Stage de Recherche Master 2 ESA

Rapport

Conversion A/N pour Capteurs d'Images à Pixels Actifs CMOS

par

Rafael SANTOS FERREIRA

Élève en 3ème année à l'ENSEA option ECM et en Master 2 ESA

ETIS - ENSEA / Université de Cergy-Pontoise / CNRS UMR 8051
6 avenue du Ponceau, 95014 Cergy-Pontoise Cedex, France

Soutenu le 30 Août 2013

Devant le jury composé de :

Pr.	Emmanuelle Bourdel	ENSEA	Responsable de stage
Pr.	Cedric Duperrier	ENSEA	Rapporteur et Professeur conseiller
Pr.	Caroline Lelandais-Perrault	Supelec	Encadrante du stage
Pr.	Filipe Vinci	Supelec	Encadrant du stage

Résumé

Ce rapport décrit un macro-modèle complet LTSPICE d'un convertisseur analogique-numérique pour des imageurs à capteurs de pixels actifs CMOS, afin d'évaluer les non-idéalités intrinsèques à l'architecture et ainsi développer une méthode de correction par calibration numérique.

Abstract

This technical note describes a complete LTSPICE macromodel of an analog-to-digital converter for CMOS active-pixel sensor imagers, in order to evaluate the architecture's inherent non-idealities and finally allow a correction method using digital calibration.

Remerciements

Je tiens à remercier dans un premier temps toute l'équipe pédagogique de l'ENSEA et du laboratoire ETIS pour m'avoir assuré une formation de qualité en ingénierie électronique primordiale pour la réussite de mon stage.

Je remercie également toute l'équipe de la Chaire Thalès en Conception Analogique Avancée et du Département Signaux et Systèmes Electroniques pour m'avoir accueilli et permis de découvrir les particularités de la conception électronique analogique.

Plus précisément, je tiens à remercier sincèrement :

- Mme. Caroline Lelandais-Perrault, mon encadrante, pour sa patience ainsi que sa pédagogie vis-à-vis de toutes mes questions de logiques/numériques et dans la rédaction du présent rapport ;
- M. Filipe Vinci, mon encadrant, pour sa disposition et son effort à m'avoir instruit sur la macro-modélisation, le simulateur SPICE et leur contexte au fur et à mesure que le stage avançait ;
- Mme. Emmanuelle Bourdel, ma responsable de stage, pour sa disponibilité à m'avoir assisté dans toutes les questions administratives concernant le master ;
- M. Cédric Duperrier, professeur conseiller, pour sa présence et son effort durant toute la période de ma formation ;
- Tous les collègues – enseignants, doctorants, stagiaires – pour avoir animé cette période de stage ;
- Ma famille, pour le soutien et la patience qu'elle m'a témoignée, malgré la distance.

Enfin je remercie Dieu pour les merveilles qu'Il a accomplies dans ma vie. A Lui je dédie tous les bons résultats de ma formation professionnelle.

Table des matières

RESUME **5**
SUPELEC **7**
CHAIRE THALES-SUPELEC 8
1. INTRODUCTION **9**
1.1 IMAGEUR APS CMOS 9
1.2 CONVERTISSEURS ANALOGIQUE-NUMERIQUE (CANS) 11
1.3 SPICE 23
1.4 MACRO-MODELISATION 25
2. CAN CYCLIQUE **26**
2.1 CAN CYCLIQUE CONVENTIONNEL 26
2.2 CAN CYCLIQUE RSD 31
2.3 IMPLEMENTATION 34
2.4 NON-IDEALITES 40
3. MACRO-MODELISATION **42**
3.1 OTA 42
3.2 CONDENSATEURS 46
3.3 COMMUTATEURS 48
3.4 COMPARATEURS SYNCHRONES 51
3.5 CIRCUITS LOGIQUES 51
4. RESULTATS ET DISCUSSION **54**
4.1 ECHANTILLONNEUR/BLOQUEUR 54
4.2 SOUSTRACTEUR ET MULTIPLICATEUR 56
4.3 CIRCUIT RSD 58
4.4 CAN CYCLIQUE RSD 58
5. CONCLUSIONS **66**
6. REFERENCES **67**

List of Figures

Figure 1-1 - Capteur d'image CMOS APS avec conversion à colonne parallèle (2) 11
Figure 1-2 - Fonction de Transfert d'un CAN de 3 bits (7) 13
Figure 1-3 - Erreur de quantification d'un CAN 3 bits (7) 14
Figure 1-4 – Interprétation des erreurs d'*offset* et de gain (7)..... 15
Figure 1-5 – Résolution X Vitesse de conversion 20
Figure 1-6 - CANs à entrées *single-ended* (8)..... 21
Figure 1-7 - CAN à entrées *pseudo-differential* (8)..... 22
Figure 18 - CAN à entrées *fully-differential* (8) 22
Figure 2-1 – Schéma fonctionnel d'un CAN Cyclique conventionnel 27
Figure 2-2 - Logigramme d'un CAN cyclique..... 28
Figure 2-3 - Fonction de transfert d'un cycle d'un CAN 1-bit/étape 28
Figure 2-4 - Effet de l'*offset* du comparateur dans un CAN Cyclique conventionnel..... 29
Figure 2-5 - Effet de l'offset de l'amplificateur dans un CAN Cyclique conventionnel 30
Figure 2-6 - Effet d'un gain constant différent de 2 sur l'amplificateur dans un CAN Cyclique conventionnel 30
Figure 2-7 - Logigramme d'un CAN Cyclique RSD 32
Figure 2-8 - Fonction de transfert d'un cycle d'un CAN RSD 33
Figure 2-9 - Effet de l'offset du comparateur dans un CAN Cyclique RSD 34
Figure 2-10 - Effet de l'offset de l'amplificateur dans un CAN Cyclique RSD..... 34
Figure 2-11 - CAN Cyclique RSD 35
Figure 2-12 – Echantillonneur/Bloqueur *fully-differential* 36
Figure 2-13 – Soustracteur et Multiplicateur *Fully-Differential* 37
Figure 2-14 – Sous-CAN d'un CAN Cyclique RSD 39
Figure 2-15 – CNA d'un CAN Cyclique RSD 39
Figure 2-16 – Circuit logique traduisant $P_{N-1}Q_{N-1}...P_0Q_0$ en bits $d_{N-1}...d_0$ 40
Figure 3-1 – Balun compris par deux transformateurs 43
Figure 3-2 – Macro-modèles de transformateur 43
Figure 3-3 – OTA *fully*-differential à limitation de tension à diodes 44
Figure 3-4 - OTA *fully*-differential à limitation de tension à varistor..... 45
Figure 3-5 - OTA *fully*-differential à limitation de tension à sources comportementales de courant 46
Figure 3-6 – Condensateurs comprenant la capacité de la plaque inférieure, la capacitance non-linéaire et les deux simultanément 48
Figure 3-7 – Schémas, symboles et paramètres des commutateurs NMOS, PMOS et CMOS... 50
Figure 3-8 –Comparateur synchrone 51
Figure 3-9 - Additionneur complet 52
Figure 3-10 - Circuit logique du CAN Cyclique RSD 53
Figure 4-1 - E/H sur LTSPICE 55
Figure 4-2 - Vérification de fonctionnement de l'E/H 56
Figure 4-3 - Soustracteur/Multiplicateur implémenté sur LTSPICE..... 57
Figure 4-4 - Test du circuit Soustracteur/Bloqueur..... 57
Figure 4-5 - Teste de la fonction RSD implementé sur LTSPICE 58
Figure 4-6 – Banc d'essai du CAN Cyclique RSD de 8 bits 60
Figure 4-7 - Signaux de Contrôle de l'E/H et du Soustracteur/Multiplicateur..... 61
Figure 4-8 - Signaux de Contrôle du commutateur de l'entrée et *enable*..... 62
Figure 4-9 - Conversion A/N Cyclique RSD de 8 bits..... 63
Figure 4-10 - Fonction de Transfert et Erreur de Quantification du CAN Cyclique RSD 8 bits.. 64
Figure 4-11 - Conversion A/N d'une sinusoïde 65
Figure 4-12 – FFT d'une conversion A/N d'une sinusoïde 65

RESUME

Dans le flot de conception électronique analogique, la simulation électrique est sans nul doute reconnue comme fondamentale. L'évolution des modèles des semi-conducteurs utilisés dans ces simulations rend possible des résultats de plus en plus proches de la réalité. Cependant l'utilisation de ces modèles dans des circuits possédant un nombre important de composants augmente l'effort calculatoire du simulateur et prolonge le temps de simulation. Un exemple d'application où le temps de simulation peut devenir prohibitif, si l'on utilise les modèles complets, est le banc de Convertisseurs Analogique-Numérique d'un Capteur d'Image à Pixels Actifs CMOS.

Un Capteur d'Image à Pixel Actifs CMOS (*CMOS Active Pixel Sensor* ou CMOS APS) est construit à partir d'une matrice de cellules photosensibles, appelées pixels, dont les échantillonnages peuvent être faits soit successivement, soit simultanément, soit par lignes entières, etc. Les échantillonnages sont effectués par des Convertisseurs Analogique-Numérique (CAN) dont les performances requises – puissance, surface occupée, vitesse de conversion – vont dépendre de la stratégie retenue pour l'échantillonnage des sorties des pixels. L'échantillonnage à lignes entières, c'est-à-dire, l'acquisition simultanée des signaux de tous les pixels d'une ligne, s'avère être un bon compromis vitesse/puissance. Or, cette topologie exige un banc de CANs, chaque CAN étant dédié à une colonne de la matrice de pixels, ce qui ajoute des contraintes pour la conception et la rend donc plus complexe.

Afin de réaliser une simulation d'un banc de CANs d'un CMOS APS de manière plus rapide et suffisamment précise, une alternative possible aux modèles complets des semi-conducteurs est l'utilisation de macro-modèles. La macro-modélisation, outre l'optimisation en vitesse, favorise aussi l'analyse du circuit quand on s'intéresse aux impacts des variations de ses paramètres sur les performances. Dans le cas des CMOS APS, cette caractéristique peut faciliter l'étude d'une calibration numérique où tous les CANs sont utilisés, voire simulés, chacun possédant des valeurs de paramètres distinctes.

Ce rapport technique vise à présenter la macro-modélisation d'un CAN cyclique en prenant en compte les non-idéalités restreignant la performance de la conversion. A cet effet, on utilise le simulateur électrique du type *SPICE* développé par *Linear Technology* et appelé LTSPICE. Ce simulateur a été choisi en raison de sa stabilité, de son accessibilité aux utilisateurs et du fait de son utilisation par un grand nombre de personnes. Les simulateurs « multi-domaine » comme *Simulink* pourraient être utilisés aussi, mais normalement ceux-ci sont moins réalistes et plus lents. En outre, à travers ces simulateurs, la transposition des modèles à un niveau électrique devient plus difficile,

c'est-à-dire, le remplacement d'une partie particulière du macro-modèle par son modèle semi-conducteur y est moins évident que sur LTSPICE.

Ce rapport discute brièvement aussi de quelques problèmes provenant du simulateur lorsque la macro-modélisation est réalisée. En effet, pour que la macro-modélisation réussisse, quelques aspects internes de la simulation doivent être mis en évidence. La conséquence la plus courante d'une macro-modélisation non maitrisée est la non-convergence de la simulation. Pour que ce type de problème soit évité, le concepteur doit connaitre suffisamment le fonctionnement du simulateur et les limites du macro-modèle utilisé.

Le travail décrit à travers ce rapport permettra une future avancée vers la calibration numérique d'un banc de CANs d'un CMOS APS. Tout d'abord, à partir de la variation des paramètres des CANs, maintenant simplifiés grâce à la macro-modélisation, l'analyse de la variation de leurs performances peut être faite de façon statistique, comme à travers la méthode Monte Carlo par exemple. Ensuite, cette variation peut être faite simultanément pour tous les CANs du CMOS APS afin que la calibration numérique soit développée. Ainsi, le focus de la recherche pourra se concentrer plus efficacement sur les algorithmes numériques.

Le rapport est organisé en cinq chapitres. Les chapitres 1 et 2 introduisent tous les concepts nécessaires pour bien comprendre le sujet du rapport. Le chapitre 1 présente le CMOS APS, les CANs, le simulateur *SPICE* et enfin la macro-modélisation. Le chapitre 2 présent le CAN cyclique conventionnel et RSD, leurs caractéristiques et leur implémentation. Le chapitre 3 présente comment la macro-modélisation des composants du CAN cyclique a été réalisée. Le chapitre 4 aura pour but de présenter les schémas du CAN Cyclique RSD et les résultats de simulation. Le chapitre 5 enfin conclura le rapport en résumant ce que l'on a eu comme résultat de ce travail.

En outre, le travail décrit par ce rapport est le produit d'un stage de Master réalisé entre avril et août 2013 à la Chaire Thales et au Département de Signaux et Systèmes Electroniques de Supelec à Gif-sur-Yvette.

SUPELEC

Grande école d'ingénieurs française, Supélec est la référence dans le domaine des sciences de l'information, de l'énergie et des systèmes : informatique, télécommunications, électronique, traitement du signal, automatique, génie électrique, etc.

La mission de Supélec est triple : formation initiale, recherche et formation continue. L'organisation interne de l'école permet une synergie entre ces trois activités confiées à des équipes rassemblées autour de thèmes spécifiques. Des équipements professionnels très performants sont mis à leur disposition.

Particulièrement dans la recherche, la caractéristique essentielle à Supélec, qu'elle soit fondamentale ou finalisée, est de découler de problématiques industrielles. Elle est menée en liaison étroite avec l'industrie, notamment grâce aux contrats de recherche. Les liens étroits entre la recherche et l'enseignement permettent d'intégrer, dans les cours et les travaux des élèves, les avancées technologiques les plus récentes. La recherche est ainsi le meilleur garant de la qualité de l'enseignement.

Créée en 1894 pour répondre aux besoins de la nouvelle industrie issue de l'électricité, Supélec est aujourd'hui une véritable entreprise de plus de 2000 personnes (personnel, enseignants, chercheurs et élèves réunis), s'adaptant en permanence aux évolutions technologiques et économiques afin d'offrir un enseignement et une recherche de haute qualité.

Supélec développe dans cet esprit une forte politique de partenariat à la fois régionale, nationale et internationale aussi bien avec les milieux de l'industrie et de la recherche qu'avec les milieux de l'enseignement supérieur.

Professionnalisme, ouverture, dynamisme, goût du travail en équipe, rigueur intellectuelle mais aussi convivialité reflètent la qualité des relations qu'élèves-ingénieurs, enseignants et chercheurs de Supélec souhaitent développer entre eux et avec leurs partenaires.

Supélec en chiffres :

- 2000 étudiants
- 460 diplômes d'ingénieurs
- 174 programmes de formation continue
- 5 mastères spécialisés

- 1 Certificat de qualification professionnelle
- 133 enseignants-chercheurs
- 620 enseignants-vacataires
- 180 personnels fonctionnels et direction
- 7 laboratoires et unités de recherche
- 243 doctorants

CHAIRE THALES-SUPELEC

La Chaire d'excellence Thales/Supélec sur la Conception Analogique Avancés est une chaire créée par Thales et Supélec pour servir de support à la recherche et à l'enseignement dans le domaine interdisciplinaire des signaux mixtes et de la conception de systèmes analogiques, favorisant ainsi l'intérêt des étudiants vers les carrières d'ingénierie des systèmes analogiques.

La Chaire Thales-Supélec est hébergée par le Dép. de Signaux et Systèmes Electroniques à Gif-sur-Yvette, et collabore étroitement avec ces équipes tant dans l'enseignement comme pour la recherche.

1. INTRODUCTION

Ce chapitre fournit au lecteur une vue d'ensemble de ce projet et de son application, ainsi que des outils et des techniques utilisés durant son exécution. L'objectif est d'expliquer au lecteur ce qu'est un Imageur CMOS APS et comment les CANs les intègrent, en soulignant les performances normalement exigées et les compromis pris en compte lors du choix de la topologie d'un CAN. Ensuite, pour que le lecteur comprenne les avantages de réaliser la macro-modélisation à travers un simulateur *SPICE*, deux paragraphes sont dédiés à ce sujet.

La Section 0 présente les Imageurs CMOS APS et l'architecture de conversion analogique/numérique en colonnes parallèles. Puis la Section 0 présente un panorama des CANs et des différentes topologies, en soulignant leurs avantages et leurs inconvénients. La Section 0 donne de brèves descriptions et un court historique des simulateurs *SPICE*. Enfin, la Section 0 introduit la macro-modélisation, explique pourquoi elle est ciblée ici et ce à quoi on s'attend à être confronté lors de son utilisation.

1.1 IMAGEUR APS CMOS

Suite à l'apparition de la micro-électronique, l'étude des capteurs d'images électroniques s'est développée depuis les années 60. Les premiers capteurs d'image ont été basés sur les technologies bipolaires, puis PMOS, NMOS et finalement CMOS. Toutefois, pendant les années 70 et 80, on s'est focalisé quasiment exclusivement sur les capteurs d'image CCD (*Charged Coupled Device*) en raison de la petite taille d'un pixel par rapport à celui d'un capteur CMOS et aussi parce que le capteur CCD possédait une meilleure robustesse vis-à-vis du bruit physique. Finalement, à partir des années 90, en raison des avancées de la technologie de fabrication des circuits intégrés, la communauté scientifique s'est tournée vers les capteurs CMOS à nouveau. Dès lors la technologie CMOS promettait des avantages sur la miniaturisation et la rentabilité face aux capteurs CCD. En outre, suite au développement de l'Intégration à Très Grand Echelle (*Very Large Scale Integration*, *VLSI*), la technologie CMOS offrait l'opportunité d'intégrer sur la même puce plusieurs d'autres fonctionnalités. Par conséquent, les systèmes d'imagerie sont devenus plus performants et leurs coûts de production à grande échelle plus faibles. Toutefois, la technologie CCD est de plus en plus utilisée dans les applications où on doit atteindre une haute performance en dépit de son coût, sa surface, etc. (1)

En partant de la technologie CMOS, il existe différentes manières d'implémenter un pixel. Habituellement on distingue les pixels passifs et les pixels actifs. Les pixels passifs sont normalement constitués d'une photodiode et d'un seul transistor utilisé pour la lecture du pixel. L'avantage du pixel passif est sa taille réduite, ce qui diminue les coûts et

augmente la mobilité du système optique. Son principal inconvénient est le bruit de lecture. Afin de minimiser le bruit de lecture, on introduit dans le pixel un *buffer*/amplificateur. Un capteur d'image dont chaque pixel possède un amplificateur actif est appelé un Capteur d'Image à Pixel Actif (APS). Régulièrement on retrouve de pixels actifs conçus à partir de *photodiodes* ou *photogates* suivis d'un amplificateur à drain commun qui n'est utilisé que pendant la lecture, d'où une consommation de puissance non excessive (1).

De manière à mesurer et quantifier numériquement les signaux capturés par les pixels, des CANs doivent être implémentés entre le capteur, voire les sorties des pixels, et l'interface dans laquelle l'image sera traitée. Les caractéristiques du CAN, expliquées en détail sur la Section 0, ne doivent pas correspondre qu'au pixel mais aussi à l'architecture globale du capteur. En effet, pour que l'on puisse avoir un capteur récupérant une image complète, il faut que l'on distribue les pixels spatialement afin de créer une matrice de lignes et colonnes où chaque élément est un pixel. Après, la manière de lire les pixels varie d'une conception à l'autre. (1)

Il est possible de réaliser la lecture d'une image complète à partir de la lecture individuelle et simultanée de chaque pixel, c'est-à-dire, réserver un CAN à chacun d'eux. Cette option rend possible un taux d'image par seconde plus grande, mais exige une large surface sur puce. A l'opposé, il existe l'option d'utiliser un CAN pour le capteur entier. Ainsi, la surface dédiée à la conversion Analogique/Numérique (A/N) peut être réduite en dépit de la fréquence d'opération du CAN qui monte. Cependant, il a été montré que dans cette architecture, en conséquence de la montée de fréquence, la puissance consommée par le capteur entier augmente aussi. En outre, la lecture à grande vitesse dans le domaine analogique introduit du bruit large bande et de la distorsion du signal (2).

Dans le but de parvenir à un haut débit d'image par seconde, alors que le bruit généré et la performance générale du capteur sont maintenus, des travaux plus récents utilisent la conversion analogique-numérique à colonnes parallèles (2) - (3). Cette architecture consiste à utiliser un CAN par colonne ou, autrement dit, à échantillonner chaque ligne de l'imageur d'un seul coup. A cette fin, des circuits de contrôle doivent être prévus pour que les conversions de chaque ligne puissent être gérées. En outre, il existe aussi des variantes de cette architecture, par exemple la conversion à colonne partagée présenté par (4) où un CAN à l'aide d'un multiplexeur est responsable de la conversion de deux colonnes. De toute façon, l'architecture de conversions à colonnes parallèles s'avère un bon compromis entre puissance, taux d'image par seconde, bruit et surface occupée. La Figure 1-1 présente un exemple de Capteur d'image CMOS APS avec conversion analogique-numérique à colonnes parallèles.

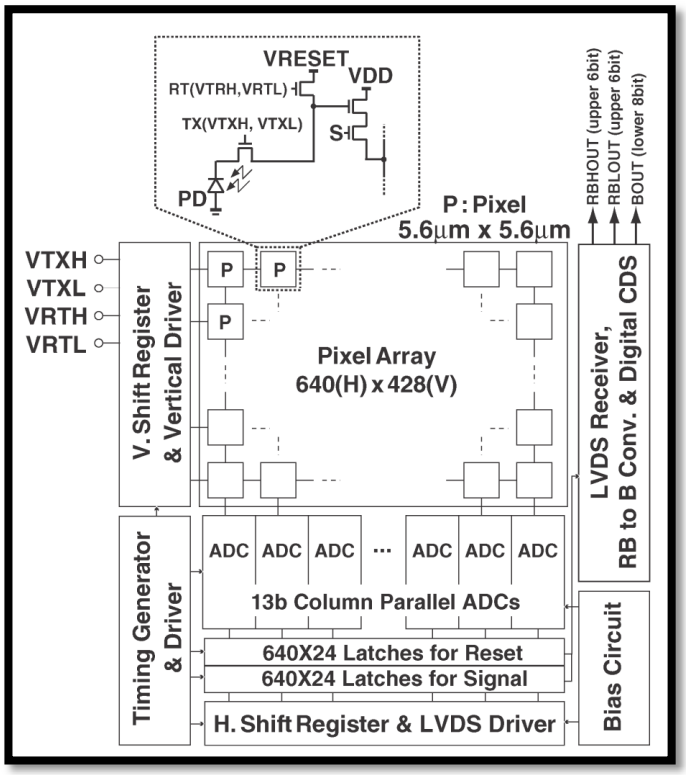


Figure 1-1 - Capteur d'image CMOS APS avec conversion à colonne parallèle (2)

D'après cette brève description, on peut déduire déjà quelques exigences aux CANs à colonnes parallèles. En plus des besoins de faible consommation, de faible niveau de bruit, etc., le convertisseur d'une architecture à colonnes parallèles doit occuper une surface étroite – normalement on limite sa largeur au pas de trame (*pitch*) du pixel – et doit avoir un taux de conversion au moins aussi vite que le taux d'images par seconde multiplié par le nombre des lignes du capteur. En outre, la résolution du convertisseur est définie par la sensibilité du capteur et par la profondeur de couleur exigée par l'application ciblée.

1.2 Convertisseurs Analogique-Numerique (CANs)

Actuellement, on entend beaucoup parler d'une transposition du domaine analogique vers le domaine numérique. En effet, le domaine numérique présente de nombreux avantages sur le domaine analogique dans le contexte du traitement de signal. Le traitement numérique du signal (DSP, *Digital Signal Processing*) est plus facile et moins cher à implémenter que le traitement analogique, en plus d'être moins sensible au bruit. Enfin, le DSP est aussi plus flexible puisqu'une modification d'algorithme numérique n'exige qu'une reprogrammation logicielle du circuit, alors qu'une modification dans le traitement analogique du signal exige une modification du hardware (5). Néanmoins, tous les signaux que l'on rencontre dans la nature sont de caractère analogique. Par conséquent, il est évident le traitement analogique ne disparaîtra jamais. Enfin, en partant

de l'avantage du traitement numérique et de la présence des signaux analogiques, le besoin d'un convertisseur entre les deux domaines est incontournable.

Les convertisseurs Numérique/Analogique et Analogique/Numérique ont été bien explorés pendant les dernières dizaines d'années. Suite à l'évolution des procédés de fabrication des circuits intégrés, à l'émergence de nouvelles applications, l'étude de ces circuits reste toujours d'actualité. Notamment, l'augmentation des performances (résolution, vitesse de conversion, puissance, surface, etc.) des CANs et CNAs impose des défis aux concepteurs analogiques.

En ce qui concerne ce travail, on ne discutera que des CANs, mais les concepts basiques de conversion A/N sont facilement transposables à la conversion N/A.

Un CAN est un système qui permet prélever périodiquement des échantillons d'un signal analogique quelconque, et de le quantifier (une valeur discrète est alors affectée à l'échantillon). Selon l'application visée, les caractéristiques du CAN varient à la fois en terme de précision, résolution (linéarité), vitesse d'échantillonnage et puissance consommée (6). Dans ce paragraphe, on décrira la vue d'ensemble d'un CAN, puis on discutera brièvement des topologies de CAN et finalement on parlera des signaux différentiels et de leur importance pour l'implémentation des CANs.

1.2.1 Caractéristiques générales d'un CAN

Tout d'abord, avant de discuter des topologies de CAN, on introduira quelques termes souvent utilisés pour décrire les performances des convertisseurs, comme les termes résolution, LSB, MSB, erreur de quantification, INL et DNL, SQNR, SNDR, ENOB et SFDR.

La résolution d'un CAN est définie par le nombre des différents niveaux analogiques correspondants à différents résultats numériques. Un CAN de N bits peut convertir 2^N niveaux analogiques distincts, donc on classifie un tel CAN comme possédant une résolution de N bits.

Par rapport à la résolution, on définit la mesure discrète la plus petite de la conversion A/N comme le pas de quantification q. Cette définition est aussi appelée V_{LSB} (*Least Significant Bit*) puisque cette variation analogique à l'entrée correspond au changement du bit le moins significatif de la sortie. LSB est utilisé pour désigner le bit conventionnellement à l'extrême droite d'un mot numérique, notamment de la sortie du CAN. D'ailleurs, le terme LSB est aussi couramment utilisé pour discuter et comparer les performances de plusieurs CANs. Ainsi :

$$q = \frac{FS}{2^N} = V_{LSB}$$

où FS signifie *pleine échelle* (en anglais *Full Scale*) et représente la différence entre les niveaux analogiques maximum et minimum.

De manière analogque on désigne par MSB (*Most Significant Bit*) le bit le plus significatif du mot numérique, celui de l'extrême gauche. Il correspond à la plus grande variation analogique, précisément la moitié de la pleine échelle, mais cette variation toute seule n'est pas utilisée durant la caractérisation des convertisseurs.

Ensuite, il existe plusieurs manières de décrire la performance d'un CAN, chacune possédant une particularité. Afin de montrer le comportement général statique d'un CAN, on présente ci-dessous une fonction de transfert idéale d'un CAN de 3 bits de résolution. L'axe horizontal représente l'entrée analogique du convertisseur et l'axe vertical sa sortie numérique. La fonction idéalement est un escalier uniforme comme montre la Figure 1-2.

