



HAL
open science

Test intégré pour Convertisseurs Analogique/Numérique

Serge Bernard

► **To cite this version:**

Serge Bernard. Test intégré pour Convertisseurs Analogique/Numérique. Micro et nanotechnologies/Microélectronique. Université Montpellier II - Sciences et Techniques du Languedoc, 2001. Français. NNT: . tel-00003665

HAL Id: tel-00003665

<https://theses.hal.science/tel-00003665>

Submitted on 31 Oct 2003

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ACADEMIE DE MONTPELLIER

UNIVERSITE MONTPELLIER II

— SCIENCES ET TECHNIQUE DU LANGUEDOC —

THESE

présentée à l'Université Montpellier II, Sciences et Techniques du Languedoc

Pour obtenir le grade de

DOCTEUR de l'Université Montpellier II

SPECIALTE : **ELECTRONIQUE, OPTRONIQUE ET SYSTEMES**

Formation doctorale : Systèmes Automatiques et Microélectroniques

Ecole doctorale : Information, Structures et Systèmes

Test Intégré pour Convertisseurs Analogique/Numérique

Par

Serge BERNARD

Soutenue le **13 avril 2001** devant le jury composé de :

D. AUVERGNE	Professeur à l'Université de Montpellier II	Président
A. KAISER	Chargé de recherche CNRS-ISEN-IEMN	Rapporteur
A. OSSEIRAN	Professeur – Directeur de recherche FLUENCE	Rapporteur
P. FOUILLAT	Professeur – ENSEIRB-Université Bordeaux I	Examineur
F. AZAIS	Chargée de recherche CNRS - Université de Montpellier II	Examineur
M. RENOVELL	Chargé de recherche CNRS - Université de Montpellier II	Examineur
Y. BERTRAND	Professeur - Université de Montpellier II	Directeur de thèse

à Adam, Nawel et Louka

Je tiens à remercier Messieurs les Professeurs Gaston CAMBON et Michel HABIB, ancien et nouveau Directeurs du Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, pour leur accueil au sein de leur laboratoire.

Je remercie vivement Monsieur Andréas KAISER, Chargé de Recherche CNRS et Monsieur Adam OSSEIRAN, Directeur de Recherche à FLUENCE, qui m'ont fait l'honneur d'accepter d'être les rapporteurs de ce mémoire de thèse.

Je voudrais remercier Monsieur Daniel AUVERGNE, Professeur à l'université Montpellier II et Monsieur Pascal FOUILLAT, Professeur à l'université Bordeaux I, pour leur participation au jury de cette thèse.

L'aboutissement de ce travail n'aurait pas été possible sans la contribution de l'ensemble de l'équipe de recherche. Ainsi, je tiens à exprimer ma profonde gratitude envers Madame Florence AZAÏS et Monsieur Michel RENOVELL, respectivement Chargé de Recherche et Directeur de Recherche CNRS. Ils ont été d'un apport essentiel autant pour les idées nouvelles que pour l'exploitation de l'ensemble du travail de cette thèse. Enfin, je tiens à exprimer toute ma reconnaissance à Monsieur Yves BERTRAND, Professeur à l'université Montpellier II, qui a dirigé mes recherches. Il m'a ouvert en grand les portes de la recherche et je ne suis pas prêt d'oublier tous ses conseils, son soutien et la confiance qu'il m'a accordée.

Je tiens à remercier Monsieur Régis LORIVAL pour son aide dans le domaine du test industriel et du test analogique.

Je tiens aussi à remercier Monsieur Guy CATHEBRAS pour ces conseils avisés dans le domaine de la conception analogique et Xavier pour son aide importante dans l'étude des générateurs de signaux analogiques.

Je tiens aussi à exprimer toute mon amitié à mes collègues thésards du laboratoire et à toutes les personnes du LIRMM qui m'ont permis de travailler dans une ambiance agréable.

Enfin je remercie Elise pour son aide et son soutien lors de la rédaction de ce manuscrit.

Table des matières

Introduction générale	7
------------------------------	----------

Chapitre 1

"État de l'Art "	11
I. Concepts de base	12
II. Caractéristiques Fonctionnelles des Convertisseurs A/N	13
II.1 Convertisseur A/N parfait	14
II.2 Convertisseur A/N réel	16
III. Test Industriel des CAN	21
III.1 Test par Histogramme	22
III.2 Analyse spectrale	29
IV. Test Intégré des CAN	34
IV.1 Test intégré avec pré-requis sur les ressources matérielles initiales	35
IV.2 Test intégré sans pré-requis sur la présence de ressources matérielles	38
V. Conclusion	43

Chapitre 2

Analyseur de réponses de test :	45
I. Introduction	46
II. Diminution du volume de données à traiter	48
III. Simplification des calculs d'exploitation	50
III.1. Signal d'entrée triangulaire	50
III.2. Signal d'entrée sinusoïdal	56
III.3. Récapitulatif	62
IV. Minimisation des ressources mémoire	63
IV.1. Stockage de l'histogramme expérimental	63
IV.2. Stockage de l'histogramme Idéal	67
IV.3. Récapitulatif	70
V. Conclusion	71

Chapitre 3

Analyseur de réponses de test :	73
I. Introduction	74
II. Analyseur de réponses pour signal d'entrée triangulaire	75
II.1. Architecture niveau porte	75
II.2. Implantation physique	87
III. Analyseur de réponses pour signal d'entrée sinusoïdal	89
III.1. Validation des approximations	89
III.2. Implantation niveau porte	93
IV. Discussion	100
IV.1. Limitation	100
IV.2. Optimisation	102
IV.3. Automatisation	105
V. Conclusion	106

Chapitre 4

Générateur de stimuli	107
I. Performances requises d'un générateur de stimuli	108
II. Générateur de signaux linéaires	110
II.1. Principe	110
II.2. Générateurs de courant	111
II.3. Schéma et performance du générateur	115
III. Générateur de rampe auto-calibré	118
III.1. Principe	118
III.2. Asservissement et intégration	120
III.3. Asservissement numérique	121
III.4. Asservissement analogique	125
III.5. Implantation du générateur de rampe auto-calibré	127
III.6. Performances et discussion	130
III.7. Application au test intégré d'un CAN	132

IV. Générateur de triangle auto-calibré	137
IV.1. Générateur classique	137
IV.2. Générateur auto calibré	138
IV.3. Validations	140
V. Conclusion	144
Conclusion générale	145
Références bibliographiques :	149
Liste des figures et tableaux	161
Annexes	167

Introduction générale

Avec le fantastique développement des technologies liées aux marchés du multimédia et des télécommunications, de plus en plus de circuits, comportent des blocs analogiques intégrés sur la même puce que les blocs numériques. Actuellement, ces circuits, appelés circuits mixtes, sont le plus souvent de véritables systèmes-sur-puce (SoC, pour "Systems-on-Chip") et consistent en un assemblage de blocs fonctionnels, appelés cœurs ou IP ("Intellectual Property"), issus de bibliothèques logicielles différentes. La complexité de ces SoC et plus généralement de l'ensemble des circuits analogiques et mixtes rend leur test particulièrement critique.

Actuellement, il est admis que la complexité des nouvelles générations de circuits intégrés est telle que les procédures de test classiques ne sont plus capables de garantir entièrement les performances de ces circuits. Pour pouvoir tester de manière efficace les circuits fabriqués, il est donc nécessaire de concevoir les circuits de manière à ce qu'il soit possible de les tester après fabrication. C'est à partir de cette observation que le concept de Conception en Vue du Test (CVT) a été développé. Un grand nombre de solutions de CVT existent pour les circuits numériques et sont peu à peu standardisées et directement intégrées dans les outils de synthèse automatique. Cependant, le fait qu'il n'existe aucune méthode générique pour tester les blocs analogiques ou mixtes implique un coût extrêmement élevé pour le test de ces blocs. Ainsi le test d'un circuit mixte peut représenter jusqu'à 50% du prix de revient total d'un circuit intégré actuel.

De plus, dans la mesure où les nouvelles technologies permettent de diminuer le coût de fabrication des circuits intégrés, il est généralement admis que le coût du test, et en particulier celui du test de la partie analogique, représentera bientôt la plus grosse partie du coût total des

circuits intégrés mixtes. Dans le contexte technologique et économique actuel, il est donc crucial pour les fabricants de CI de réduire le coût du test des circuits mixtes.

Ce coût excessif est dû en grande partie à l'utilisation d'équipements sophistiqués nécessaires au test de circuits toujours plus performants. Parallèlement à cela, pour les systèmes-sur-puce, l'accès aux différents nœuds internes des blocs devient problématique et rend le test particulièrement difficile et coûteux. Dans ce contexte, une solution attractive pour réduire le coût du test consiste à intégrer directement sur la puce tout ou une partie des ressources nécessaires au test. Ainsi les structures de test intégré BIST (Built-In-Self-Test) permettent de rendre le circuit (ou les cœurs qu'il contient) auto testable ou tout au moins de réduire considérablement l'équipement externe nécessaire pour le tester.

Dans la majeure partie des applications multimédia et de télécommunication, il est nécessaire de faire communiquer le domaine réel qui est forcément analogique avec le processeur numérique. De ce fait, le Convertisseur Analogique/Numérique (CAN) est l'élément incontournable de la plupart des circuits mixtes.

Dans ce contexte, l'objectif des travaux présentés dans le présent manuscrit consiste à proposer des structures de test intégré dédiées aux Convertisseurs Analogique/Numérique. Les solutions proposées sont basées sur une technique classique de test externe couramment utilisée dans le domaine industriel : le test par histogramme. Comme toute structure BIST les structures de test intégré proposées sont constituées d'un générateur de stimuli et d'un analyseur de réponses de test.

Dans le premier chapitre, nous présentons le contexte des travaux effectués. Tout d'abord, nous rappelons les concepts de base qui permettent de définir les caractéristiques fonctionnelles d'un Convertisseur Analogique/Numérique. Nous nous intéressons ensuite aux différents types de test couramment utilisés dans l'industrie pour évaluer ces caractéristiques fonctionnelles. Enfin, nous réalisons un état de l'art des différentes techniques de test intégré pour convertisseurs A/N proposées dans la littérature.

Dans le second chapitre nous développons une architecture originale d'analyseur de réponses de test pour les différents types de signaux d'entrée (linéaire ou sinusoïdal). Les solutions proposées permettent une parfaite évaluation des paramètres fonctionnels du convertisseur tout en nécessitant une surface extrêmement réduite. Pour obtenir ces résultats, nous avons été

amenés à repenser la technique de test par histogramme qui n'est pas directement intégrable dans sa forme initiale.

Dans le troisième chapitre, nous validons les résultats obtenus au chapitre précédent en effectuant une implantation de l'analyseur de réponses de test. Ainsi, dans le cas d'un signal d'entrée triangulaire, nous effectuons un ensemble d'optimisations de l'architecture haut niveau présentée au chapitre 2. Celles-ci nous permettent de proposer une implantation physique de la structure finale pour une surface de silicium très faible. Dans le cas d'un signal d'entrée sinusoïdal, nous développons un outil logiciel permettant de simuler la procédure du test par histogramme et de comparer les résultats de mesure avec ceux obtenus en utilisant notre technique de test. Cet outil permet de valider les approximations utilisées pour simplifier les expressions d'extraction des paramètres fonctionnels du CAN.

Le quatrième chapitre est consacré à la génération des stimuli analogiques de test. Dans un premier temps nous rappelons les différentes contraintes imposées par le contexte BIST de nos travaux. Nous prenons en compte ces contraintes pour élaborer des structures de générateurs de signaux linéaires analogiques. Ainsi, dans un premier temps nous proposons un générateur de rampe. Ce générateur utilise un système particulièrement innovant d'auto-calibration qui lui permet de générer une rampe précise et insensible aux variations des paramètres technologiques tout en impliquant une surface de silicium minimale. Ensuite, nous proposons une architecture de générateur de signaux triangulaires basée sur le même principe de calibration. L'association d'un de ces générateurs avec l'analyseur de réponses de test développé au chapitre deux, nous permet de construire une structure BIST globale permettant d'intégrer les qualités du test par histogramme en impliquant une surface de silicium extrêmement réduite.

Chapitre 1 :

"État de l'Art "

Il existe actuellement une très grande variété de circuits analogiques et mixtes. Il serait difficile de donner une liste exhaustive de ces circuits et des différentes techniques de test qui leur sont associées. C'est pourquoi, dans ce chapitre d'état de l'art, nous nous limitons au domaine, pourtant très vaste de l'élément charnière des circuits mixtes : le Convertisseur Analogique/Numérique.

Dans un premier temps, nous rappellerons les concepts de base de la Conversion Analogique/Numérique. Nous décrirons ensuite les caractéristiques fonctionnelles des Convertisseurs Analogique/Numérique. Dans une troisième partie, nous présenterons les deux techniques de test industriel les plus utilisées pour évaluer ces caractéristiques. Enfin, nous présenterons les différentes solutions de test intégré pour convertisseurs A/N, proposées dans la littérature.

I. Concepts de base

Les Convertisseurs Analogique/Numérique (CAN) permettent de faire la liaison entre le domaine réel analogique et le domaine numérique. La figure 1.1 représente les deux phases de la conversion d'un signal analogique en codes numériques.

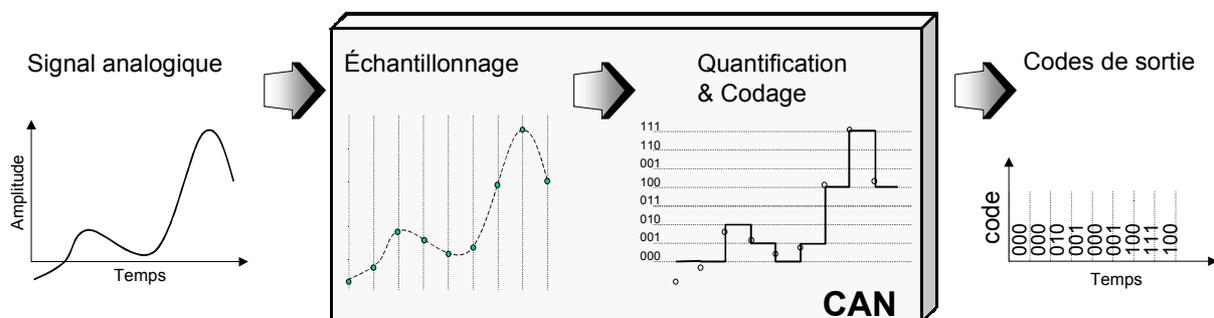


Figure 1.1 : Conversion Analogique/Numérique

La première phase consiste à échantillonner le signal analogique. L'échantillonnage correspond à une discrétisation temporelle du signal analogique.

Il existe deux techniques d'échantillonnage dans le cadre du test de circuits intégrés :

- **l'échantillonnage cohérent**, qui consiste à répartir les échantillons sur un nombre entier N_P de périodes du signal d'entrée sans avoir de répétition d'échantillons à l'intérieur de la séquence d'échantillonnage. Il impose une condition particulière entre la fréquence d'échantillonnage $F_{\text{éch}}$ et la fréquence du signal d'entrée F_{in} pour un nombre d'échantillons N_T donné:

$$\frac{F_{\text{in}}}{F_{\text{éch}}} = \frac{N_P}{N_T} \quad \text{avec } N_P \text{ et } N_T \text{ premiers entre eux.} \quad (1.1)$$

- **l'échantillonnage non cohérent**, qui consiste à répartir les échantillons de manière aléatoire sur l'ensemble du signal d'entrée. Pour satisfaire cette condition, il ne doit pas y avoir de synchronisation entre le signal d'entrée et la fréquence d'échantillonnage.

La deuxième phase de conversion consiste en une discrétisation spatiale du signal sur un nombre fini de valeurs. Cette opération appelée quantification est complétée par un codage qui permet d'associer un code numérique à chacune de ces valeurs discrètes.

II. Caractéristiques Fonctionnelles des Convertisseurs A/N

Indépendamment de la structure du CAN (approximations successives, flash, Sigma-Delta ...), un ensemble de paramètres permettent de spécifier les caractéristiques fonctionnelles du convertisseur. Il est possible de classer ces erreurs en deux catégories. On a d'une part les erreurs systématiques dues à la conversion de signaux analogiques en signaux numériques qui sont présentes sur un CAN idéal (erreur de quantification), et d'autre part les erreurs dues aux imperfections présentes sur un CAN réel.

Pour illustrer les différents paramètres des convertisseurs, nous prendrons l'exemple simple d'un CAN de 3 bits.

II.1 Convertisseur A/N parfait

a) Fonction de transfert

Rappelons que la fonction d'un CAN est de transformer un signal continu analogique en une succession de codes numériques représentatif du signal. La fonction de transfert traduit la réponse du convertisseur à une entrée analogique.

La dynamique maximale ou Pleine Echelle (PE) du convertisseur est divisée en intervalles délimités par les tensions de seuil V_T . A chaque valeur analogique à convertir est associé un code numérique de sortie qui correspond à la plus proche tension de seuil V_T inférieure à cette valeur analogique. Finalement, dans le cas le plus classique d'un codage binaire naturel sans loi de compression, la fonction de transfert d'un convertisseur est une fonction en marches d'escalier de la forme illustrée figure 1.2 :

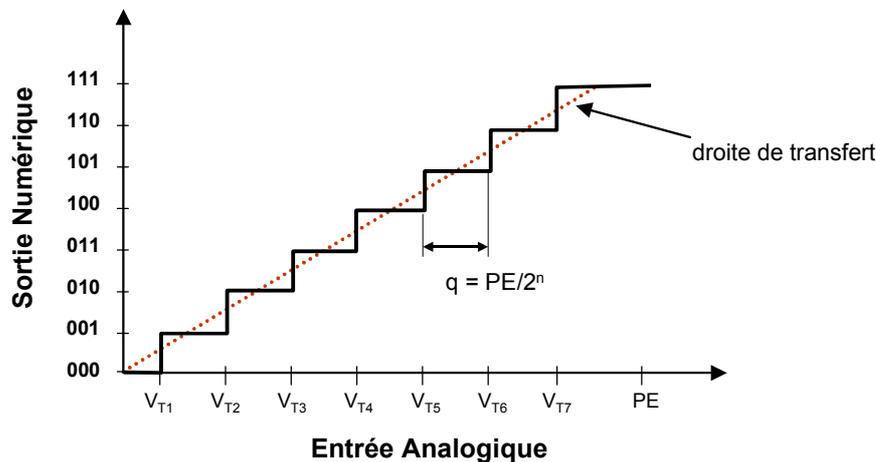


Figure 1.2 : Fonction de transfert

La largeur de chaque palier, appelée **pas de quantification** ou **quantum q** , correspond à la plus petite différence de tension analogique entre deux codes successifs ; elle est souvent exprimée en terme de LSB (Least Significant Bit). Dans le cas d'un CAN parfait sans loi de compression, le pas de quantification est constant pour tous les codes :

$$q = PE/2^n = 1 \text{ LSB} \quad \text{où } n \text{ représente le nombre de bits du convertisseur.} \quad (1.2)$$

Nous pouvons remarquer que le premier et le dernier palier ont une largeur égale respectivement à $\frac{q}{2}$ et $\frac{3.q}{2}$. Cette dissymétrie permet de limiter l'influence de l'erreur de quantification que nous définirons à la section suivante.

La valeur de chaque tension V_{Ti} est exprimée à l'aide de l'équation suivante :

$$V_{Ti} = q \times (i - 0,5) \quad (1.3)$$

En plus de la fonction en marches d'escalier, on définit une courbe de transfert passant par le centre des différentes marches. Dans le cas d'un convertisseur parfait, cette courbe de transfert est une droite.

b) Erreur de quantification

Nous pouvons observer sur la fonction de transfert qu'une plage entière de tension analogique comprise entre deux tensions de seuil successives est convertie en un code unique. Ceci constitue une perte d'information, appelée **erreur de quantification**. La figure 1.3 donne l'évolution de cette erreur, exprimée en LSB, en fonction de la tension analogique appliquée en entrée du convertisseur.

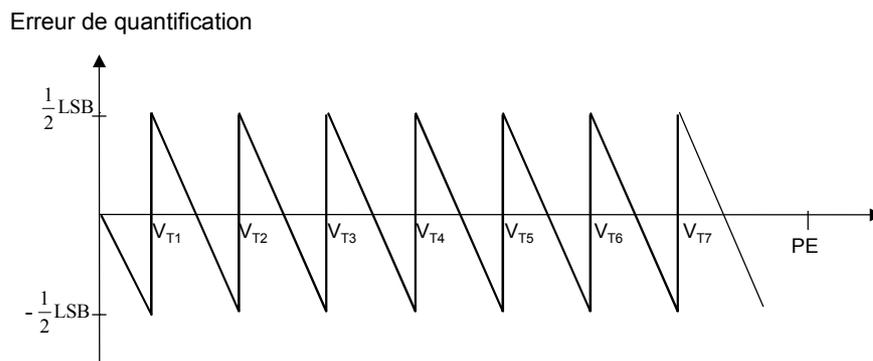


Figure 1.3 : Variation de l'erreur de quantification

A cette erreur de quantification, on associe un bruit de quantification qui est considéré comme une variable aléatoire uniformément répartie sur l'intervalle $\left[-\frac{q}{2}; \frac{q}{2}\right]$. Sa densité de probabilité F_q est donc constante sur cet intervalle (figure 1.4).

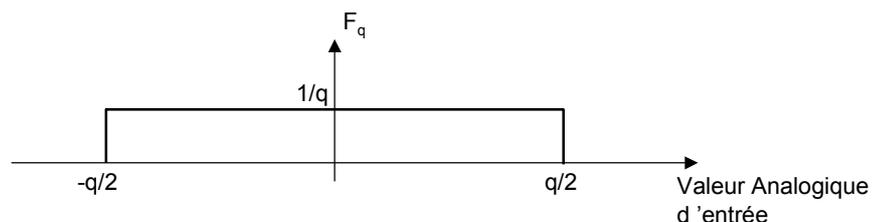


Figure 1.4 : Bruit de quantification

Nous pouvons en déduire la valeur efficace du bruit de quantification B_q :

$$B_q = \left(\int_{-\infty}^{\infty} x^2 F_q(x) dx \right)^{1/2} = \left(\frac{1}{q} \int_{-q/2}^{q/2} x^2 dx \right)^{1/2} = \frac{q}{\sqrt{12}} \quad (1.4)$$

Nous pouvons observer que cette erreur dépend directement du pas de quantification q et donc de la résolution du convertisseur.

c) Rapport signal sur bruit

Le rapport signal sur bruit représente le rapport entre la valeur efficace du signal à convertir et celle du bruit (en anglais : SNR pour Signal to Noise Ratio). Généralement, ce paramètre est défini pour un signal d'entrée sinusoïdal d'amplitude crête à crête égale à la pleine échelle du convertisseur. La valeur efficace A_{RMS} du signal d'entrée est donnée par l'expression :

$$A_{RMS} = \frac{PE}{2\sqrt{2}} = \frac{2^{n-1}q}{\sqrt{2}} \quad (1.5)$$

Dans le cas d'un CAN parfait, le bruit étant dû uniquement à la quantification du signal d'entrée, nous pouvons en déduire l'expression suivante du rapport signal sur bruit exprimé en dB :

$$SNR_{dB} = 20 \log \left(\frac{A_{RMS}}{B_q} \right) = 20 \log \left(\frac{2^{n-1}q/\sqrt{2}}{q/\sqrt{12}} \right) = 20 \log \left(\sqrt{\frac{3}{2}} \cdot 2^n \right)$$

$$SNR_{dB} = 6,02.n + 1,76 \quad (1.6)$$

Ainsi, nous pouvons observer que le rapport signal sur bruit d'un convertisseur analogique/numérique parfait dépend directement du nombre de son nombre de bit n .

II.2 Convertisseur A/N réel

Dans le cas d'un convertisseur analogique/Numérique réel, il est nécessaire de définir d'autres paramètres fonctionnels. Dans ce paragraphe, nous détaillerons les paramètres les plus couramment utilisés.

a) Erreur d'offset

Nous pouvons définir d'abord l'erreur d'offset, qui correspond, dans le cas d'un convertisseur réel, à un décalage identique de toutes les tensions de seuil. Elle est généralement exprimée en fraction de LSB.

Au niveau de la fonction de transfert, l'erreur d'offset correspond à une translation de l'ensemble de la caractéristique (figure 1.5).

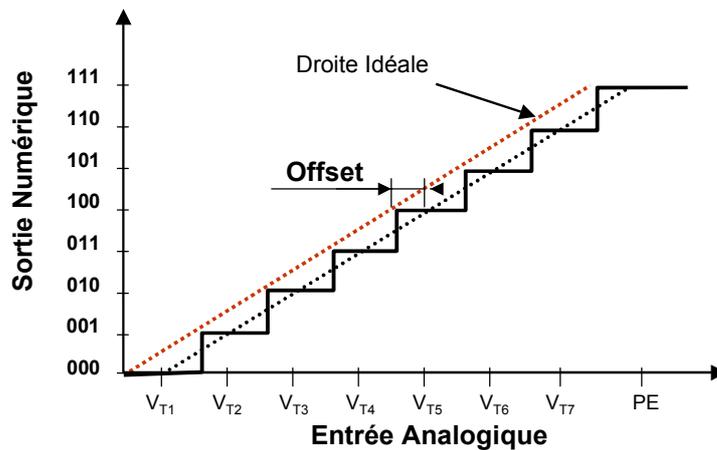


Figure 1.5 : Erreur d'offset

b) Erreur de Gain

L'erreur de gain correspond à une variation identique de la largeur des différents paliers dont l'influence sur la fonction de transfert est représentée figure 1.6. Cette erreur est équivalente à une variation de la pente de la droite de transfert idéale.

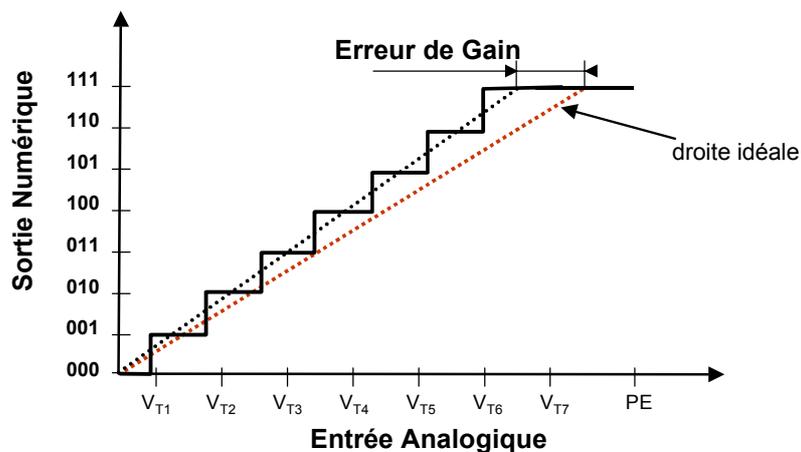


Figure 1.6 : Erreur de gain

Généralement, l'erreur de gain est définie comme la différence, au niveau du dernier code, entre la droite de transfert idéale et la droite de transfert réelle.

c) Erreur de Non-Linéarité Différentielle et Non-Linéarité Intégrale

Pour un CAN parfait, le pas de quantification q est constant pour l'ensemble des codes du convertisseur. En réalité, la largeur $q(i)$ de chaque palier i peut varier d'un code à l'autre.

La Non-linéarité Différentielle (NLD) d'un code i représente la différence, exprimée en LSB, de la largeur du palier associé au code i par rapport à la valeur idéale de 1 LSB :

$$NLD(i) = \frac{q(i) - q}{q} \quad (1.7)$$

La Non-linéarité Intégrale (NLI) d'un code i représente la variation entre la courbe réelle et la droite de transfert idéale au niveau de ce code. Elle correspond, pour chaque code i , à la somme des non-linéarités différentielles des codes précédents :

$$NLI(i) = \sum_{j=1}^i NLD(j) \quad (1.8)$$

La figure 1.7 illustre l'influence de ces deux types de non-linéarités sur la fonction de transfert du convertisseur.

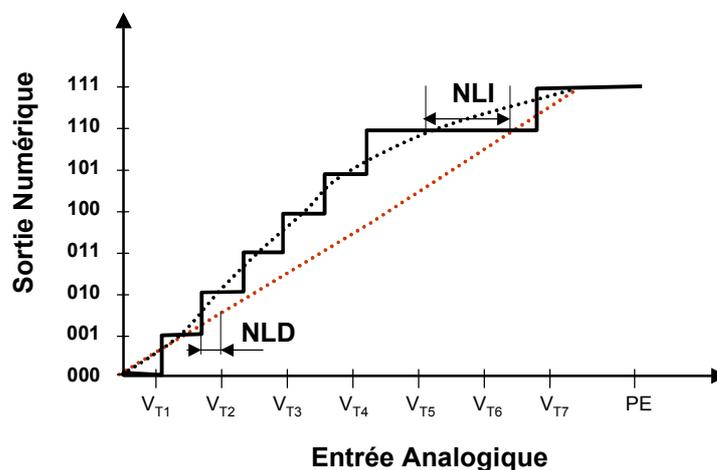


Figure 1.7 : Non-linéarité Différentielle et Intégrale

d) Erreur de code manquant

Une erreur de code manquant signifie qu'un code n'est jamais présent en sortie du convertisseur. Cette erreur est due à une erreur de non-linéarité trop importante. Nous pouvons remarquer qu'un convertisseur dont les non-linéarités différentielles sont comprises entre + 1 LSB et - 1 LSB ne peut pas comporter de code manquant.

e) Taux de distorsion harmonique

Dans le cas d'un signal d'entrée sinusoïdal pur de fréquence F_{in} , les non-linéarités du convertisseur engendrent des harmoniques de fréquence $k.F_{in}$ (k entier). Le Taux de Distorsion Harmonique (THD : Total Harmonic Distortion) permet d'évaluer l'importance de ces harmoniques.

Il représente le rapport entre la somme des amplitudes H_k des harmoniques d'ordre k et l'amplitude A_{rms} du signal d'entrée.

$$(THD)_{dB} = 20 \cdot \log \left(\frac{\sqrt{\sum_{k>1} H_k^2}}{A_{rms}} \right) \quad (1.9)$$

f) Taux de distorsion par inter-modulation

Comme pour le THD, le taux de distorsion par inter modulation (IMD : Inter-Modulation-Distortion) est dû aux non-linéarités du convertisseur.

Il nous renseigne sur la présence de fréquences parasites engendrées par les non-linéarités du convertisseur à partir des fréquences du spectre du signal d'entrée. Généralement, le taux de distorsion par inter-modulation est évalué en appliquant un signal composé de deux sinusoïdes de fréquences F_1 et F_2 et de valeurs efficaces A_1 et A_2 . Il correspond au rapport entre la somme des valeurs efficaces des raies parasites, de fréquence $k_1.F_1 + k_2.F_2$ (k_1 et k_2 entiers) et de valeur efficace $IM_{k_1k_2}$, et l'amplitude du signal d'entrée.

$$(IMD)_{dB} = 10 \cdot \log \left(\frac{\sum_{k_1, k_2} IM_{k_1, k_2}^2}{A_1^2 + A_2^2} \right) \quad (1.10)$$

g) Rapport signal sur bruit avec distorsion

Le rapport signal sur bruit avec distorsion (SINAD : Signal to Noise And Distorsion) est défini comme le rapport entre la valeur efficace du signal d'entrée et la valeur efficace du bruit.

Dans le cas d'un convertisseur réel, le bruit n'est pas seulement dû au pas de quantification mais à l'ensemble des variations des paramètres fonctionnels (non-linéarités, le jitter...).

Le SINAD est donné par l'expression suivante, où B_{RMS} représente la valeur efficace du bruit (bruit de quantification compris) :

$$\text{SINAD}_{dB} = 20 \cdot \log \left(\frac{A_{RMS}}{B_{RMS}} \right) \quad (1.12)$$

Il permet de calculer le nombre de bits effectifs n_{eff} du convertisseur donné par l'expression (1.13), établie en considérant que le bruit total (quantification intrinsèque + distorsions) d'un convertisseur A/N réel est équivalent à une erreur de quantification.

$$n_{eff} = \frac{\text{SINAD}_{dB} - 1,76}{6,02} \quad (1.13)$$

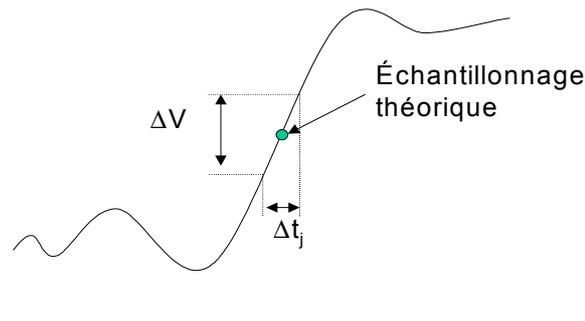
h) Dynamique de codage

La dynamique de codage (SFDR : Spurious Free Dynamic Range) correspond à la différence entre l'amplitude A_{RMS} de la raie fondamentale et l'amplitude B_i de la plus forte raie parasite (harmoniques comprises). Le SFDR, exprimé en dB, est défini par l'expression suivante :

$$(\text{SFDR})_{dB} = 20 \cdot \log \left(\frac{\max(B_i)}{A_{RMS}} \right) \quad (1.14)$$

i) Incertitude au point d'ouverture (jitter)

L'échantillonnage de l'entrée analogique peut ne pas avoir lieu exactement à l'instant désiré. Cette incertitude Δt_j sur l'instant d'échantillonnage, appelée couramment jitter, produit une erreur de conversion ΔV proportionnelle à la pente du signal comme illustré à la figure 1.8.



$$\Delta V = \Delta t_j \cdot \frac{dV}{dt} \quad (1.11)$$

Figure 1.8 : jitter

j) Bande passante

La bande passante d'un convertisseur représente le domaine de fréquence pour lequel le convertisseur a un gain qui reste constant. Classiquement, si on considère le signal analogique reconstitué à partir des codes de sortie du convertisseur, la bande passante du convertisseur représente la plage de fréquence où l'atténuation du signal de sortie due au convertisseur est inférieure à 3dB.

Pour illustrer l'ensemble de ces définitions, nous proposons en annexe 1 quelques exemples de spécifications fonctionnelles de CAN réels tirées de catalogue de composants

III. Test Industriel des CAN

D'une façon générale, il existe deux types d'approche pour tester un circuit intégré : l'approche structurelle et l'approche fonctionnelle.

- a) L'approche structurelle est basée sur l'étude de la présence de défauts au niveau de la structure du circuit. Dans un premier temps, elle consiste à établir des modèles de fautes qui correspondent le mieux aux défauts réels possibles du circuit. Dans un deuxième temps, elle nécessite la recherche de stimuli de test permettant la meilleure détection de fautes pour les modèles considérés. Généralement, l'efficacité des techniques de test structurel est exprimée en terme de taux de couverture de fautes.

Dans le cas des circuits numériques, le modèle utilisé est généralement le modèle de collage à 1 ou à 0 de nœuds du circuit, ce qui correspond à des modèles de courts-circuits avec les alimentations du circuit. Même si ces modèles ne représentent pas toujours les défauts physiques réels, ils permettent d'obtenir de très bons taux de détection de circuits

défectueux. Malheureusement, dans le cas de circuits analogiques, il n'existe pas de modèle de fautes qui soit à la fois aussi simple et aussi performant.

- b) L'approche fonctionnelle consiste à étudier le circuit à un niveau d'abstraction supérieur, indépendant de la structure du circuit. Généralement, ce type d'approche est basé sur l'évaluation des différents paramètres qui définissent la fonctionnalité du circuit. L'approche fonctionnelle est très fiable, mais elle nécessite un temps de test qui peut devenir prohibitif pour des circuits complexes.

Le but de cette section n'est pas de faire l'état de l'art de toutes les techniques de test des convertisseurs A/N, mais de détailler les deux types de test les plus couramment utilisés dans l'industrie. A noter que ces deux techniques sont basées sur une approche fonctionnelle. Nous prendrons souvent comme référence dans cette section l'ouvrage de M. Mahoney intitulé "DSP-based Testing" [Mah84].

III.1 Test par Histogramme

a) Principe

Le test par histogramme est basé sur une analyse statistique de la probabilité d'apparition des codes de sortie du convertisseur. La figure 1.9 décrit la mise en œuvre des différentes étapes de cette technique de test.

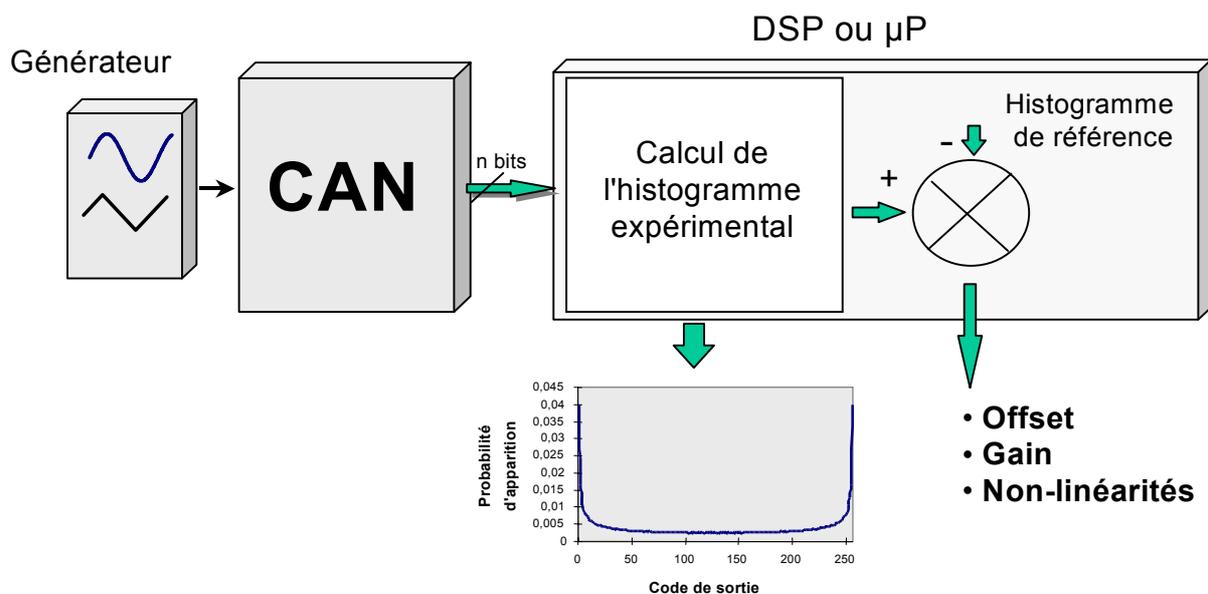


Figure 1.9 : Structure du test par histogramme

La première étape consiste à construire l'histogramme expérimental associé au convertisseur sous test. Cet histogramme représente le nombre de fois $H^{\text{exp}}(i)$ que les codes i sont présents en sortie du convertisseur pour un nombre d'échantillons et un signal d'entrée donnés. En général, le signal d'entrée est soit un signal sinusoïdal soit un signal linéaire (rampe ou triangle), et sa fréquence est choisie non cohérente avec la fréquence d'échantillonnage.

Pour avoir une bonne validité statistique, il est nécessaire d'acquérir un très grand nombre d'échantillons. Si le nombre N_T d'échantillons tend vers l'infini, alors $\frac{H^{\text{exp}}(i)}{N_T}$ converge vers la densité de probabilité du signal d'entrée.

La deuxième étape consiste à comparer cet histogramme expérimental avec l'histogramme de référence obtenu avec un convertisseur parfait. Cette comparaison permet d'extraire les paramètres fonctionnels du convertisseur sous test. Le test par histogramme permet ainsi de déterminer la valeur de **l'erreur d'offset**, de **l'erreur de gain**, des **non-linéarités différentielles**, des **non-linéarités intégrales** et permet aussi de détecter **les codes manquants**.

Avant de détailler le calcul de l'histogramme de référence pour les deux types de signaux d'entrée (linéaire et sinusoïdal), nous devons faire deux remarques importantes.

La première remarque concerne l'amplitude du signal d'entrée. Pour que le test par histogramme soit viable, il faut que tous les codes soient traités. Or, comme il est difficile de maîtriser la valeur exacte de l'amplitude du signal d'entrée, les codes extrêmes risquent de ne pas être présents en sortie du convertisseur. Généralement, pour palier ce problème, l'amplitude crête à crête du signal d'entrée est choisie supérieure à la pleine échelle du convertisseur $A_{\text{in}} > PE$.

La deuxième remarque concerne la valeur moyenne du signal. Précédemment (§ II.1), nous avons présenté la fonction de transfert d'un convertisseur. Nous avons vu que cette fonction de transfert n'est pas symétrique puisque le premier palier a une largeur égale à 0,5 LSB et le dernier palier a une valeur égale à 1,5 LSB. Pour le test par histogramme, il est préférable de symétriser cette fonction de transfert. Pour cela, il faut que la valeur moyenne du signal d'entrée soit égale à $\frac{PE - q}{2}$.

Par la suite, nous nous placerons toujours dans le cas où l'amplitude du signal d'entrée est supérieure à la pleine échelle du convertisseur et où sa valeur moyenne est égale à $\frac{PE - q}{2}$.

b) Calcul de l'histogramme de référence

➤ *Histogramme de référence pour un signal d'entrée linéaire*

Soit un signal aléatoire de la forme :

$$Y = k \times \frac{A_{in}}{2} + V_M \quad (1.15)$$

où k représente la variable aléatoire uniformément répartie sur $[-1 ; 1]$, A_{in} l'amplitude crête à crête et V_M la valeur moyenne du signal.

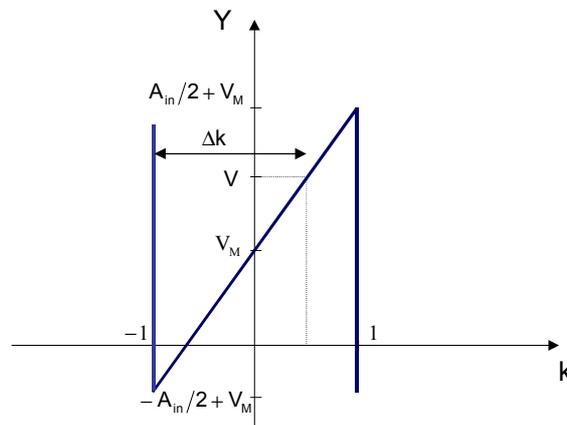


Figure 1.10 : Signal linéaire aléatoire

D'après la figure 1.10, la probabilité pour que la valeur x d'un échantillon soit inférieure à une valeur V est donnée par la relation suivante :

$$F(V) = P(x < V) = \frac{\Delta k}{2} = \frac{V - V_M}{A_{in}} + \frac{1}{2} \quad (1.16)$$

La densité de probabilité $p(V)$ est égale à la dérivée de $F(V)$: $p(V) = \frac{dF(V)}{dV} = \frac{1}{A_{in}}$. Elle est proportionnelle à l'inverse de la pente du signal. Cette pente étant constante, la densité de probabilité est constante.

