et l'utilisation de tant de convertisseurs de tension dans

le seul but de sous-alimenter cette partie n'est pas forcément rentable étant donné qu'on atteint le seuil critique où la consommation de tous les *level shifters* ajoutés est supérieure à celle du bloc alimenté normalement.

La deuxième solution est intéressante mais moins que la troisième qui constitue un compromis idéal dans notre cas. L'idée est de placer les *level shifters* en amont des *DAC driver*, qui fonctionneront donc en haute tension, mais en aval des convertisseurs binaire-thermomètre (cf. figure 2.2, page 28). Avec cette solution, deux de nos trois blocs numériques fonctionnent à basse tension d'alimentation. Pour l'implémenter sur l'ensemble de la puce, on utilise 78 *level-shifters* pour une surface cumulée de  $0.2\text{mm}^2$ .

On remarquera pour finir que sur le schéma 4.6, le signal *Search* n'est pas associé à un *level shifter* car celui-ci est en provenance du bloc *Clk S/H* inclut dans le sous-ensemble *Analog Wrapper* (cf. figure 4.5, page 27) et est donc déjà à haute tension.

#### 4.2.2 Testabilité

Comme on a pu le constater avec les essais en laboratoire de la version 2015 du circuit, la testabilité est un point à privilégier si on veut être en mesure d'identifier les éléments non fonctionnels en cas de problème. Dans cette optique, des points de mesure numériques et analogiques ont été implantés sur différents nœuds stratégiques du circuit (*cf.* figure 4.4, page 26). Ces nœuds stratégiques correspondent aux entrées et à la sortie du comparateur.

Sur le plan numérique, on veut donc mesurer la sortie du comparateur. Celle-ci étant numérique, il n'y a pas de problème de perturbation du signal par la mesure. On se contente simplement de placer deux inverseurs en série sur le signal de mesure pour ne pas obliger le comparateur à *driver* cette partie du circuit en plus de l'entrée du cœur logique. La particularité de ce point de mesure est qu'il est réversible. Concrètement, on va pouvoir utiliser ce point comme sortie (mode mesure) ou comme entrée (on force un signal en provenance de l'extérieur).

Du côté analogique, on mesure les deux entrées du comparateur. Cette fois-ci, il est impératif de ne pas perturber le signal analogique en sortie des *DAC*. Pour cela, on utilise des amplificateurs opérationnels montés en suiveur. Ils vont permettre de découpler le point de mesure du reste du circuit et ainsi de ne pas perturber le fonctionnement normal de l'*ADC*.

Grâce à ces deux mesures analogiques et au mode "forcé" du point de mesure numérique il sera alors possible, si nécessaire, de shunter le comparateur pour effectuer la comparaison à l'extérieur de la puce. Ces trois points de mesure nous donnent ainsi la marge de manœuvre nécessaire pour identifier et éventuellement corriger un maximum de problèmes sur la puce.

Nous allons à présent analyser unes à unes les conceptions des différents éléments constitutifs du circuit.

## CHAPITRE 5

### CONCEPTION DE LA PUCE

Cette partie du rapport est consacrée à l'étude de chacun des éléments du circuit.

#### 5.1 Éléments analogiques

##### 5.1.1 *Sample-and-Hold*

Le *sample-and-hold* (abréviation *S/H*) est une composante importante du convertisseur analogique numérique. Son architecture est tirée de (Dessouky & Kaiser, 2001). Le schéma du circuit est représenté en figure 5.1 (page 32). Celui-ci fonctionne via deux signaux d'horloge non recouvrant (*i.e.*  $\phi_{1n}$  et  $\phi_{2n}$ ) et un troisième (*i.e.*  $\phi_{2p}$ ). Ces signaux sont issus du générateur d'horloge décrit dans 5.3 en page 33. Dans l'idéal, l'objectif de ce montage est de transmettre le signal d'entrée en sortie avec le maximum de fidélité lorsque le montage est passant. En revanche, lorsque le montage est bloqué, il ne doit fournir aucun effet parasite au reste du circuit. Dans cette perspective, on utilise une capacité qui sera chargée lorsque le montage est inactif, puis connectée à la grille de la demi-porte de transmission (*i.e.* le transistor  $M_2$  sur la figure 5.1) lors de l'acquisition afin de favoriser la commutation de cette dernière. On appelle cette technique un *bootstrap*.

On commence le cycle avec  $\phi_{1n} = \phi_{2p} = V_{SS}$  et  $\phi_{2n} = V_{DD}$ . Les transistors  $M_8$ ,  $M_5$  et  $M_1$  sont donc passants alors que  $M_6$  est bloqué.  $M_5$  passant implique  $M_3$  bloqué. Par conséquent, la capacité est chargée entre  $V_{DD}$  et  $V_{SS}$  (par intermédiaire de  $M_4$  et  $M_8$ ) alors que la grille du transistor  $M_2$  (*i.e.* la demi porte de transmission) est connectée à la masse. Pour résumer, on charge C alors que le transistor  $M_2$  est inactif.

Lors de la deuxième étape, les signaux  $\phi_{1n}$  et  $\phi_{2p}$  montent au potentiel  $V_{DD}$  alors que  $\phi_{2n}$  redescend à  $V_{SS}$ . En conséquence, les transistors  $M_8$ ,  $M_5$  et  $M_1$  sont alors bloqués et  $M_6$  devient passant. La connexion de la capacité sur  $M_3$  le rend également passant. Ainsi, la présence d'un

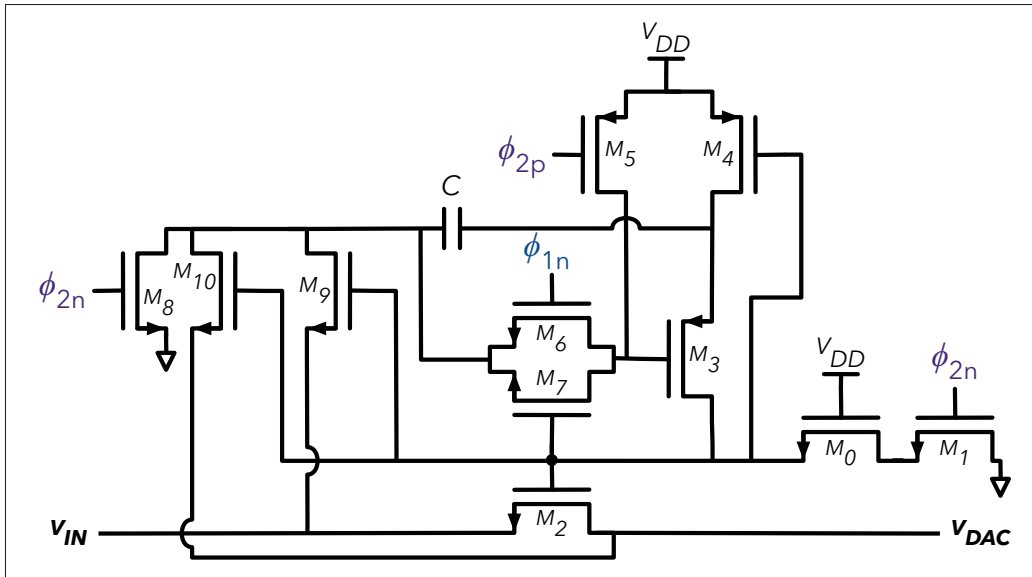


Figure 5.1 Schéma électrique du *sample-and-hold*

signal en entrée, combiné au passage de  $M_3$  passant, entraîne la commutation de  $M_9$ . Ainsi ce n'est plus un potentiel de  $V_{DD}$  qui se présente sur la grille de  $M_2$  mais un potentiel légèrement inférieur à  $V_{IN}+V_{DD}$ . Concrètement le potentiel est de  $V_{IN}+850\text{mV}$  soit  $V_{GS_{M_2}}=850\text{mV}$ . Une façon plus simple de comprendre le fonctionnement du système est de considérer le schéma 5.2 (page 32) et inspiré de (Dessouky & Kaiser, 2001) :

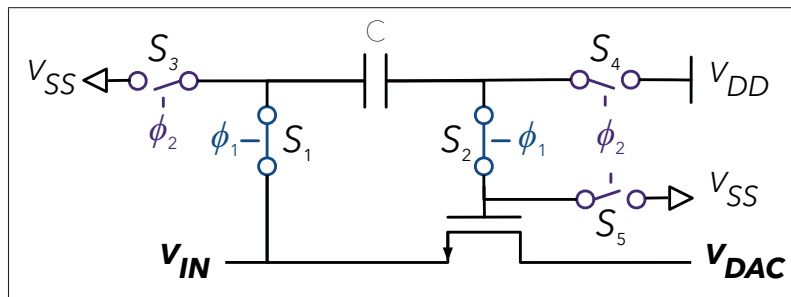


Figure 5.2 Vue schématique du *sample-and-hold*

On observe sur ce schéma que lorsque les interrupteurs  $S_3$ ,  $S_4$  et  $S_5$  sont fermés, alors  $S_1$  et  $S_2$  sont ouverts et on charge la capacité  $C_{offset}$ . La demi-porte de transmission est alors bloquée. Lorsque les signaux commutent, on constate que la capacité est bien connectée entre la



source et la grille du transistor par l'intermédiaire de  $S_1$  et  $S_2$ . On notera que la présence de signaux d'horloge non recouvrant est due à la nécessité de ne pas activer tous les transistors simultanément (ce qui connecterait  $V_{IN}$  à  $V_{SS}$  mais surtout  $V_{DD}$  à  $V_{SS}$  par  $S_4$ ,  $S_2$  et  $S_5$ ). Le générateur d'horloge dont le schéma est disponible en figure 5.3 (page 33) joue sur le temps de transmission des portes logiques pour créer les signaux nécessaires. Enfin, la présence de transistors supplémentaires dans le schéma transistor réside dans la nécessité d'obtenir un système *rail-to-rail* (i.e. fonctionnant pour  $V_{IN}$  allant de  $V_{SS}$  à  $V_{DD}$ ).

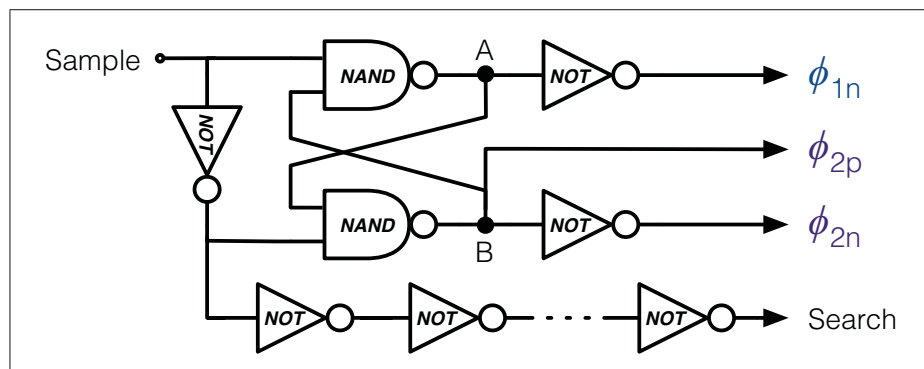


Figure 5.3 Schéma bloc du générateur de signaux commandant les *sample-and-hold*

On utilise également le générateur d'horloge pour créer un signal retardé du signal *sample* (cf. signal *search* sur le schéma 5.3). L'association des quatre signaux ainsi créés et du signal *sample* permet de garantir la séquence suivante entre la fin de l'échantillonnage et le début de la convergence par approximations successives (cf. figure 5.4, page 34). Cette séquence est très importante pour s'assurer que la charge des *DAC* ne sera pas affectée par la commutation du *sample-and-hold*. En effet, toute autre séquence aurait pour effet de connecter  $V_{cm}$  à  $V_{DD}/V_{SS}$  ou de connecter le *sample-and-hold* à  $V_{DD}/V_{SS}$ . Dans le premier cas on connecte la base des condensateurs à  $V_{cm}$  et  $V_{DD}/V_{SS}$  en même temps au lieu de les connecter uniquement à  $V_{DD}/V_{SS}$ . Dans le deuxième cas le *sample-and-hold* continuerait d'échantillonner alors que les condensateurs seraient connectés à  $V_{DD}/V_{SS}$  au lieu de  $V_{cm}$ . Dans les deux cas on introduit un fort biais sur la charge du *DAC* ce qui a nécessairement un impact sur la précision finale de l'*ADC*.

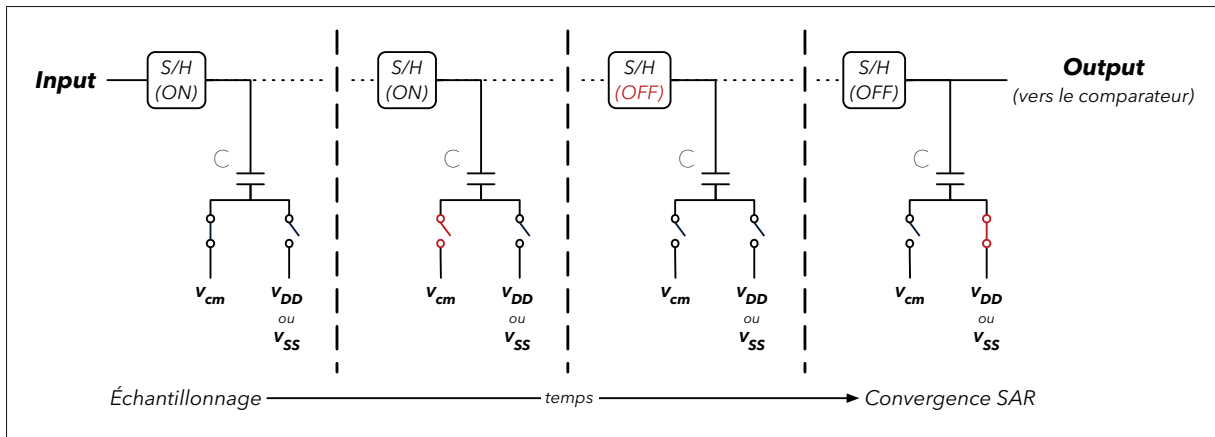


Figure 5.4 Chronologie des commutations des interrupteurs des DAC entre la fin de l'échantillonnage et le début de l'algorithme de recherche SAR

### 5.1.2 Convertisseur numérique/analogique (DAC)

Hormis le changement imposé par la modification de la séquence de *dithering*, la structure du DAC n'a pas changé par rapport à la version 2015 de la puce. Ce que l'on considère comme le DAC dans cette partie se résume à un peigne de capacités associées à leurs interrupteurs respectifs. Le module drivant les DAC sera étudié plus tard dans la partie numérique (cf. section 5.3.3, page 44).

Deux types de capacités ont été envisagés pour ce DAC : les capacités *MIM* pour *Metal Insulator Metal*, et *MOM* pour *Metal Oxide Metal* (cf. figure 5.5, page 35). Elles sont étudiées dans (Chiu & Ker, 2014). Les capacités *MIMs* utilisées sur un projet antérieur ont été écartées au profit des capacités *MOMs*. Ces dernières possèdent une capacité minimale inférieure à celle des capacités *MIMs*. Les DAC traditionnels étant ordonnés de façon binaire (chaque capacité est égale au double de la précédente), minimiser la capacité unitaire (la plus petite du DAC) revient à minimiser la capacité totale du DAC. Des capacités plus petites sont synonymes de temps de chargement moins longs et d'une consommation de courant réduite. Le principal inconvénient des condensateurs *MOMs* est la surface qu'ils occupent par rapport aux *MIMs* à

capacité égale. On verra par la suite que les *DAC* occupent une surface très importante de la puce à cause de ces capacités.

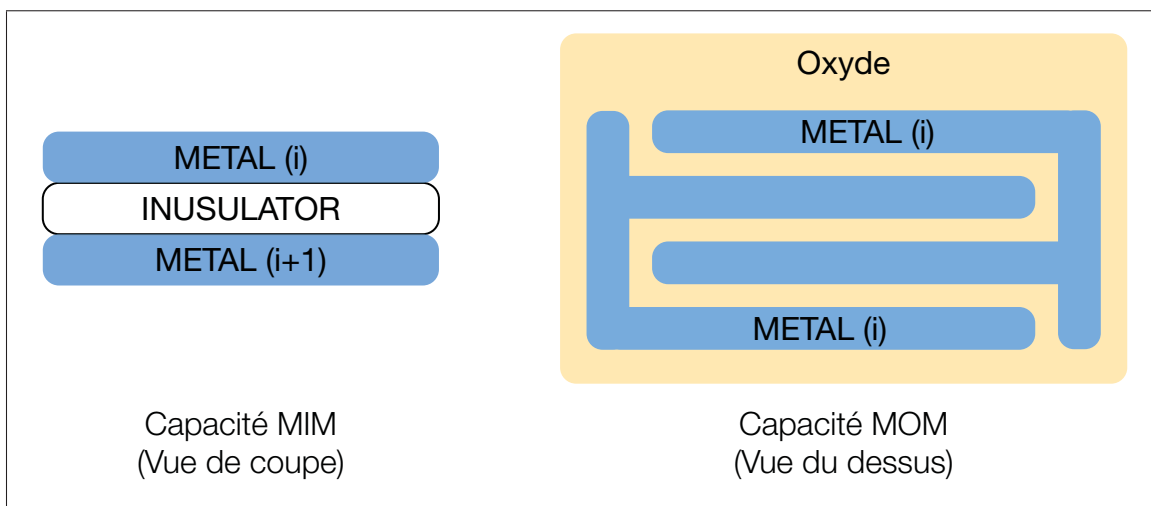


Figure 5.5 Structures des capacités *MIM* et *MOM*

En observant la structure des capacités *MOM*, on constate que celle-ci n'utilise qu'une seule et unique couche de métallisation. Il est possible de superposer ces couches pour obtenir des capacités plus grosses tout en occupant moins de surface sur la puce. La technologie 130nm *GlobalFoundries* que nous utilisons ici donne accès à huit couches de métal et nous permettrait donc de superposer les condensateurs. Néanmoins, cette solution est délicate à mettre en œuvre. En effet, la capacité *MOM* dépend des dimensions du peigne qui la constitue mais aussi des propriétés de la couche de métal sur laquelle on l'implante. Il est alors difficile de garantir la proportionnalité entre les différentes capacités si celles-ci ne sont pas toutes issues de la même capacité unitaire. La régularité du *DAC* étant cruciale pour les performances de l'*ADC* on préférera travailler sur une unique couche de métal afin de garantir que toutes les capacités sont exactement proportionnelles à la capacité unitaire, même si cela implique d'attribuer aux *DAC* une portion très importante de la surface disponible.

Avant de poursuivre, il est important de vérifier que le bruit thermique généré par les capacités ne nous empêchera pas d'atteindre la résolution souhaitée de 13 bits. Pour cela, on applique la

formule 5.1 issue de (Lotfi, Majidi, Maymandi-Nejad, & Serdijn, 2009) :

$$V_{Thermal\ Noise} = \sqrt{\frac{k_B \cdot T}{C_{DAC}}} \quad (5.1)$$

Avec une capacité unitaire  $C_U$  de  $13.4fF$ , on aboutit à une capacité totale de  $27.4pF$  pour un  $DAC$ . Dès lors, à  $300K$ , la tension à considérer pour le bruit est égale à  $12\mu V$ . L' $ADC$  étant différentiel, la valeur de la tension générée par le plus petit condensateur (*i.e.*  $V_{LSB}$ ) est définie par la formule suivante :

$$V_{LSB} = \frac{2 \cdot V_{ref}}{2^{Resolution}} = 220\mu V \quad (5.2)$$

La tension du  $LSB$  étant bien supérieure à  $12\mu V$  (100%) le bruit thermique induit par les condensateurs constituant les  $DAC$  ne devrait pas impacter le fonctionnement de l' $ADC$ .

### 5.1.3 Comparateur

La fonction du comparateur est de déterminer laquelle des deux entrées est la plus haute pour fournir au  $SAR$  l'information nécessaire pour commander les  $DAC$ . Concrètement, c'est lui qui effectue la conversion bit par bit de la tension différentielle résultant des tensions générées par les deux  $DAC$ . Son architecture est tirée de (Zhang *et al.*, 2013) et est représentée en figure 5.6, page 37. Il fonctionne avec l'horloge et actualise son résultat à chaque front descendant de l'horloge.