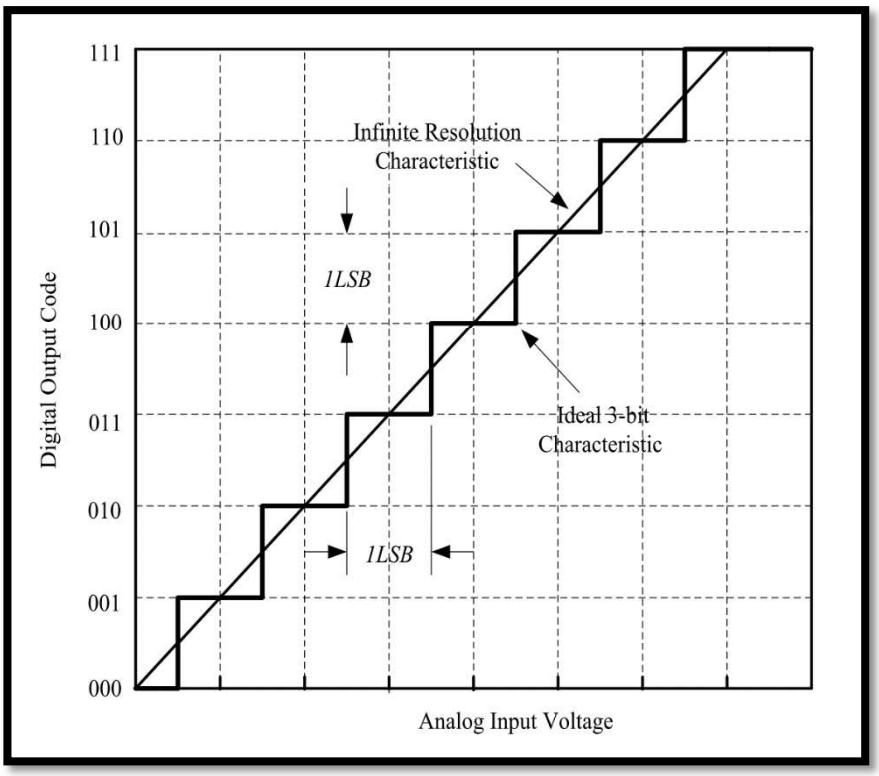


Figure 1-2 - Fonction de Transfert d'un CAN de 3 bits (7)

Ensuite, on définit aussi l'erreur de quantification comme l'erreur intrinsèque à une conversion A/N, une fois que la sortie est discrète alors que l'entrée est continue. La courbe d'erreur de quantification est produite à partir de la différence entre la fonction de transfert idéal (d'un CAN de résolution infinie) et la fonction de transfert du CAN. Pour le cas du CAN idéal de 3 bits de résolution l'erreur de quantification est présentée sur la Figure 1-3.

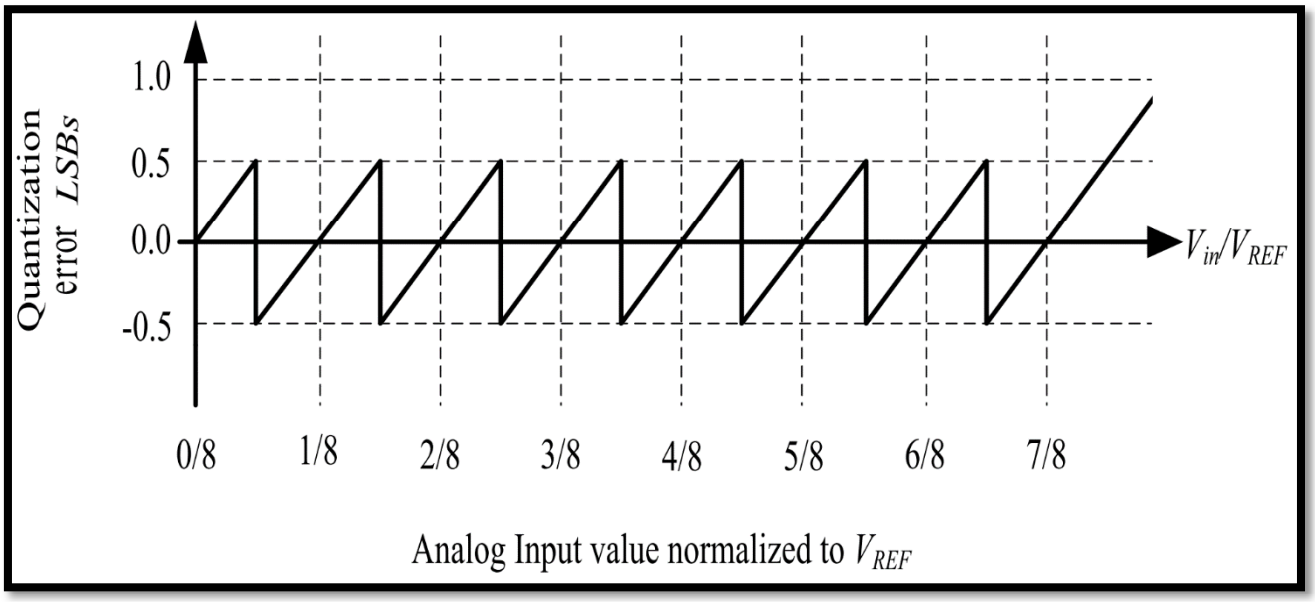


Figure 1-3 - Erreur de quantification d'un CAN 3 bits (7)

Lorsque l'on analyse un CAN non-idéal, outre l'inévitable erreur de quantification, d'autres erreurs apparaissent. Ces erreurs peuvent être mesurées de façon statique ou dynamique. On en présente brièvement quelques-unes.

A partir de la fonction de transfert statique d'un CAN, tout d'abord on extrait les erreurs 'linéaires', soit l'erreur d'*offset* et l'erreur de gain.

L'erreur d'*offset* est calculée par l'équation ci-dessous, où $V_{0...01}$ est la valeur du premier seuil de tension du CAN :

$$E_{off} = \frac{V_{0...01}}{V_{LSB}} - \frac{1}{2}LSB$$

L'erreur de gain, à son tour, est définie par l'équation ci-dessous, où, de façon similaire, $V_{1...1}$ est la valeur du dernier seuil de tension :

$$E_{gain} = \left(\frac{V_{1...1}}{V_{LSB}} - \frac{V_{0...01}}{V_{LSB}}\right) - (2^N - 2)$$

Ces deux erreurs sont interprétées comme la déviation horizontale et la déviation de la pente d'une ligne droite extraite de la fonction de transfert du CAN, comme montre la Figure 1-4.

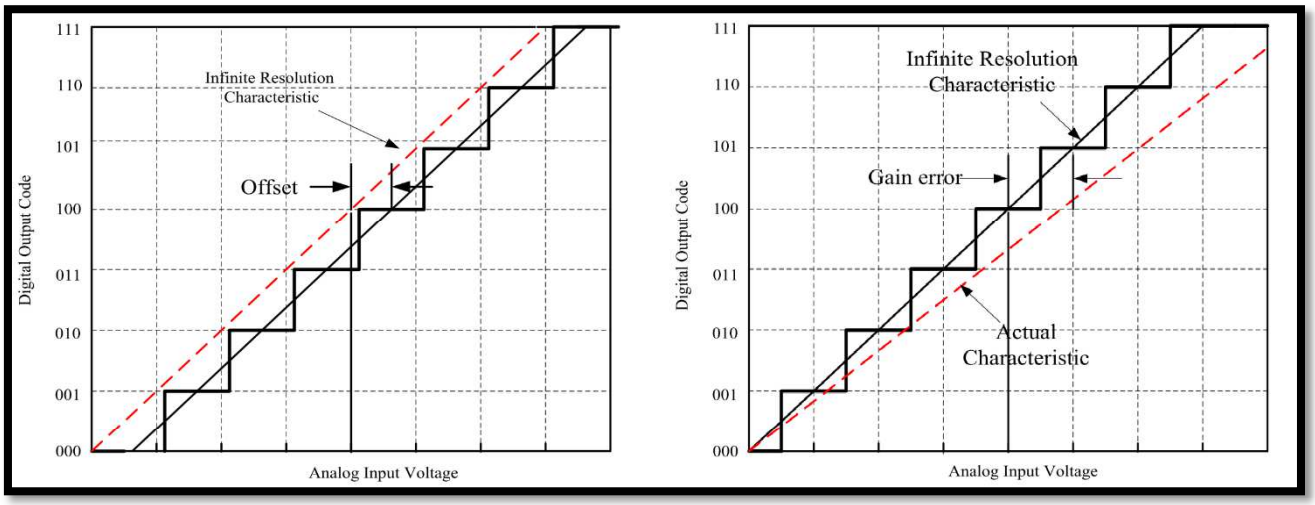


Figure 1-4 – Interprétation des erreurs d'*offset* et de gain (7)

Egalement, on définit deux erreurs nommées non-linéaires : INL et DNL.

INL (*Integral Nonlinearity Error*) est la déviation de la fonction de transfert par rapport une ligne droite après la correction des erreurs d'*offset* et de gain. Pour que ces deux erreurs soient corrigées, couramment cette droite est créée en reliant les points de transition extrêmes de la fonction de transfert du CAN. On remarque que souvent l'INL est exprimée en LSB's et que quelques auteurs utilisent le terme INL comme la valeur maximale de l'erreur que l'on a décrit.

DNL (*Differential Nonlinearity Error*) est la déviation d'un pas analogique par rapport la valeur idéale, égale à 1 V_{LSB}. Comme l'INL, le DNL est aussi couramment exprimée en LSB et quelques auteurs l'utilisent comme la valeur d'erreur maximale.

$$DNL = Largeur\ du\ pas\ réel - Largeur\ du\ pas\ idéel$$

Ensuite, il existe aussi les mesures dites dynamiques. Tout d'abord le SQNR (*Signal to Quantization-Noise Ratio*) est le rapport signal sur bruit idéal d'un CAN puisqu'il ne prend en compte que le bruit de quantification. SQNR est définit comme le rapport de la tension RMS d'un signal sinusoïdal à l'entrée du CAN sur la tension RMS du bruit de quantification. Son résultat est donné en dB et peut être exprimé en fonction du nombre de bits :

$$SQNR = 6.02N + 1.76\ dB$$

Le bruit de quantification, provenant de l'erreur de quantification, est un phénomène attendu dans la théorie de CAN. Or, durant les mesures de performance d'un CAN réel, on rencontre d'autres sources de non-idéalités, ce qui peut dégrader le rapport signal sur bruit. Pour mesurer ce rapport, il suffit d'injecter un signal sinusoïdal à l'entrée du CAN et de réaliser une FFT à partir des résultats de conversion. Ensuite, on calcule le

rapport entre le signal d'entrée et le signal de sortie en excluant la raie de l'harmonique principale. Ce rapport mesuré est appelé SNDR (*Signal to Noise and Distortion Ratio*).

En remplaçant SQNR par SNDR dans l'équation ci-dessus, on trouve une autre définition courante dans la littérature appelé ENOB (*Effective Number of Bits*). Ce paramètre nous informe dans des conditions réales quelle est le nombre de bits effectif pour une qualité de signal semblable au cas d'un CAN idéal.

$$SNDR = 6.02\,ENOB + 1.76\,dB$$

$$ENOB = \frac{SNDR - 1.76\,dB}{6.02}$$

Finalement on définit le SFDR (*Spurious-Free Dynamic Range*), toujours à l'aide du spectre de sortie du CAN, comme le rapport de l'harmonique principale sur le signal non-désiré le plus fort. Le SFDR est couramment exprimé en dBc.

1.2.2 Classification des CAN

Les CANs peuvent être classés en deux types de convertisseurs : les Convertisseurs de type *Nyquist-rate* et les Convertisseurs à sur-échantillonnage.

Les convertisseurs de type *Nyquist-rate* sont définis en gros comme des convertisseurs générant un signal de sortie dont chaque valeur possède une correspondance unique avec une valeur du signal d'entrée. Il faut remarquer que, malgré son nom, un convertisseur de type *Nyquist-rate* est rarement utilisé sous un échantillonnage égal à la limite du principe de Shannon-Nyquist, en raison de la difficulté pratique de réaliser des filtres anti-repliement de spectre et de reconstruction. De toute façon, le taux d'échantillonnage varie normalement entre 1.5 et 10 fois la fréquence de Nyquist du signal d'entrée.

Les convertisseurs à sur-échantillonnage sont, à leur tour, ceux qui opèrent à partir d'une fréquence beaucoup plus haute que la fréquence de Nyquist du signal d'entrée, typiquement entre 10 et 500 fois celle-là, et augmentent le rapport signal sur bruit en filtrant le bruit de quantification situé en dehors de la bande de fréquence du signal ciblé. Ces convertisseurs utilisent normalement des filtres numériques et la technique *noise shaping*. Ces convertisseurs sont habituellement appelés Delta-Sigma (ΔΣ).

Parmi les convertisseurs analogique/numérique, on liste ci-dessous les plus souvent rencontrés sur la littérature en mettant en évidence quelques caractéristiques particulières mais sans les beaucoup détailler. On remarque que, en dehors des convertisseurs Delta-Sigma, tous les CANs mentionnés ci-dessous sont des convertisseurs de type *Nyquist rate* :

- Convertisseurs à Intégration. Normalement appelés CAN à rampe ou double-rampe, ces convertisseurs réalisent la conversion à l'aide du temps en intégrant le niveau de tension instantané de l'entrée (considéré constant durant la conversion). Dans ce but, des intégrateurs et des compteurs sont utilisés. Les Convertisseurs à Intégration présentent l'avantage d'avoir des erreurs de gain et d'*offset* très faibles en plus d'une petite surface. En revanche ils ne peuvent être utilisés que dans des applications de signaux très lents.

- Convertisseurs à Approximations Successives (*SAR*). Les SARs réalisent une conversion A/N à partir du principe de la dichotomie, en comparant la valeur échantillonnée du signal d'entrée avec plusieurs valeurs discrètes –générées par un CNA interne – qui s'approchent graduellement du signal quantifié en plusieurs étapes. Ces convertisseurs sont capables de réaliser une conversion de vitesse et précision moyennes. Ils sont avantageux en tant que convertisseurs simples, qui utilisent un relativement petit nombre de composants et qui consomment peu. Le principal inconvénient des SARs est l'exigence de plusieurs cycles d'horloge pour réaliser une seule conversion, puisque la dichotomie demande des itérations.

- Convertisseurs Cycliques (ou Algorithmiques). Les convertisseurs cycliques sont caractérisés par la réalisation de plusieurs conversions d'un seul bit enchaînées. Ainsi, l'une après l'autre, les conversions récupèrent les bits du plus significatif jusqu'au moins significatif. Ces convertisseurs sont similaires aux SARs, sauf qu'alors que ceux-ci exigent un CNA précis, les convertisseurs cycliques exigent une soustraction et une multiplication par deux précises. Plus de détails sur le fonctionnement de ce type de CAN seront données sur le chapitre 2. Les convertisseurs cycliques possèdent conventionnellement une vitesse et une précision de conversion moyennes, et n'ont pas besoin de nombreux composants pour leur implémentation, ce qui signifie une petite surface occupée. Aussi comme les SAR, les CAN Cycliques exigent plusieurs cycles d'horloge pour une seule conversion.

- Convertisseur Pipeline. Ces convertisseurs réalisent les mêmes opérations qu'un convertisseur cyclique, mais au lieu de boucler le signal afin d'exécuter nombreuses itérations, les convertisseurs pipeline répliquent spatialement le même circuit plusieurs fois en cascade. Ainsi, malgré le même délai du cyclique, le convertisseur pipeline complète une conversion à chaque cycle d'horloge. Par conséquent, le convertisseur pipeline parvient à une grande vitesse de conversion avec une précision moyenne. En revanche, l'augmentation du nombre de circuits augmente aussi la surface et la puissance consommées par le CAN.

- Convertisseurs Flash. Les convertisseurs flash réalisent la conversion A/N en comparant directement la tension d'entrée avec plusieurs références de tension connues. Ces convertisseurs sont normalement les convertisseurs les plus rapides, où chaque coup d'horloge conduit à une nouvelle conversion sans délai. Cependant les convertisseurs Flash sont limités par une faible résolution. En effet, l'augmentation de bits de résolution entraîne l'augmentation sévère de la surface occupée et de la puissance consommée. En outre, la résolution est rapidement limitée car elle est directement dépendante de la précision du procédé de fabrication.

- Convertisseur semi-flash (*Two-step ADC*). Ce convertisseur peut être considéré comme un CAN pipeline à deux étages où chacun possède un CAN flash extrayant plusieurs bits. Dans un CAN semi flash de n bits, le premier CAN flash, de *j*-bit par exemple, obtient les *j* bits le plus significatifs et le deuxième est responsable des autres $n - j$ bits. Entre ces deux CANs internes cependant, le signal d'entrée est soustrait de la valeur quantifiée par le premier CAN et multiplié par 2^j, de façon similaire au CAN pipeline lorsque *j* est égal à 1. Le CAN semi-flash, en dépit d'être plus lent que le flash, possède quelques avantages car il permet une surface occupée plus petite, une puissance consommée plus faible et un relachement d'autres contraintes.

- Convertisseur à interpolation. Ce convertisseur suit le principe d'un CAN flash, mais il utilise moins de comparateurs à l'entrée de telle sorte que la charge capacitive soit réduite, en diminuant la consommation totale. Or, une réduction au nombre de comparateurs signifie au départ une réduction de niveaux de quantification. Les niveaux exclus, néanmoins, sont récupérés après les comparateurs à travers une interpolation – d'où le nom du convertisseur - réalisée, par exemple, par une chaîne de résistances. Les CANs à interpolation, outre la simplification des circuits, présentent par rapport aux CANs flash des avantages en termes de de surface occupée et de consommation.

- Convertisseur à repliement (*Folding*). Ce convertisseur part toujours du principe d'un CAN flash. Son objectif est aussi de réduire la surface occupée et la consommation en comparaison au CAN flash. Son amélioration comparée au CAN à interpolation est la réduction de bascules. Cependant, les CANs à repliement sont souvent conçus en partant aussi du principe du CAN à interpolation, pour que leurs optimisations soient combinées.

- Convertisseur à entrelacement temporel (*time-interleaved*). Ce convertisseur peut atteindre une haute vitesse de conversion à travers l'utilisation de plusieurs CANs

plus lents en parallèle opérant à tour de rôle. La principale entrave de ce CAN est la synchronisation des CANs utilisés. Les décalages imprécis d'horloge entre les CAN dans cette topologie deviennent le point le plus critique à la performance ciblée. La topologie de chaque CAN interne doit être choisie d'après l'application ciblée. Les CANs à entrelacement temporel accomplissent une haute vitesse en dépit d'une augmentation de surface occupée et de consommation de puissance.

- Convertisseurs Sigma-Delta. Ces convertisseurs sont souvent rencontrés dans les applications qui ne demandent pas une conversion rapide mais qui exigent une bonne résolution. Son utilisation est actuellement courante dû au fait que, dans les CANs ΔΣ, les contraintes analogiques sont relaxées en dépit d'un circuit numérique plus complexe. Au fur et à mesure que la technologie submicronique se développe l'utilisation des CAN ΔΣ devient de plus en plus avantageuse. L'inconvénient de cette topologie, comme déjà cité, est la faible vitesse de fonctionnement.

Afin de comparer les performances des CANs mentionnés, le Tableau 1-1 et la Figure 1-5 présentent à la fois la vitesse et la résolution de quelques CANs. Ce type de comparaison est souvent rencontré et est bien utile aux concepteurs lorsqu'ils doivent réaliser le choix d'un convertisseur pour son application.

Tableau 1-1 – Comparaison entre les CANs

Basse Vitesse, Haute résolution	**Vitesse Moyenne, Résolution Moyenne**	**Haute Vitesse, Faible résolution**
A intégration Sur-échantillonné	Approximations Successives Cyclique	Flash Semi-flash A Interpolation A repliement A entrelacement temporal Pipeline

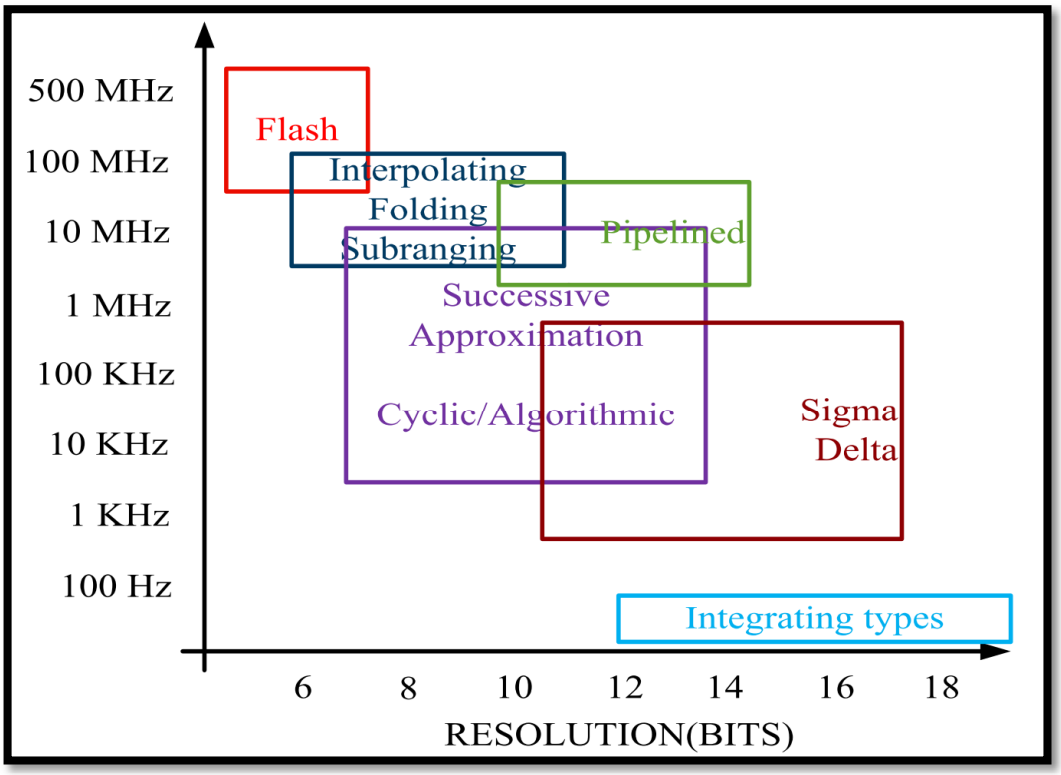


Figure 1-5 – Résolution X Vitesse de conversion

1.2.3 Types de Signaux d'Entrées

Malgré la même fonctionnalité et le même principe d'opération, un CAN peut être implémenté différemment afin de manipuler des signaux analogiques semblables représentés de manières différentes. On parle de signaux *single-ended*, *pseudo-differential* et *fully-differential* (8). Souvent la littérature présente des circuits électroniques utilisant des entrées *single-ended* afin d'être plus claire et synthétique durant l'explication de son fonctionnement. Or, en pratique, les signaux *differential* sont plus souvent choisis car ils présentet des quelques avantages que ceux-là possèdent sur les *single-ended*.

Un CAN à entrées *single-ended* numérise une tension d'entrée analogique en la référençant à la masse. Les signaux *single-ended* simplifient les exigences du *driver*, réduisent la complexité des circuits et baissent la consommation de puissance du CAN. En outre, les signaux *single-ended* peuvent être classés comme unipolaire ou vrai bipolaire, lorsqu'ils se confinent entre la masse (GND) et une tension préfixée (V_{ref}) ou au-dessus et au-dessous de la masse entre les tensions $-V_{ref}$ et $+V_{ref}$. La Figure 1-6 montre ces deux représentations.

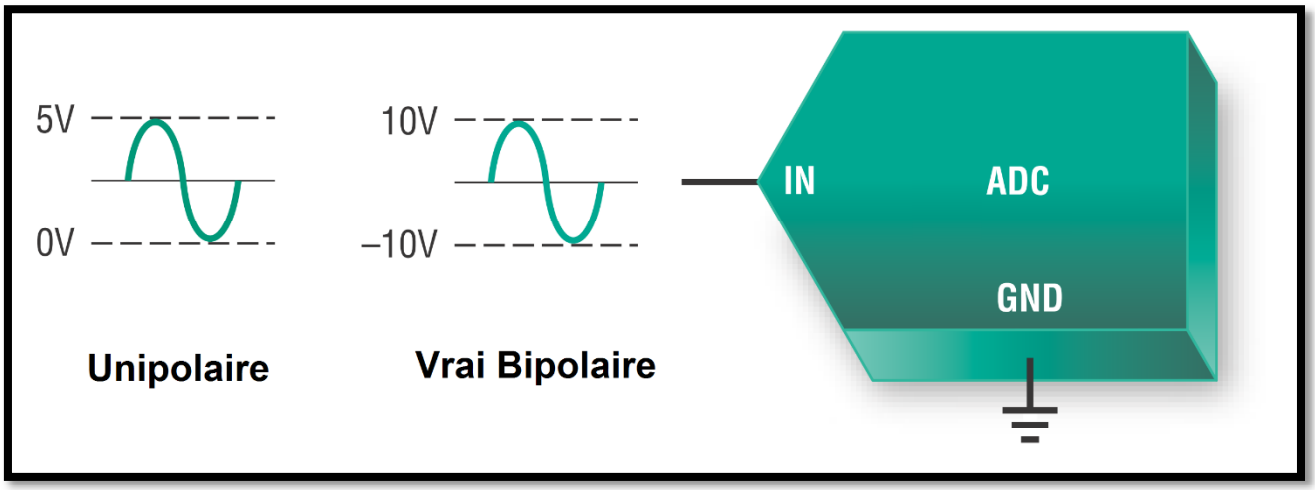


Figure 1-6 - CANs à entrées *single-ended* (8)

Un CAN à entrées *pseudo-differential* numérise la tension entre ses deux portes d'entrées IN^+ et IN^-. Cependant, ce n'est que la porte IN^+ qui possède le signal utile, alors que le signal à la porte IN^- ne sert que de référence. Ainsi, trois catégories de signaux *pseudo-differential* sont définies : unipolaire, bipolaire et vrai bipolaire. Dans le cas d'un CAN à signaux *pseudo-differential* unipolaires, un signal *single-ended* unipolaire est entrainé sur l'entrée IN^+ alors que l'entrée IN^- est reliée à la masse. Pour un CAN *pseudo-differential* bipolaire, le même signal unipolaire est présent sur l'entrée IN^+ mais l'entrée IN^- est reliée à la tension moyenne du signal unipolaire. Enfin, sur un CAN *pseudo-differential* vrai bipolaire, l'entrée IN^+ est reliée à un signal vrai bipolaire et l'entrée IN^- est reliée à la masse. La Figure 1-7 résume les CANs *pseudo-differential.*

Les signaux *pseudo-differential* aident à isoler la masse du signal de la masse du CAN, ce qui peut être visée par quelques applications. Il s'agit d'applicationsqui exigent une bonne tolérance à la tension de mode commun DC du signal d'entrée ou celles qui désirent profiter des caractéristiques d'un signal différentiel mais préfèrent éviter la complexité des circuits *fully-differential*.

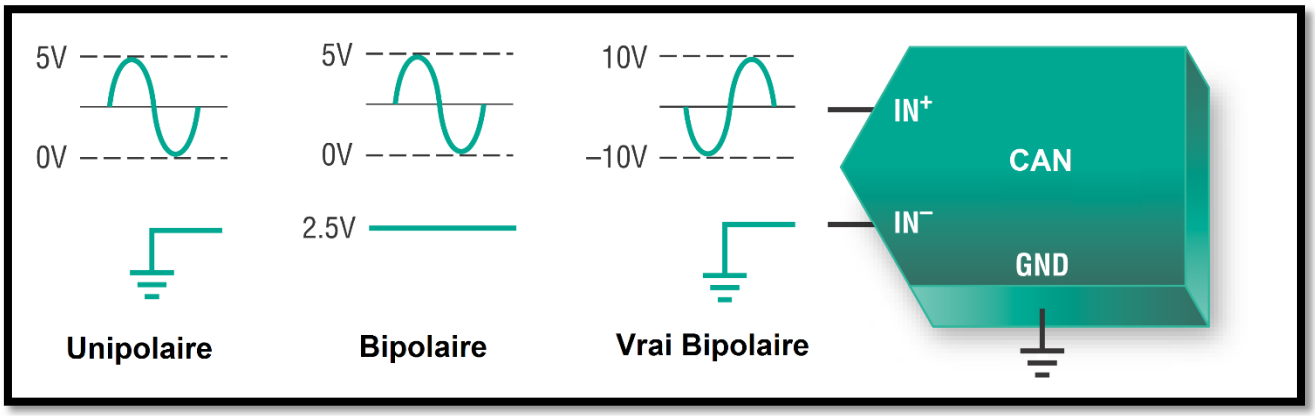


Figure 1-7 - CAN à entrées *pseudo-differential* (8)

Finalement, un CAN à entrées *fully-differential* numérise la tension entre ses deux portes d'entrées IN^+ et IN^-, mais cette fois-ci le signal visé est présent dans les deux portes de façon complémentaire, c'est-à-dire, déphasés de 180° l'un par rapport à l'autre. La tension moyenne pour les deux signaux est appelée mode commun. A nouveau, les signaux *fully-differential* peuvent être séparés en deux catégories : unipolaire et vrai bipolaire. Les CANs *fully-differential* unipolaires sont caractérisés par ses deux entrées *single-ended* unipolaires déphasées de 180°. Les CANs *fully-differential* vrai bipolaires, à leur tour, sont caractérisés par leurs deux entrées *single-ended* bipolaires, aussi déphasées de 180° l'une par rapport à l'autre. Ces deux catégories sont résumées sur la Figure 18.