La valeur de $p(V)$ correspond à la probabilité d'apparition d'une tension V pour un signal d'entrée linéaire. Pour calculer la probabilité d'apparition des codes de sortie d'un CAN, ce ne sont plus des valeurs discrètes de tension qu'il faut prendre en compte mais tous les domaines

de tension correspondants à chaque code. Pour obtenir la probabilité d'apparition du code i dont la tension analogique est située entre les tensions de seuil V_{Ti-1} et V_{Ti} , il faut intégrer la relation précédente entre ces deux valeurs de tension. Ceci équivaut à faire la différence entre la probabilité que le signal soit inférieur à V_{Ti} et la probabilité que le signal soit inférieur à V_{Ti-1} .

$$p(i) = P(V < V_{Ti}) - P(V < V_{Ti-1}) = \frac{V_{Ti} - V_{Ti-1}}{A_{in}} = \frac{PE}{2^n \cdot A_{in}} \quad (1.17)$$

Nous pouvons en déduire l'expression (équation 1.18) de la fréquence d'apparition $H^{réf}(i)$.

$$H^{réf}(i) = H_{idéale} = N_T \times \frac{PE}{2^n \cdot A_{in}} \quad \forall i \in [2 ; 2^n - 1] \quad (1.18)$$

La fréquence d'apparition $H^{réf}(i)$ est constante pour tous les codes i , sauf pour les deux codes extrêmes (codes 1 et 2^n). En effet, comme l'amplitude du signal d'entrée est plus grande que la pleine échelle du CAN, ces codes seront présents plus souvent que les autres en sortie du convertisseur (équation 1.19).

$$H^{réf}(1) = H^{réf}(2^n) = H_{extrême} = \left(\frac{2^{n-1} (A_{in} - PE)}{PE} + 1 \right) \times H_{idéale} = N_T \times \left(\frac{2^{n-1} (A_{in} - PE) + PE}{2^n A_{in}} \right) \quad (1.19)$$

La figure 1.11 représente l'histogramme obtenu dans le cas d'un signal linéaire d'amplitude $A_{in} > PE$ pour un convertisseur de 5 bits :

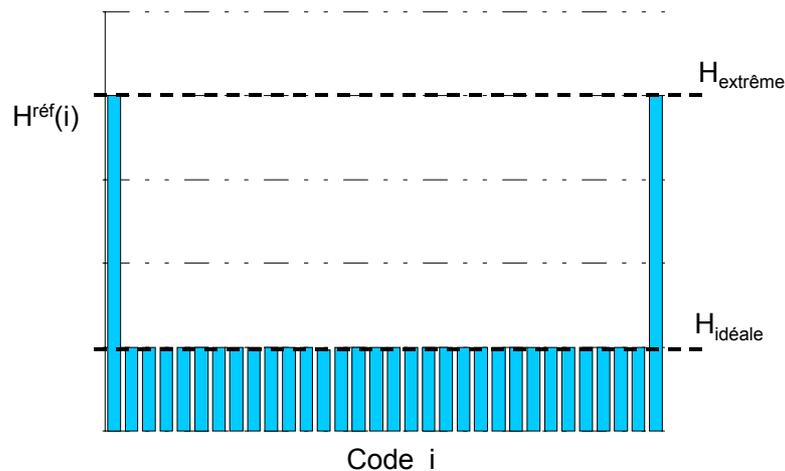


Figure 1.11 : Histogramme de référence pour un signal linéaire

➤ *Histogramme de référence pour un signal d'entrée sinusoïdal*

Soit un signal aléatoire de la forme :

$$Y = \frac{A_{in}}{2} \sin(\varphi) + V_M \quad (1.20)$$

où φ représente une variable aléatoire uniformément répartie sur $[-\pi ; \pi]$, A_{in} l'amplitude crête à crête et V_M la valeur moyenne du signal.

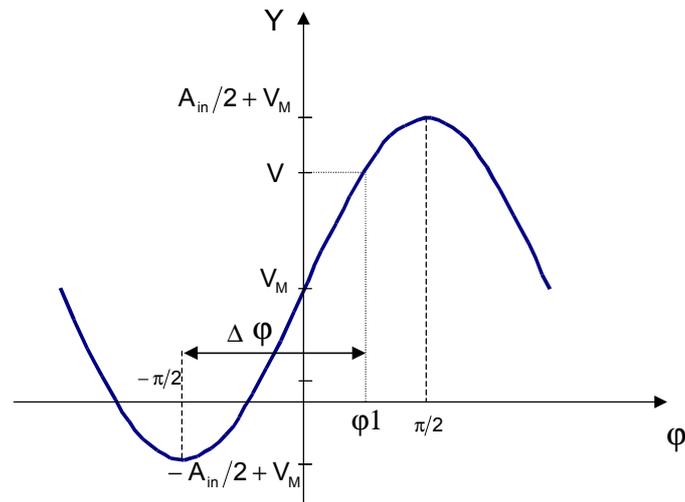


Figure 1.12 : Signal sinusoïdal aléatoire

D'après la figure 1.12, la probabilité pour que la valeur x d'un échantillon soit inférieure à une valeur V est donnée par la relation suivante :

$$F(V) = \frac{\Delta\varphi}{\pi} = \frac{\varphi_1 - \left(-\frac{\pi}{2}\right)}{\pi} = \frac{\text{Arcsin}\left(\frac{2 \cdot (V - V_M)}{A_{in}}\right) + \frac{\pi}{2}}{\pi} \quad (1.21)$$

La densité de probabilité correspondante, exprimée par l'équation 1.22 et illustrée par la figure 1.13, est égale à la probabilité d'apparition d'une valeur V pour un signal sinusoïdal.

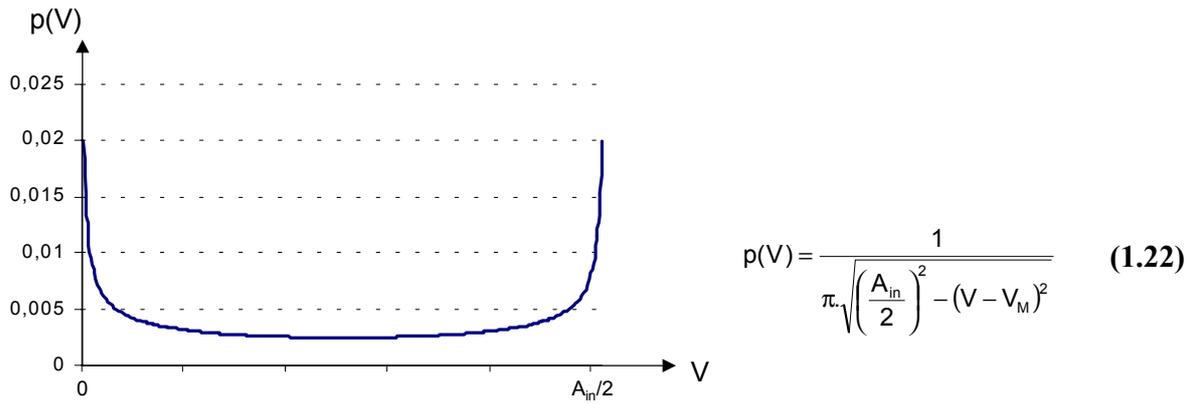


Figure 1.13 : Densité de probabilité d'une sinusoïde

Par conséquent, la probabilité d'apparition d'un code i en sortie du convertisseur est donnée par l'expression suivante :

$$p(i) = P(V < V_{T_i}) - P(V < V_{T_{i-1}}) = \frac{1}{\pi} \left[\arcsin\left(\frac{V_{T_i} - V_M}{A_{in}/2}\right) - \arcsin\left(\frac{V_{T_{i-1}} - V_M}{A_{in}/2}\right) \right] \quad (1.23)$$

Nous pouvons alors en déduire la fréquence d'apparition $H^{réf}(i)$ du code i exprimée par les équations 1.24 et 1.25.

$$H^{réf}(1) = H^{réf}(2^n) = \frac{N_T}{\pi} \left(\arcsin\left[\left(\frac{1}{2^{n-1}} - 1\right) \cdot \frac{PE}{A_{in}}\right] - \frac{\pi}{2} \right) \quad (1.24)$$

et

$$H^{réf}(i) = \frac{N_T}{\pi} \left(\arcsin\left[\left(\frac{2i - 2^n}{2^n}\right) \cdot \frac{PE}{A_{in}}\right] - \arcsin\left[\left(\frac{2i - 2^n - 2}{2^n}\right) \cdot \frac{PE}{A_{in}}\right] \right) \quad \forall i \in [2 ; 2^n - 1] \quad (1.25)$$

Pour un CAN de 5 bits et une amplitude crête à crête $A_{in} = PE + 2 \text{ LSB}$, nous obtenons l'histogramme de référence suivant :

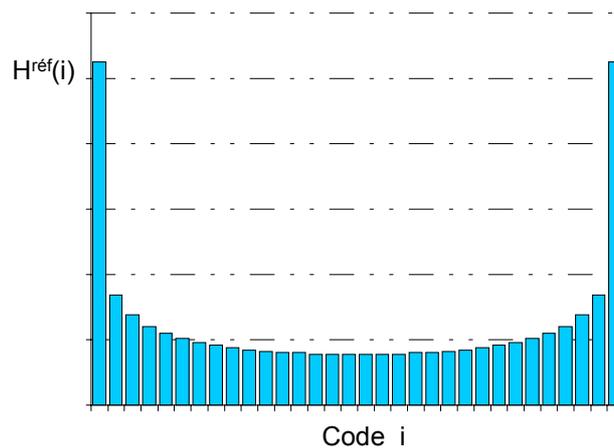


Figure 1.14 : Histogramme de référence pour un signal d'entrée sinusoïdal

c) Histogramme verrouillé

La technique de test par histogramme présentée précédemment permet d'évaluer des paramètres fonctionnels fondamentaux des convertisseurs (offset, gain...). En effectuant quelques modifications sur cette technique de test, il est possible de déterminer d'autres paramètres intéressants.

Une des variantes du test par histogramme est le test par histogramme verrouillé [Ben93]. Il permet de déterminer l'incertitude au point d'ouverture (jitter) du convertisseur. Cette technique de test consiste à appliquer un signal d'entrée de fréquence égale à la fréquence d'échantillonnage du convertisseur et à construire l'histogramme à partir des codes de sortie du convertisseur.

En théorie, le code de sortie devrait être toujours le même puisque la valeur du code de sortie devrait être unique puisque le point d'échantillonnage devrait correspondre, à chaque période, à la même tension analogique. En pratique, le jitter induit des variations sur la tension échantillonnée que doit convertir le CAN et certains codes adjacents peuvent être émis. La figure 1.15 représente un exemple de distribution de codes (histogramme verrouillé) obtenue en utilisant cette technique.

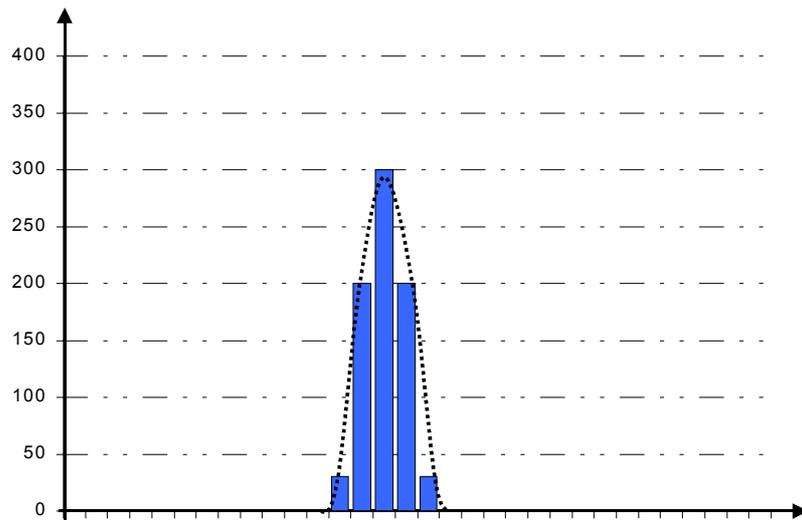


Figure 1.15 : Histogramme verrouillé

Généralement, comme le jitter (Δt_j) ne dépend que de la fréquence d'échantillonnage, cette mesure est effectuée sur un seul code. Le code considéré est le code le plus sensible à l'erreur de jitter sachant que l'erreur de jitter est définie par l'équation (1.11) rappelée ci-dessous :

$$\Delta V = \Delta t_j \cdot \frac{dV}{dt} \quad (1.11)$$

l'erreur de tension sera d'autant plus importante que la pente du signal est importante. Par conséquent, dans le cas d'un signal sinusoïdal, le code le plus sensible sera le code médian (code = $\frac{2^n}{2}$).

L'histogramme obtenu reflète la loi de distribution du jitter. Sachant que le jitter suit habituellement une loi gaussienne, on peut déterminer sa valeur en analysant la loi de probabilité suivie par l'histogramme.

III.2 Analyse spectrale

a) Principe

L'analyse spectrale consiste à appliquer un signal sinusoïdal à l'entrée du CAN sous test et à analyser sa sortie dans le domaine fréquentiel. Le spectre est obtenu en réalisant une Transformée de Fourier Discrète (TFD).

En pratique, comme les calculateurs ont une mémoire limitée, l'analyse fréquentielle ne peut s'effectuer que sur une durée T finie. Par conséquent, dans le domaine temporel, l'analyse

fréquentielle implique d'effectuer un échantillonnage et d'appliquer une fenêtre temporelle de longueur T sur les échantillons obtenus. La figure 1.16 illustre l'influence de ces deux opérations sur le spectre de sortie.

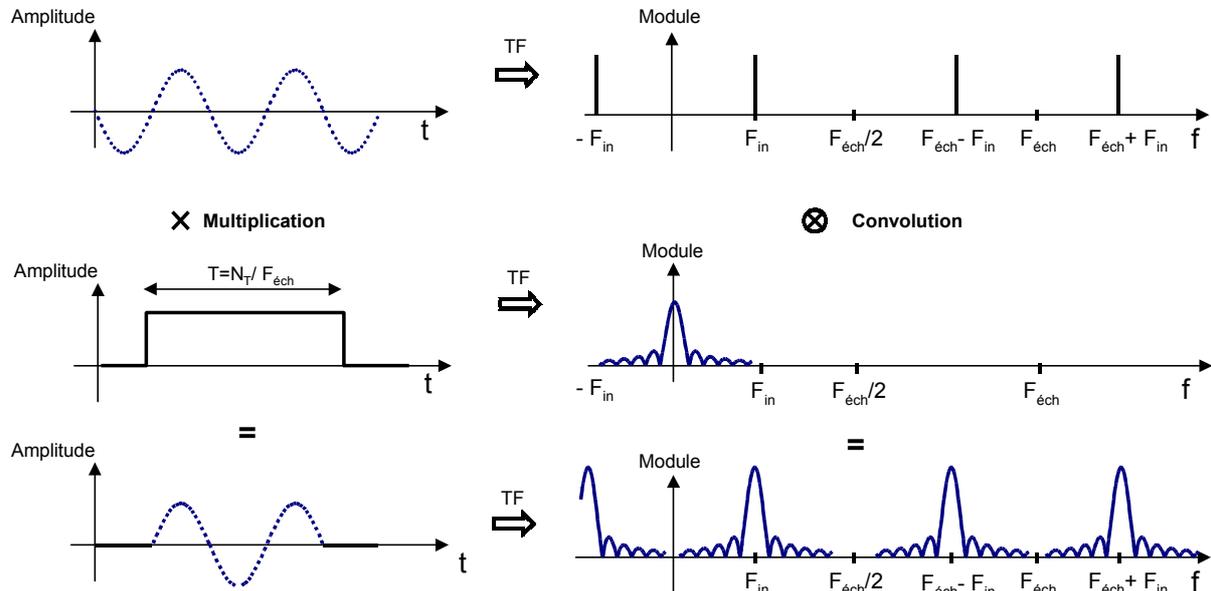


Figure 1.16 : Signal sinusoïdal aléatoire

D'une part, l'échantillonnage temporel, de fréquence $F_{\text{éch}}$, périodise le spectre (théorème de l'échantillonnage). D'autre part, le produit, dans le domaine temporel du signal rectangulaire (correspondant à la fenêtre temporelle) et du signal sinusoïdal, revient à un produit de convolution de leur transformée de Fourier. Ces différentes opérations peuvent être responsables d'une perte d'information au niveau du spectre.

En effet, il peut y avoir d'une part repliement du spectre dans la bande utile si le signal d'entrée ne respecte pas la condition de Shannon : $F_{\text{in}} < F_{\text{éch}}/2$. D'autre part, la transformée de Fourier de la fenêtre temporelle est un sinus cardinal dont les lobes secondaires peuvent introduire des raies parasites pouvant fausser l'analyse du spectre. Pour assurer la condition de Shannon, un filtre anti-repliement (filtre passe-bas) est placé à l'entrée du convertisseur. L'erreur due aux lobes secondaires du sinus cardinal est appelée effet de "leakage". Celui-ci peut être totalement supprimé pour un signal d'entrée de fréquence fixe et connue en imposant une relation particulière entre la taille de la fenêtre temporelle et la fréquence du signal d'entrée.

En effet, comme l'analyse spectrale est effectuée par un ordinateur numérique, le spectre vu à travers le ordinateur n'est plus continu mais discrétisé avec un pas :

$$\Delta F_{\text{discrétisation}} = \frac{F_{\text{éch}}}{N_T} \quad (1.26)$$

où N_T représente le nombre total d'échantillons traités.

Or, comme les lobes secondaires du sinus cardinal sont espacés de $\Delta F = \frac{1}{T} = \frac{F_{\text{éch}}}{N_T} = \Delta F_{\text{discrétisation}}$, il suffit que la fréquence F_{in} du signal d'entrée, qui représente la localisation du lobe principal, soit égale à k fois (avec k entier) le pas fréquentiel de discrétisation $\Delta F_{\text{discrétisation}}$. Dans le domaine temporel, cette condition est équivalente au fait d'utiliser une fenêtre temporelle qui englobe un nombre entier de période N_p du signal d'entrée :

$$F_{\text{in}} = N_p \frac{F_{\text{éch}}}{N_T} \Leftrightarrow T = \frac{N_p}{F_{\text{in}}} \quad (1.27)$$

De plus, l'échantillonnage doit être cohérent pour éviter une perte d'information due à des répétitions à l'intérieur de la séquence d'échantillonnage. Cette condition restrictive impose, en outre, que N_T et N_p soient premiers entre eux.

Dans le cas où la première condition ne pourrait pas être assurée, il est possible d'utiliser des fenêtres temporelles autres que rectangulaires (Hamming, Hanning...) qui modifient la valeur des échantillons dans le domaine temporel pour diminuer les effets de "leakage" au niveau spectral, et donc permettent d'améliorer l'analyse du spectre.

b) Transformée de Fourier Rapide

L'expression de la transformée discrète d'un signal $x(t)$ pour un échantillonnage à la fréquence $F_{\text{éch}}$ est donnée par l'expression 1.28 suivante :

$$X(n_{\text{éch}} F_{\text{éch}}) = \sum_{k=0}^{N_T-1} x(k \cdot T_{\text{éch}}) \times e^{-j2\pi \frac{n_{\text{éch}} k}{N_T}} \quad (1.28)$$

Où $T_{\text{éch}}$, $n_{\text{éch}}$ et N_T sont respectivement la période d'échantillonnage $T_{\text{éch}} = \frac{1}{F_{\text{éch}}}$, le numéro d'échantillon et le nombre total d'échantillons traités.

Le calcul d'une valeur $X(n_{\text{éch}}, F_{\text{éch}})$ de la transformée de Fourier discrète nécessite (N_T-1) additions et N_T multiplications complexes. Par conséquent, dans le cas classique d'un calcul de N_T valeurs de TFD, il est nécessaire d'effectuer $N_T \times (N_T-1)$ additions et N_T^2 multiplications complexes. Ce nombre important d'opérations constitue un handicap pour une analyse en temps réel.

Etant donné que la durée d'une addition complexe est négligeable devant la durée d'une multiplication complexe, le temps de calcul dépend directement du nombre de multiplications complexes à effectuer. Des algorithmes, appelés algorithmes de Transformée de Fourier Rapides TFR (en anglais : FFT pour Fast Fourier Transform), permettent de réduire ce nombre de multiplications. Le plus utilisé, l'algorithme de Cooley-Tukey, nécessite $\frac{N_T}{2} \log_2(N_T)$ multiplications au lieu des N_T^2 multiplications dans le cas d'un TFD classique.

Elle permet donc de réduire considérablement le temps de calcul, à la condition restrictive d'effectuer l'analyse sur un nombre d'échantillons égal à une puissance de 2. Par exemple, pour traiter 2048 échantillons (2^{11}), il faut 300 fois moins de temps avec une TFR qu'avec la TFD classique.

Indépendamment de la technique de calcul du spectre, il existe deux techniques d'analyse spectrale :

- technique "single tone",
- technique "dual tone".

c) Technique "single tone"

La technique "single tone" consiste à appliquer une sinusoïde pure en entrée du convertisseur pour effectuer l'analyse spectrale. La figure 1.17 donne un exemple de spectre obtenu pour un signal d'entrée sinusoïdal pur.

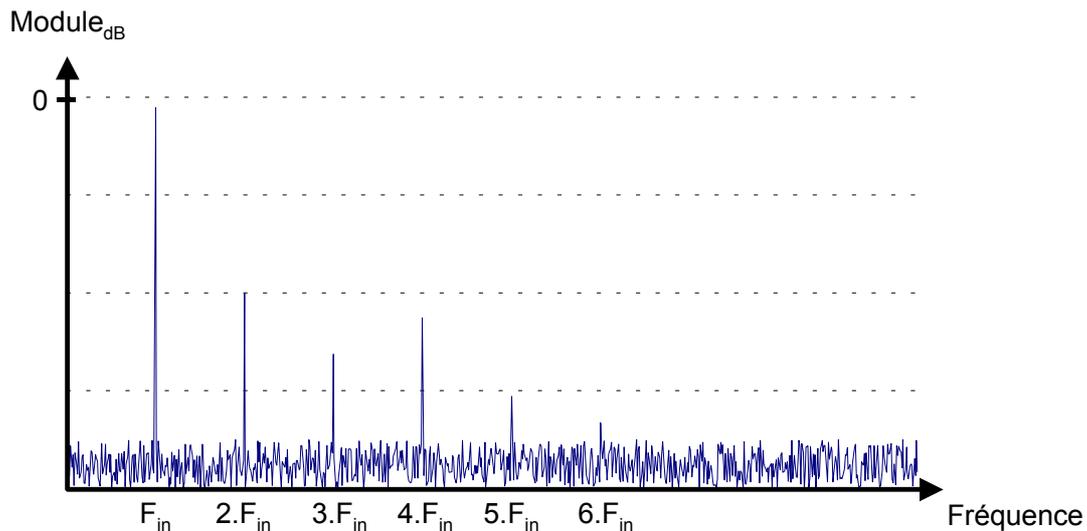


Figure 1.17 : Analyse spectrale "single Tone"

Nous pouvons observer le fondamental du signal d'entrée de fréquence F_{in} ainsi que ses harmoniques de fréquence $k.F_{in}$ (k entier) et le bruit créé par les non-linéarités du convertisseur. Avec cette technique de test fréquentielle, il est possible de déterminer le rapport signal sur bruit avec distorsion **SINAD**, le taux de distorsion harmonique **THD**, la dynamique de codage **SFDR** et le **jitter** du convertisseur testé.

d) Technique "dual tone"

La technique "dual tone" utilise comme signal d'entrée la somme de deux signaux sinusoïdaux de fréquence F_1 et F_2 n'ayant pas de relation harmonique entre eux ($F_1 \neq k.F_2$ avec k entier). Cette technique permet de mettre en évidence tous les phénomènes d'intermodulation induits par la conversion de ce signal composite. La figure 1.18 représente un exemple de spectre obtenu avec la technique "dual tone".

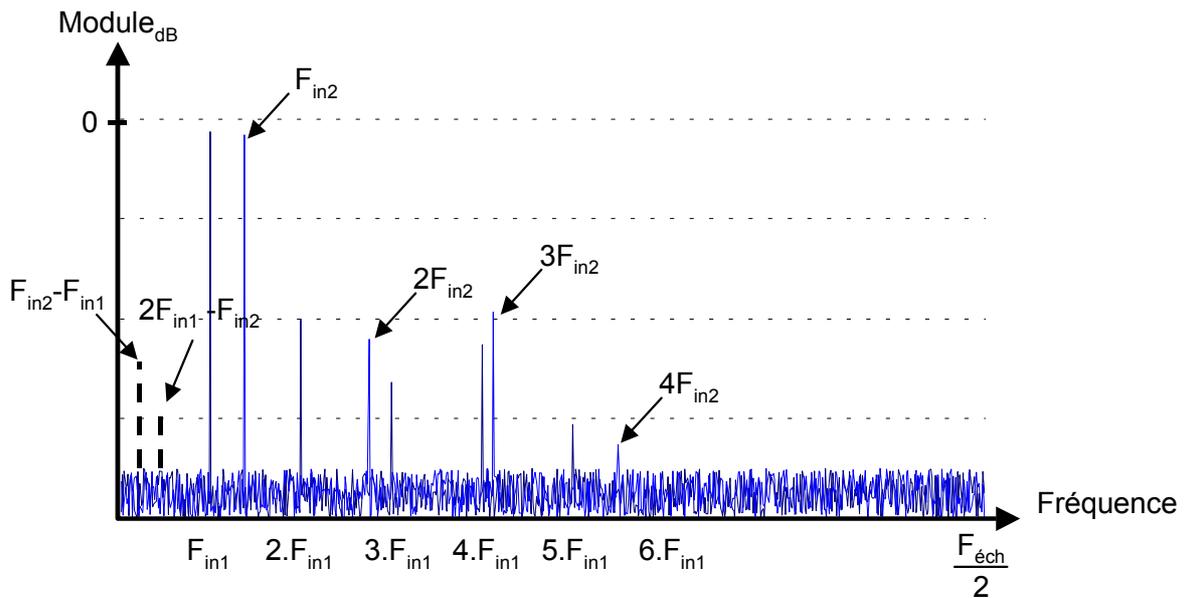


Figure 1.18 : Analyse spectrale "dual Tone"

Nous pouvons observer la présence de raies aux fréquences $i.F_1 + j.F_2$ (i et j entiers) qui s'ajoutent aux harmoniques des composantes du signal d'entrée aux fréquences $k_1.F_1$ et $k_2.F_2$ (k_1 et k_2 entier) : ce sont les raies d'intermodulation.

Avec cette technique de test "dual tone", il est donc possible de déterminer le taux de distorsion par intermodulation **IMD**.

IV. Test Intégré des CAN

Les deux techniques présentées précédemment (§ III) sont des techniques de test externe nécessitant l'utilisation d'équipements spécifiques. Or, l'amélioration des performances des circuits intégrés mixtes exige l'utilisation d'équipements de test de plus en plus sophistiqués. Le coût très élevé de ces équipements de test se répercute sur le coût global du circuit. Pour réduire ce coût, il est possible d'intégrer sur la puce tout ou partie des fonctions du testeur. Ainsi, les structures de test intégré ou BIST (Built In Self Test) obtenues sont constituées d'un ou des deux éléments suivants:

- un générateur de stimuli de test,
- un analyseur des réponses du circuit sous test.

Dans cette section, nous présentons brièvement un éventail des différentes techniques de test intégré (BIST: Built in Self Test) pour CAN proposées dans la littérature. Selon les cas, ces

structures BIST sont conçues soit dans une approche fonctionnelle soit dans une approche structurelle. Comme le paramètre le plus important dans une structure BIST est le rapport entre la surface de la structure et la surface de l'élément à tester, nous avons classé ces structures en deux catégories : d'une part, les structures nécessitant la présence de ressources matérielles particulières dans le circuit original, d'autre part, les structures utilisables sur n'importe quel type de circuit intégré.

IV.1 Test intégré avec pré-requis sur les ressources matérielles initiales

a) Structures BIST nécessitant un Convertisseur Numérique Analogique

Les deux techniques présentées dans ce paragraphe nécessitent l'utilisation d'un Convertisseur Numérique Analogique (CNA) pour le test du Convertisseur Analogique/Numérique. La surface d'un CNA étant du même ordre de grandeur que la surface d'un CAN, ce type de structure BIST n'est viable que si le CNA utilisé est déjà présent sur la puce.

Les techniques de test intégré fonctionnel proposées par Frish et Almy [Fri97][Fri98], et par Toner et Roberts [Ton92] sont toutes les deux basées sur une étude de l'histogramme construit à partir des codes de sortie du convertisseur sous test. La procédure de test est divisée en deux étapes successives.

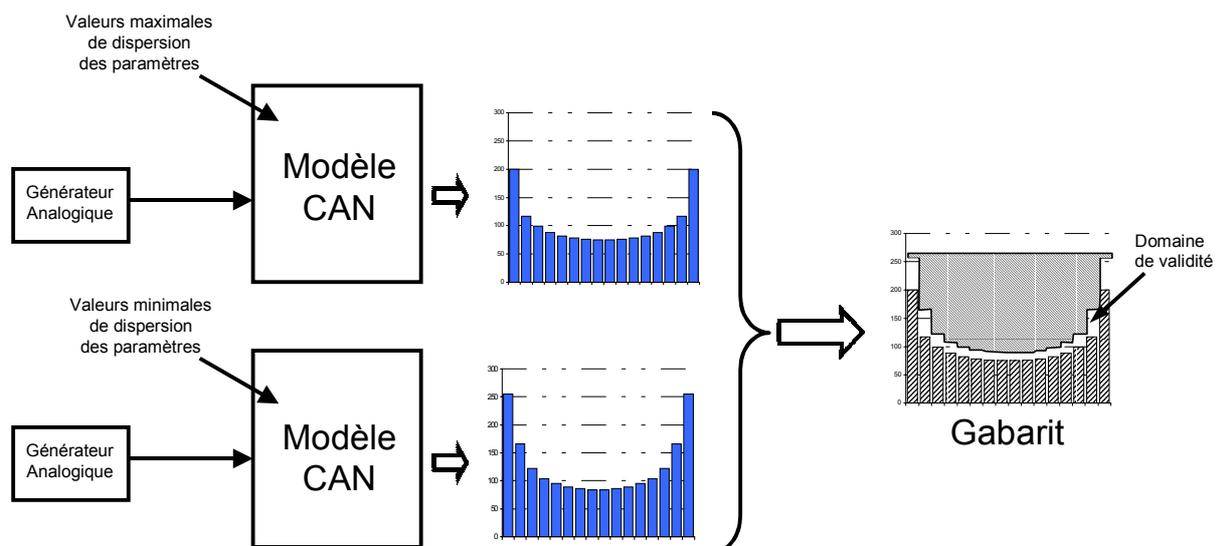


Figure 1.19 : Construction du gabarit

La première étape, représentée figure 1.19, consiste à construire un gabarit pour l'histogramme, qui tient compte des dispersions permises sur les paramètres fonctionnels à

tester. Pour obtenir ce gabarit, il est nécessaire d'effectuer un très grand nombre de simulations avec un modèle de CAN dont les caractéristiques prennent en compte les dispersions tolérées sur les paramètres à tester. Cette étape est très importante puisqu'elle influe directement sur la précision de l'évaluation des paramètres du convertisseur. Dans le cas de la technique HABIST proposée par Frish et Almy, le gabarit est construit dans le but d'évaluer l'erreur d'offset, l'erreur de gain et les non-linéarités. Dans le cas de la technique proposée par Toner et Roberts, le gabarit est construit pour évaluer l'erreur de gain et la dynamique de codage (SFDR).

La deuxième étape consiste à construire l'histogramme expérimental pour le convertisseur testé et à le comparer au gabarit élaboré durant la première étape de la technique de test. Finalement, les convertisseurs sont déclarés défectueux si l'histogramme qui leur est associé ne rentre pas dans le gabarit.

b) Structures BIST nécessitant un CNA et un DSP

Les deux structures BIST présentées dans ce paragraphe nécessitent, en plus d'un CNA, la présence d'un module de traitement numérique complexe sur la puce initiale (DSP ou Microprocesseur). Ce bloc numérique, incluant les éléments numériques des CNA et CAN, est testé à l'aide des techniques classiques de test intégré numérique.

➤ ***HABIST***

La structure de test intégré HBIST proposée par Ohletz et Dam [Ohl91][Dam95] permet d'effectuer entièrement le test du convertisseur A/N dans le domaine numérique à l'aide de l'unité de traitement numérique et du CNA. Cette technique de test est basée sur une approche structurelle.

Après le test des blocs numériques du circuit, les stimuli analogiques de test sont créés par la combinaison d'un registre, configuré en générateur de séquence pseudo-aléatoire, et d'un CNA, qui transforme cette séquence numérique en séquence analogique. Parallèlement, le module de traitement numérique analyse la réponse du convertisseur à cette séquence afin de détecter les fautes structurelles du CAN.

➤ **MADBIST**

La structure de test intégré MADBIST proposée par Toner et Roberts [Ton93][Ton95][Ton96] permet d'effectuer un test fonctionnel des circuits mixtes comportant un CNA **Sigma-Delta** ($\Sigma-\Delta$). Ce type de convertisseurs ne comporte qu'une petite partie analogique : le filtre de lissage de sortie.

Compte tenu des pré-requis en ressources matérielles déjà importants, les éléments à rajouter au circuit initial sont relativement peu nombreux. Ils sont constitués d'un générateur de signal sinusoïdal entièrement numérique composé d'additionneurs et de registres, et d'un multiplexeur analogique en entrée du CAN. La structure finale est présentée figure 1.20.

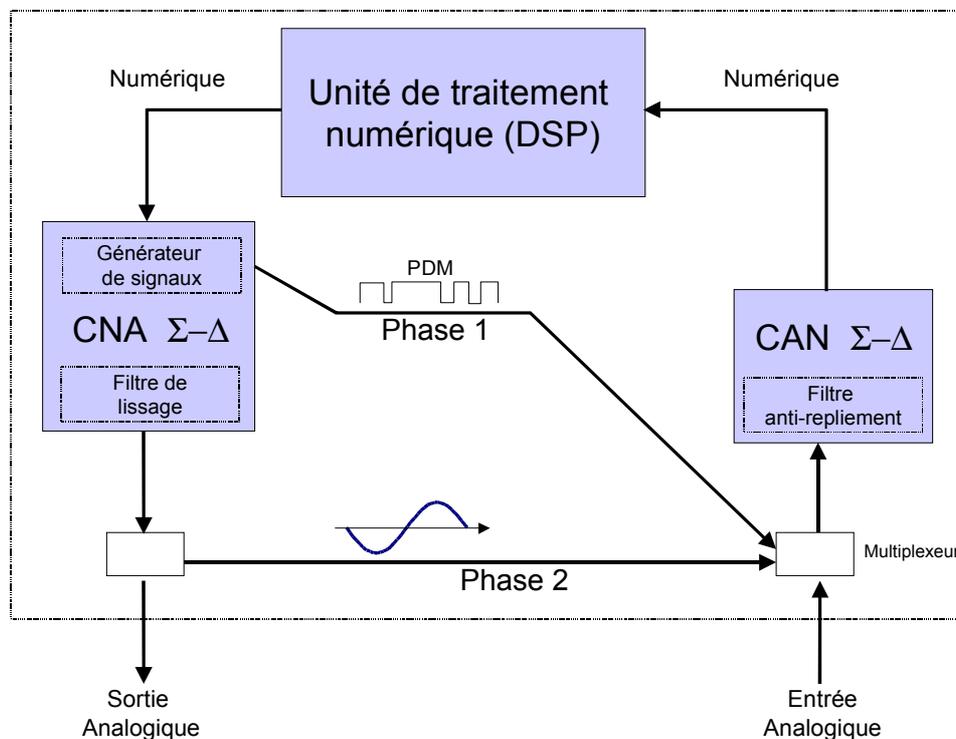


Figure 1.20 : Structure MADBIST

La technique proposée utilise les propriétés de mise en forme du bruit et de sur-échantillonnage des modulateurs $\Sigma-\Delta$ pour générer un signal sinusoïdal analogique précis à partir du générateur de sinus numérique et du modulateur présent dans le CNA.

Le test se déroule en plusieurs étapes :

Au cours de la première étape, la sortie du générateur de signal sinusoïdal numérique est appliquée à l'entrée du modulateur $\Sigma-\Delta$. Le signal de sortie, à modulation de densité d'impulsions (PDM : Pulse-Density-Modulation), attaque directement le Convertisseur A/N sans passer par le filtre de lissage du CNA. En fait, le filtre anti-repliement du CAN permet de filtrer le signal PDM provenant du modulateur et ainsi d'obtenir un signal sinusoïdal analogique précis. Finalement, cette structure de test intégré permet d'évaluer le taux de distorsion harmonique **THD**, le rapport signal sur bruit avec distorsion **SINAD**, le taux de distorsion par intermodulation **IMD** et le **gain** du CAN.

La deuxième étape consiste à tester le filtre de lissage du CNA en lui appliquant le signal PDM et en connectant sa sortie à l'entrée du CAN. Le filtre de lissage est caractérisé grâce au bloc de traitement numérique.

IV.2 Test intégré sans pré-requis sur la présence de ressources matérielles

a) Techniques utilisant les propriétés du bit de poids faible

Le bit de poids faible (le LSB) d'un Convertisseur A/N est le bit qui contient le plus d'informations sur les variations des codes de sortie parce qu'il commute à chaque transition de code. La technique présentée dans ce paragraphe utilise cette propriété pour évaluer les paramètres fonctionnels du convertisseur.

La structure proposée par De Vries [Vri97][Vri98] est une structure de test qui peut être soit partiellement soit totalement intégrée sur la puce. Le principe du test statique proposé consiste à générer une rampe lente en entrée du convertisseur et à évaluer le temps entre chaque commutation du bit de poids faible du CAN. Comme le signal d'entrée est linéaire, chaque intervalle de temps est directement proportionnel à la largeur du palier correspondant de la fonction de transfert. L'ensemble de ces intervalles de temps permet de reconstruire la fonction de transfert du convertisseur et donc de déterminer le **gain**, l'**offset** et les **non-linéarités** du convertisseur.

Pour mesurer la durée écoulée entre deux commutations du LSB, un compteur est incrémenté à chaque front du signal d'échantillonnage (figure 1.21). Plus la fréquence d'échantillonnage est grande par rapport au signal d'entrée, plus la technique de test est précise.

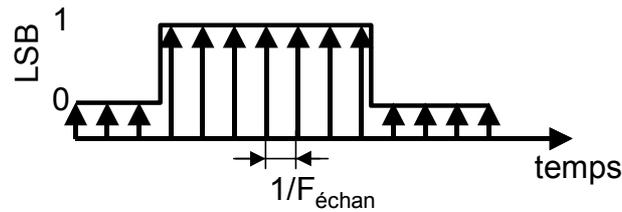


Figure 1.21 : Calcul de l'intervalle de temps entre deux commutations du LSB

Pour illustrer cette technique de test intégré, nous pouvons considérer l'exemple de la procédure de test des non-linéarités du convertisseur, présenté à la figure 1.22.

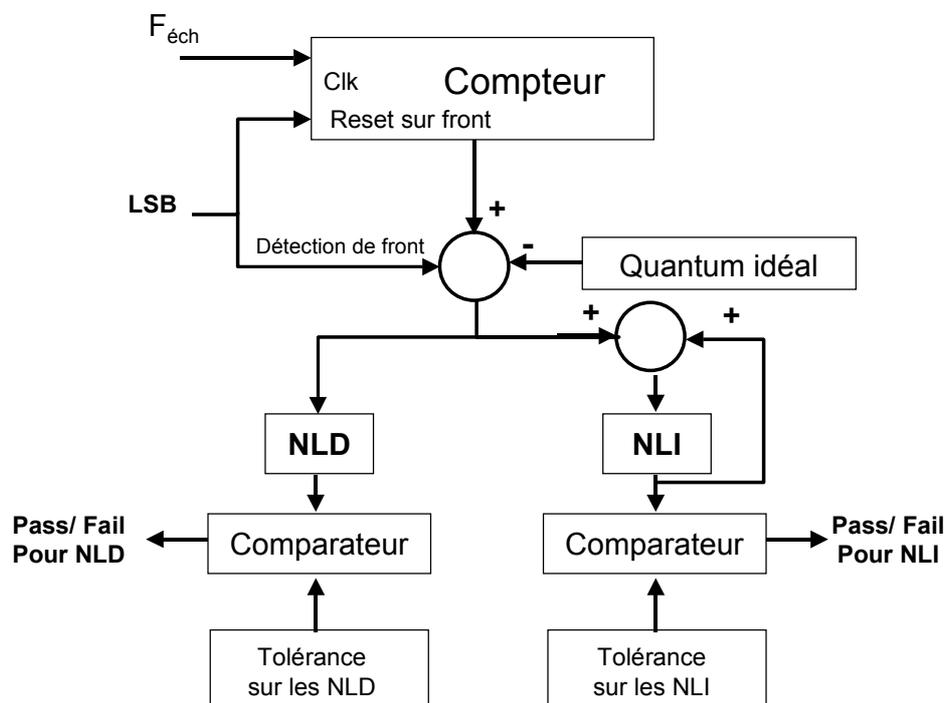


Figure 1.22 : Procédure de test des non-linéarités

Un compteur est incrémenté à chaque impulsion du signal d'horloge. Si une transition du LSB apparaît, la valeur du compteur est soustraite à la valeur idéale qui serait obtenue avec un convertisseur parfait. Le résultat de cette soustraction permet de calculer la valeur des non-linéarités et de les comparer avec les tolérances admises. Cela permet de vérifier que le convertisseur respecte bien les spécifications attendues.

b) Structure BIST pour convertisseur pipeline

Les convertisseurs pipeline sont des convertisseurs constitués de plusieurs étages identiques connectés en série. Généralement, ce type de convertisseurs utilise des techniques de

calibration numériques ou analogiques qui permettent de limiter la sensibilité du convertisseur aux variations technologiques des valeurs des composants. Ces techniques de calibration peuvent masquer certaines erreurs en phase de test et donc rendre le test relativement peu fiable. Dans ce paragraphe, nous présentons des structures de test intégré dédiées à ce type de convertisseur.

➤ ***Structure de test intégré fonctionnel***

La structure de test intégré proposée par Provost, Palermo et Sanchez-Sinencio [Pro98] est élaborée dans le cas particulier des convertisseurs de type pipeline. La technique de test associée est basée sur une approche fonctionnelle. Elle consiste à évaluer la bande passante et le gain de chaque étage et à vérifier la monotonie de la courbe de transfert du convertisseur pour détecter les codes manquants.

Les ressources supplémentaires nécessaires à l'intégration de cette technique de test sont un générateur de rampe et quelques éléments de multiplexage. Le test est effectué en trois phases successives, correspondant à l'évaluation de chaque paramètre.

Durant la première phase, l'analyse de la monotonie du convertisseur est effectuée en appliquant une rampe lente en entrée du convertisseur et en vérifiant que tous les codes apparaissent en sortie du convertisseur. Comme il est difficile de générer une rampe intégrée lente et précise, les auteurs proposent une structure de générateur de rampe auto-calibré qui corrige les éventuelles dispersions sur les valeurs des composants.

Dans la deuxième phase, la bande passante de chaque étage est évaluée en effectuant une étude dans le domaine temporel. Pour cela, le générateur de rampe est configuré en timer analogique qui permet d'effectuer des mesures de grands intervalles de temps.

Au cours de la dernière phase le gain de chaque étage est évalué en le comparant au gain d'un bloc de référence.

➤ ***Structures de test intégré structurel***

Les structures proposées par Peralias, Rueda et Huertas sont deux structures BIST utilisant l'approche structurelle pour convertisseurs pipeline de haute résolution [Per97][Per98][Per00]. Ces deux structures sont basées sur le même schéma général illustré figure 1.23, dans lequel chaque étage du convertisseur est testé séparément en utilisant un élément de référence.

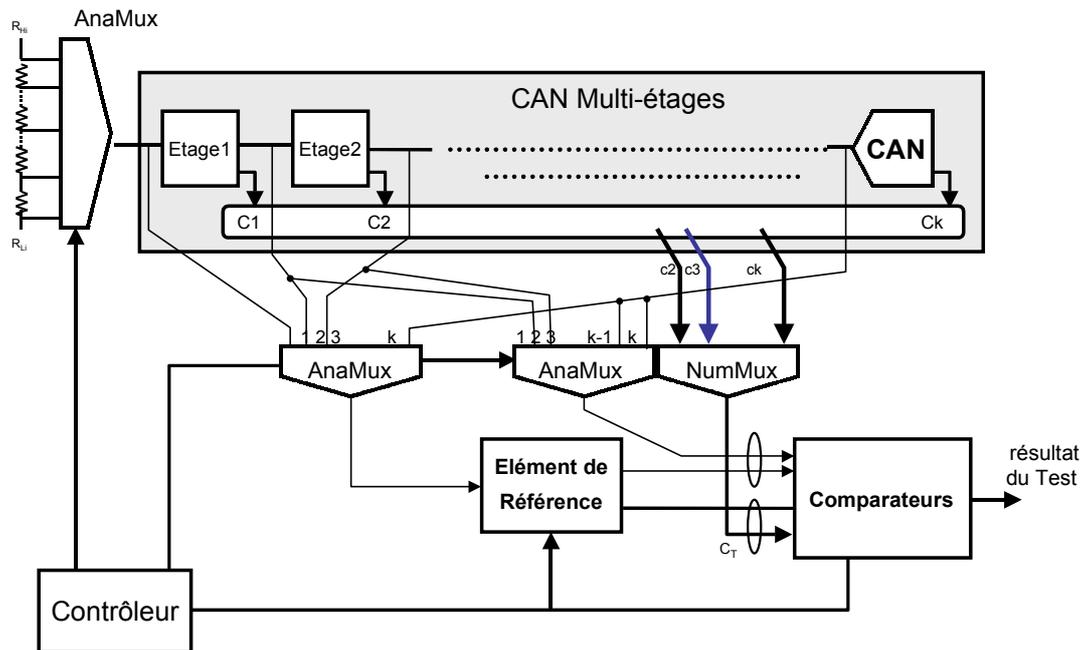


Figure 1.23 : Structure BIST pour CAN pipeline

La structure générale est constituée d'un générateur de tensions continues analogiques, d'un ensemble de multiplexeurs analogiques et numériques et d'un module de comparaison. L'ensemble des multiplexeurs permet de sélectionner l'étage à tester. Le module de comparaison permet de comparer les sorties de l'étage testé avec les sorties d'un élément de référence. La différence majeure entre les deux structures proposées vient principalement du choix de l'élément de référence.