Le circuit peut se décomposer en deux sous-parties. Tout d'abord, un ensemble constitué des transistors  $M_1, M_2, M_3, M_4$  et  $M_5$ . Ce premier sous-circuit constitue l'entrée du comparateur et prend la forme d'une paire différentielle (constituée ici des transistors  $M_3$  et  $M_4$ ). La polarisation de cette paire différentielle est conditionnée par la valeur du signal d'horloge puisque celui-ci contrôle les bases des transistors  $M_1, M_2$  et  $M_5$ . Ainsi celle-ci n'est polarisée que lorsque l'horloge est au niveau logique bas. Sur la seconde moitié du circuit, si l'horloge est à l'état logique haut, les potentiels  $V_{out+}$  et  $V_{out-}$  sont tirés vers le bas. On peut conclure que lorsque

l'horloge est à '1', le circuit présente une phase de *reset* où les signaux de sortie sont nuls. Dès que l'horloge passe au niveau logique bas, la paire différentielle est polarisée. En fonction du signe de  $(V_{in+} - V_{in-})$  la tension augmente dans  $V_{d+}$  et diminue dans  $V_{d-}$  jusqu'à atteindre  $V_{DD}$  et  $GND$  ou inversement. À terme, les transistors  $M_6, M_7, M_{11}$  et  $M_{12}$  sont bloqués ou en régime de saturation. La forte rétroaction positive qui existe ici maintient les sorties du comparateur à '0' ou '1'. Dans ce régime stable, les courants sont quasiment nuls et le circuit voit sa consommation chuter de manière importante. Le circuit quittera cet état dès que l'horloge passera au niveau logique haut lors du cycle suivant.

Cette architecture garantit une consommation de courant très faible et n'est possible que parce que le comparateur fonctionne de façon séquentielle. En aucun cas une telle conception pourrait être mise en place sur un comparateur à temps continu.

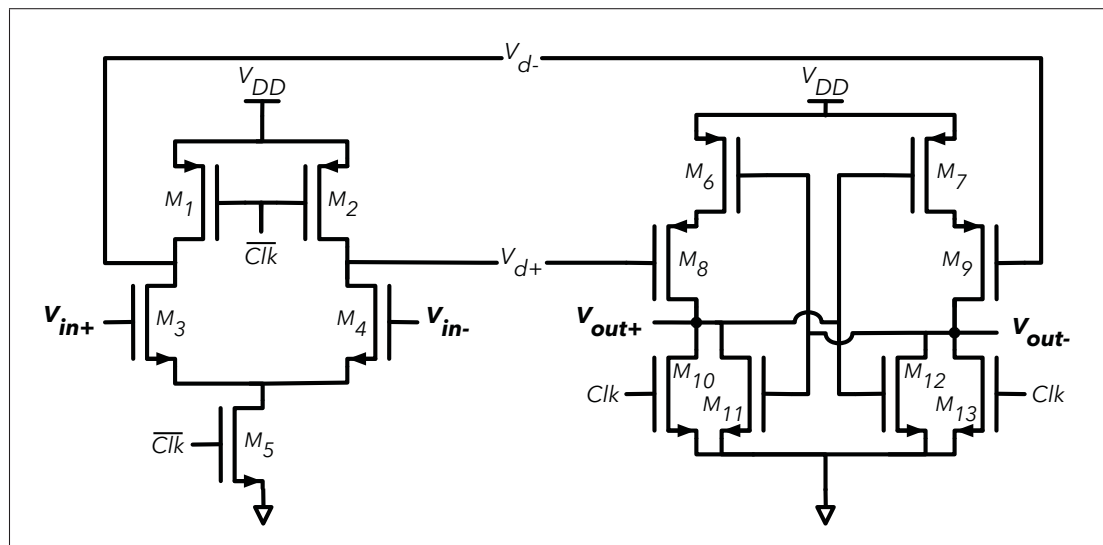


Figure 5.6 Vue schématique du comparateur

## 5.2 Level-Shifters

Comme précisé dans la partie liée à l'architecture (cf. partie 4.2.1, page 26), l'architecture *MSV* (*Multiple Supply Voltages*) mise en place ici impose l'utilisation de *level-shifters* pour convertir les signaux utilisés par le bloc numérique (en basse tension) et le reste de la puce

comme précisé dans la partie liée à l'architecture (*cf.* partie 4.2.1, page 26). Sans eux, les signaux en provenance du cœur logique ne pourraient être compris de façon satisfaisante par les parties numériques fonctionnant à haute tension (*i.e.* les *DAC drivers*). Prenons l'exemple où un niveau logique haut doit transiter entre ces deux éléments. La tension de ce signal est alors égale à  $V_{DDL}$ . Or la tension de transition du transistor récepteur dans le *DAC driver* est potentiellement supérieure à  $V_{DDL}$  puisque celui-ci est alimenté avec  $V_{DDH}$ . Dans le pire des cas, le transistor ne commutera pas et au mieux il commutera bien plus lentement que la norme. De plus une tension de commande trop basse ne permettrait pas de fermer le transistor alimenté en haute tension ce qui entrainerait une consommation de courant excessive. L'ajout de ces sous-circuits est donc indispensable.

L'architecture de ces *level-shifters* est tirée de (Lotfi *et al.*, 2009) et est représentée en figure 5.7 (page 38). Son principe de fonctionnement est analogue à celui du *bootstrap* utilisé pour le *sample-and-hold*.

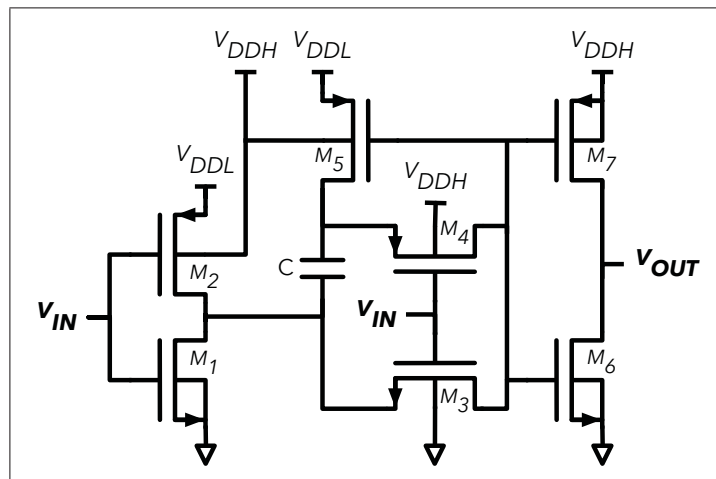


Figure 5.7 Schéma des *level-shifters*

Le circuit est composé de deux inverseurs en série séparés par un ensemble de portes de transmission et d'une capacité. Lorsqu'on présente un '1' en entrée, le premier inverseur transmet alors un '0' à sa sortie. L'entrée à '1' impose également que le transistor  $M_3$  soit passant. Ainsi le deuxième inverseur voit un '0' à son entrée et transmet un '1', mais cette fois la tension as-

sociée n'est plus  $V_{DDLOW}$  comme à l'entrée du montage mais  $V_{DDHIGH}$ . Parallèlement, le fait que  $M_3$  soit passant implique que  $M_5$  soit passant et donc que l'on charge la capacité  $C$  entre  $V_{SSLOW}$  et  $V_{DDLOW}$ . Si ensuite on applique un '0' (*i.e.*  $V_{DDLOW}$ ) en entrée du montage, le premier inverseur transmet un '1' et le transistor  $M_4$  est passant. Ainsi, on attaque le second inverseur avec  $V_{DDLOW}$  plus la charge du condensateur, soit deux fois  $V_{DDLOW}$ . La tension est alors suffisante pour faire commuter le deuxième inverseur qui transmet alors  $V_{SSHIGH}$  à sa sortie. À ce stade, il est important de noter que l'inégalité  $V_{DDHIGH} < 2 \cdot V_{DDLOW}$  doit être respectée pour garantir le fonctionnement du montage.

Abordons maintenant l'architecture et la conception des éléments numériques de cette puce.

### 5.3 Éléments numériques

La conception des éléments numériques constitue l'amélioration majeure de ce projet par rapport à la version de 2015. Les avantages apportés par l'automatisation du processus de synthèse du bloc logique sont multiples : consommation d'énergie, fiabilité, facilité d'ajout de nouvelles fonctionnalités, optimisation surfacique des éléments ainsi créés... Nous allons aborder dans cette partie les spécificités des deux blocs créés grâce au flot de conception automatisé. Commençons par décrire ce flot pour les parties numériques. Le flot de conception complet sera quant à lui présenté en section 5.6 (page 49).

#### 5.3.1 Flot de conception des blocs numériques

La base du flot de conception automatisé d'un système numérique est une description *VHDL* (*i.e.* *VHSIC Hardware Description Language*) de ce même système. À partir de cette description *VHDL*, on utilise un compilateur *RTL* qui va traduire la description *VHDL* en schéma *RTL* à partir des portes logiques disponibles dans la technologie utilisée. Le design de ces portes est caché à l'utilisateur mais on obtient un fichier *verilog* décrivant le circuit (il décrit toutes les interconnexions entre les différentes portes utilisées). On utilise alors un logiciel de synthèse (ici *Encounter*) qui va relier physiquement les portes entre elles à partir du fichier *verilog* et

des paramètres choisis. On obtient alors le *layout* partiel de l'élément logique. Ce design est partiel car les portes logiques n'y sont présentes que sous forme de boîtes noires qui ne seront remplacées par le fondeur qu'à l'envoi du design pour la fabrication de la puce.

Décrivons à présent les deux blocs créés à partir de ce flot de conception.

### 5.3.2 Cœur logique

Le cœur logique regroupe deux éléments numériques de la puce : le *SAR* avec son étage de sortie et le convertisseurs binaire-thermomètre chargés de convertir les signaux du *SAR* à destination du *DAC* unaire.

#### 5.3.2.1 Le *SAR*

La description *VHDL* du *SAR* est inspirée de celle proposée dans (Short, 2008) tout en étant adaptée pour permettre le fonctionnement du mode *dithering*. La machine à états, utilisée pour implémenter le système, est représentée en figure 5.8 (page 40).

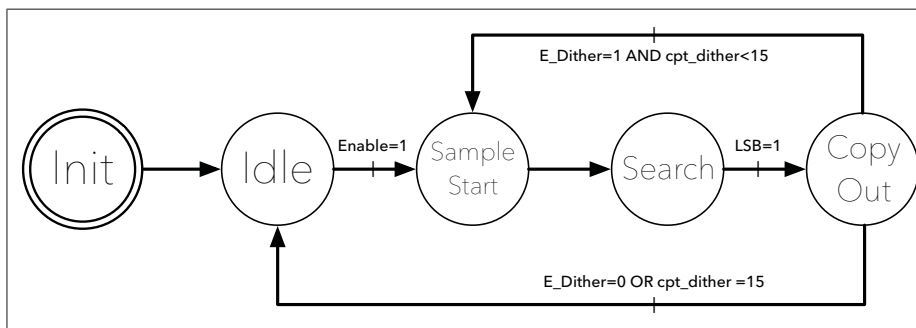


Figure 5.8 Représentation de la machine à états utilisée

À l'issue de l'étape d'initialisation (notée *Init* sur la figure 5.8), l'étape *Idle* précède la conversion qui commence dès que les signaux *Enable* et l'horloge passent au niveau logique haut. On retrouve alors successivement les deux étapes de la recherche par approximations successives. Tout d'abord l'échantillonnage avec l'étape *Sample Start* puis la recherche à proprement parler



avec l'étape *Search*. La recherche est effectuée via le module représenté en figure 5.9 (page 42). Celui-ci est composé de deux registres et d'une porte *OR*.

Le premier registre, nommé *Shiftreg*, va imposer la commutation de chacun des condensateurs, les uns après les autres. Ainsi, ces codes de sortie sont les suivants (cf. tableau 5.1, page 41) :

Tableau 5.1 Évolution des codes de sortie du registre *Shiftreg*

# Clock Cycle	Output Code
1	10000...
2	01000...
3	00100...
⋮	⋮

Le deuxième registre est *Approxreg*, c'est lui qui mémorise le réglage des *DAC*. Ses bits de sortie sont initialement tous au niveau logique bas. Ce registre est mis à jour bit après bit (du *MSB* vers le *LSB*) au fur et à mesure de l'avancement de l'algorithme et en fonction des résultats du comparateur. Ainsi, le registre *Shiftreg* a toujours un cycle d'horloge d'avance sur *Approxreg*. Si *Approxreg* met à jour son  $i^{\text{ème}}$  bit, *Shiftreg* place un '1' sur le  $i+1^{\text{ème}}$  bit. Dès lors, la porte *OR* génère un signal permettant de tester si le  $i+1^{\text{ème}}$  condensateur du *DAC* doit être commuté ou non. Si c'est le cas *Approxreg* mémoriser ce réglage au prochain coup d'horloge en plaçant un '1' sur le  $i+1^{\text{ème}}$  bit, dans le cas contraire il y placera un '0'. Lorsque l'ensemble des bits ont été parcourus par *Shiftreg*, il reste un coup d'horloge pour que le comparateur évalue la valeur du *LSB* et que *Approxreg* la mémorise avant que la machine à états passe à l'état *Copy Out*. Ce changement d'état est consécutif au passage à '1' du dernier bit de *Shiftreg*.

L'état *Copy Out* donne au système le temps nécessaire (*i.e.* un cycle d'horloge) pour stocker le résultat de la conversion dans un registre annexe (cf. *Result Buffer* en Figure 5.10, page 43) et si nécessaire mettre à jour les réglages de *dithering* pour une prochaine conversion. L'état postérieur à *Copy Out* est déterminé par le réglage de l'*ADC*. En fonctionnement standard, on retourne à l'état d'attente *Idle*. En mode *dithering*, on retourne à l'état d'attente si les quinze

valeurs ont été calculées. Dans le cas contraire, on ré-échantillonne le signal avec les nouveaux réglages de *dithering* sur l'état *Sample Start*.

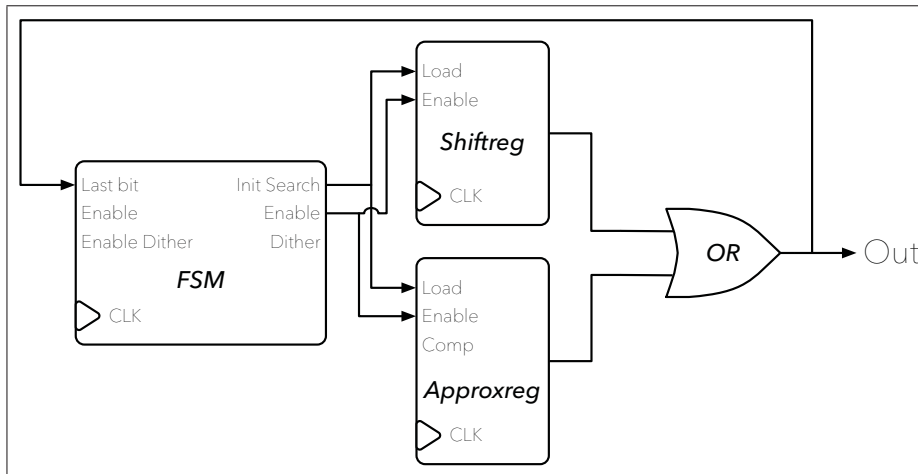


Figure 5.9 Diagramme RTL du registre à approximations successives (SAR)

Lorsque l'algorithme est arrivé à son terme, le résultat doit être traité et acheminé en sortie de la puce. Cette opération est effectuée par le module représenté en figure 5.10 (page 43). En tout temps, un module *PISO* (*i.e. Parallel In - Serial Out*) permet de récupérer les données brutes (*i.e. les données directement issues de chaque conversion que l'ADC soit réglé en mode *dithering* ou non*) de façon sérielle.

Outre cette sortie sérielle, une sortie parallèle sur 14 bits est également disponible. Des multiplexeurs permettent alors de choisir le traitement à appliquer au signal en fonction du mode de fonctionnement. En mode standard, on se contente de concaténer "0000" avec le code de sortie pour obtenir un mot de 14 bits qui est alors acheminé en sortie. En mode *dithering*, on va sommer les résultats au fil des quinze conversions successives grâce à la boucle formée du sommateur et du registre *Sum Buffer* (*cf. figure 5.10, page 43*). Comme expliqué précédemment (*cf. section 4.1, page 21*), la valeur initiale est sommée deux fois pour obtenir seize valeurs et donc un résultat sur 14 bits. La résolution théorique maximale de l'ADC, également détermi-

née en partie 4.1, est de 13 bits. Néanmoins, les 14 bits du résultat de la moyenne calculée ici seront acheminés en sortie à titre expérimental.

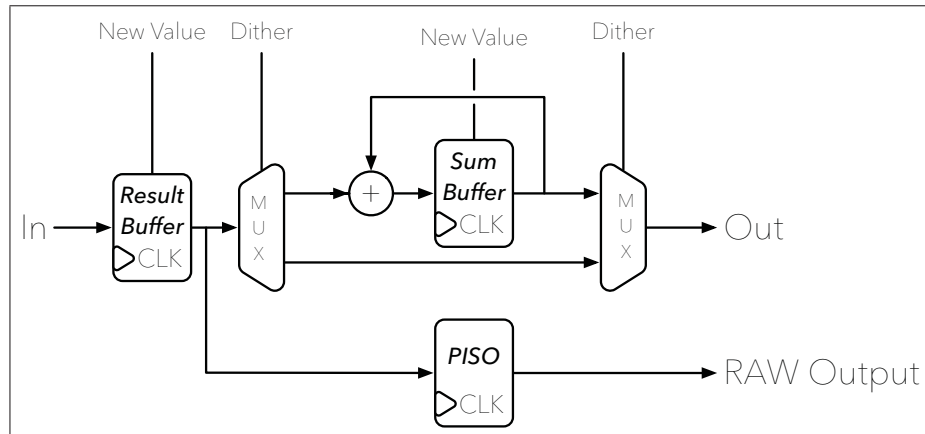


Figure 5.10 Diagramme *RTL* de l'étage de sortie du cœur logique

### 5.3.2.2 Les convertisseurs binaire-thermomètre

Ceux-ci ont pour mission de convertir le signal 5 bits de commande des 5 *MSBs* (en provenance du *SAR*) vers un signal 31 bits susceptible de commander les 31 condensateurs du *DAC* unaire. La description *VHDL* de ces convertisseurs est très simple, il s'agit d'une simple assignation des codes d'entrées vers leur code de sortie respectif (*cf.* tableau 5.2, page 43).

Tableau 5.2 Évolution des codes de sortie des convertisseurs binaire-thermomètre

Decimal	Input Code	Output Code
0	00000	00000000000000000000000000000000
1	00001	00000000000000000000000000000001
2	00010	00000000000000000000000000000011
3	00011	00000000000000000000000000000111
4	00100	00000000000000000000000000011111
⋮	⋮	⋮ ⋮ ⋮

Les condensateurs des *DAC* unaires étant par définition tous identiques, l'ordre des bits n'a pas d'importance. L'essentiel est de conserver le bon ratio de condensateurs chargés/déchargés en accord avec la table 5.2 (page 43). Cette description se traduit ensuite sur la puce par une structure combinatoire relativement simple et économe en énergie. Passons à présent à la description des *DAC Drivers*.

### 5.3.3 *DAC Drivers*

À titre de rappel, le rôle de ces *drivers* est d'interfacer les bits de commande en provenance du SAR et les interrupteurs commandant les condensateurs. Concrètement, le module doit, à partir des signaux *dither*, *sample*, et des bits commande, déterminer lequel des trois interrupteurs actionner pour connecter un condensateur à *Vcm*, *Vrefp* ou *Vrefm*. Les drivers sont identiques pour les deux *DAC* des deux branches de l'*ADC* et sont basés sur un réseau de multiplexeurs. Un certain nombre de ces multiplexeurs sont simplifiés et remplacés par des portes standards lors de la compilation *RTL*.

Les *MUXs* sont commandés par les signaux *Sample* (pendant l'échantillonnage), *Search* (pendant la recherche *SAR*) et le signal 4 bits de *dithering* (uniquement pour les trois condensateurs concernés). À partir de ces éléments, les *MUXs* transmettent '0', '1' ou la valeur de la commande du *SAR* aux interrupteurs des *DAC*. Ainsi, lors de l'échantillonnage, la majeure partie des condensateurs est connectée à *Vcm* (sauf ceux utilisés pour le *dithering* qui sont eux connectés à *Vrefp* ou *Vrefm*). Pendant la recherche algorithmique, les capacités sont connectées à *Vrefp* ou *Vrefm* dépendamment de la valeur de la commande provenant du *SAR*.