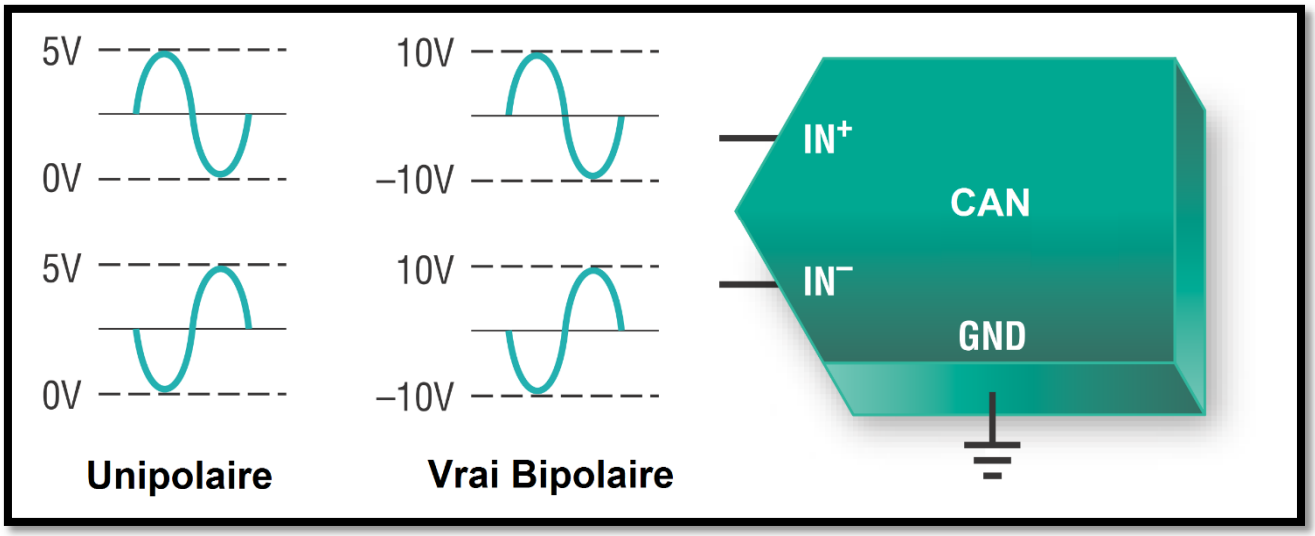


Figure 18 - CAN à entrées *fully-differential* (8)

L'avantage des signaux *fully-differential* est facile à comprendre. Tout d'abord, comme les signaux d'entrées sont déphasés de 180° et que le signal détecté par le circuit est la différence entre ces deux signaux, la gamme dynamique est dupliquée pour une même excursion de tension de chaque signal *single-ended*. Ensuite, les circuits *fully-*

differential sont plus robustes aux distorsions, surtout au bruit de mode commun et à la distorsion harmonique d'ordre pair, une fois que ces distorsions apparaissent aux deux entrées de façon pareille et ne sont pas prises en compte par le circuit d'entrée (9).

1.3 SPICE

Déterminer le comportement de circuits électriques à partir de leurs schémas exige la résolution simultanée de nombreuses équations. Trouver la solution DC pour un point de polarisation est un problème simple dès que le circuit est linéaire. Si le circuit est simple, les lois de courant et tension de Kirchhoff et les équations constitutives des éléments peuvent être résolues à la main en parvenant ainsi à une solution unique et exacte. Lorsque le circuit devient plus important et non-linéaire, l'analyse DC n'est plus évidente. Si on a besoin d'une autre analyse – AC ou temporelle, par exemple – la solution se trouve encore plus difficilement. D'où la nécessité d'une autre type de solution que celle analytique. Une solution prometteuse vient des hypothèses d'approximations, comme celle de petits signaux, et de l'analyse numérique, avec le développement de divers algorithmes de résolution de systèmes linéaires, d'équations intégration-différentielles, etc. Néanmoins, malgré des résultats satisfaisants, la complexité calculatoire est trop importante .

Outre l'avancée des algorithmes d'analyse numérique, un autre développement primordial pour la conception de circuits électroniques a été celui des ordinateurs et des programmes de calcul associés. La technologie numérique présente dans les ordinateurs, de plus en plus efficaces et accessibles au public, accompagnée des algorithmes de résolution de circuits, a rendu possible les simulations au service de l'analyse et de la conception. Parmi les types de simulateurs, on met en évidence dans ce rapport les simulateurs électriques de type SPICE.

SPICE (*Simulation Program with Integrated Circuit Emphasis*) est un simulateur de circuits électriques polyvalent pour les analyses DC non-linéaires, temporelles non-linéaires et AC linéaires. Il résout les équations du circuit envisagé à partir des tensions nodales. Ce simulateur est approprié pour des circuits linéaires et non-linéaires de nombreuses applications, depuis les références de tension/courant jusqu'aux cellules de mémoire RAM par exemple. Les circuits peuvent contenir des résistances, des capacités, inductances, des sources indépendantes et dépendantes de tension ou de courant, des lignes de transmission ou encore les composants semi-conducteurs les plus courants, comme les diodes, TBJs, JFETs, MOSFETs et MESFETs (10).

Le programme maintenant connu sous le nom de SPICE a été lancé tout d'abord sous le nom de CANCER (*Computer Analysis of Nonlinear Circuits, Excluding Radiation*) en

1970 et a acquis le nom SPICE1 en 1972, grâce aux travaux d'une équipe de chercheurs du Département d'Ingénierie Electrique et Informatique de l'Université de Californie à Berkley. Son acceptation a été bonne étant donné les modèles semi-conducteurs utilisés. En 1975, la deuxième version du simulateur a été lancée avec beaucoup d'améliorations par rapport à l'antérieure. Cette version, appelée SPICE2, a subi diverses mises à jour et est une version utilisée encore aujourd'hui par une grande partie des simulateurs électriques. Une troisième version a été encore diffusée en 1985 avec le nom SPICE3. La rupture entre cette version et SPICE2 était le langage de programmation avec laquelle on a écrit le simulateur. SPICE2, écrit en FORTRAN, avait posé des difficultés aux développeurs utilisateurs de la plateforme UNIX qui ne pouvaient pas utiliser toutes les potentialités du *C-Shell*. Ainsi, le principal objectif du SPICE3 était de traduire le simulateur SPICE2 vers le langage C, en ajoutant quelques perfectionnements. Actuellement, tous les logiciels de conception analogique assistée par ordinateur sont basés sur SPICE2 ou SPICE3, mais avec des renforcements particuliers ajoutés et bien soutenus. (10)

Les analyses les plus connus et utilisées sur SPICE sont l'analyse DC, l'analyse AC et l'analyse temporelle, brièvement expliquées ci-dessous :

- L'analyse DC calcule le point de polarisation du circuit en remplaçant les inductances par des court-circuit et les capacités par des circuits ouverts. SPICE peut utiliser plusieurs itérations pour résoudre les équations de circuits non-linéaires.
- L'analyse AC à son tour calcule les valeurs complexes de tension en chaque nœud en fonction de la fréquence d'une sinusoïde à l'entrée du circuit. Ce calcul, cependant, est réalisé après la linéarisation du circuit autour de son point de polarisation suite à l'hypothèse de petits signaux.
- L'analyse temporelle calcule les tensions de chaque nœud du circuit en fonction du temps. Cette fois-ci l'analyse ne possède aucune restriction de niveau de signal. Par conséquent, on dit qu'il s'agit d'une analyse de grands signaux où toutes les non-linéarités sont prises en compte.

Suite à l'évolution du simulateur SPICE, plusieurs sociétés, liées à fabricants de circuits intégrés, ont développé leurs propres logiciels basés sur SPICE. Habituellement ces logiciels sont payés, mais bien suivis et maintenus par les sociétés. De plus, les modèles fournis favorisent normalement les sociétés propriétaires du logiciel. Parmi ces nombreux logiciels disponibles, dans le cadre de ce projet, il a été choisi le logiciel LTSPICE, développé par Linear Technology.

LTSPICE est un simulateur basé sur la version SPICE 3f4/5 (deux patches de la version SPICE3). Outre le simulateur, LTSPICE possède aussi une interface graphique pour la conception de schémas, l'analyse des courbes de simulation, etc. Suite à son utilisation bien répandue, LTSPICE est considéré comme une plateforme stable et sure. Egalement, le logiciel est continuellement mis à jour puisque les utilisateurs de LTSPICE maintiennent sur internet des groupes de discussion que sont ouverts et accompagnés par les développeurs.

Dans ce travail on a utilisé surtout la version LTSPICE 4.18x.

1.4 MACRO-MODELISATION

On a vu dans la section antérieure comment le simulateur SPICE est devenu populaire grâce à la recherche de modèles semi-conducteurs cohérents avec la réalité des circuits intégrés. Cependant, au fur et à mesure que les circuits devenaient de plus en plus complexes et étendus, les simulations devenaient aussi lentes. La simulation d'un système possédant un nombre raisonnable de composants semi-conducteurs pouvait prendre un temps prohibitif. D'où une nouvelle approche utilisant SPICE à travers une autre type de modélisation appelée macro-modélisation.

Les macro-modèles combinent des éléments fonctionnels avec des modèles non-linéaires précis afin d'obtenir un circuit comportemental qui possède la même réponse que le circuit original. Or, vu que le circuit modélisé est beaucoup moins complexe que le circuit original, ce nouveau circuit est simulé plusieurs fois plus rapide (10).

Le concept de macro-modélisation a été appliqué par la première fois pour les Amplificateur Opérationnels (11) qui peuvent se composer d'une dizaine à une centaine de transistors sur une représentation détaillée. La popularité des AmpOps a débuté dans les années 1960, lorsque les circuits intégrés ont été introduits, et ont été utilisés dans un grand nombre d'applications. Toutefois, à nouveau, l'utilisation de deux AmpOps est déjà suffisante pour effectuer la simulation dans des temps prohibitifs, à cause de la complexité calculatoireexigée par le simulateur. Actuellement la macro-modélisation est l'approche principale pour représenter des circuits équivalents SPICE pour une grande variété de circuits intégrés complexes (10).

Dans l'intention de réaliser la macro-modélisation, quelques techniques peuvent être utilisées. Par exemple, sur (11) les auteurs décrivent deux techniques : *simplification* et *build-up*. La première technique consiste à remplacer une partie du circuit réel par des éléments idéaux. La deuxième technique utilise aussi des éléments idéaux pour accorder le circuit à quelques spécifications, mais le résultat ne ressemble forcément pas au circuit original. Des techniques similaires ont été aussi utilisées durant le travail décrit par ce rapport.

2. CAN CYCLIQUE

Les CAN Cycliques, aussi appelés Algorithmiques, réalisent une conversion A/N complète à partir de plusieurs conversions successsives d'un seul bit. Son avantage principal est que le CAN ne comprend qu'un seul circuit pour toutes ces *sous-conversions*, puisque le signal sortant de ce circuit est rebouclé à son entrée. Ainsi, outre l'avantage d'une surface et d'un coût réduits, une calibration numérique est plus facilement réalisée puisque un nombre plus petit de transistors va généralement de pair avec une quantité plus réduite de sources d'erreurs.

Dans ce chapitre, les signaux analogiques sont implémentés comme des signaux *fully-differentials* unipolaires. Plus précisément, les signaux *single-ended* resteront entre 0 et $+V_{ref}$ avec une tension de mode commun égale à $V_{ref}/2$ et la tension différentielle variera entre $-V_{ref}$ et $+V_{ref}$. Cependant, lors des explications, les figures et les schémas montreront plutôt des signaux *single-ended* vrai bipolaires pour que la compréhension soit plus claire. En outre, N représentera le nombre de bits du convertisseur A/N alors que d_i est l'i-iéme bit du mot numérique converti.

Tout d'abord, on expliquera le fonctionnement général d'un CAN Cyclique conventionnel, en présentant l'algorithme de 1-bit par étape, la fonction de transfert d'un cycle tout seul et les effets des non-idéalités les plus courants. Après on introduira l'algorithme RSD et le CAN Cyclique RSD, en présentant aussi sa fonction de transfert et ses avantages à l'égard des non-idéalités toujours présents. Finalement on montrera l'implémentation sur laquelle on s'est basé pour faire le travail décrit sur ce rapport, en spécifiant les circuits internes.

2.1 CAN CYCLIQUE CONVENTIONNEL

Dans un CAN Cyclique Conventionnel, la sortie d'un cycle est rebouclée à l'entrée pour qu'elle soit traitée durant le cycle prochain. A chaque cycle, un bit est déterminé en comparant l'entrée à une tension de seuil et la sortie, appelée résidu, est produite à partir de la multiplication par deux de la différence entre l'entrée et l'équivalent analogique du bit déterminé. Ainsi, afin de compléter une conversion A/N de N bits, N cycles sont exécutés.

Un CAN Cyclique comprend un échantillonneur/bloqueur (E/H), un amplificateur de résidus, un sous-CAN, un CNA et une unité logique synthétisant les bits extraits à chaque étape. Un schéma fonctionnel d'un CAN cyclique conventionnel est présenté sur la Figure 2-1. Comme chaque cycle extrait un bit, le sous-CAN d'un bit peut être implémenté par un Comparateur et le CNA par un multiplexeur d'un bit.

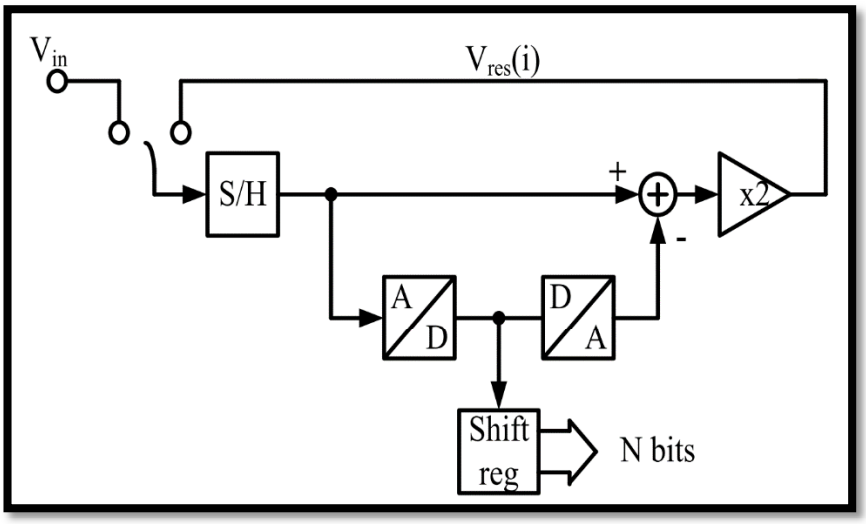


Figure 2-1 – Schéma fonctionnel d'un CAN Cyclique conventionnel

Lors du premier cycle, la tension d'entrée V_{IN} est à la fois échantillonnée et comparée à la tension nulle afin de trouver le MSB (d_{N-1}) : si $V_{IN} > 0$ alors MSB = 1, sinon MSB = 0. Ce bit sert comme entrée du CNA, qui évalue sa valeur analogique. Cette estimation est soustraite au signal original, maintenu par l'E/H, en générant le résidu du premier cycle. Pour que le résidu puisse être traité comme un nouveau signal à l'entrée du circuit, il est multiplié par deux pour revenir à la pleine échelle :

$$V_{res}(i+1) = 2 \times V_{res}(i) - b \times V_{ref}$$

où $V_{res}(i)$ est le résidu de l'i-iéme cycle, V_{ref} est la tension de référence du CAN et *b* est le résultat de la comparaison. V_{ref} est soustrait si $V_{res}(i)>0$ (b = 1) et est sommé si $V_{res}(i)<0$ (b = -1). On remarque que $V_{res}(1) = V_{IN}$.

Cette tension est rebouclée à l'entrée et l'opération se répète jusqu'à ce que tous les bits soient extraits. Puis l'E/H échantillonne l'entrée à nouveau et une autre conversion A/N commence. L'opération logique du CAN cyclique est présentée sur le logigramme de la Figure 2-2. Lorsque b=-1, le bit converti d est égal à 0, alors que quand b=+1, d=1.

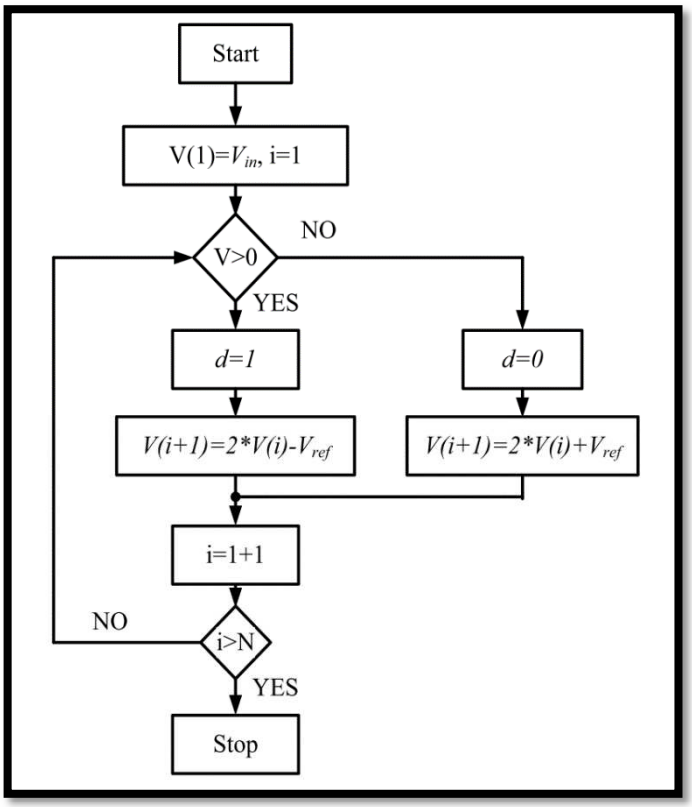


Figure 2-2 - Logigramme d'un CAN cyclique

Couramment, on utilise la courbe de la fonction de transfert d'un cycle du CAN pour analyser son fonctionnement. La fonction de transfert d'un cycle du CAN cyclique de 1-bit/étape est montrée sur la Figure 2-3. A travers cette figure, on peut explorer les effets des non-idéalités présentes dans le CAN.

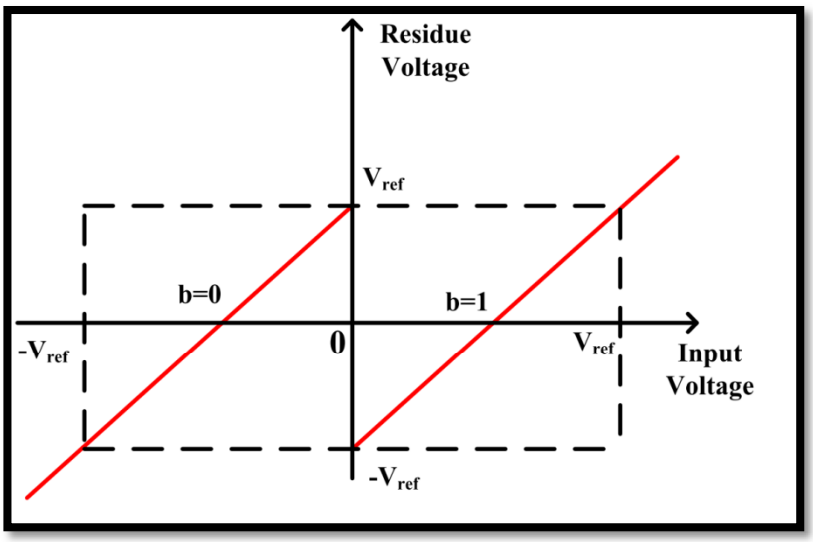


Figure 2-3 - Fonction de transfert d'un cycle d'un CAN 1-bit/étape

En général, les sources d'erreurs les plus critiques du CAN Cyclique sont les *offsets* tant du comparateur que de l'amplificateur de résidu, et le gain de l'amplificateur précisément égal à 2. On les discutera séparément.

Premièrement, l'effet créé par l'*offset* du comparateur est montré sur la Figure 2-4. Le seuil de tension à l'entrée est déplacé, créant ainsi une zone où le résidu est plus grand que V_{ref}. Cette non-idéalité n'est gênante que lorsque l'on est dans cette zone étroite. Si c'est le cas, une divergence se produit et tous les bits postérieurs sont coincés à un seul état de comparaison. C'est pourquoi on conclut que le comparateur doit être au moins aussi précis que la résolution du CAN, dans le cas d'un convertisseur cyclique conventionnel.

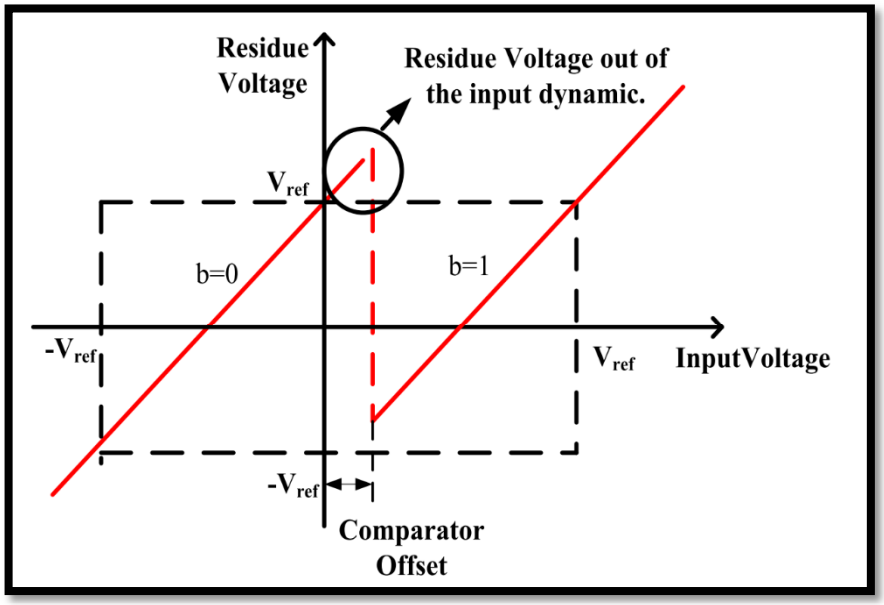


Figure 2-4 - Effet de l'*offset* du comparateur dans un CAN Cyclique conventionnel

L'*offset* présent dans l'amplificateur de gain égal à 2 crée l'effet sur la fonction de transfert présenté sur la Figure 2-5. La fonction de transfert entière est déplacée verticalement. A nouveau, des zones où la tension de résidu dépasse V_{ref} sont créées. Ainsi, le fonctionnement du CAN reste normal jusqu'au moment où une des tensions résidus tombe sur une de ces zones, ce qui entraîne une série d'erreurs systématiques dans la conversion A/N.

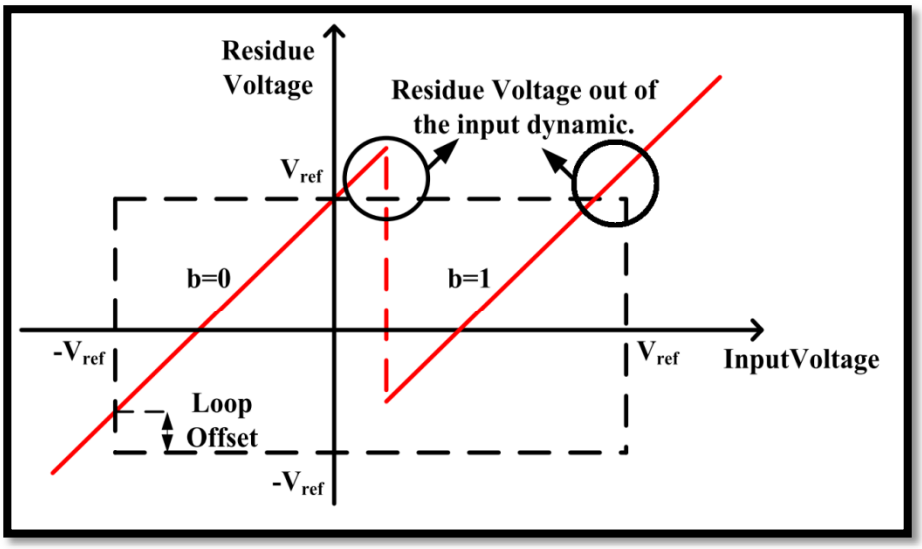


Figure 2-5 - Effet de l'offset de l'amplificateur dans un CAN Cyclique conventionnel

Finalement, l'erreur sur le gain de l'amplificateur est la non-idéalité la plus critique du CAN Cyclique (12). L'effet d'un gain constant légèrement différent de 2 est présenté sur la Figure 2-6. Toute la plage de tension d'entrée est déjà gênée mais, dans le cas où le gain est plus grand que 2, aux extrêmes et au seuil de tension, il apparaît des zones de non-convergence semblables à celles des *offsets*.

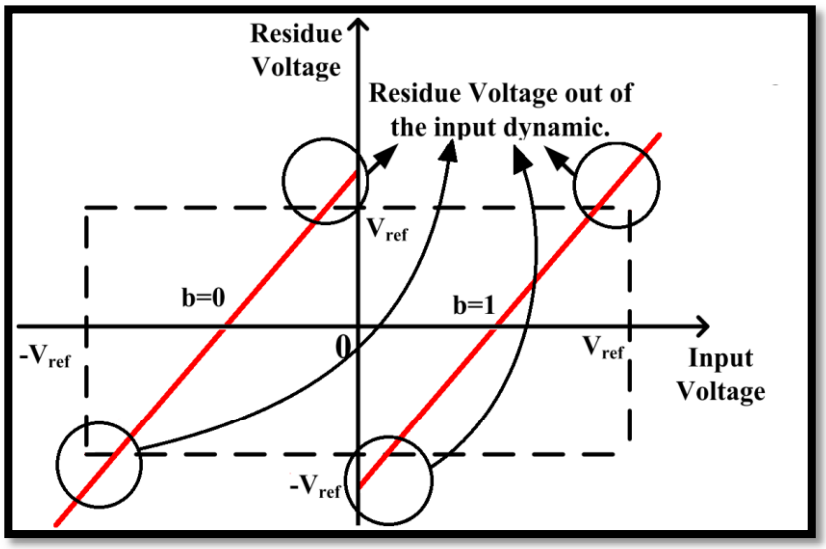


Figure 2-6 - Effet d'un gain constant différent de 2 sur l'amplificateur dans un CAN Cyclique conventionnel

Plusieurs techniques ont été rapportées dans le but de corriger ou d'atténuer ces erreurs. L'imprécision du gain est attaquée de manières différentes, comme par exemple en (13) (14) (15). Pour corriger l'erreur d'*offset* du comparateur, la technique de l'algorithme RSD (*Redundant Signed Digit*) est bien répandue (16) (17) (13) (12). L'*offset* du comparateur est partiellement corrigé à travers l'algorithme RSD, mais son influence

est éliminée de manière plus efficace durant la commutation des capacités, comme on verra durant l'implémentation du CAN.

2.2 CAN CYCLIQUE RSD

Afin de s'affranchir des erreurs d'*offset* dans le CAN Cyclique conventionnel, une nouvelle architecture a été proposée par (16). Cette architecture explore un algorithme nommé RSD (*Redundant Signed Digit*) qui rend plus facile la conception d'un convertisseur de haute résolution sans avoir des comparateurs de tension très précis. L'algorithme RSD est bien répandu et son architecture est parfois appelée 1.5-bit/étape. Outre la correction des erreurs d'*offset,* l'architecture RSD permet aussi la correction numérique à travers une calibration (17).

L'algorithme RSD utilise deux comparateurs au lieu d'un seul comme dans l'architecture d'un CAN Cyclique conventionnel. Ainsi, trois états de comparaison sont créés (b = -1, 0 ou +1), d'où vient le nom 1.5-bit/étape. Etant donné que le sous-CAN est changé pour extraire trois niveaux de quantification, le CNA est aussi changé afin qu'il produise trois niveaux analogiques : $+V_{ref}$, $-V_{ref}$ et 0. En revanche, la multiplication précise par 2 reste inchangée.