Dans la première structure, l'élément de référence est un étage supplémentaire considéré comme élément de référence. Du fait de la structure multi-étages du convertisseur la surface rajoutée par un seul étage supplémentaire ne représente pas un surcoût important en surface.

La deuxième structure compare les sorties numériques de l'étage testé avec des valeurs idéales qui sont stockées dans une ROM (Read Only Memory).

c) Autres techniques de test intégré

➤ *Test intégré par estimation de la courbe de transfert*

La technique proposée par Sunter et Nagi [Sun97a] [Sun97b] consiste à évaluer directement sur la puce la courbe de transfert du convertisseur à l'aide d'un polynôme du 3^{ème} ordre. Les opérations nécessaires à cette estimation sont relativement simples et peuvent être facilement implantées sur la puce. La connaissance de la courbe de transfert permet de reconstruire la fonction de transfert et par conséquent de déterminer la valeur de **l'erreur de l'offset**, de **l'erreur de gain** et des **non-linéarités** du convertisseur sous test.

➤ *OBIST*

La technique de test intégré par oscillation proposée par Arabi et Kaminska [Ara97] permet de tester n'importe quel circuit analogique ou mixte en utilisant une approche structurelle ou une approche fonctionnelle. Le principe consiste à introduire le circuit sous test dans une boucle et de le forcer à osciller. L'étude de la fréquence d'oscillation du circuit permet alors de déterminer les paramètres recherchés.

Dans le cas du test des CAN, cette technique de test permet d'effectuer un test fonctionnel dans le but de déterminer les **non-linéarités différentielles**, les **non-linéarités intégrales** et le **temps de conversion** du CAN. La structure de test obtenue est représentée figure 1.24.

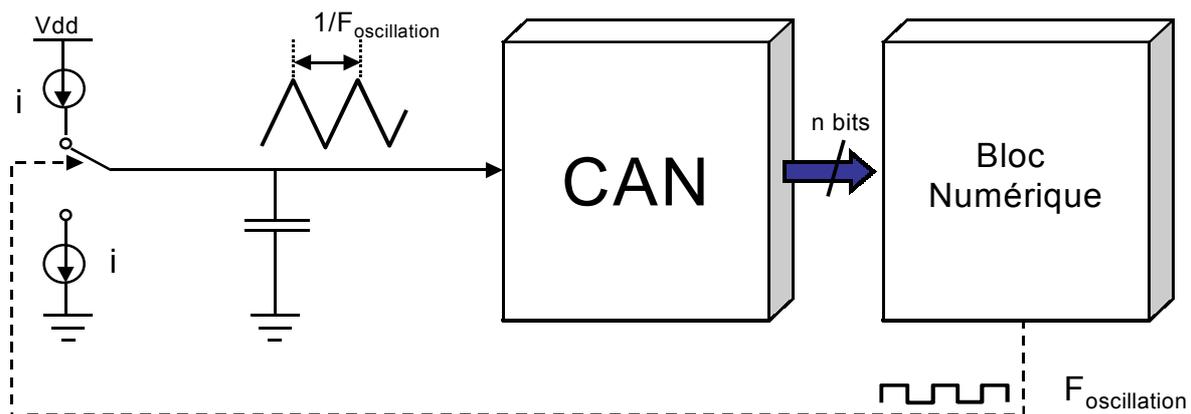


Figure 1.24 : Technique basée sur l'oscillation du CAN sous test

Elle est composée d'un bloc numérique, de deux générateurs de courant, d'un interrupteur et d'une capacité de charge. Le bloc numérique commande l'interrupteur pour charger la capacité avec un courant i ou $-i$ et fournit au convertisseur un signal triangulaire dont la fréquence dépend des caractéristiques fonctionnelles à évaluer.

V. Conclusion

Ce premier chapitre a permis de préciser le contexte de notre travail.

Dans un premier temps, nous avons introduit les différents paramètres fonctionnels permettant de caractériser un convertisseur analogique/Numérique. Ces paramètres sont de deux types :

- les erreurs inhérentes à l'opération de conversion de signaux analogiques en signaux numériques (bruit de quantification),
- les erreurs présentes dans un CAN réel, dues aux imperfections des composants.

Ensuite, nous avons détaillé les deux techniques de test les plus utilisées dans l'industrie : le test par histogramme et le test par analyse fréquentielle. Le test par histogramme est basé sur une étude statistique de la fréquence d'apparition des codes de sortie. Il permet de déterminer l'erreur d'offset, l'erreur de gain, les non-linéarités et les codes manquants du convertisseur. Le test par analyse fréquentielle consiste à étudier le spectre calculé à partir des codes de sortie du convertisseur pour un signal d'entrée sinusoïdal. Cette technique permet de déterminer le rapport signal sur bruit avec distorsion, le taux de distorsion harmonique, le taux de distorsion par intermodulation et le jitter du convertisseur. Ces deux techniques de test sont globalement complémentaires puisqu'elles permettent de déterminer des paramètres fonctionnels différents.

Enfin, la dernière partie du chapitre est consacrée à la présentation des différentes structures de test intégré proposées dans la littérature. Ces structures peuvent être divisées en deux catégories distinctes. D'une part, les structures utilisant des éléments déjà présents sur la puce, et d'autre part, les structures sans pré-requis sur les ressources matérielles présentes sur le circuit intégré à tester.

Chapitre 2:

Analyseur de réponses de test : architecture haut niveau

I. Introduction

Dans le premier chapitre, nous avons présenté les deux techniques les plus couramment utilisées dans l'industrie pour tester les Convertisseurs Analogique/Numérique : les techniques de test par histogramme et par Transformée de Fourier Rapide (TFR). Ces techniques de test externe étant utilisées pour leur capacité à évaluer les paramètres fonctionnels du convertisseur, il est intéressant d'étudier leur possible intégration sur la même puce que le CAN sous test. Le test par Transformée de Fourier Rapide nécessite un traitement en temps réel des sorties du convertisseur. Ce type de traitement impose l'utilisation de calculateurs rapides et complexes impliquant une surface de silicium prohibitive dans un contexte de test intégré où la surface additionnelle doit être minimale.

Nos travaux se sont par conséquent tournés vers le développement d'une structure de test intégré basée sur la technique de test par histogramme. De manière classique, l'intégration d'une technique de test suppose d'être capable de générer de manière interne les stimuli de test et d'analyser de manière interne les réponses de test. La structure BIST dédiée au test des CAN présentée figure 2.1 comporte donc un générateur de stimuli analogiques et un analyseur de réponses numériques de test.

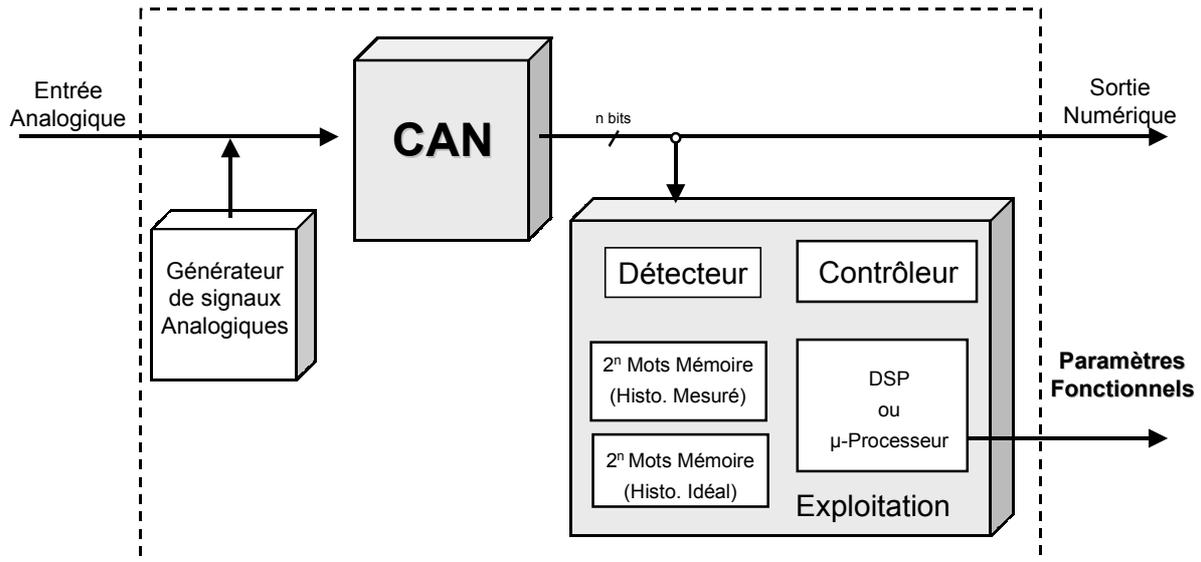


Figure 2.1 : Structure BIST dédiée au test d'un CAN

L'étude de l'intégration du générateur de signaux analogiques sera développée au chapitre 4. Dans ce chapitre, nous ne traiterons que les problèmes liés à l'intégration de l'analyseur de réponses de test.

La fonction principale de l'analyseur de réponses de test consiste à collecter les échantillons en sortie du convertisseur et à extraire les paramètres fonctionnels du CAN à partir d'une analyse statistique de ces échantillons. L'architecture générale de ce bloc comporte donc trois modules distincts :

- Un module de détection du code en sortie du convertisseur sous test.
- Un module d'exploitation effectuant l'ensemble des opérations d'extraction des paramètres fonctionnels.
- Un contrôleur gérant l'ensemble des phases de la procédure de test.

Le module d'exploitation représente le module critique de la structure dans la mesure où il comporte la plupart des ressources matérielles nécessaires à l'intégration de la technique de test, à la fois en termes de ressources mémoires et de ressources opératives. Plus précisément pour un convertisseur A/N de n bits, 2 RAM (Random Access Memory) de 2^n mots mémoire sont nécessaires pour stocker l'histogramme expérimental et l'histogramme de référence. L'extraction des caractéristiques du convertisseur à partir de la comparaison de ces deux histogrammes fait ensuite intervenir des opérations complexes qui nécessitent l'utilisation d'un calculateur sophistiqué (DSP : Digital Signal Processor ou Microprocesseur). Ainsi, l'intégration directe de la technique de test par histogramme conduit à un coût prohibitif en

surface. Afin de rendre cette technique viable dans un contexte de test intégré, nous devons apporter des modifications de manière à n'avoir qu'un nombre réduit d'informations à stocker sur la puce et n'avoir que de simples opérations à effectuer pour évaluer les caractéristiques du convertisseur. Ce chapitre est dédié à la présentation des concepts mis en œuvre permettant de définir une architecture optimisée pour le module d'exploitation.

La première partie de ce chapitre est consacrée à l'étude des différentes techniques d'échantillonnage dans l'objectif de limiter le nombre d'échantillons à traiter. Ensuite, nous présenterons les simplifications faites sur les expressions permettant l'extraction des différents paramètres fonctionnels des convertisseurs. Enfin, nous proposerons des solutions pour réduire la mémoire nécessaire à l'intégration de la technique de test par histogramme.

II. Diminution du volume de données à traiter

Le test par histogramme est basé sur une étude statistique de la fréquence d'apparition des codes de sortie du convertisseur sous test. Or, pour avoir une bonne validité statistique, un très grand nombre d'échantillons doit être collecté de manière aléatoire sur toute la dynamique du signal d'entrée. Dans le cas d'un signal d'entrée sinusoïdal, les mathématiques statistiques permettent d'estimer ce nombre d'échantillons par l'expression [DOE84] [LIB96] :

$$N_T \geq \frac{Z_{\alpha/2}^2 \cdot \pi \cdot 2^{n-1}}{\delta_{\text{NLD}}^2} \quad (2.1)$$

où N_T représente le nombre d'échantillons nécessaires pour effectuer une mesure de non-linéarité avec une précision δ_{NLD} (exprimée en LSB) dans au moins $100 \cdot (1-\alpha)$ pour-cent des cas, n représentant le nombre de bits du convertisseur et $Z_{\alpha/2}$ la valeur de la fonction de distribution normale standard pour un taux de réussite de $(1-\alpha)$.

Par exemple, pour une précision sur la mesure de la non-linéarité différentielle de $\delta_{\text{NLD}} = 0,2$ LSB dans 95% des cas ($\alpha = 0,05$), il sera nécessaire d'acquérir 67 000 échantillons pour un convertisseur de 8 bits et 17 125 000 échantillons pour un convertisseur de 16 bits. La surface d'une structure BIST dépendant généralement du volume de données à traiter, l'échantillonnage aléatoire ne semble pas être la solution la plus appropriée pour l'intégration du test par histogramme.

En fait, il existe une alternative à l'échantillonnage aléatoire permettant de garder la même précision en terme de statistique, c'est l'échantillonnage cohérent. Cette technique d'échantillonnage consiste à répartir uniformément les échantillons sur un nombre fixe de périodes du signal d'entrée. Pour cela, la fréquence d'échantillonnage $F_{\text{éch}}$ et la fréquence du signal d'entrée F_{in} sont choisies cohérentes entre elles. Pour obtenir cette cohérence, ces fréquences sont calculées pour que l'acquisition des N_T échantillons corresponde exactement à un nombre entier N_P de périodes du signal d'entrée, et qu'il n'y ait aucune répétition au sein de cette séquence. Toutes ces conditions peuvent se résumer par l'équation (1.1) donnée au chapitre 1.

$$\frac{F_{\text{in}}}{F_{\text{éch}}} = \frac{N_P}{N_T} \quad \text{avec } N_P \text{ et } N_T \text{ premiers entre eux.} \quad (1.1)$$

Reprenons l'exemple du signal d'entrée sinusoïdal dans le cas d'un échantillonnage cohérent. Il est possible d'utiliser l'expression de la probabilité d'apparition du code i :

$$p(i) = \frac{1}{\pi} \left\{ \arcsin \left[\left(\frac{2i - 2^n}{2^n} \right) \cdot \frac{PE}{A_{\text{in}}} \right] - \arcsin \left[\left(\frac{2i - 2^n - 2}{2^n} \right) \cdot \frac{PE}{A_{\text{in}}} \right] \right\} \quad (2.2)$$

où PE correspond à la pleine échelle du CAN et A_{in} correspond à l'amplitude crête à crête du signal d'entrée. Le code le plus sensible à une erreur de calcul est le code dont la probabilité d'apparition est minimum. Dans le cas d'un signal d'entrée sinusoïdal, le code milieu (2^{n-1}) a la plus petite probabilité d'apparition donnée par l'expression suivante :

$$p(2^{n-1}) = \frac{2}{\pi} \cdot \sin^{-1} \left[\frac{PE}{2^n \cdot A_{\text{in}}} \right] = \frac{k}{N_T} \quad (2.3)$$

où k représente le nombre d'échantillons associés au code $i = 2^{n-1}$ et N_T le nombre total d'échantillons.

Le nombre d'échantillons par code étant un entier, nous pouvons considérer que l'erreur maximale de mesure est de 1 échantillon pour chaque code. La précision minimale sera alors de $\delta_{\text{NLD}} = 1/k$. Finalement, le nombre d'échantillons nécessaires pour assurer une précision de δ_{NLD} peut être calculé à l'aide de l'expression suivante :

$$N_T = \frac{\pi}{\delta_{\text{NLD}} \cdot \sin^{-1} \left[\frac{PE}{2^n A_{\text{in}}} \right]} \quad (2.4)$$

Par exemple, pour un signal d'entrée pleine échelle ($A_{in} = PE$) et une précision sur la mesure des NLD $\delta_{NLD} = 0,2 \text{ LSB}$, il sera nécessaire d'acquérir 4 022 échantillons pour un convertisseur de 8 bits et 1 029 437 échantillons pour un convertisseur de 16 bits. Ces valeurs sont à comparer avec les 67 000 échantillons pour un convertisseur de 8 bits et les 17 125 000 échantillons pour un convertisseur de 16 bits nécessaires dans le cas d'un échantillonnage aléatoire.

Le tableau suivant permet de comparer les nombres d'échantillons nécessaires selon le type d'échantillonnage utilisé.

n	Echantillonnage	
	Aléatoire	Cohérent
8	67 000	4 022
10	267 500	16 085
12	1 050 000	64 340
14	4 282 500	257 360
16	17 125 000	1 029 437

Tableau 2.1 : Nombre d'échantillons pour le test par histogramme

Pour limiter le volume de données à traiter nous considérerons toujours par la suite que nous travaillons avec un échantillonnage cohérent.

III. Simplification des calculs d'exploitation

La structure BIST originale comporte un DSP ou un microprocesseur pour effectuer les opérations complexes d'extraction des paramètres. Dans cette section nous présentons des simplifications importantes de ces calculs d'extraction. Mais avant toute chose, rappelons que les opérations d'extraction dépendent directement du type de signal d'entrée utilisé et que par conséquent, nous traiterons séparément le cas d'un signal d'entrée triangulaire et le cas d'un signal d'entrée sinusoïdal.

III.1. Signal d'entrée triangulaire

Dans le cas d'un signal d'entrée triangulaire, l'histogramme de référence obtenu pour un convertisseur parfait est très simple (figure 2.2).

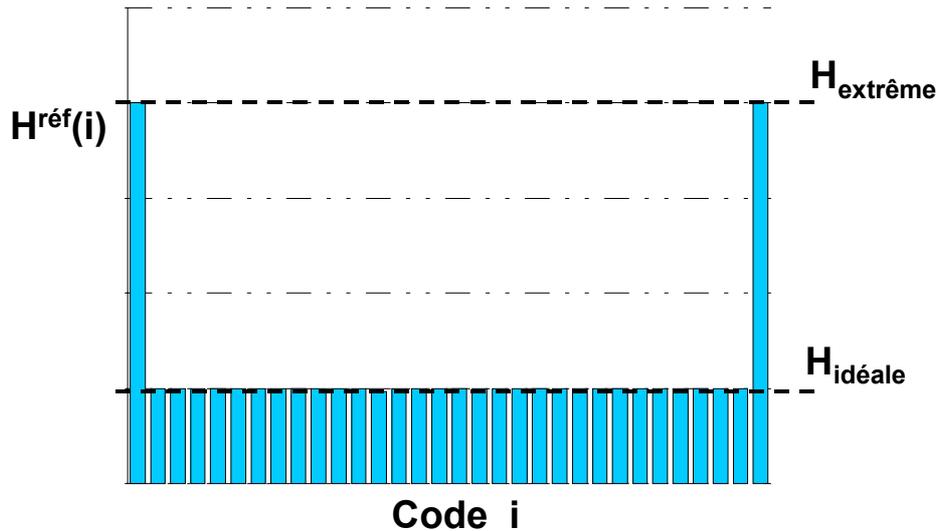


Figure 2.2 : Histogramme idéal pour un signal d'entrée triangulaire

L'histogramme a les mêmes propriétés linéaires que le signal dont il découle. En effet, comme la pente du signal est constante, la probabilité ($H_{idéale}/N_T$) pour qu'un code soit présent en sortie du convertisseur est elle aussi constante. Seule la fréquence d'apparition $H_{extrême}$ des deux codes extrêmes est plus élevée dans le cas où le signal d'entrée a une amplitude plus grande que la pleine échelle du CAN.

La précision de l'évaluation des paramètres à l'aide de l'histogramme dépend directement de la valeur de $H_{idéale}$. Plus clairement, la précision sur la mesure des non-linéarités différentielles δ_{NLD} est égale à la précision sur la construction de l'histogramme :

$$\delta_{NLD} = \frac{\delta H^{exp}(i)}{H_{idéale}} \quad \text{où } \delta H^{exp}(i) \text{ est l'erreur de mesure sur le code } i \quad (2.5)$$

En pratique, la fréquence d'apparition $H^{exp}(i)$ est forcément égale à un entier et donc l'erreur maximale de mesure sur un code est égale à 1. Ainsi, la précision sur la mesure des NLD est donnée par l'expression suivante :

$$\delta_{NLD} = \frac{1}{H_{idéale}} \quad (2.6)$$

En utilisant l'expression (1.18) obtenue au chapitre 1, nous avons une relation entre le nombre total d'échantillons traités et la valeur de $H_{idéale}$:

$$H_{idéale} = N_T \times \frac{PE}{2^n \cdot A_{in}} \quad (1.18)$$

Connaissant l'amplitude du signal d'entrée et la précision désirée sur les mesures des paramètres, nous pouvons déterminer le nombre d'échantillons qu'il sera nécessaire d'acquérir :

$$N_T = \frac{2^n \times A_{in}}{PE \times \delta_{NLD}} \quad (2.7)$$

Pour évaluer la valeur de l'erreur d'offset et de gain, il peut être intéressant de considérer le signal « vu » à travers le convertisseur sous test. Plus précisément, nous allons étudier le signal analogique virtuellement reconstitué à partir de la sortie numérique du CAN pour un signal d'entrée parfait. Comme le signal analogique obtenu a les mêmes propriétés (gain, offset) que le convertisseur sous test, l'étude de ses caractéristiques fonctionnelles est équivalente à l'étude des caractéristiques du convertisseur.

a) Erreur d'offset

La figure 2.3 illustre l'influence d'une erreur d'offset sur le signal reconstitué et sur l'histogramme expérimental. Nous pouvons observer que pour un signal d'entrée d'amplitude supérieure à la pleine échelle du convertisseur, une erreur d'offset ne modifie que la fréquence d'apparition des deux codes extrêmes. Nous pouvons aussi remarquer que l'erreur d'offset du CAN est opposée à l'erreur d'offset du signal reconstitué.

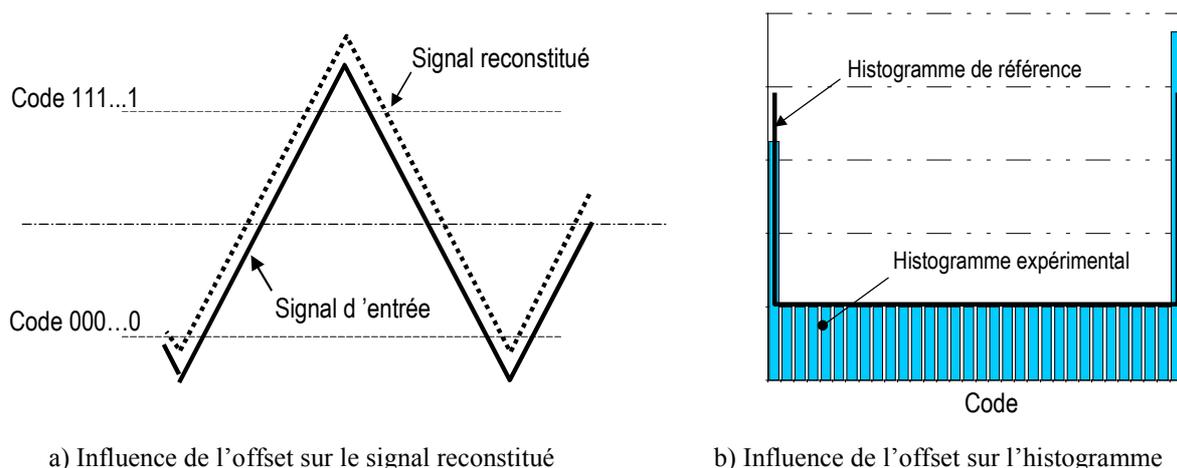


Figure 2.3 : Influence de l'erreur d'offset

L'erreur d'offset a un effet linéaire sur la fréquence d'apparition des deux codes extrêmes. Ainsi, l'expression de l'erreur d'offset du convertisseur peut facilement se déduire à partir de la fréquence d'apparition du code 2^n :

$$\text{Offset} = \frac{H_{\text{extrême}} - H^{\text{exp}}(2^n)}{H_{\text{idéale}}} \quad (2.8)$$

Or, cette mesure peut être faussée si le convertisseur comporte aussi une erreur de gain. Pour que la mesure de l'offset soit indépendante de la valeur du gain, il suffit d'utiliser le même type d'expression que l'équation (2.8), mais cette fois pour le code 1 :

$$\text{Offset} = \frac{H^{\text{exp}}(1) - H_{\text{extrême}}}{H_{\text{idéale}}} \quad (2.9)$$

En combinant ces deux équations, nous obtenons l'expression suivante pour l'erreur d'offset :

$$\boxed{\text{Offset} = \frac{H^{\text{exp}}(1) - H^{\text{exp}}(2^n)}{2 \cdot H_{\text{idéale}}}} \quad (2.10)$$

Ainsi, l'erreur d'offset est simplement proportionnelle à la soustraction entre les fréquences d'apparition des deux codes extrêmes. Par conséquent, les ressources opératives nécessaires à l'évaluation de cette erreur se résument à un additionneur pour calculer les fréquences d'apparition $H^{\text{exp}}(1)$ et $H^{\text{exp}}(2^n)$, un soustracteur pour effectuer la différence entre ces deux fréquences d'apparition et un diviseur pour la division par $2 \cdot H_{\text{idéale}}$.

b) Erreur de gain

Comme pour l'évaluation de l'erreur d'offset, nous étudions le signal virtuellement reconstitué en sortie du convertisseur. La figure 2.4 illustre l'influence d'une erreur de gain sur ce signal reconstitué et sur l'histogramme obtenu. Nous pouvons observer qu'une erreur de gain affecte de manière uniforme tous les codes du convertisseur (excepté les codes extrêmes dans la mesure où l'amplitude du signal d'entrée est supérieure à la pleine échelle).

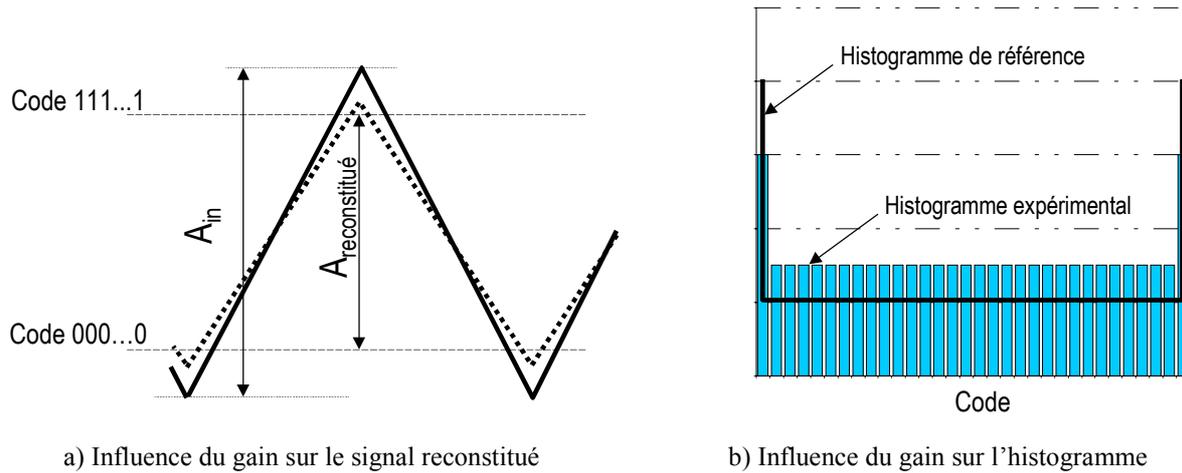


Figure 2.4 : Influence de l'erreur de gain

De par sa définition même, le gain g du convertisseur s'exprime comme le rapport entre l'amplitude du signal d'entrée et l'amplitude du signal reconstitué : $g = \frac{A_{in}}{A_{reconstitué}}$. Grâce aux propriétés linéaires du signal triangulaire, l'amplitude du signal reconstitué est directement proportionnelle à la fréquence d'apparition d'un code quelconque du convertisseur $H^{exp}(i)$ (hormis les codes extrêmes).

Le gain du convertisseur s'exprime par conséquent comme le rapport entre la valeur $H_{idéale}$ de l'histogramme idéal et la valeur mesurée $H^{exp}(i)$ de l'histogramme expérimental pour un code quelconque :

$$\text{Gain} = \frac{A_{in}}{A_{reconstitué}} = \frac{H_{idéale}}{H^{exp}(i)} \quad (2.11)$$

L'erreur de gain, exprimée en LSB ou en fraction de Pleine Echelle, peut être déterminée à partir des expressions suivantes :

$$\text{Erreur_de_Gain}_{(LSB)} = \left(\frac{H^{exp}(i)}{H_{idéale}} - 1 \right) \times 2^{n-1} \text{ (LSB)} \quad (2.12)$$

$$\text{Erreur_de_Gain}_{(PE)} = \left(\frac{H^{exp}(i)}{H_{idéale}} - 1 \right) \text{ (PE)} \quad (2.13)$$

En théorie, la mesure de la fréquence d'apparition $H^{\text{exp}}(i)$ d'un code quelconque du convertisseur (hormis les codes extrêmes) permet donc d'obtenir une image de l'erreur de gain. Toutefois en pratique, il peut se révéler dangereux d'estimer l'erreur de gain en utilisant la mesure de la fréquence d'apparition d'un seul code. En effet, si l'on prend en compte les variations du pas de quantification dues à d'éventuelles non-linéarités du convertisseur, toute erreur de non-linéarité sur le code pris comme référence se répercutera sur la mesure du gain. Pour limiter cette influence, nous proposons d'effectuer la moyenne de mesure du gain sur m codes différents. Ces m codes sont choisis dans la partie centrale de l'histogramme. Nous obtenons alors l'expression suivante pour l'erreur de gain, exprimée en fraction de PE :

$$\text{Erreur_de_Gain}_{PE} = \left(\frac{\sum_{i=N1}^{N2} H^{\text{exp}}(i)}{m \cdot H_{\text{idéale}}} - 1 \right) \quad \text{avec } N1 = \frac{2^n - m}{2} + 1 \text{ et } N2 = \frac{2^n + m}{2} \quad (2.14)$$

Concernant les ressources nécessaires pour extraire ce paramètre, les différentes opérations utilisées sont l'addition, la division par une constante et la soustraction par 1.

c) Erreurs de Non-linéarité

La figure 2.5 illustre l'influence des non-linéarités du convertisseur sur l'allure de l'histogramme expérimental.

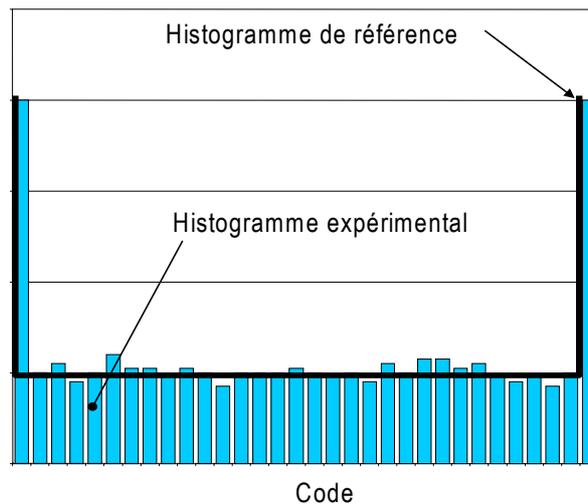


Figure 2.5 : Influence des erreurs de non-linéarités

La Non-Linéarité Différentielle $NLD(i)$ d'un code i donné est définie comme la différence entre la valeur de l'histogramme expérimental $H^{\text{exp}}(i)$ et la valeur de l'histogramme de référence correspondante $H^{\text{réf}}(i)$ [Mah87]. La Non-Linéarité Intégrale $NLI(i)$ d'un code i est alors exprimée comme la somme cumulative des NLD de tous les codes précédents [Mah87]. Pour la détermination des Non-Linéarités Différentielle (NLD) et Intégrale (NLI), nous utilisons les expressions classiques suivantes :

$$NLD(i) = \frac{H^{\text{exp}}(i) - H_{\text{idéale}}}{H_{\text{idéale}}} = \frac{H^{\text{exp}}(i)}{H_{\text{idéale}}} - 1 \quad (2.15)$$

$$NLI(i) = \sum_{j=1}^i NLD(j) \quad (2.16)$$

Ces deux expressions ne comportent que des fonctions simples. Il n'y a par conséquent aucun besoin de modifier ces expressions pour intégrer le calcul et les ressources opératives nécessaires sont simplement un additionneur, un soustracteur et un diviseur.

III.2. Signal d'entrée sinusoïdal

Dans le cas d'un signal d'entrée sinusoïdal, l'histogramme de référence est beaucoup plus compliqué que dans le cas d'un signal d'entrée triangulaire (figure 2.6).

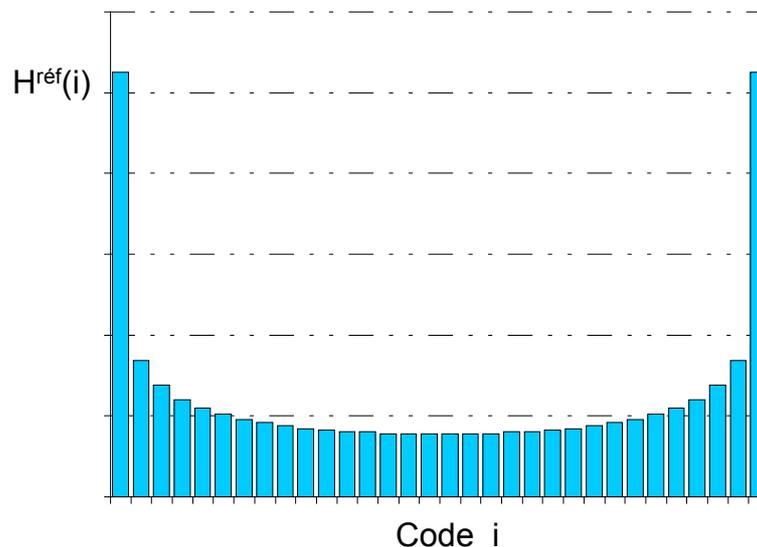


Figure 2.6 : Histogramme de référence pour un signal sinusoïdal

En effet, les valeurs de l'histogramme suivent la loi de distribution du signal sinusoïdal. Or, cette loi de distribution n'est pas constante car elle dépend de la pente du signal (plus la pente

du signal est grande, moins il y aura d'échantillons en sortie). La valeur de l'histogramme pour chaque code peut être calculée à l'aide des expressions suivantes définies au chapitre précédent :

$$H^{\text{réf}}(1) = H^{\text{réf}}(2^n) = \frac{N_T}{\pi} \left\{ \sin^{-1} \left[\left(\frac{1}{2^{n-1}} - 1 \right) \cdot \frac{PE}{A_{\text{in}}} \right] - \frac{\pi}{2} \right\} \quad (1.24)$$

et

$$H^{\text{réf}}(i) = \frac{N_T}{\pi} \left\{ \sin^{-1} \left[\left(\frac{2i - 2^n}{2^n} \right) \cdot \frac{PE}{A_{\text{in}}} \right] - \sin^{-1} \left[\left(\frac{2i - 2^n - 2}{2^n} \right) \cdot \frac{PE}{A_{\text{in}}} \right] \right\} \quad \forall i \in [2 ; 2^n - 1] \quad (1.25)$$

Le nombre total d'échantillons nécessaires dépend de l'amplitude du signal d'entrée. Nous avons démontré au paragraphe II que le nombre d'échantillons N_T dans le cas d'un signal sinusoïdal est égal à :

$$N_T = \frac{\pi}{\delta_{\text{NLD}} \cdot \sin^{-1} \left[\frac{PE}{2^n A_{\text{in}}} \right]} \quad (2.4)$$

a) Erreur d'offset

Habituellement, l'erreur d'offset est déterminée à l'aide de la valeur de l'histogramme expérimental pour les deux codes extrêmes. La figure 2.7 représente l'influence de cette erreur sur le signal reconstitué à partir des sorties du convertisseur sous test et l'influence sur l'histogramme expérimental.

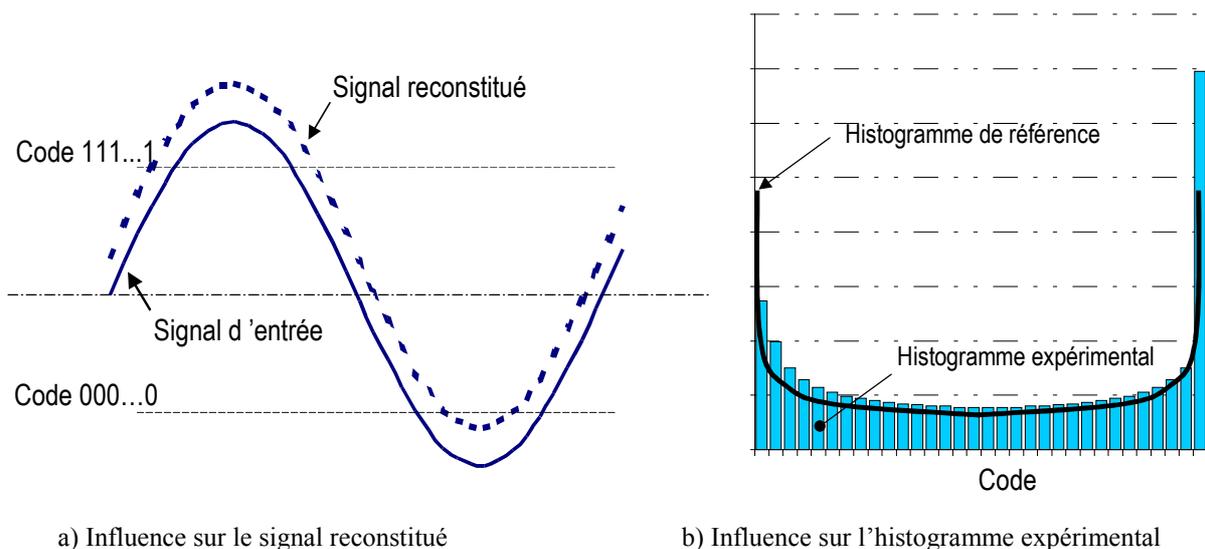


Figure 2.7 : Influence de l'erreur de d'offset

L'expression originale de l'erreur d'offset donnée dans [MAH 84] est la suivante :

$$\text{Offset} = -\frac{\text{PE}}{2} \cdot \frac{\cos \frac{\pi H^{\text{exp}}(1)}{N_T} - \cos \frac{\pi H^{\text{exp}}(2^n)}{N_T}}{\cos \frac{\pi H^{\text{exp}}(1)}{N_T} + \cos \frac{\pi H^{\text{exp}}(2^n)}{N_T}} \quad (2.17)$$

où PE représente la pleine échelle du CAN et N_T le nombre total d'échantillons. A l'aide des formules classiques de trigonométrie, on peut transformer l'équation précédente et obtenir une nouvelle formulation de l'offset :

$$\text{Offset} = \frac{\text{PE}}{2} \cdot \frac{2 \cdot \sin \frac{\pi(H^{\text{exp}}(1) + H^{\text{exp}}(2^n))}{2N_T} \cdot \sin \frac{\pi(H^{\text{exp}}(1) - H^{\text{exp}}(2^n))}{2N_T}}{2 \cdot \cos \frac{\pi(H^{\text{exp}}(1) + H^{\text{exp}}(2^n))}{2N_T} \cdot \cos \frac{\pi(H^{\text{exp}}(1) - H^{\text{exp}}(2^n))}{2N_T}} \quad (2.18)$$

Ensuite, de cette expression, nous pouvons faire apparaître un produit de tangentes :

$$\text{Offset} = -\frac{\text{PE}}{2} \cdot \tan \frac{\pi(H^{\text{exp}}(1) + H^{\text{exp}}(2^n))}{2N_T} \cdot \tan \frac{\pi(H^{\text{exp}}(1) - H^{\text{exp}}(2^n))}{2N_T} \quad (2.19)$$

L'analyse de cette nouvelle expression conduit à deux observations.

La première observation importante concerne la somme des valeurs de l'histogramme pour les deux codes extrêmes $H^{\text{exp}}(1) + H^{\text{exp}}(2^n)$. Dans le cas d'une erreur d'offset, cette somme peut être considérée comme constante. En effet, une erreur d'offset positive (figure 2.8) entraîne une diminution de $H^{\text{exp}}(1)$ et une augmentation de $H^{\text{exp}}(2^n)$, et inversement dans le cas d'une erreur d'offset négative. Pour estimer l'erreur induite par cette approximation, nous considérons l'exemple d'un histogramme construit pour un convertisseur de 8 bits avec 8191 échantillons collectés sur le signal d'entrée d'amplitude crête à crête égale à 262 LSB (PE=256 LSB). Les équations (2.20) et (2.21) définissent la valeur de l'histogramme pour les deux codes extrêmes dans le cas d'une erreur d'offset, exprimée en LSB ($\text{Offset}_{\text{LSB}}$).

$$H^{\text{exp}}(2^n) = \frac{N_T}{\pi} \left\{ \frac{\pi}{2} - \sin^{-1} \left[\left(1 - \frac{1 + \text{Offset}_{\text{LSB}}}{2^{n-1}} \right) \cdot \frac{\text{PE}}{A_{\text{in}}} \right] \right\} \quad (2.20)$$

et

$$H^{\text{exp}}(1) = \frac{N_T}{\pi} \left\{ \sin^{-1} \left[\left(\frac{1 - \text{Offset}_{\text{LSB}}}{2^{n-1}} - 1 \right) \cdot \frac{\text{PE}}{A_{\text{in}}} \right] - \frac{\pi}{2} \right\} \quad (2.21)$$

Dans le cas particulier d'une erreur d'offset de 1 LSB, la fréquence d'apparition du code 2^n est égale à 722 et la fréquence d'apparition du code 1 est égale à 559.

De plus, la valeur de l'histogramme de référence pour le code 1 est donnée par l'expression :

$$H^{\text{réf}}(1) = \frac{N_T}{\pi} \left\{ \sin^{-1} \left[\left(\frac{1}{2^{n-1}} - 1 \right) \cdot \frac{\text{PE}}{A_{\text{in}}} \right] - \frac{\pi}{2} \right\} \cong 646$$

Nous pouvons en déduire l'erreur faite par l'approximation grâce à l'expression suivante :

$$\text{Erreur} = \left(\frac{H^{\text{exp}}(2^n) + H^{\text{exp}}(1)}{2 \cdot H^{\text{réf}}(1)} - 1 \right) \times 100 = 0.7\%$$

Ainsi, une erreur d'offset aussi importante que 1 LSB n'est responsable que de 0,7% de variation sur la somme $H^{\text{exp}}(1) + H^{\text{exp}}(2^n)$. En conséquence, notre première approximation consiste à remplacer cette somme expérimentale par une valeur constante dérivée de l'histogramme de référence $H^{\text{exp}}(1) + H^{\text{exp}}(2^n) \cong 2 H^{\text{réf}}(1)$, où $H^{\text{réf}}(1)$ est une valeur prédéterminée pour l'application envisagée.

Une deuxième observation sur l'équation (2.19) concerne la valeur de la différence entre les fréquences d'apparition des deux codes extrêmes $H^{\text{exp}}(1) - H^{\text{exp}}(2^n)$. Cette valeur est très petite par rapport au nombre total d'échantillons N_T , et par conséquent nous pouvons utiliser l'approximation classique du premier ordre : $\tan(\alpha) \cong \alpha$, pour α petit. Nous obtenons alors l'expression suivante :

$$\text{Offset} \cong - \frac{\text{PE}}{2} \cdot \left(\tan \frac{\pi \cdot H(1)}{N_T} \right) \times \left(\frac{\pi \cdot (H^{\text{exp}}(2^n) - H^{\text{exp}}(1))}{2N_T} \right) \quad (2.22)$$

Finalement, l'erreur d'offset peut être exprimée par :

$$\boxed{\text{Offset} \approx \frac{(H^{\text{exp}}(1) - H^{\text{exp}}(2^n))}{K}} \quad (2.23)$$

où $K = \left(\frac{\pi \cdot PE}{4 \cdot N_T} \times \left[\tan \frac{\pi \cdot H^{réf}(1)}{N_T} \right] \right)^{-1}$ est une valeur constante qui peut être prédéterminée.

Nous obtenons une expression simple dans laquelle l'offset est proportionnel à la différence entre les fréquences d'apparition des deux codes extrêmes. Le calcul de l'erreur d'offset nécessite donc que des opérateurs simples : additionneur, soustracteur, diviseur.

b) Erreur de gain

Comme pour l'évaluation de l'erreur d'offset, nous étudions le signal virtuellement reconstitué en sortie du convertisseur. La figure 2.8 illustre l'influence d'une erreur de gain sur ce signal reconstitué et sur l'histogramme obtenu.