## 5.4 Éléments de mesure

Comme précisé précédemment (*cf.* section 4.2.2, page 30) la nouvelle architecture inclut des éléments de mesure facilitant la mise en évidence de problèmes potentiels sur la puce réelle. Ces circuits de testabilité prennent des formes analogique (avec deux amplificateurs opérationnels, un sur chaque entrée du comparateur) et numérique comme nous allons le voir à présent.

### 5.4.1 Amplificateur Opérationnel

L'utilisation d'un amplificateur opérationnel monté en suiveur permet de découpler le point de mesure et le reste du système de mesure (*i.e.* *I/Opad*, circuit de test, sonde et oscilloscope). Ainsi on peut réaliser une mesure sur une section analogique du circuit sans perturber son fonctionnement. La conception d'un amplificateur opérationnel n'étant pas un des objectifs de ce projet nous nous contenterons ici d'une description succincte de son architecture sans entrer dans les détails et techniques de design. L'amplificateur utilisé est relativement classique et se décompose en trois sous-circuits (*cf.* figure 5.11, page 45) : un circuit de polarisation (noté *Bias*), un circuit d'amplification (noté *Amp*), et un circuit de démarrage (noté *Start*). Les schémas des ces trois sous-circuits sont disponibles en annexe (*cf.* figure III-1, page 93).

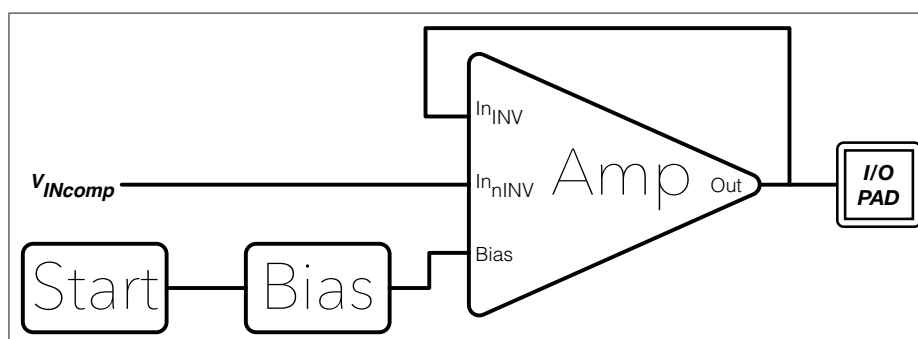


Figure 5.11 Vue schématique des sous-circuits de l'amplificateur opérationnel (AOP)

L'adjonction d'un circuit de démarrage au circuit de polarisation permet de forcer ce dernier à converger vers le bon point de fonctionnement. En effet, si on observe la schématisation de la caractéristique du circuit de polarisation (*cf.* figure 5.12, page 46), on trouve deux points de fonctionnement stables et distincts (*cf.* les points *A* et *B*). Plus concrètement, la stabilité est rencontrée lorsque  $I_{D_{M_1}} = I_{D_{M_2}}$ <sup>1</sup>.

1.  $M_1$  et  $M_2$  sont référencés sur le schéma du circuit de polarisation en annexe III-1, page 93

Ces deux courants sont fonction de  $V_{GS_{M1}}$  et suivent les formules suivantes :

$$I_{D_{M1}} = \frac{\beta_1}{2} \cdot (V_{GS_{M1}} - V_{th_{M1}})^2 \quad (5.3)$$

$$I_{D_{M2}} = \frac{V_{GS_{M1}}}{R} \quad (5.4)$$

Plus simplement, le point *A* correspond à une situation où le circuit de polarisation ne produit pas de courant alors que le point *B* représente le point de fonctionnement nominal du circuit. À la mise sous tension, le circuit de démarrage a ainsi pour mission de forcer le circuit de polarisation à converger vers *B*. Dès lors, le circuit de démarrage ne doit plus influencer le comportement du reste de l'amplificateur opérationnel.

Le circuit de polarisation est quant à lui chargé de fournir un courant stable au circuit d'amplification afin de le polariser. Tous les transistors des circuits de polarisation et d'amplification fonctionnent en régime saturé. Le circuit d'amplification est décomposé en deux étages. Un premier étage constitué d'une paire différentielle qui convertit un signal différentiel (*i.e.* la différence de tension entre ses deux entrées) en signal commun. Le deuxième étage est constitué d'un transistor *PMOS* qui réalise la majeure partie de l'amplification.

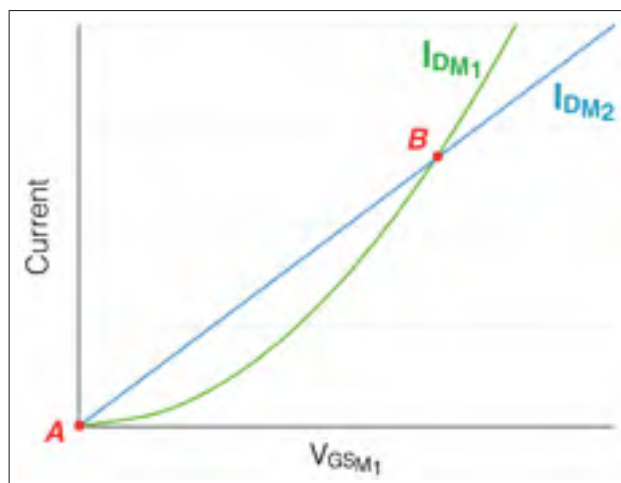


Figure 5.12 Schématisation de la caractéristique du circuit de polarisation

Un des éléments indispensable de ce type de circuit est la compensation en fréquence qui doit garantir le respect de la règle de stabilité de Barkhausen. Sans elle, le circuit est instable car très sensible au bruit et de fortes oscillations apparaissent sur le signal de sortie rendant l'AOP inutilisable. La compensation en fréquence est ici assurée par une capacité réglée de manière à assurer la stabilité (*i.e.* au minimum une marge de phase de  $45^\circ$ ) tout en maximisant la fréquence de coupure. Il s'agit là de trouver un compromis satisfaisant entre gain et réponse fréquentielle. Les performances générales de l'AOP sont disponibles en annexe (*cf.* tableau III-2, page 94).

L'AOP a été dimensionné pour une tension d'alimentation de 1.2V alors que le reste du circuit analogique est alimenté en 0.9V. Cette structure a l'avantage d'éviter l'utilisation d'une structure *rail-to-rail* qui serait plus complexe. En effet, le fait d'utiliser une alimentation supérieure pour l'AOP évite à celui-ci de saturer lorsque l'on s'approche des 0.9V qui constituent la limite de tension du reste de la partie analogique. Le principal inconvénient, qui est par certains aspects avantageux, est l'obligation d'utiliser un plot d'alimentation dédié aux AOP. Ceci peut permettre de ne pas alimenter les AOP si on ne les utilise pas et ainsi de les exclure du bilan de puissance du reste de l'ADC.

## 5.4.2 Points de test numérique

Le point de test numérique est bien plus simple à traiter que les blocs analogiques. On rappelle que l'objectif de ce point de test numérique est de fournir un accès réversible (*i.e.* entrée et sortie) au signal de sortie du comparateur. L'intérêt étant de pouvoir mesurer cette sortie en fonctionnement nominal, mais également de pouvoir imposer un autre signal (provenant de l'extérieur de la puce) à l'entrée du SAR en lieu et place du signal provenant du comparateur interne. Pour réaliser cette fonction, trois signaux sont nécessaires : un signal de sélection du mode de fonctionnement (noté *Comp\_In\_Enable*), le signal de mesure (noté *Measure*) et le signal externe (noté *Comp\_In\_Forced*). Afin de limiter le nombre d'IOPads utilisés par ce système de mesure, on peut mettre en place l'architecture présentée en figure 5.13 (page 48).

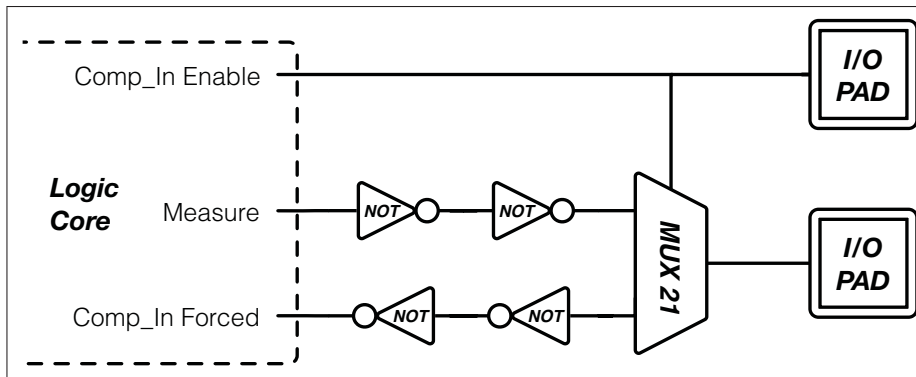


Figure 5.13 Vue schématique du point de mesure numérique

Les signaux *Measure* et *Comp\_In\_Forced* n'étant pas destinés à fonctionner en même temps ils sont connectés au même plot d'entrée/sortie. Un multiplexeur permet de sélectionner (via *Comp\_In\_Enable*) la branche du système à utiliser. Sur ces deux branches, on trouve des inverseurs destinés à *driver* les signaux. Pour *Measure*, l'objectif est de reformer le signal à l'entrée de la puce pour supprimer un maximum de parasites. Pour *Comp\_In\_Forced* l'objectif est de *driver* le plot de sortie. On minimise ainsi les perturbations sur le reste du circuit car on évite au comparateur d'avoir à *driver* à la fois l'entrée du SAR et le plot de sortie.

## 5.5 Distribution de l'alimentation

La structure à alimentation multiple de la puce nous donne l'occasion de structurer l'alimentation des différents sous-circuits en les regroupant à l'intérieur de rails d'alimentation (*power rings* en anglais). Chaque rail est constitué de deux anneaux imbriqués (un pour  $V_{DD}$ , l'autre pour  $V_{SS}$ ). Ces rails d'alimentation sont représentés en figure 5.14 (page 49) et distribuent les trois tensions d'alimentation utilisées à l'intérieur de la puce.

Le cœur logique (qui regroupe le SAR et les convertisseurs binaire-thermomètre) utilise un rail d'alimentation sur la tension  $V_{DIGITAL}$ . Les signaux sont ensuite acheminés vers les *level shifters* qui sont eux alimentés par un double rail d'alimentation qui leur fournit les tensions  $V_{DIGITAL}$  et  $V_{ANALOG}$  nécessaires à leur fonctionnement. Arrivent ensuite les deux DAC *drivers* qui sont munis de leurs rails respectifs sous la tension  $V_{ANALOG}$ . La partie analogique regroupe



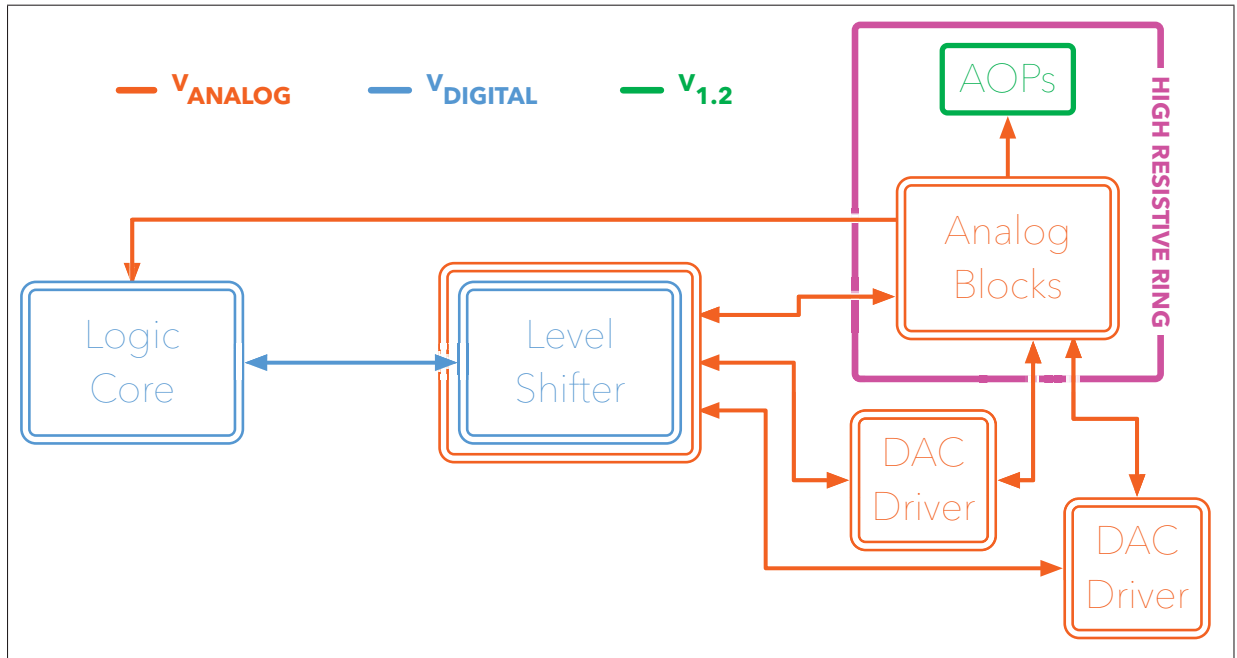


Figure 5.14 Vue schématique de la distribution de l'alimentation

l'ensemble des blocs analogiques et les deux amplificateurs opérationnels. L'ensemble est encerclé par un anneau de haute résistivité destiné à isoler les substrats des parties analogique et numérique. L'objectif étant de limiter l'échange de signaux parasites entre les deux sous-ensembles. Les *AOP* sont intégrés à ce groupe mais bénéficient de leur alimentation propre. Étant donné que se sont les deux seuls éléments connectés à l'alimentation 1.2V, ils ne possèdent pas de *power rings* mais sont directement connectés aux *IOPads* 1.2V et  $V_{SS}$ .

L'ensemble des interconnexions des différents sous-circuits ainsi que la création des rails d'alimentation ont été réalisées via un flot de conception automatisé. Nous allons à présent décrire les grandes étapes de ce flot ainsi que le *layout* final de la puce.

## 5.6 Présentation globale du *layout* et du flot de conception

Les trois outils utilisés pour ce design sont *ISE Xilinx* (pour les descriptions *VHDL* et *Verilog*), *Cadence Encounter* (pour la synthèse des blocs numériques et l'assemblage général) et

*Cadence Virtuoso* (pour le design des éléments analogiques et les finitions manuelles). Les grandes étapes de conception sont représentées en figure 5.15 (page 50).

Le design de la puce commence par l'élaboration des différents sous-circuits qui la composent. Pour les parties numériques on utilise donc *ISE Xilinx* pour décrire l'architecture en *VHDL* puis élaborer les simulations théoriques de celle-ci. On utilise ensuite *Encounter* et un compilateur *RTL* pour synthétiser les blocs numériques. On obtient ainsi le cœur logique et les *DAC drivers*. Parallèlement, on met au point et on simule les blocs analogiques sous *Cadence Virtuoso*. La première étape est complétée par la description des interconnexions en langage *Verilog* sous *ISE Xilinx*. Cette description est vérifiée sous *ISE* en créant des boîtes noires en lieu et place des sous-circuits. Ceci permet de vérifier directement les interconnexions sur le diagramme *RTL* proposé par *ISE*.

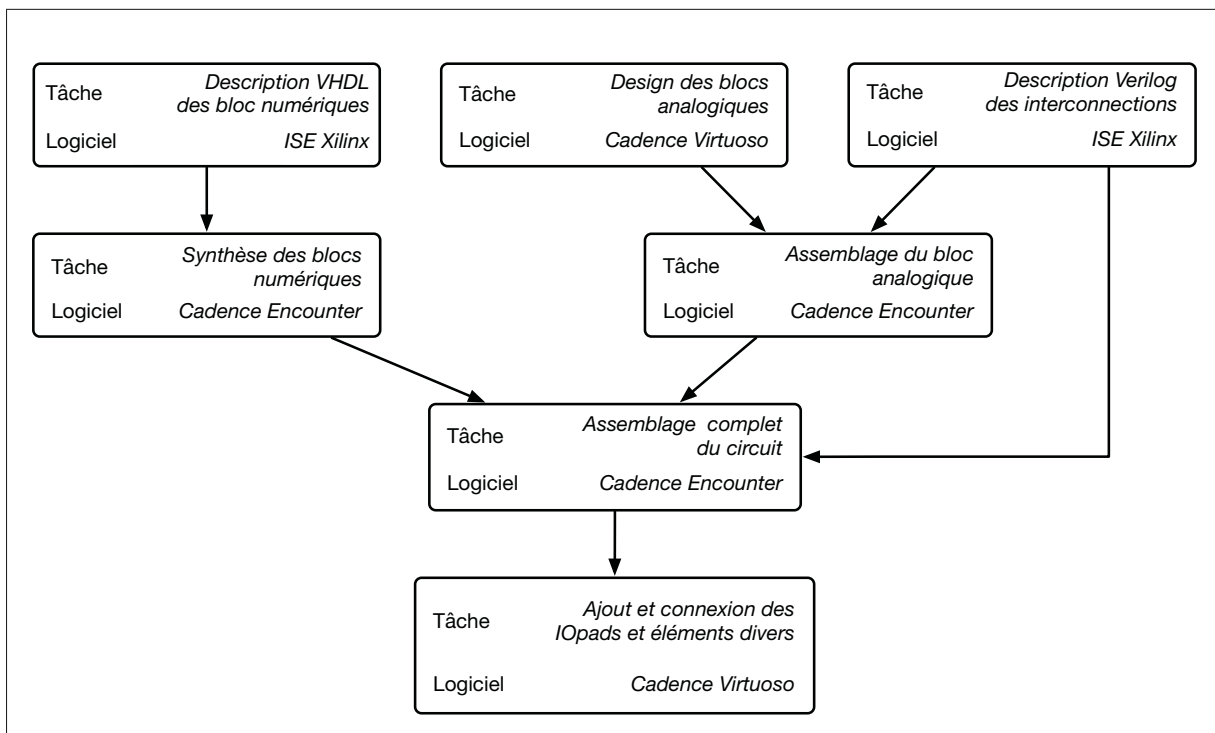


Figure 5.15 Grandes étapes de conception de la puce

Une fois tous les éléments de base mis au point, on utilise *Encounter* pour les assembler. C'est ici qu'on utilise la description *Verilog* précédemment élaborée en commençant par assembler le bloc analogique à l'intérieur de son rail d'alimentation. Avec tous ces éléments, on peut alors procéder à l'assemblage final sous *Encounter* à partir du reste de la description *Verilog*. Lorsque toutes ces étapes ont été menées à bien on peut importer le design sous *Cadence Virtuoso* pour y ajouter les *IOPads* et mener des simulations sur la puce complète via le simulateur *AMS (Analog Mixed-Signal)*.

Le design global de l'ADC ainsi que ceux de ses différents éléments constitutifs est représenté en figure 5.16 (la version avec les plots d'entrée/sortie est disponible en annexe I, page 87). On retrouve sur cette vue les différents sous-circuits de l'ADC. Le tableau 5.3 (page 51) donne la correspondance des numérotations de la figure ainsi que la surface des différents sous-circuits. On constate que la partie analogique occupe la majeure partie de l'espace utilisé. Le reste de l'espace est principalement couvert par les plots d'entrée/sortie et les interconnexions entre les divers sous-circuits.

Tableau 5.3 Numérotations et dimensions des sous-circuits constitutifs de la puce

#	Nom	Surface ( $\mu\text{m}^2$ )	Surface (% de la puce entière)
1	Bloc analogique	307104	30.7
2	Cœur logique	16900	1.69
3	<i>Level Shifters</i>	43964	4.40
4	<i>DAC Drivers</i>	7200	0.72

Ainsi, le bloc analogique est l'élément occupant la surface la plus importante sur la puce, on le retrouve en figure 5.17 (page 54) et en annexe I-2, page 88). La majorité de l'espace est occupée par les peignes de condensateurs *MOM* et particulièrement ceux des *DAC* unaires comme le montre le tableau 5.4 (page 52). Cette représentation est à mettre en regard avec le schéma général de l'ADC en figure 4.4, (page 26).