En résumé, l'entrée de chaque cycle est comparée à deux seuils de tensions situés entre $-V_{ref}/2$ et 0 et entre 0 et $V_{ref}/2$, couramment choisis égaux à $-V_{ref}/4$ et à $+V_{ref}/4$. L'équation générale de la fonction de transfert du cycle du CAN reste inchangée :

$$V_{res}(i+1) = 2 \times V_{res}(i) - b \times V_{ref}$$

mais cette fois-ci b=-1 dans le cas où $V_{res}(i)<-V_{ref}/4$, b=+1 si $V_{res}(i)>V_{ref}/4$ et finalement b=0 si $-V_{ref}/4<V_{res}(i)<V_{ref}/4$.

En rebouclant ce cycle on construit un CAN Cyclique RSD. Le logigramme de l'algorithme RSD est présenté sur la Figure 2-7.

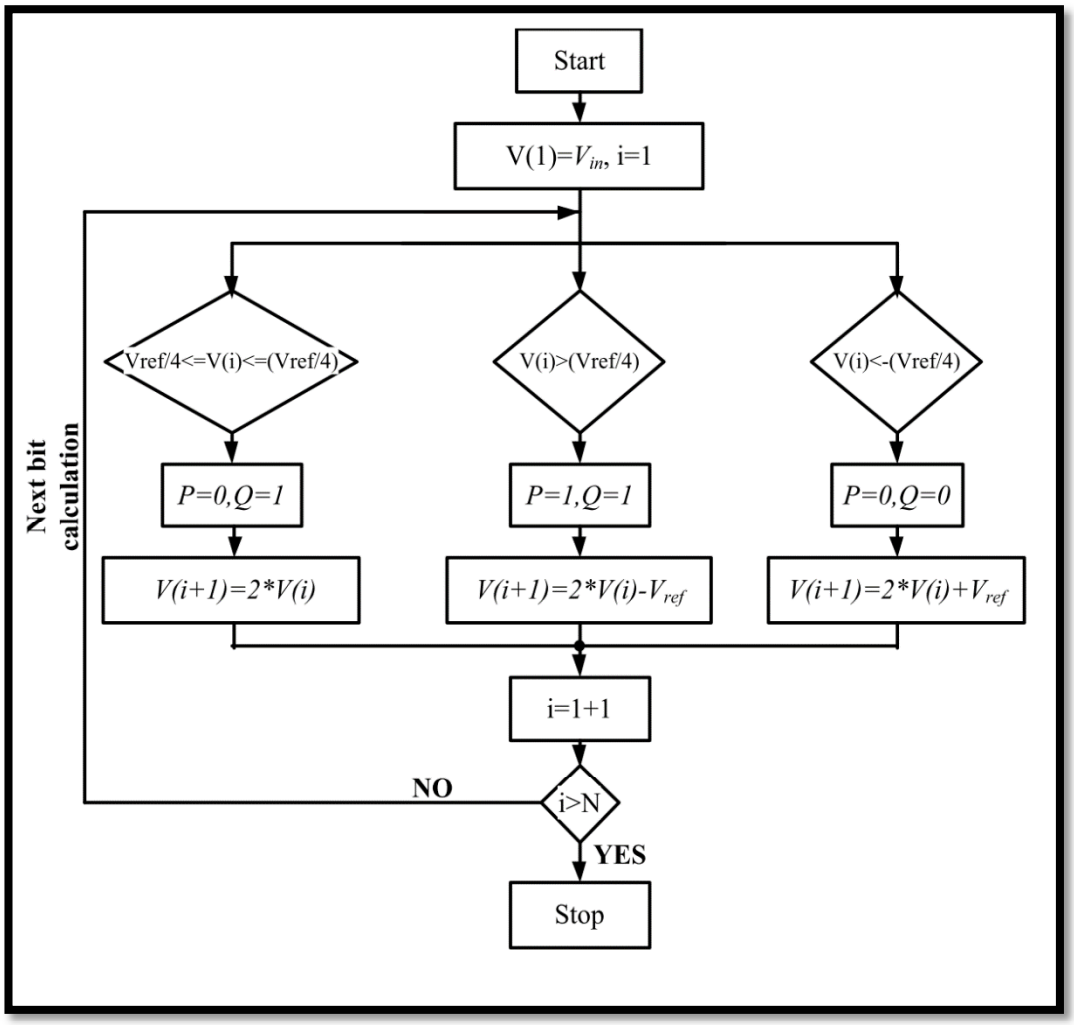


Figure 2-7 - Logigramme d'un CAN Cyclique RSD

Les bits produits durant l'opération décrite au-dessus ne constituent pas encore l'équivalent numérique de l'entrée analogique. En effet, pour que la conversion soit complète, une opération logique doit être faite avec les résultats de conversion de chaque cycle de conversion. Basé sur l'algorithme RSD, pour que le mot numérique soit créé analytiquement, à chaque fois que b=+1 est produit, on doit multiplier le mot partiel par 2 et sommer 1, c'est-à-dire déplacer le mot binaire une fois vers la gauche et sommer 1. Après, si b=-1 on doit multiplier également le mot par deux, mais tout de suite on doit soustraire 1. Finalement, si b=0 est produit on ne fait que multiplier le résultat partiel par 2. Dans le but d'implémenter cette opération, les états de comparaison b_i, représentés par deux bits chacun, sont manipulés à travers un circuit logique numérique traduisant l'opération décrite au-dessus. P et Q, par exemple sur le logigramme de la Figure 2-7, sont utilisés comme deux bits représentant b : PQ=00 si b=-1, 01 si b=0 et 10 si b=+1.

Dans l'intention de comparer cet algorithme avec celui d'un CAN Cyclique conventionnel 1-bit/étape, on présente sur la Figure 2-8 la fonction de transfert d'un cycle

tout seul. Les seuils de tension utilisés $+V_{ref}/4$ et $-V_{ref}/4$ déterminent les valeurs binaires de Q et P, respectivement.

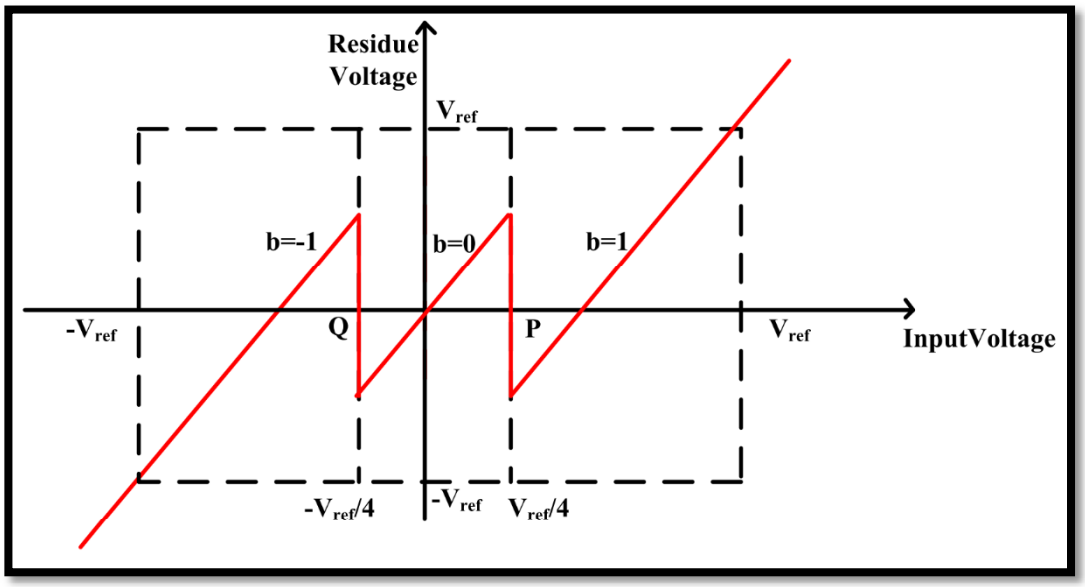


Figure 2-8 - Fonction de transfert d'un cycle d'un CAN RSD

Le grand avantage de l'algorithme RSD est vu sur les figures ci-dessous. La Figure 2-9 montre l'effet de l'*offset* du comparateur sur la fonction de transfert du cycle du CAN RSD. On remarque que, pour que la tension résiduelle sortante parvienne à sortir de la plage de tension d'entrée $\pm V_{ref}$, les *offset* de tension doivent dévier de plus de $\pm V_{ref}/4$, ce qui est normalement une marge suffisante pour les imperfections du comparateur. Ensuite, la Figure 2-10 présente l'effet de l'*offset* de l'amplificateur sur la même fonction de transfert. On rencontre toujours une zone de divergence à une des extrémités de la plage de tension d'entrée, mais celle située au milieu de la plage antérieurement est supprimée.

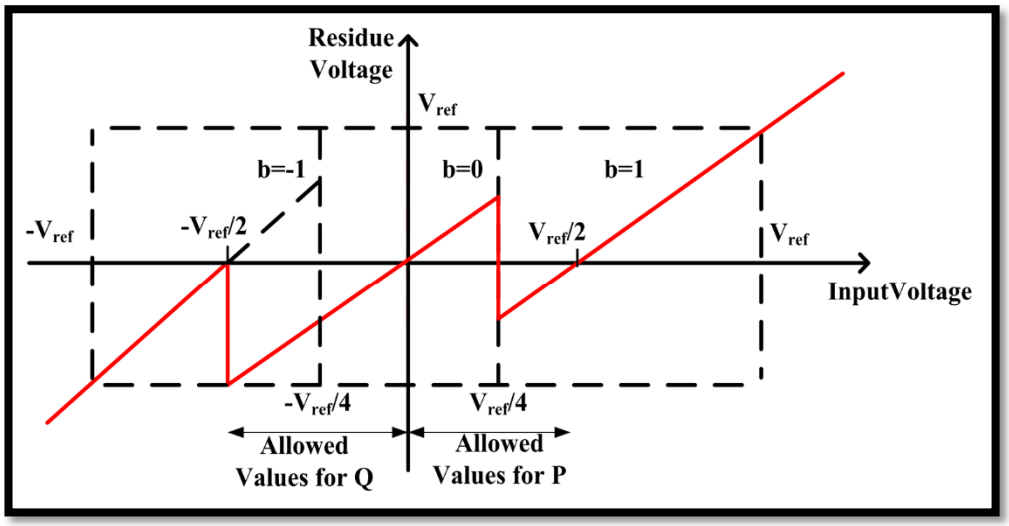


Figure 2-9 - Effet de l'offset du comparateur dans un CAN Cyclique RSD

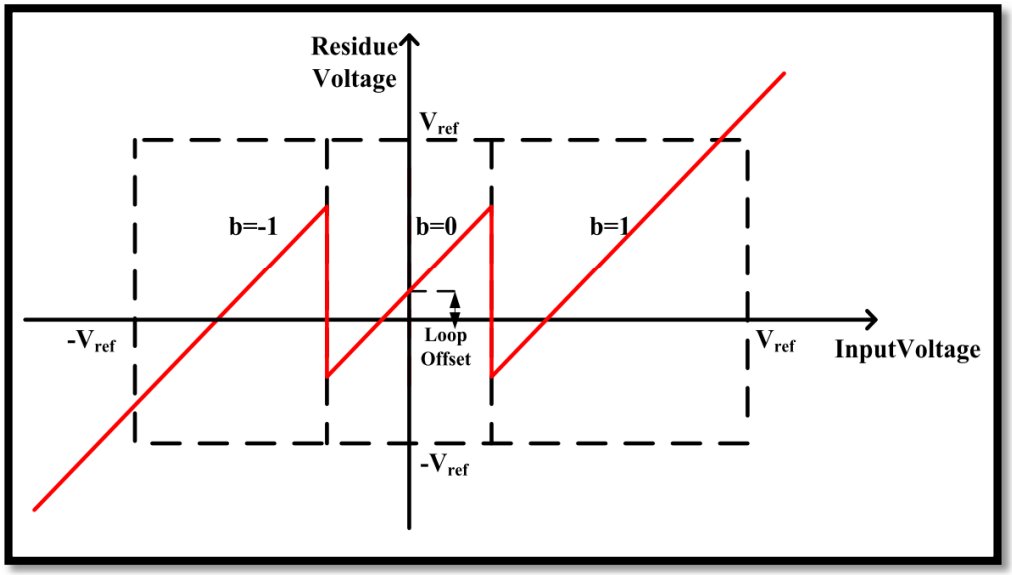


Figure 2-10 - Effet de l'offset de l'amplificateur dans un CAN Cyclique RSD

2.3 IMPLEMENTATION

Après avoir expliqué le fonctionnement d'un CAN Cyclique, conventionnel et RSD, on présente ensuite une manière de l'implémenter. Cette implémentation se base surtout sur les principes présentés en (18).

2.3.1 Vue d'ensemble

Comme on l'a dit antérieurement, un CAN comprend un E/H, un sous-CAN, un CNA, un soustracteur, un amplificateur de gain égal à 2 et un circuit logique. La Figure 2-11 présente un diagramme fonctionnel plus détaillé de comment le CAN Cyclique sera conçu.

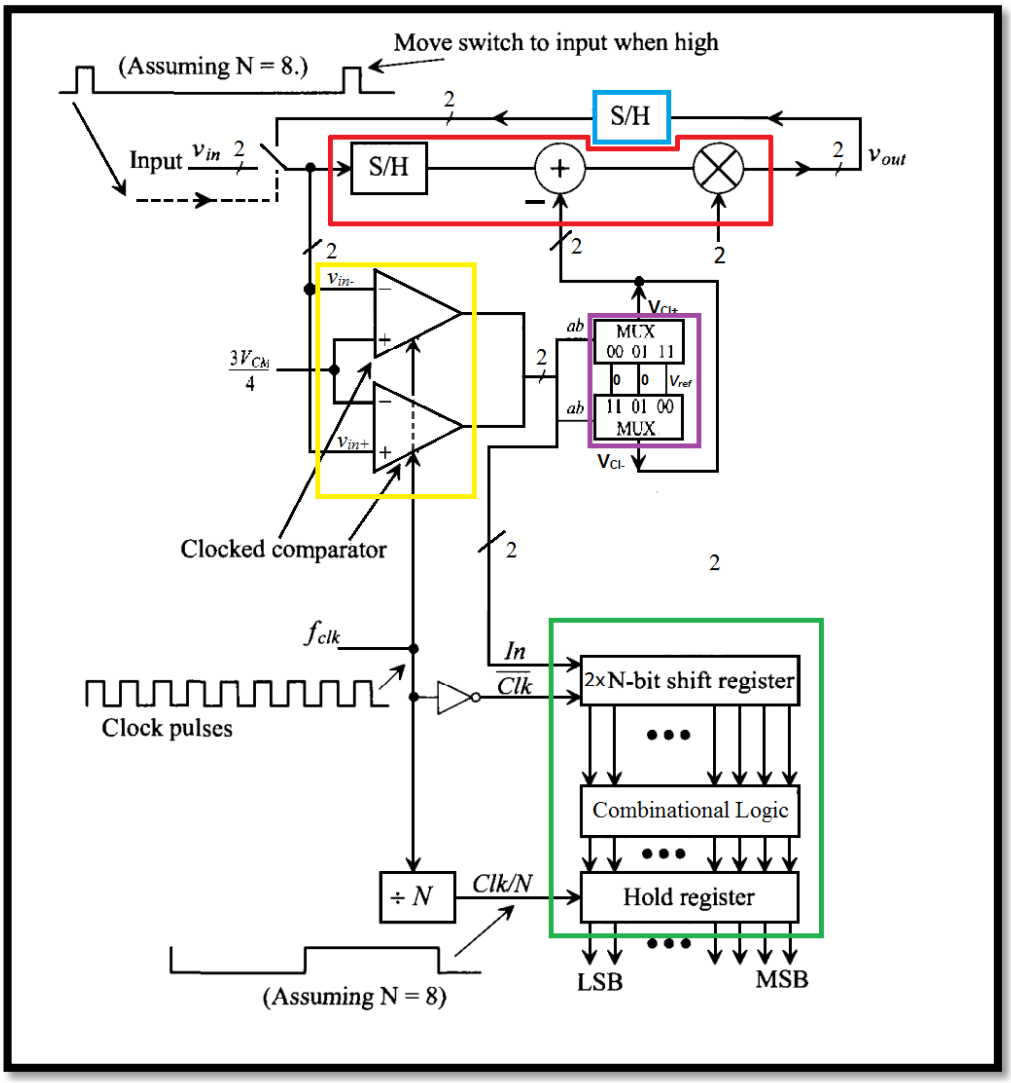


Figure 2-11 - CAN Cyclique RSD

En pratique, le sous-CAN peut être réalisé à partir de comparateurs (deux pour un CAN RSD, entourés en jaune), le CNA est créé à partir de commutateurs (MUXs, entourés en violet), le soustracteur et l'amplificateur peuvent être réalisés ensemble par un seul circuit qui fonctionne aussi comme E/H (entouré en rouge) et le circuit logique (entouré en vert) est réalisé avec des bascules et portes logiques combinatoires. En outre, on perçoit qu'un autre E/H (en anglais S/H ou *Sample and Hold*) est nécessaire lorsque le résidu est rebouclé (élément entouré en bleu). Ces circuits seront étudiés sur les prochains paragraphes.

2.3.2 Echantillonneur/Bloqueur (E/H)

Un circuit E/H a pour but de détecter une tension analogique à l'entrée à un instant déterminé et la bloquer pendant un temps suffisant pour que le reste du CAN la traite afin de la quantifier. Normalement l'E/H est réalisé à partir d'un circuit à capacités commutées. La Figure 2-12 présente un E/H *fully-differential* utilisant trois signaux de commutation.

L'amplificateur différentiel sur la figure, en réalité, peut être un OTA (*Operational Transconductance Amplifier* ou Amplificateur Opérationnel à Transconductance) dès lors qu'une faible impédance de sortie n'est pas exigée, ce qui est le cas puisque l'amplificateur ne va actionner que de capacités.

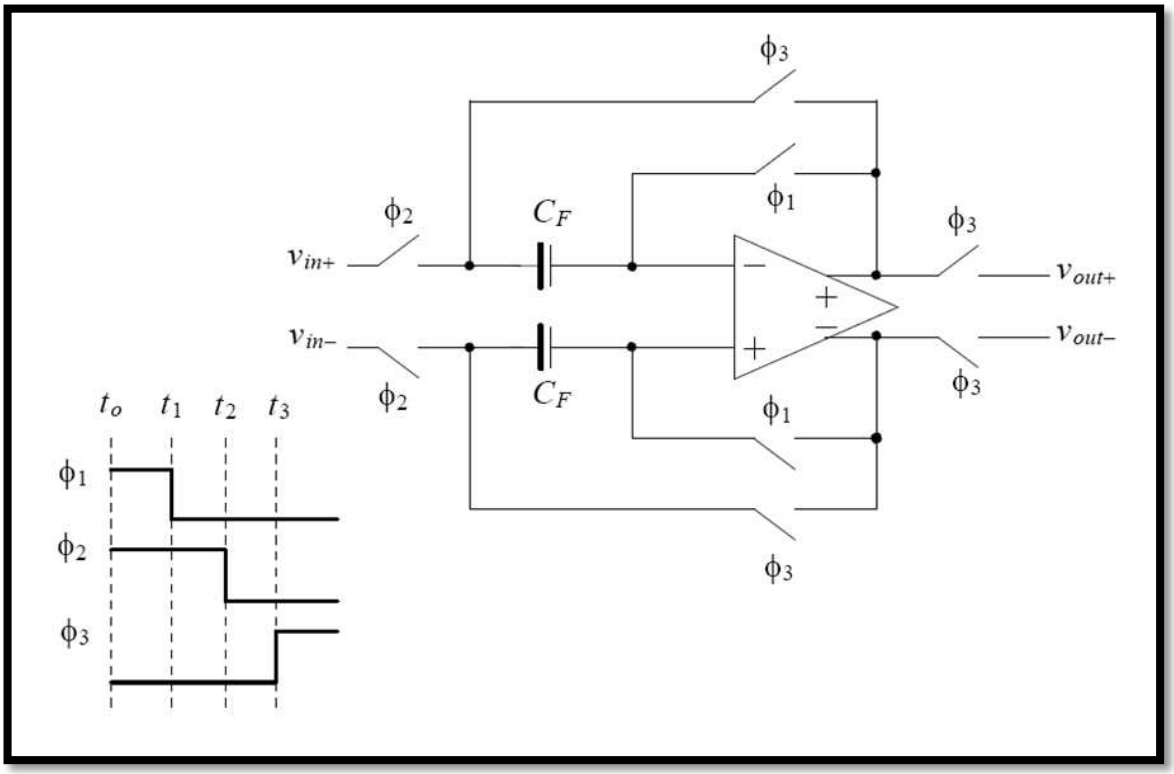


Figure 2-12 – Echantillonneur/Bloqueur *fully-differential*

Entre t_0 et t_1 la tension d'entrée est échantillonnée. L'OTA fonctionne comme un suiveur et présente sur la sortie sa tension d'*offset*. Au même instant les capacités sont chargées avec la différence entre la tension d'entrée et la tension de mode commun (V_{CM}) plus l'*offset* (V_{OS}).

$$Q_F^{\varphi_1} = C_F \cdot (V_{in+,in-} - V_{CM} \pm V_{OS})$$

A partir de t_1 jusqu'à t_3 l'OTA fonctionne en boucle ouverte, ce qui peut le forcer à un régime de saturation. Ainsi on désire que cette période soit la plus courte possible. En effet, après que le commutateur φ_1 s'ouvre, φ_2 s'ouvre tout de suite. Cette séquence est couramment appelée *Bottom Plate Sampling*. Son objectif est d'éviter des erreurs dues aux injections de charge créées par les commutateurs. Lorsque le commutateur φ_2 s'ouvre, la charge injectée fuit vers le nœud de plus petite impédance, v_{in} puisque φ_1 a été déjà ouvert.

Lorsque t_3 passe au niveau haut, l'OTA fonctionne en boucle fermée à nouveau et la capacité est reliée à la sortie de l'amplificateur. A partir du principe de la conservation de

charge, on conclut que la tension à travers la capacité se maintient et donc l'OTA répète et tient la tension échantillonnée.

$$Q_F^{\varphi_3} = C_F \cdot (V_{out+,out-} - V_{CM} \pm V_{OS})$$

$$Q_F^{\varphi_1} = Q_F^{\varphi_3}$$

$$V_{out+,out-} = V_{in+,in-}$$

2.3.3 Soustracteur et Multiplicateur

Le circuit chargé de soustraire et multiplier le signal dans le cycle du CAN suit le même principe qu'un E/H. En effet les signaux φ_1, φ_2 et φ_3 sont aussi utilisés de façon identique. Cependant, maintenant on utilise quatre capacités pour implémenter la multiplication et deux entrées de tension supplémentaires pour implémenter la soustraction. La Figure 2-13 montre un tel circuit.

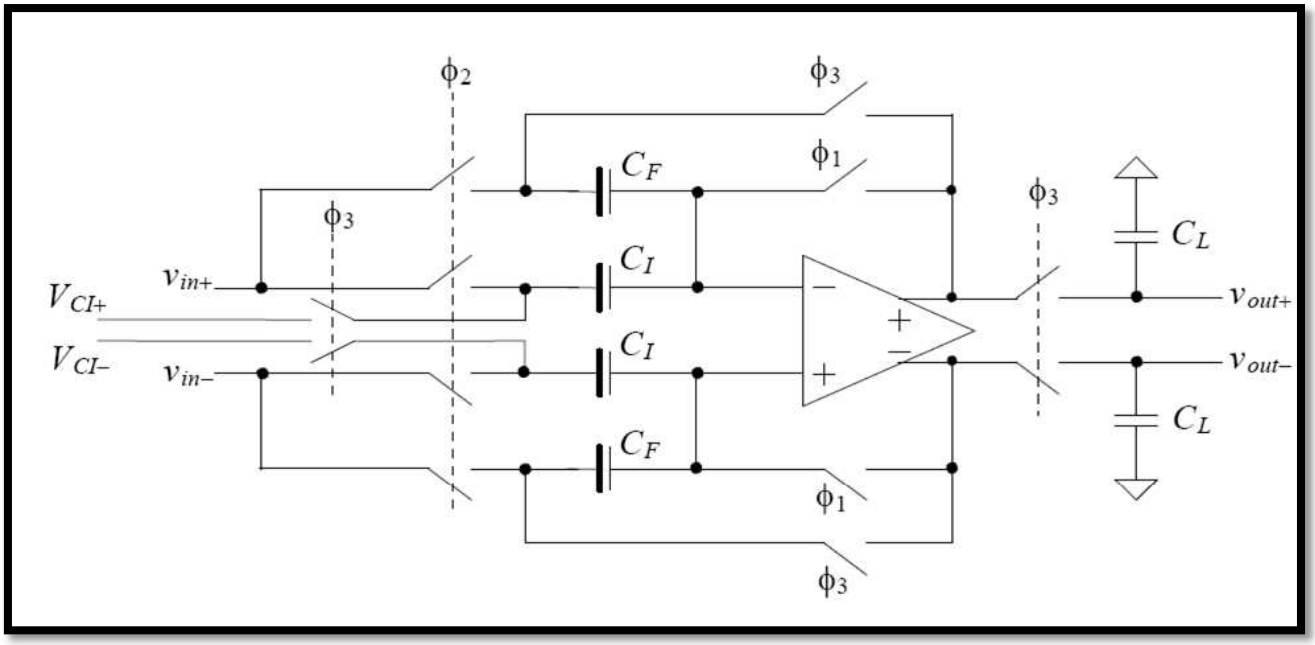


Figure 2-13 – Soustracteur et Multiplicateur *Fully-Differential*

Avant t_1 toutes les capacités sont chargées comme suit :

$$Q_{I,F}^{\varphi_1} = C_{I,F} \cdot \left(V_{in+,in-} - V_{CM} \pm V_{OS}\right)$$

Après que les commutateurs φ_3 sont fermés, les charges deviennent :

$$Q_I^{\varphi_3} = C_I \cdot (V_{CI+,CI-} - V_{CM} \pm V_{OS})$$

$$Q_F^{\varphi_3} = C_F \cdot \left(V_{out+,out-} - V_{CM} \pm V_{OS}\right)$$

A nouveau, en partant du principe de la conservation de charge :

$$Q_I^{\varphi_1} + Q_F^{\varphi_1} = Q_I^{\varphi_3} + Q_F^{\varphi_3}$$

$$C_F \cdot \left(V_{out+,out-} - V_{CM} \pm V_{OS}\right)$$
$$= C_I \cdot \left(V_{in+,in-} - V_{CM} \pm V_{OS}\right) + C_F \cdot \left(V_{in+,in-} - V_{CM} \pm V_{OS}\right) - C_I \cdot \left(V_{CI+,CI-} - V_{CM} \pm V_{OS}\right)$$

$$V_{out+,out-} = \left(1 + \frac{C_I}{C_F}\right) \cdot V_{in+,in-} - \frac{C_I}{C_F} \cdot V_{CI+,CI-}$$

Comme $V_{out}=V_{out+} - V_{out-}$ et $V_{in} = V_{in+} - V_{in-}$:

$$V_{out} = \left(1 + \frac{C_I}{C_F}\right) \cdot V_{in} - \frac{C_I}{C_F} \cdot (V_{CI+} - V_{CI-})$$

Par conséquent, pour que l'on puisse multiplier le signal d'entrée par deux, il suffit de faire C_I égal à C_F. D'ailleurs, pour réaliser la soustraction il suffit de choisir une différence de tension exact, V_{CI+}-V_{CI-}, qui traduisent la valeur que l'on veut soustraire.

2.3.4 Sous-CAN

Comme on l'avait déjà évoqué, dans un CAN Cyclique RSD, le sous-CAN est conçu à partir de deux comparateurs dont on extrait trois états de comparaison. Dans cette implémentation, après les comparateurs les deux sorties suivent le *thermometer code*, c'est-à-dire, b=-1, 0, +1 sont représentés respectivement par P_pQ_p=00, 01, 11. Pour que le codage soit converti au binaire, on ajoute une porte logique pour que $Q=P_p'Q_p$ et $P=P_p$.