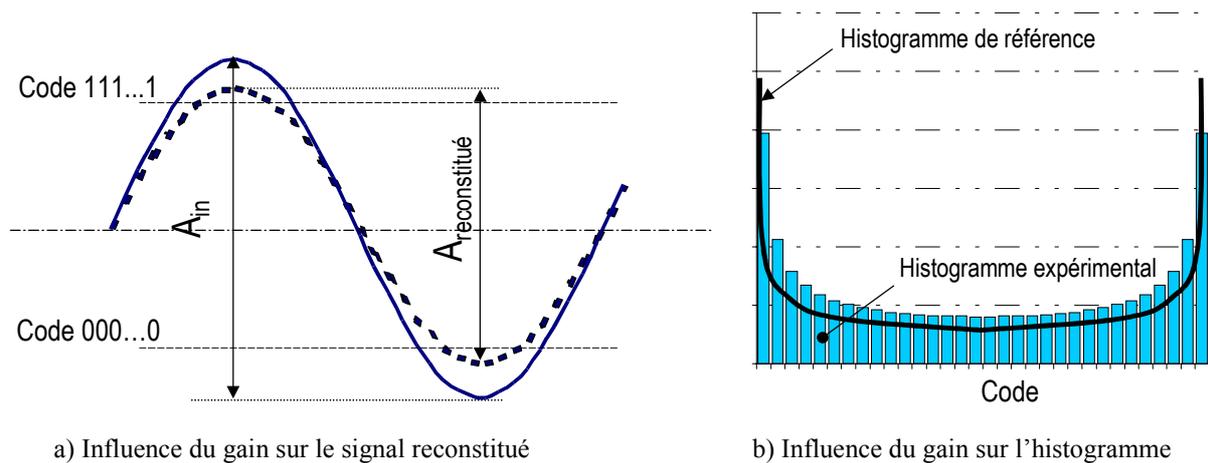


Figure 2.8 : Influence de l'erreur de gain

L'expression classique du gain [Mah87] pour un signal d'entrée sinusoïdal est la suivante :

$$\text{Gain} = \frac{A_{in}}{A_{reconstitué}} = \frac{\frac{A_{in}}{2} \cdot \cos \frac{\pi H^{exp}(2^n)}{N_T}}{\frac{PE}{2} - \text{Offset}_{LSB}} \quad (2.25)$$

où A_{in} correspond à l'amplitude crête à crête de l'entrée sinusoïdale. Cette expression implique le calcul d'une fonction cosinus qu'il est difficile de réaliser directement sur la puce. En outre, elle dépend de la valeur de l'erreur d'offset mesurée, ce qui implique que n'importe quelle erreur sur la détermination de l'offset affectera la détermination de l'erreur de gain.

Dans ce contexte, nous proposons une autre approche pour estimer le gain. Notre solution est basée sur l'observation suivante. L'histogramme obtenu pour un signal d'entrée sinusoïdal présente une section relativement plate dans la partie centrale. Cette section plate correspond aux sorties du CAN pour la partie la plus linéaire de la sinusoïde d'entrée. Par conséquent, nous pouvons faire une analogie avec l'histogramme obtenu pour un signal d'entrée triangulaire. On peut alors appliquer la même expression :

$$\text{Gain} = \frac{H^{\text{réf}}(i)}{H^{\text{exp}}(i)} \quad \text{Avec } H^{\text{exp}}(i) \cong \text{constante} \quad (2.26)$$

On ne pourra utiliser cette expression que pour les codes i dont la fréquence d'apparition est relativement constante, c'est à dire pour i proche de la valeur $2^n/2$. Ainsi, il est possible d'obtenir une évaluation du gain en utilisant simplement la valeur de l'histogramme expérimental $H^{\text{exp}}(i)$ et la valeur de référence correspondante $H^{\text{réf}}(i)$ pour un code i donné dans la section linéaire de la sinusoïde. Cependant comme dans le cas du signal triangulaire, il est raisonnable de faire la moyenne des mesures sur plusieurs codes. Si on considère m codes autour du code central $2^n/2$, nous obtenons l'expression suivante pour l'erreur de gain :

$$\boxed{\text{Erreur_de_Gain}_{(PE)} = \left(\frac{1}{m} \times \sum_{i=N1}^{N2} \frac{H^{\text{exp}}(i)}{H^{\text{réf}}(i)} - 1 \right)} \quad \text{Avec } N1 = \frac{2^n - m}{2} + 1 \text{ et } N2 = \frac{2^n + m}{2} \quad (2.27)$$

Naturellement, plus le nombre m de codes considérés est grand, plus le calcul de gain est précis, sous réserve bien sûr que tous les codes utilisés soient dans la section " plate " de l'histogramme.

Par exemple, dans le cas d'un histogramme obtenu pour un convertisseur de 8 bits ($PE = 256$ LSB) en utilisant 8191 échantillons avec une entrée sinusoïdale d'amplitude crête à crête de 262 LSB, nous observons que la fréquence d'apparition reste constante à plus ou moins un échantillon pour les codes compris entre 92 et 163. Par conséquent, dans ce cas particulier, il sera possible de calculer l'erreur de gain sur 70 codes.

Pour résumer, les ressources opératives nécessaires à l'évaluation de l'erreur de gain sont de simples opérateurs élémentaires : additionneur, soustracteur et diviseur.

c) Erreurs de non-linéarité

Pour la détermination des Non-Linéarités Différentielles (NLD) et Intégrales (NLI), nous utilisons les expressions classiques suivantes [Mah87] :

$$\text{NLD}(i) = \frac{H^{\text{exp}}(i) - H^{\text{réf}}(i)}{H^{\text{réf}}(i)} = \frac{H^{\text{exp}}(i)}{H^{\text{réf}}(i)} - 1 \quad (2.28)$$

$$\text{NLI}(i) = \sum_{j=1}^i \text{NLD}(j) \quad (2.29)$$

Ces expressions n'utilisent que des fonctions simples. Par conséquent aucune simplification n'est nécessaire pour évaluer ces paramètres et les ressources opératives sont là encore : un additionneur, un soustracteur et un diviseur.

III.3. Récapitulatif

Que se soit pour un signal d'entrée triangulaire ou sinusoïdal, seul des opérateurs élémentaires du type additionneur, soustracteur et diviseur sont réellement nécessaires pour évaluer les paramètres fonctionnels du convertisseur sous test compte tenu des expressions proposées. Une première optimisation de l'architecture du module d'exploitation (illustrée figure 2.9) consiste donc à remplacer le DSP ou le microprocesseur utilisé de manière classique en test externe par un module opératif beaucoup plus simple ne comportant que des opérateurs élémentaires.

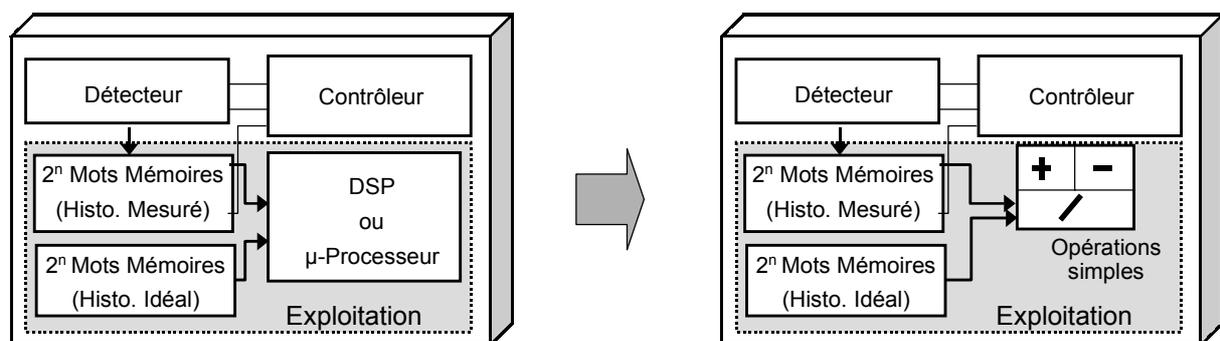


Figure 2.9 : Optimisation des ressources opératives

IV. Minimisation des ressources mémoire

IV.1. Stockage de l'histogramme expérimental

a) Principe de décomposition temporelle

Habituellement, pour effectuer le test par histogramme, il est nécessaire de stocker l'ensemble de l'histogramme expérimental avant d'effectuer l'extraction des paramètres fonctionnels. La première modification que nous proposons consiste à décomposer temporellement la procédure de test. A un haut niveau, cette technique de décomposition temporelle correspond à une division du test en différentes phases successives. A un plus bas niveau, chacune des phases de test est elle-même décomposée en plusieurs étapes élémentaires. En fait, avec ce principe relativement simple les mêmes ressources peuvent être réutilisées pour les différents calculs. Bien sur, cette technique de décomposition temporelle du test illustrée à la figure 2.10 ne peut être appliquée qu'à condition de pouvoir dissocier l'évaluation des différents paramètres.

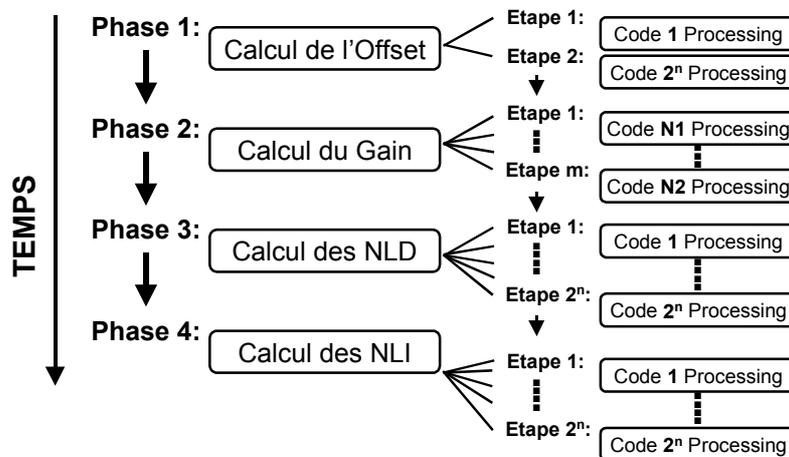


Figure 2.10 : Décomposition de test en différentes phases

Ainsi, dans chacune des phases, une seule caractéristique fonctionnelle du CAN est extraite. L'idée consiste à procéder à un traitement séquentiel de l'histogramme en utilisant qu'un nombre minimum de codes alors que la technique classique ne peut commencer les calculs qu'après avoir acquis la totalité des codes. Finalement, à chaque étape de la procédure de test seul le nombre d'apparition d'un code est calculé, les ressources mémoire requises ne concernent donc que le code en cours. Dans la section suivante, nous évaluerons le nombre de

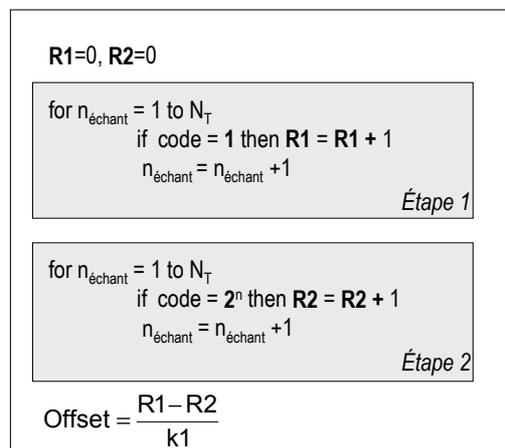
mots mémoire nécessaires au calcul de l'ensemble des caractéristiques fonctionnelles du CAN au cours du test complet.

b) Procédure d'évaluation de l'erreur d'offset

Nous avons démontré à la section III qu'indépendamment du type de signal d'entrée, l'erreur d'offset peut être évaluée à l'aide d'une expression de la forme suivante :

$$\text{Offset} = \frac{(H^{\text{exp}}(1) - H^{\text{exp}}(2^n))}{K} \quad (2.23)$$

Pour pouvoir évaluer plus facilement les ressources nécessaires à l'extraction de l'erreur d'offset, nous allons étudier l'algorithme suivant qui détaille la procédure de calcul de ce paramètre en appliquant le principe de décomposition temporelle.



Algorithme 2.1 : Procédure de calcul de l'offset

Pour chaque étape du processus de test, une trame complète du signal d'entrée (N_T échantillons) est utilisée. Dans le cas de l'erreur d'offset, deux étapes de calcul sont nécessaires. Dans la première étape, le registre R1 est incrémenté lorsque le code de sortie du convertisseur est égal au code 1 (premier code). Dans la deuxième étape, le registre R2 est incrémenté lorsque le code de sortie du convertisseur est égal au code 2^n (dernier code). A la fin des deux étapes, le registre R1 contient la valeur de $H^{\text{exp}}(1)$ et le registre R2 contient la valeur de $H^{\text{exp}}(2^n)$. Ainsi, la soustraction des valeurs stockées dans ces 2 registres est égale à la différence entre les fréquences d'apparition des deux codes extrêmes. Pour obtenir la valeur de l'offset, il suffit ensuite de diviser la valeur obtenue par une constante K prédéterminée.

Dans le cas d'un signal d'entrée linéaire cette constante est égale à $2.H_{id\acute{e}ale}$ et dans le cas d'un

$$\text{signal d'entrée sinusoïdal } K = \left(\frac{\pi.PE}{4.N_T} \times \left[\tan \frac{\pi.H^{réf}(1)}{N_T} \right] \right)^{-1}.$$

Par conséquent, les ressources nécessaires à l'évaluation de l'offset sont un additionneur, un soustracteur et un diviseur pour effectuer les calculs d'exploitation et deux registres R1 et R2 pour le stockage des valeurs de l'histogramme expérimental.

c) Procédure d'évaluation de l'erreur de gain

A la section III, nous avons montré que l'erreur de gain peut être évaluée à l'aide d'une expression dont la forme ne dépend pas du type de signal d'entrée utilisé :

$$\text{Erreur_de_Gain}_{(PE)} = \left(\frac{1}{m} \times \sum_{i=N1}^{N2} \frac{H^{exp}(i)}{H^{réf}(i)} - 1 \right) \quad (2.27)$$

$$\text{avec } N1 = \frac{2^n - m}{2} + 1 \text{ et } N2 = \frac{2^n + m}{2}$$

A noter que dans le cas d'un signal d'entrée triangulaire, la valeur de la fréquence d'apparition idéale d'un code i ($H^{réf}(i)$) est une constante ($H^{réf}(i) = H_{id\acute{e}ale}$).

L'algorithme suivant décrit les différentes étapes nécessaires à l'évaluation de l'erreur de gain en appliquant le principe de décomposition temporelle.

<pre> R2 = 0 for i = N1 to N2 R1 = 0 for n_échant = 1 to N_T if code = i then R1 = R1 + 1 n_échant = n_échant + 1 R1 = R1 / R^réf R2 = R2 + R1 </pre> <p style="text-align: right; margin-right: 20px;"><i>Étape i</i></p> $\text{Erreur_de_Gain} = \frac{R2 - 1}{m}$

Algorithme 2.2 : Procédure de calcul de l'erreur du gain

Le calcul est effectué séquentiellement, code après code. Pour un code donné, une trame complète (N_T échantillons) est utilisée. Par conséquent, m trames d'entrée sont nécessaires pour effectuer cette mesure. Tout d'abord, pour chaque code i , le registre R1 est incrémenté

chaque fois que le code de sortie est égal à i . Ainsi, à la fin de la trame correspondante, le registre R1 contient la valeur de la fréquence d'apparition du code i : $H^{\text{exp}}(i)$. Nous supposons que la valeur de référence correspondante $H^{\text{réf}}(i)$ est disponible au même moment dans un registre dédié $R^{\text{réf}}$. Nous pouvons alors diviser R1 par cette valeur de référence et stocker le résultat dans R1. En fait, en répétant ces opérations pour i variant de la valeur $N1 = \left(\frac{2^n - m}{2} + 1\right)$ à $N2 = \frac{2^n + m}{2}$, la somme cumulative $H^{\text{réf}}(i)/H^{\text{exp}}(i)$ est progressivement calculée dans R2. Après avoir soustrait 1, la valeur du gain est donc disponible dans le registre R2 à la fin des m trames du signal d'entrée.

Concernant les ressources nécessaires pour extraire ce paramètre, les différentes opérations utilisées sont l'addition, la division par une constante et une soustraction par 1. Du point de vue mémoire, deux registres sont nécessaires. Il faut noter que ces ressources sont identiques à celles utilisées lors de la phase de détermination de l'erreur d'offset. Par conséquent, grâce à la décomposition temporelle, les mêmes ressources peuvent être utilisées dans ces deux phases.

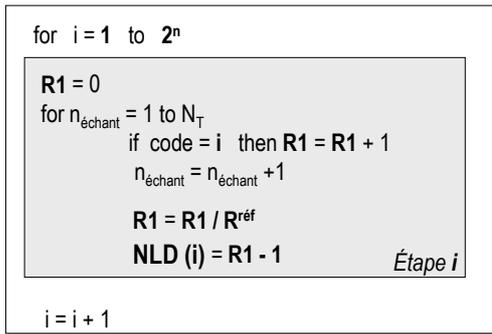
d) Procédure d'évaluation des non-linéarités

Nous avons défini les Non-Linéarités Différentielles et les Non-Linéarités Intégrales par des expressions de la forme suivante :

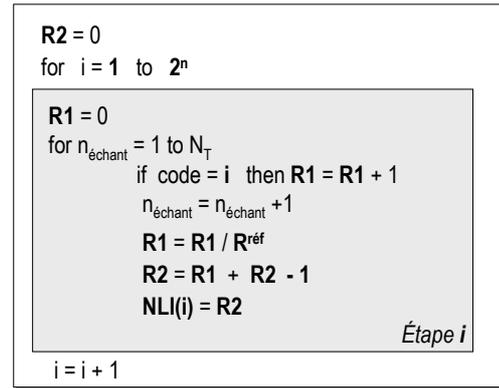
$$\text{NLD}(i) = \frac{H^{\text{exp}}(i) - H^{\text{réf}}(i)}{H^{\text{réf}}(i)} = \frac{H^{\text{exp}}(i)}{H^{\text{réf}}(i)} - 1 \quad (2.28) \quad \text{NLI}(i) = \sum_{j=1}^i \text{NLD}(j) \quad (2.29)$$

Comme pour le calcul de l'erreur de gain, la principale différence entre les calculs des non-linéarités effectués pour un signal d'entrée triangulaire ou sinusoïdal concerne la valeur de $H^{\text{exp}}(i)$ qui est constante dans le cas d'un signal d'entrée triangulaire ($H^{\text{réf}}(i) = H_{\text{idéale}}$).

Les algorithmes suivants détaillent les procédures de calcul des non-linéarités du convertisseur sous test.



a) Calcul des NLD



b) Calcul des NLI

Algorithme 2.3 : Procédures de calcul des Non-Linearités

Les NLD et les NLI doivent être calculées pour tous les codes de sortie du convertisseur. Par conséquent, 2^n trames sont nécessaires pour un convertisseur de n bits. Pour chaque code i , le registre $R1$ est d'abord initialisé à 0, puis incrémenté chaque fois que le code de sortie du convertisseur est égal à i . A la fin de la trame correspondante, $R1$ contient la valeur de l'histogramme expérimental $H^{\text{exp}}(i)$ pour le code i . Ensuite, la division du résultat par $R^{\text{réf}}$ qui contient la valeur de $H^{\text{réf}}(i)$ et la soustraction par 1 permettent d'obtenir la Non-Linearité Différentielle du code i . La procédure de calcul des NLI est identique à celle du calcul des NLD, à la différence que les NLD des codes précédents sont progressivement accumulés dans le registre $R2$.

Concernant les ressources nécessaires à l'évaluation de ces paramètres, deux registres sont utilisés et nous retrouvons les mêmes opérateurs (addition, soustraction, et division) déjà utilisés pour l'évaluation de l'offset et de l'erreur de gain. Finalement, ces deux registres et ces opérateurs élémentaires permettent donc d'implanter l'ensemble des calculs nécessaires à la détermination des paramètres fonctionnels.

IV.2. Stockage de l'histogramme Idéal

Précédemment, nous avons proposé des simplifications des expressions d'évaluation des paramètres fonctionnels du convertisseur et des solutions d'optimisation de la mémoire nécessaire au stockage de l'histogramme expérimental. Dans ce paragraphe, nous proposons des solutions d'optimisation de la mémoire nécessaire au stockage de l'histogramme de référence. Habituellement, l'histogramme de référence est entièrement stocké sur la puce avant toute opération de traitement.

Dans le cas d'un signal d'entrée triangulaire, l'histogramme de référence ne comporte que deux valeurs distinctes $H_{\text{idéale}}$ et $H_{\text{extrême}}$. La mémoire nécessaire au stockage de l'histogramme de référence pour ce type de signal d'entrée ne nécessite donc que deux registres.

Dans le cas d'un signal d'entrée sinusoïdal, l'histogramme étant différent pour chaque code, il est nécessaire de stocker l'ensemble des 2^n valeurs de l'histogramme de référence. Mais, comme l'histogramme de référence est symétrique, la première minimisation évidente de la mémoire consiste à ne stocker que la moitié de l'histogramme. Cependant, une mémoire de 2^{n-1} mots reste beaucoup trop importante dans le cadre d'une structure BIST.

Pour minimiser cette mémoire nous proposons de remplacer le stockage de l'histogramme complet par un calcul de ces valeurs directement sur la puce. Cette solution est rendue possible par l'utilisation de la technique de décomposition temporelle. Grâce à cette décomposition séquentielle du test, dans chaque étape élémentaire de la procédure de test globale une seule valeur de l'histogramme de référence est nécessaire. L'idée consiste donc à calculer cette valeur de référence dans l'étape considérée et la stocker dans une mémoire réutilisable pour les étapes suivantes de la procédure de test.

L'idée bien qu'attrayante présente une certaine difficulté puisque l'expression de l'histogramme de référence rappelée ci-dessous est relativement complexe, et surtout comporte des fonctions trigonométriques dont l'intégration représente une surface beaucoup trop importante dans une optique de test intégré.

$$H^{\text{réf}}(i) = \frac{N_T}{\pi} \left\{ \arcsin \left[\left(\frac{2i - 2^n}{2^n} \right) \cdot \frac{PE}{A_{\text{in}}} \right] - \arcsin \left[\left(\frac{2i - 2^n - 2}{2^n} \right) \cdot \frac{PE}{A_{\text{in}}} \right] \right\} \quad (1.25)$$

Par conséquent, nous proposons d'approximer cette fonction par une fonction plus simple et surtout plus facilement intégrable. Bien sûr, nous pourrions utiliser une approximation classique par un polynôme, mais cette solution n'est pas optimale en terme de surface puisqu'elle nécessite l'ajout d'un multiplieur dans la structure BIST. Il est plus intéressant d'estimer l'histogramme de référence à l'aide d'une fonction qui n'utilise que des opérateurs déjà présents sur la puce. Ainsi, l'additionneur et le diviseur étant déjà nécessaires pour l'évaluation des caractéristiques fonctionnelles du CAN, nous pouvons les réutiliser pour calculer la fonction d'estimation suivante :

$$H^{est}(i) = \frac{\alpha}{i + \beta} \quad (2.30)$$

où α et β sont des coefficients prédéterminés.

Pour améliorer la précision de l'estimation de l'histogramme, nous avons utilisé cette expression simple dans le cadre d'une approximation par morceaux sur la moitié des codes de l'histogramme (figure 2.11). Nous obtenons alors un ensemble de domaines, et dans chaque domaine les coefficients α_k et β_k sont déterminés pour obtenir la meilleure estimation possible. Le nombre de domaines à considérer dépend directement de la précision désirée pour l'histogramme estimé.

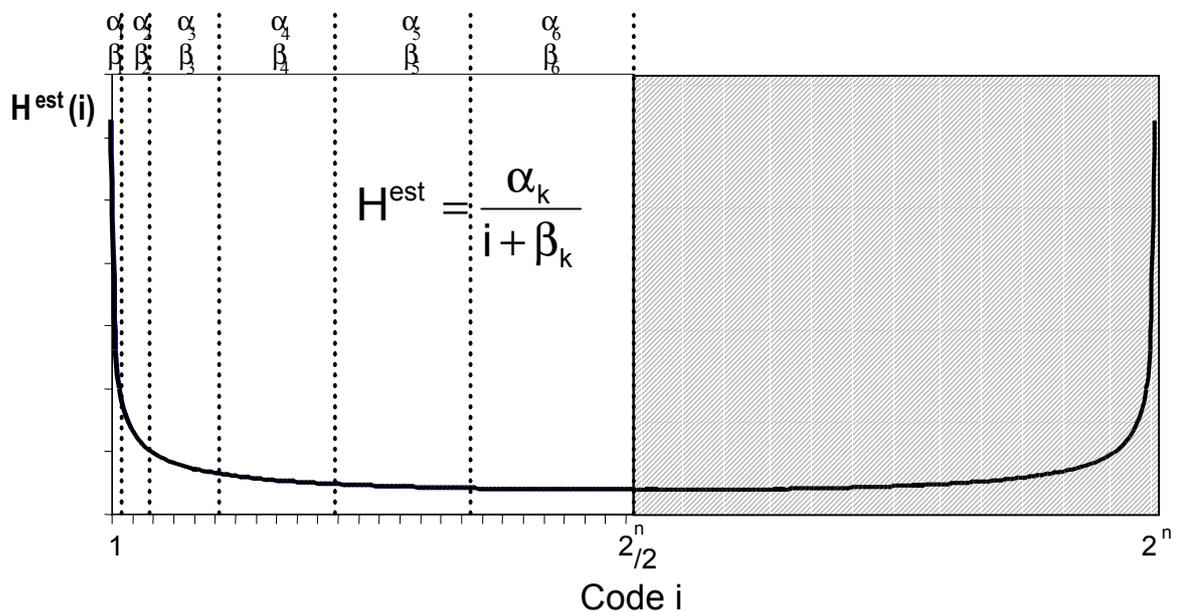


Figure 2.11 : Approximation par morceaux de l'histogramme de référence

Pour effectuer cette estimation de l'histogramme de référence, nous avons développé un programme en MATLAB[®] qui permet de calculer la taille des différents domaines et la valeur des coefficients associés pour une précision imposée sur la mesure des Non-Linéarités Différentielles.

Par exemple, pour une erreur maximale de 0,05 LSB sur les mesures des NLD dans le cas d'un convertisseur de 8 bits, seuls 6 domaines sont nécessaires pour une estimation valable de l'histogramme de référence.

Concernant les ressources nécessaires, nous avons vu que les opérateurs utilisés dans l'estimation de l'histogramme étaient déjà présents dans la structure BIST initiale. Par

conséquent, seuls quelques registres supplémentaires seront nécessaires pour estimer l'histogramme de référence. Plus précisément, pour chaque domaine k nous avons besoin de deux registres pour stocker les valeurs de α_k et β_k , plus un registre $R^{\text{réf}}$ commun à tous les domaines pour stocker la valeur d'histogramme estimé.

Finalement, la RAM de 2^n mots mémoire nécessaire à la technique classique de test par histogramme a été remplacée ici par $2k+1$ registres. Ainsi, dans l'exemple décrit précédemment pour un convertisseur de 8 bits, le stockage de l'histogramme de référence nécessite seulement 13 mots mémoire au lieu des 256 mots mémoire initiaux.

IV.3. Récapitulatif

Pour les deux types de signaux d'entrée, nous avons simplifié les opérations d'extraction des paramètres et minimisé la mémoire nécessaire à l'intégration de la technique de test par histogramme. Dans le cas d'un signal d'entrée sinusoïdal, nous avons effectué plusieurs approximations pour obtenir les simplifications des expressions. La validation de ces approximations sera présentée au chapitre 3.

Concernant les ressources matérielles, deux registres sont nécessaires et suffisants pour évaluer chaque paramètre à l'aide d'opérations simples : addition, soustraction et division. En outre, le fait d'utiliser le principe de décomposition temporelle permet de réutiliser les mêmes ressources matérielles pour chaque phase et pour chaque étape élémentaire. Ce même principe de décomposition temporelle permet aussi de minimiser la mémoire nécessaire au stockage de l'histogramme de référence en procédant à une estimation séquentielle des différentes valeurs de référence.

En résumé, les optimisations proposées permettent de définir l'architecture haut niveau présentée figure 2.12 pour le module d'exploitation. Ce module est composé de quelques mots mémoire pour le stockage de l'histogramme de référence, deux mots mémoire pour le stockage de l'histogramme expérimental et des opérateurs simples pour effectuer l'ensemble des calculs d'extraction des paramètres.

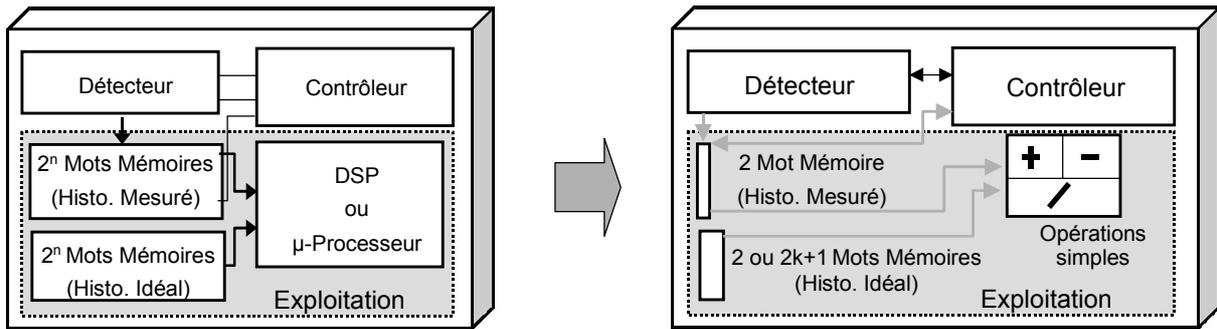


Figure 2.12 : Architecture optimale du module d'exploitation

V. Conclusion

Dans ce chapitre, nous avons proposé une architecture optimisée pour implanter un analyseur de réponses de test dédié à l'intégration de la technique de test par histogramme. En effet, l'intégration directe de cette technique n'est pas viable en terme de surface dans le cadre d'une structure BIST. Nous avons donc adapté et optimisé la technique initiale de test par histogramme pour réduire la surface nécessaire à son intégration.

Dans un premier temps, pour réduire le volume des données à traiter, nous avons étudié les deux types d'échantillonnage possibles : l'échantillonnage cohérent et l'échantillonnage aléatoire. L'échantillonnage cohérent permet de réduire considérablement le nombre d'échantillons à acquérir par rapport à un échantillonnage aléatoire. La surface de la structure BIST étant dépendante du volume de données à traiter, nous avons toujours travaillé par la suite dans le cas d'un l'échantillonnage cohérent.

Dans un deuxième temps, nous avons étudié les expressions permettant d'évaluer les différents paramètres fonctionnels. Nous avons constaté que ces expressions étaient relativement complexes et faisaient intervenir des fonctions trigonométriques difficiles à intégrer sur la puce sans une augmentation importante de la surface. Dans ce contexte, nous avons proposé des simplifications de ces expressions permettant l'évaluation des paramètres fonctionnels à l'aide d'opérateurs simples : addition, soustraction, division. Evidemment, l'intégration de ces opérateurs n'implique qu'une surface réduite de silicium.

Ensuite, nous avons présenté des solutions d'optimisation de la mémoire nécessaire au stockage de l'histogramme expérimental. Toutes ces solutions sont basées sur l'utilisation de

la technique de décomposition temporelle du test. Cette technique permet de limiter considérablement les ressources nécessaires à un instant donnée de la procédure de test. Ainsi, avec cette nouvelle technique de test, nous n'avons besoin que de deux mots mémoire pour stocker l'histogramme expérimental au lieu des 2^n mots mémoire nécessaires initialement.

Enfin, nous nous sommes proposé de réduire la mémoire nécessaire au stockage de l'histogramme de référence. Pour cela, nous avons proposé une nouvelle approche qui consiste à calculer l'histogramme code à code plutôt que de le stocker sur la puce. Dans le cas d'un signal triangulaire, la mémoire ne nécessite pas d'optimisation. En revanche, pour un signal sinusoïdal, l'expression de l'histogramme étant complexe, nous avons effectué des approximations à l'aide d'une fonction plus simple ne faisant intervenir que des opérateurs déjà présents dans la structure BIST.

Chapitre 3

Analyseur de réponses de test : validations

I. Introduction

Dans le chapitre 2, nous avons présenté les concepts mis en œuvre pour intégrer la technique classique de test par histogramme dans un analyseur de réponses de test optimisé. Les résultats obtenus ont permis de proposer une architecture haut niveau ne comportant que des opérateurs simples et quelques mots mémoire.

Dans ce chapitre, nous nous intéressons à l'implantation bas niveau de l'analyseur de réponses de test et présentons dans un premier temps, les résultats obtenus dans le cas d'un signal d'entrée triangulaire. Un exemple d'implantation physique sur silicium permet de valider la structure.

Dans un deuxième temps, nous présentons l'implantation bas niveau de l'analyseur de réponses de test associé à un signal d'entrée sinusoïdal en proposant une méthodologie de vérification de l'ensemble des approximations utilisées pour évaluer les paramètres fonctionnels.

La dernière partie de ce chapitre est consacrée à l'étude des performances et limitations de l'analyseur de réponses de test. Différentes possibilités d'optimisation du temps de test sont proposées et nous décrivons les étapes permettant d'automatiser l'élaboration de notre structure BIST pour une application donnée.

II. Analyseur de réponses pour signal d'entrée triangulaire

L'analyseur de réponses de test est composé de trois modules : un détecteur de code, un contrôleur et un module d'exploitation. Au chapitre 2, nous avons proposé une architecture haut niveau du module d'exploitation permettant de minimiser sa surface. Dans ce paragraphe, nous présentons l'architecture bas niveau de ces trois blocs et une implantation de la structure complète.

II.1. Architecture niveau porte

a) Détecteur de code

La technique de test que nous proposons est basée sur la technique classique de test par histogramme. La première phase de cette technique consiste à construire l'histogramme expérimental. Cet histogramme est une représentation de la fréquence d'apparition des différents codes de sortie du convertisseur. Dans ce contexte, le rôle du détecteur de code consiste à déterminer quel est le code de sortie du convertisseur à un instant donné et de vérifier s'il fait partie des codes à traiter.

Le principe de décomposition temporelle utilisé dans la solution BIST proposée implique le traitement séquentiel d'un seul code à la fois à chaque instant de la procédure de test. Ainsi, la fonction principale du détecteur de code consiste à positionner la valeur du code à traiter selon la phase de test considérée et à comparer le code en sortie du convertisseur avec ce code de référence.

Cette double fonction de mise en place du code de référence et comparaison de ce code avec le code de sortie du CAN a été intégrée dans une structure originale de registre *compteur-comparateur* présentée figure 3.1.

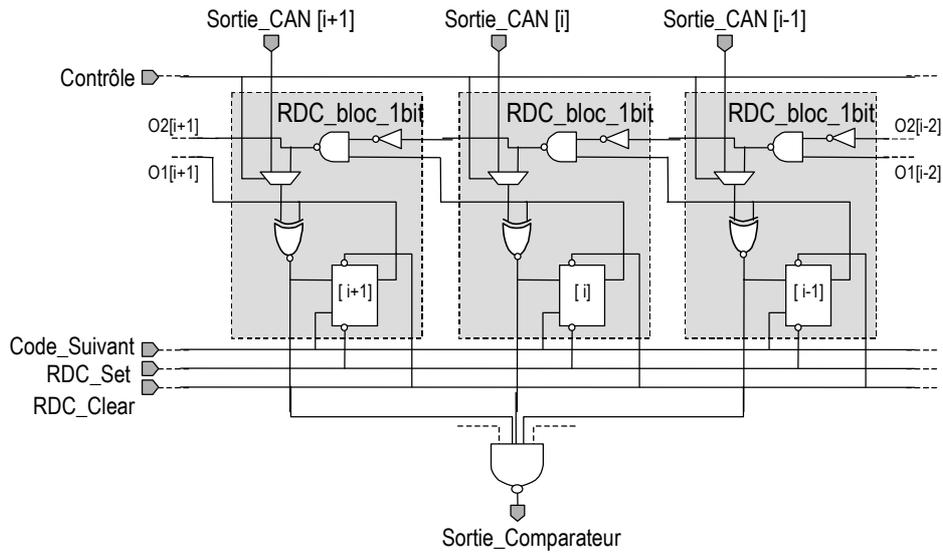


Figure 3.1 : Détecteur de code

Selon la valeur du signal *Contrôle*, le Registre Détecteur de Code (RDC) est configuré, soit en compteur, soit en comparateur. En mode compteur, le registre est incrémenté à chaque nouveau front du signal *Code_suivant*. Ainsi, au bout de k fronts successifs sur ce signal, le registre contient la valeur correspondant au code de référence k . En mode comparateur, le registre effectue la comparaison entre ce code de référence et le code de sortie du convertisseur *Sortie_CAN[1;n]*. Le signal *Sortie_Comparateur* prend alors la valeur logique "0" chaque fois que le code de sortie du convertisseur correspond au code de référence positionné dans le registre RDC.

Nous observons que la structure est composée d'un assemblage de blocs élémentaires identiques, notés *RDC_bloc_1_bit*. Comme chaque bloc analyse un seul bit de sortie du convertisseur, le nombre total de blocs nécessaires pour implanter le détecteur de code est égal au nombre de bits du CAN à tester.

b) Module d'exploitation

Le module d'exploitation est utilisé pour évaluer la valeur des différents paramètres fonctionnels du CAN sous test. Il est composé d'une partie mémoire et d'une partie opérative.

L'architecture haut niveau du module d'exploitation proposée au chapitre 2 comporte :

- 2 mots mémoire pour le stockage de l'histogramme expérimental,
- 2 mots mémoire pour le stockage de l'histogramme de référence,
- 1 additionneur,
- 1 soustracteur,
- 1 diviseur.

En fait, l'étude bas niveau de cette structure permet d'en optimiser encore l'architecture. Ainsi, dans cette section, nous allons détailler les procédures de calcul des différents paramètres et nous allons voir qu'il est possible d'intégrer toutes les fonctions nécessaires à l'extraction des paramètres dans un simple registre configurable, appelé Registre d'Exploitation (RE).

➤ *Erreur d'offset*

Nous avons montré au chapitre 2 que l'offset peut être déterminé par l'équation (2.10) rappelée ci-dessous :

$$\text{Offset} = \frac{H^{\text{exp}}(1) - H^{\text{exp}}(2^n)}{2 \cdot H_{\text{idéale}}} \quad (2.10)$$

Pour ce calcul, le module d'exploitation doit effectuer la soustraction entre les fréquences d'apparition des deux codes extrêmes, $H^{\text{exp}}(1)$ et $H^{\text{exp}}(2^n)$.

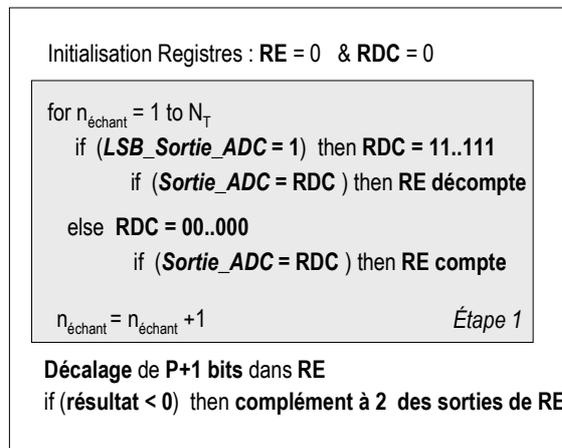
En pratique, cette opération est effectuée en configurant le registre RE en compteur-décompteur. Dans cette configuration, le registre RE est décrémenté si le code de sortie du convertisseur est égal au code 2^n , ou incrémenté si le code de sortie est égal au code 1. Ainsi, à la fin d'une trame complète du signal d'entrée, le registre contient la valeur de la différence entre les nombres d'apparition des deux codes extrêmes, $H^{\text{exp}}(1) - H^{\text{exp}}(2^n)$.

Après avoir effectué cette opération, il est nécessaire de diviser le résultat obtenu par $2 \times H_{\text{idéale}}$. L'intégration d'un diviseur classique impliquerait une augmentation substantielle de la surface de la structure. Or dans le contexte BIST de nos travaux, la surface est un critère extrêmement important. Nous proposons donc une alternative basée sur le fait que le nombre diviseur est une constante. L'idée consiste à utiliser une propriété particulière de la division entre nombres binaires. En effet, la division d'un nombre binaire par un nombre égal à une puissance de

deux revient à un simple décalage de bits. Ainsi, en choisissant $H_{idéale} = 2^P$, la division nécessaire au calcul de l'offset est équivalente à un décalage de P+1 bits du registre RE.

Finalement, le calcul de l'offset peut être simplement implanté par des opérations de comptage/décomptage, suivies d'un décalage de bits. Le résultat obtenu est alors un nombre binaire signé correspondant à l'offset mesuré. En pratique, toutes les caractéristiques fonctionnelles mesurées à l'aide de notre technique de test intégré sont spécifiées en terme de valeur absolue. Par conséquent pour faciliter l'exploitation de la mesure, nous prévoyons la possibilité de configurer le registre RE de manière à effectuer le complément à deux du résultat s'il est négatif.

L'algorithme suivant résume l'ensemble de la procédure de calcul de l'offset.



Algorithme 3.1 : Procédure de calcul de l'erreur d'offset

➤ **Erreur de gain**

L'erreur de gain est définie par l'expression (2.14) rappelée ci-dessous :

$$\text{Erreur_de_Gain} = \left(\frac{\sum_{i=N1}^{N2} H^{\text{exp}}(i)}{m \cdot H_{idéale}} - 1 \right) \tag{2.14}$$

Dans un premier temps, pour évaluer ce paramètre, il faut calculer la somme des fréquences d'apparition $H^{\text{exp}}(i)$ des m codes centraux de l'histogramme. Cette opération est effectuée en configurant le registre RE en compteur incrémenté à chaque fois que le code de sortie du CAN est égal au code référence. Une trame complète du signal d'entrée est utilisée pour

chaque code de référence compris entre les valeurs N1 et N2, avec $N1 = \frac{2^n - m}{2} + 1$ et $N2 = \frac{2^n + m}{2}$. Ainsi, à la fin des m trames de signal d'entrée, le registre RE contient la somme des

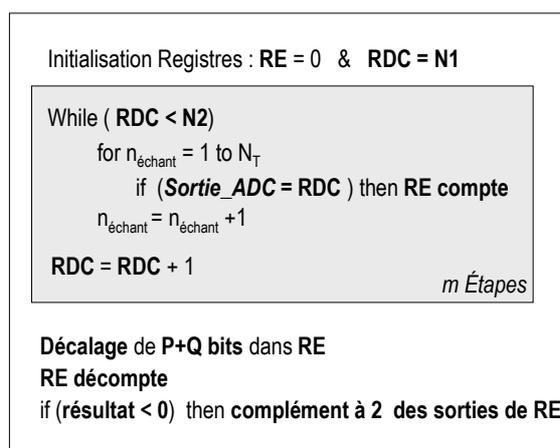
fréquences d'apparition des m codes centraux : $RE = \sum_{i=N1}^{N2} H^{exp}(i)$.

Ensuite, il est nécessaire de diviser le résultat obtenu par $m \times H_{idéale}$. Comme dans le cas du calcul de l'erreur d'offset, nous utilisons les propriétés de la division entre nombres binaires pour remplacer cette division difficilement intégrable par de simples décalages de bits. Pour cela, il est nécessaire d'imposer la valeur de $m \times H_{idéale}$ égale à une puissance de 2. Précédemment nous avons imposé $H_{idéale} = 2^P$, donc pour que l'élément diviseur ($m \times H_{idéale}$) soit égal à une puissance de deux, il faut que la valeur de m soit aussi égale à une puissance de deux : $m = 2^Q$. Ainsi, la division par $m \times H_{idéale}$ est équivalente à un simple décalage de P+Q bits du registre RE.

L'étape suivante consiste à soustraire 1 au résultat de la division précédente. Pour cela, il suffit de décrémenter le registre RE.

Enfin, si la valeur obtenue est négative, le registre RE est configuré pour effectuer le complément à deux de ses sorties.

L'algorithme suivant résume l'ensemble des étapes de la procédure de détermination de l'erreur de gain.