Tableau 5.4 Numérotations et dimensions des sous-circuits constitutifs de la partie analogique du convertisseur

#	Nom	Surface ( $\mu\text{m}^2$ )	Surface (% de la partie analogique)
1	DAC unaires	248888	81.0
2	DAC binaires	11880	3.87
3	Sample-and-hold <sup>1</sup>	1152	0.375
4	Amplificateurs opérationnels	4235	1.38
5	Comparateur <sup>2</sup>	130	0.0423
6	Générateur de d'horloge <sup>3</sup>	153	0.0498

Outre ces informations sur les dimensions des sous-circuits, on peut retrouver sur ces vues l'organisation décrite en figure 5.14 (page 49). On constate que l'architecture à alimentation multiple à totalement définit l'organisation générale de la puce. L'agencement des différents sous-ensembles dans des rails d'alimentation spécifiques structure la puce et tend ainsi à limiter les interférences entre les sous-circuits qui la constituent. Nous allons à présent aborder la dernière étape ce flot de conception en retraçant les diverses simulations et tests informatiques réalisés sur ce design en vue de valider sa fabrication.

---

1. notés *S/Hp* et *S/Hn* sur la figure 4.4, page 26

2. noté *Comp* sur la figure 4.4, page 26

3. noté *Clk S/H* sur la figure 4.4, page 26

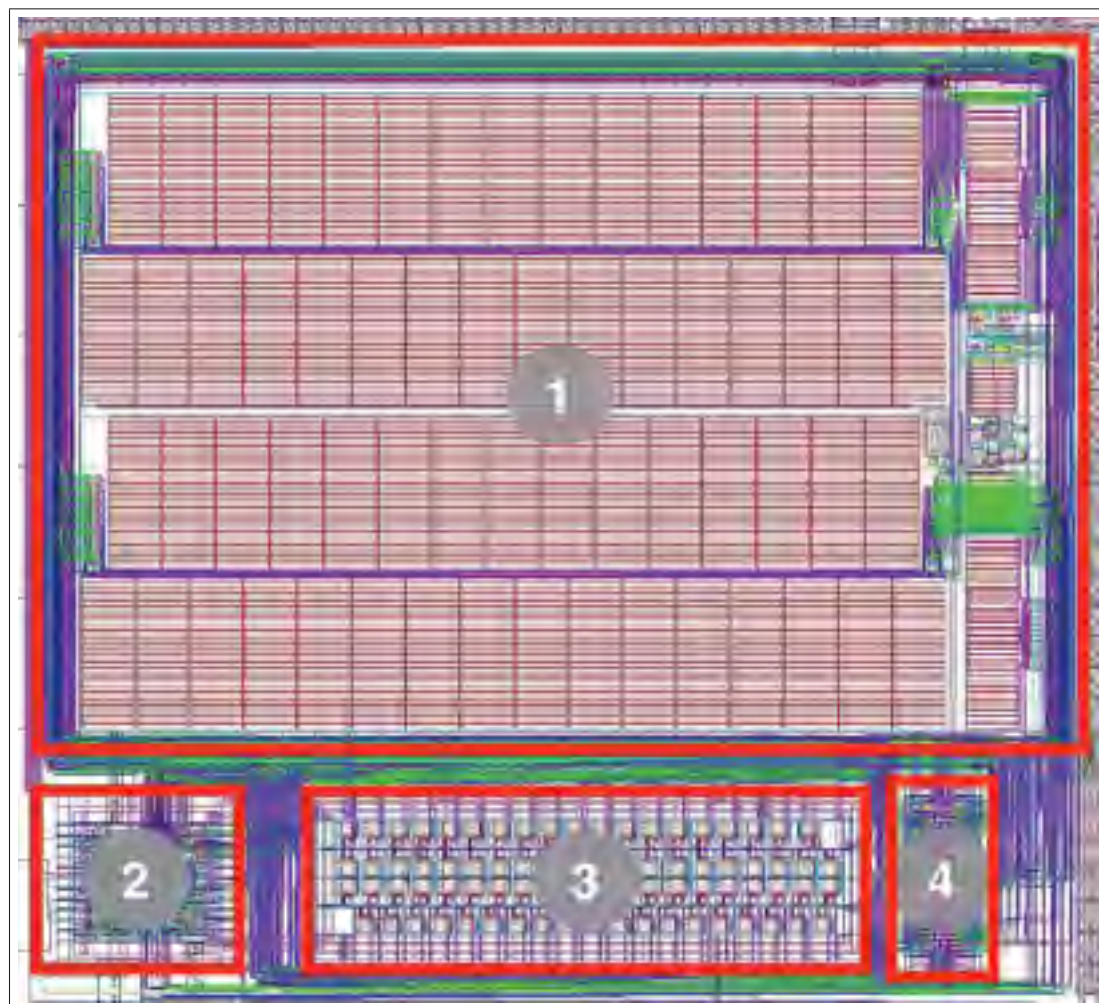


Figure 5.16 Dessin des masques de la puce sans les plots d'entrée/sortie

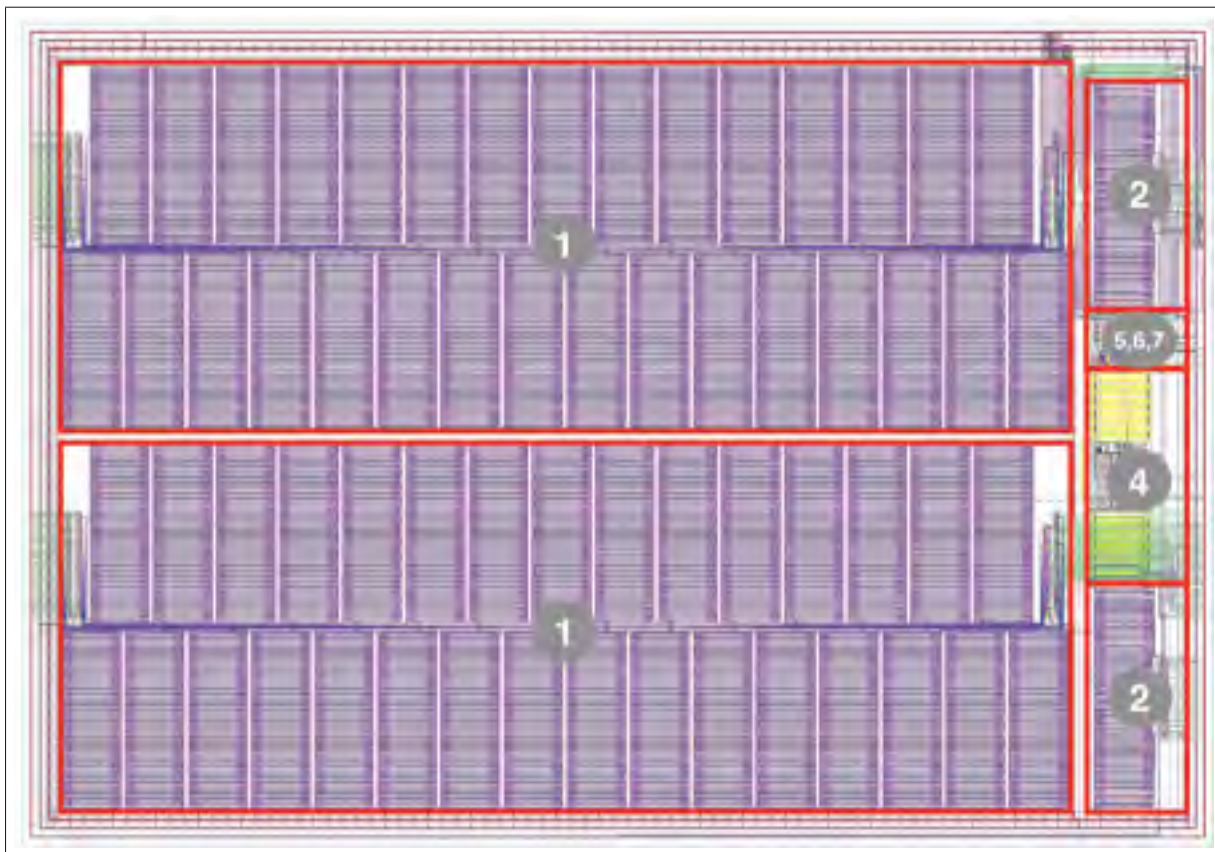


Figure 5.17 Dessin des masques de la partie analogique

## CHAPITRE 6

### TESTS ET SIMULATIONS

Dans cette partie nous allons mettre en avant les différentes simulations qui nous ont permis de valider le fonctionnement de la puce avant de lancer sa production. Nous allons, dans un premier temps, aborder les tests spécifiques à chaque sous-circuit avant d'aborder les tests globaux.

#### 6.1 Tests sur les éléments analogiques

À l'exception de quelques modifications, les éléments analogiques sont en grande partie issus des travaux antérieurs décrits dans (Favre, 2015). Toutefois, les problèmes expérimentaux rencontrés lors des tests de cette version de l'ADC ainsi que les modifications apportées sur le *dithering* m'ont forcés à modifier certains éléments et à re-tester l'ensemble des sous-circuits. Ces tests sont en grande partie des simulations temporelles visant à vérifier le comportement d'un module dans des conditions de fonctionnement nominales. La liste exhaustive de ces vérifications étant assez longue, celle-ci ne sera pas traitée ici. Le seul élément analogique que nous traiterons ici est le comparateur. En effet, l'augmentation de la résolution de l'ADC impose de vérifier la capacité du comparateur à fonctionner dans de telles conditions. C'est ce que nous vérifions ci-dessous (*cf.* figure 6.1, page 56).

Comme explicité avec la formule 5.2 (*cf.* page 36), pour garantir les performances de l'ADC, le comparateur doit être capable de discriminer des tensions séparées de  $220\mu V$ . Ces calculs sont effectués pour une tension de référence  $V_{ref}$  de 0.9V et pour une résolution de 13 bits. Cette résolution correspond à la résolution maximale que l'on attend du système en mode *dithering*. On vérifie donc que le comparateur est opérationnel en lui faisant comparer un signal carré d'amplitude égale à  $V_{LSB}$  à la tension de  $\frac{V_{DD}}{2}$ . L'observation de la simulation montre que le circuit fonctionne convenablement en faisant commuter les sorties en fonction du signe de la différence entre  $V_{inN}$  et  $V_{inP}$ . La synchronisation du comparateur sur le signal d'horloge impacte les sorties. On y retrouve en effet les oscillations du signal d'horloge lorsque la sortie



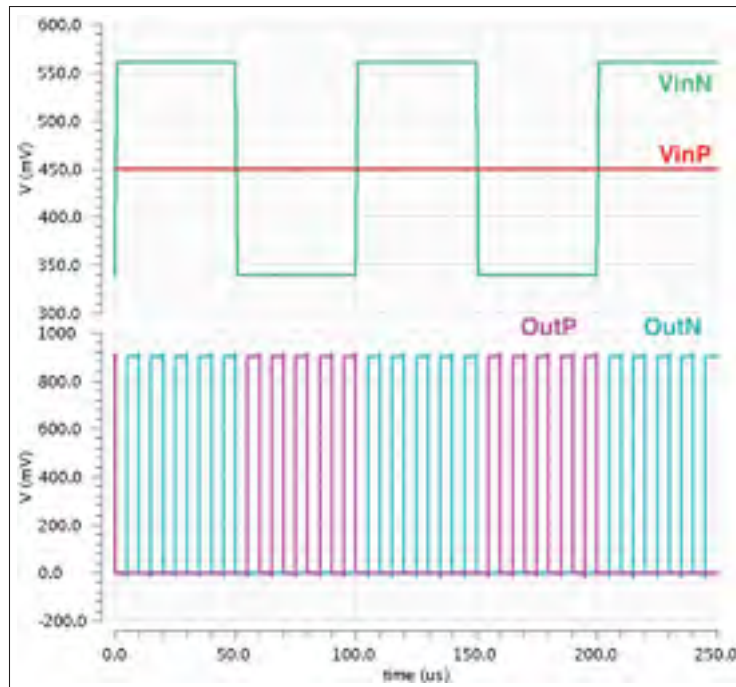


Figure 6.1 Simulation temporelle réalisée sur le comparateur

est au niveau logique haut. Il faut en être conscient afin de prélever le signal de comparaison au bon moment.

Nous allons à présent aborder les vérifications associées à la partie numérique du design.

## 6.2 Tests sur les éléments numériques

Les tests numériques se divisent en deux étapes. Débutons par les tests pré-synthèse. Ceux-ci visent à démontrer le bon fonctionnement du système au niveau fonctionnel. On vérifie ici que la description *VHDL* que l'on a réalisée fournit bien les réponses qu'on attend d'elle pour des sollicitations spécifiques. Dans un deuxième temps, les simulations post-synthèse visent à vérifier que le fonctionnement du circuit implémenté sous forme de portes logiques est identique à celui observé lors des simulations pré-synthèse. Ici encore la liste des configurations à vérifier étant assez longue nous nous contenterons de mettre en avant les spécificités des deux types de simulation (*i.e.* simulation pré et post synthèse).



### 6.2.1 Simulations pré-synthèse

Comme nous venons de le préciser, les simulations pré-synthèse sont uniquement basées sur la description en langage *VHDL* de l'architecture. Elles ne traduisent pas les propagations induites par les portes logiques mais sont nécessaires pour valider une première fois le comportement du système avant la synthèse en porte logique. Ces simulations prennent la forme de chronogrammes et sont directement effectuées sous *ISE Xilinx* à partir d'un fichier *Testbench* où l'on impose des contraintes au système. À titre d'exemple on retrouve un de ces chronogrammes en figure 6.2 (page 57).



Figure 6.2 Chronogramme pré-synthèse du cœur logique de la puce

Sans entrer dans les détails, on peut rapidement constater sur ce chronogramme qu'après chaque échantillonnage (*i.e.* passage de *sample* à '1') l'algorithme se déroule normalement avec l'incrémentement des signaux de commande des *DAC*. Cette incrémentation commence avec le *DAC* unaire (*cf.* signaux *n\_unary*[30 : 0] et *p\_unary*[30 : 0]) puis se poursuit avec le *DAC* binaire (*cf.* signaux *dacn\_binary*[4 : 0] et *dacp\_binary*[4 : 0]). Dès lors que l'algorithme *SAR* est terminé, le résultat est actualisé en sortie (*cf.*  $t = 2235ns$  en figure 6.2, page 57). Passons à présent aux simulations post-synthèse.

## 6.2.2 Simulations post-synthèse

Une fois que la synthèse a été réalisée, la simulation post-synthèse est indispensable pour valider le comportement du système au niveau matériel. Pour le cela, le simulateur (*i.e.* ici *AMS*) introduit des délais arbitraires sur les portes logiques. On peut ainsi vérifier que les signaux restent synchronisés vis-à-vis du signal d'horloge et ce malgré ces délais. Dans l'idéal, on reproduit ici les mêmes vecteurs de test que pour les simulations pré-synthèse et on s'attend à obtenir exactement les mêmes résultats.

Outre les chronogrammes post-synthèse, le compilateur *RTL* (qui transforme la description *VHDL* en circuit de portes logiques) nous donne des informations sur le circuit créé. Ces informations sont des estimations des performances de la puce en terme de consommation d'énergie, de temps de transition ou encore de surface occupée.

Considérons les résultats récoltés pour les deux éléments conçus via le flot de conception automatisé.

### 6.2.2.1 *Logic Core*

Commençons par nous intéresser aux données fournies sur l'énergie par le compilateur *RTL*. On rappelle que le circuit est destiné à fonctionner à basses fréquences avec peu d'énergie. Ainsi, dans les réglages du compilateur *RTL*, on privilégie l'optimisation des courants de fuites par rapport aux courants de commutation.

Le circuit proposé consomme une énergie liée aux courants de fuite de  $75.04nW$  alors que la consommation d'énergie liée au fonctionnement s'élève à  $15900nW$ . Ces paramètres permettront de limiter au maximum la consommation globale du bloc numérique. Il est à noter que ces valeurs sont déterminées pour la tension d'alimentation nominale de la technologie (*i.e.*  $1.2V$ ) et ne tiennent pas compte du fonctionnement réel du circuit mais de transitions arbitraires liées à la fréquence d'utilisation choisie. Notre circuit fonctionnant à  $0.9V$ , on peut

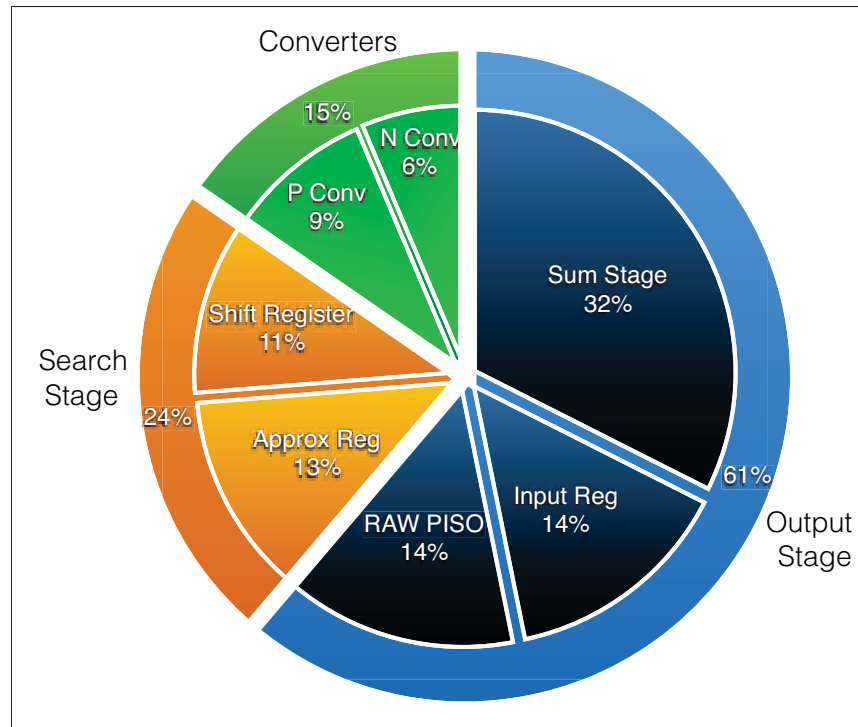


Figure 6.3 Répartition de la consommation d'énergie dans le bloc numérique

estimer sa consommation réelle à  $56.25nW$  pour les courants de fuite et  $11925nW$  pour le fonctionnement.

Les informations données par le compilateur *RTL* permettent de connaître la répartition de la consommation d'énergie du cœur logique dans les différents éléments qui la constitue (cf. figure 6.3, page 59). Ainsi on constate que l'étage de sortie est le plus consommateur d'énergie (61% du total) alors qu'au contraire les convertisseurs binaire-thermomètre présentent une consommation très faible (15% du total).

Au terme du flot de conception, on obtient sous *Encounter* l'architecture du cœur logique (cf. figure I-3, page 89, annexe I). Le cœur seul se compose de 470 cellules élémentaires qui occupent une surface de  $8100\mu m^2$ . Avec l'ajout des anneaux d'alimentation, cette surface est portée à  $16900\mu m^2$ . La surface totale disponible est de  $1mm^2$ . Ce bloque numérique n'occupe donc que 1.7% de la surface disponible.

### 6.2.2.2 DAC Driver

Ici encore, on privilégie l'optimisation des courants de fuite. La consommation d'énergie liée à ces courants s'élève à  $7.68nW$  (soit  $5.76nW$  pour une alimentation de  $0.9V$ ) alors que la consommation d'énergie liée au fonctionnement est de  $6930nW$  (soit  $5198nW$  pour une alimentation de  $0.9V$ ). Il est inutile d'analyser la répartition de cette énergie. En effet, la consommation de courant est équirépartie entre les multiples processus similaires qui composent les *DAC Drivers*.

D'un point de vue surfacique, les *DAC Drivers* sont constitués de 124 portes logiques occupant une surface de  $900\mu m^2$  (cf. figure I-4, page 89, section I). Avec les *power rings*, la surface totale est de  $3600\mu m^2$ . Les deux *DAC Drivers* utilisés occupent ainsi  $0.72\%$  du design total.

## 6.3 Simulations Globales

Sur une simulation en mode mixte (*i.e.* analogique/numérique) telle que nous l'exploitons ici, on utilise le simulateur *AMS* pour vérifier le bon interfacage des signaux reliant les parties analogique et numérique. On en profite pour mettre en place des simulations globales visant à évaluer les performances du circuit. Avant d'aborder ces simulations globales et leurs résultats nous allons définir les principaux paramètres généralement utilisés pour caractériser les *ADC*.