En outre, les seuils de tension analogique envisagés pour v_{in} sont $-V_{ref}/4$ et $+ V_{ref}/4$. Ainsi, pour que l'on utilise les deux signaux *single-ended* comme entrées des comparateurs, la valeur de tension aux entrées des comparateurs est égale à $3V_{ref}/8$:

$$v_{in+} = \frac{V_{ref}}{2} + \frac{v_{in}}{2}$$

$$v_{in-} = \frac{V_{ref}}{2} - \frac{v_{in}}{2}$$

$$v_{in} = 2 \cdot v_{in+} - V_{ref} = V_{ref} - 2 \cdot v_{in-}$$

$$v_{in} > -\frac{v_{ref}}{4} \rightarrow v_{in+} > \frac{3 \cdot v_{ref}}{8} \quad et \;\; Q = 1$$

$$v_{in} > \frac{v_{ref}}{4} \rightarrow v_{in-} < \frac{3 \cdot v_{ref}}{8} \quad et \; P = 1$$

A partir de ces équations, il est facile d'imaginer une implémentation à travers deux comparateurs. La Figure 2-14 présente un schéma avec des comparateurs, une porte logique ET et des bascules pour que le sous-CAN devienne synchrone.

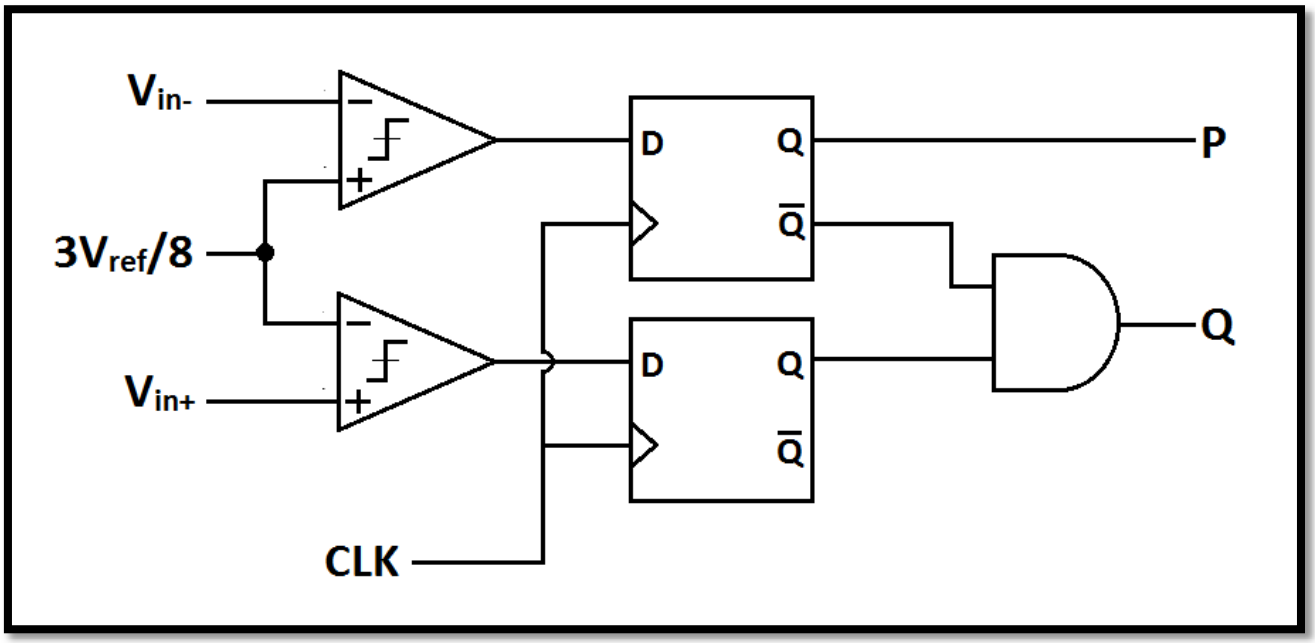


Figure 2-14 – Sous-CAN d'un CAN Cyclique RSD

2.3.5 CNA – Multiplexeur

D'après l'équation du circuit chargé de multiplier et soustraire le signal, le CNA du CAN Cyclique RSD doit fournir son résultat de conversion à partir des tensions V_{CI+} et V_{CI-}. Présenté sur la Figure 2-11, le CNA est facilement implémenté grâce à deux multiplexeurs. Les sorties de ces multiplexeurs sont les tensions V_{CI+} ou V_{CI-} et elles doivent être choisies de façon que la différence entre eux soit $+V_{ref}$, 0 ou $-V_{ref}$.

La Figure 2-15 donne le diagramme fonctionnel du CNA et montre une façon de réaliser les multiplexeurs.

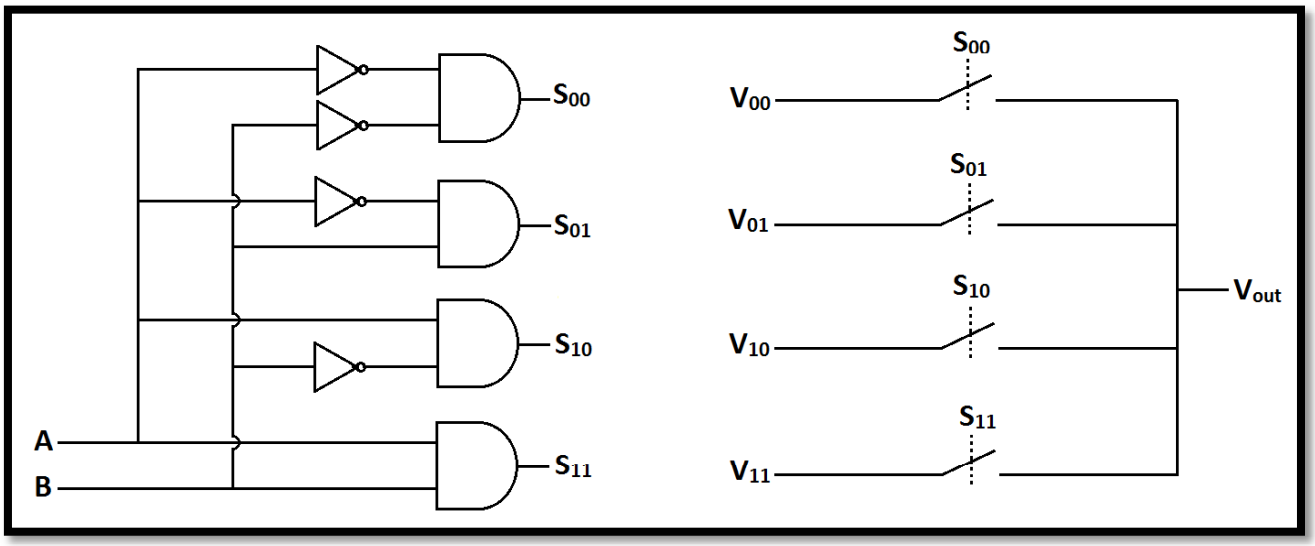


Figure 2-15 – CNA d'un CAN Cyclique RSD

2.3.6 Logique

On a vu dans le paragraphe 2.2 qu'après chaque cycle d'un CAN Cyclique RSD on peut obtenir trois états de comparaison b=-1, 0 ou 1 et qu'à partir des états de tous les cycles on peut extraire le mot numérique résultant de la conversion A/N. Dans ce paragraphe, on présente un circuit logique pour réaliser cette opération à partir des bits PQ créées par le sous-CAN.

La Figure 2-16 présente une chaîne d'additionneurs complets. Chacun possède comme entrées Q_i et P_{i-1} et la retenue de l'additionneur antérieur. On remarque que le résultat de cette addition comprend N+1 bits. En effet, les bits intéressants pour la conversion A/N sont les N bits le plus significatifs. Or, le bit le moins significatif est utilisé pour ajuster la fonction de transfert du CAN au regard de la largeur du pas du premier niveau discret. C'est pourquoi sur le premier additionneur on utilise comme entrée un bit de niveau logique 1.

De plus, pour que les P_i et Q_i soient manipulés ensemble, des registres à décalage sont instanciés à l'entrée du circuit logique à partir de bascules D. Finalement, une autre chaîne de bascules D (en parallèle) a été instanciée à la sortie du circuit pour que le résultat de conversion soit mis à jour à un instant précis déterminé par un signal exclusif *enable*.

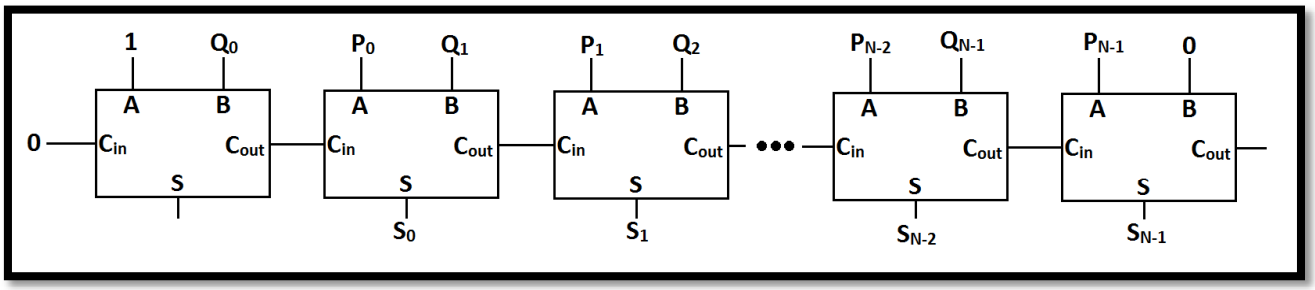


Figure 2-16 – Circuit logique traduisant $P_{N-1}Q_{N-1}...P_0Q_0$ en bits $d_{N-1}...d_0$

2.4 NON-IDEALITES

Dans les paragraphes 2.1 et 2.2 on a parlé de l'erreur d'*offset* du comparateur et des erreurs de gain et *offset* de l'amplificateur. Ainsi, on a vu brièvement les effets sur la fonction de transfert d'un seul cycle d'un CAN Cyclique conventionnel et RSD. Maintenant, sachant comment réaliser les fonctions discutées, on parlera brièvement de quelques sources d'erreurs dans l'implémentation du paragraphe 2.3.

Tout d'abord, il est rappelé que les circuits à capacités commutées sont clairement dépendants de leurs condensateurs et de leurs commutateurs. Néanmoins, dans la pratique, ces composants présentent des défauts ou incertitudes.

Premièrement, dans la technologie CMOS, les commutateurs sont normalement réalisés à partir de transistors NMOS, PMOS ou les deux en parallèle. Pour que ce commutateur soit fermé, il faut que le condensateur MOS, entre la grille et le substrat, soit chargé et le canal créé. En revanche, quand ce même commutateur est ouvert, cette charge est restituée, ce qui peut affecter la charge des condensateurs utilisés.

Une autre source d'erreurs du circuit est la nature des capacités. Les condensateurs dans la technologie microélectronique sont habituellement créés à partir de deux plaques métalliques dont une est plus proche du substrat semi-conducteur que l'autre. Ainsi, le premier effet non-désiré rencontré est la capacité parasitique de la plaque inférieure, surtout si l'on parle des capacités formées par les couches de *polysilicium*. Le deuxième est la non-linéarité dépendant surtout du matériau présent entre les deux plaques, ce qui fait varier la valeur de capacité en fonction de la tension appliquée à ses extrémités et de la température. Finalement, la dernière source d'erreur liée aux condensateurs est l'imprécision du procédé de fabrication. Par exemple, le circuit décrit en 2.3.3 exige que les capacités C_F et C_I soient identiques pour que la multiplication soit exactement égale à 2. Or, pendant la fabrication une différence entre elles peut exister et la multiplication est donc dégradée.

En ce qui concerne les OTAs présents dans l'E/H et dans le circuit Multiplicateur/Soustracteur, quelques non-idéalités sont à souligner. Outre l'*offset* de l'OTA, provenant de son architecture interne, le gain non infini est aussi une source d'erreur. En effet, les équations développées dans le paragraphe 2.3 sont parties de l'hypothèse que le gain de boucle ouvert de l'OTA est infini, mais en pratique le gain trouve une limite et ainsi la multiplication par deux acquiert une imprécision. Egalement, durant le développement des équations, le gain de l'OTA a été considéré constant, ce qui est notamment faux puisque la sortie d'un OTA devient saturée lorsque la tension s'approche des tensions d'alimentation. Ainsi, la conversion aux extrémités de la plage de tension analogique sera aussi dégradée.

Finalement, les comparateurs possèdent aussi des imprécisions, surtout liées à son *offset*. Celui-ci peut exister à cause de l'architecture interne du comparateur ou à cause d'incertitudes sur la tension de comparaison. Cependant, il était déjà vu que l'algorithme RSD relaxe les contraintes sur l'*offset* du comparateur. Ainsi, ces non-idéalités ne sont pas un obstacle à la performance du CAN Cyclique RSD.

Il existe dans la littérature des propositions pour corriger ou réduire les effets d'un gain fini, des injections de charge, des non-linéarités des capacités, etc. L'objectif de ce travail n'est pas de les comparer et/ou trier ni de proposer une nouvelle technique, mais de créer un modèle qui les prennent en compte.

3. MACRO-MODELISATION

Ce chapitre vise à montrer comment les circuits d'un CAN Cyclique RSD, présentés au fur et à mesure du chapitre 2, seront construits et ensuite simulés sur le logiciel LTSPICE. Dès que possible on utilisera des composants LTSPICE dans le macro-modèles car il est attendu que ces modèles soient déjà optimisés en terme d'efficacité calculatoire.

Tout d'abord un OTA différentiel est présenté dans le paragraphe 0. Ensuite les condensateurs sont présentés avec leurs non-idéalités intrinsèques en 0. Les commutateurs, à leur tour, sont modélisés au paragraphe 3.3. Ensuite, les comparateurs synchrones sont décrits en 3.4. Finalement le circuit logique sera présenté en dernier lieu au paragraphe 3.5.

3.1 OTA

L'amplificateur opérationnel à transconductance utilisé dans ce travail doit prendre en compte, en plus des paramètres d'un OTA *single-ended* idéal, la conversion vers une sortie *fully-differential* et sa limitation de tension.

LTSPICE possède une fonction nommée OTA, tel que dans les codes SPICE, qui fonctionne comme un amplificateur de tension. Cette fonction permet le changement de quelques-uns de ses paramètres. On en liste les plus importants :

- Gain de transconductance (paramètre *G*). Défaut G=1.
- Résistance de sortie (paramètre *Rout*). Défaut Rout=1/Gmin= la résistance la plus grande possible d'après le simulateur.
- Capacitance de sortie (paramètre *Cout*). Défaut Cout=0.
- Tension d'*offset* (paramètre *Ref*). Défaut Ref=0.
- Courant de saturation de sortie (paramètre *Iout*). Défaut Iout=1Ou.
  - Si le paramètre *linear* est défini égal à 1, la saturation de courant est déactivée.
  - Si *linear* est égal à 0 et si le flag *Asym* est égal à 1, les courants de saturation injecté ou prélevé de la sortie peuvent être différents. Le paramètre *Isrc* (ou *Isource*) et le paramètre *Isink* sont les valeurs de courant de saturation poussé et tiré respectivement. Défauts Isrc=Iout et Isink=-Iout.
- Courant d'*offset* de sortie (paramètre *Ioffset*). Défaut Ioffset=0.

Un symbole a été créé pour que cette fonction ait pu être utilisée sur l'éditeur de schémas, puisque LTSPICE ne le rend pas disponible.

Ensuite, pour que la sortie de l'amplificateur soit convertie de *single-ended* vers *fully-differential*, un *balun* (*balanced-unbalanced*) a été utilisé. Ce composant peut être conçu à partir de deux transformateurs électriques, comme vu sur la Figure 3-1. Ces transformateurs, à leur tour, peuvent être macro-modélisés de différentes façons dont deux sont présentés sur la Figure 3-2. Les deux ont montré des résultats satisfaisants et n'ont pas présenté de problèmes de convergence. Tout de même, la deuxième a été choisie.

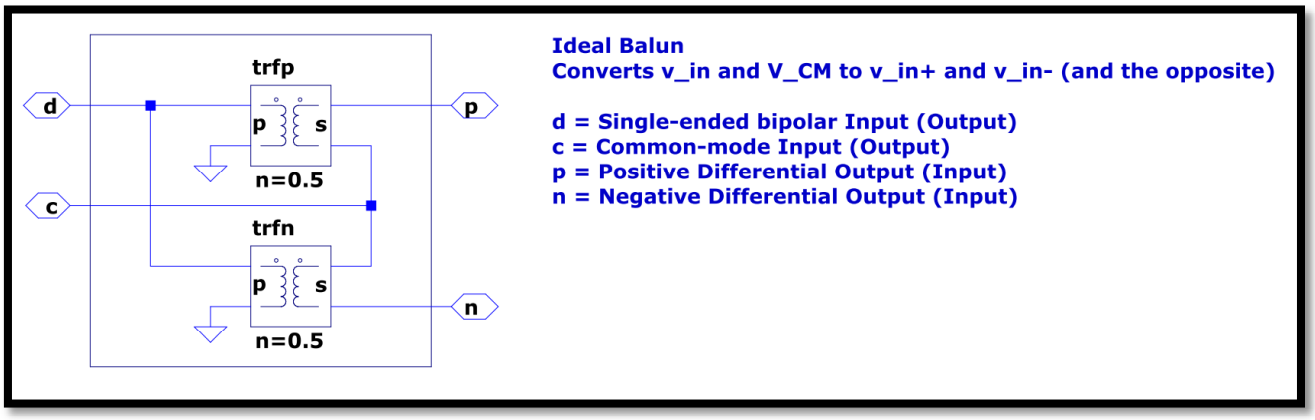


Figure 3-1 – Balun compris par deux transformateurs

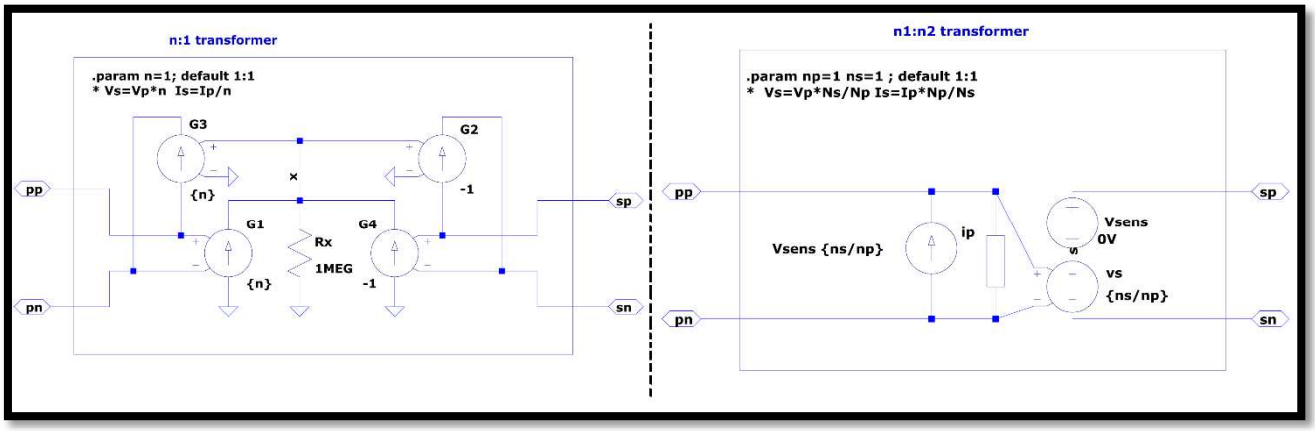


Figure 3-2 – Macro-modèles de transformateur

Ensuite, à la sortie du *balun* une autre étape doit être faite pour que la tension soit limitée à un certain niveau, comportement rencontré sur les vrais circuits, lorsque les sorties s'approchent des tensions d'alimentation. Dans ce travail, on présente trois façons différentes de l'implémenter : par diodes, par *varistor* et par des sources comportementales de courant.

La limitation de tension à diode est conçue à partir de deux diodes sur chaque sortie et deux sources de tension. Les diodes sont placées de façon à ce que les extrémités positive-négative soient sortie-Vdd et Vss-sortie. Ainsi, lorsque la sortie est plus grande

que Vdd plus la tension de seuil ou plus petite que la tension de Vss plus la tension de seuil, la diode commence à conduire et la tension est limitée. Evidemment, puisque la macro-modélisation envisage l'optimisation de la simulation à l'égard de la vitesse, le modèle de diode utilisé n'est pas le modèle semi-conducteur, mais un modèle à partir duquel la diode se comporte comme une faible résistance quand elle est passante et une forte résistance quand elle est bloquée, les deux étant définies comme des paramètres. La transition entre une valeur et l'autre est faite de façon quadratique pour que le simulateur n'ait pas de problèmes de convergence. Les paramètres du modèle de la diode utilisés sont *Ron* (résistance passante), *Roff* (résistance bloquée), *Vfwd* (tension de seuil) et *Epsilon* (l'épaisseur de la région quadratique de transition). D'ailleurs, la diode pourrait être aussi en état *reverse*, mais cela n'est pas intéressant pour la limitation de tension. Finalement, on remarque qu'Epsilon est choisi relativement petit et que Vfwd est choisi négatif et petit (pour que la limite de tension reste proche et au-dessous Vdd). Le circuit est montré sur la Figure 3-3.

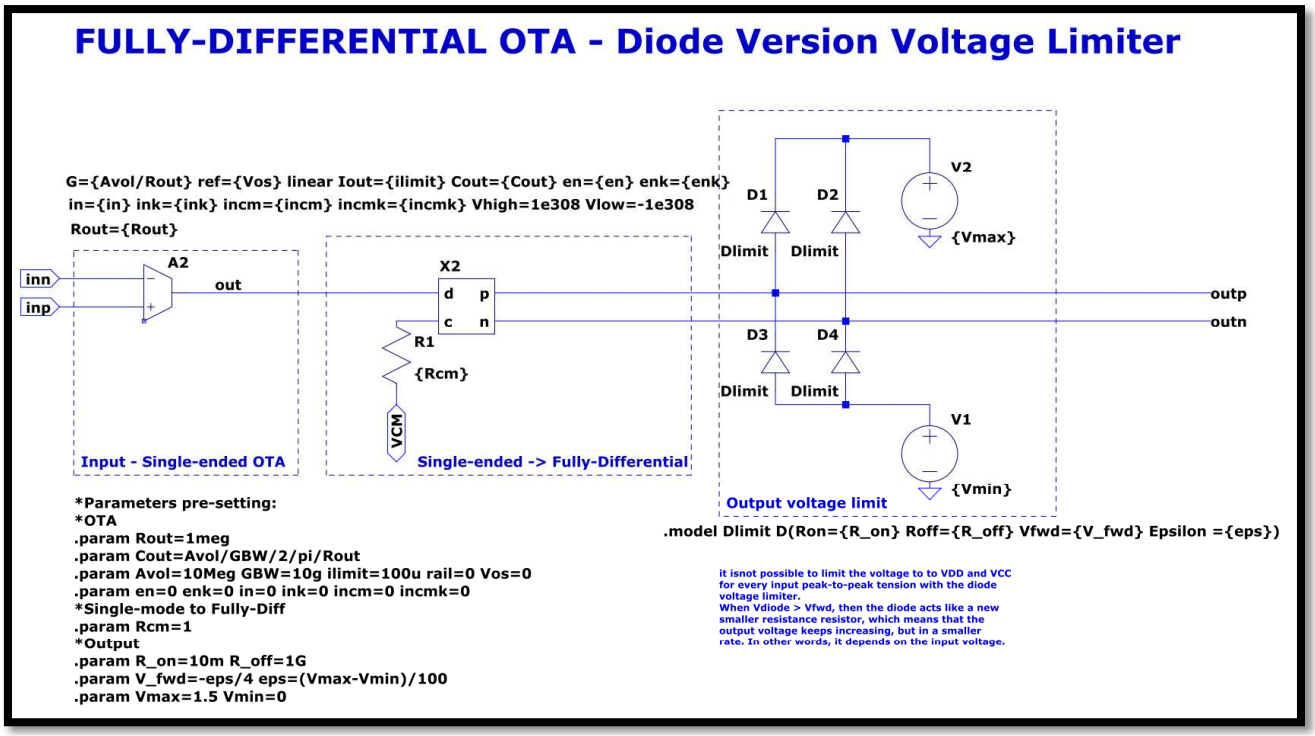


Figure 3-3 – OTA *fully*-differential à limitation de tension à diodes

La limitation à varistor est conçue à travers un varistor sur chaque sortie et une source de tension. Le composant utilisé, appelé varistor, possède deux paires de points d'accès de tension que l'on appellera $V_{control}$ et V_{cond}. Le varistor fonctionne comme deux résistances, une faible et une autre élevée, qui sont commutées selon une règle. Si $V_{cond} > |V_{control}|$ alors la résistance sur les accès de V_{cond} sera faible, sinon elle sera élevée. Les résistances sont définies par les paramètres *Roff* (haute résistance) et *Rclamp* (faible

résistance). L'OTA implémenté avec limitation de tension à varistor est présenté sur la Figure 3-4.

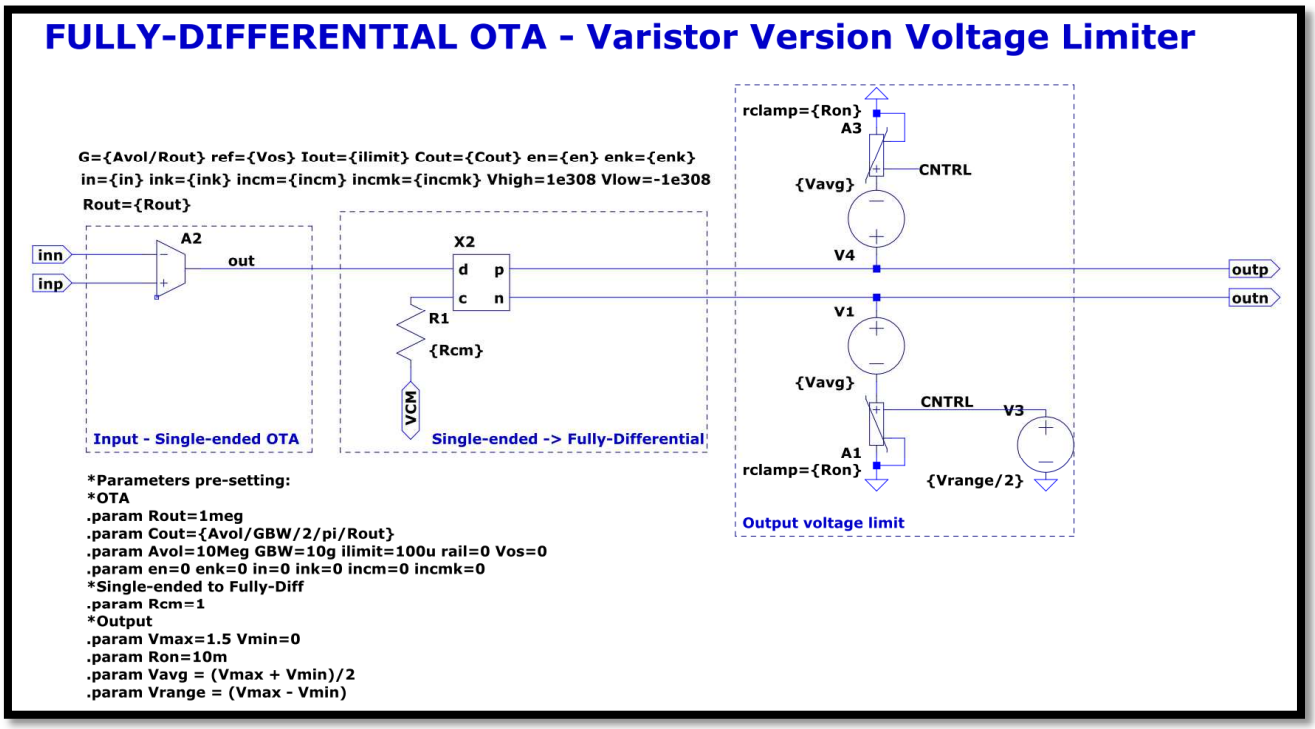


Figure 3-4 - OTA *fully*-differential à limitation de tension à varistor

Enfin, la limitation de tension à sources comportementales de courant est réalisée à partir d'une source comportementale de courant par sortie. Le principe de fonctionnement ressemble à celui des autres limiteurs de tension : réduire l'impédance – ou tirer le courant davantage – dans les conditions de tension au-dessus de la tension maximale ou au-dessous de la tension minimale. L'intérêt pour les sources comportementales vient du fait qu'avec elles on peut implémenter n'importe quelle fonction de transition entre les faibles et les hautes impédances, ce qui permet de concilier analyse de convergence avec vitesse de calcul. Particulièrement, afin de rendre possible l'analyse des ordres de polynômes, la fonction ci-dessous a été utilisée en faisant variant n.

$$I = \begin{cases} -(V_{min} - V_{cond})^n \ , \ si \ V_{cond} < V_{min} \\ 0 \ , \ si \ V_{max} < V_{cond} < V_{min} \\ (V_{cond} - V_{max})^n \ , \ si \ V_{cond} > V_{max} \end{cases}$$

Le circuit de l'OTA avec une limitation de tension à sources comportementales de courant est présenté sur la Figure 3-5.