Algorithme 3.2 : Procédure de calcul de l'erreur de gain

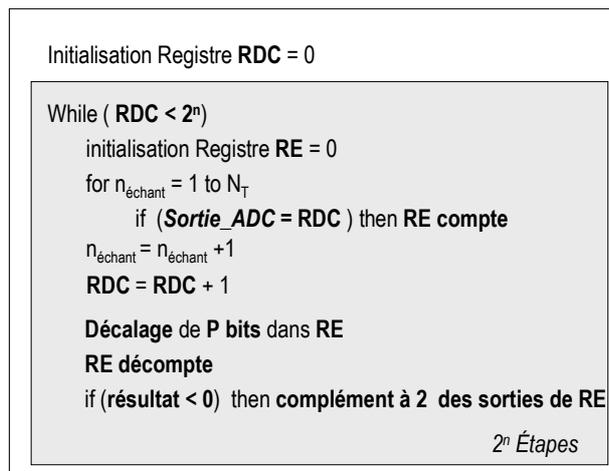
➤ **Non-Linéarités Différentielles et Non-Linéarités Intégrales**

Les Non-Linéarités Différentielles (NLD) sont définies par l'expression suivante :

$$\text{NLD}(i) = \frac{H^{\text{exp}}(i)}{H_{\text{idéale}}} - 1 \quad (2.15)$$

Ces non-linéarités doivent être évaluées pour chacun des 2^n codes du convertisseur, supposant l'application de 2^n trames de test. Pour chaque code i , le registre RE est d'abord initialisé à 0 puis le calcul de $H^{\text{exp}}(i)$ est effectué en configurant le registre RE en compteur incrémenté à chaque fois que le code i apparaît en sortie du convertisseur. Ainsi, à la fin de chaque trame, le registre RE contient la valeur de $H^{\text{exp}}(i)$. Comme $H_{\text{idéale}} = 2^P$, la division par $H_{\text{idéale}}$ revient à un décalage de P bits du registre RE et la soustraction par 1 est équivalente à une décrémentation de ce même registre. Si le résultat est négatif, le registre RE est configuré pour effectuer le complément à 2 de ses sorties.

L'algorithme suivant résume les différentes étapes du calcul des non-linéarités différentielles.



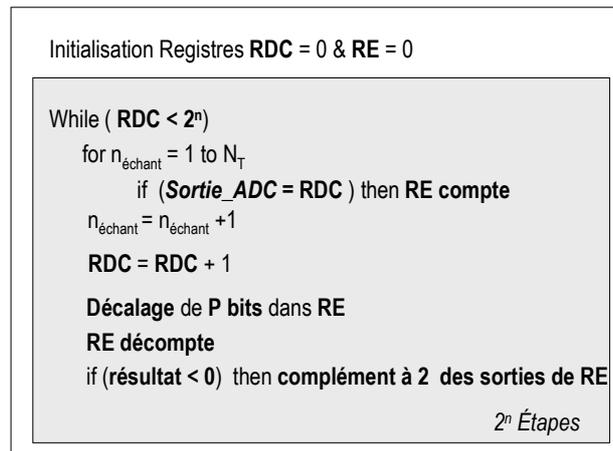
Algorithme 3.3 : Procédure de calcul des NLD

Enfin, la non-linéarité intégrale d'un code est définie comme la somme des non-linéarités différentielles des codes précédents :

$$\text{NLI}(i) = \sum_{j=1}^i \text{NLD}(i) \quad (2.16)$$

Par conséquent pour le calcul des NLI, on retrouve la même procédure que pour les NLD, à la différence que le registre RE n'est pas initialisé entre deux trames successives du signal d'entrée mais une seule fois au début de la procédure afin d'implanter la somme cumulative.

L'algorithme suivant permet de résumer l'ensemble de la procédure de calcul des non-linéarités intégrales.



Algorithme 3.4 : Procédure de calcul des NLI

➤ *Architecture finale*

Le détail des procédures d'évaluation des paramètres fonctionnels a permis de montrer qu'il est possible d'intégrer les ressources du module d'exploitation dans un simple registre. Ce registre est muni d'un bloc logique de contrôle qui permet de modifier sa fonctionnalité. Ainsi, la structure finale du module d'exploitation présentée figure 3.2 comporte deux entrées de contrôle *c1* et *c2* qui permettent de configurer le registre en compteur, en décompteur, ou enfin en module effectuant le complément à 2 de ses sorties.

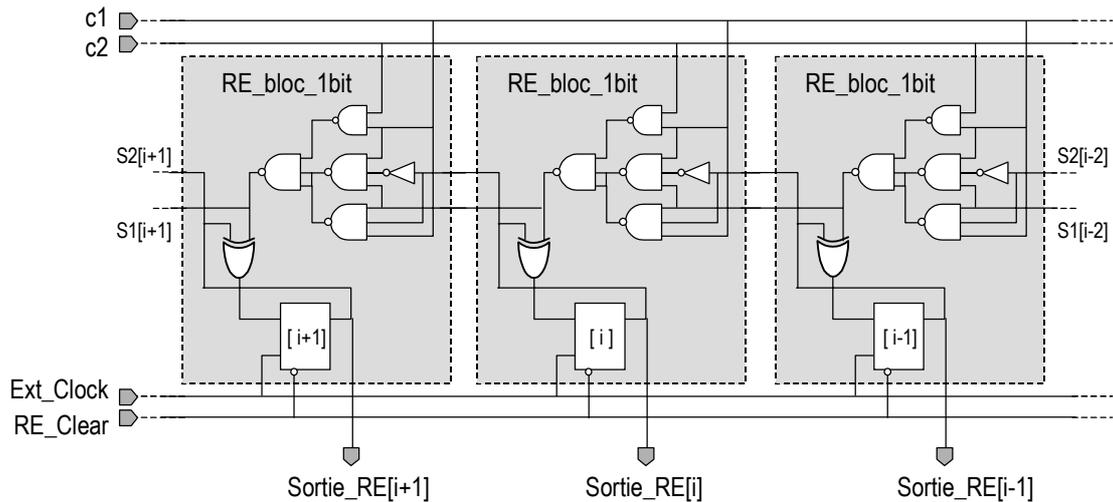


Figure 3.2 : Module d'exploitation

Comme dans le cas du registre détecteur de code, le module d'exploitation est composé d'une succession de blocs élémentaires, appelé *RE_bloc_1_bit*. Le nombre de blocs dépend du nombre de décalages de bits nécessaires à l'évaluation des paramètres fonctionnels.

Précédemment, nous avons vu que le calcul de l'erreur d'offset nécessite $P+1$ décalages de bits, que le calcul de l'erreur de gain nécessite $P+Q$ décalages de bits, et que le calcul des NLD et NLI nécessite P décalage de bits. Par conséquent, le nombre minimal de blocs élémentaires est égal à $P+Q+1$, où $P = \log_2(H_{id\acute{e}ale})$ et $Q = \log_2(m)$.

La valeur de $H_{id\acute{e}ale}$ dépend directement de la précision désirée sur la mesure des différents paramètres fonctionnels. Par exemple, la précision δ_{NLD} de la mesure des NLD peut être déduite facilement à partir de l'équation 1.2 présentée au chapitre 1. Nous obtenons alors l'expression suivante :

$$\delta_{NLD} = \frac{\delta H^{exp}(i)}{H_{id\acute{e}ale}} \quad (3.1)$$

où $\delta H^{exp}(i)$ représente l'erreur de mesure de la fréquence d'apparition du code i .

En pratique, l'erreur de mesure $\delta H^{exp}(i)$ est due au fait que la fréquence d'apparition $H^{exp}(i)$ ne peut être qu'un entier. Ainsi, sa valeur maximale étant égale à 1, la valeur de $H_{id\acute{e}ale}$ dépend uniquement de la précision sur la mesure des NLD :

$$H_{id\acute{e}ale} = \frac{1}{\delta_{NLD}} \quad (3.2)$$

Pour une application donnée où la précision sur la mesure de la NLD est imposée, nous pourrions donc déterminer la valeur de $H_{id\acute{e}ale}$ et en déduire la valeur de P .

Pour déterminer le nombre de blocs élémentaires, il faut aussi calculer le nombre m de codes à traiter pour effectuer la mesure de l'erreur de gain. Plus le nombre de codes utilisés est important et plus la mesure du gain est précise. En contrepartie, plus ce nombre est élevé, plus la surface du module d'exploitation augmente.

Il est difficile d'évaluer précisément la valeur de m pour une application donnée. En pratique, partant des nombreux résultats de simulation, nous considérerons que traiter un quart des codes possibles pour l'évaluation du gain permet d'obtenir une précision acceptable :

$$m = \frac{2^n}{4} \text{ soit } Q = n - 2 \quad (3.3)$$

où n correspond au nombre de bits du convertisseur.

De ce fait, le nombre de blocs élémentaires constituant le module d'exploitation est donné par l'expression suivante :

$$\text{Nombre de } RE_Bloc_Ibit \geq n - 1 - \log_2(\delta_{NLD}) \quad (3.4)$$

Par exemple, pour un convertisseur de 6 bits et une précision sur la mesure des non-linéarités égale $0,05\text{LSB}$, la valeur de δ_{NLD} impose $H_{id\acute{e}ale} \geq 20$ soit $P = 5$ et le fait de mesurer le gain sur $2^n/4$ codes impose $Q = 4$. Donc au final, le nombre de blocs élémentaires est égal à $P + Q + 1 = 10$.

A ce stade, nous pouvons remarquer que la mémoire initialement nécessaire au stockage de l'histogramme de référence n'est plus utile à présent. En effet, le fait que $H_{id\acute{e}ale}$ soit égale à une puissance de 2 ($H_{id\acute{e}ale} = 2^P$) rend implicite son stockage dans le $P^{\text{ème}}$ bit du registre d'exploitation.

En conclusion, toutes les ressources nécessaires à l'évaluation des paramètres fonctionnels d'un convertisseur sont intégrables dans un simple registre configurable.

c) Contrôleur

Le rôle du contrôleur consiste à gérer l'ensemble des étapes de la procédure de test. Il doit effectuer chaque phase de test dans l'ordre présenté figure 3.3.

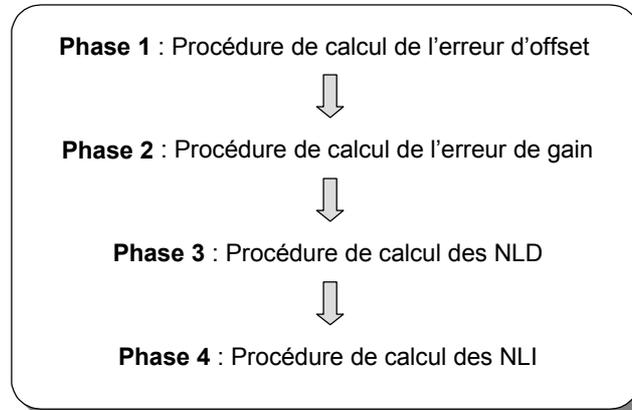


Figure 3.3 : Phases du test

La procédure de test a fait l'objet d'une description haut niveau écrite en VHDL. Partant du programme VHDL, donné en annexe 2, nous avons développé la structure niveau porte (figure 3.4) à l'aide des outils de synthèse automatique (Synopsys®).

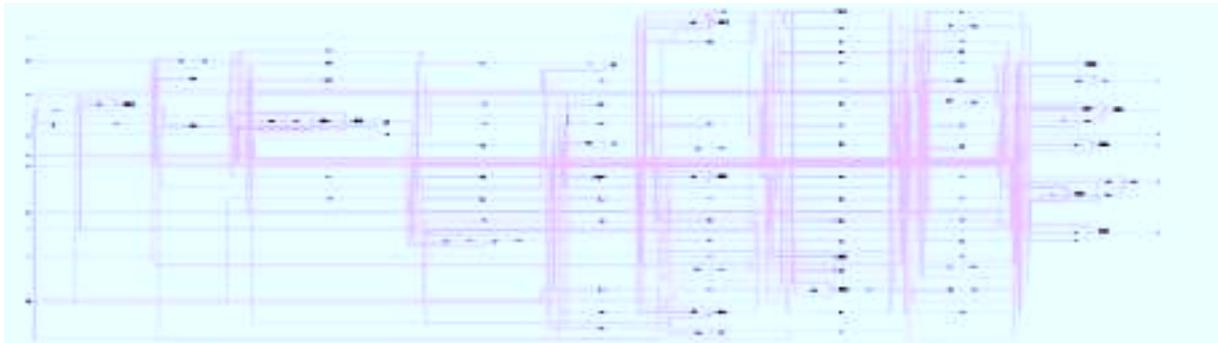


Figure 3.4 : Schéma niveau porte du contrôleur

d) Simulations

Nous avons simulé la structure finale de l'analyseur de réponses de test. Pour illustrer les résultats obtenus, nous présentons dans cette section un exemple de mesure d'un paramètre fonctionnel à l'aide de l'analyseur de réponses de test.

Les deux diagrammes suivants représentent respectivement le début et la fin de la phase d'évaluation de l'erreur d'offset. Sur la figure 3.5, nous pouvons observer que le registre RE est incrémenté si le code de sortie du convertisseur *Sortie_CAN* est égal au code 00...000.

Ainsi sur l'ensemble de la trame du signal d'entrée, il est possible de calculer la valeur de la fréquence d'apparition du code 00...000 (code 1). Dans l'exemple considéré, le code 1 est apparu 39 fois : $H^{\text{exp}}(1) = 39$.

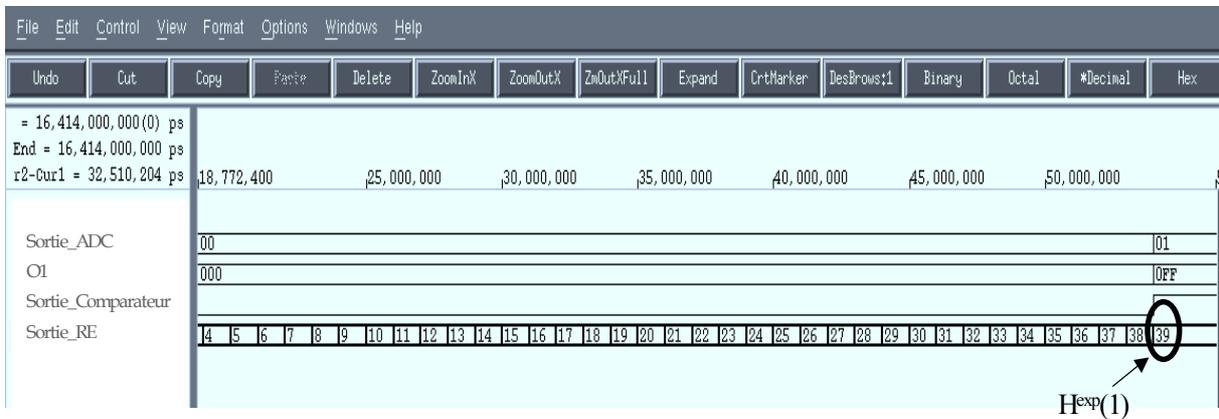


Figure 3.5 : Début de la phase de détermination de l'erreur d'offset

Sur la figure 3.6, nous observons que le registre est décrémenté à chaque fois que le code de sortie du CAN est égal à 1111.111 (code 2^n). Ainsi, à la fin de la trame du signal d'entrée, le registre RE contient la valeur de la différence entre les fréquences d'apparition des deux codes extrêmes : $H^{\text{exp}}(1) - H^{\text{exp}}(2^n) = 14$.

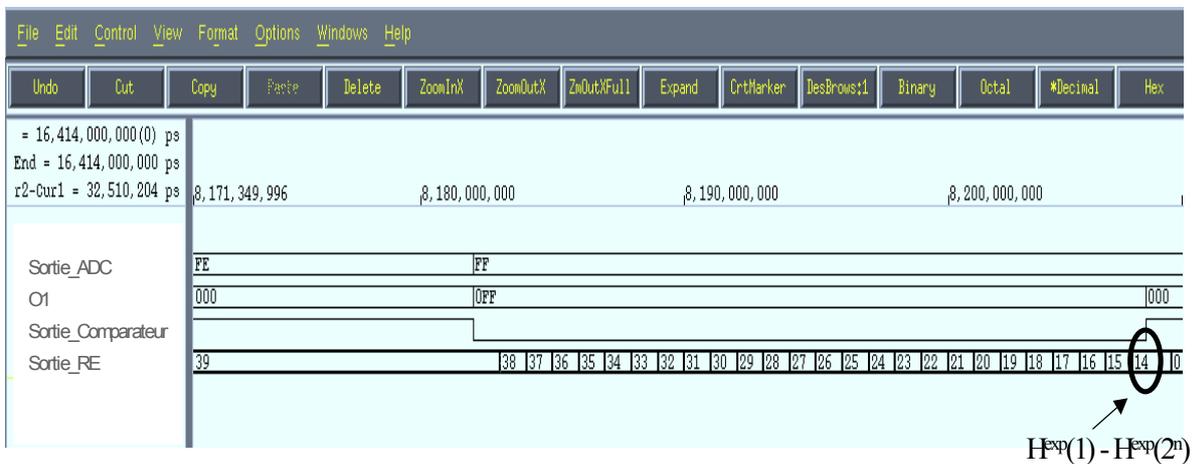


Figure 3.6 : Fin de la phase de détermination de l'erreur d'offset

Plus généralement, pour valider la structure de l'analyseur de réponses de test, nous avons effectué un grand nombre de simulations en injectant diverses erreurs dans le convertisseur sous test. A titre d'illustration, le tableau 4.1 compare les valeurs mesurées par l'analyseur (données en binaire colonne 2 et en décimal colonne 3) aux valeurs injectées dans le

convertisseur (données colonne 1) pour les différents paramètres fonctionnels. Ces simulations ont été effectuées pour un convertisseur de 6 bits et en imposant une précision sur la mesure des NLD supérieure à 0,05 LSB. Dans ce cas particulier, nous avons vu à la section précédente que le module d'exploitation de l'analyseur de réponses de test est composé de 10 blocs élémentaires de 1 bit.

		Valeur	Valeur mesurée (binaire)	Valeur mesurée (décimal)	Erreur de mesure
Paramètres fonctionnels	Erreur d'Offset (LSB)	-0,75	0000,110000	0,75	0,00
		-0,20	0000,001100	0,19	0,01
		1,10	0001,000110	1,09	0,01
		1,40	0001,011001	1,39	0,01
	Erreur de Gain (PE)	0,10	0,000110011	0,10	0,00
		-0,05	0,000011001	0,05	0,00
		0,04	0,000010100	0,04	0,00
		0,02	0,000001010	0,02	0,00
	NLD (LSB)	-0,55	00000,10001	0,53	0,02
		-0,30	00000,01001	0,28	0,02
		0,50	00000,10000	0,50	0,00
		0,95	00000,11110	0,94	0,01
	NLI (LSB)	-0,55	00000,10001	0,53	0,02
		-0,30	00000,10001	0,28	0,02
		1,00	00001,00000	1,00	0,00
		0,40	00000,01100	0,37	0,03

Tableau 3.1 : Résultats de simulation

Sur ce tableau, nous pouvons observer les différents décalages de bits (colonne 2) selon le paramètre fonctionnel à déterminer. Nous avons un décalage de $P+1 = 6$ bits pour la mesure de l'erreur d'offset, un décalage de $P + Q = 9$ bits pour la mesure de l'erreur de gain et un décalage de $P = 5$ bits pour la mesure des non-linéarités. Nous observons aussi que plus le nombre de décalages est grand, plus la précision de mesure est importante. Ainsi concernant la mesure du gain, nous obtenons une erreur nulle (colonne 4) entre la valeur mesurée par l'analyseur et la valeur injectée dans le convertisseur pour la simulation. Concernant les autres paramètres, nous constatons que les erreurs de mesure respectent la contrainte de précision imposée. Pour toutes les simulations présentées dans ce tableau, l'erreur maximale de mesure est égale à 0,03 LSB, ce qui est donc bien inférieur à la limite permise de 0,05 LSB.

II.2. Implantation physique

Afin de valider la structure finale, nous avons effectué son implantation en technologie AMS 0,6 micron. Nous avons vu précédemment que le nombre de blocs élémentaires constituant le détecteur de code et le registre d'exploitation dépend directement de l'application envisagée. Pour illustrer la solution proposée, nous considérons l'exemple d'une structure BIST permettant de tester les caractéristiques fonctionnelles réalistes suivantes :

- nombre de bits du convertisseur $n = 6$,
- tolérance sur l'erreur d'offset = $\pm 1,5$ LSB,
- tolérance sur l'erreur de Gain = $\pm 0,2$ % de PE,
- tolérance sur les NLD = $\pm 0,5$ LSB,
- précision sur la mesure des NLD : $\delta_{\text{NLD}} = 0,05$ LSB,
- tolérance sur les NLI = ± 1 LSB.

Comme nous l'avons vu précédemment, pour ce cahier des charges, la structure BIST est constituée de 6 blocs élémentaires pour le détecteur de code et de 10 blocs élémentaires pour le module d'exploitation.

La figure 3.7 représente le circuit obtenu au niveau physique (layout).

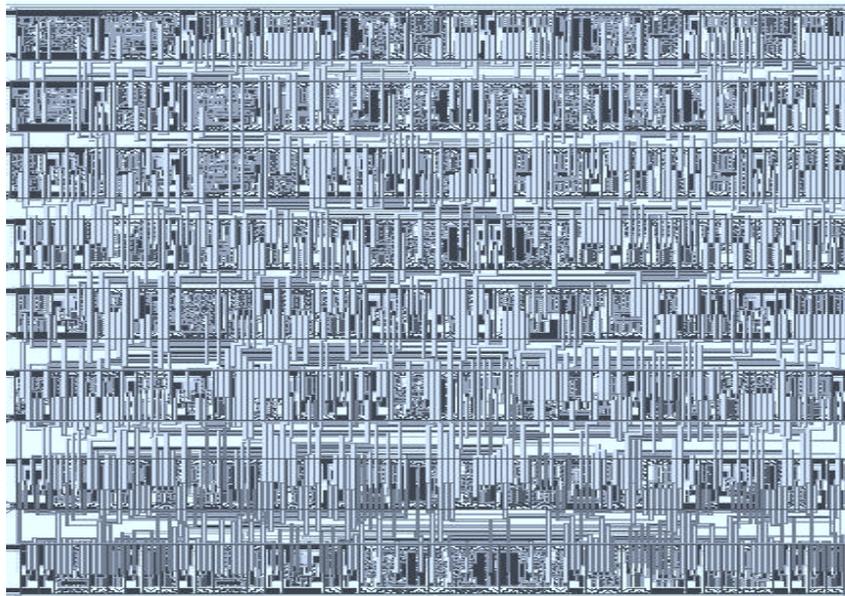


Figure 3.7 : Implantation physique de l'analyseur de réponses de test

Avant de présenter les résultats de simulations obtenus, nous devons étudier l'une des caractéristiques les plus importantes d'une structure BIST : sa surface par rapport au composant à tester.

a) Surface

Pour le cahier des charges considéré, l'analyseur de réponses de test n'est constitué que d'une quinzaine de bascules et de quelques portes logiques. La surface totale engendrée par son intégration est par conséquent réduite : elle est inférieure à 0,116 mm².

Le tableau suivant donne la surface de chaque bloc de l'analyseur de réponses de test et permet de comparer la surface de la structure complète avec la surface d'un CAN de 6 bits et de fréquence d'échantillonnage 100MHz de la bibliothèque AMS 0,6 micron.

	Surface d'un bloc élémentaire	Surface du bloc	Surface de l'Analyseur de réponses de test	Surface du CAN	Surface $\frac{\text{Analyseur}}{\text{CAN}}$
Détecteur de code	0,00377 mm ²	0,023mm ²	0,116 mm ²	3,3 mm ²	4,1 %
Module d'exploitation	0,0041 mm ²	0,041mm ²			
Contrôleur	0,052 mm ²				

Tableau 3.2 : Surface de l'analyseur de réponses de test

La surface de l'analyseur de réponses de test ne représente que 4,1% de la surface du convertisseur 6 bits présent dans la bibliothèque de la même technologie. A ce stade de l'étude, il s'agit d'un résultat très encourageant qui nous laisse espérer pouvoir développer une solution BIST complète incluant non seulement l'analyseur de réponses de test mais aussi le générateur de stimuli, avec une surface additionnelle inférieure à 10 %.

b) Validation expérimentale

Le circuit obtenu a été testé à son retour de fonderie à l'aide du testeur industriel HP83000. Pour illustration, la figure 3.8 représente le diagramme obtenu à la fin de la phase de mesure de l'erreur d'offset. Cette mesure étant effectuée dans les mêmes conditions que pour l'exemple donné au paragraphe I.1, nous pouvons aisément remarquer la parfaite cohérence du résultat.

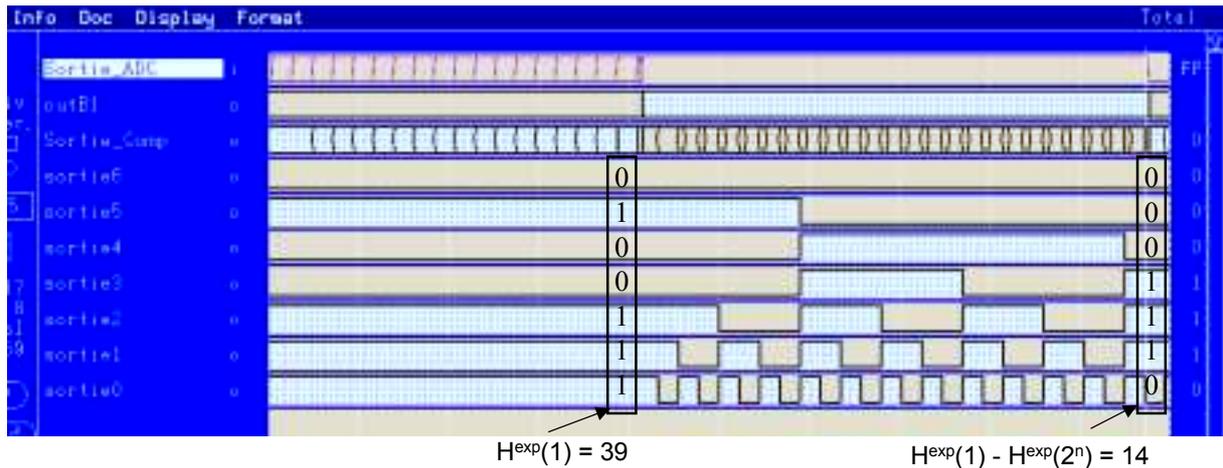


Figure 3.8 : Simulation après fabrication

Nous ne présenterons pas ici le tableau récapitulatif des différentes mesures de paramètres fonctionnels car il est totalement identique au tableau 3.2 présenté au paragraphe I.1.

Ainsi, cette implantation nous permet de valider notre solution d'analyseur intégré de réponses de test dans le cas d'un signal d'entrée triangulaire.

III. Analyseur de réponses pour signal d'entrée sinusoïdal

L'architecture haut niveau développée pour l'analyseur de réponses de test dans le cas d'un signal d'entrée sinusoïdal s'appuie sur un certain nombre d'approximations pour l'évaluation des paramètres fonctionnels. La première partie de ce paragraphe est par conséquent consacrée à la validation de ces approximations, avant de détailler dans une deuxième partie l'implantation niveau porte de l'analyseur.

III.1. Validation des approximations

Pour simplifier les expressions des paramètres fonctionnels dans le cas d'un signal d'entrée sinusoïdal, plusieurs approximations ont été développées dans le chapitre 2.

Les premières de ces approximations interviennent dans le calcul de l'erreur d'offset. En effet, rappelons que nous avons considéré que la somme des fréquences d'apparition des deux codes extrêmes était égale à une constante ($H^{\text{exp}}(1) + H^{\text{exp}}(2^n) = \text{constante}$) et nous avons remplacé

$\text{Tan}\left(\frac{\pi.H^{\text{exp}}(1)}{N_T}\right)$ par $\left(\frac{\pi.H^{\text{exp}}(1)}{N_T}\right)$. Concernant le calcul de l'erreur de gain, nous avons là encore introduit une approximation. En effet, l'erreur de gain est estimée à partir des m codes centraux en utilisant l'expression développée pour l'histogramme linéaire, ce qui suppose que $H^{\text{exp}}(i)$ soit constant pour chacun des m codes. Finalement, la dernière approximation introduite concerne le calcul de l'histogramme de référence puisque la fonction originale complexe est remplacée par une approximation par morceaux à l'aide d'une fonction simple.

Durant les différentes étapes de calcul, ces diverses approximations ont été vérifiées de manière isolée. Or en pratique, toutes ces approximations sont utilisées simultanément et il est donc indispensable d'évaluer l'influence de l'ensemble des approximations sur l'évaluation des paramètres fonctionnels.

a) Programme d'évaluation

Pour effectuer la validation de ces approximations, nous avons développé un programme d'évaluation qui permet de comparer notre technique avec la technique classique de test par histogramme. Plus précisément, nous avons implanté dans ce programme, écrit en LABVIEW[®], deux algorithmes permettant de réaliser le test par histogramme. Le premier algorithme utilise les expressions originales définies dans [Mah87] pour la définition de l'histogramme de référence et l'évaluation des paramètres fonctionnels. Il s'agit donc de l'approche classique du test par histogramme. Le deuxième algorithme utilise les différentes approximations développées dans le cadre de notre solution BIST, à la fois pour le calcul de l'histogramme de référence et l'évaluation des paramètres. La comparaison des résultats fournis par les deux algorithmes permet donc de vérifier la validité de nos approximations. La figure 3.9 représente la vue conceptuelle du programme d'évaluation proposé.

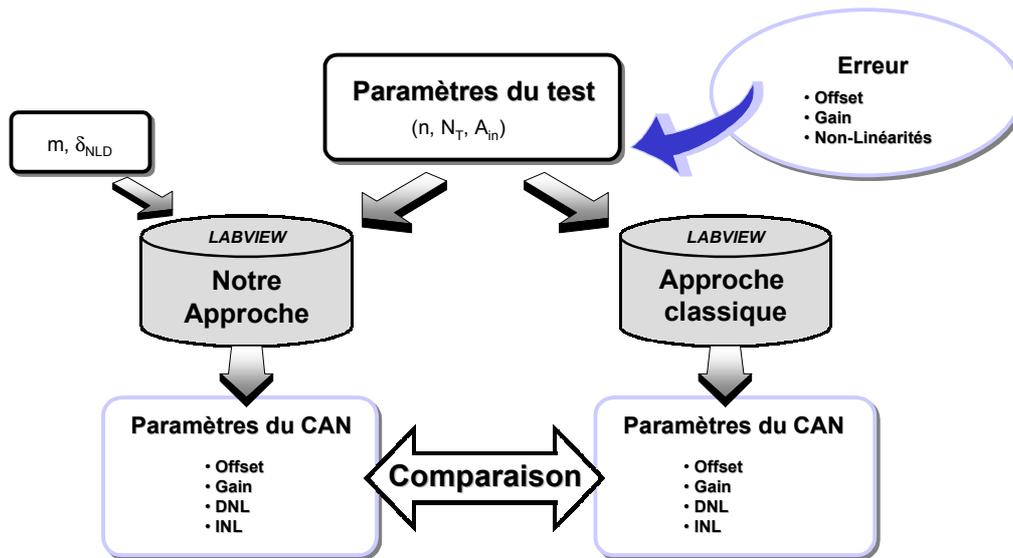


Figure 3.9 : Programme d'évaluation

Ce programme comporte deux types d'entrées. D'une part, les entrées permettant de définir le modèle du convertisseur utilisé : nombre de bits, erreur d'offset, erreur de gain ... D'autre part, les entrées dédiées à la définition des paramètres spécifiques de la procédure de test : nombre d'échantillons à traiter, choix du signal d'entrée et de son amplitude... En plus des paramètres classiques utilisés pour le test par histogramme, le programme permet d'entrer les paramètres spécifiques à notre technique de test : la précision désirée sur la mesure des NLD et le nombre de codes traités pour le calcul du gain.

La figure 3.10 représente l'interface utilisateur du programme avec les deux types d'entrées sur la gauche, les résultats graphiques au centre et les résultats numériques sur la droite.

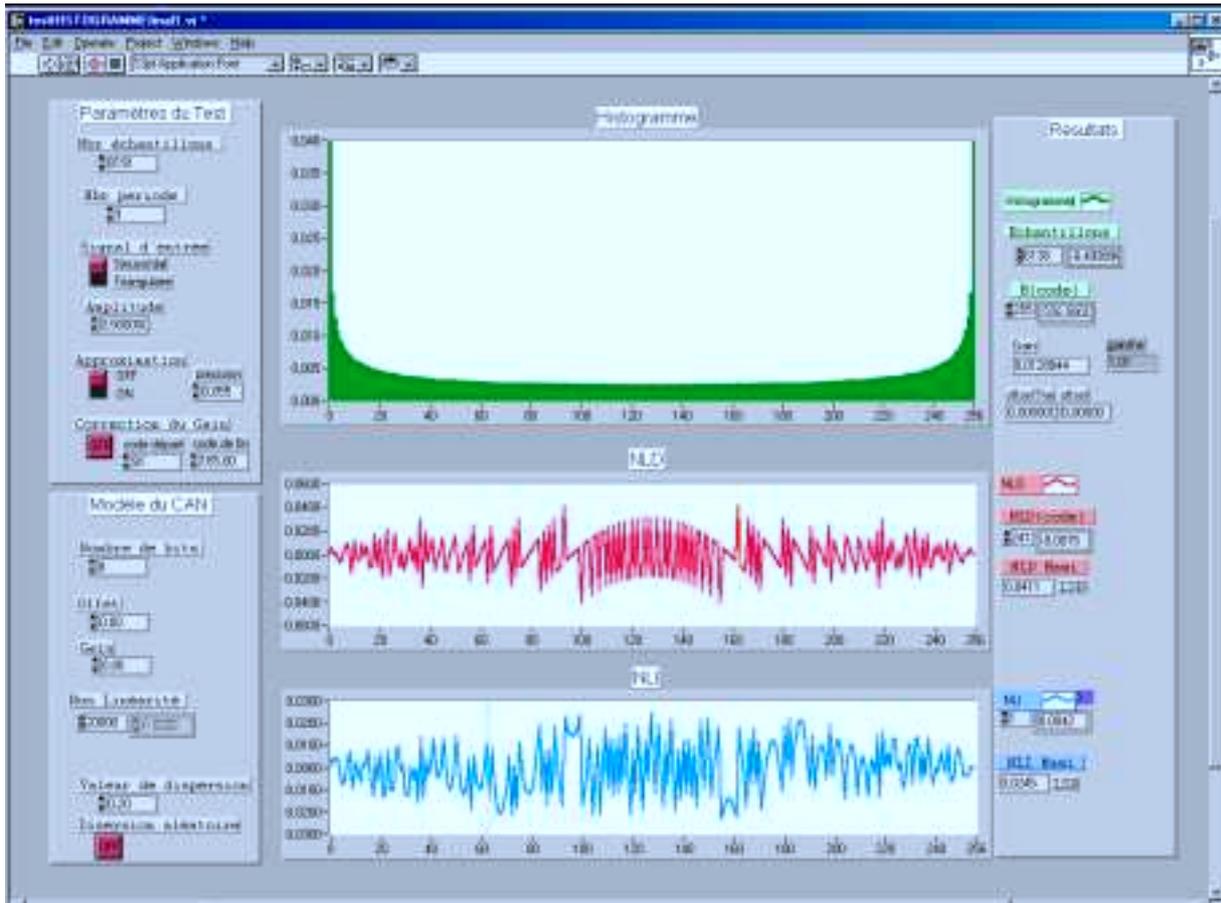


Figure 3.10 : Interface utilisateur

En utilisant les valeurs des différentes données entrées par l'utilisateur, le programme effectue la comparaison entre les valeurs des paramètres fonctionnels mesurés avec notre technique de test et celles obtenues avec la technique classique de test par histogramme.

b) Résultats de simulation

Pour valider nos approximations nous avons effectué un grand nombre de mesures. Le tableau 3.3 donne quelques exemples de résultats obtenus dans le cas particulier du cahier des charges suivant :

- Convertisseur de 8 bits,
- Amplitude crête à crête du signal d'entrée : $A_{in} = PE + 6 \text{ LSB}$,
- Précision sur la mesure des NLD : $\delta_{DNL} = 0,005\text{LSB}$,
- Calcul de l'erreur gain effectué sur 64 codes.

		Valeurs Théoriques (LSB)	Technique Classique (LSB)	Erreur mesure (%)	Technique Proposée (LSB)	Erreur mesure (%)
Paramètres fonctionnels	Erreur d'Offset	-0,750	-0,772	2,9	-0,775	3,3
		-0,250	-0,260	4,1	-0,260	4,1
		1,000	1,027	2,7	1,035	3,5
		1,500	1,547	3,1	1,575	5,0
	Erreur de Gain	-1,500	-1,501	0,1	-1,517	1,1
		-1,000	-1,002	0,2	-1,010	1,0
		-0,500	-0,502	0,4	-0,504	0,8
		1,500	1,522	1,5	1,569	4,6
	NLD	-0,500	-0,509	1,8	-0,510	2,0
		-0,250	-0,263	5,3	-0,265	6,0
		0,500	0,474	5,3	0,470	6,0
		0,750	0,719	4,1	0,715	4,7
	NLI	-0,550	-0,579	5,2	-0,577	4,8
		-0,250	-0,256	2,4	-0,260	4,0
		1,000	1,063	6,3	1,063	6,3
		0,500	0,477	4,7	0,479	4,3

Tableau 3.3 : Comparaison entre notre technique et la technique classique de test par histogramme

Globalement, les valeurs des paramètres obtenues avec notre technique et celles obtenues avec la technique classique sont extrêmement proches et valident donc nos approximations. Les erreurs dues à l'utilisation de la technique de test que nous proposons sont du même ordre de grandeur que des erreurs de mesure dues à l'utilisation de la technique classique de test par histogramme.

III.2. Implantation niveau porte

Le paragraphe précédent ayant permis de valider les approximations utilisées dans le cas d'un signal sinusoïdal, nous pouvons maintenant nous intéresser à l'implantation bas niveau de l'analyseur de réponses de test. Plus précisément, il s'agit de traduire l'architecture générale définie au chapitre 2 en une description au niveau porte en essayant d'optimiser les ressources matérielles.

Comme dans le cas d'un signal d'entrée triangulaire, l'analyseur de réponses de test est composé de trois blocs :

- Le détecteur de code,
- L'analyseur de réponses de test,
- Le contrôleur.

a) Détecteur de code

La fonction du détecteur de code est de positionner la valeur du code de référence et comparer ce code avec le code de sortie du convertisseur. Cette fonction est totalement indépendante du type de signal d'entrée. Par conséquent, nous pouvons réutiliser le détecteur de code développé pour un signal triangulaire.

b) Module d'exploitation

Selon l'architecture haut niveau définie au chapitre 2, le module d'exploitation est constitué des éléments suivants :

- 2 mots mémoire pour le stockage de l'histogramme expérimental,
- $2k+1$ mots mémoire pour le stockage l'histogramme de référence,
- 1 additionneur,
- 1 soustracteur,
- 1 diviseur,

la constante k représentant le nombre de domaines nécessaires à l'approximation par morceaux de l'histogramme de référence.

Comme dans le cas d'un signal d'entrée triangulaire, le rôle du module d'exploitation consiste à évaluer la valeur des paramètres fonctionnels. Il n'est cependant pas possible de réutiliser directement le module d'exploitation développé pour un signal triangulaire. En effet à partir de l'architecture haut niveau, ce module a été optimisé en tirant parties des propriétés liées à un histogramme linéaire. Grâce à la distribution uniforme de l'histogramme de référence, il a notamment été possible de réaliser de manière implicite le stockage de cet histogramme dans le registre d'exploitation et de remplacer les opérations de division par de simples décalages dans ce registre. Dans le cas d'un signal sinusoïdal, l'histogramme de référence présente une probabilité de distribution non uniforme, imposant au module d'exploitation d'être capable de fournir les différentes valeurs de l'histogramme de référence et de réaliser des divisions avec ces valeurs. Une nouvelle structure doit donc être développée. Toutefois, il est à noter que comme dans le cas du signal triangulaire, les opérations d'addition et de soustraction vont autant que possible être ramenées à de simples opérations de comptage/décomptage afin de baser la structure du module d'exploitation autour d'un registre compteur/décompteur.

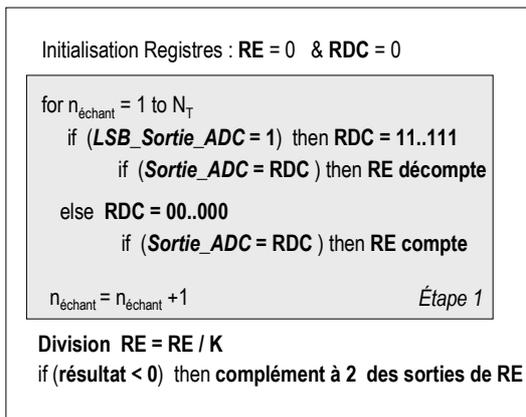
Avant de détailler plus en avant l'implantation, nous devons rappeler les expressions définissant chaque paramètre fonctionnel :

$$\bullet \text{ Erreur d'Offset} = \frac{H^{\text{exp}}(2^n) - H^{\text{exp}}(1)}{K} \quad (2.23)$$

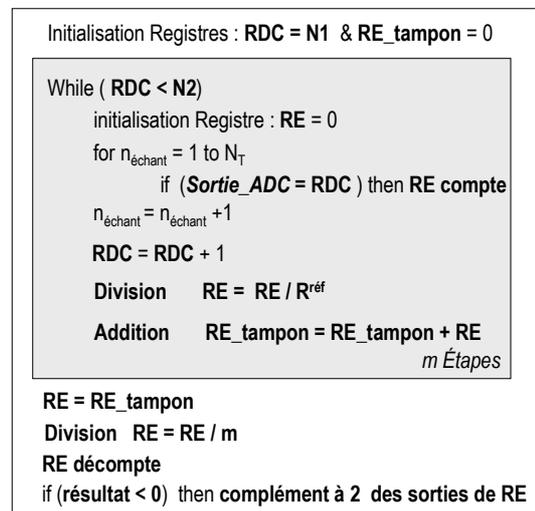
$$\bullet \text{ Erreur de Gain \% PE} = \left(\frac{1}{m} \times \sum_{i=N_1}^{N_2} \frac{H^{\text{exp}}(i)}{H^{\text{réf}}(i)} - 1 \right) \quad (2.27)$$

$$\bullet \text{ NLD}(i) = \frac{H^{\text{exp}}(i)}{H^{\text{réf}}(i)} - 1 \quad (2.28) \quad \text{NLI}(i) = \sum_{j=1}^i \text{NLD}(j) \quad (2.29)$$

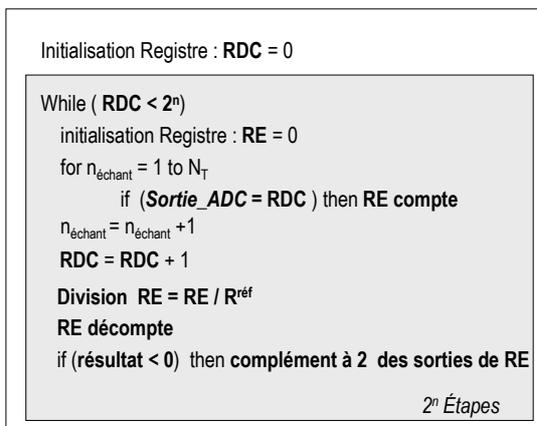
Pour illustration, les algorithmes qui définissent les procédures d'évaluation des paramètres fonctionnels sont présentés ci-après :



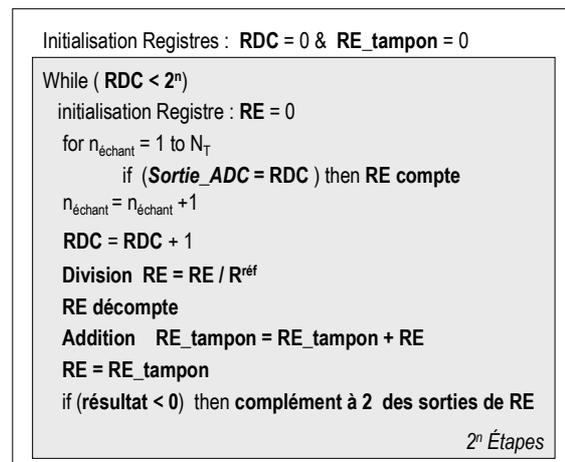
a) Calcul de l'erreur d'offset



b) Calcul de l'erreur de gain



c) Calcul des NLD



d) Calcul des NLI

Algorithme 3.4 : Procédures de calcul des différents paramètres fonctionnels

Comme les calculs effectués pour l'évaluation des paramètres sont relativement proches de ceux effectués dans le cas d'un signal d'entrée triangulaire, nous ne détaillerons pas ces algorithmes. En revanche, nous allons analyser précisément les différences qui existent entre ces deux cas.