### 6.3.1 Paramètres caractéristiques des *ADC*

Les paramètres traditionnellement utilisés pour caractériser les *ADC* sont définis ci-après. On peut les décomposer en deux grandes parties : les paramètres statiques (*i.e.* *INL* et *DNL*) et les paramètres dynamiques (*i.e.* *SNR*, *SINAD* et *ENOB*).

#### *INL* et *DNL* :

Ces deux variables permettent d'évaluer la linéarité de l'*ADC*. Ainsi, le *DNL* (*i.e.* *Differential NonLinearity*) représente l'écart entre la valeur théorique idéale du code de sortie et sa valeur réelle alors que l'*INL* (*i.e.* *Integral NonLinearity*) correspond à la somme différentielle de

chacun des *DNLs* antérieurs au point mesuré. Ces deux mesures sont effectuées à partir des résultats expérimentaux corrigés en *offset* et en gain. Ainsi on commence par définir la courbe  $DNL(i)$  représentative de l'écart entre la fonction de transfert parfaite et le résultat corrigé obtenu :

$$DNL(i) = \text{Résultats Théoriques}(i) - \text{Résultats Corrigés}(i) \quad (6.1)$$

avec :

$$\text{Résultats Corrigés}(i) = \frac{\text{Résultats Bruts}(i)}{\text{gain}} + \text{offset} \quad (6.2)$$

Le *DNL* que l'on retient est alors défini comme le maximum de  $DNL(i)$  :

$$DNL = \max(\text{abs}(DNL(i))) \quad (6.3)$$

Et l'*INL* comme la somme des *DNLs* précédemment calculés :

$$INL(i) = \sum_{j=1}^i DNL(j) \quad (6.4)$$

Pour ces deux variables, on retient également la valeur maximale obtenue. Pour les calculer, on utilise une entrée de type rampe avec laquelle on balaye l'ensemble de la plage de fonctionnement d'entrée du convertisseur analogique numérique. L'ensemble des codes de sortie obtenus constitue *Résultats Bruts*(*i*).

### ***SNR* et *SINAD* :**

Ces deux autres mesures font appel à des notions de puissance. Il s'agit ici de déterminer le ratio entre le signal utile et le signal complet. Dans le calcul du *SNR* (*i.e. Signal Noise Ratio*), on prend en compte le signal et le bruit alors que dans le calcul *SINAD* (*i.e. Signal-to-Noise And Distortion ratio*), on inclut également la notion de distorsion. Ces deux valeurs caractéristiques sont calculées à partir du signal de sortie obtenu pour une entrée sinusoïdale.

$$SNR = \frac{\text{Full Signal}}{\text{Noise}} \quad \text{et} \quad SINAD = \frac{\text{Full Signal}}{\text{Noise} + \text{Distortion}} \quad (6.5)$$

***ENOB :***

C'est sans doute le paramètre le plus important concernant les *ADC*. Celui-ci caractérise le nombre de bits utilisables (*i.e.* Effective Number Of Bits). Cette valeur n'est pas identique à la résolution totale de l'*ADC* et est calculée à partir de la valeur du *SINAD* via la formule suivante :

$$ENOB = \frac{SINAD - 1.76}{6.02} \quad (6.6)$$

***Figure of Merit (FoM) :***

Cette dernière valeur caractéristique représente la consommation d'énergie pour une conversion. Celle-ci est calculée à partir de la puissance totale, de la fréquence d'échantillonnage  $f_S$  et de la valeur de l'*ENOB* :

$$FoM = \frac{Power\ Consumption}{f_S \cdot 2^{ENOB}} \quad (6.7)$$

### 6.3.2 Résultats

Cette section s'attache au fonctionnement général de l'ADC et aux valeurs des paramètres caractéristiques précédemment décrits. Ces paramètres ont été déterminés pour le fonctionnement standard de l'ADC et partiellement pour le fonctionnement en mode *dithering*. Le temps de calcul des paramètres statiques en mode *dithering* serait bien trop long. Néanmoins, des vérifications ont été effectuées sur quelques conversions pour valider le fonctionnement de ce mode particulier de conversion. Nous allons à présent mettre en avant les résultats obtenus pour le mode standard.

À titre d'exemple, la figure 6.4 (page 63) propose un fonctionnement typique de l'ADC et permet de constater que la convergence du SAR s'effectue bien comme on l'attend.

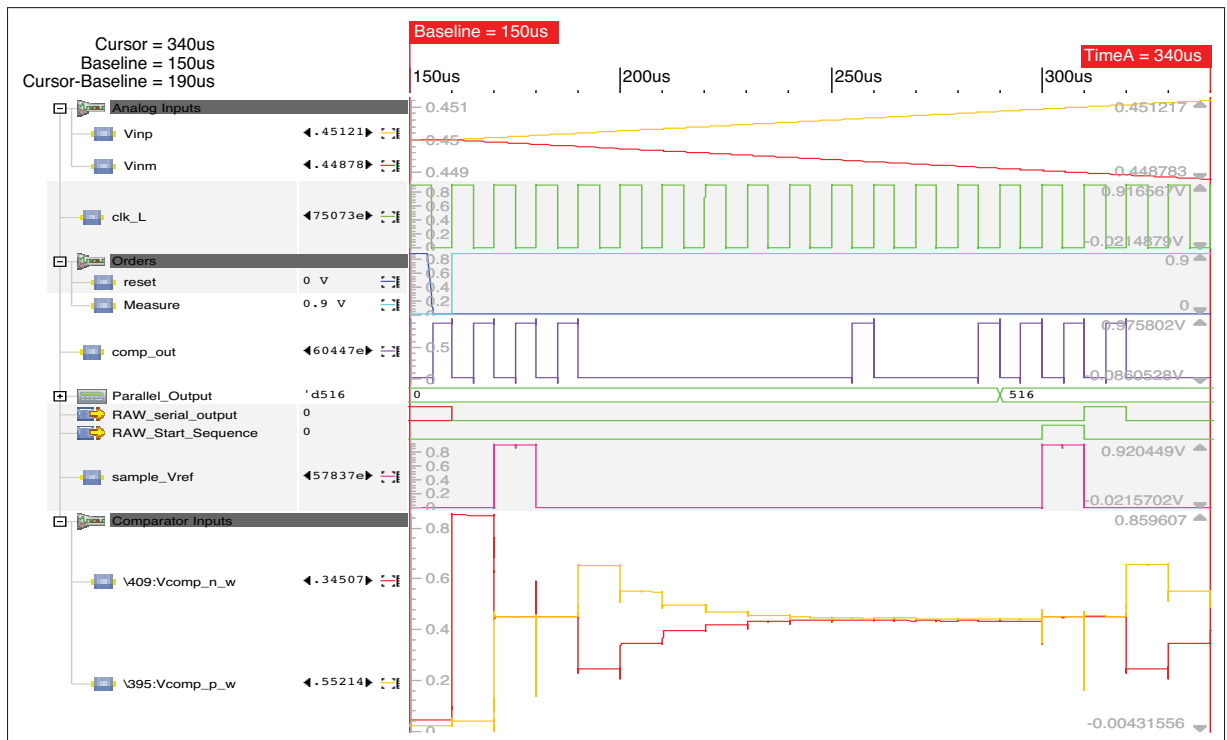


Figure 6.4 Chronogramme typique d'une conversion (données : simulateur AMS)

On retrouve bien l'enchaînement des différentes étapes de l'algorithme *SAR*. Une fois la phase de *reset* terminée et l'activation du signal *measure*, qui ordonne une conversion à l'*ADC*, l'échantillonnage s'effectue avec le signal *sample\_Vref* (cf.  $t = 170\mu s$  à  $t = 180\mu s$ ). On observe ensuite la convergence des *DAC* sur les tensions d'entrées du comparateur (cf. signaux  $V_{comp\_p\_w}$  et  $V_{comp\_n\_w}$ ). Lorsque l'algorithme est terminé, le résultat est actualisé en sortie (cf.  $t = 290\mu s$ ). Dès le début de la conversion suivante, on traduit le dernier résultat obtenu de façon sérielle. *RAW\_Start\_Sequence* indique le début de la séquence alors que *RAW\_serial\_output* constitue la séquence de données sérielles à proprement parlé.

Ainsi, l'enchaînement des actions est bien conforme à nos attentes et le principe de fonctionnement du *SAR* est bien respecté. Outre les multiples vérifications possibles via les chronogrammes évalués sous *AMS*, il est indispensable de déterminer les paramètres définis en section 6.3.1 (page 60). Pour cela, on traite les données obtenues via *AMS* sous *Matlab*.

Commençons par les paramètres statiques de l'*ADC*. À partir des codes de sortie issus d'une entrée rampe, on détermine le *DNL* et l'*INL*. Pour cela on calcule les écarts avec les résultats idéaux comme il est précisé précédemment (cf. section 6.3.1, page 60). Le *DNL* s'établit ainsi à la valeur de  $0.59LSB$  et l'*INL* maximal enregistré est de  $2.3LSB$ .

Les paramètres dynamiques utilisent les codes de sortie d'une entrée sinusoïdale. Cette courbe de sortie est représentée en figure 6.5, page 65. Grâce à *Matlab* et sa fonction *snr(...)* on trace la représentation fréquentielle du sinus de sortie. La fonction nous donne un *SNR* de  $59.19dB$ <sup>1</sup>. Outre cette valeur, le tracé nous confirme bien que la puissance du bruit est bien plus faible que celle du fondamental mais aussi que celui-ci est nettement plus important que les harmoniques. Ceci est de bonne augure pour nos prochains calculs car on constate ici que la distorsion et le bruit sont relativement faibles face au signal utile.

Le *SINAD* est calculé avec la fonction *sinad(...)* de *Matlab*. Différentes simulations ont été effectuées avec des amplitudes et des fréquences différentes pour le sinus d'entrée. Il en résulte un *SINAD* variant de  $59.1dB$  à  $61.9dB$  et donc un *ENOB* allant de  $9.52$  bits à  $9.99$  bits par

---

1. *N.B.* Ces puissances sont fictives car calculées à partir du code décimal de sortie



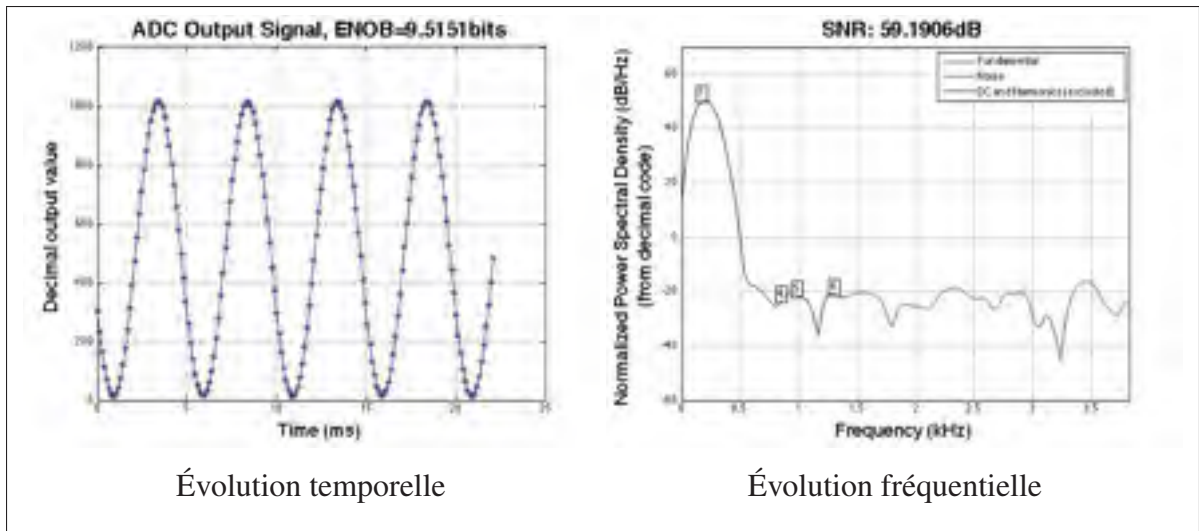


Figure 6.5 Résultats de simulation en mode natif avec  
 $f_{CLK} = 100kHz$  et  $f_{IN} = 200Hz$

application de la formule (6.6), page 62. En moyenne, le *SINAD* semble s'établir à 59.9dB et donc l'*ENOB* à 9.65 bits.

Terminons cette présentation des résultats par les aspects énergétiques. L'utilisation du flot automatisé de conception pour les parties numériques rend impossible le calcul de la consommation de ces éléments car le fabricant cache ces données. En réalité, nous n'avons pas accès aux dessins des masques, aux temps de transitions ou à la consommation énergétique des cellules proposées. Néanmoins, il reste possible d'évaluer la consommation globale de la puce en combinant les résultats du compilateur *RTL* (cf. section 6.2.2, page 58) et la consommation de la partie analogique.

Sous le simulateur *AMS*, la consommation liée à la partie analogique (amplificateurs opérationnels exclus) s'établit à  $0.133\mu W$ . Pour rappel, la consommation du cœur logique s'élevait, pour une tension d'alimentation de 0.9V, à  $11925nW$  et celle d'un *DAC Drivers* à  $5198nW$ . Par application de la formule ci-dessous, on aboutit à une consommation globale de  $22.5\mu W$ . À titre de comparaison, la consommation de la version 2015 de l'*ADC* s'établissait à  $90\mu W$ .

$$P_{Chip} = P_{Logic Core} + 2 \cdot P_{DAC Drivers} + P_{Analog Blocks} \quad (6.8)$$

L'amélioration obtenue ici (75%) est à considérer avec du recul car il est fort possible que cette consommation soit très différente lors des tests sur la puce réelle. En effet, cette estimation utilise des transitions arbitraires sur les éléments numériques et ne tient pas compte de la tension d'alimentation qui leur sera réellement appliquée mais considère une tension de 0.9V pour toute la puce. La tension minimale de fonctionnement ne sera connue que lors des tests réels et donc la consommation aussi.

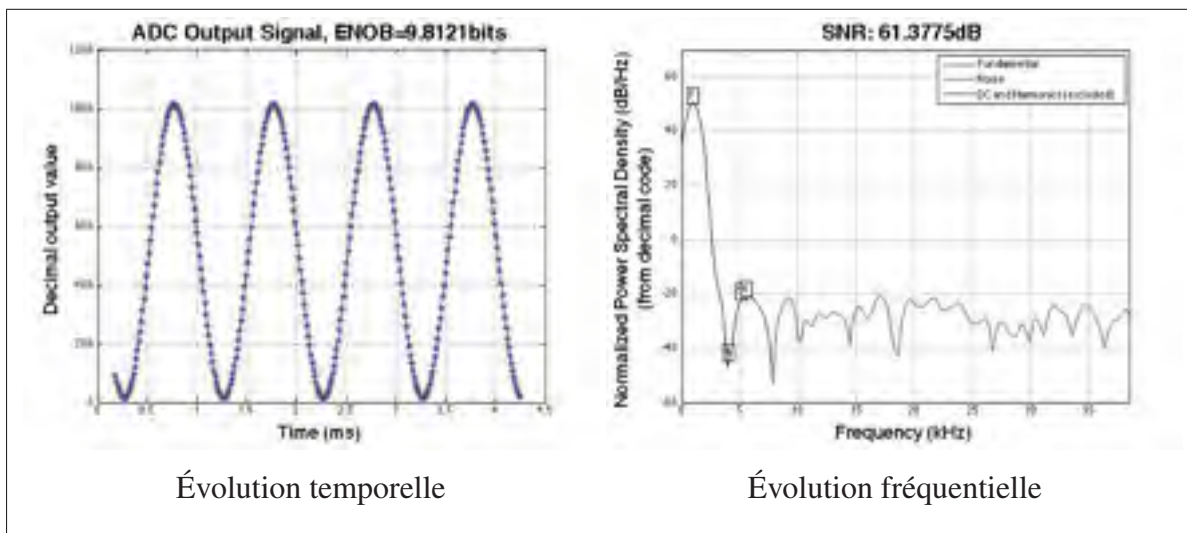


Figure 6.6 Résultats de simulation en mode natif avec  $f_{CLK} = 1MHz$  et  $f_{IN} = 1kHz$

Si on considère une consommation de  $22.5\mu W$ , une fréquence d'échantillonnage de  $8kS \cdot s^{-1}$  et un  $ENOB$  de 9.65 bits, on aboutit à une  $FoM$  de  $2.73pJ/conv$ . Cette valeur est supérieure à l'état de l'art mais ne peut pas être considérée comme fiable si on considère la méthode de calcul de la consommation utilisée. Néanmoins, les dernières simulations réalisées montrent qu'il est possible d'augmenter la fréquence d'échantillonnage à  $83kS \cdot s^{-1}$  (cf. figure 6.6, page 66) avec une consommation portée à  $23.6\mu W$ . L' $ENOB$  est alors de 9.88 bits ce qui abaisserait la  $FoM$  à  $0.30pJ/conv$ .

L'article présenté à la conférence *ISCAS 2017* (Sauvé *et al.*, 2017) annonce une consommation de  $4\mu W$ . Cette consommation est uniquement associée aux processus participant à la re-

cherche du résultat. Le processus de traitement des données de sortie est en effet exclu de ce bilan de puissance car il n'est pas l'objet de l'étude. La consommation totale présentée dans l'article est basée sur une estimation de la consommation de chaque sous-processus en fonction des tensions d'alimentation et des fréquences auxquelles ils sont sollicités (ces données sont présentées dans le tableau 6.1 en page 67). Ainsi, on attribue au processus de recherche une consommation de  $2900nW$ . Les convertisseurs binaire-thermomètre et les *DAC drivers* n'étant utilisés qu'en fonction des commutations des condensateurs, on diminue leur consommation d'un facteur  $\frac{1}{13}$  (ce facteur correspond aux 13 cycles d'horloge nécessaires à une conversion). On se permet cette minoration car la consommation de courant liée aux courants de fuite est ici très minoritaire, donc réduire la fréquence d'utilisation équivaut à réduire d'autant les courants de commutation et donc la consommation. On aboutit à une consommation de  $3.8\mu W$  à laquelle on ajoute  $1.23\mu W$  pour la partie analogique. Pour finir, si on considère une sous-alimentation du cœur logique (*i.e.* SAR et convertisseurs) à  $0.6V$  on aboutit à une consommation numérique de  $2.8\mu W$ . On retient alors  $4\mu W$  pour la consommation totale.

Tableau 6.1 Évaluation de la consommation des circuits logiques

<b>Digital Process</b>	<b>1.2V Data (Encounter)</b>	<b>1<sup>st</sup> reduction (V<sub>DD</sub> = 0.9V)</b>	<b>2<sup>nd</sup> reduction (fréquentielle)</b>	<b>3<sup>rd</sup> reduction (V<sub>DD</sub> = 0.9V/0.6V)</b>
<b>Converters</b>	2385nW	1789nW	138nW	92nW
<b>SAR</b>	3816nW	2862nW	2862nW	1908nW
<b>DAC Drivers</b>	13860nW	10395nW	800nW	800nW

Concernant le *dithering*, une simulation d'entrée rampe permet de vérifier le bon fonctionnement de l'algorithme sur quelques codes (*cf.* figure 6.7, page 68). Même si elle souffre de quelques irrégularités, cette courbe permet de vérifier la monotonie du système et le bon enchaînement des codes de sortie. On remarque notamment une modification du gain du convertisseur à l'approche de la tension d'entrée différentielle nulle. Cette variation est un problème commun qui peut être corrigé en utilisant la technique décrite dans (Taherzadeh-Sani *et al.*, 2014). Pour cette entrée rampe on peut également déterminer la tension moyenne d'un *LSB*. Celle-ci est de  $0.20mV$  pour une valeur théorique de  $0.22mV$  soit un écart relatif de 9%. Même

si ce résultat n'est pas parfait, il démontre que l'amélioration de la résolution est effective. Les simulations dynamiques tendent également dans ce sens.