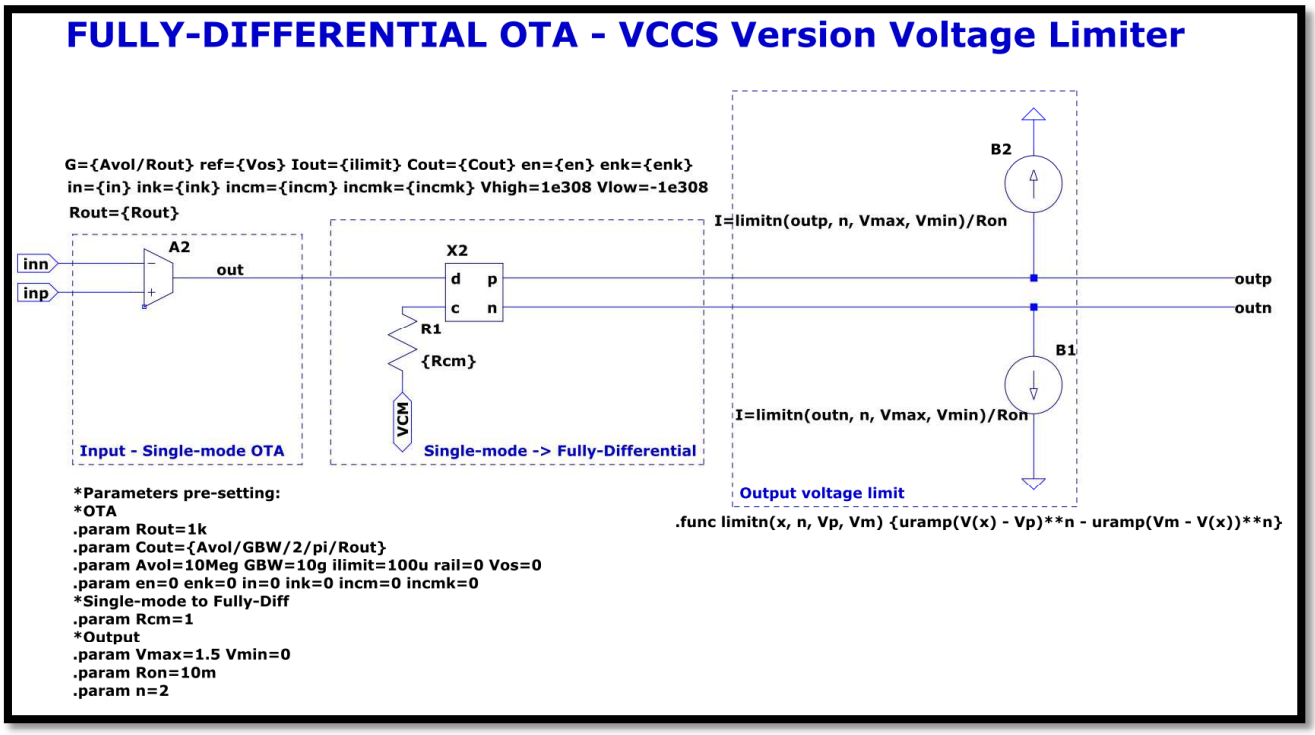


Figure 3-5 - OTA *fully*-differential à limitation de tension à sources comportementales de courant

Enfin, après avoir testé les trois limiteurs de tension, on a conclu que le dernier prenait plus de temps à être simulé, alors que les deux premiers se montraient relativement identiques. Pour l'avancement des simulations, le limiteur de tension à diodes a été retenu.

3.2 CONDENSATEURS

Comme déjà introduit dans la section 2.4, les condensateurs utilisés lors de ce travail doivent prendre en compte deux caractéristiques indépendantes : le condensateur parasitique de la plaque inférieure (*bottom plate*) et la capacitance non-linéaire.

La capacité parasitique de la plaque inférieure est caractérisée simplement par une capacité, pour l'instant linéaire, reliant une des extrémités du condensateur nominal au substrat. Puisque le condensateur n'est plus symétrique, il faut remarquer déjà sur le symbole quel est le côté de la plaque inférieure. On rencontre dans la littérature plusieurs illustrations différentes. Dans ce travail, la plaque inférieure sur le symbole est représentée par une trait plus épais.

Le deuxième effet pris en compte est la capacité non-linéaire. Le logiciel LTSPICE permet déjà qu'une capacité soit définie à partir de sa charge, c'est-à-dire, au lieu de définir une valeur de capacitance, une expression de la charge en fonction de la tension Q=f(v) est formulée. Pour que le modèle ressemble à ce que l'on rencontrerait en

pratique, un modèle de condensateur de la technologie TSMC 0.18um a été utilisé. Comme cette modèle exprime la capacité comme une fonction directe C=g(v), pour la traduire vers une déclaration que LTSPICE comprend, il faut remarquer que au lieu de la définition C=Q/V, on utilise C=dq/dv tant pour LTSPICE que pour le modèle TSMC. Ainsi :

$$Q = \int_0^V C(v)\, dv$$

Finalement, on a créé aussi un condensateur comprenant à la fois les deux effets. La Figure 3-6 montre les symboles et les schémas simplifiés des trois condensateurs discutés.

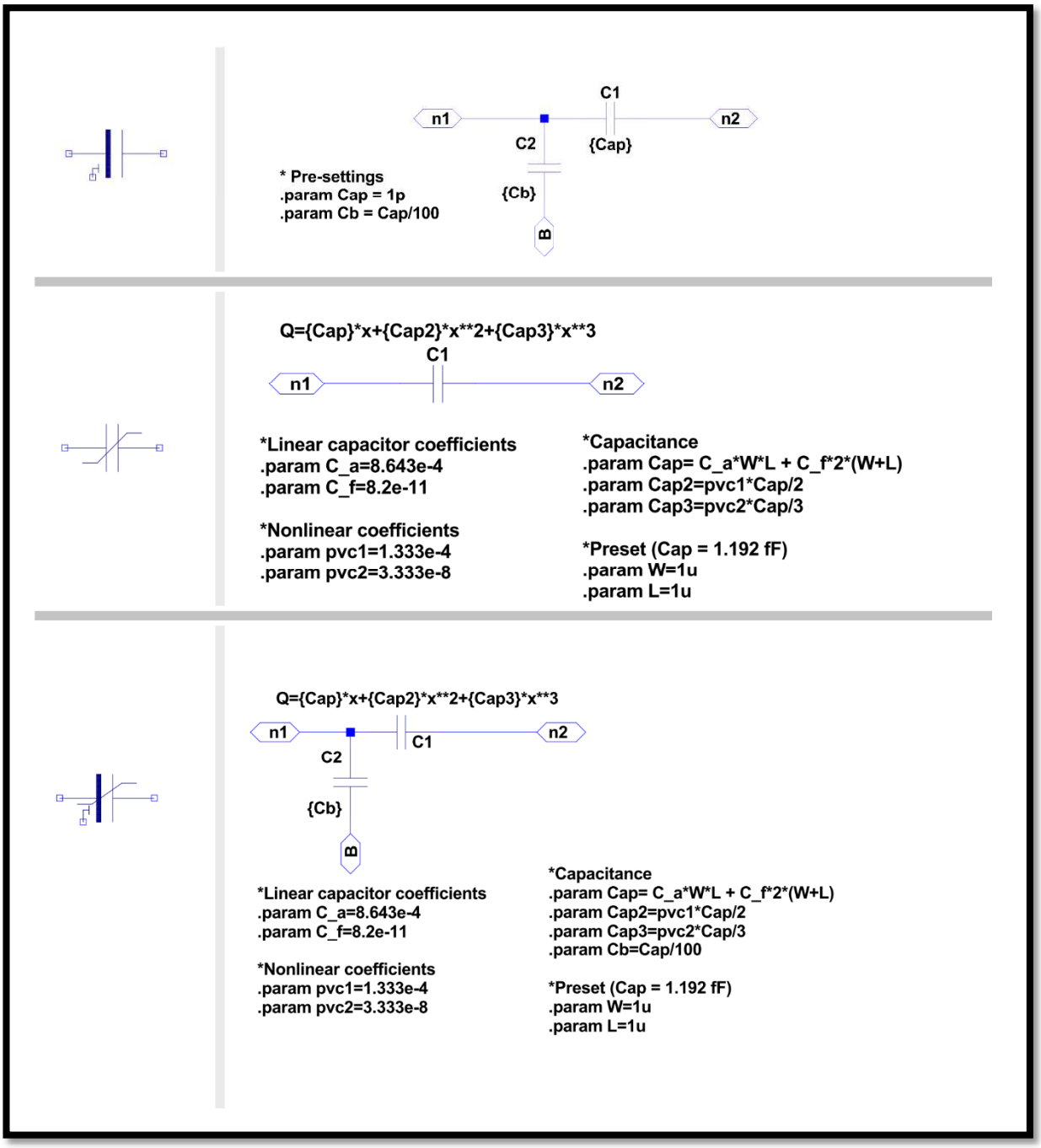


Figure 3-6 – Condensateurs comprenant la capacité de la plaque inférieure, la capacitance non-linéaire et les deux simultanément

3.3 COMMUTATEURS

Les modèles de commutateur utilisés pour implémenter le CAN Cyclique RSD sont basés sur des commutateurs électroniques composés soit d'un transistor NMOS, soit d'un PMOS, soit des deux en parallèle (appelé CMOS). Dans les trois cas l'effet d'injection de charge est gênant pour les circuits à capacités commutées comme les circuits de convertisseurs A/N. Ce phénomène provient de la décharge de la capacité MOS, présente dans tous les transistors MOSFET, lorsque le commutateur est ouvert.

Dans ce travail, l'injection de charge est modélisée par un condensateur entre la grille et le drain et un autre entre la grille et la source. De plus, les transistors, PMOS et NMOS, sont remplacés par des composants LTSPICE appelés SW.

SW caractérise un commutateur contrôlé par une tension. Le composant SW possède quatre portes dont deux fournissent la tension de contrôle V_{cont} et les deux autres fournissent la voie que l'on veut commuter entre circuit ouvert ou court-circuit. Les paramètres de SW utilisés sont *Ron* (résistance série lorsque SW est passant), *Roff* (résistance série lorsque SW est bloqué), *Vt* (seuil de tension sans hystérésis) et *Vh* (tension d'hystérésis).

Pour utiliser SW, il faut choisir entre deux niveaux de comportements, un niveau 1 ou un niveau 2. Le niveau 1 décrit un commutateur avec hystérésis où les seuils de tension sont Vt-Vh et Vt+Vh. Par ailleurs, si Vh<0 alors la région Vt-Vh< V_{cont}<Vt+Vh devient une région de transition quadratique entre le régime passant et le régime bloqué. Le niveau 2, à son tour, est une variation du niveau 1 lorsque Vh<0, mais dans ce cas la résistance série du commutateur n'arrive jamais aux valeurs Ron et Roff. La résistance série du commutateur au niveau 2 suit l'équation ci-dessous :

$$\frac{1}{R_{serie}} = e^{\left(A\cdot \mathrm{atan}\left(\frac{V_{cont}-Vt}{|Vh|}\right)+B\right)}$$

$$A = \frac{ln\left(\frac{Roff}{Ron}\right)}{\pi}$$

$$B = \frac{ln\left(\frac{1}{Ron\cdot Roff}\right)}{2}$$

Enfin, afin de rendre possible l'analyse de leur influence, six versions de commutateurs ont été créées : NMOS, PMOS et CMOS avec ou sans injection de charge. Les schémas et les symboles sont présentés sur la Figure 3-7 .

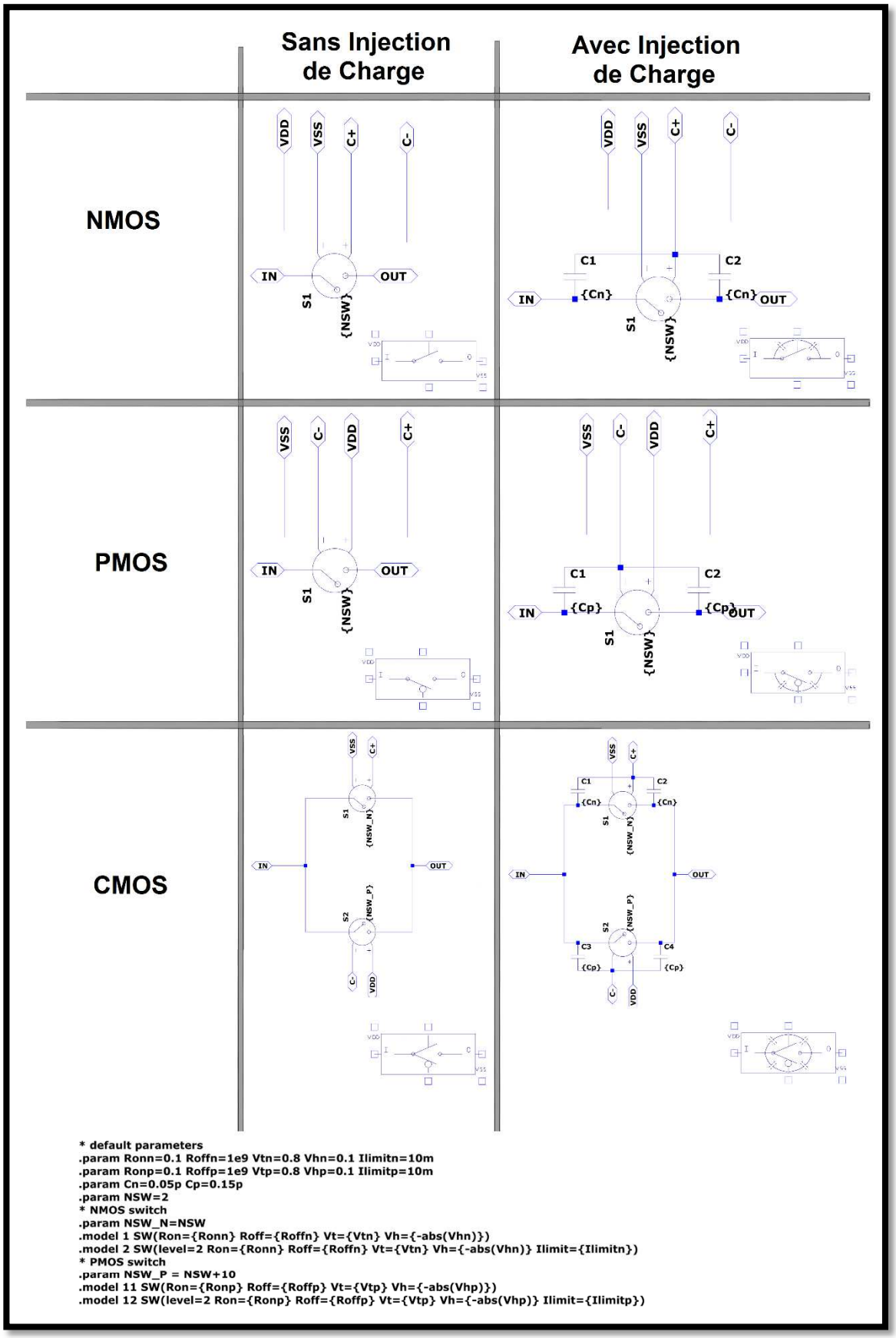


Figure 3-7 – Schémas, symboles et paramètres des commutateurs NMOS, PMOS et CMOS

3.4 COMPARATEURS SYNCHRONES

Suite à la description vue sur le paragraphe 0, les comparateurs utilisés dans l'architecture du CAN Cyclique RSD doivent comparer les signaux d'entrée à un instant défini par l'horloge du convertisseur, cela veut dire que le comparateur doit être synchrone. Dans ce but, un sous-circuit a été créé avec un comparateur comportemental asynchrone et une bascule D.

Le comparateur comportemental asynchrone utilise la fonction LTSPICE appelée SCHMITT possédant les paramètres *Vhigh*, *Vlow*, *Trise*, *Tfall*, *Vt* et *Vh*. Les quatre premiers paramètres cités caractérisent un signal numérique, notamment de la sortie du comparateur. Vhigh/Vlow sont les tensions de bits 1 et 0 respectivement et Trise/Tfall sont les temps de montée et de descente du signal numérique. *Vt* est le seuil de tension à l'entrée du comparateur et *Vh* est la valeur de l'hystéresis (définie à 0 dans cette conception).

Ensuite, la bascule D est instanciée en ayant son entrée reliée à la sortie du comparateur asynchrone. Les paramètres numériques sont les mêmes que ceux de la fonction SCHMITT.

Le schéma du comparateur synchrone complet est présenté sur la Figure 3-8 .

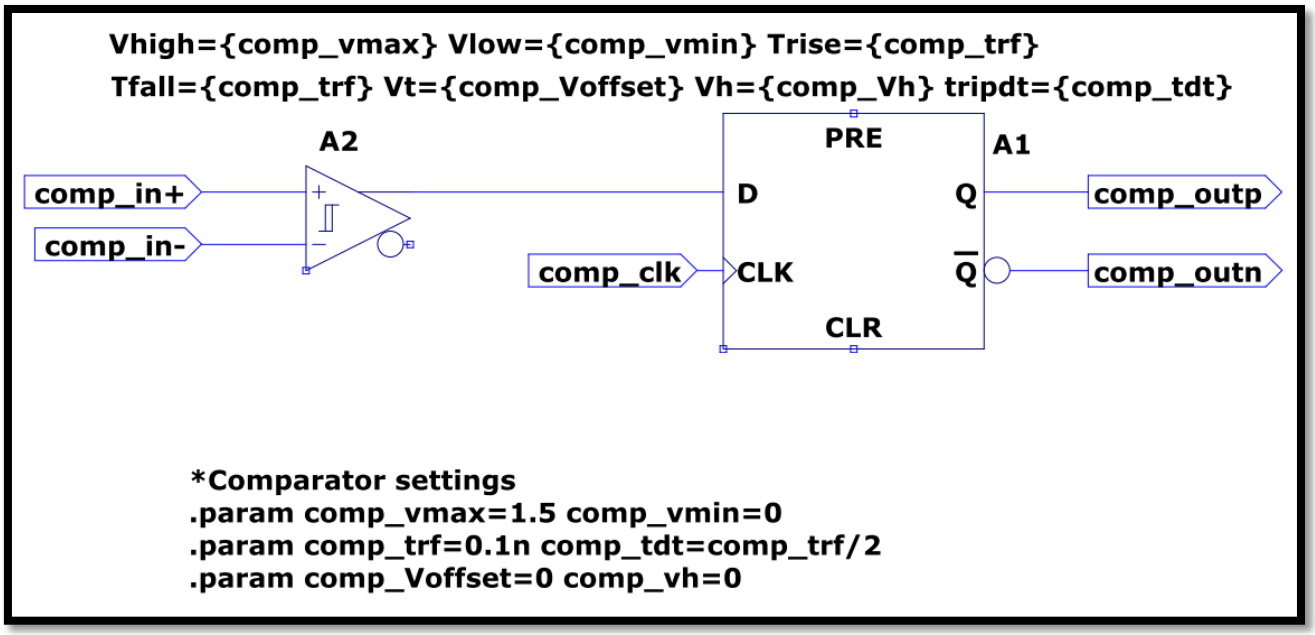


Figure 3-8 –Comparateur synchrone

3.5 CIRCUITS LOGIQUES

Pour que les signaux P_iQ_i sortant du sous-CAN soient convertis vers le mot numérique résultant de la conversion A/N, on rappelle qu'une chaîne d'additionneurs doit

être instanciée en plus de deux registres à décalage de N bits à l'entrée et N registres à décalage de 1 bit à la sortie.

Tout d'abord, les additionneurs complets sont créés à partir de trois portes logiques XOR comme le montre la Figure 3-9. On remarque que ce circuit explore une particularité de la porte XOR sur le logiciel LTSPICE. Dans le cas présent, la porte XOR à plusieurs entrées retourne une sortie positive lorsqu'une entrée seule est positive, à la différence de la plus courante qui retourne la sortie positive quand un nombre impair d'entrées est positif. Clairement, lorsque le nombre d'entrées est égal à 2, les deux types de XOR sont identiques. Ainsi, la porte XNOR produisant C_out ne retourne une sortie négative que dans les cas où une seule entrée est positive ou aucune entrée n'est positive.

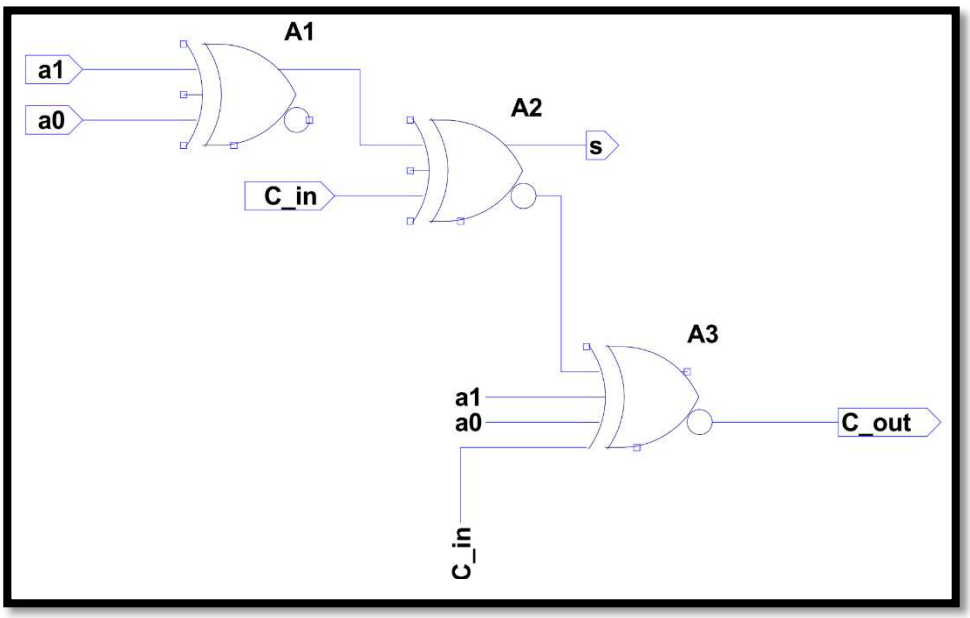


Figure 3-9 - Additionneur complet

Ensuite, pour que les signaux P_i et Q_i de toutes les étapes du CAN Cyclique RSD puissent être manipulés ensemble, deux registres à décalage de N bits sont instanciés à l'entrée du circuit logique à travers N bascules D en cascade. En outre, à la sortie du circuit, N autres bascules D sont utilisées pour instancier une registre à décalage de N bits. Ce dernier registre évite que la sortie du convertisseur ne bouge avant que la conversion soit terminée. Dans ce but, le signal *enable* est utilisé comme horloge.

Le circuit logique complet est présenté sur la Figure 3-10. *ain* représente P_i, *bin* représente Q_i, *clk* est l'horloge et *en* est le signal *enable*.

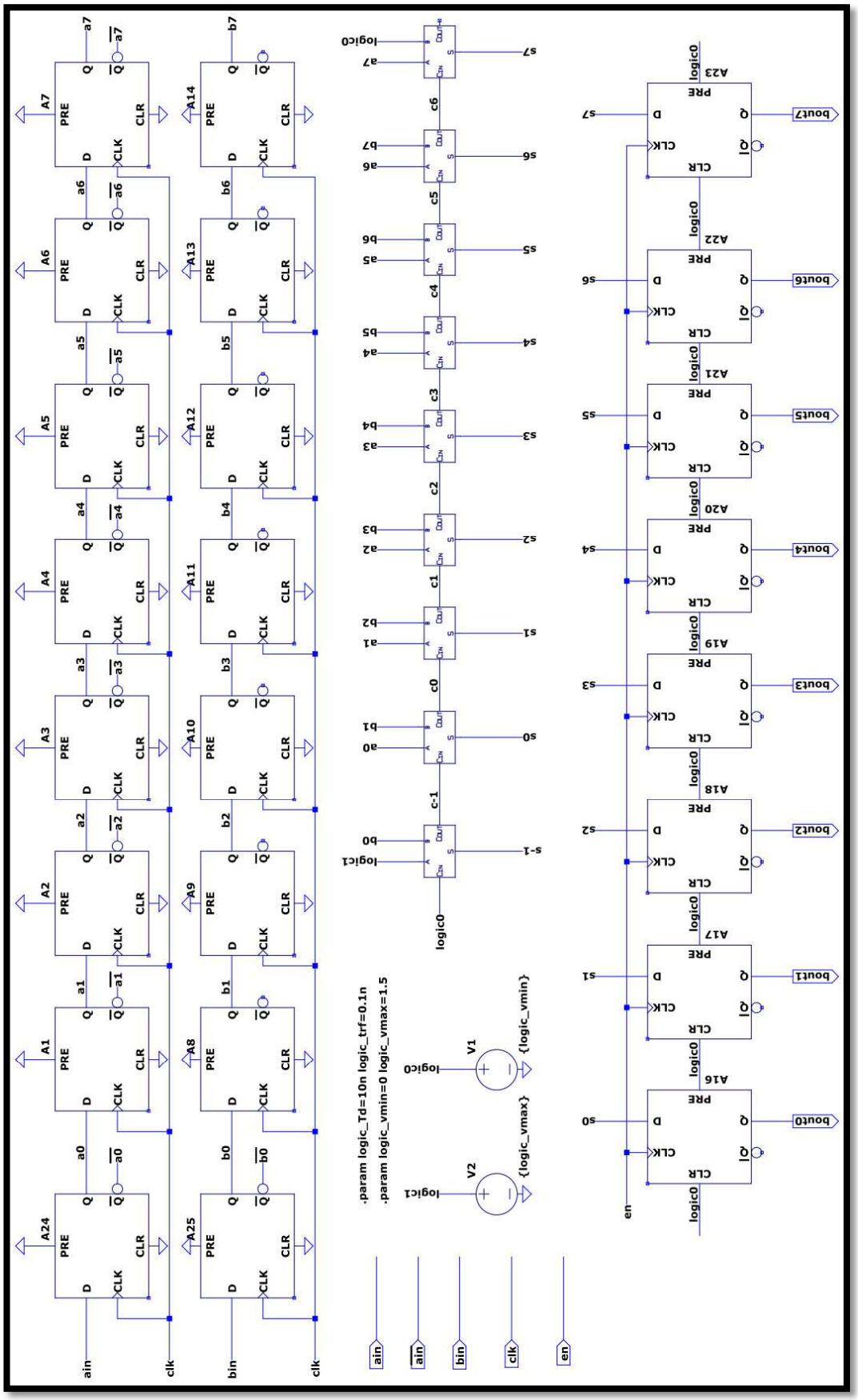


Figure 3-10 - Circuit logique du CAN Cyclique RSD

4. RESULTATS ET DISCUSSION

Suite aux développements réalisés au fur et à mesure des chapitres précédents, un CAN Cyclique RSD à 8 bits de résolution a été conçu et simulé. Ce chapitre vise à présenter le circuit complet, les résultats de simulations et des analyses.

Le circuit modélisé présenté dans ce chapitre est une première étape en vue d'une future analyse complète des non-idéalités d'un CAN Cyclique RSD. Ainsi, le circuit développé utilise les modèles de condensateurs linéaires et commutateurs NMOS sans injection de charge. Toutefois, le modèle retenu ici pour les OTAs est le modèle complet avec limiteur de tension à diodes.