L'analyse des différents algorithmes de calcul des paramètres conduit à plusieurs observations. Tout d'abord, il apparaît que le registre configurable RE développé dans le cas linéaire est effectivement utilisé dans les diverses procédures pour compter, décompter ou délivrer le complément à 2. Cependant ce registre configurable seul ne suffit pas. En particulier, l'utilisation d'un diviseur apparaît nécessaire dans la mesure où, contrairement au cas linéaire, l'élément diviseur n'est plus une constante.

On peut noter à ce niveau que dans la plupart des procédures, l'élément diviseur correspond à la fréquence d'apparition du code en cours de traitement pour l'histogramme de référence. Cela suppose donc que la valeur de $H^{\text{réf}}(i)$ soit disponible à chaque étape. Durant les procédures d'évaluation des paramètres définies par les algorithmes 4.4, nous considérons avoir calculé la valeur de $H^{\text{réf}}(i)$ avant chaque étape à l'aide de l'estimation suivante :

$$H^{\text{est}}(i) = \frac{\alpha}{i + \beta} \approx H^{\text{réf}}(i) \quad (2.30)$$

Cette valeur est alors stockée dans un registre dédié, noté $R^{\text{réf}}$.

L'analyse des différents algorithmes laisse aussi apparaître qu'il n'est pas possible de ramener toutes les additions à de simples opérations de comptage. Ainsi, l'implantation des sommes cumulatives utilisées lors des calculs du gain et des NLI nécessite l'utilisation d'un additionneur et d'un registre tampon. Notons que cet additionneur ainsi que le diviseur précédemment défini sont aussi indispensables au calcul de l'histogramme de référence d'après l'expression 2.30.

Finalement du point de vue des ressources matérielles, la structure du module d'exploitation comporte :

- 1 registre configurable accompagné de 1 registre tampon pour le traitement de l'histogramme expérimental,
- $2k+1$ registres pour le calcul de l'histogramme de référence,
- 1 additionneur,
- 1 diviseur.

La structure optimisée du registre configurable est totalement similaire à la structure proposée dans le cas d'un signal d'entrée triangulaire. Les autres registres correspondent à de simples mots mémoire. Nous nous intéressons par conséquent dans la suite à l'implantation optimisée du diviseur et de l'additionneur.

➤ *Diviseur*

Un diviseur classique sur 10 bits est constitué de 10 additionneurs de 10 bits. Intégrer un tel dispositif conduirait à une surface excessive pour une structure BIST.

Afin de minimiser la surface, nous avons développé une structure de diviseur série qui permet d'effectuer une décomposition temporelle de l'opération de division. L'architecture du diviseur série proposé (figure 3.11) est constituée de 5 registres à décalage, d'un multiplexeur numérique de 2 vers 1, d'un soustracteur/additionneur 1 bit et d'un mot mémoire 1 bit M.

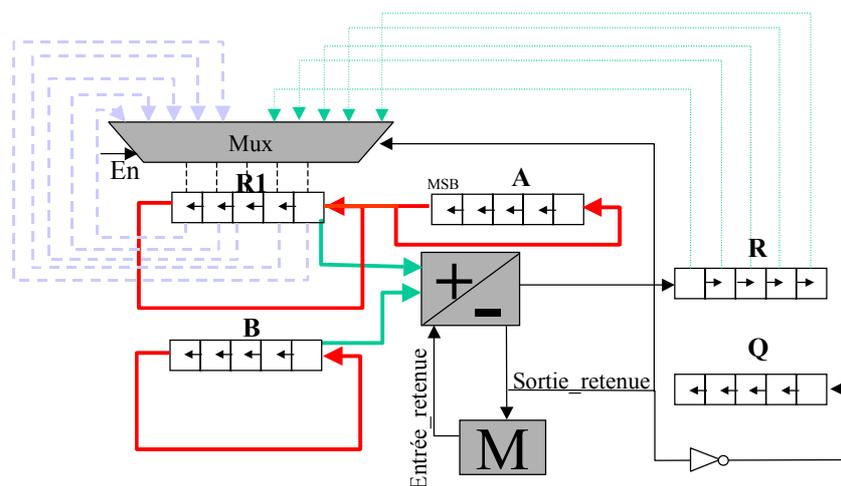
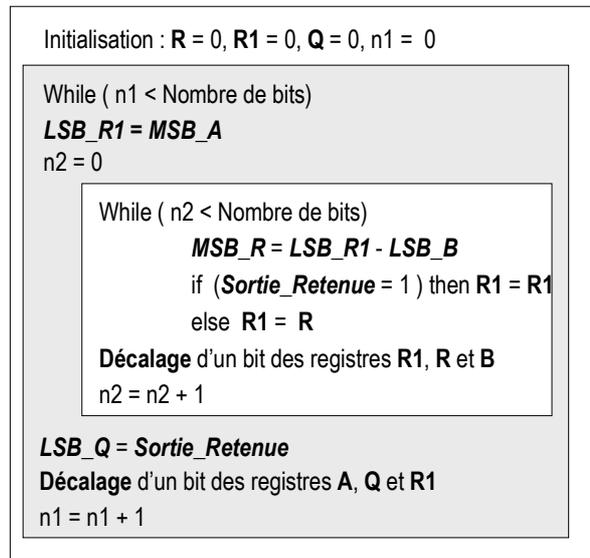


Figure 3.11 : Diviseur série

Le module d'addition/soustraction est composé de quelques portes logiques et le mot mémoire est réalisé à l'aide d'une simple bascule. Par conséquent, l'intégration du diviseur obtenu ne nécessite qu'une surface réduite de silicium.

L'algorithme 3.5 décrit les différentes étapes nécessaires au calcul de la division d'un nombre binaire stocké dans le registre A par un nombre binaire stocké dans le registre B.



Algorithme 3.5 : Division du registre A par le registre B

Après la phase d'initialisation de l'ensemble des registres, la première étape consiste à transférer la valeur du bit de poids fort du registre A (MSB_A) dans le bit de poids faible du registre R1 (LSB_{R1}). Ensuite, après avoir effectué une soustraction bit à bit du registre R1 par le registre B, le résultat complété de la retenue est stocké dans le bit de poids faible du registre Q (LSB_Q). La valeur de la retenue indique si le résultat de la soustraction entre R1 et B est négatif ou positif. S'il est négatif, le registre R1 est égal à lui-même et s'il est positif, le registre R1 est égal au résultat de la soustraction préalablement stocké dans le registre R. Ensuite, les trois registres A, Q et R1 subissent un décalage d'un bit et on revient à la première étape de l'algorithme : transfert du MSB du registre A dans le LSB du registre R1. A la fin des différentes itérations, le résultat de la division est stocké dans le registre Q et le reste de la division est stocké dans le registre R.

➤ Additionneur

Pour effectuer les opérations d'addition il suffit d'utiliser une partie des ressources matérielles du diviseur série présenté précédemment. En effet, en utilisant le module d'addition/soustraction et 3 registres du diviseur série nous pouvons construire une architecture d'additionneur série (figure 3.12).

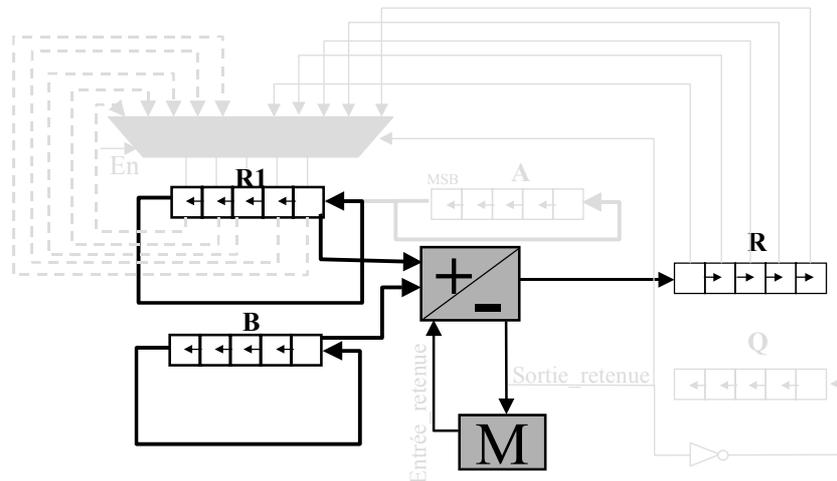
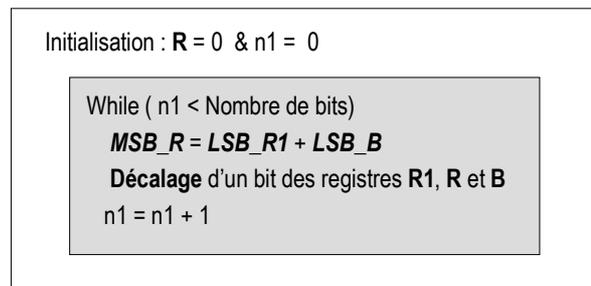


Figure 3.12 : Additionneur série

L'algorithme suivant détaille les différentes étapes permettant d'effectuer l'addition série entre deux mots binaires stockés dans les registres R1 et B.



Algorithme 3.6 : Addition entre le registre R1 et registre B

Après avoir initialisé le registre R, les LSB des registres R1 et B sont additionnés à l'aide du module additionneur/soustracteur 1 bit et le résultat est stocké dans le MSB du registre R. Ensuite, les registres B et R1 subissent un décalage d'un bit vers le MSB et les bits du registre R sont décalés d'un bit vers le LSB. Après ces décalages, les nouvelles valeurs des LSB des registres R1 et B sont ajoutés et ainsi de suite tant que tous les bits non pas été traités. Finalement, le registre R contient bien la valeur de la somme entre le registre B et le registre R1.

c) Contrôleur

La gestion des différentes phases de la procédure de test s'avère plus complexe dans le cas d'un signal d'entrée sinusoïdal que dans le cas d'un signal d'entrée triangulaire. En effet, en plus des opérations de comptage/décomptage ou complément à 2 du registre RE, le contrôleur doit gérer les différentes phases de calcul des opérations de division et d'addition décrites par les algorithmes 3.5 et 3.6. Néanmoins ce module correspond à un bloc numérique classique qui peut être synthétisé automatiquement à partir d'une description VHDL en utilisant un outil de synthèse standard.

IV. Discussion

Dans ce paragraphe, nous présentons la principale limitation de notre solution de test intégré et les améliorations possibles que nous proposons. Pour illustrer nos explications, l'ensemble de l'étude est effectué dans le cas d'un signal d'entrée triangulaire.

IV.1. Limitation

La technique de décomposition temporelle utilisée pour limiter les ressources mémoire nécessaires à l'intégration de la technique de test par histogramme implique l'utilisation d'un grand nombre de trames du signal d'entrée. L'optimisation en surface de l'analyseur de réponses de test est donc obtenue au détriment du temps de test.

Afin d'évaluer le temps de test requis par l'application de notre technique, nous devons tout d'abord calculer le nombre de trames de test utilisées.

Que ce soit pour un signal d'entrée triangulaire ou sinusoïdal, nous avons vu que les algorithmes d'évaluation des paramètres nécessitent :

- 1 étape élémentaire pour le calcul de l'erreur d'offset,
- m étapes élémentaires pour le calcul de l'erreur de gain,
- 2ⁿ étapes élémentaires pour le calcul des NLD,
- 2ⁿ étapes élémentaires pour le calcul des NLI.

Chaque étape élémentaire nécessitant l'application d'une trame de test complète, le nombre total de trames est donné par l'expression (3.5) :

$$\text{Nombre_Trames} = 2,25 \times 2^n \quad (3.5)$$

en supposant que la mesure du gain est réalisée sur un quart des codes ($m = 2^{n-2}$)

Pour chaque trame, N_T échantillons sont collectés en sortie du convertisseur. Ainsi, le temps correspondant à une trame est donné par l'expression (3.6) :

$$\text{Temps_Trame} = \frac{N_T}{F_{\text{éch}}} \quad (3.6)$$

où $F_{\text{éch}}$ correspond à la fréquence d'échantillonnage.

Nous pouvons alors utiliser l'expression 2.7 rappelée ci-dessous qui permet de calculer le nombre d'échantillons nécessaires dans chaque trame :

$$N_T = \frac{2^n \times A_{\text{in}}}{PE \times \delta_{\text{NLD}}} \quad (2.7)$$

Finalement, si nous négligeons l'influence de l'amplitude du signal d'entrée ($PE \cong A_{\text{in}}$) sur le nombre d'échantillons, nous obtenons l'expression suivante pour le temps de test :

$$\text{Temps de test} = \text{Nombre_Trames} \times \text{Temps_Trame} \cong \frac{2,25 \times 2^{2n}}{F_{\text{éch}} \times \delta_{\text{NLD}}} \quad (3.7)$$

Le tableau 3.5 représente le temps de test en fonction du nombre n de bits du convertisseur et de la fréquence d'échantillonnage $F_{\text{éch}}$ pour une précision sur la mesure des NLD inférieure à 0,05 LSB.

Nombre de bits n	Temps de test			
	$F_{\text{éch}} = 1\text{MHz}$	$F_{\text{éch}} = 10\text{MHz}$	$F_{\text{éch}} = 50\text{MHz}$	$F_{\text{éch}} = 100\text{MHz}$
6	180 ms	18 ms	4 ms	2 ms
8	2,9 s	290 ms	59 ms	29 ms
10	47 s	4,7 s	0,9 s	470 ms
12	12 mn	1 mn	15 s	7,5 s
14	3 h 21mn	32 mn	4 mn	2 mn

Tableau 3.5 : Temps de test

Nous pouvons observer que le temps de test augmente considérablement avec le nombre de bits du convertisseur. Notons que cette augmentation du temps de test est moins critique dans un contexte de test intégré que dans le cas d'un test externe si la solution de test intégré

permet par ailleurs de simplifier l'équipement de test nécessaire. Toutefois, si nous ne voulons pas dépasser un temps de test de quelques secondes, la solution de test intégré que nous proposons ne peut être appliquée qu'aux convertisseurs de faible et moyenne résolution.

IV.2. Optimisation

L'objectif des travaux présentés dans cette section consiste à proposer des optimisations permettant de diminuer le temps nécessaire au test. Dans tous les cas, ces optimisations impliquent une augmentation de la surface de la structure BIST et donc ne devront être utilisées que dans les cas où le temps de test est réellement critique.

La première optimisation évidente consiste à évaluer simultanément les DNL et les INL. On obtient alors les algorithmes suivants pour le cas d'un signal d'entrée triangulaire et sinusoïdal :

```

Initialisation Registres : RDC = 0 & RE2 = 0

While ( RDC < 2n )
  initialisation Registre RE = 0
  for néchant = 1 to NT
    if (Sortie_ADC = RDC ) then RE & RE2 comptent
    néchant = néchant + 1
  RDC = RDC + 1
  Décalage de P bits dans RE
  RE décompte
  if (résultat < 0) then complément à 2 des sorties de RE
  *** NLD disponible sur les sorties de RE ***
  RE = RE2
  RE décompte
  if (résultat < 0) then complément à 2 des sorties de RE
  *** NLI disponible sur les sorties de RE ***
  2n Étapes
    
```

a) Signal d'entrée triangulaire

```

Initialisation Registres : RDC = 0 & RE2 = 0

While ( RDC < 2n )
  initialisation Registre RE = 0
  for néchant = 1 to NT
    if (Sortie_ADC = RDC ) then RE compte
    néchant = néchant + 1
  RDC = RDC + 1
  Division RE = RE / Rréf
  RE décompte
  Addition RE_tampon = RE_tampon + RE
  if (résultat < 0) then complément à 2 des sorties de RE
  *** NLD disponible sur les sorties de RE ***
  RE = RE_tampon
  RE décompte
  if (résultat < 0) then complément à 2 des sorties de RE
  *** NLI disponible sur les sorties de RE ***
  2n Étapes
    
```

b) Signal d'entrée sinusoïdal

Algorithme 3.7 : Nouvelles procédures de calcul des NLD et des NLI

Du point de vue des ressources matérielles, le calcul simultané des NLD et NLI nécessite l'ajout d'un compteur RE2 dans le cas d'un signal d'entrée triangulaire. En effet, les NLI dans ce cas peuvent s'exprimer par :

$$NLI(i) = \frac{1}{H_{idéale}} \times \left(\sum_{j=1}^i H^{\exp}(j) \right) - i \quad (3.8)$$

Le compteur supplémentaire permet ainsi d'accumuler les valeurs de l'histogramme expérimental $H^{\text{exp}}(i)$ sans les initialisations intermédiaires requises à chaque étape pour le calcul des NLD.

Dans le cas d'un signal d'entrée sinusoïdal, toutes les ressources matérielles requises sont déjà disponibles dans la structure. En effet, le module d'exploitation original comporte deux registres, autorisant par conséquent le calcul parallèle des NLD et NLI.

Ainsi, avec une augmentation de surface extrêmement faible dans le cas linéaire et nulle dans le cas sinusoïdal, le calcul simultané des NLD et NLI permet d'obtenir une réduction du temps de test approchant un facteur 2.

La deuxième optimisation possible concerne le calcul de l'erreur de gain. Rappelons que la solution initiale repose sur l'évaluation du gain à partir des m codes centraux, impliquant l'application de m trames de test successives. De plus, l'évaluation du gain s'appuie sur l'hypothèse que la fréquence d'apparition des m codes centraux est constante pour l'histogramme de référence, à la fois pour un signal d'entrée triangulaire ou sinusoïdal. Ainsi l'erreur de gain s'exprime par :

$$\text{Erreur_de_Gain} = \left(\frac{1}{m \cdot H^{\text{réf}}} \times \sum_{i=N1}^{N2} H^{\text{exp}}(i) \right) - 1 \quad (3.9)$$

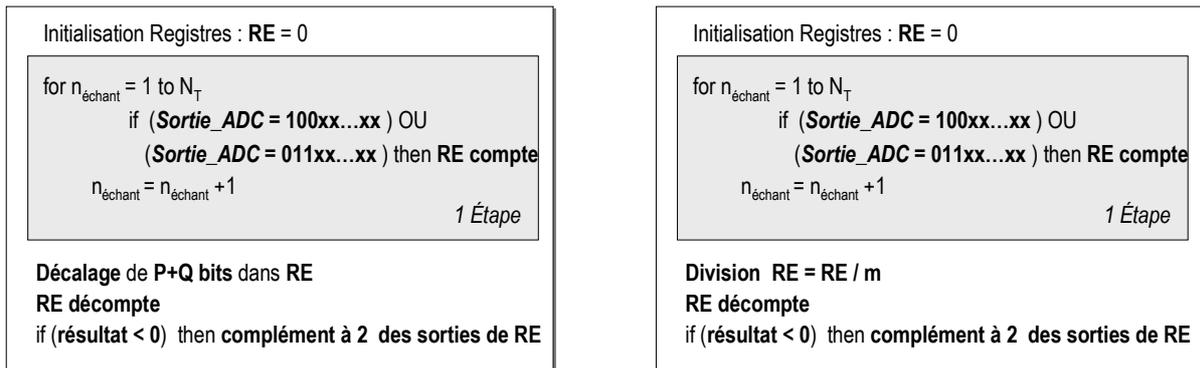
Notons à ce niveau que cette expression suppose que l'approximation par morceaux utilisée pour l'estimation de l'histogramme de référence dans le cas sinusoïdal, soit réalisée par une constante dans la zone centrale de l'histogramme.

Afin de réduire le temps de test, l'idée consiste à accumuler en une seule trame de test les valeurs de l'histogramme expérimental pour les m codes centraux, au lieu d'une accumulation en m trames de test successives. Pour cela, nous proposons de modifier le détecteur de code pour que, dans la phase de détermination du gain, il délivre un signal de sortie au niveau logique "0" chaque fois qu'un des m codes centraux est présent en sortie du convertisseur.

En pratique, il suffit de comparer les 3 bits de poids fort des codes de sortie du CAN avec les codes 011 et 100 pour sélectionner les échantillons correspondant à un quart de codes centraux. Si l'on désire réaliser la mesure sur un nombre moins important de codes, par exemple seulement sur un huitième des codes, il suffit de comparer les 4 bits de poids fort des

codes de sortie du CAN avec les codes 0111 et 1000. Ce type de comparaison s'implante très facilement à l'aide de quelques portes logiques. Notons que dans ce cas, la structure compteur-comparateur du détecteur de code n'est pas utilisée.

L'algorithme suivant décrit la nouvelle procédure de calcul du gain, dans laquelle les échantillons correspondants aux $m = 2^{n-2}$ codes centraux sont collectés en une seule trame de test.



Algorithme 3.8 : Nouvelles procédures de calcul de l'erreur de gain

Finalement, à l'aide de ces deux optimisations, il est possible de réduire le temps de test. Ce temps est maintenant défini par l'expression suivante :

$$\text{Temps de test} = \frac{2^{2n}}{F_{\text{éch}} \times \delta_{\text{NLD}}} \quad (3.8)$$

Pour illustration, le temps de test d'un convertisseur 12 bits échantillonné à 100MHz est ramené de 7,5 s à 3.3s. Il s'agit d'une réduction significative du temps de test compte tenu de l'augmentation négligeable de la surface de l'analyseur de réponse de test.

Néanmoins, si la réduction du temps de test ainsi obtenu se révèle insuffisante pour une application donnée, une solution consiste à dupliquer les ressources de traitement pour analyser simultanément plusieurs codes de sortie du convertisseur. L'augmentation en surface de la structure BIST est dans ce cas plus importante puisqu'en première approximation, on peut considérer que la surface de l'analyseur est proportionnelle au nombre de codes traités en parallèle. Il existe donc un compromis à définir entre surface additionnelle et temps de test.

IV.3. Automatisation

La complexité des circuits intégrés actuels nécessite l'utilisation d'outils d'aide à la conception. Depuis déjà plusieurs années, la génération automatique de structures BIST pour circuits numériques est incorporée dans les outils de synthèse. Actuellement, il n'existe aucun équivalent pour les circuits analogiques et mixtes.

Précédemment, nous avons montré que l'analyseur de réponses de test est constitué d'un ensemble de blocs élémentaires dont le nombre dépend de l'application envisagée. Le calcul du nombre de blocs est facilement automatisable ce qui rend possible l'intégration de la structure BIST proposée dans des outils de synthèse automatique.

Par exemple, dans le cas du cahier des charges proposé dans la partie I, la construction de la structure BIST illustrée figure 3.13 permet d'obtenir une structure composée de 10 blocs élémentaires pour le Registre d'Evaluation, de 6 blocs pour le Registre de détection de code et d'un contrôleur synthétisé à partir d'une description en VHDL.

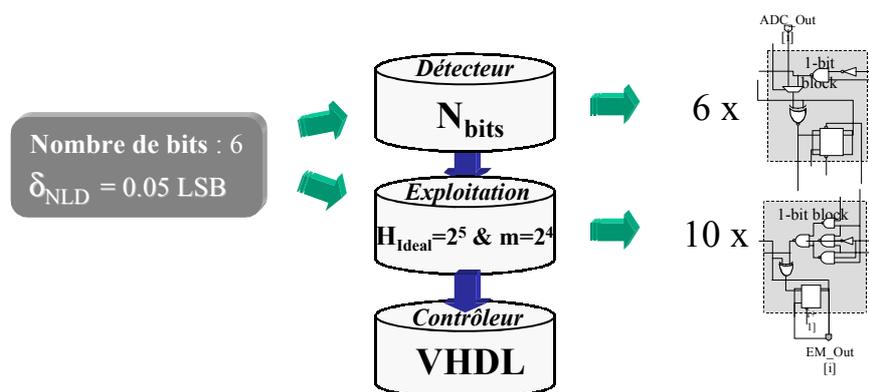


Figure 3.13 : Automatisation de construction de la structure

Nous avons développé un programme qui permet de calculer le nombre des différents blocs élémentaires nécessaires à une application donnée. Il permet aussi d'estimer la surface totale de la structure complète et le temps nécessaire pour tester le convertisseur considéré. Si le temps de test s'avère trop long pour l'application envisagée, il est possible d'imposer une durée de test maximale. Dans ce cas, le programme utilise les différentes possibilités de parallélisation proposées dans la section III et fournit la valeur de la nouvelle estimation de surface de la structure BIST.

V. Conclusion

Dans ce chapitre, nous avons présenté des solutions d'implantation bas niveau pour l'analyseur intégré de réponses de test. A partir de l'architecture générale composée d'un contrôleur, d'un module de détection de code et d'un module d'exploitation, nous avons proposé des solutions physiques pour ces différents blocs, à la fois pour le cas d'un signal d'entrée triangulaire et sinusoïdal.

Dans les deux cas, le contrôleur a été décrit en VHDL et synthétisé à l'aide d'un outil de synthèse automatique (Synopsis). Le module de détection des codes est identique dans les deux cas et une structure originale de compteur/comparateur a été développée. Finalement, une architecture basée autour d'un registre configurable a été proposée pour le module d'exploitation. Ce registre permet d'effectuer des opérations d'addition en mode compteur et de soustraction en mode décompteur. Il peut aussi délivrer le complément à 2 d'un résultat en cas de valeur négative. Ce registre configurable permet en fait de réaliser toutes les opérations nécessaires aux calculs des paramètres fonctionnels dans le cas d'un signal d'entrée triangulaire. Dans le cas d'un signal d'entrée sinusoïdal, le module d'exploitation a été complété par un registre tampon supplémentaire et un bloc original de diviseur/additionneur série.

Ces différentes solutions ont été validées par simulation et une implantation physique de l'analyseur de réponses de test pour signal d'entrée triangulaire a été réalisée. Concernant l'analyseur pour un signal d'entrée sinusoïdal, la validité des approximations introduites lors du calcul des paramètres fonctionnels a été vérifiée au travers d'un programme d'évaluation permettant de comparer les résultats de mesure obtenus avec notre structure BIST et ceux obtenus avec une approche classique de test par histogramme.

Dans la dernière partie du chapitre, nous avons mis en évidence le problème de la durée du test lié à l'utilisation de la technique de décomposition temporelle. En effet, si cette technique permet de réduire considérablement la surface nécessaire à l'implantation la technique de test par histogramme, elle introduit nécessairement une augmentation du temps de test. Pour réduire la durée du test, nous avons proposé plusieurs optimisations de la technique de test permettant de limiter le nombre de trames du signal d'entrée utilisées. Enfin, nous avons détaillé les différentes étapes permettant de construire notre structure BIST pour une application donnée. En pratique, ces étapes de conception sont facilement automatisables et peuvent être incorporées dans des outils d'aide à la conception de circuits intégrés.

Chapitre 4

Générateur de stimuli

L'objectif des travaux présentés dans ce manuscrit concerne le développement de structures de test intégré (BIST) dédiées aux Convertisseurs Analogique/Numérique. Jusqu'à présent, nous nous sommes intéressés au problème de l'analyse intégrée des réponses de test et nous avons proposé des solutions d'analyseurs de réponses pour les différents types de signaux d'entrée utilisés lors du test par histogramme. En vue de l'intégration d'une solution BIST complète, nous nous consacrons dans ce chapitre au problème de la génération intégrée de stimuli de test, et plus particulièrement au problème de la génération intégrée de signaux analogiques

Ce chapitre est organisé en 4 parties distinctes. Dans une première partie, nous exposons les contraintes spécifiques que doit respecter un générateur de signaux dédié au test intégré. Les principes généraux de génération de signaux analogiques linéaires sont alors introduits dans la seconde partie. La troisième partie est consacrée au développement d'une structure originale de générateur de rampe auto-calibré. Finalement, la structure proposée est étendue à la génération de signaux triangulaires auto-calibrés dans la dernière partie.

I. Performances requises d'un générateur de stimuli

Notre objectif est de générer le stimulus de test analogique sur la même puce que le convertisseur à tester. Ce contexte de test intégré impose des contraintes particulières sur le

Tout d'abord, le signal analogique généré doit présenter des qualités supérieures au circuit à tester pour ne pas fausser l'interprétation des résultats. En effet, pour le test d'un Convertisseur Analogique/Numérique, toute imperfection au niveau du signal d'entrée peut

être interprétée en sortie du CAN comme une imperfection du convertisseur. Dans ce contexte, les deux caractéristiques principales permettant d'estimer la qualité d'un signal linéaire appliqué comme stimulus de test sont l'amplitude et la linéarité du signal généré. Examinons plus en détail, l'influence de ces caractéristiques sur l'évaluation des paramètres fonctionnels d'un CAN. Pour cela, l'approche utilisée consiste à appliquer en entrée d'un convertisseur parfait le signal analogique linéaire présentant des imperfections et évaluer l'histogramme résultant. Il est clair que toute modification de cet histogramme par rapport à l'histogramme idéal entraînera une erreur dans l'évaluation des paramètres fonctionnels du CAN.

Examinons tout d'abord le cas d'une variation de l'amplitude du signal d'entrée. A fréquence constante, toute variation d'amplitude correspond en fait à une variation de la pente du signal d'entrée, et provoque par conséquent une modification de la fréquence d'apparition des codes en sortie du convertisseur parfait. Ainsi, la figure 4.1 illustre l'histogramme idéal et l'histogramme obtenu pour une réduction de l'amplitude du signal d'entrée. Il apparaît clairement une modification uniforme de la fréquence d'apparition des codes de sortie du convertisseur. Cet histogramme peut donc être interprété de la même manière qu'un histogramme obtenu en présence d'une erreur de gain au niveau du CAN.

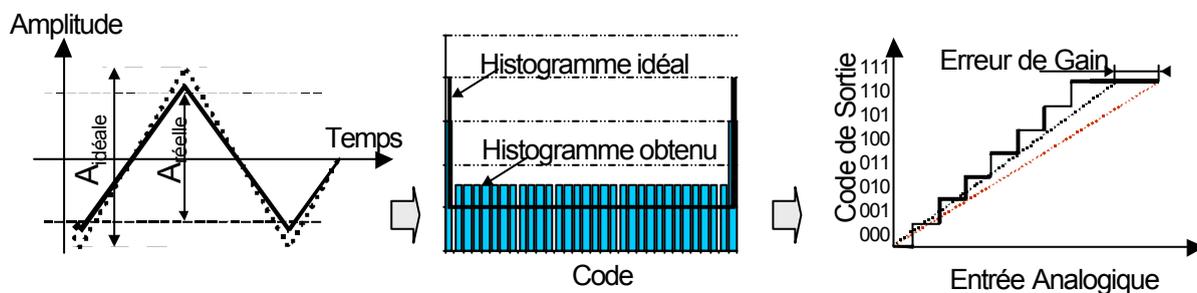


Figure 4.1 : Influence de l'amplitude du signal d'entrée

Supposons maintenant que le signal analogique appliqué à l'entrée du convertisseur parfait comporte des non-linéarités. La figure 4.2 illustre l'histogramme mesuré en sortie du convertisseur. Cet histogramme est en fait identique à un histogramme obtenu en présence d'erreurs de Non-Linéarités Différentielles (NLD) et d'erreurs de Non-Linéarités Intégrales (NLI) au niveau du convertisseur sous test.

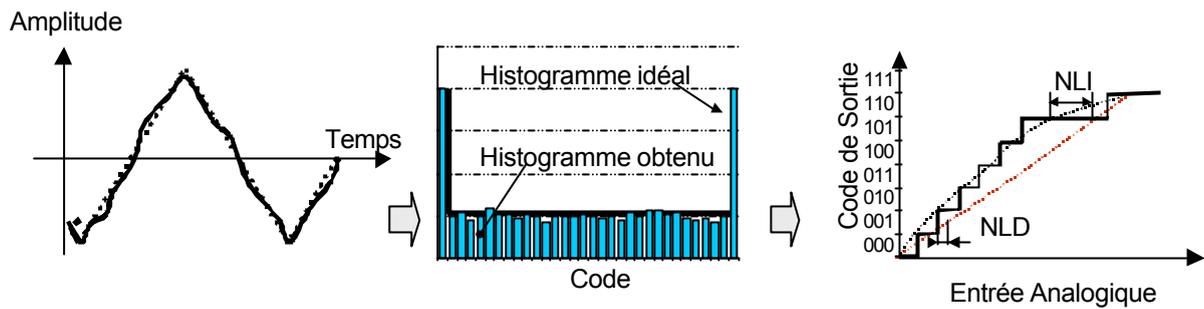


Figure 4.2 : Influence des non-linéarités du signal d'entrée

En plus de ces performances purement fonctionnelles, le contexte BIST de nos travaux nous impose des contraintes de surface très fortes. En effet, une solution BIST ne sera acceptée par le monde industriel que si la surface additionnelle nécessaire à l'intégration du test ne représente qu'un faible pourcentage de la surface du convertisseur à tester. Ainsi, les deux principales contraintes que doit respecter le générateur intégré de stimuli analogiques sont :

- surface minimale,
- qualité du signal généré en terme de linéarité et précision de la pente.

II. Générateur de signaux linéaires

II.1. Principe

Deux techniques sont principalement utilisées pour générer une rampe en interne :

- intégrer une tension constante,
- charger une capacité avec un courant constant.

Pour obtenir l'intégration linéaire d'une tension sur une plage convenable, il est nécessaire d'utiliser un Amplificateur Opérationnel (AO) ayant de très bonnes spécifications fonctionnelles. A première vue, intégrer un tel AO implique une surface trop importante dans un contexte de test intégré. Nous nous orientons donc vers la deuxième technique.

La figure 4.3 représente le schéma de principe d'un générateur de rampe basé sur le principe de la charge d'une capacité par un courant constant. La valeur du courant de charge I_c dépend de la tension de contrôle V_{ctr}

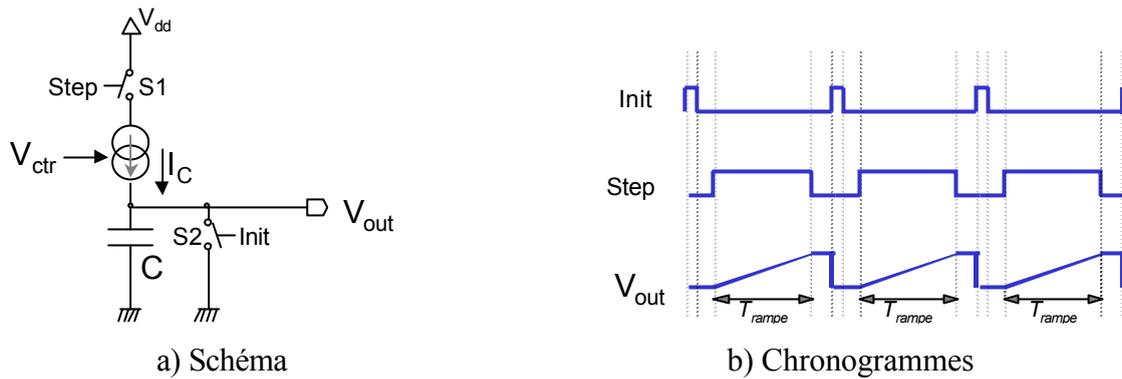


Figure 4.3 : Génération de rampe basée sur la charge d'une capacité à courant constant

Le montage est constitué d'un générateur de courant, d'une capacité et de deux interrupteurs S1 et S2 commandés respectivement par les signaux *Step* et *Init*. Le signal *Step* permet de contrôler la durée de la rampe et le signal *Init* permet de réinitialiser le signal. A priori, l'intégration d'un tel circuit ne paraît pas engendrer une surface de silicium excessive. Nous envisageons donc de baser notre générateur sur ce principe de charge d'une capacité à courant constant.

La valeur du signal de sortie V_{out} est donnée par l'expression suivante :

$$V_{out} = \frac{I_c}{C} \times t \quad (4.1)$$

Le signal de sortie est donc directement proportionnel au temps selon la pente définie par le rapport I_c/C . Le fait de contrôler précisément ce rapport conditionne aussi les performances du générateur en terme de précision de la pente. La linéarité du signal généré dépend quant à elle des variations du courant I_c . Dans la section suivante, nous présentons différentes implantations du générateur de courant permettant de limiter les fluctuations du courant de charge I_c .

II.2. Générateurs de courant

Idéalement, un générateur de courant délivre un courant constant quelle que soit la tension de sortie et quelles que soient les conditions de charge. Un bon générateur de courant est donc un montage délivrant un courant constant dans une grande dynamique de sortie et présentant une résistance équivalente de sortie aussi grande que possible.

$V_{\text{eff}} = V_{\text{gs}} - V_{\text{th}}$ comme la tension effective des transistors.

Pour les transistors M1 et M3, la condition de saturation est toujours respectée puisque $V_{\text{ds3}} = V_{\text{gs3}}$ et $V_{\text{ds1}} = V_{\text{gs1}}$. De plus, tous les transistors M1, M2, M3 et M4 sont identiques et

traversés par le même courant impliquant une différence de potentiel identique entre la grille et la source si on néglige l'influence du potentiel entre le substrat et la source sur la tension de seuil :

$$V_{gs1} = V_{gs2} = V_{gs3} = V_{gs4} = V_{gs} \quad (4.2)$$

De plus, nous pouvons écrire l'expression du potentiel de la grille du transistor M3 et en déduire l'expression de la tension entre le drain et la source du transistor M2 :

$$V_{g3} = V_{gs1} + V_{gs3} = 2 V_{gs} \quad (4.3)$$

$$V_{ds2} = V_{g3} - V_{gs4} = V_{g3} - V_{gs} = V_{gs} = V_{eff} + V_{tn} \quad (4.4)$$

Ainsi, la condition de saturation est respectée pour le transistor M2 avec une marge de fonctionnement égale à $V_{tn} \cong 1$ Volt.

En fait, les problèmes de saturation se posent uniquement pour le transistor M4 dont la tension drain/source dépend de la tension de sortie V_{out} .

$$V_{ds4} = V_{out} - V_{ds2} \quad (4.5)$$

Pour respecter la condition de saturation, il faut donc s'assurer que la valeur $V_{out} - V_{ds2}$ est supérieure à la tension effective V_{eff} . Nous obtenons alors la condition suivante sur le signal V_{out} :

$$V_{out} > 2 V_{eff} + V_{tn} \quad (4.6)$$

Pour être sûr d'être dans le fonctionnement désiré, il est nécessaire de prendre une marge de sécurité. Ainsi, nous prendrons généralement $V_{eff} \geq 0,2$ Volt. Or, comme $V_{tn} \cong 1$ Volt, la valeur de la tension de sortie V_{out} doit être supérieure à 1,4 Volt pour que le miroir de courant fournisse toujours un courant constant. Cette condition limite considérablement la dynamique utile du signal de sortie.

b) Miroir "grande excursion"

Pour augmenter la dynamique de sortie, nous utilisons le miroir de courant "grande excursion" présenté figure 4.5.

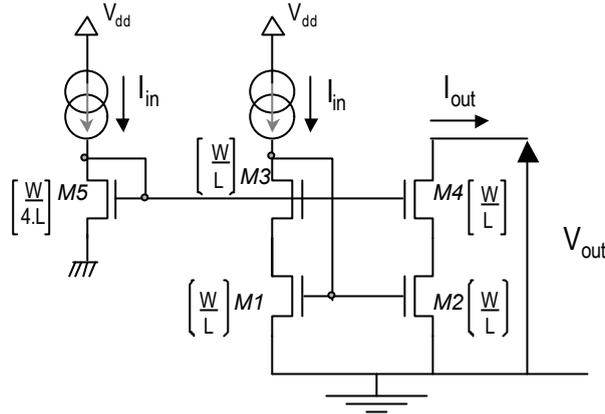


Figure 4.5 : Miroir de courant grande excursion

Comme dans le cas d'un miroir cascode, les quatre transistors M1, M2, M3 et M4 ont la même taille, sont parcourus par le même courant de drain I_d donné par l'expression (4.7) et ont la même tension effective V_{eff} .

$$I_d \cong K_n \cdot \frac{W}{L} \cdot (V_{eff})^2 \quad (4.7)$$

Dans cette expression, K_n est une constante qui dépend de la technologie utilisée.

Concernant le transistor M5, on a de la même manière $I_{d5} \cong K_n \cdot \frac{W}{4L} \cdot (V_{eff5})^2$

$$\text{soit } V_{eff5} = 4 V_{eff} \quad (4.8)$$

Pour que le miroir de courant fournisse un courant constant, il faut que tous les transistors fonctionnent en régime saturé.

$$\text{Or comme } V_{gs4} = V_{gs3} \quad (4.9)$$

$$\text{et que } V_{gs5} = V_{gs3} = V_{gs4} = 4 \cdot V_{eff} + V_{tn} \quad (4.10)$$

nous pouvons déterminer l'expression de la tension drain source du transistor M2 :

$$V_{ds2} = V_{gs5} - V_{gs4} = 4 \cdot V_{eff} + V_{tn} - V_{eff} - V_{tn} = 3V_{eff} \quad (4.11)$$

Ce transistor fonctionne donc bien en régime saturé.

Finalement, la condition pour que le transistor M4 soit en fonctionnement saturé impose la condition suivante sur la tension V_{out} :

$$V_{out} = V_{ds4} + V_{ds2} > 4 V_{eff} \quad (4.12)$$

Enfin, il faut s'assurer que le transistor M3 est bien saturé et respecte donc la condition $V_{ds3} > V_{eff}$. Pour trouver la valeur de V_{ds3} , nous pouvons remarquer que la grille du transistor M1 est connectée au drain du transistor M3. Il en résulte l'expression suivante :

$$V_{ds3} = V_{g1} - V_{ds1} = V_{gs} - V_{eff} = V_{tn} \quad (4.13)$$

En conclusion, tous les transistors du montage sont saturés si la condition (4.12) est vérifiée.

($V_{eff} = 0,2 \text{ Volt}$) que dans le cas du miroir de courant cascode, cette condition impose $V_{out} > 0,8 \text{ Volt}$. Par conséquent, ce montage permet d'augmenter sensiblement la dynamique du signal de sortie.

II.3. Schéma et performance du générateur

Dans cette section, nous présentons la structure finale d'un générateur de rampe utilisant le miroir de courant grande excursion décrit précédemment. Sur le schéma du générateur donné figure 4.6, nous retrouvons les deux signaux de commande *Step* et *Init* permettant de contrôler la durée de la rampe et d'initialisation de la structure. La tension de contrôle V_{ctr} permet d'ajuster la valeur du courant de charge I_c .

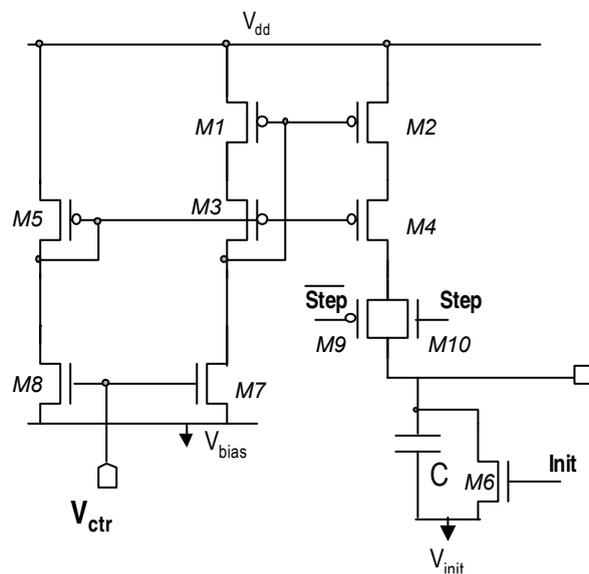


Figure 4.6 : Générateur de rampe

Le dimensionnement des différents éléments du générateur dépend de l'application envisagée. Pour valider la structure proposée, nous considérons le cahier des charges suivant :

- durée de la rampe égale à $100 \mu\text{s}$,
- amplitude de la rampe égale à 3 Volt,
- alimentation symétrique $\pm 2.5 \text{ V}$,
- tension d'initialisation $V_{\text{init}} = -1.5 \text{ V}$.

Il s'agit par conséquent de générer une rampe symétrique de pente 30 V/ms . Afin de limiter la surface du générateur, nous devons choisir une valeur raisonnable de la capacité de charge. Compte tenu de notre cahier des charges, nous imposons $C = 10 \text{ pF}$, ce qui implique $I_c = 0,3 \mu\text{A}$ pour assurer la pente de 30 V/ms . Le tableau 4.1 donne la largeur W et la longueur L des transistors du schéma. Le détail du calcul des dimensions des transistors est disponible en annexe 3.

	M1		M2		M3		M4		M5		M6		M7		M8		M9		M10	
	W	L	W	L	W	L	W	L	W	L	W	L	W	L	W	L	W	L	W	L
Tailles des transistors (μm) ($V_{\text{bias}} = 900 \text{ mV}$)	5	5	5	5	5	5	5	5	1.3	5	0.8	0.6	2	42.4	2	11.8	0.8	0.6	0.8	0.6

Tableau 4.1 : Tailles des transistors utilisés

Le fonctionnement de la structure finale du générateur de rampe a été vérifié au travers de simulations électriques. La figure 4.7 illustre la rampe obtenue qui présente une amplitude de 3 V pour une durée de $100 \mu\text{s}$.