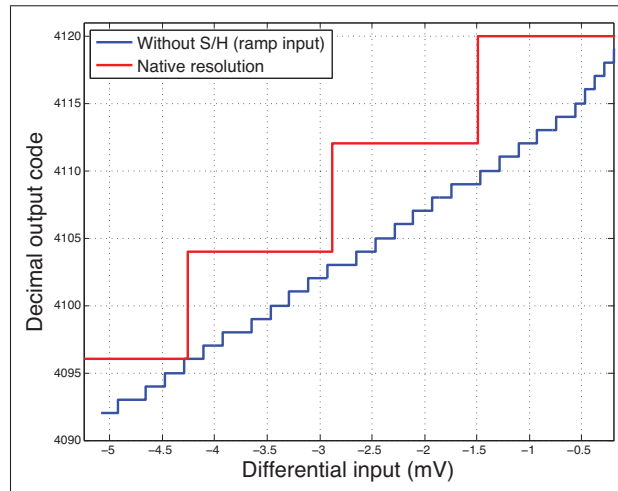


Figure 6.7 Portion de la fonction de transfert de l'ADC en résolution native et en mode *dithering* (données : simulateur AMS)

Moyennant des temps de calculs très longs (plusieurs semaines), une estimation du comportement dynamique du convertisseur a été calculée. Avec une entrée sinusoïdale de fréquence  $50\text{Hz}$  et une horloge cadencée à  $1\text{MHz}$ , on obtient les résultats présentés en figures 6.8 (page 69). Ces simulations ont été effectuées en fractionnant la sinusoïde en différents tronçons qui ont ensuite été rassemblés. Cette méthode a l'avantage d'être plus rapide qu'une simulation continue mais entraîne également une dégradation de l'*ENOB*. En effet, le résultat d'une conversion dépend bien évidemment des tensions d'entrées mais également des tensions résiduelles présentes dans les *DAC* à l'issue de la conversion précédente. De ce fait, le résultat d'une conversion n'est pas tout à fait le même si le convertisseur vient d'être activé (comme c'est le cas en simulation fractionnée) ou si il convertit de manière continue. Bien entendu, on s'arrange pour rassembler les courbes en des points où elles présentent de bonnes corrélations mais le résultat reste imparfait. Il en résulte de légers problèmes de distorsion qui viennent dégrader l'*ENOB*.

Dans cette configuration, si on considère la formule usuelle de l'*ENOB*, basée sur le *SINAD*, l'*ENOB* s'élève à une valeur de 11.4 bits. Si on exclut l'aspect distorsion en considérant le *SNR* au lieu du *SINAD* dans la formule de l'*ENOB*, la valeur de celui-ci passe à 11.6 bits dans le meilleur des cas.

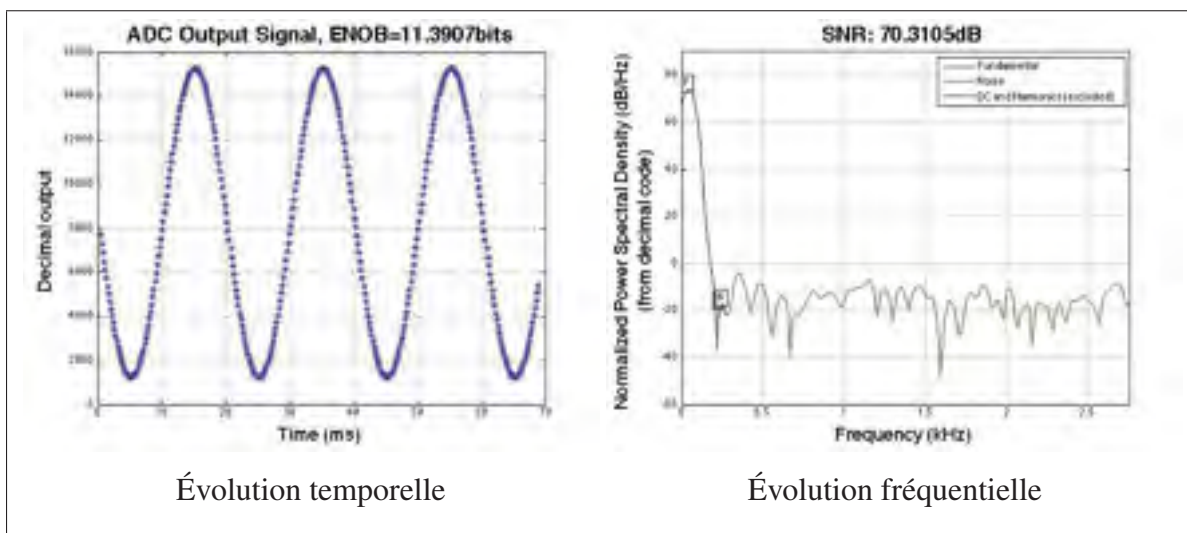


Figure 6.8 Résultats de simulation en mode *dithering* avec  $f_{CLK} = 1MHz$  et  $f_{IN} = 50Hz$

Ces résultats ne sont pas aussi bons que les 13 bits théoriques mais mettent tout de même en évidence l'augmentation de la résolution. Par cette méthode, la résolution initiale de 9.88 bits a été portée à 11.4 bits soit une augmentation de 15.4%. L'hypothèse la plus probable est que les défauts de la puce existant en mode natif sont amplifiés, ou a minima conservés, en mode *dithering*. En effet, le calcul du résultat en mode *dithering* étant directement basé sur les valeurs obtenues en résolution native, il est impossible d'atteindre la résolution maximale si les résultats natifs ne sont pas parfaits. On met ici en avant une des premières limites du *dithering*. Nous en étudierons une autre au chapitre 8.

L'ensemble des résultats de simulation sont regroupés dans le tableau 6.2 (page 70). Ces données utilisent la puissance de  $4\mu W$  calculée pour l'article *ISCAS*. On constate que la *FoM* en mode *dithering* est fortement dégradée. Ce problème n'est pas entièrement lié au fonctionne-

ment du convertisseur mais en partie au concept même du *dithering*. L'explication sera fournie au chapitre 8 (page 77).

La complexité des calculs en mode *dithering* rend difficile l'obtention de résultats chiffrés fiables (notamment pour l'*INL* et le *DNL*). Par ailleurs, l'utilisation de cellules logiques standards de la technologie *CMOS* 130nm rend complexe le calcul d'un bilan de puissance fiable et complexifie l'optimisation du système. Néanmoins, ces simulations prouvent l'amélioration de la résolution grâce au *dithering*. Passons à présent à l'étude expérimentale réalisée sur la puce.

Tableau 6.2 Résumé des performances simulées de l'ADC

	Mode natif	Mode dithering
<b>Technologie</b>	<i>CMOS</i> 130nm	
<b>Alimentation (mV)</b>	900 <sub>(analogique)</sub> / 600 <sub>(numérique)</sub>	
<b>Power (<math>\mu</math>W)</b>	4	
<b>F<sub>sample</sub> (kS.s<sup>-1</sup>)</b>	83	5.5
<b><i>DNL</i> (LSB)</b>	0.5	NA
<b><i>INL</i> (LSB)</b>	2	NA
<b><i>SNR</i> (dB)</b>	61.4	70.3
<b><i>SINAD</i> (dB)</b>	61.2	70.3
<b><i>ENOB</i> (bits)</b>	9.88	11.4
<b><i>FoM</i> (fJ / conv step)</b>	53	269

## CHAPITRE 7

### EXPÉRIMENTATIONS ET TESTS DE LA PUCE

#### 7.1 Banc de test

À l'issue de l'étape de fabrication, dix *die* ont pris place dans des boîtiers *QFN* de 32 pins. Une vue au microscope de la puce nue est disponible en figure 7.1 (page 71). On y distingue la couche de métallisation supérieure (nommée *MA* pour notre technologie) ainsi que les 32 plots d'entrée/sortie.

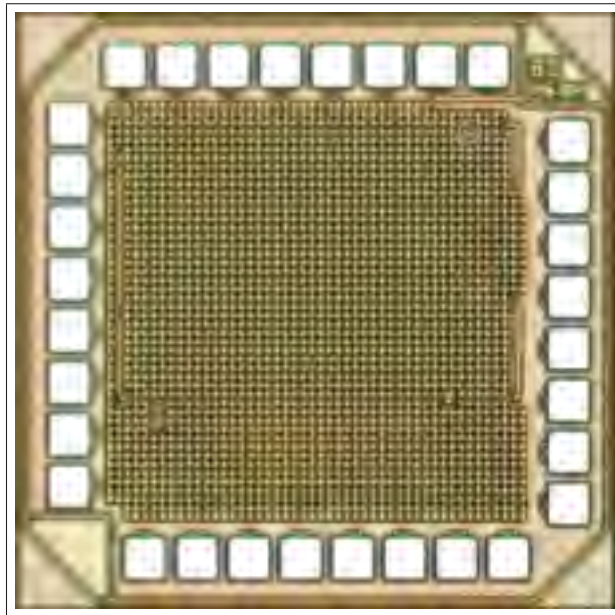


Figure 7.1 Vue au microscope d'un *die*

Pour faciliter la mise en œuvre des tests, j'ai eu recours à une *socket* qui, soudée au *PCB* de test (cf. figure 7.2, page 72), m'a permis d'interchanger plus facilement les circuits à tester. Afin de maximiser les résultats expérimentaux, les tensions d'alimentation (*i.e.*  $V_{DD\ HIGH}$ ,  $V_{DD\ LOW}$  et  $V_{DD\ 1.2V}$ ) et de référence (*i.e.*  $V_{Refp}$  et  $V_{CM}$ ) sont fournies par des régulateurs de tension. La tension  $V_{Refm}$  étant nulle, on se limite à l'élimination d'éventuels parasites grâce à des



capacités de découplage. Tous ces régulateurs de tension sont connectés à la même tension d'alimentation.



Figure 7.2 Premier *PCB* de test de la nouvelle puce

L'entrée différentielle est fournie par un générateur de signaux à deux voies. Ceci permet de générer les deux sinus en opposition de phase, nécessaires à l'obtention des paramètres dynamiques.

Le premier *PCB* de test présentant divers problèmes, notamment liés au bruit, une deuxième carte de test a été fabriquée (*cf.* figure 7.3, page 73). L'absence de plan de masse sur la première carte entraînait la formation d'oscillations parasites sur les différents signaux (en particulier l'horloge). Le deuxième *PCB* est plus simple. Un plan de masse est mis en place sur l'ensemble de la carte et les régulateurs de tension sont remplacés par des trio de capacités de découplages. De plus, la *socket* est remplacée par une soudure directe de la puce sur le *PCB* (celle-ci n'ayant pu être convenablement désoudée de la première carte de test).



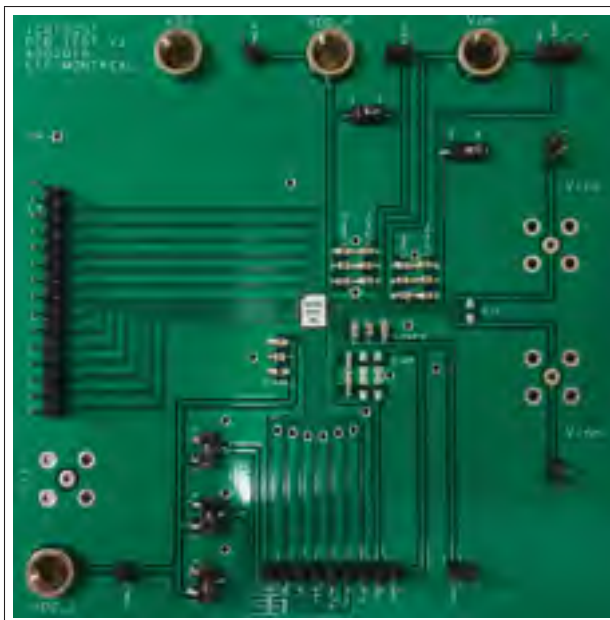


Figure 7.3 Deuxième *PCB* de test de la nouvelle puce

Pour finir, on place, en parallèle des sorties des amplificateurs opérationnels permettant de mesurer les tensions générées par les deux *DAC*, des capacités de  $6.8\text{pF}$  afin de stabiliser leur sortie.

## 7.2 Résultats expérimentaux

L'ensemble des tests en laboratoire ont permis de tester neuf puces différentes. Les résultats expérimentaux associés à ces tests ont confirmé le fonctionnement de la puce d'un point de vue logique. En effet, les conversions s'enchaînent bien à la fréquence voulue et les différents points de contrôle agissent bien comme prévu (*reset*, *dithering*, activation du comparateur, etc.). De plus, en mode *dithering*, le calcul du résultat final se fait bien comme décrit en partie 4.1 (page 21). La vérification est possible grâce à la sortie sérielle qui donne la valeur de chacune des conversions constitutives d'une séquence de *dithering*. Une vérification du fonctionnement analogique du *dithering* est en revanche impossible, l'oscilloscope n'étant pas en mesure de détecter les tensions de biais ajoutées au signal. Ces expériences ont aussi révélé la présence d'un problème majeur dans la détermination des résultats de sortie. Ce problème

semble être d'origine analogique, le comportement de la partie logique étant pour sa part tout à fait cohérent.

La figure 7.4 (page 74) illustre bien le problème. On constate que la sinusoïde évolue par palier comme si les *LSBs* étaient inopérants. De plus on constate un phénomène de saturation qui a pour effet de tronquer la sinusoïde. La faible valeur de l'*ENOB*, mesuré ici à 4.18 bits en mode natif, est bien en deçà des attentes. D'un point de vue spectral, on retrouve bien le fondamental à la fréquence de la sinusoïde d'entrée mais le bruit est important sur le reste du spectre. L'*ENOB* est, en moyenne, égal à 3.82 bits en mode natif et 4.04 bits en mode *dithering*. Ces moyennes sont basées sur 32 configurations différentes (tests de plusieurs puces, carte de test, fréquences d'utilisation et tensions d'alimentation).

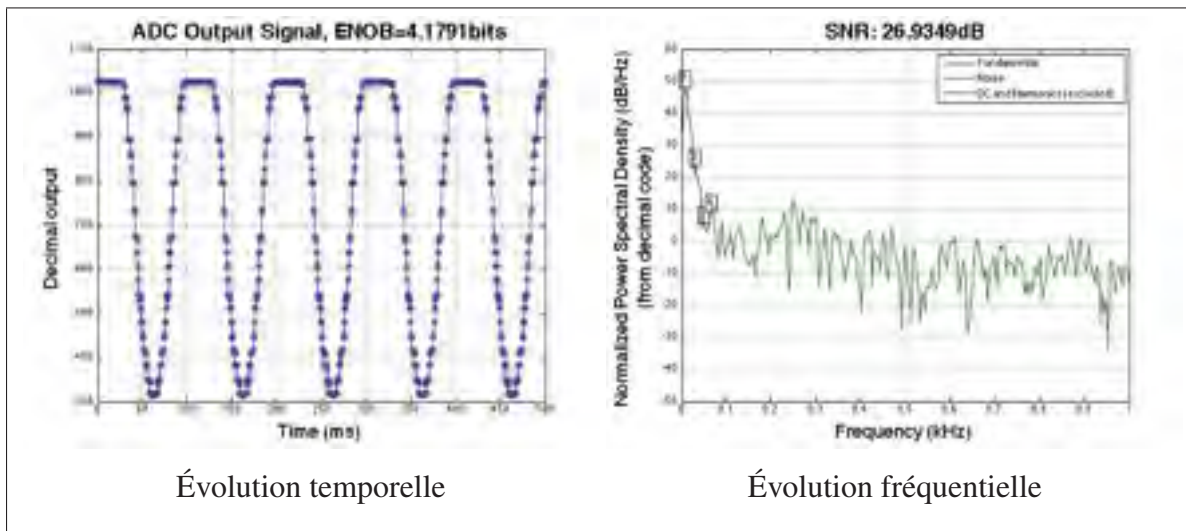


Figure 7.4 Résultats expérimentaux en mode natif avec  
 $f_{CLK} = 1MHz$  et  $f_{IN} = 10Hz$

Les mesures de puissance tendent également vers un dysfonctionnement majeur de la puce. En effet, la puissance mesurée ici est de  $29\mu W$  avec une alimentation haute de  $900mV$  et une alimentation basse de  $600mV$ . Le circuit alimenté à  $0.6V$  (*i.e.* le cœur logique) consomme  $10.4\mu W$  alors que le reste de la puce (alimenté à  $0.9V$ ) consomme  $18.7\mu W$ . Si on reprend les données *Encounter*, la consommation du cœur logique était de  $15900nW$  à  $0.9V$  soit  $7950nW$

à 0.6V. Ces valeurs sont différentes mais restent assez proches (écart relatif de 31%). Sur la partie de la puce alimentée à 0.9V, les simulations prévoyaient une consommation de 11625nW (5198nW pour chaque *DAC Driver* et 1230nW pour la partie analogique). L'écart relatif est ici de 61% ce qui soutient l'hypothèse que le problème est d'ordre analogique. On notera que si on exclu tout problème sur la partie numérique, il faut alors remettre en question l'approximation liée à la fréquence d'utilisation utilisée en partie 6.3.2 (cf. tableau 6.1, page 67). En effet, les résultats présentés ici sont relativement proches alors qu'ils tiennent uniquement compte de la tension d'alimentation et non de la fréquence de commutation des signaux <sup>1</sup>.

Cherchons à présent l'origine de ce problème. L'implémentation de pins de test sur les entrées/sortie du comparateur a permis de valider le bon fonctionnement de celui-ci mais a aussi mis en évidence un dysfonctionnement majeur du *DAC* négatif (cf. figure 7.5, page 75). On rappelle qu'en activant les amplificateurs opérationnels placés sur les entrées du comparateur nous pouvons mesurer les tensions générées par les *DAC*.

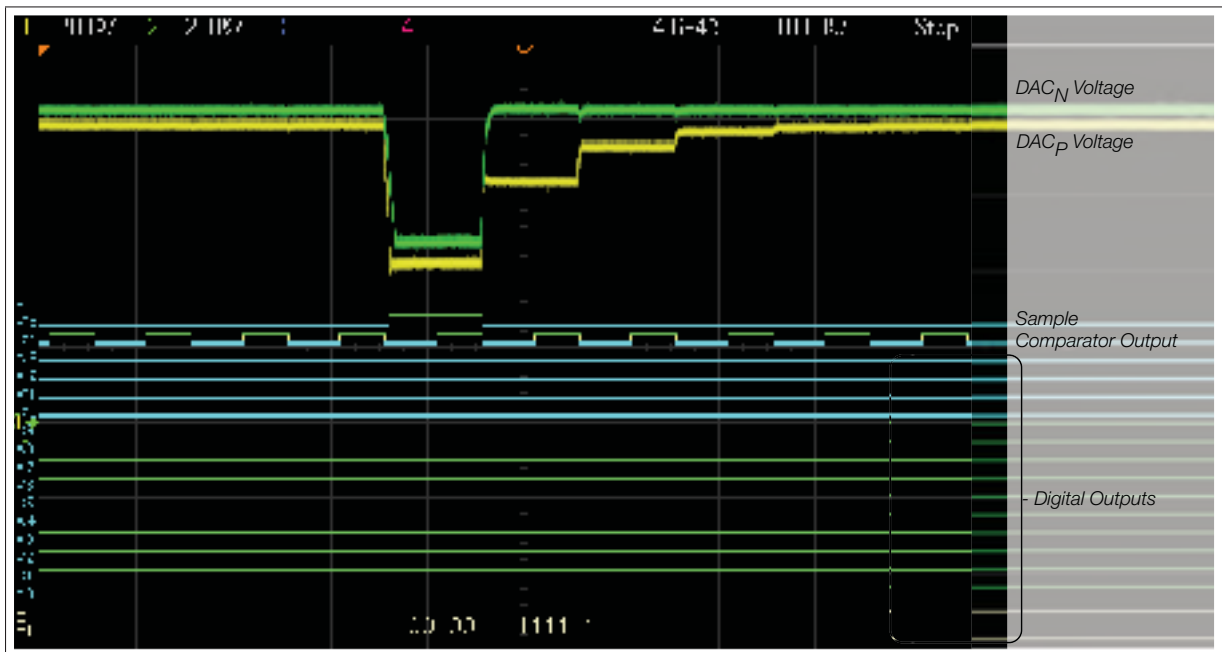


Figure 7.5 Évolution des tensions générées par les *DAC* lors une conversion

1. On utilise ici les données *Encounter* simplement transposées de 1.2V à 0.9V/0.6V

La conversion représentée ici a été effectuée avec une horloge cadencée à 100kHz et pour des tensions d'entrées constantes et identiques (*i.e.*  $V_{INp} = V_{INn}$ ). On visualise bien le comportement anormal du convertisseur sur cet enregistrement. En effet, à l'instar de la simulation représentée en figure 6.4 (page 63), les tensions des deux *DAC* devraient converger l'une vers l'autre à l'issue de l'étape d'échantillonnage. Par ailleurs, un problème est présent dès l'étape d'échantillonnage. En effet, les deux tensions d'entrée étant identiques les tensions des *DAC* devraient être identiques pendant l'étape d'échantillonnage.