L'architecture conçue est une hiérarchie de sous-circuits de sorte que le système complet reste clair et organisé. En outre, la hiérarchie permet que les changements réalisés dans un sous-circuit spécifique ne nécessite pas de modification dans les autres circuits qui l'englobe. On remarque encore que, afin de rendre plus facile la manipulation du CAN, tous les composants, sous-circuits et simulations mettent à disposition leurs paramètres importants en symboles, ce qui permet leur modification à un niveau plus élevé. Ainsi, ces paramètres sont accessibles aussi bien au niveau des sous-circuits mais aussi au niveau du circuit complet.

4.1 ECHANTILLONNEUR/BLOQUEUR

Le circuit E/H décrit sur la Section 0 est construit sous LTSPICE, à l'aide des modèles de commutateur et OTA précédents. La Figure 4-1 montre sa modélisation.

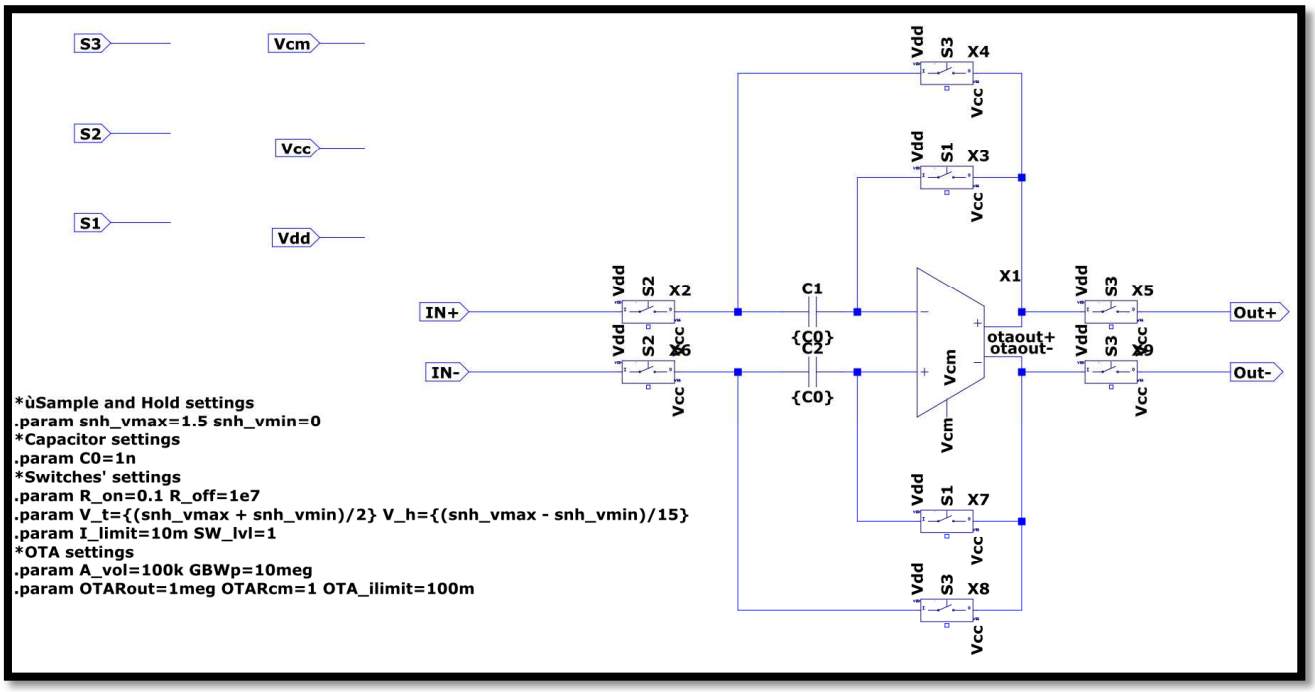


Figure 4-1 - E/H sur LTSPICE

Pour tester l'E/H, un circuit de banc d'essai a été créé dans lequel on simule un signal sinusoïdal à l'entrée de l'E/H.. Les résultats validant le fonctionnement de l'E/H sont montrés sur la Figure 4-2. Les pointes apparues sur la courbe sont dues à la courte période de temps pendant laquelle l'OTA reste en boucle ouverte. En tout cas, on constate que la sortie du circuit suit bien l'entrée.

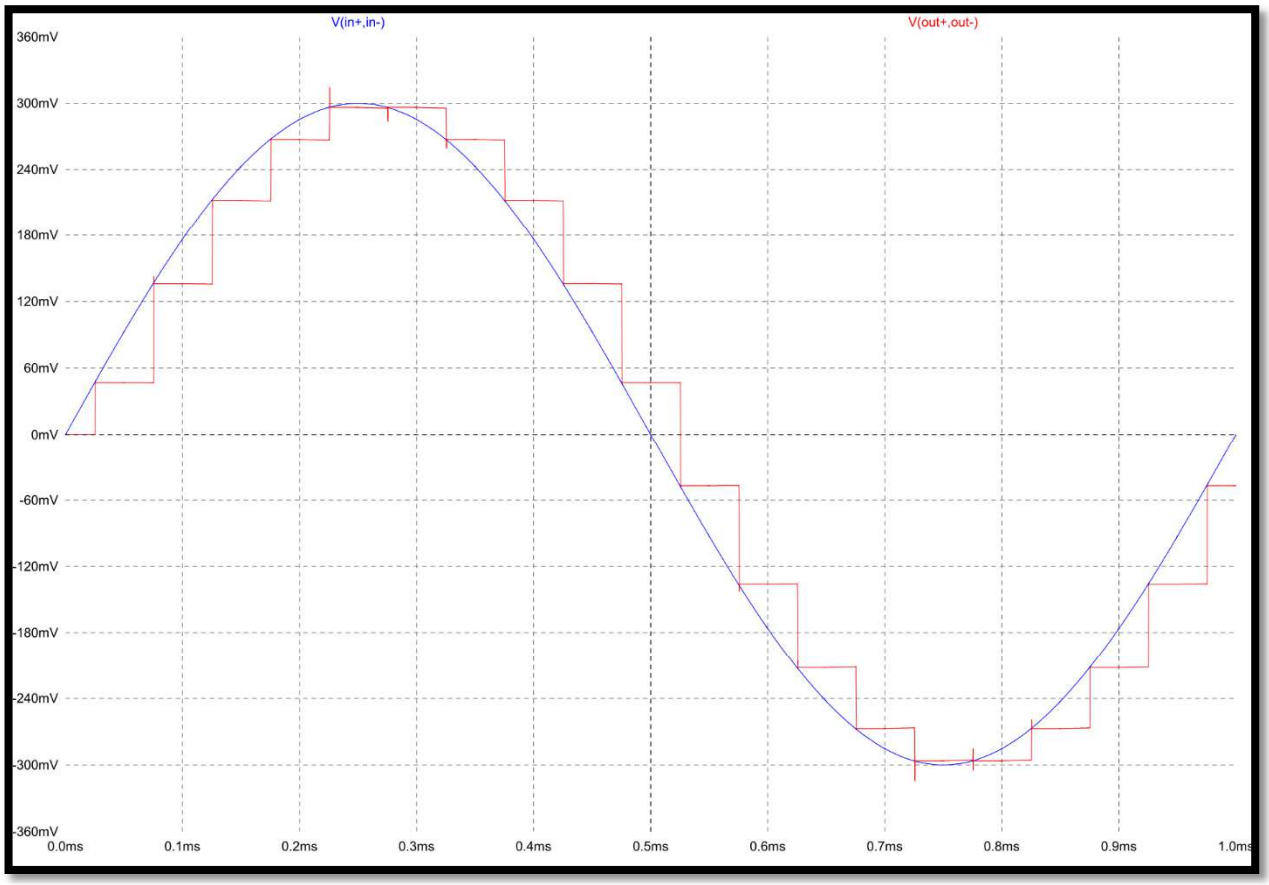


Figure 4-2 - Vérification de fonctionnement de l'E/H

4.2 Soustracteur et Multiplicateur

Le circuit Soustracteur/Multiplicateur est modélisé en s'appuyant sur de la description de l'E/H et la description donnée dans la Section 2.3.3. Le circuit implémenté sur LTSPICE est montré sur la Figure 4-3.

Comme pour l'E/H, pour que le fonctionnement du circuit soit validéle circuit soustracteur/bloqueur est simulé à partir d'un banc d'essai où une sinusoïde est utilisée comme entrée. Le résultat est affiché sur la Figure 4-4. On remarque qu'il est présenté avec un signal référence divisé par 2 et sommé par la valeur soustrait (V_{cm}), si considérée avant la multiplication) de manière à ce que la comparaison soit plus facile.

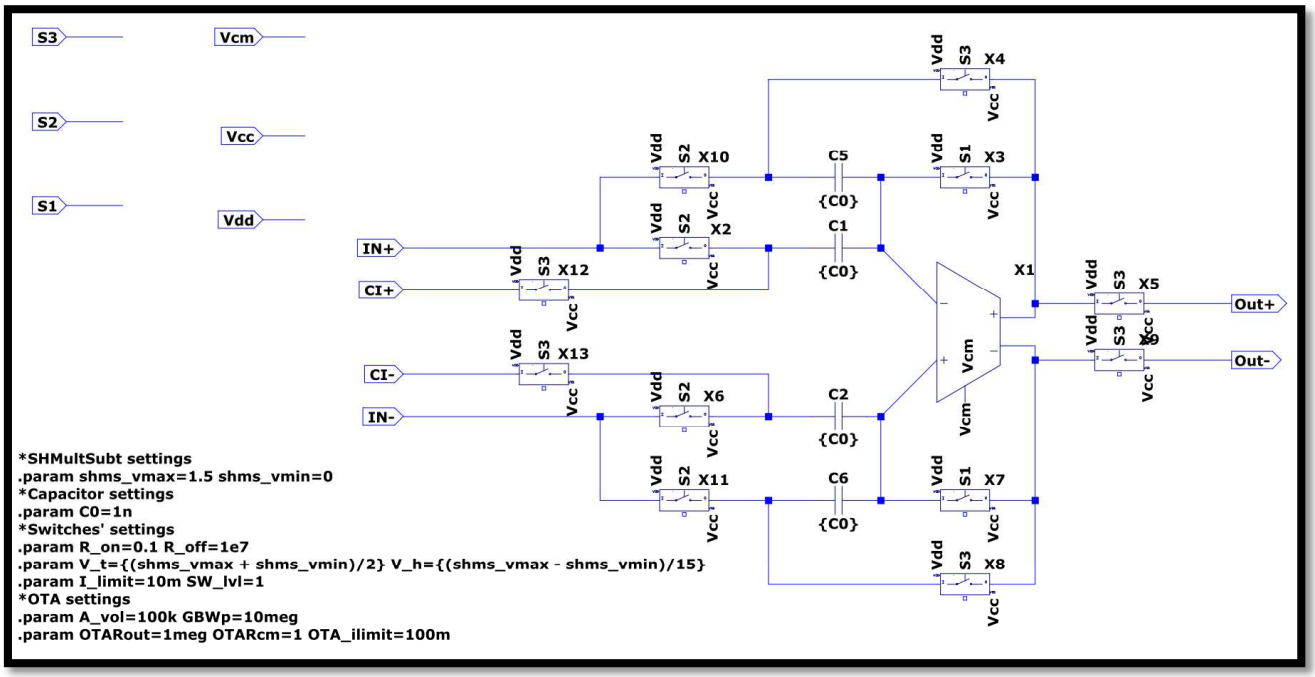


Figure 4-3 - Soustracteur/Multiplicateur implémenté sur LTSPICE

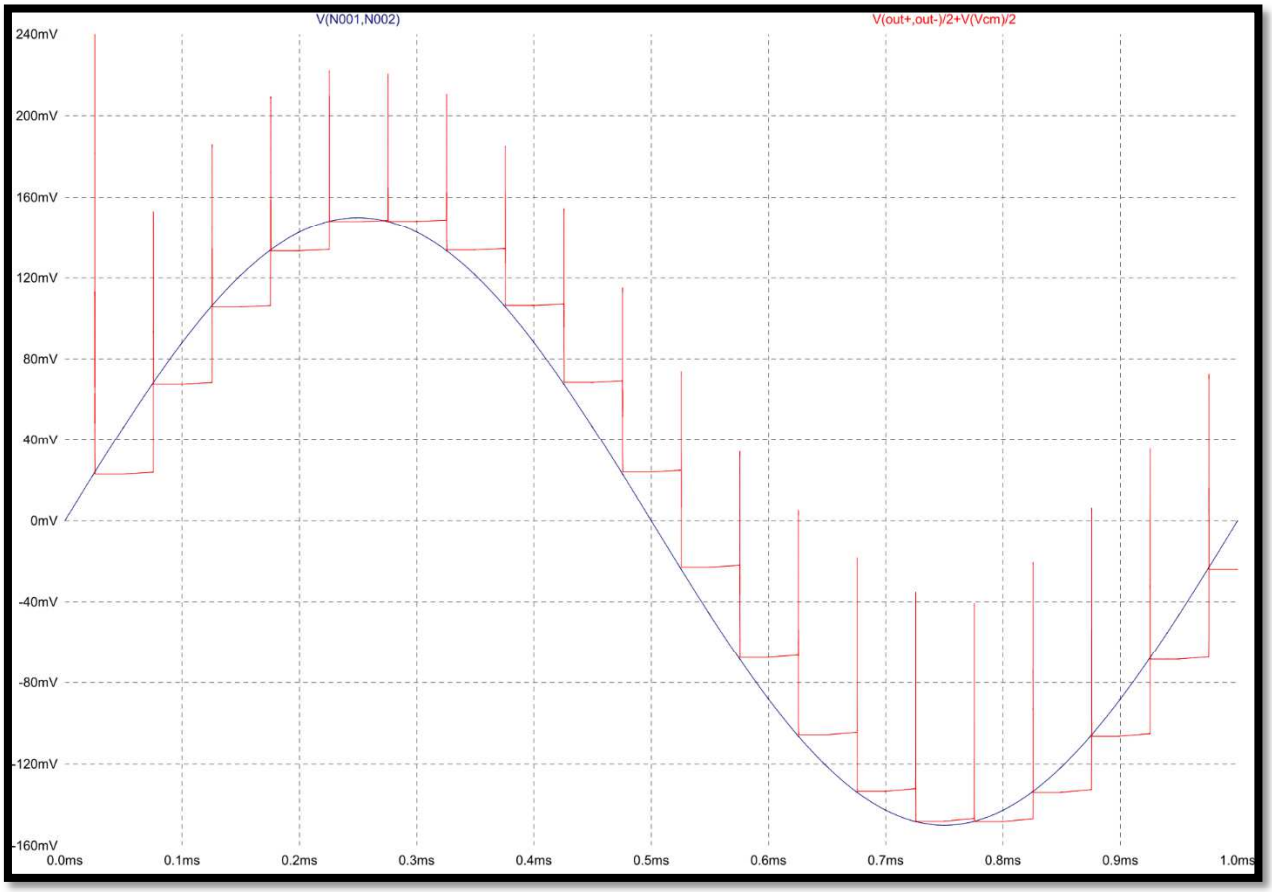


Figure 4-4 - Test du circuit Soustracteur/Bloqueur

4.3 CIRCUIT RSD

Dans l'intention de valider le fonctionnement du sous-CAN (comparateurs synchrones) et du CNA (MUXs) du CAN Cyclique RSD, la fonction de transfert d'un cycle seul du CAN a été testée. La fonction de transfert a été affichée avec sa réponse idéale sur la Figure 4-5 pour que l'on puisse les comparer.

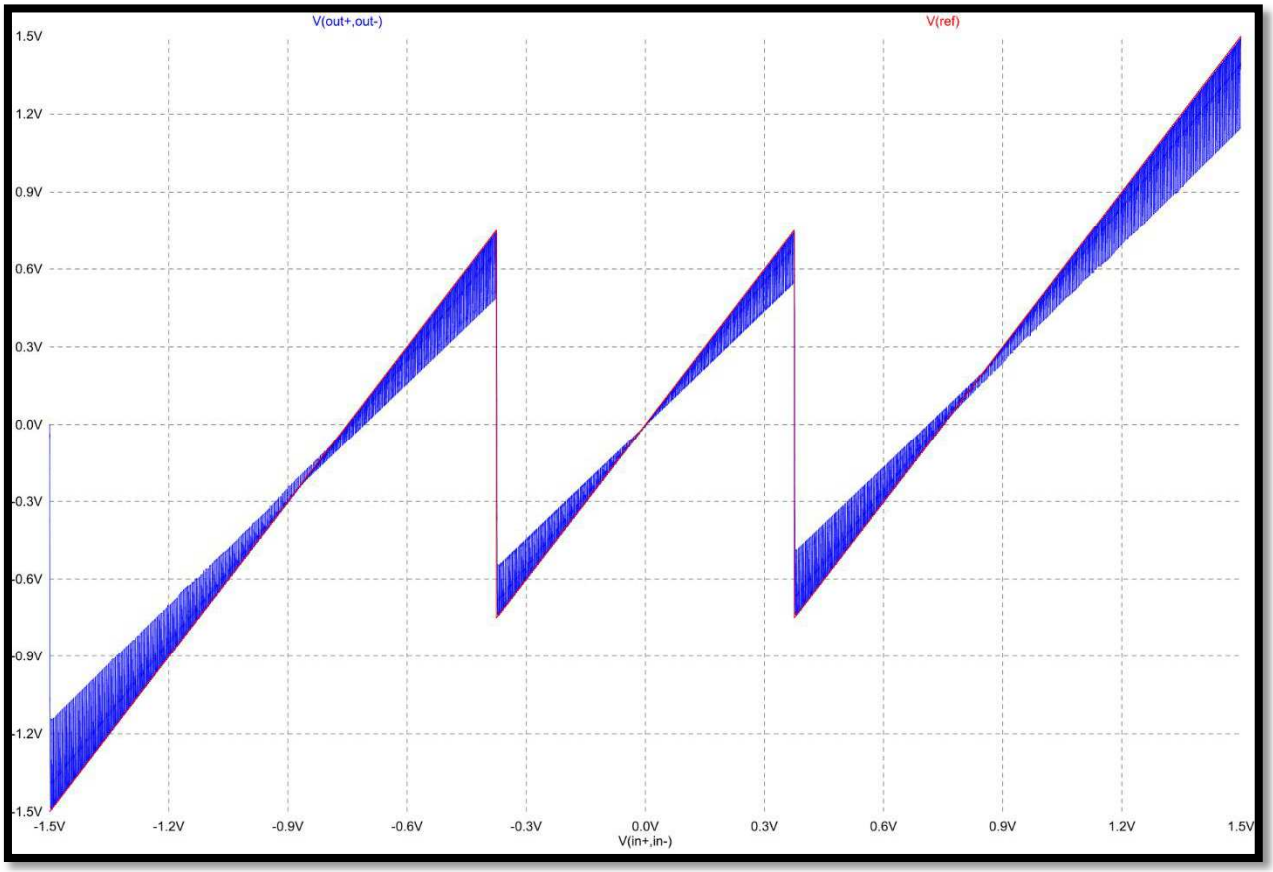


Figure 4-5 - Teste de la fonction RSD implementé sur LTSPICE

Il est vu que la réponse du circuit suit le réponse idéale, mais une sorte de bruit est présent sur toute la courbe. Cette ondulation observée est due à la période pendant laquelle l'OTA reste en boucle ouverte. Durant cette période, un régime instable est déclenché et la sortie commence à bouger de manière instable jusqu'au moment où la boucle est fermée à nouveau ce qui stabilise la sortie.

4.4 CAN CYCLIQUE RSD

Après avoir présenté chaque circuit séparément, on les rassemble tous afin de modéliser et simuler le CAN Cyclique RSD. Le schéma du plus haut niveau est montré sur la Figure 4-6. On trouve sur ce schéma le convertisseur A/N complet (entouré en rouge), les références de tensions continues (entourés en jaune), les signaux d'entrées (entourés en orange), les signaux de contrôle du CAN (entourés en bleu), un CNA idéal pour

reconstruire la conversion A/N (entouré en violet) et finalement les paramètres de l'ensemble du circuit (entouré en vert).

Dans cette version du CAN, on définit V_{dd}=1.5, V_{ss}=0, V_{ref}=V_{dd} et V_{cm}=V_{dd}/2. Afin de tester la conversion A/N, deux types de simulations ont été faites : une utilisant une sinusoïde à l'entrée et l'autre utilisant une rampe. Pour le deux cas on remarque que le signal appelé *ref,* considéré comme une entrée (dans le cadre orange), n'est utilisé que pour comparer, à l'affichage de courbes, la sortie du convertisseur avec l'entrée. En effet, *ref* est égal au signal d'entrée, in=in_+-in_-, mais décalé 8 F_s, ce qui est la latence de la conversion du CAN Cyclique de 8 bits.

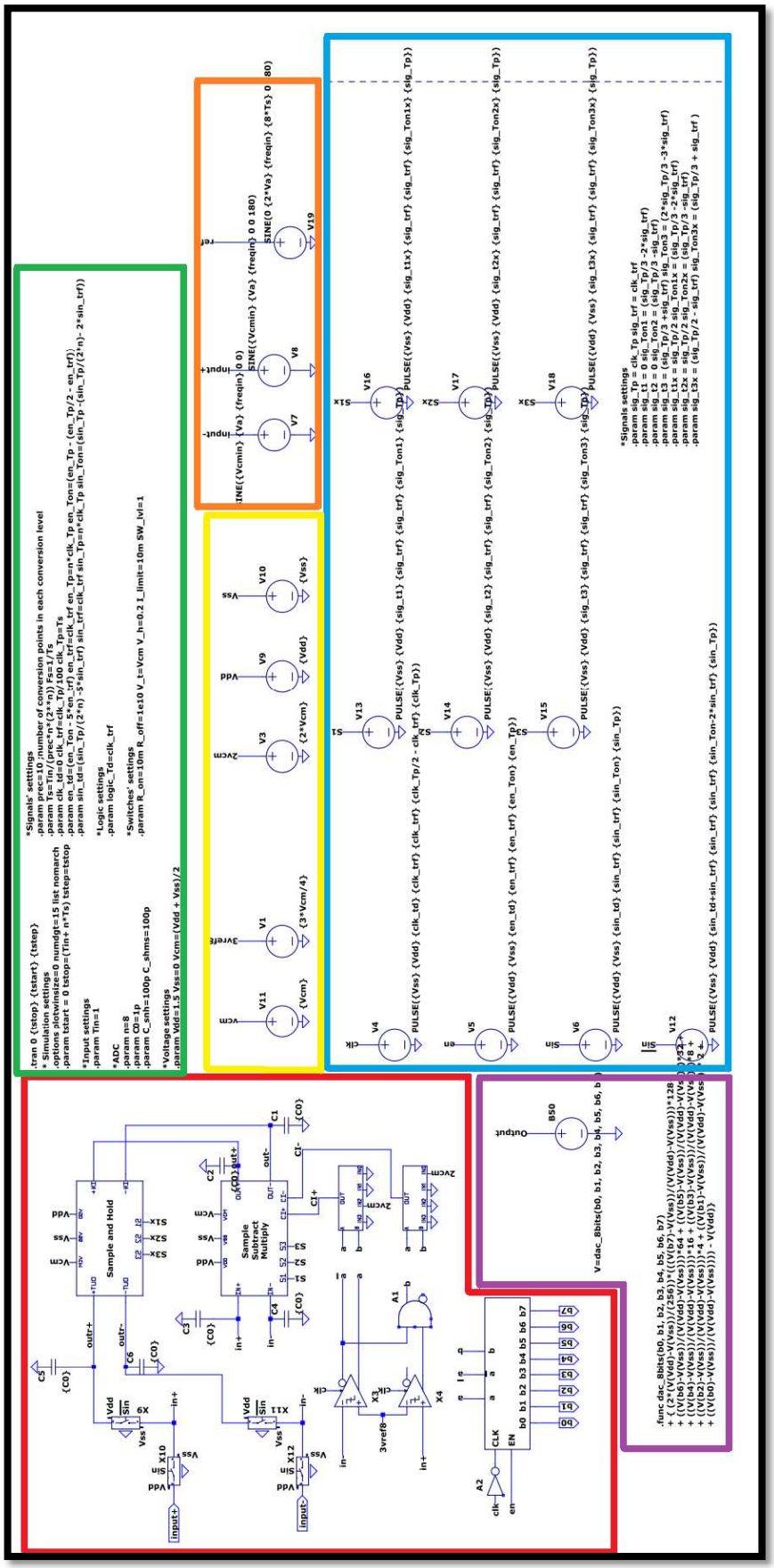


Figure 4-6 – Banc d'essai du CAN Cyclique RSD de 8 bits

Pour que la conversion soit accomplie, il est primordial de bien respecter l'ordre entre les signaux de contrôle. En général, deux commutateurs adjacents ne doivent pas devenir ouverts au même temps. Dans un circuit de capacités commutées, la fuite de charge générée serait prohibitive pour le résultat de la conversion.

Les signaux de contrôle utilisés dans les circuits E/H et Soustracteur/Multiplicateur sont présentés sur la Figure 4-7. Il est possible de voir que S1 descend juste avant la descente de S2 qui se fait juste avant la montée de S3. Ensuite, pour revenir à l'état initial, S3 descend avant e tout de suite S1 et S2 montent, mais cette fois-ci ensemble, car lors de la fermeture des commutateurs, le problème d'injection de charge n'est pas gênant.

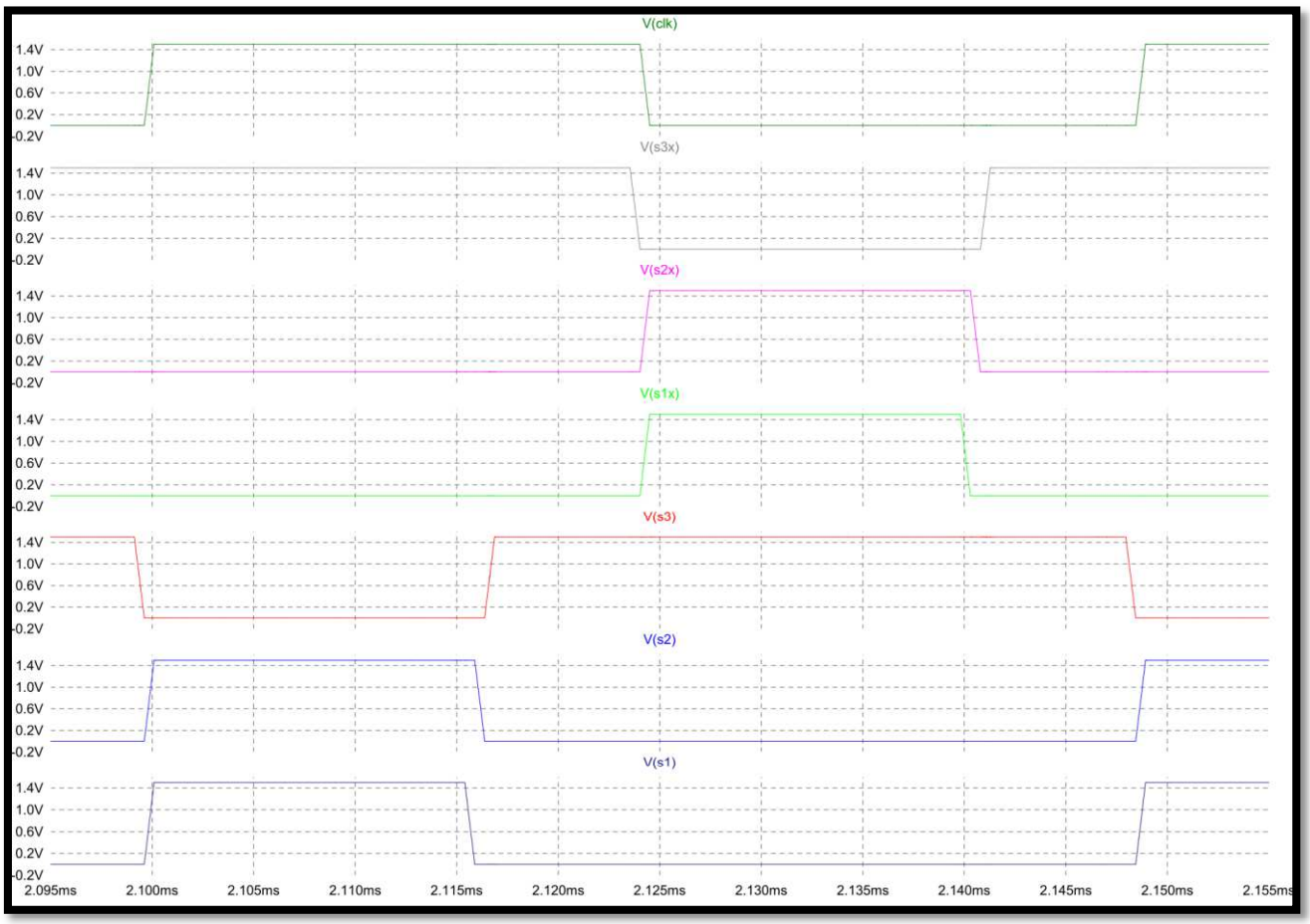


Figure 4-7 - Signaux de Contrôle de l'E/H et du Soustracteur/Multiplicateur

La Figure 4-8 présente, à son tour, les courbes du signal *sin*, qui contrôle le commutateur à l'entrée du CAN et du signal *en*, signal *enable,* qui contrôle la sortie du circuit logique. La première remarque est que signal sin et son inverse ne se chevauchent pas afin d'éviter toujours les fuites de charge. La deuxième remarque est que le signal *en*

monte juste avant la montée de l'horloge *clk* pour assurer que le circuit combinatoire dans le circuit logique du CAN utilise les bits P_iQ_i corrects.