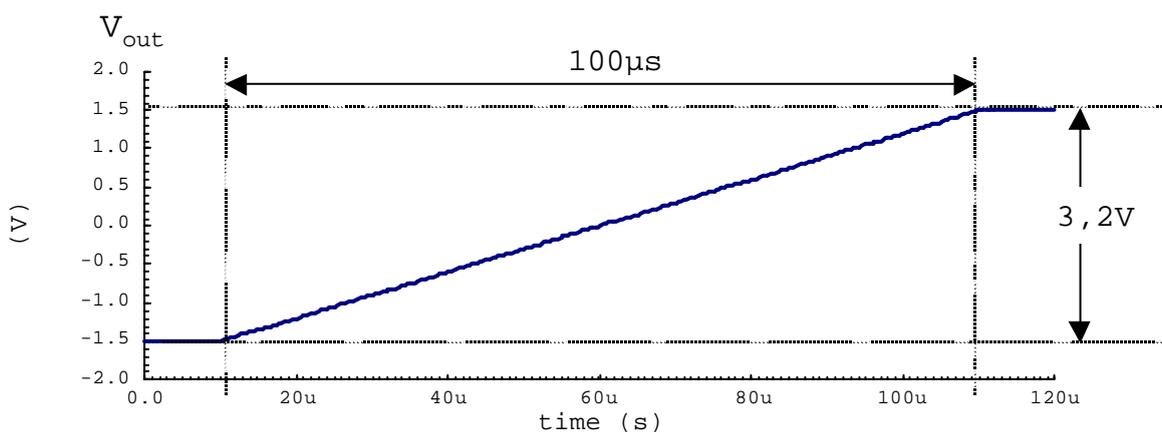


Figure 4.7 : Signal de sortie du générateur

L'évaluation plus précise de la qualité du générateur de rampe nécessite maintenant une analyse plus fine de la linéarité et de la précision de la pente du signal généré.

NL_{\max} est égale au LSB pour une amplitude de 3 volts, nous pouvons déterminer la résolution équivalente du générateur exprimée en nombre de bits à :

$$n = \log_2 \left(\frac{3}{NL_{\max}} \right) \quad (4.14)$$

Le générateur de rampe proposé présente donc une résolution équivalente égale à 15 bits. Généralement on considère que les ressources matérielles utilisées pour tester un CAN doivent avoir au moins deux bits de résolution de plus que le CAN sous test. Par conséquent, nous pouvons espérer tester des convertisseurs de 13 bits avec le générateur de rampe présenté, sous réserve que la pente du signal soit imposée de manière précise.

b) Précision de la pente

La deuxième contrainte à satisfaire pour le générateur de rampe est d'assurer une très grande précision sur la valeur de la pente du signal.

Or, la pente du signal est particulièrement difficile à fixer de manière précise pour le générateur proposé. En effet, les valeurs de la capacité C et du courant de charge I_c définissant la pente I_c/C sont très sensibles aux dispersions qui peuvent apparaître durant la fabrication du circuit. La figure 4.9 représente le résultat des simulations effectuées en prenant en compte ces dispersions. Nous pouvons observer que dans les pires cas, la valeur finale de la tension de sortie varie de 0,2 V à 2,5 V, ce qui correspond à des variations de la pente du signal de l'ordre de $\pm 50\%$.

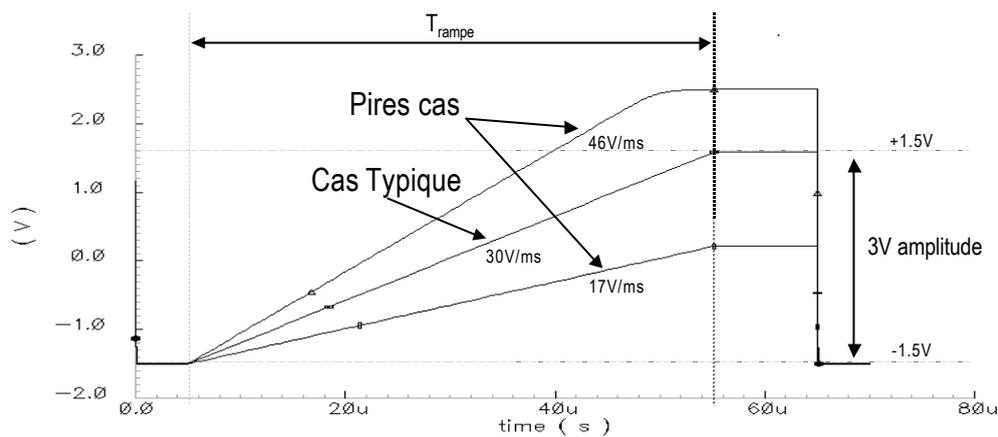


Figure 4.9 : Variation de la pente de la rampe

En théorie, pour résoudre ce problème, il est possible d'augmenter la valeur du courant de charge et la valeur de la capacité en maintenant constant le rapport courant sur capacité. Malheureusement, l'augmentation du courant et de la capacité a une répercussion directe sur la surface totale du circuit. Cette solution n'étant pas viable dans le contexte de test intégré où la surface additionnelle doit être minimale, nous proposons une approche alternative consistant à auto-calibrer le générateur à l'aide d'un système de compensation.

III. Générateur de rampe auto-calibré

III.1. Principe

La rampe idéale est définie avec une amplitude V pour une durée T_{rampe} (figure 4.10).

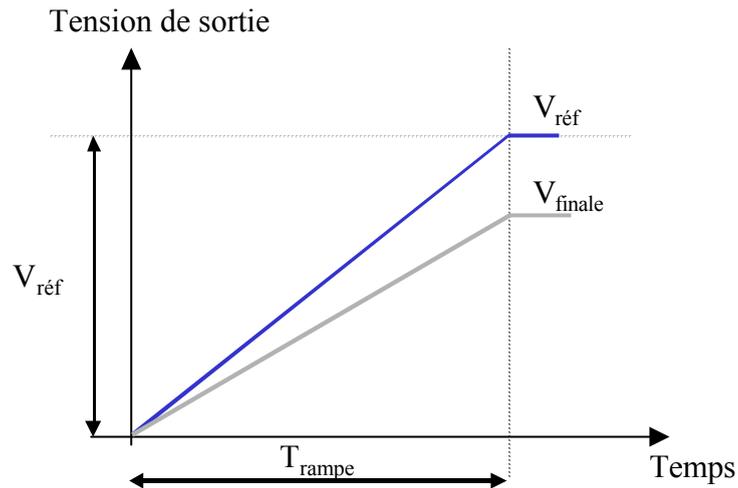


Figure 4.10 : Définition de la pente de la rampe

Le principe de base du système de calibration consiste à comparer la tension V_{finale} atteinte par la rampe au bout d'un temps fixe T_{rampe} avec une tension de référence $V_{\text{réf}}$. Suivant le résultat de la comparaison, le système de compensation augmente ou diminue la tension de réglage V_{ctr} , modifiant ainsi le courant de charge I_c et la pente (I_c/C) de la rampe. Finalement, au bout de plusieurs cycles, la tension V_{finale} converge vers la tension de référence.

Le schéma de principe du système de calibration présenté figure 4.11 est constitué de trois blocs :

- le générateur défini précédemment qui fournit une rampe dont la durée est fixée par le signal *Step* et dont la pente dépend de la tension de réglage V_{ctr} ,
- un comparateur qui indique si l'amplitude de la rampe est inférieure ou supérieure à la tension de référence imposée,
- le module de compensation qui modifie la valeur de la tension de réglage en fonction du résultat fourni par le comparateur.

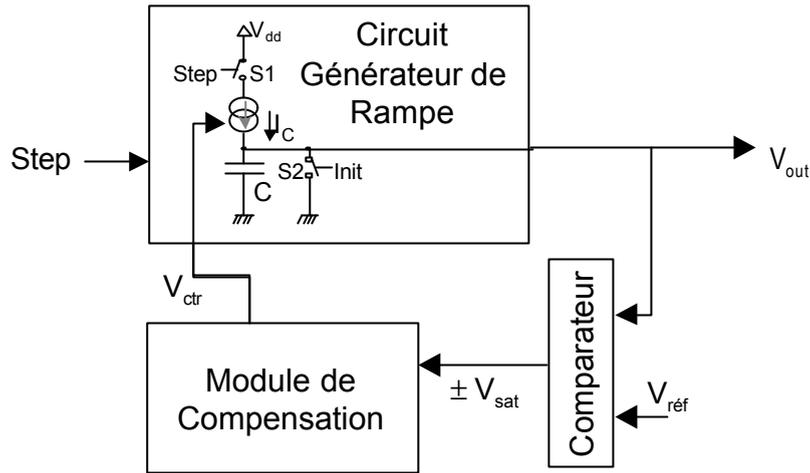


Figure 4.11 : Schéma de principe du système de compensation

III.2. Asservissement et intégration

Partant du schéma de principe du système de calibration présenté figure 4.11, Provost et Sanchez-Sinencio ont proposé une structure de générateur auto-calibré [Pro99a] [Pro99b]. L'idée consiste à calculer la valeur exacte de la variation de V_{ctr} nécessaire à compenser une erreur $V_{réf} - V_{finale}$ donnée.

Ainsi, ils ont démontré que cette valeur idéale de compensation est donnée par l'expression suivante :

$$V_{ctr} = e^{\frac{\mu C_{ox} W}{L} \frac{T_{rampe}}{C} \int (V_{réf} - V_{finale}) dt} + V_T \quad (4.15)$$

où W , L et V_T sont respectivement la largeur, la longueur et la tension de seuil du transistor générateur de courant, C est la valeur de la capacité de charge, T_{rampe} représente la durée de la rampe et V_{finale} la valeur maximale atteinte par la rampe.

Cette expression comporte des calculs de fonction exponentielle qu'il est extrêmement difficile d'intégrer sur une puce sans une augmentation importante de la surface de silicium.

Pour réduire la surface, les auteurs proposent d'utiliser une approximation du premier ordre de l'expression et obtiennent la nouvelle expression suivante :

$$V_{ctr} \cong \frac{\mu C_{ox} W}{L} \cdot \frac{T_{rampe}}{C} \int (V_{réf} - V_{finale}) dt \quad (4.16)$$

Du fait de cette approximation, la tension de contrôle ne permet pas une correction parfaite de la rampe dès la première itération. Il est par conséquent nécessaire d'effectuer plusieurs cycles pour obtenir une valeur de correction valable.

Finalement, cette solution intéressante permet d'obtenir de bons résultats en terme de précision sur la pente et de linéarité de la rampe générée. Cependant, l'intégration du système de compensation nécessite l'utilisation de deux ou trois Amplificateurs Opérationnels et de plusieurs grosses capacités. L'intégration d'une telle structure implique une surface de silicium difficilement acceptable pour une structure BIST. Partant de ce constat, notre objectif est de proposer des systèmes de calibration basés sur le même principe de rebouclage, mais dont l'intégration implique une surface de silicium plus faible.

III.3. Asservissement numérique

La première solution de système de calibration que nous avons développée est basée sur le principe d'un asservissement numérique. La figure 4.12 décrit l'implantation de cet asservissement. Le module de compensation est simplement constitué d'un compteur/décompteur et d'un Convertisseur Numérique/Analogique (CNA). L'idée consiste à calculer progressivement dans le compteur/décompteur la valeur numérique correspondant à la tension de contrôle analogique appropriée. Cette valeur est alors appliquée au générateur de rampe au travers du CNA. En pratique, plusieurs cycles de compensation sont donc nécessaires pour obtenir la tension de contrôle appropriée. A chaque cycle, le compteur/comparateur est incrémenté ou décrémenté suivant le résultat de la comparaison et la valeur numérique de sortie du compteur est convertie en tension analogique V_{ctr} par l'intermédiaire du Convertisseur Numérique/Analogique.

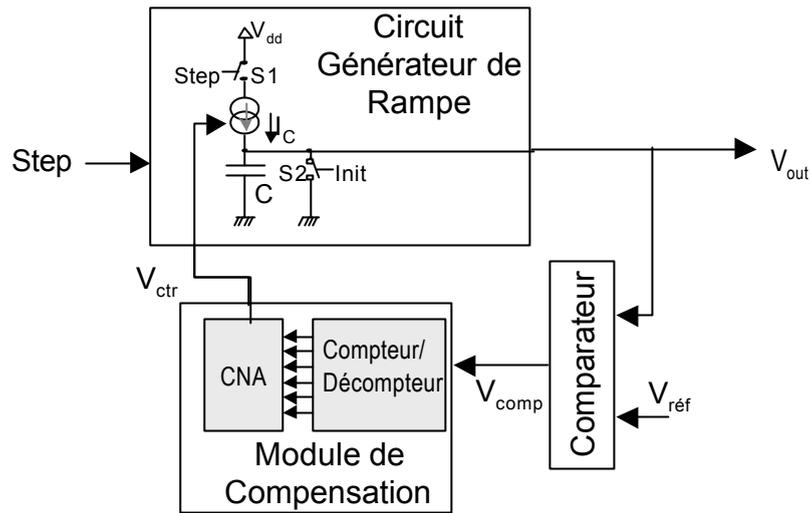


Figure 4.12 : Système de compensation par asservissement numérique

Le chronogramme présenté figure 4.13 permet d'illustrer le fonctionnement du système de calibration. Chaque cycle de compensation comprend une durée propre à la génération de la rampe T_{rampe} et une durée T_{cal} permettant la calibration. Avant de commencer la phase de calibration, le compteur est initialisé à la moitié de sa plage numérique de sortie pour permettre le maximum de dynamique de correction. Supposons qu'à la première itération, la valeur de la rampe n'atteint pas la tension de référence. La sortie du comparateur délivre un signal logique à l'état haut et le compteur est incrémenté. Après le passage au travers du CNA, la tension analogique de contrôle est par conséquent augmentée de l'équivalent de 1 LSB. Ce processus sera alors répété un certain nombre de cycles jusqu'à ce que la valeur finale de la rampe dépasse la tension de référence. La sortie du comparateur délivre dans ce cas un niveau logique bas et le compteur est décrémenté. La tension analogique de contrôle est par conséquent réduite de l'équivalent de 1 LSB. Lors du cycle suivant, la valeur finale de la rampe sera inférieure à la tension de référence et le compteur sera incrémenté. Ainsi, en fin de procédure de calibration, la tension analogique de contrôle V_{ctr} oscille autour de la valeur désirée avec une amplitude d'oscillation égale au LSB de l'association compteur-CNA.

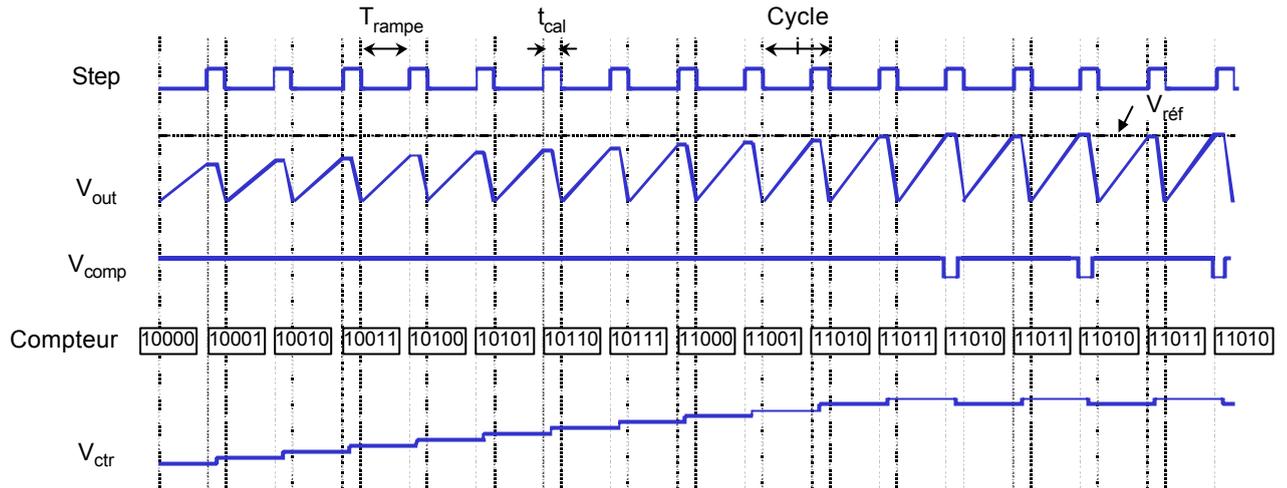


Figure 4.13 : Chronogramme du système de calibration par asservissement numérique

Le contexte BIST de nos travaux impose des contraintes très strictes en terme de surface. Il est par conséquent primordial d'obtenir une surface réduite pour l'implantation de la structure de générateur de rampe auto-calibré présenté précédemment. Le compteur/décompteur est une structure numérique classique qui peut être intégrée à faible coût. Concernant le CNA, il existe un grand nombre de structures possibles pour implanter ce bloc. Une solution intéressante du point de vue de la surface nécessaire est le CNA série, appelé aussi CNA cyclique. Le principe de ce type de convertisseur consiste à traiter successivement avec la même structure chaque bit du mot numérique à convertir, permettant ainsi de limiter les ressources matérielles nécessaires. De ce fait, le schéma du CNA série présenté figure 4.14 n'est constitué que de deux capacités et de quelques interrupteurs.

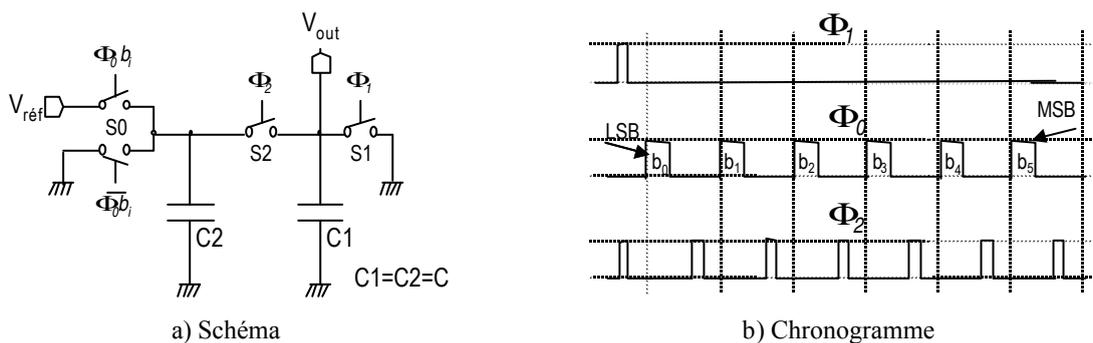


Figure 4.14 : Convertisseur Numérique/Analogique série

Le principe de fonctionnement de ce type de convertisseur est basé sur une succession de transferts de charges entre deux capacités identiques $C1$ et $C2$ commandés par la valeur des différents bits du code numérique à convertir. La première étape consiste à décharger les

capacités C1 et C2 en fermant les deux commutateurs S1 et S2. Ensuite la capacité C1 est ou à la masse en fonction de la valeur du LSB b_0 qui commande l'interrupteur S0. Cette précharge effectuée, le commutateur S0 s'ouvre à nouveau, isolant le système. Quand l'interrupteur S2 se referme, les charges se répartissent entre les deux capacités C1 et C2. Enfin, la conservation de la charge totale et l'égalité des tensions aux bornes des deux capacités permet d'exprimer la tension de sortie V_{out} en fonction de $V_{réf}$ et de b_0 :

$$V_{out} = b_0 \cdot \frac{V_{réf}}{2} \quad (4.17)$$

Cette valeur correspond à la conversion d'un bit du signal numérique. La conversion des bits suivants est faite en effectuant le même cycle, mais sans décharge initiale de la capacité C1. Ainsi, après avoir converti le bit de poids fort b_{n-1} (MSB pour Most Significant Bit), la tension de sortie sera donnée par l'expression:

$$V_{out} = V_{réf} \cdot \left(\sum_{i=0}^{n-1} b_i \cdot \frac{1}{2^{n-i}} \right) \quad (4.18)$$

Ce type de convertisseur se révèle très intéressant pour notre module de compensation de par sa faible surface d'intégration. On peut d'ailleurs noter que cette surface est indépendante de la résolution du convertisseur, seul le temps de conversion augmente avec le nombre de bits du convertisseur.

Du point de vue des performances du générateur de rampe auto-calibré, la précision obtenue sur la valeur de la pente du signal généré dépend principalement de la résolution du système compteur-CNA. En effet, nous avons vu précédemment qu'en fin de calibration, la tension de contrôle analogique oscille autour de la valeur désirée avec une amplitude correspondant à 1 LSB. Par conséquent, pour obtenir une calibration précise du générateur de rampe il est nécessaire de réduire autant que possible la valeur du LSB en augmentant le nombre de bits du module de compensation. Par exemple, une valeur de dispersion de 1mV sur la tension V_{ctr} pour une tension d'alimentation de ± 2.5 V impose une résolution supérieure à 12 bits.

L'augmentation de la résolution du module de compensation implique une augmentation de surface qu'il est nécessaire de quantifier. Comme mentionné précédemment, la résolution du CNA série ne dépend pas du nombre de bits du mot numérique à traiter, mais de l'appariement ,de la valeur des injections de charges à l'ouverture des

interrupteurs S1 et S2. Par contre, la surface du compteur augmente linéairement avec le nombre de bits.

Pour donner une idée de la surface nécessaire, nous utilisons comme élément de référence un Amplificateur Opérationnel (AO) qui constitue généralement un élément de base des circuits analogiques. Dans la bibliothèque AMS 0,6 micron, un AO représente une surface de 0,018 mm². Intégrer un compteur de 12 bits dans cette même technologie nécessite une surface de 0,03 mm². Un compteur de 12 bits est donc équivalent à 1,7 AO. Cette surface additionnelle est relativement importante. Par conséquent, nous envisageons une nouvelle approche basée sur un asservissement analogique.

III.4. Asservissement analogique

Dans cette nouvelle approche de calibration par asservissement analogique, nous désirons conserver le même principe d'incrément ou décrémentation de la tension de contrôle. Toutefois, cet ajustement doit se faire directement dans le domaine analogique pour éviter l'utilisation d'un compteur. La solution proposée est illustrée sur la figure 4.15. La fonction d'incrément ou décrémentation analogique est assurée par un système de transfert de charges entre deux capacités C1 et C2, et l'idée consiste à accumuler progressivement dans la capacité C2 la quantité de charges correspondant à la valeur désirée de la tension de contrôle.

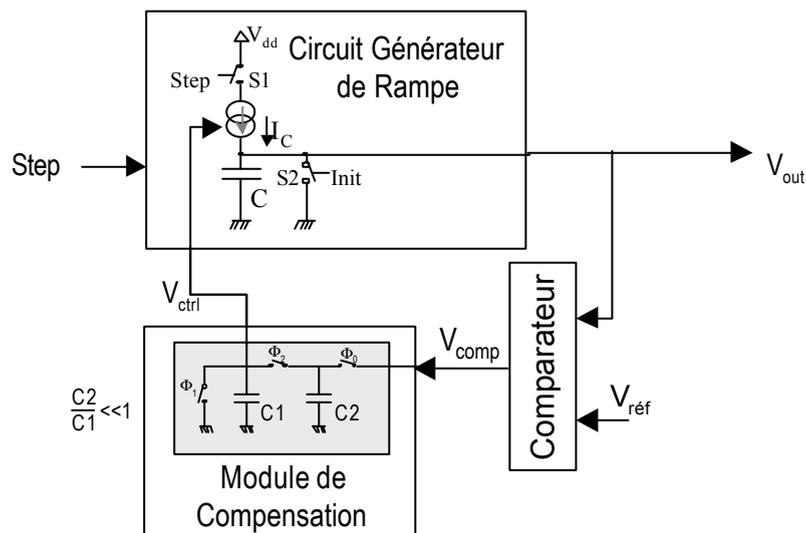


Figure 4.15 : Système de compensation par asservissement analogique

En fait le système de transfert de charges décrit figure 4.16 est similaire à celui utilisé dans la structure de CNA série. Les deux différences majeures entre ces structures concernent la valeur du rapport des capacités et le type de commande utilisée. En effet, les deux capacités ne sont plus identiques et la structure est directement commandée par le signal de sortie du comparateur.

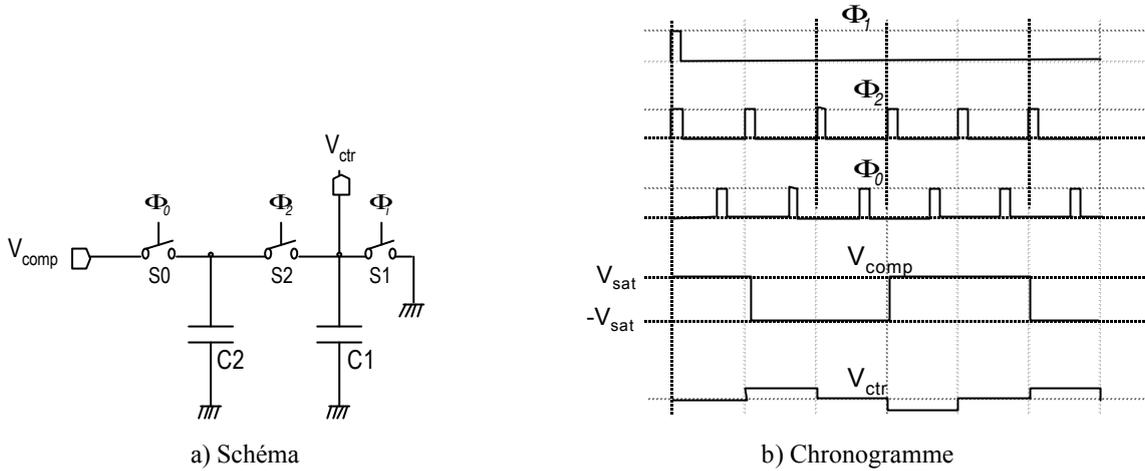


Figure 4.16 : Incrémenteur/décrémenteur analogique

Le principe de fonctionnement des interrupteurs S1 et S2 reste identique à celui utilisé dans le cas du CNA série. Ainsi, le transfert de charge décrit figure 4.16b dépend de la tension de sortie du comparateur et de la valeur de la tension V_{ctr} obtenue à l'itération précédente. L'expression de la tension de contrôle au $i^{\text{ème}}$ cycle est donnée par l'expression suivante :

$$V_{ctr}(i) = \frac{C1}{C1+C2} \cdot V_{ctr}(i-1) \pm \frac{C2}{C1+C2} \cdot V_{sat} \quad (4.19)$$

où V_{sat} correspond à la tension de saturation du comparateur.

Pour que la variation de tension de contrôle soit minimale, il faut choisir le rapport des capacités aussi petit que possible, ce qui impose alors la condition: $\frac{C2}{C1} \ll 1$. En utilisant cette condition nous pouvons simplifier l'expression (4.19) et obtenir la nouvelle expression de la tension de contrôle :

$$V_{ctr}(i) \cong V_{ctr}(i-1) \pm \frac{C2}{C1} \cdot V_{sat} \quad (4.20)$$

Cette expression traduit clairement qu'à chaque itération, la tension de contrôle délivrée par le module de compensation est ajustée d'un incrément constant $\Delta V_{\text{ctr}} = \frac{C_2}{C_1} \cdot V_{\text{sat}}$. La valeur de cet incrément dépend uniquement du rapport des capacités et de la valeur de la tension de saturation du comparateur.

Le principe de fonctionnement du système d'asservissement analogique est similaire à celui du système d'asservissement numérique. Plusieurs itérations sont nécessaires pour corriger une erreur sur la pente de la rampe générée. Dans chaque cycle, la tension de contrôle est décrémentée) si la tension atteinte par la rampe est inférieure (respectivement supérieure) à la tension de référence imposée. A la fin des cycles de calibration, la tension de contrôle V_{ctr} oscille autour de la valeur idéale de correction. Comme dans le cas de l'asservissement numérique, la précision de la correction dépend de l'amplitude des oscillations de la tension V_{ctr} , c'est à dire de la valeur de l'incrément ΔV_{ctr} . Cet incrément étant proportionnel au rapport des capacités C_2/C_1 , il est nécessaire de choisir un rapport aussi réduit que possible pour limiter l'amplitude des oscillations. Par exemple, une précision de 1 mV sur la tension de contrôle impose un rapport de capacités inférieur à $4 \cdot 10^{-4}$ dans le cas d'une tension d'alimentation égale à $\pm 2,5$ Volt.

I.5. Implantation du générateur de rampe auto-calibré

Ce paragraphe est consacré à l'implantation du générateur de rampe auto-calibré basé sur un asservissement analogique. La structure finale du générateur de rampe auto-calibré (figure 4.17) est constitué de trois blocs distincts :

- le générateur de rampe présenté à la section II,
- le module de compensation,
- un comparateur.

Le comparateur n'a pas besoin d'être rapide puisque la durée de calibration T_{cal} peut être réglée en fonction du temps de réponse du comparateur. Nous choisissons par conséquent un simple Amplificateur Opérationnel de bibliothèque AMS 0,6 μm utilisé en comparateur. Concernant le module de compensation, l'objectif est d'obtenir une précision maximale sur la valeur de la pente de la rampe en impliquant une surface minimale de silicium. La précision sur la pente de la rampe générée dépend directement du rapport des C_2/C_1 . Afin

d'implanter un très faible rapport capacitif tout en conservant une surface minimale, nous suggérons de n'utiliser pour la capacité C2 que la capacité parasite équivalente présente entre les deux portes de transmission M14-15 et M16-17. La valeur de cette capacité dépend bien sûr de la technologie utilisée. Pour la technologie AMS 0,6 μm et compte tenu de la taille des transistors considérés, cette capacité parasite équivalente est de l'ordre de quelques fF. Ainsi en choisissant une capacité C1 égale à 10pF, nous sommes sûrs d'obtenir un rapport de 10^{-4} , et donc une dispersion inférieure à 1mV sur la valeur de la tension

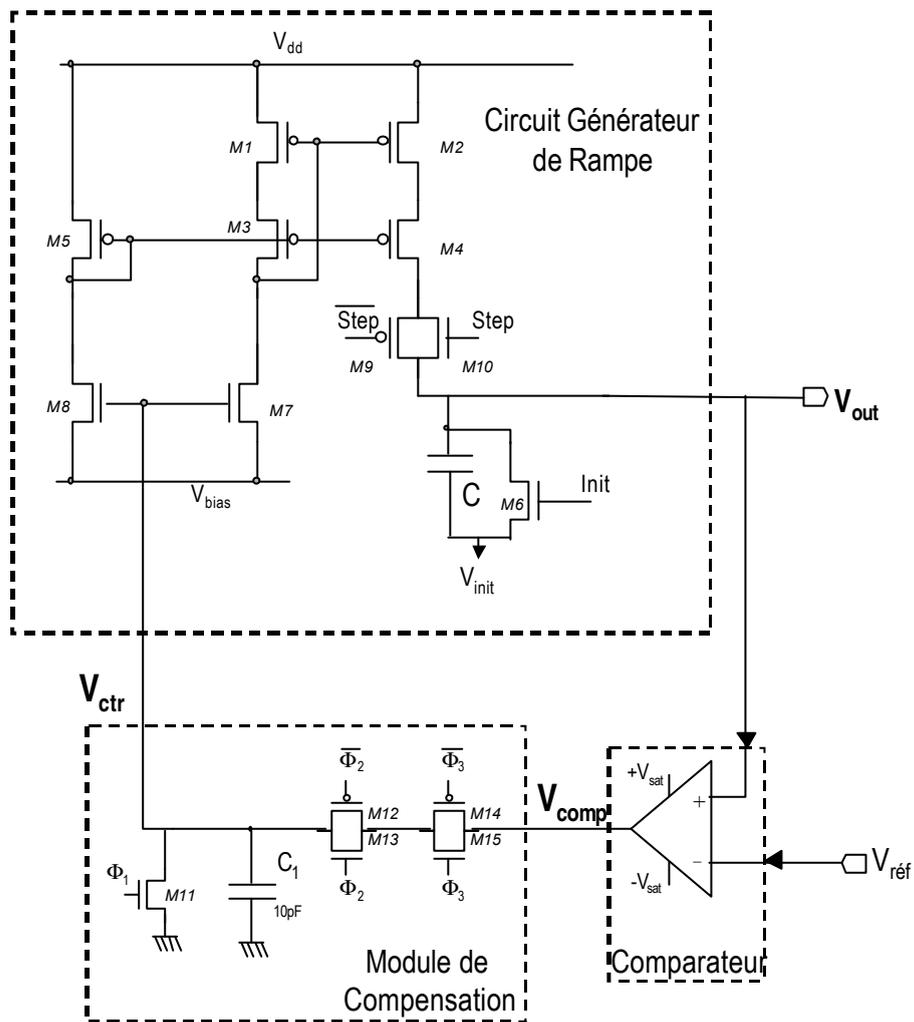


Figure 4.17 : Implantation du générateur de rampe auto-calibré

Pour valider la solution proposée, un grand nombre de simulations ont été effectuées en considérant des variations extrêmes des paramètres du circuit (température, taille des transistors, valeur de capacité de charge, valeur du courant). Dans tous les cas de figures considérés, le système de calibration permet de compenser ces dispersions avec un nombre

réduit de cycles de calibration. Pour illustration, la figure 4.18 représente un exemple de simulation dans le cas d'une variation sur la valeur de la capacité de charge. Nous pouvons remarquer que le système de calibration permet de corriger cette erreur en moins de 10 cycles.

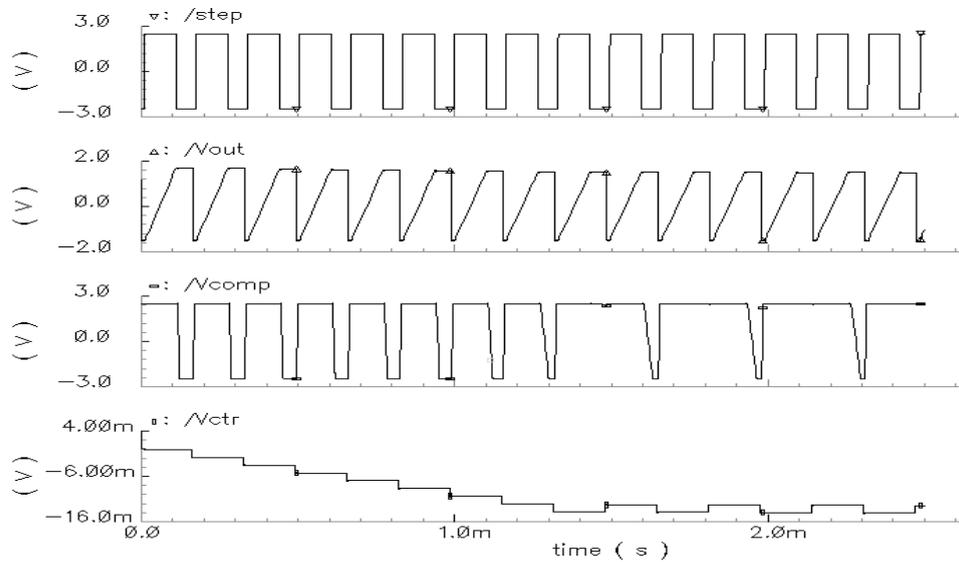


Figure 4.18 : Simulation du générateur de rampe auto-calibré

Finalement, nous avons implanté le circuit en technologie AMS 0,6 μm . Sur le layout du circuit présenté figure 4.19, nous pouvons retrouver les différents éléments constituant le générateur de rampe auto-calibré : la capacité de charge du circuit générateur de rampe, la capacité C1 du module de compensation, l'amplificateur opérationnel et les divers transistors

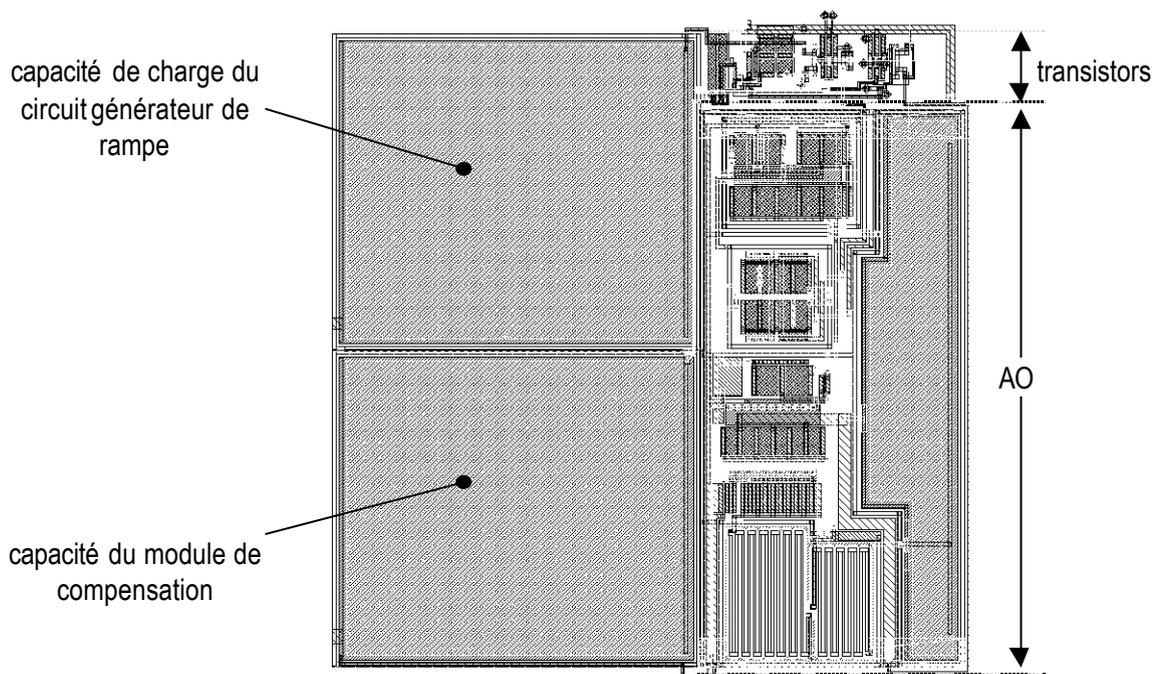


Figure 4.19 : Layout du générateur de rampe auto-calibré

I.6. Performances et discussion

Rappelons que les principaux critères de qualité d'un générateur de rampe utilisé dans le contexte du test intégré sont d'une part, la surface du générateur et d'autre part, la linéarité et

a) Surface

La structure finale du générateur de rampe auto-calibré implanté en technologie $0,6\ \mu\text{m}$ représente une surface de $0,047\ \text{mm}^2$. Le tableau 4.2 récapitule la surface occupée par les différents composants du générateur.

	Circuit générateur de rampe		Module de compensation		Comparateur	Générateur de rampe auto-calibré
	Capacité de Charge	Circuit Complet	Capacité Ccal	Circuit Complet		
Surface (μm^2)	11750	15510	11750	14100	17390	47000
Surface (%)	25%	33%	25%	30%	37%	100%

Tableau 4.2 : Surface des éléments constituant le générateur de rampe

L'analyse de ce tableau conduit à plusieurs observations. Tout d'abord, nous pouvons remarquer que la capacité de charge du circuit générateur de rampe et la capacité du module de compensation représentent à elles deux la majeure partie de la surface du générateur. La deuxième remarque concerne la surface totale du module de compensation. Cette surface reste extrêmement faible puisqu'elle correspond simplement à 0,8 fois la surface d'un Amplificateur Opérationnel classique. Enfin, nous pouvons observer que la surface de l'AO utilisé en comparateur représente une part importante de la surface totale du circuit. Néanmoins, il faut noter que nous avons utilisé dans cette première implantation un AO standard de bibliothèque. Nous pouvons donc espérer réduire sensiblement la surface du générateur de rampe auto-calibré en développant une structure de comparateur dédiée à l'application.

b) Linéarité et précision sur la pente de la rampe générée

➤ **Linéarité**

Diverses simulations électriques ont été effectuées sur la structure finale du générateur de rampe auto-calibré afin de vérifier que la circuiterie additionnelle de la structure de compensation n'introduit pas de dégradation de performance significative en terme de linéarité. Les résultats de ces simulations sont résumés sur la figure 4.20. La non-linéarité maximale mesurée est égale à $98 \mu\text{V}$. Cette valeur est très proche de celle mesurée pour le générateur de rampe sans système de calibration (NL maximum = $80 \mu\text{V}$), démontrant ainsi que le système de calibration ne perturbe que très légèrement la qualité du signal.

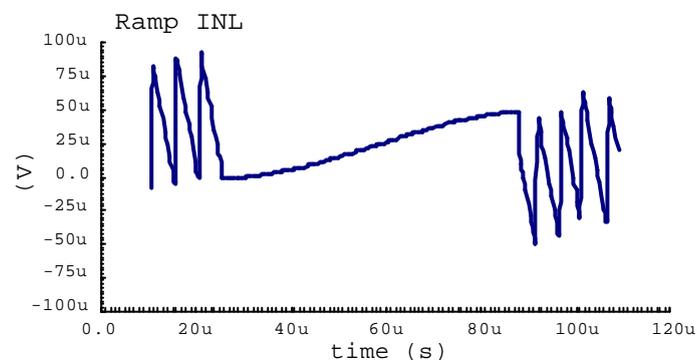


Figure 4.20 : Non-linéarité de la rampe générée par la structure auto-calibrée

➤ **Précision de la pente**

La précision sur la pente que permet d'obtenir la structure de calibration a aussi été évaluée plus finement. Un grand nombre de simulations a été réalisé en prenant en compte les différentes sources d'erreur possibles telles que la présence d'une erreur d'offset au niveau du comparateur ou les éventuelles fluctuations technologiques du processus de fabrication.

Dans le cas le plus défavorable, le système de calibration permet effectivement d'obtenir une précision sur la tension de contrôle inférieure à 1mV . L'erreur maximale mesurée entre la valeur finale de la rampe et la tension de référence reste dans tous les cas inférieure à 19mV , ce qui correspond à une erreur de $0,6\%$ sur la pente de la rampe. Le résultat démontre l'efficacité du système de calibration qui permet de rendre le générateur de rampe insensible aux variations technologiques du processus de fabrication.

I.7. Application au test intégré d'un CAN

De par ses qualités de linéarité et précision sur la pente, le générateur de rampe auto-calibré constitue une excellente base en vue du développement d'une solution de test intégré complète dédiée au test des convertisseurs A/N. Néanmoins, il est nécessaire de réaliser une légère adaptation au générateur pour prendre en compte les spécificités du test par histogramme. En effet, nous avons vu au chapitre 2 que l'analyseur intégré de réponses de test collecte les échantillons en sortie du convertisseur sur plusieurs périodes du signal d'entrée. Il est donc nécessaire que le générateur délivre une succession continue de rampes sans périodes de

Afin de gérer cette contrainte supplémentaire, nous proposons d'adapter les signaux de commande du générateur de façon à inhiber le processus de calibration lors de la phase de test du convertisseur, la calibration étant réalisée dans une phase initiale.

Durant la phase initiale de calibration, chaque période du signal délivré par le générateur rampe propre à la génération de la rampe définie tant que le signal de commande *Step* est à l'état logique haut, et une durée T_{cal} propre à la calibration du système définie tant que le signal de commande *Step* est à l'état logique bas.

Une fois la calibration terminée, le générateur est utilisé en mode test en inhibant les signaux de commande du module de compensation Φ_2 et Φ_3 et en maintenant le signal de commande *Step* à l'état logique haut pour toute la durée du test. Le générateur délivre alors une succession continue de rampes dont la période est simplement définie par la période du signal de commande *Init* contrôlant l'initialisation successive des rampes.