L'existence d'un problème sur la partie analogique du convertisseur est corroborée par la mesure de la consommation du circuit au repos (*i.e.* avec le signal *reset* actif). En effet, cette consommation s'élève à  $26\mu W$  alors que la consommation liée aux courants de fuites estimée par le compilateur *RTL* pour la partie numérique est de l'ordre de la centaine de nanowatt.

La cause exacte de ce problème reste difficile à déterminer. L'absence d'erreur de *DRC* ou de *LVS* ainsi que la validité des résultats de simulation excluent, a priori, un oubli de connexion entre deux éléments ou une interversion de deux connections. Nous sommes donc en présence d'une erreur qui s'est présentée lors de la fabrication de la puce et qui n'est pas prise en compte par les modèles du simulateur. Pour expliquer ce problème, on peut supposer qu'une trace de métal soit sous dimensionnée et ne supporterait pas l'intensité du courant qu'elle est supposé convoyer. Néanmoins, la probabilité que cette hypothèse soit confirmée reste limitée compte tenu des faibles puissances mises en jeu sur ce projet.

## CHAPITRE 8

### BILAN SUR LE FONCTIONNEMENT DU CONVERTISSEUR

#### 8.1 Considérations sur les limites du *dithering*

Revenons à des considérations plus théoriques. On rappelle que le *dithering* permet d'augmenter la résolution native du convertisseur au prix d'un temps de conversion plus long. Dans notre cas, la résolution est théoriquement améliorée de 3 bits (passant de 10 bits à 13 bits) alors que le temps de conversion est multiplié par 15. En considérant la formule de la figure de mérite (cf. formule (6.7), page 62), on peut évaluer l'impact du *dithering* sur ce paramètre dynamique (cf. formule (8.1), page 77). On rappelle que ce paramètre illustre la consommation d'énergie nécessaire pour réaliser une conversion.

$$\frac{FoM_{Dithering}}{FoM_{Native}} = \frac{Power\ Consumption}{\frac{1}{15} \cdot f_s \cdot 2^{13}} \cdot \frac{f_s \cdot 2^{10}}{Power\ Consumption} = 15 \cdot \frac{2^{10}}{2^{13}} = \frac{15}{8} \simeq 1.9 \quad (8.1)$$

L'augmentation de la résolution améliore la *FoM* d'un facteur  $\frac{2^{10}}{2^{13}}$  soit  $\frac{1}{8}$ . À l'inverse, la dégradation de la fréquence de conversion augmente la *FoM* d'un facteur 15. Si on considère que la consommation de puissance est identique entre les modes natif et *dithering* (ce qui est à peu de chose près le cas), on aboutit à une dégradation de la figure de mérite d'un facteur  $\frac{15}{8}$  soit environ 1.9.

Ce résultat souligne l'importance d'optimiser la séquence des condensateurs utilisés pour le *dithering* afin de maximiser la résolution du convertisseur comme expliqué en partie 4.1 (page 21). Sur la version 2015 de l'ADC la résolution visée en mode *dithering* était de 12 bits mais en utilisant une séquence de condensateurs inadaptée. La *FoM* n'était alors pas dégradée d'un facteur 1.9 mais d'un facteur 3.8. On visualise bien ici tout l'impact du choix des condensateurs de biais dans les performances finales du convertisseur en mode *dithering*.

Au regard de ces résultats, il est possible de modifier une nouvelle fois la séquence des condensateurs utilisés pour le *dithering* dans le but d'optimiser la *FoM*. En effet, la partie 4.1 (page

21) met en évidence le lien entre la taille des condensateurs utilisés et la résolution maximale accessible. La conclusion de cette partie étant que la tension parasite maximale générée par la séquence doit au minimum être égale à  $V_{LSB}$  si on veut s'affranchir des problèmes de paliers présentés en figure 4.2 (page 23). Il est toutefois possible d'éliminer ces paliers en augmentant le nombre de condensateurs de la séquence. Ainsi, comme l'illustre le tableau ci-dessous (cf. tableau 8.1, page 78), à chaque fois que l'on divise la plus petite des capacités de biais par deux, on doit ajouter un condensateur supplémentaire à la séquence et donc multiplier le nombre de combinaisons de cette séquence (et donc le nombre de conversions) par deux pour conserver la linéarité du convertisseur.

Tableau 8.1 Lien entre la séquence de capacités utilisées et la résolution maximale accessible

Capacités de la séquence	Nombre de combinaisons	Résolution maximale
$2 \cdot Cu$	2	11
$Cu, 2 \cdot Cu$	6	12
$\frac{1}{2} \cdot Cu, Cu, 2 \cdot Cu$	14	13
$\frac{1}{4} \cdot Cu, \frac{1}{2} \cdot Cu, Cu, 2 \cdot Cu$	30	14
$\vdots \quad \quad \quad \vdots$	$\vdots$	$\vdots$

Ainsi, pour un nombre donné de condensateurs, il existe une capacité minimale garantissant la linéarité. La séquence utilisée dans ce projet utilisait 3 condensateurs de taille  $\frac{1}{2} \cdot Cu$ ,  $Cu$  et  $2 \cdot Cu$  pour atteindre une résolution théorique de 13 bits. On rappelle que la séquence de biais produite sur le fonctionnement du convertisseur est alors  $\pm \frac{V_{LSB}}{8}, \pm \frac{V_{LSB}}{4}, \pm \frac{3V_{LSB}}{8}, \pm \frac{V_{LSB}}{2}, \pm \frac{5V_{LSB}}{8}, \pm \frac{3V_{LSB}}{4}, \pm \frac{7V_{LSB}}{8}$ . Ceci nous permet d'ajouter huit nouveaux codes entre deux codes successifs de la résolution d'origine. Ces huit nouveaux codes sont rendus accessibles grâce aux huit plus petites tensions soit :  $\pm \frac{V_{LSB}}{8}, \pm \frac{V_{LSB}}{4}, \pm \frac{3V_{LSB}}{8}$  et  $\pm \frac{V_{LSB}}{2}$ . Qu'en est-il des six tensions restantes ? En réalité, elles sont inutiles car elles donnent la même information que les tensions de biais plus petites mais avec un décalage d'un  $V_{LSB}$ . Par exemple, on conviendra aisément que connaître le code de sortie correspondant à  $V_{IN} + \frac{5V_{LSB}}{8}$  n'est d'aucune utilité si on connaît déjà celui associé

à  $V_{IN} - \frac{3V_{LSB}}{8}$ . En effet, avec un écart en tension égal à  $V_{LSB}$ , le premier code est par définition égal au deuxième plus un.

En tenant compte de ces nouvelles considérations, le nombre de conversion effectuée peut être réduit de 15 (la conversion d'origine plus 14 avec biais d'échantillonnage) à 9 (la conversion d'origine plus 8 avec biais d'échantillonnage) et ceci avec un résultat final strictement identique. La principale conséquence de cette observation est l'amélioration de la *FoM*. Le facteur de dégradation de la *FoM* passe ainsi de 1.9 à 1.1 (soit 40% d'amélioration).

Dans l'optique d'améliorer au maximum les performances, il semblerait également intéressant de reconsidérer la méthode de détermination du résultat final. Actuellement, le système fournit en sortie la moyenne des conversions successives mais cette méthode pourrait être abandonnée au profit d'une recherche de type séquentielle ou dichotomique. En effet, on pourrait envisager de se rapprocher du code de résolution supérieure en testant des tensions parasites de plus en plus petites comme l'illustre la figure 8.1.

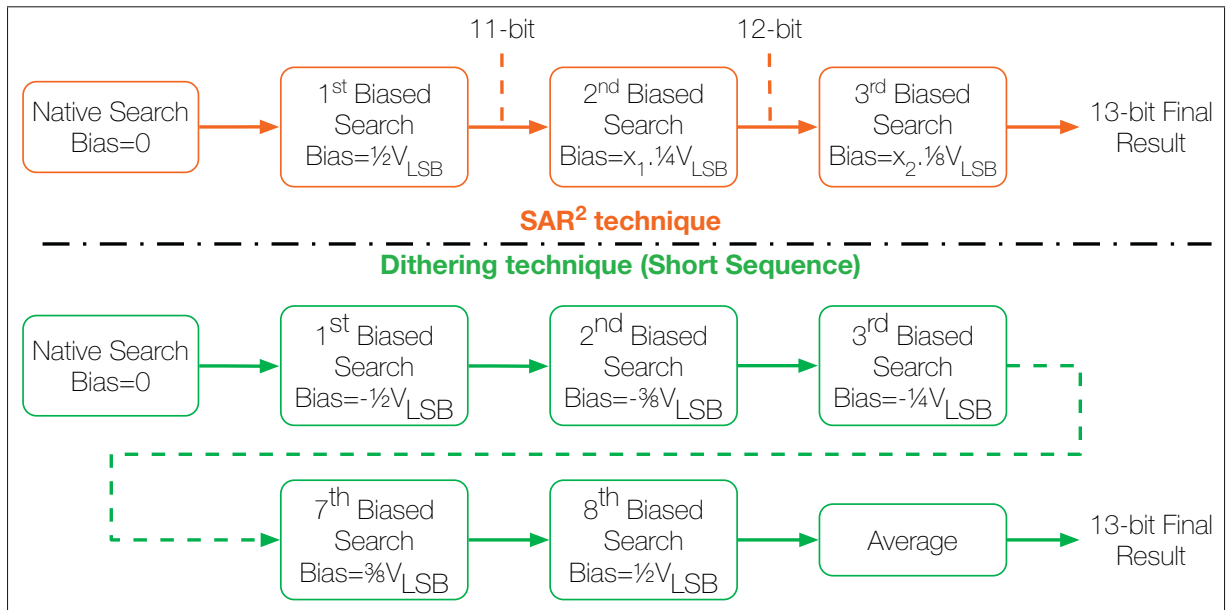


Figure 8.1 Étapes des techniques SAR<sup>2</sup> et dithering

L'utilisation de la recherche dichotomique, qui présente une complexité en  $O(\log_2(n))$ , permettrait ainsi d'améliorer la figure de mérite de manière significative. Ainsi pour une résolution de 13 bits le nombre maximal de conversions supplémentaires serait de  $\log_2(2^3) = 3$ . Avec la conversion initiale on aboutit à 4 conversions entraînant une division de la  $FoM$  par deux (cf. formule 8.2, page 80).

$$\frac{FoM_{Dithering\ dichotomique}}{FoM_{Native}} = \frac{Power\ Consumption}{\frac{1}{4} \cdot f_s \cdot 2^{13}} \cdot \frac{f_s \cdot 2^{10}}{Power\ Consumption} = \frac{1}{2} \quad (8.2)$$

La figure 8.2 (page 80) représente l'évolution des facteurs de dégradation de la  $FoM$  en fonction de la résolution souhaitée en mode *dithering* et ce pour les différentes solutions proposées.

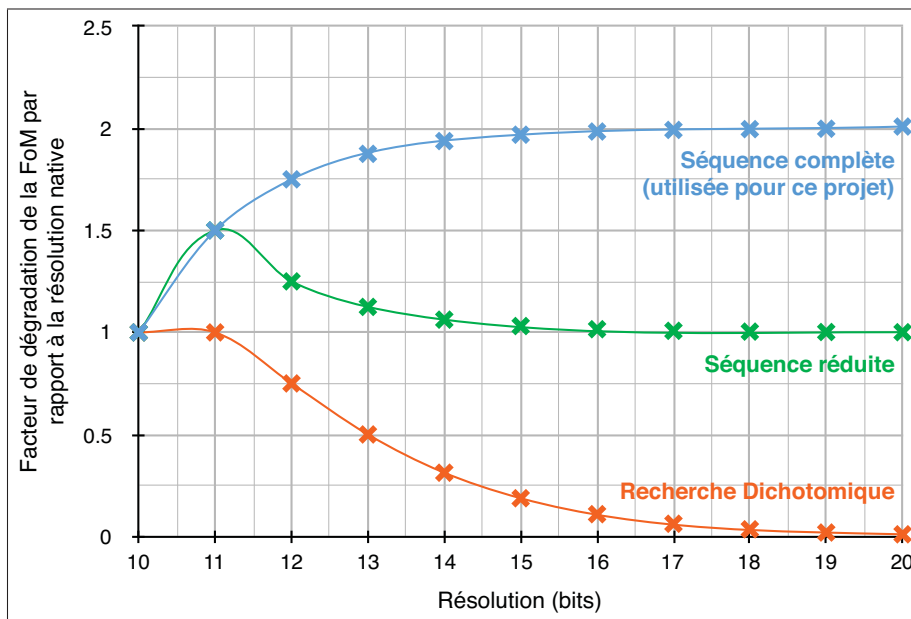


Figure 8.2 Évolutions du facteur de modification de la  $FoM$  en fonction de la résolution

On constate avec cette séquence réduite que l'augmentation de la résolution va de pair avec la réduction du facteur de dégradation de la  $FoM$ . Contrairement à la version actuelle de l'ADC qui est pénalisée par l'augmentation de la  $FoM$ , cette nouvelle solution permettrait d'atteindre des performances similaires à celles du convertisseur en mode natif en terme de consomma-



tion d'énergie par conversion. La recherche dichotomique quant à elle, permet d'atteindre les meilleures performances puisqu'elle entraîne une réduction de la  $FoM$  de plus en plus grande à mesure que la résolution augmente. Ce dernier résultat est toutefois à prendre avec un certain recul. En effet, il ne tient pas compte de l'augmentation de la consommation de courant générée par l'implémentation d'un algorithme de recherche dichotomique au sein du cœur logique. Les résultats réels pourraient ainsi être légèrement moins bons.

## 8.2 Évolution du projet

Cette ultime section s'attache à décrire brièvement l'architecture d'une puce figure (cf. 8.3, page 81) qui permettrait d'évaluer avec précision les avantages et les inconvénients du *dithering*. La résolution de cette puce serait de revue à la baisse afin de simplifier le système et de gagner de la place pour embarquer deux circuits logiques. Prenons ici un exemple d'un convertisseur 8 bits (on diviserait la taille du *DAC* par quatre).

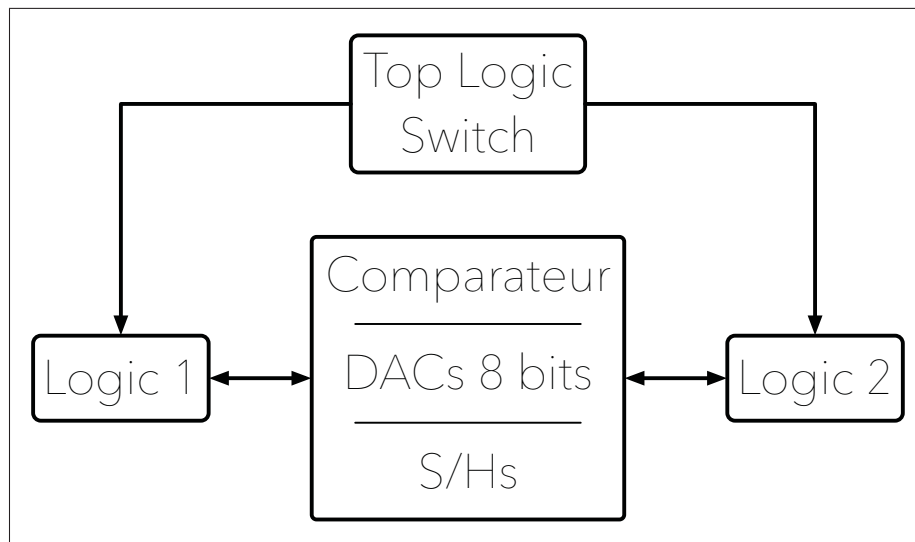


Figure 8.3 Organisation possible d'une puce testant les performances du *dithering*

Un premier circuit nommé *Logic 1* permettrait d'évaluer le fonctionnement de l'*ADC* en mode natif sur 8 bits alors que le cœur *Logic 2* utiliserait 6 bits de résolution native pour finalement

atteindre 8 bits par recherche dichotomique. De cette façon, il sera possible de comparer les performances des deux modes pour des résolutions et des sous circuits analogiques identiques.

Un module de gestion nommé *Top Logic Switch* sera en charge de gérer le mode de fonctionnement du convertisseur. Via un système de *power gating*, il pourra alors désalimenter le cœur logique inutilisé pour l'exclure du bilan de puissance. On peut également envisager de désalimenter la partie inutilisée du *DAC* en mode *dithering* afin d'être le plus précis possible sur la consommation d'énergie. Avec cette méthode il sera possible de chiffrer les gains et les coûts d'une telle technique de résolution variable. De plus, il serait intéressant d'augmenter la tension  $V_{LSB}$  de tel sorte que les tensions parasites ajoutées au signal en mode *dithering* soient discernables à l'oscilloscope (et pas noyée dans le bruit). De cette façon, on pourrait, grâce aux *AOP*, vérifier le fonctionnement du mode *dithering*.

Du point de vue de la testabilité, il semble essentiel de conserver les points de test actuel (entrées/sortie du comparateur et bits de contrôle). Cependant on pourra ajouter deux commandes séparées permettant de désactiver l'un ou l'autre des deux *sample-and-hold* afin de faire fonctionner le comparateur en mode commun et non différentiel. L'idée étant que si un problème similaire à celui rencontré ici survient on soit tout de même en mesure de tirer des paramètres caractéristiques viables.

Afin d'anticiper au mieux les performances du convertisseur il serait également intéressant de recréer une bibliothèque de cellules logiques pour remplacer celle de la technologie. On serait alors à même de fournir un bilan de puissance de la partie numérique et de simuler des tests sur les tensions d'alimentation sous *AMS*. Cela permettrait de statuer sur l'impact de l'architecture *MSV* dans la réduction de la consommation et d'optimiser les cellules logiques pour sauver de la puissance.

L'objectif de ces mesures est de recentrer le projet sur le *dithering* sans forcément chercher à faire tout de suite une puce hautes performances. Dans un second temps, et une fois le concept pleinement validé, il sera possible de complexifier le système. Parmi les éléments qu'ils seraient intéressant d'ajouter à ce convertisseur on peut penser à une gestion interne de l'alimen-

tation (stabilisée en température par exemple) ou encore à un circuit de protection contre les décharges électrostatiques (en anglais *ESD* pour *Electrostatic discharge*). Il serait également intéressant de combiner le *DAC* hybride avec une architecture à *split-capacitor* à l'instar de la solution proposée dans (Tahezadeh-Sani *et al.*, 2014). Ceci aurait pour effet de réduire la consommation de courant et la surface de silicium occupée.



## CONCLUSION

L'objectif de ce projet est de concevoir et tester un *SAR-ADC* utilisant une fonction *dithering* pour augmenter sa résolution. Dans cette optique, une architecture relativement complexe a été mise en place. Malgré des résultats de simulations encourageant avec un *ENOB* à 9.88 bits en mode natif et 11.4 bits en mode *dithering* les puces testées ont révélé un défaut majeur absent lors des simulations informatiques. Outre l'objectif de rendre la puce fonctionnelle, le projet devrait à présent se tourner vers l'optimisation de la puissance et la mise en place d'une véritable structure de test permettant de quantifier les avantages et inconvénients du *dithering* (*cf.* section 8.2, 81).

La majeure partie de mon temps de travail a été utilisée pour la mise au point d'un flot de conception automatisé permettant d'associer des briques numériques et analogiques en technologie *CMOS* 130nm. Le travail effectué sur l'analyse théorique et l'optimisation du concept sont également un apport important de cette étude. De plus, on pourra retenir la forte réduction de 75% de la consommation théorique de courant par rapport à la version 2015 de la puce, notamment grâce à l'implémentation d'une architecture à alimentation multiple. L'intégration d'une architecture de tests analogique/numérique est également un nouvel apport à ce projet au même titre que l'obtention de résultats de simulation plus avancés qui nous permettent de mieux évaluer les performances du *dithering* dans le cadre des *SAR-ADC*.

Un article a été soumis et accepté au *IEEE International Symposium on Circuits and Systems (ISCAS)* qui se tiendra en mai 2017 à Baltimore. De plus, une *IEEE Electronic Letter* est également en cours de rédaction. Celle-ci complétera l'article avec les résultats de simulation en mode *dithering* mais aussi en expliquant les améliorations possibles du concept décrites au chapitre 8 (page 77).