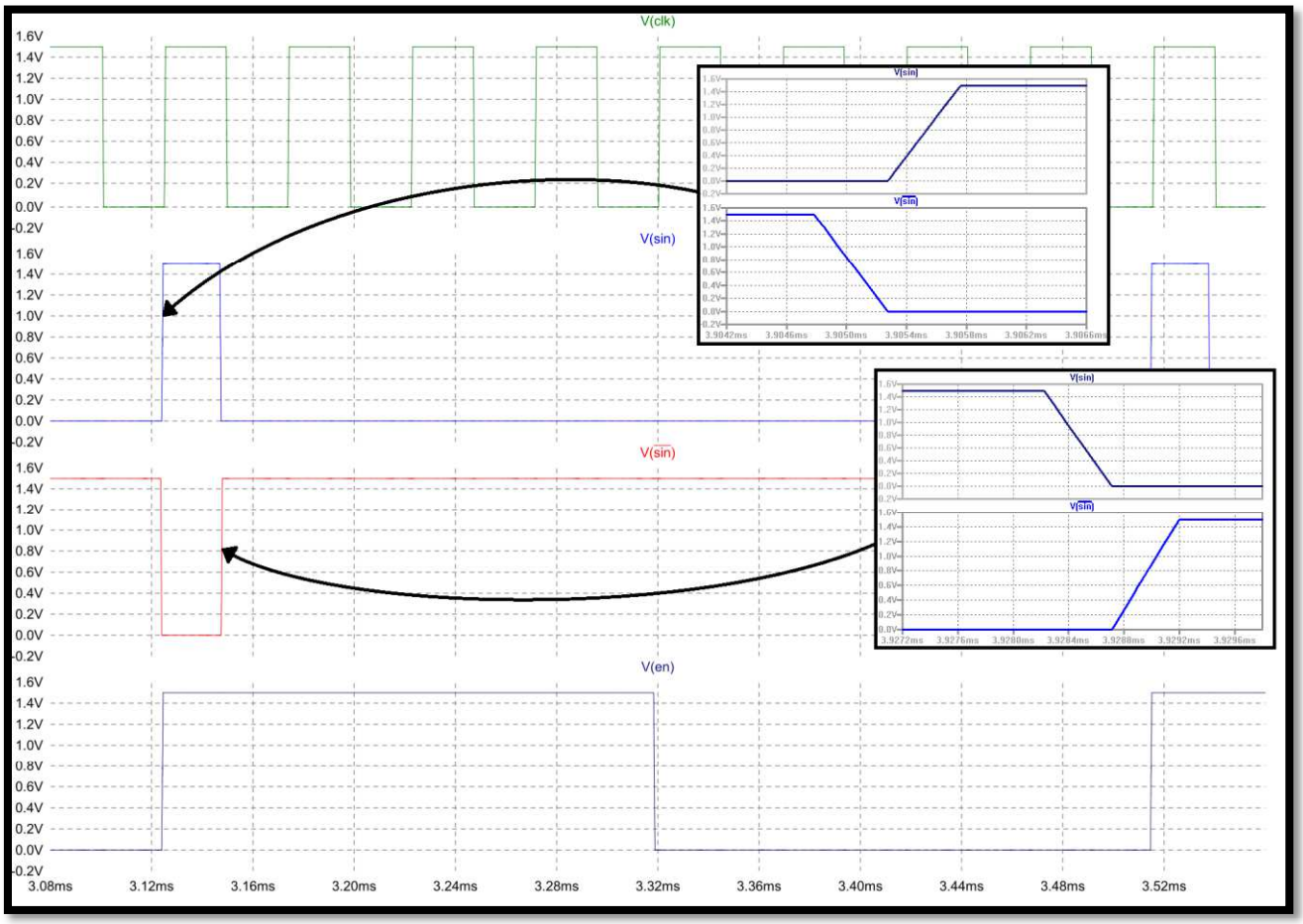


Figure 4-8 - Signaux de Contrôle du commutateur de l'entrée et *enable*

Le premier résultat de conversion, montré sur la Figure 4-9, est celui d'une rampe à l'entrée. La première courbe est l'entrée du convertisseur, alors que celles qui sont au-dessous sont les bits convertis. A l'aide du CNA idéal, la tension analogique construiteà partir des bits convertis caractérise la fonction de transfert du CAN, dans le cas où la rampe est utilisée comme entrée. Cette fonction, accompagnée de l'erreur de quantification, est présentée sur la Figure 4-10

Sur cette dernière figure, il est difficile d'évaluer visuellement la qualité de la fonction de transfert puisque la conversion comprend 8 bits de résolution et donc 256 niveaux discrets de tension. Cependant le calcul des Erreurs d'*offset* et de gain se calculent facilement à partir des équations décrites dans la Section 0. On obtient :

$$E_{offset} = 0.010\ LSB$$

$$E_{gain} = -0.036$$

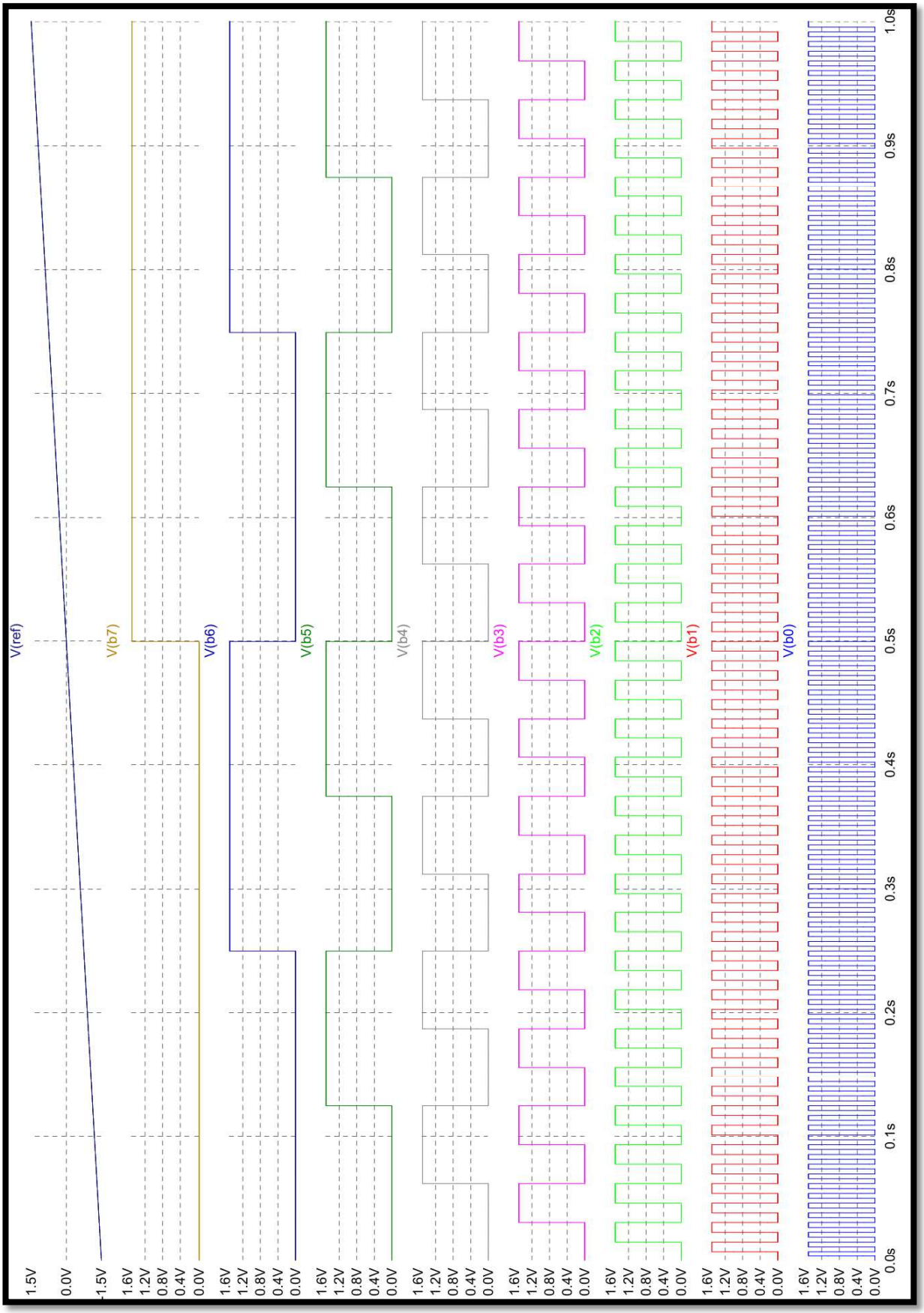


Figure 4-9 - Conversion A/N Cyclique RSD de 8 bits

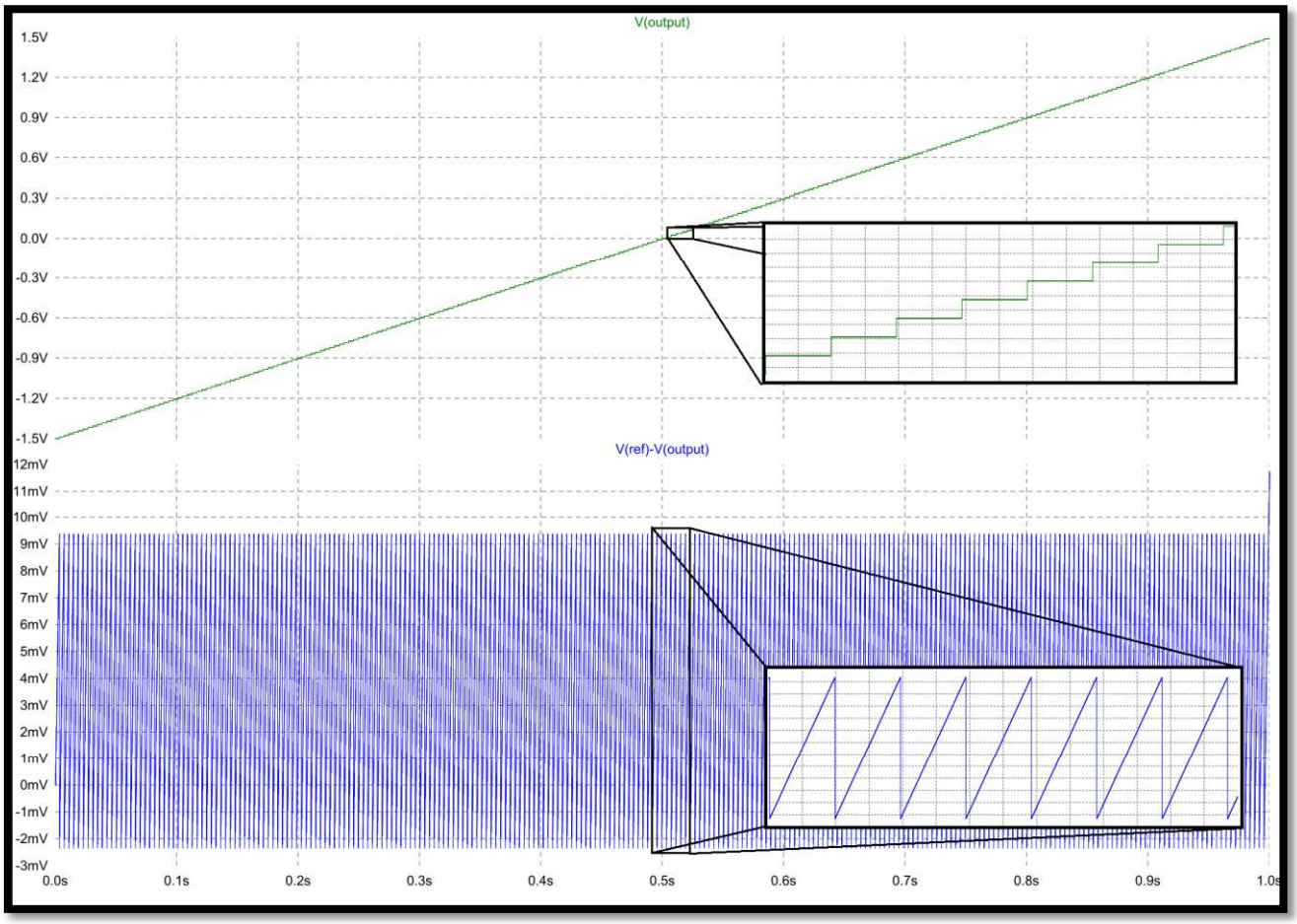


Figure 4-10 - Fonction de Transfert et Erreur de Quantification du CAN Cyclique RSD 8 bits

Ensuite, pour le deuxième test on a excité l'entrée du CAN avec une sinusoïde de fréquence F_{in}=1 Hz et de tension variant entre -V_{ref} et V_{ref} (variation entre 0 e V_{ref} sur chaque branche in_+ et in_-). De plus, la fréquence de l'horloge du convertisseur a été définie comme F_s=20,45 kHz, ce qui commande 2560 conversions par seconde. Le résultat de simulation est montré sur la Figure 4-11.

Enfin, pour confirmer un bon résultat de linéarité, la FFT de la sortie du CAN est affichée sur la Figure 4-12. Pour arriver à ce résultat, 8192 points de la sortie du CAN ont été utilisés après une simulation durant cinq périodes de la sinusoïde à l'entrée. Les mêmes résultats sont obtenus avec la commande SPICE *.four* ci-dessous, d'où on calcule que le SFDR vaut 70,2 dBc.

$$SFDR = 20 \cdot \log\left(\frac{1.499}{4.624 \cdot 10^{-4}}\right) = 70.2\ dBc$$

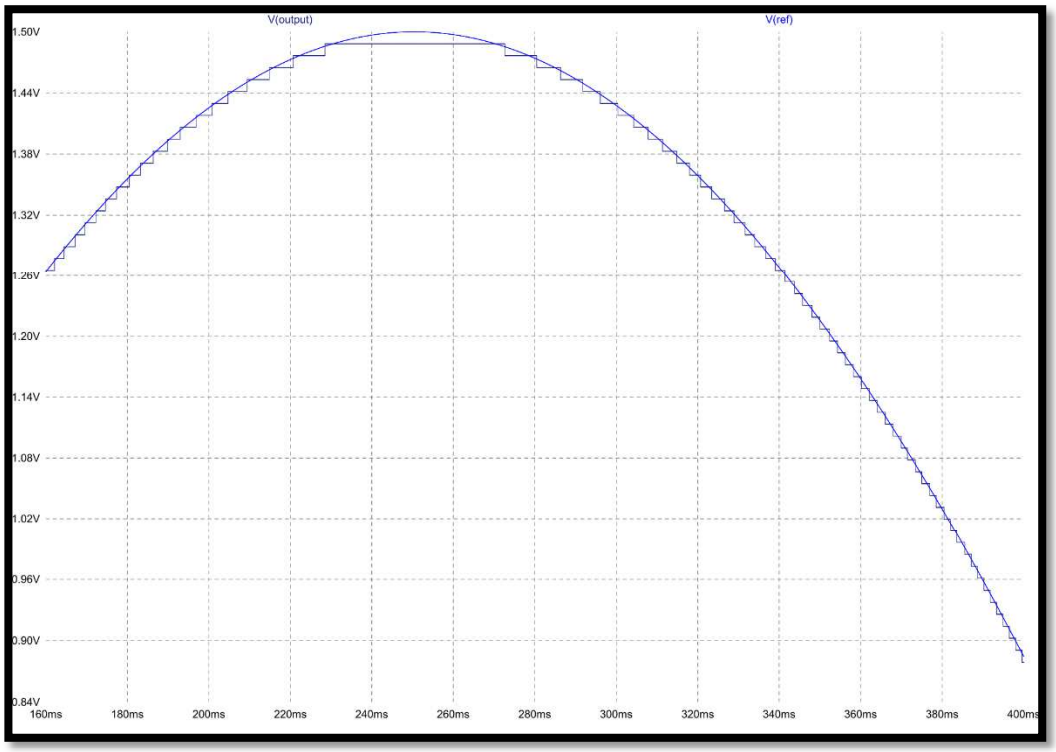
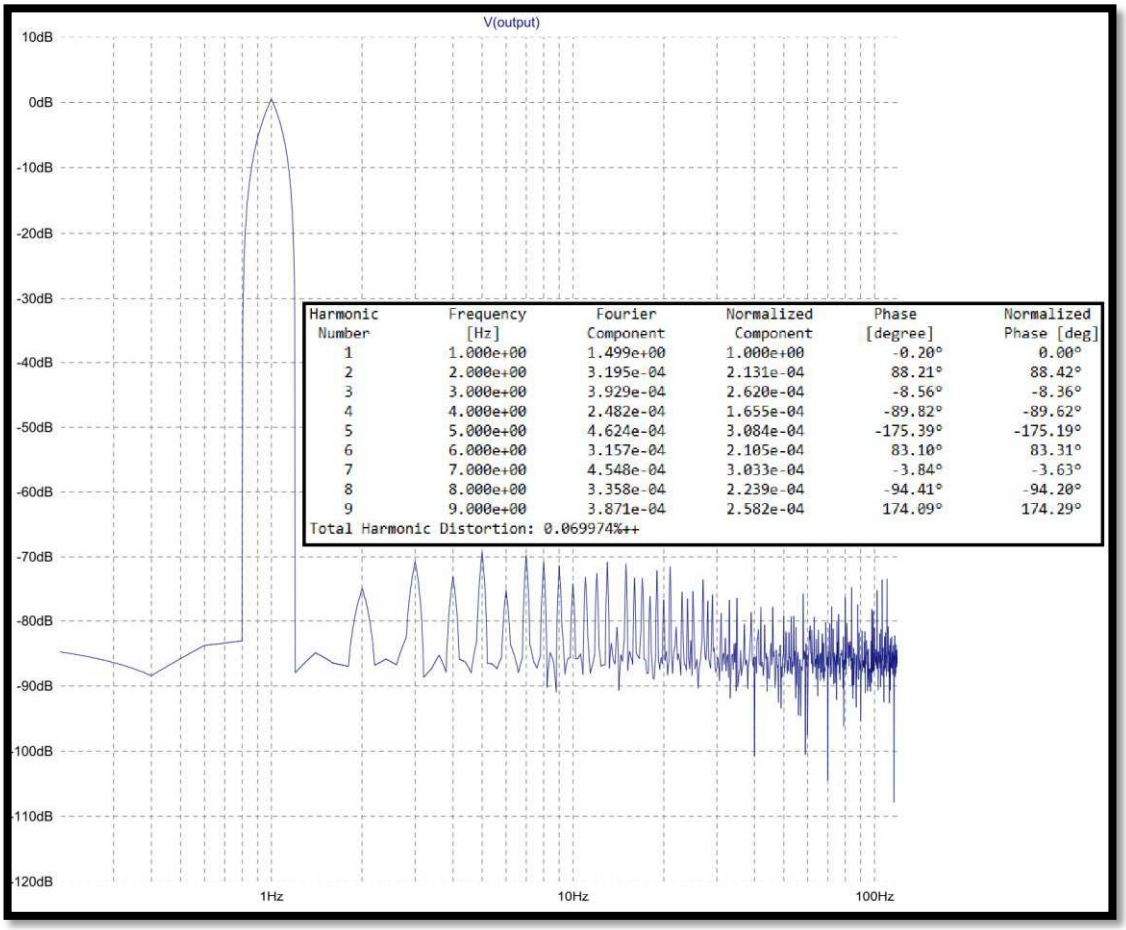


Figure 4-11 - Conversion A/N d'une sinusoïde



Harmonic Number	Frequency [Hz]	Fourier Component	Normalized Component	Phase [degree]	Normalized Phase [deg]
1	1.000e+00	1.499e+00	1.000e+00	-0.20°	0.00°
2	2.000e+00	3.195e-04	2.131e-04	88.21°	88.42°
3	3.000e+00	3.929e-04	2.620e-04	-8.56°	-8.36°
4	4.000e+00	2.482e-04	1.655e-04	-89.82°	-89.62°
5	5.000e+00	4.624e-04	3.084e-04	-175.39°	-175.19°
6	6.000e+00	3.157e-04	2.105e-04	83.10°	83.31°
7	7.000e+00	4.548e-04	3.033e-04	-3.84°	-3.63°
8	8.000e+00	3.358e-04	2.239e-04	-94.41°	-94.20°
9	9.000e+00	3.871e-04	2.582e-04	174.09°	174.29°

Total Harmonic Distortion: 0.069974%++

Figure 4-12 – FFT d'une conversion A/N d'une sinusoïde

5. CONCLUSIONS

Dans le contexte des simulations SPICE, la macro-modélisation est une approche très efficace pour optimiser la simulation de circuits complexes. Les macro-modèles utilisent des éléments fonctionnels avec des modèles non-linéaires précis dans le but d'évaluer de manière comportementale les réponses d'un circuit semi-conducteur complexe. Pour des analyses de circuits complexes, comme celle d'un banc de Convertisseurs Analogiques/Numériques en colonnes parallèles utilisé dans un Capteur d'Image à Pixels Actifs CMOS, la macro-modélisation est très puissante. Pour cela, il faut modéliser l'ensemble d'un CAN à l'aide de macro-modèle.

Mon travail a été de concevoir une architecture pour le modèle du CAN cyclique qui permette dans l'avenir de faire des analyses fines et efficaces de l'impact des défauts de chaque sous-circuit qui le compose. A cette fin, j'ai dû macro-modéliser des circuits sur le logiciel LTSPICE de sorte que leurs paramètres, leurs défauts y compris, soient facilement accessibles sur les niveaux plus hauts de la hiérarchie de conception. Aussi, il m'a fallu acquérir des connaissances en Convertisseurs A/N, capteurs CMOS APS, macro-modélisation et simulateurs *SPICE*.

En somme, à la fin des travaux j'ai pu simuler un CAN cyclique RSD complet à travers la macro-modélisation des sous-circuits utilisés dans l'architecture proposée. Ces derniers sont : l'OTA différentiel avec limitation de tension, les commutateurs NMOS, PMOS et CMOS avec ou sans injection de charge, les condensateurs linéaires et non-linéaires avec ou sans des condensateurs parasitiques de plaque inférieure, etc. Sur le schéma du plus haut niveau les paramètres des modèles sont affichés et peuvent être modifiés. Les résultats de conversion d'un CAN cyclique RSD de 8 bits sont en accord avec la théorie, ce qui nous mène à conclure que la modélisation est correcte.

En ce qui concerne les travaux à l'avenir, le développement de ces macro-modèles permettra de futures analyses de la topologie cyclique et de l'influence de toutes les non-idéalités sur les performances de la conversion A/N. De plus, ces modèles rendront éventuellement possible la simulation de plusieurs CANs en parallèle dans le but d'élaborer une calibration numérique d'un banc de CANs. Une fois que la macro-modélisation et la simulation des CANs à colonnes parallèles seront accomplies, l'effort du concepteur pourra se concentrer sur les algorithmes de calibration en s'affranchissant des problématiques de simulation et modélisation.

6. REFERENCES

1. **Fossum, E. R.** CMOS Image Sensors: Electronic camera-on-chip. *IEEE Transactions on Electron Devices.* 1997, Vol. 44, 10.
2. **Jong-Ho Park, Satoshi Aoyama, Takashi Watanabe, Keigo Isobe and Shoji Kawahito.** A High-Speed Low-Noise CMOS Image Sensor with 13-b Column-Parallel Single-Ended Cyclic ADCs. *IEEE Transactions on Electron Devices.* 2009, Vol. 56, 11.
3. *A Low-Noise High Intrascene Dynamic Range CMOS Image Sensor with a 13 to 19b Variable-Resolution Column-Parallel Folding-Integration/Cyclic ADC.* **Min-Woong Seo, Sung-Ho Suh, Tetsuya Iida, Taishi Takasawa, Keigo Isobe, Takashi Watanabe, Shinya Itoh, Keita Yasutomi and Shoji Kawahito.** 1, s.l. : IEEE Journal of Solid-State Circuits, 2012, Vol. 47.
4. *A 240-frames/s 2.1 Mpixel CMOS Image Sensor with Column-Shared Cyclic ADCs.* **Seunghyun Lim, Toungcheol Chae, Wunki Jung, Dong-Hun Lee, Kwisung Yoo, Seoghen Ham and Gunhee Han.** 9, s.l. : IEEE Journal of Solid-State Circuits, 2011, Vol. 46.
5. **Medawar, Samer.** *Pipeline Analog-Digital Converters Dynamic Error Modeling for Calibration.* Stockholm : KTH Electrical Engineering, 2012. Doctoral Thesis.
6. **Bernal, Olivier.** *Conception de Convertisseurs Analogique-Numérique en technologie CMOS basse tension pour chaînes Vidéo CCD Spatiale.* Toulouse : Institut National Polytechnique de Toulouse, 2006. p. 3, Thèse.
7. **Puppala, Ajith kumar.** *Design of a Low Power Cyclic/Algorithmic Analog-to-Digital Converter in a 130nm CMOS Process.* s.l. : Linköping Institute of Technology, 2012.
8. **Linear Technology.** SAR ADC Input Types. *Linear Technology.* [En ligne] 24 07 2013. [Citation : 03 08 2013.] http://www.linear.com/docs/43498.
9. *Fully Differential amplifiers.* **Karki, Jim.** s.l. : Texas Instruments Analog Applications Journal, 2000. slyt165.pdf.
10. **Vladimirescu, Andrei.** *The Spice Book.* s.l. : John Wiley & Sons, Inc., 1994.
11. *Macromodeling of Integrated Circuit Operation Amplifiers.* **Boyle, Graeme R., et al., et al.** 6, s.l. : IEEE Journal of Solid-State Circuits, 1974, Vol. 9.
12. **Jespers, Paul G. A.** *Integrated Converters.* s.l. : Oxford, 2001.
13. *A CMOS 13-b Cyclic RSD A/D Converter.* **Bernard Ginetti, Paul G. Jespers, André Vandemeulebroecke.** 7, s.l. : IEEE Journal of Solid-State Circuits, 1992, Vol. 27.
14. *A CMOS Ratio-Independent and Gain-Insensitive Algorithmic Analog-to-Digital Converter.* **Shu-Yuan Chin, Chung-Yu Wu.** 8, s.l. : IEEE Journal of Solid-State Circuits, 1996, Vol. 31.
15. *A Cyclic A/D Converter that does not require Ratio-Matched Components.* **Hidetsohi Onodera, Tetsuo Tateishi, Keikichi Tamaru.** 1, s.l. : IEEE Journal of Solid-State Circuits, 1988, Vol. 23.
16. **B. Ginetti, A. Vandemeulebroecke, P. Jespers.** RSD Cyclic Analog-to-Digital Converter. *Symposium on VLSI Circuits.* 1988.
17. *A 12-b 600 ks/s Digitally Self-Calibrated Pipelined Algorithmic ADC.* **Lee, Hae-Sung.** 4, s.l. : IEEE Journal of Solid-State Circuits, 1994, Vol. 29.
18. **Baker, R. Jacob.** *CMOS Mixed-Signal Circuit Design.* s.l. : John Wiley & Sons, 2002.