## ANNEXE I

### DESSINS DES MASQUES DE LA PUCE

#### 1. Assemblages

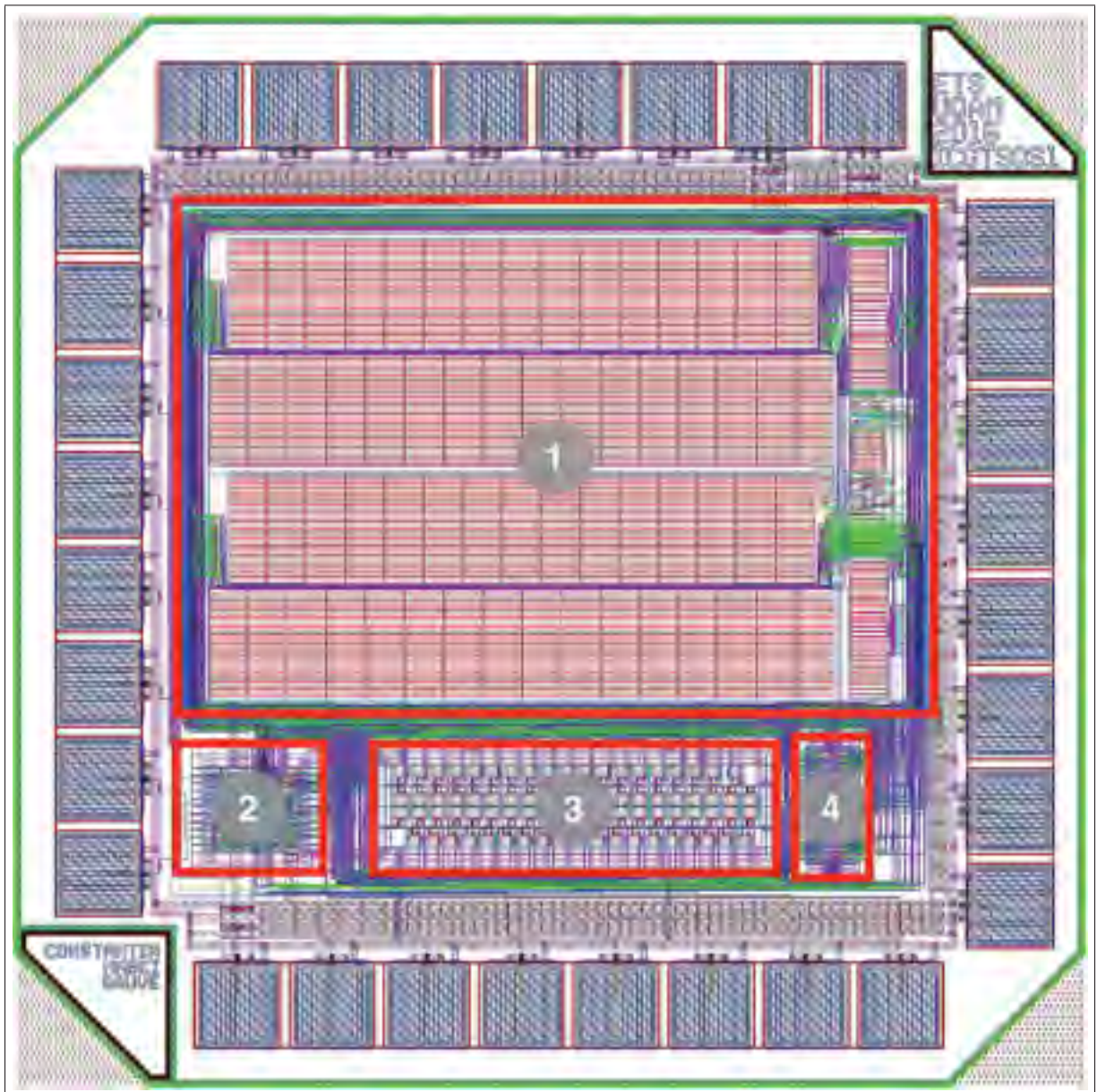


Figure-A I-1 Vue générale du dessin des masques de la puce

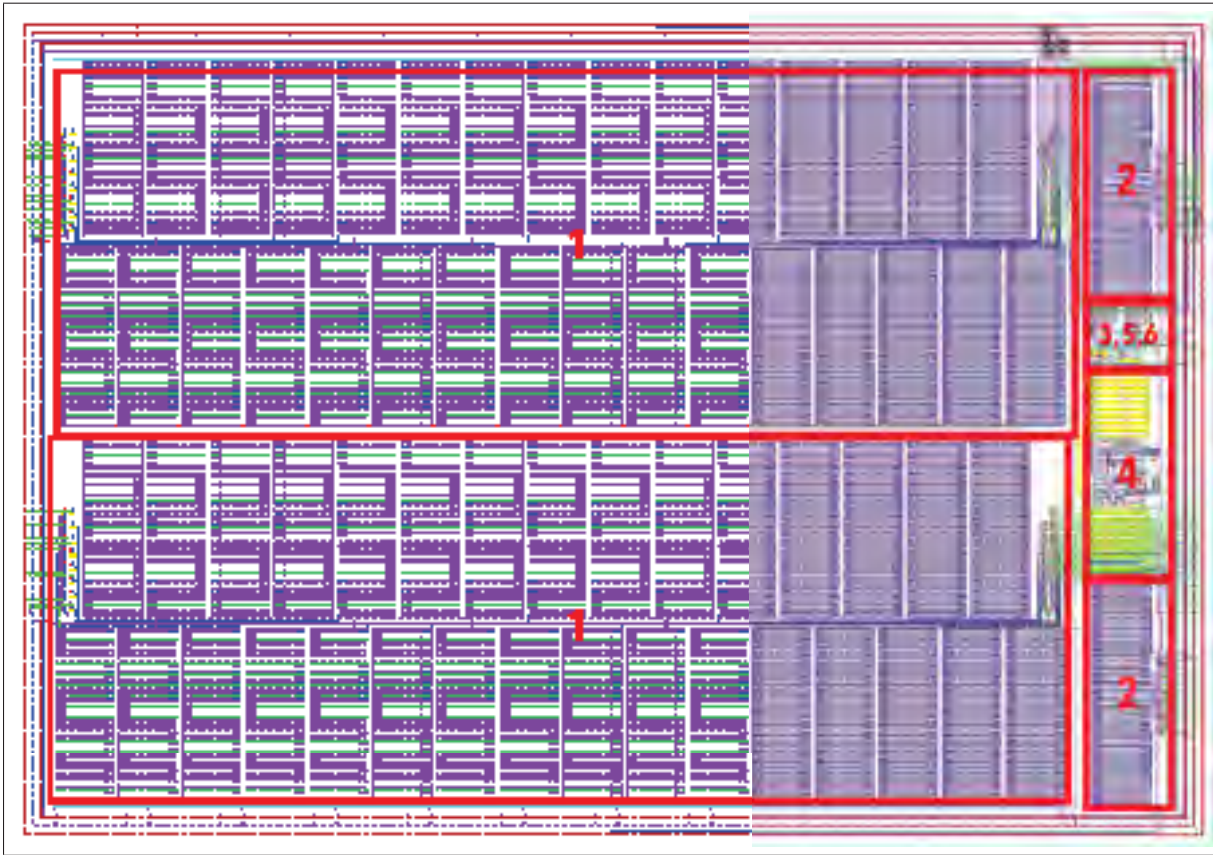


Figure-A I-2 Vue du dessin des masques de la partie analogique



## 2. Blocs logiques

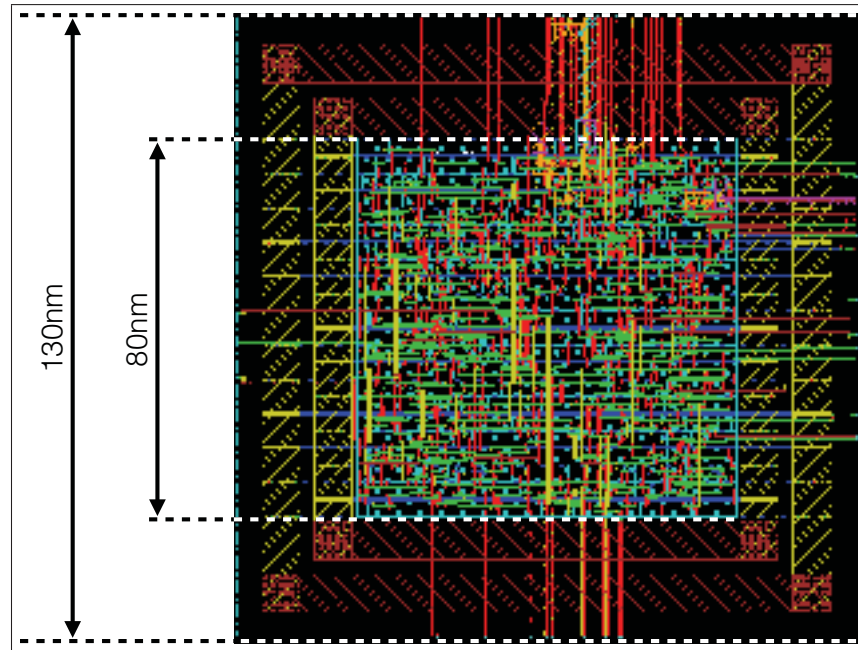


Figure-A I-3 Vue du dessin des masques du cœur logique

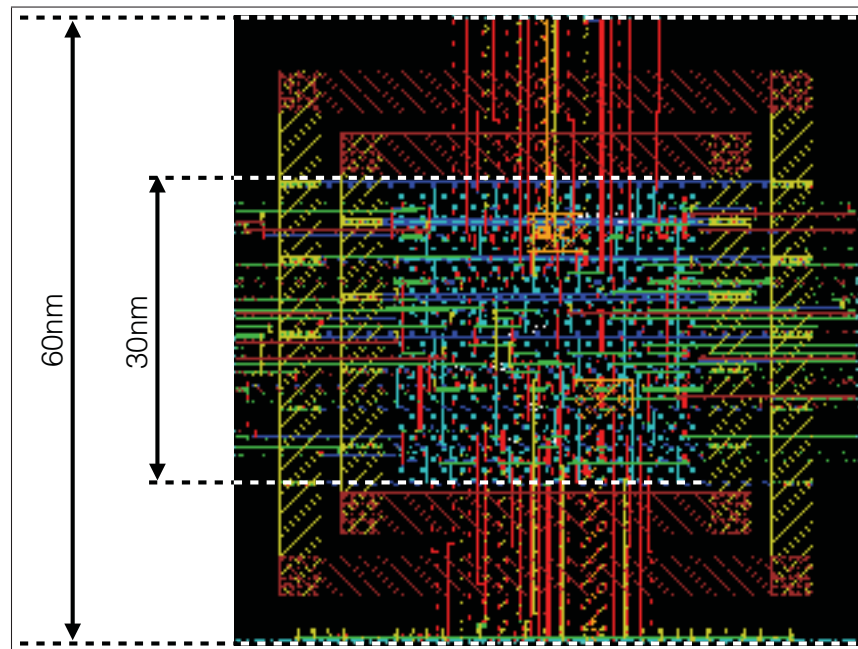


Figure-A I-4 Vue du dessin des masques d'un *DAC Driver*



## ANNEXE II

### RÉSUMÉ DES SPÉCIFICATIONS DU CONVERTISSEUR

Tableau-A II-1 Résumé des spécifications simulées et expérimentales de l'ADC

		Mode natif	Mode dithering
<b>Général</b>	<b>Technologie</b>	<i>CMOS</i> 130nm	
	<b>Alimentation (mV)</b>	900 <sub>(analogique)</sub> / 600 <sub>(numérique)</sub>	
	<b>Power (μW)</b>	4	
	<b>f<sub>CLK</sub> (MHz)</b>	1	
<b>Simulation</b>	<b>F<sub>sample</sub> (kS.s<sup>-1</sup>)</b>	83	5.5
	<b>DNL (LSB)</b>	0.5	NA
	<b>INL (LSB)</b>	2	NA
	<b>SNR (dB)</b>	61.4	70.3
	<b>SINAD (dB)</b>	61.2	70.3
	<b>ENOB (bits)</b>	9.88	11.4
	<b>FoM (fJ / conv step)</b>	53	269
<b>Limitations</b>	<b>Alimentation</b>	$V_{DDHIGH} < 2 \cdot V_{DDLLOW}$	
	<b>V<sub>DD</sub> max (V)</b>	1.2	
	<b>V<sub>DD</sub> min (V)</b>	0.6	
	<b>T<sub>start up</sub> digital</b>	au moins $T_{CLK}$ avec <i>reset</i> actif	
	<b>T<sub>start up</sub> analog (ps)</b>	25	
	<b>f<sub>CLK</sub> max (MHz)</b>	5	
	<b>f<sub>S</sub> max</b>	415	27.7



## ANNEXE III

### AMPLIFICATEUR OPÉRATIONNEL

#### 1. Schémas des sous-circuits de l'AOP

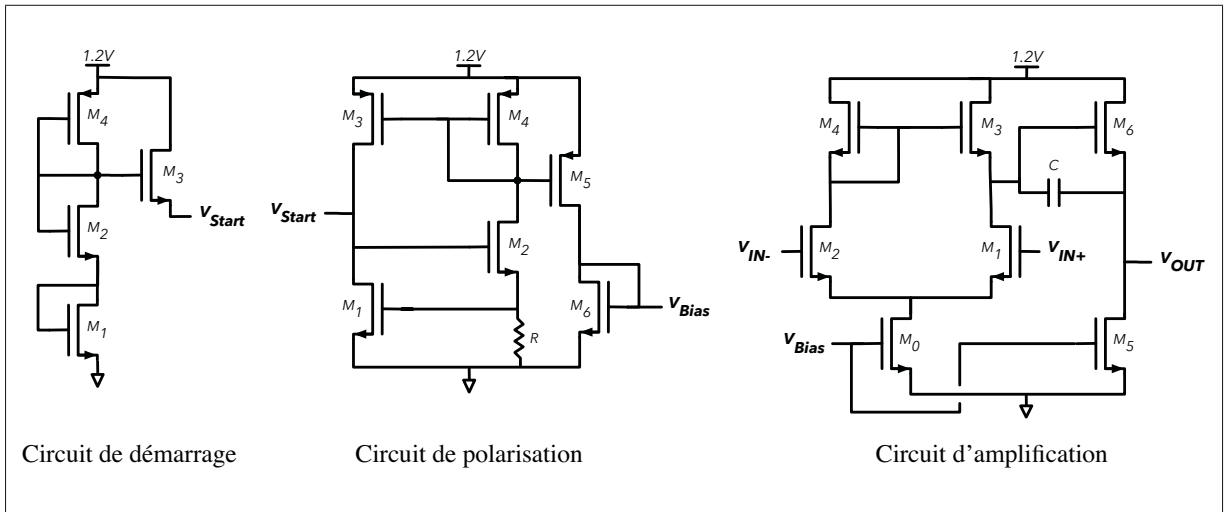


Figure-A III-1 Vue schématique des trois sous-circuits composant l'amplificateur opérationnel

Tableau-A III-1 Dimensionnement des différentes composantes des amplificateurs opérationnels

Circuit de polarisation	Circuit d'amplification	Circuit de démarrage
$M_1$ 280nm	$M_0$ 280nm	$M_1$ 280nm
$M_2$ 280nm	$M_1$ 280nm	$M_2$ 280nm
$M_3$ $2\mu\text{m}$	$M_2$ 280nm	$M_3$ 280nm
$M_4$ $2\mu\text{m}$	$M_3$ 280nm	$M_4$ 280nm
$M_5$ $2\mu\text{m}$	$M_4$ 280nm	
$M_6$ 280nm	$M_5$ 830nm	
$R_B$ 60.66k $\Omega$	$M_6$ $2\mu\text{m}$	
	$C_{comp}$ 30fF	

(N.B. tous les transistors sont de longueur minimale soit 120nm).

## 2. Performances de l'amplificateur opérationnel

Tableau-A III-2 Résumé des performances des amplificateurs opérationnels (simulations *Spectre post-layout*)

		$C_{OUT} = 1pF$	$C_{OUT} = 10pF$
<b>DC</b>	$V_{OUT}$	602.1mV	602.1mV
	$I_{bias-OTA\#1}$	6.81 $\mu$ A	6.81 $\mu$ A
	$I_{bias-OTA\#2}$	19.8 $\mu$ A	19.8 $\mu$ A
<b>AC</b>	$G_0$	37.6dB	37.7dB
	$f_{-3dB}$	1.60MHz	373kHz
	$M_G$	17.2dB	29.8dB
	$M_\phi$	36.4°	50.8°
<b>Transient</b> ( $V_{IN} = \text{pulse}$ , $T = 10\mu s$ , $V_H = 700mV$ et $V_L = 500mV$ )	<i>Offset</i>	0.03mV	0.03mV
	$\Delta_H$	1.59mV	2.77mV
	$\Delta_L$	2.42mV	5.57mV

## BIBLIOGRAPHIE

- Chang, Y.-K., Wang, C.-S. & Wang, C.-K. (2007). A 8-bit 500-KS/s Low Power SAR ADC for Bio-Medical Applications. *2007 IEEE Asian Solid-State Circuits Conference*, pp. 228-231.
- Chiu, P.-Y. & Ker, M.-D. (2014). Metal-layer capacitors in the 65 nm CMOS process and the application for low-leakage power-rail ESD clamp circuit. *Microelectronics Reliability*, 54(1), 64 - 70. doi : <http://dx.doi.org/10.1016/j.microrel.2013.08.011>.
- Dessouky, M. & Kaiser, A. (2001). Very Low-Voltage Digital-Audio  $\Delta\Sigma$  Modulator with 88-dB Dynamic Range Using Local Switch Bootstrapping. *IEEE Journal Of Solid State Circuit*, 36(3), 349-355.
- Favre, D. (2015). *Conception d'un SAR ADC 10bits avec dithering*. UQAM & CPE Lyon, Rapport de stage de fin d'étude (cursus école d'ingénieur française) de M. Damien Favre.
- Gulati, K. & Lee, H.-S. (2001). A Low-Power Reconfigurable Analog-to-Digital Converter. *IEEE Journal Of Solid State Circuit*, 36(12), 1900-1911.
- Hong, H.-C. & Lee, G.-M. (2007). A 65-fj/Conversion-Step 0.9-v 200-kS/s Rail-to-Rail 8-bit Successive Approximation ADC. *IEEE Journal Of Solid State Circuit*, 42(10), 2161-2168.
- Lotfi, R., Majidi, R., Maymandi-Nejad, M. & Serdijn, W. A. (2009). An Ultra-Low-Power 10-Bit 100-kS/s Successive-Approximation Analog-to-Digital Converter. *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1117-1120.
- Maxim Integrated. (2001). Understanding SAR ADCs : Their Architecture and Comparison with Other ADCs. Consulté le 5 octobre 2016, Repéré à <http://pdfserv.maximintegrated.com/en/an/AN1080.pdf>.
- Maxim Integrated. (2002). Understanding Single-Ended, Pseudo-Differential and Fully-Differential ADC Inputs. Consulté le 5 octobre 2016, Repéré à <http://pdfserv.maximintegrated.com/en/an/AN1108.pdf>.
- Przyborowski, D. & Idzik, M. (2010). A 10-bit Low-Power Small-Area High-Swing CMOS DAC. *IEEE Transactions on Nuclear Science*, 57(1), 292-299.
- Sauvé, Q., Favre, D., Morin-Laporte, G., Taherzadeh-Sani, M., Constantin, N. & Nabki, F. (2017). A Calibration-Free 13-Bit 0.9 V Differential SAR-ADC with Hybrid DAC and Dithering. *IEEE International Symposium on Circuits and Systems*.
- Short, K. L. (2008). *VHDL for engineers* (pp. 449 - 457). Copyright 2009, Pearson Education, Inc., Upper Saddle River, New Jersey 07458.

- Taherzadeh-Sani, M., Lotfi, R. & Nabki, F. (2014). A 10-bit 110 kS/s 1.16  $\mu$ W SA-ADC With a Hybrid Differential/Single-Ended DAC in 180-nm CMOS for Multichannel Biomedical Applications. *IEEE Transactions on Circuits and Systems II : Express Briefs*, 61(8), 584-588.
- Zhang, E., Mawelo, R. M., Fayomi, C. & Nabki, F. (2013). A 0.9 V Low-Power Reconfigurable Successive Approximation ADC for Integrated Sensors. *2013 IEEE 11th International new Circuits and Systems Conference (NEWCAS)*, pp. 1-4.