L'association du générateur de rampe auto-calibré et de l'analyseur de réponses de test décrit au chapitre 3 permet ainsi de proposer une structure BIST complète dont schéma de principe

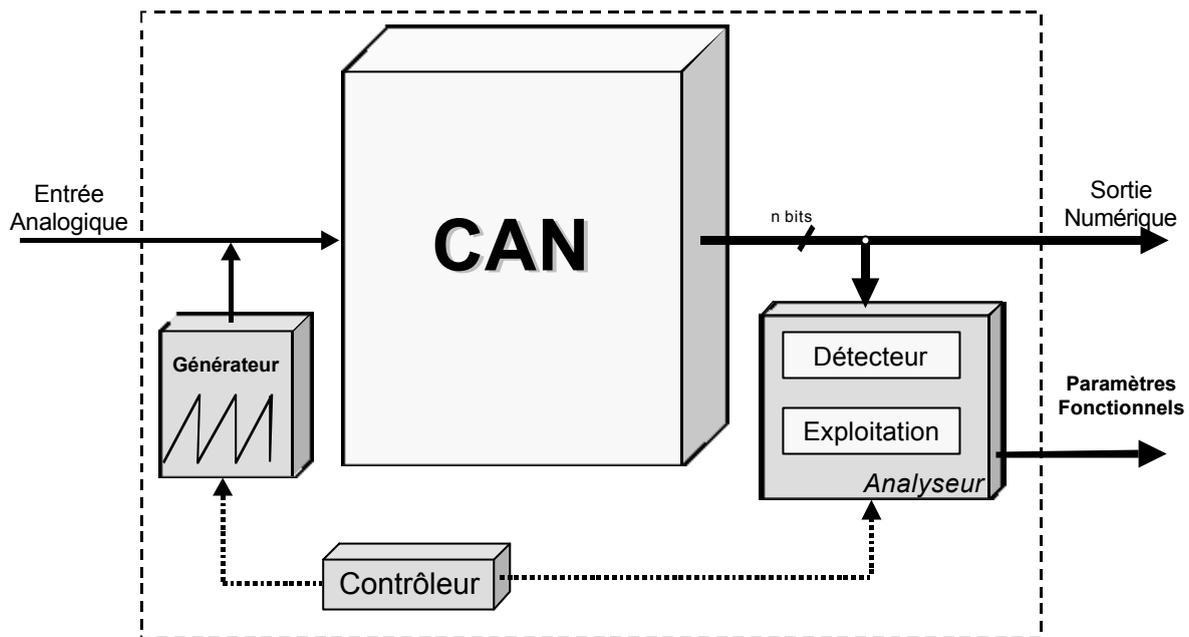


Figure 4.21 : Structure BIST globale

Dans cette structure BIST, un contrôleur unique permet de gérer l'ensemble du test. Ce contrôleur gère dans un premier temps la phase de calibration initiale du générateur permettant de corriger une éventuelle erreur sur la pente des rampes générées. Une fois la calibration terminée, le test proprement dit du convertisseur peut commencer. Le contrôleur gère alors les différentes phases de détermination des paramètres fonctionnels du CAN.

a) Surface de la solution BIST complète

Afin d'estimer la surface nécessaire à l'intégration de la solution BIST complète, nous avons réalisé l'implantation de la structure en technologie AMS 0,6 μm pour le test d'un convertisseur 6 bits. Le générateur de rampes correspond simplement à la structure définie au paragraphe précédent. L'analyseur de réponses de test est composé du détecteur de code comportant 6 blocs élémentaires et du module d'exploitation comportant 10 blocs élémentaires comme défini au chapitre 3. Le contrôleur commun gérant l'ensemble de la procédure de test a été synthétisé automatiquement à partir d'une description VHDL prenant en compte les nouvelles fonctionnalités liées à la gestion du générateur. Le tableau 4.2 résume l'ensemble des résultats obtenus et permet de comparer la surface de la structure BIST avec celle d'un convertisseur 6 bits de la bibliothèque AMS 0,6 μm .

	Surface	Surface BIST / Surface CAN
Analyseur de réponses de test (sans le contrôleur)	0,063 mm ²	< 6 %
Générateur de rampes auto-calibré	0,047 mm ²	
Contrôleur	0,065 mm ²	
Structure BIST complète	0,175 mm²	
CAN	3,3 mm²	

Tableau 4.3 : Surface de la structure BIST

Nous pouvons remarquer que pour le cahier des charges retenu, la structure BIST complète est tout à fait viable dans une optique de test intégré puisque sa surface est inférieure à 6 % de

b) Performances de la solution BIST complète

Afin d'évaluer les performances de la structure BIST, nous étudions l'histogramme mesuré en sortie d'un convertisseur parfait lorsque le générateur intégré délivre le stimulus de test appliqué au circuit. Nous pourrions ainsi estimer la linéarité équivalente du stimulus de test

Nous devons tout d'abord préciser les conditions de l'étude. Soit un convertisseur parfait échantillonné à 20 MHz avec une résolution de 10 bits et une dynamique d'entrée de 3V. On peut appliquer à ce convertisseur un test par histogramme en utilisant 25750 échantillons collectés à la fréquence de fonctionnement nominale.

La première étape consiste à choisir l'amplitude et la période du stimulus de test. Classiquement, l'amplitude est choisie légèrement supérieure à la dynamique d'entrée du convertisseur. Dans notre cas, nous fixons l'amplitude du stimulus de test à 3,2 V, ce qui impose $V_{\text{init}} = -1,6 \text{ V}$ et $V_{\text{réf}} = +1,6 \text{ V}$ pour le circuit de génération. Concernant la période du stimulus de test, cette période doit respecter la condition d'échantillonnage cohérent définie par la relation :

$$\frac{F_{in}}{F_{éch}} = \frac{N_p}{N_T} \quad (1.1)$$

où N_p et N_T sont premiers entre eux. F_{in} , $F_{éch}$, N_p et N_T représentent respectivement la fréquence du stimulus du test, la fréquence d'échantillonnage, le nombre de périodes du signal d'entrée constituant une trame de test et le nombre total d'échantillons collectés durant la trame de test. De plus, le générateur de stimulus de test a été conçu pour fonctionner à une fréquence proche de 10 kHz. La relation (1.1) impose donc $N_p = 13$ et $F_{in} = 10097$ Hz, ce qui correspond à une période $T_{rampe} = 99 \mu s$.

Les caractéristiques principales (amplitude et période) du stimulus de test étant déterminées, la deuxième étape consiste à calibrer le générateur avec ces valeurs de référence. La figure 4.22 illustre une trame de test constituée de 13 périodes du signal délivré par le générateur une fois la calibration terminée.

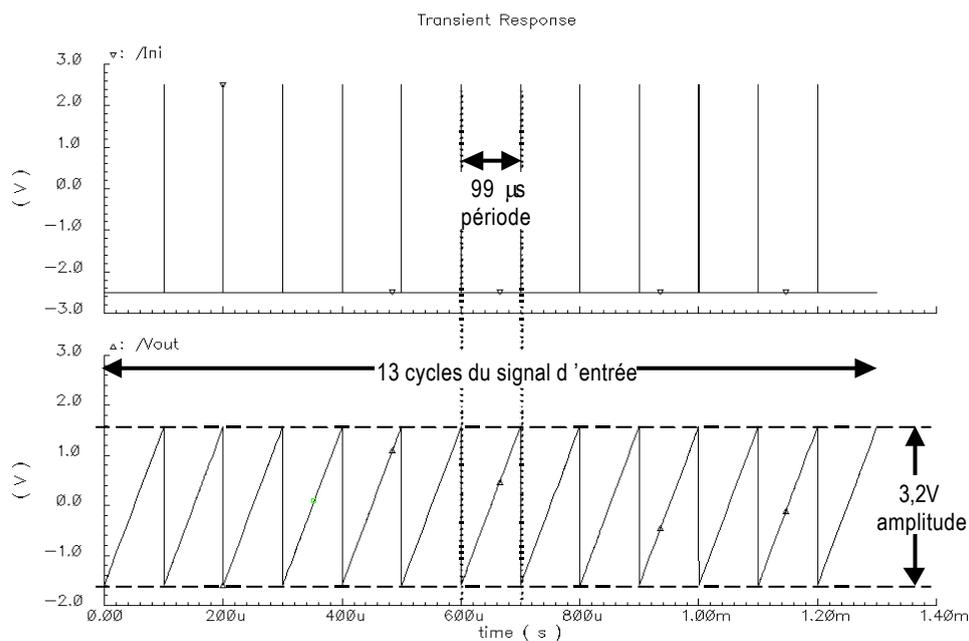


Figure 4.22 : Trame du signal utilisé comme stimulus de test

Finalement, la dernière étape consiste à évaluer l'histogramme obtenu en appliquant cette trame de test à un convertisseur parfait. Le programme d'évaluation présenté au chapitre 3 est utilisé pour cette étude. Les résultats obtenus sont résumés sur la figure 4.23 qui donne l'histogramme mesuré ainsi que les NLD et NLI associées. Les non-linéarités maximales

mesurées sont de l'ordre de 0,25 LSB, ce qui correspond à une résolution équivalente de 12 bits pour la trame de test. Cette résolution est bien inférieure à celle déterminée lors de l'étude d'une rampe unique. Cette dégradation de performance est en fait principalement due à la présence d'une phase d'initialisation dans chaque période du signal généré. En effet, le courant de décharge de la capacité C n'est pas constant durant cette phase. Le signal généré n'est donc plus linéaire surtout pour la fin de la décharge correspondant à une tension proche de V_{init} .

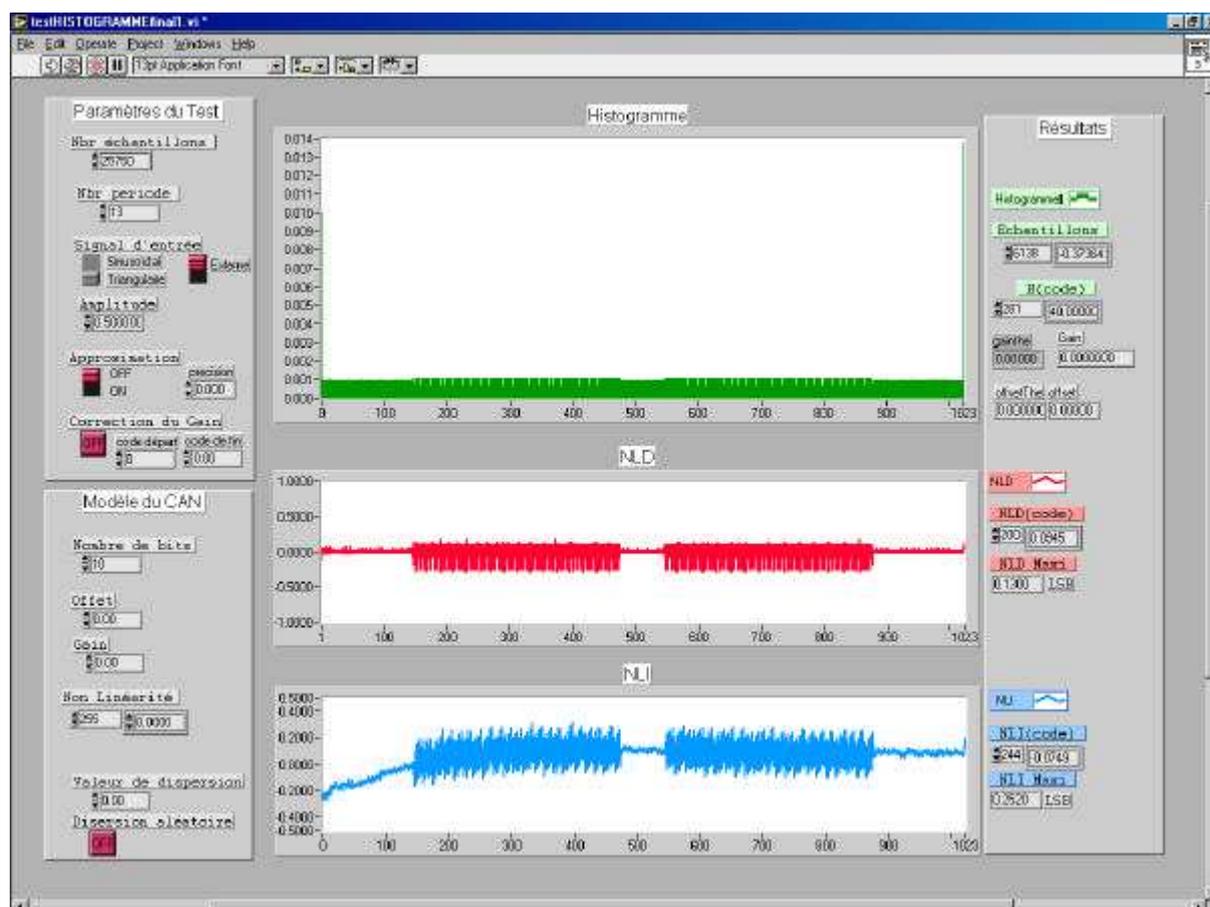


Figure 4.23 : Linéarité du signal d'entrée estimée par la technique de test par histogramme

Généralement, il est admis que le stimulus de test doit présenter une résolution supérieure de 2 bits au moins à celle du CAN sous test. Par conséquent, le générateur de rampe auto-calibré présenté dans ce chapitre permet de proposer une structure BIST complète pour le test de convertisseurs de résolution inférieure à 11 bits. Pour augmenter le domaine d'application de notre technique de test intégré, nous envisageons d'utiliser un nouveau type de générateur ne : le générateur de signaux triangulaires.

IV. Générateur de triangle auto-calibré

IV.1. Générateur classique

L'objectif est de générer en interne un signal triangulaire. Pour générer ce signal sur le même principe de charge d'une capacité avec un courant constant, il est nécessaire d'utiliser deux générateurs fournissant des courants opposés. En pratique, nous utilisons deux générateurs de courant en parallèle de valeur $-I$ et $+2I$. L'avantage de ce système est qu'il permet de limiter les parasites de commutation en maintenant l'un des générateurs toujours conducteur. La structure du générateur de signaux triangulaires est donnée figure 4.24a.

Le signal qui relie les deux générateurs de courant est directement commandé par la sortie d'un trigger de Schmitt dont les tensions de seuil $\pm V_{\text{seuil}}$ sont définies par l'expression :

$$V_{\text{seuil}} = \frac{V_{\text{sat}} \cdot R1}{R1 + R2} \quad (4.20)$$

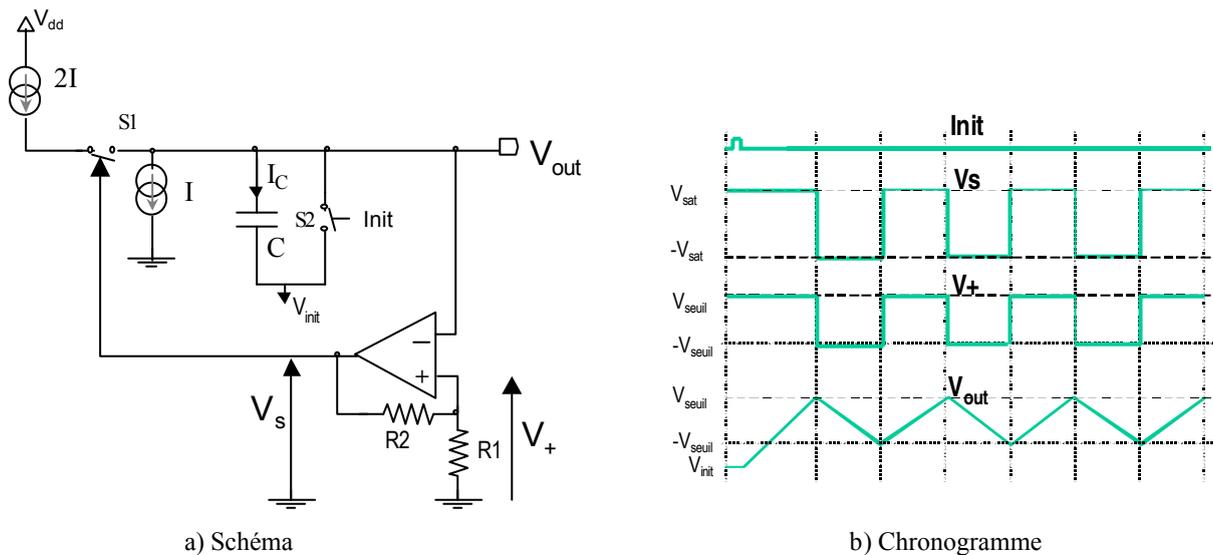


Figure 4.24 : Générateur de signaux triangulaires

Le chronogramme de la figure 4.25b illustre le fonctionnement interne du générateur. La première phase consiste à initialiser la sortie à la tension V_{init} par l'intermédiaire de l'interrupteur S2. Or, si on suppose que la tension V_{init} est inférieure à la tension présente sur l'entrée non-inverseuse du trigger, la sortie du trigger est égale à $+V_{\text{sat}}$ et l'interrupteur S1 est fermé. De ce fait, la capacité C est chargée par un courant $+I$ et le générateur délivre une rampe de pente positive $+I/C$ (rampe montante). Quand la tension de sortie atteint la tension de seuil $+V_{\text{seuil}}$, le trigger commute à $-V_{\text{sat}}$ et ouvre l'interrupteur S1. La capacité C est alors

déchargée par un courant $-I$ et le générateur délivre une rampe de pente négative $-I/C$ (rampe descendante). Lorsque la tension de sortie devient égale à $-V_{\text{seuil}}$, le trigger commute à nouveau impliquant la charge de la capacité C par un courant $+I$ et donc la génération d'une rampe montante. Ainsi, la répétition de ce processus correspond à la génération d'un signal triangulaire compris entre les deux tensions de seuil prédéterminées $+V_{\text{seuil}}$ et $-V_{\text{seuil}}$.

On peut noter que le trigger de Schmitt assure une double fonction. D'une part il détermine l'amplitude du signal généré par l'intermédiaire de ses tensions de seuil. D'autre part, il assure la commande de l'interrupteur définissant la commutation d'une rampe montante à une rampe descendante et vice versa.

IV.2. Générateur auto calibré

Pour les mêmes raisons que dans le cas du générateur de rampe, les variations technologiques lors de la fabrication rendent indispensable l'utilisation d'un système de calibration. La structure du générateur de signaux triangulaires auto-calibré présenté figure 4.25 utilise un module de compensation basé sur l'incrémenteur/décrémenteur analogique défini à la section III.

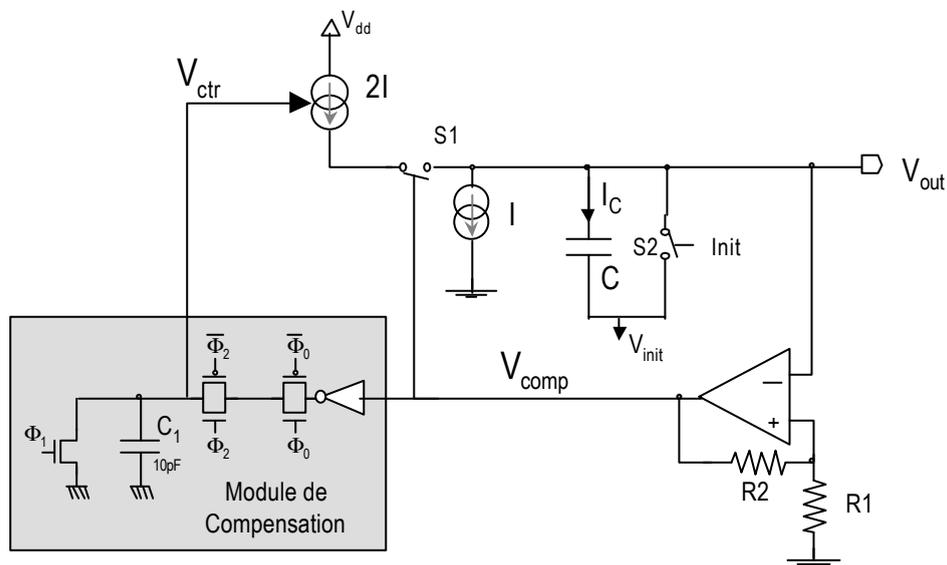


Figure 4.25 : Générateur de rampe auto-calibré

Le principe de fonctionnement du système de compensation est cependant légèrement différent de celui du générateur de rampe auto-calibré. L'idée consiste ici à ajuster la valeur du générateur de courant $+2I$ pour qu'une période complète du signal triangulaire soit générée en

un temps donné T_{triangle} . Il s'agit par conséquent d'augmenter (respectivement de diminuer) la pente de la rampe montante du signal si la période du signal généré est supérieure à T_{triangle} .

En pratique, l'observation de la sortie du trigger permet d'avoir une information sur le signe de la pente de la rampe générée à tout instant. Nous avons vu que le montage délivre une rampe montante lorsque la sortie du trigger est à $+V_{\text{sat}}$, et une rampe descendante lorsque la sortie du trigger est égale à $-V_{\text{sat}}$. L'observation de la sortie du trigger au temps T_{triangle} permet donc de déduire l'information relative à la période du signal triangulaire généré. En effet, supposons que cette période du signal soit supérieure à T_{triangle} . Dans ce cas, le montage n'a pas le temps de délivrer une période complète du signal. Par conséquent, la pente de la rampe T_{triangle} est négative et la tension de sortie du trigger est égale à $-V_{\text{sat}}$. De même, si la période du signal généré est inférieure à T_{triangle} , une période complète du signal triangulaire aura été générée et le montage aura recommencé la génération d'une rampe montante. La tension de sortie du trigger est dans ce cas égale à $+V_{\text{sat}}$.

Dans ce contexte, le rôle du module de compensation consiste à incrémenter la tension v_{ctr} , c'est à dire augmenter la pente de la rampe montante et donc réduire la période du signal généré, si la sortie du trigger est égale à $-V_{\text{sat}}$ au temps T_{triangle} . De même, le module de compensation doit décrémenter la tension de contrôle si la sortie du trigger est égale à $+V_{\text{sat}}$ de façon à augmenter la période du signal généré. Le module de compensation est donc simplement constitué d'un inverseur précédant l'incrémenteur/décrémenteur analogique défini à la section III.

Le fonctionnement du système de calibration est illustré sur la figure 4.26. Chaque cycle de la procédure débute par une phase d'initialisation de la capacité de charge à la tension $-V_{\text{seuil}}$ ($V_{\text{init}} = -V_{\text{seuil}}$). Si la période du signal triangulaire est supérieure à la valeur imposée T_{triangle} , la sortie du trigger est égale à $-V_{\text{sat}}$ au temps T_{triangle} et la tension de contrôle est incrémentée par le module de compensation, impliquant une augmentation de la pente de la rampe montante du signal triangulaire. Finalement, après plusieurs cycles de correction, la période du signal triangulaire converge vers la valeur T_{triangle} .

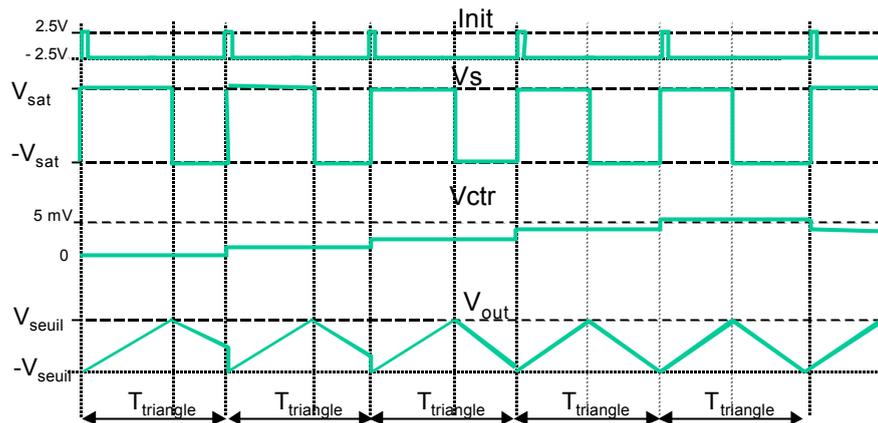


Figure 4.26 : Fonctionnement du système de calibration

A ce stade, nous pouvons remarquer que le fait de modifier un seul des deux générateurs de courant ne nous permet pas d'assurer la symétrie du signal triangulaire généré. En fait, ce générateur de signaux triangulaires est conçu pour être utilisé en générateur de stimuli dédié au test par histogramme des convertisseurs analogique/numérique. Or, l'histogramme résultant d'un signal triangulaire ne dépend que de la période T_{triangle} et non pas de la symétrie du signal.

IV.3. Validations

Le générateur de signaux triangulaires auto-calibré a été implanté au niveau transistor en utilisant des miroirs de courant "grande excursion" pour les deux générateurs de courant (figure 4.27). Le dimensionnement des transistors de la structure est donné en annexe 4.

Pour l'implantation du trigger de Schmitt, un amplificateur opérationnel standard de bibliothèque a été utilisé et les deux résistances R1 et R2 sont des résistances externes très précises. En effet, l'amplitude du signal triangulaire dépend directement des seuils de commutation du trigger définis par la valeur des résistances. Afin d'obtenir une bonne précision sur l'amplitude du signal, il est préférable de ne pas intégrer ces résistances.

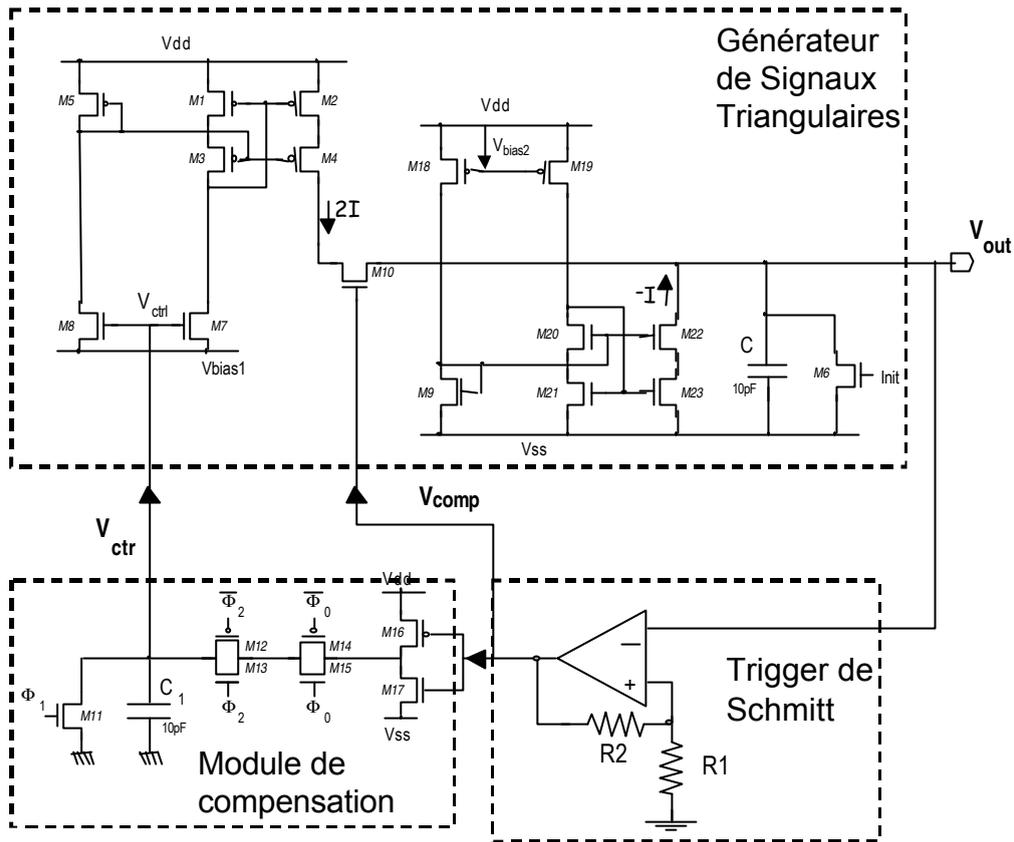


Figure 4.27 : Implantation du générateur de signaux triangulaires auto-calibré

L'implantation physique finale du générateur de signaux triangulaires auto-calibré n'a pas été réalisée. Il est toutefois possible d'obtenir une estimation de la surface du module. Comme dans le cas du générateur de rampe auto-calibré, la majeure partie de la surface sera consacrée à l'intégration des deux capacités (de charge et de compensation) et de l'amplificateur opérationnel du trigger, les différents transistors de la structure de contribuant que pour une faible part à la surface totale. Nous pouvons ainsi espérer obtenir une surface du même ordre de grandeur que pour le générateur de rampe, soit une surface proche de $0,05 \text{ mm}^2$.

Concernant la validation fonctionnelle de la structure, la figure 4.28 présente les résultats de simulation du générateur de signaux triangulaires en phase de calibration. Comme dans le cas du générateur de rampe auto-calibré, nous observons l'ajustement progressif de la tension de contrôle durant les premiers cycles, puis l'oscillation de cette tension autour de la valeur appropriée une fois la calibration achevée. La précision obtenue sur la période du signal triangulaire généré dépend bien sûr de l'amplitude de ces oscillations.

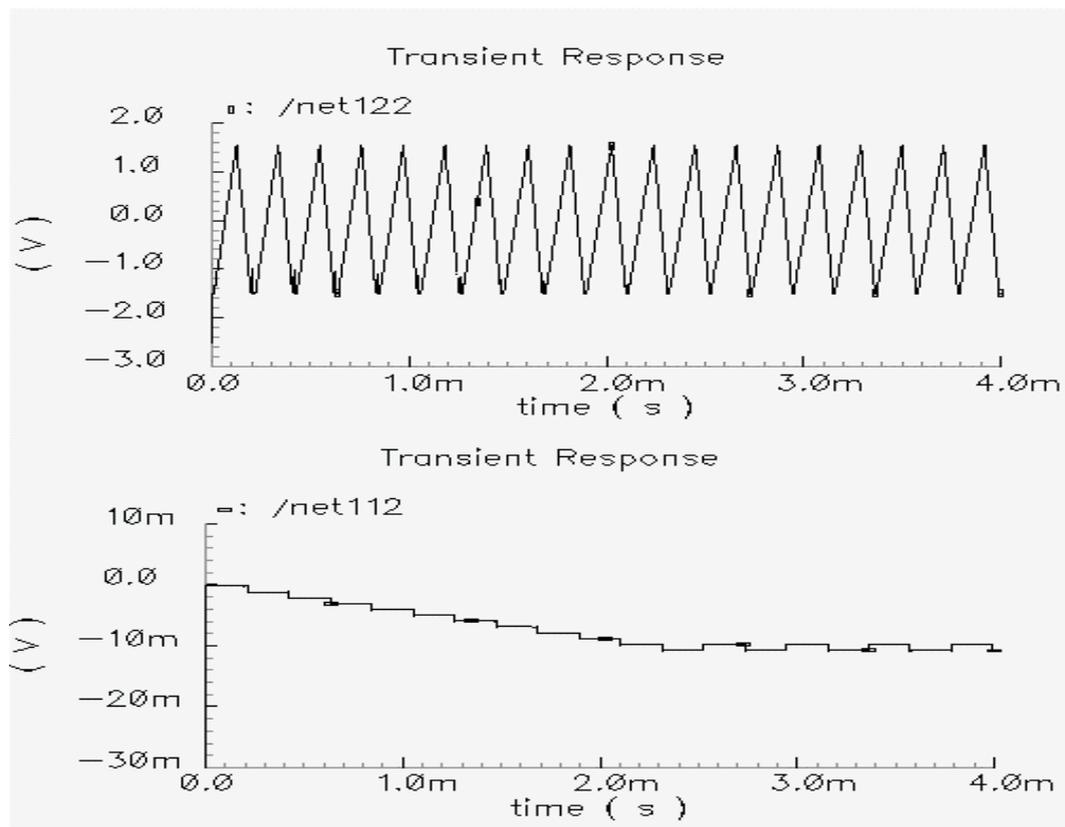


Figure 4.28 : Simulation du générateur de triangle auto-calibré

Finalement, les performances de la structure utilisée comme générateur de stimulus de test ont été évaluées au travers de l'analyse de l'histogramme mesuré en sortie d'un convertisseur parfait. Afin de pouvoir comparer les résultats avec ceux du générateur de rampe auto-calibré, les mêmes conditions expérimentales ont été utilisées :

- convertisseur parfait échantillonné à 20 MHz avec une résolution de 10 bits et une dynamique d'entrée de 3V,
- histogramme construit à partir de 25750 échantillons collectés à fréquence nominale en utilisant un échantillonnage cohérent,
- trame de test constituée de 13 périodes du signal d'entrée calibré pour une amplitude de 3.2 V est une période de 99 μ s.

la figure 4.30 illustre l'histogramme mesuré en sortie du convertisseur ainsi que les NLD et NLI associées. Les non-linéarités maximales mesurées sont de l'ordre de 0.12 LSB, ce qui correspond à une résolution équivalente de 13 bits pour les trames de test délivrées par le générateur de signaux triangulaires auto-calibré. le générateur de triangle que nous avons développé permet de proposer une structure BIST complète dédiée au test des convertisseurs de résolution inférieure à 12 bits.

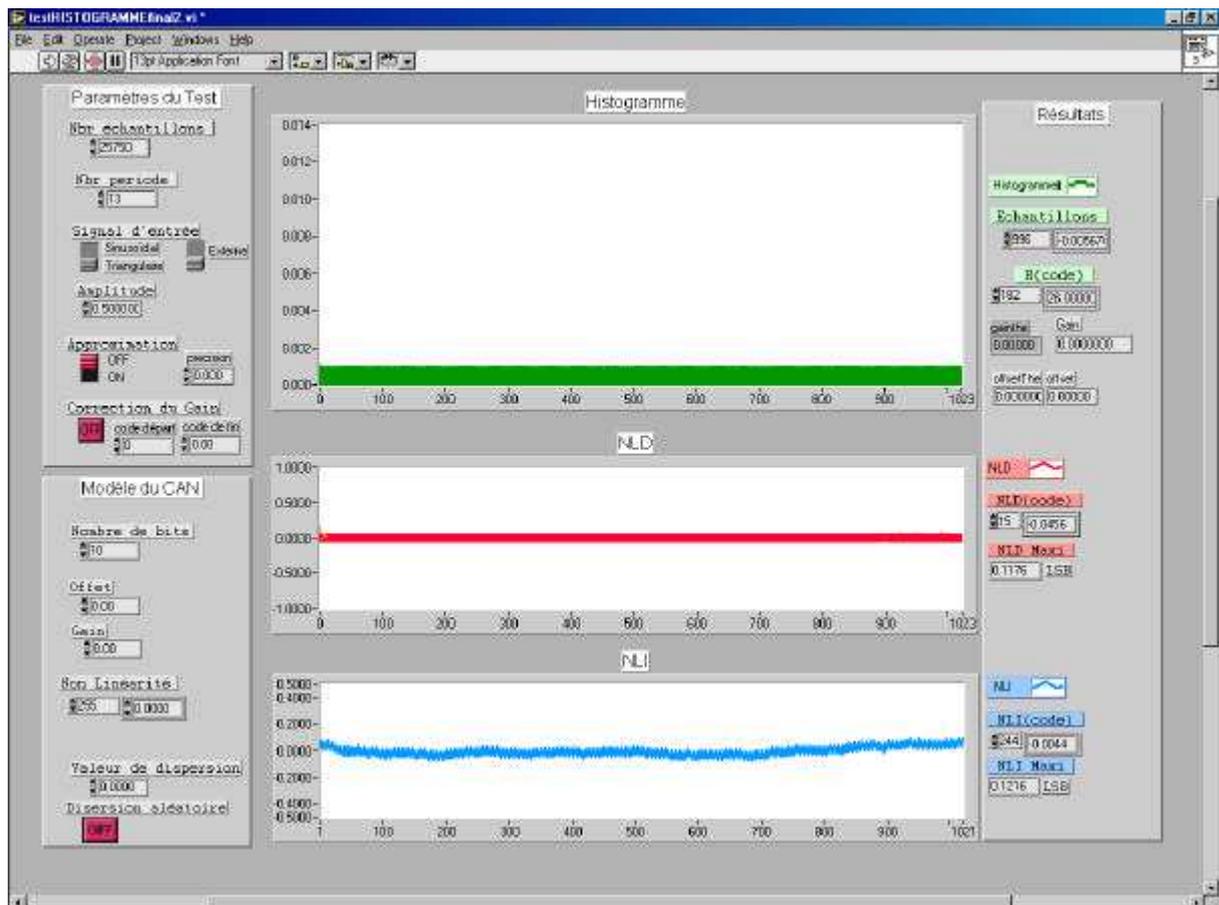


Figure 4.29 : Linéarité du signal d'entrée estimée par la technique de test par histogramme

V. Conclusion

Dans ce chapitre nous avons développé plusieurs architectures de générateurs de signaux linéaires basés sur le principe de la charge d'une capacité à courant constant. Dans le contexte de test intégré de nos travaux, les générateurs proposés doivent avoir d'excellentes caractéristiques fonctionnelles sous des contraintes de surface très fortes.

Les deux caractéristiques fonctionnelles principales d'un signal linéaire sont sa pente et sa linéarité. Nous avons justifié notre choix d'un miroir de courant *grande excursion* qui nous a permis d'obtenir une linéarité maximale sur toute la dynamique du signal généré. En revanche, la pente du signal est difficile à imposer de manière précise. En effet, pour limiter la surface du générateur nous avons utilisé des valeurs de courant et de capacité de charge faibles qui sont très sensibles aux variations technologiques lors de la fabrication du circuit.

Pour compenser les possibles fluctuations de la valeur de la pente du signal, nous avons proposé une structure originale de générateur de rampe auto-calibré. Le premier système de calibration utilise un principe d'asservissement numérique qui permet de corriger la pente de la rampe en modifiant la polarisation du générateur de courant de charge. Cette solution bien que performante implique une surface de silicium relativement importante. Nous avons donc proposé une autre structure de générateur de rampe auto-calibré nécessitant une surface plus réduite en utilisant un asservissement analogique du courant de charge. Le système est basé sur un système de transfert de charges entre deux capacités de valeurs très différentes. Ainsi, l'intégration d'une telle structure impose seulement l'intégration d'une capacité supplémentaire et ne représente donc pas un surcoût important. Le générateur de rampe obtenu peut être utilisé en association avec l'analyseur de réponses de test présenté au chapitre 3 pour le test interne de convertisseurs de résolution inférieure ou égale à 10 bits.

Pour agrandir le champ d'application de la structure BIST aux convertisseurs de résolution supérieure à 10 bits, nous avons proposé une structure de générateur de signaux triangulaires. Ce générateur permet d'obtenir une meilleure linéarité du signal tout en utilisant le même principe de calibration.

Conclusion générale

Les technologies de fabrication des Circuits Intégrés (CI) évoluent à une vitesse telle que la capacité en terme de transistors dans un CI double tous les 18 mois. Cette forte densité d'intégration complique considérablement le test des CI. Ainsi, pour tester correctement ces circuits, il devient indispensable de tenir compte du test dès la phase de conception. Cette approche appelée Conception en Vue du Test (CVT) se présente sous deux formes différentes : soit elle consiste à modifier le circuit pour faciliter son test, soit elle suppose l'intégration de modules d'auto-test (BIST pour Built-In-Self-Test) qui effectuent tout ou partie du test du circuit directement sur la puce. L'intérêt des structures BIST est admis par la communauté scientifique et par le milieu industriel. Pour que ces structures soient utilisables, elles doivent pouvoir être intégrées dans les outils logiciels d'aide à la conception. Actuellement, on peut trouver des modules complets de test intégré incorporés dans les outils de synthèse automatique pour les circuits numériques. Par contre, dans le cas des circuits analogiques ou mixtes, les avancées dans le domaine du test sont beaucoup moins importantes et il n'existe aucune solution BIST globale pour ces circuits. De ce fait, même si la partie analogique ne représente souvent que 10% d'un circuit mixte, le coût de son test peut représenter jusqu'à 90 % du coût du test du circuit complet.

Dans ce contexte, il est primordial d'être capable de proposer des architectures de test intégré pour circuits analogiques ou mixtes. Généralement, dans ces circuits mixtes, le Convertisseur Analogique/Numérique (CAN) joue un rôle prépondérant.

L'objectif des travaux présentés dans ce manuscrit consiste à proposer des architectures BIST dédiées au test fonctionnel des convertisseurs A/N. Ces structures comportent une partie génération de signaux analogiques et une partie analyse des réponses du convertisseur sous test.

Le premier chapitre nous a permis de décrire le contexte de ce travail. Dans un premier temps, nous avons rappelé les définitions des différents paramètres fonctionnels qui permettent la caractérisation un convertisseur analogique/numérique. Pour évaluer ces paramètres fonctionnels, différentes techniques peuvent être utilisées. Nous avons détaillé les deux techniques les plus souvent utilisées dans l'industrie : le test par histogramme et le test par analyse spectrale. Enfin, nous avons présenté un état de l'art des techniques de test intégrées dédiées aux convertisseurs A/N proposées dans la littérature. Ces techniques de test ont souvent recours à des éléments extérieurs, soit pour la génération des stimuli, soit pour le traitement des données de sortie, ou à des éléments déjà présents sur la puce, ce qui suppose des pré-requis sur l'architecture du circuit contenant le convertisseur à tester.

Dans ce cadre de ce travail, nous avons proposé dans le chapitre 2 des architectures haut niveau d'analyseurs de réponses de test basées sur la technique classique du test par histogramme. Cependant, si cette technique de test permet une évaluation précise des paramètres fonctionnels, les ressources matérielles nécessaires à son intégration directe sur silicium sont importantes et impliquent une surface additionnelle prohibitive dans un contexte de test intégré. Par conséquent, nous avons proposé un ensemble de simplifications et d'optimisations basées sur la technique de décomposition temporelle qui nous ont permis de réduire considérablement les ressources nécessaires à l'intégration de la technique de test. En résumé, nous avons remplacé la RAM de 2^{n+1} mots et le DSP nécessaires à l'intégration de la technique initiale du test par histogramme par quelques registres et des opérateurs simples.

Le troisième chapitre a été consacré à la validation des solutions haut niveau de l'analyseur de réponses de test présentées au chapitre 2. Nous avons proposé des architectures niveau porte de l'analyseur de réponses de test pour les deux types de signaux d'entrée : linéaire et sinusoïdal. Dans le cas d'un signal d'entrée linéaire, l'architecture proposée n'est constituée que d'un registre configurable pour extraire les paramètres fonctionnels et stocker l'histogramme de référence et l'histogramme expérimental nécessaires à la technique de test utilisée. Nous avons proposé par la suite une implantation physique de cet analyseur. Nous avons obtenu une surface réduite qui nous a permis de valider l'architecture proposée dans le contexte de test intégré où la surface est un critère important. Dans le cas d'un signal d'entrée sinusoïdal, nous avons présenté un outil qui nous a permis de valider les approximations utilisées pour simplifier et optimiser l'architecture de l'analyseur de réponses de test.

Enfin, le dernier chapitre, nous avons proposé des solutions pour la génération de signaux linéaires. Ces générateurs étant dédiés au test intégré, ils doivent posséder des qualités particulières pour une surface d'intégration minimale. Nous avons étudié la linéarité du signal obtenu et montré que la pente d'un signal généré en interne est extrêmement sensible aux dispersions sur les paramètres des composants inhérentes à la technologie de fabrication du circuit. Pour minimiser l'influence de ces dispersions, nous avons proposé deux techniques de d'auto-calibration. La première technique de calibration est basée sur asservissement numérique. Elle permet de générer une rampe avec une pente précise. Cependant, son intégration nécessite une surface de silicium relativement élevée. La deuxième solution que nous avons proposée, basée sur un asservissement analogique, permet pour un faible coût en surface, de générer en interne une rampe précise. Cependant, dans le cas d'une utilisation en test et pour le cas particulier d'une rampe, les phases d'initialisation nécessaires à la génération d'une trame de test, ont tendance à dégrader la linéarité équivalente du signal. En conséquence, nous avons développé une architecture de générateur de triangle ne nécessitant pas de phase d'initialisation et permettant ainsi une utilisation optimale du signal. Ce générateur utilise la technique de calibration par asservissement que nous avons mis au point, qui nous permet de générer un signal triangulaire insensible aux variations technologiques.

L'objectif de la présente thèse est de proposer des structures de test intégré pour convertisseurs analogique permettant d'évaluer les paramètres fonctionnels du convertisseur directement sur la puce et n'utilisant qu'une surface additionnelle minimale de silicium. Nous avons atteint cet objectif en proposant une structure complète de test intégré allant du générateur de stimuli analogiques à l'analyseur de réponses numériques dans le cas d'un signal linéaire. A partir de cette structure, il s'agit d'étendre le champ d'application au cas de signaux sinusoïdaux en proposant un générateur de signal sinusoïdal intégré. Par la suite, l'idée est de se servir de ces structures BIST de CAN pour faciliter le test des blocs analogiques enfouis dans les circuits mixtes. En effet, les signaux analogiques pourront alors être testés dans le domaine numérique au travers d'un convertisseur A/N utilisant notre structure BIST.

Références bibliographiques

Références bibliographiques

- [Ana97] Analog Devices
"Data Converters", *Design-In Reference Manual Analog Devices*, 1997.
- [Ara97] K. Arabi and B. Kaminska
"Oscillation Built-In-Self-Test (OBIST) Scheme for Functional and Structural Testing of Analog and Mixed-Signal Integrated Circuits", *Proc. International Test Conference*, pp. 786-795, 1997.
- [Ben93] M. Benkais
"Méthodologie de caractérisation des circuits de conversion de données : application aux convertisseurs analogique-numérique à facteur de mérite élevé. Mise en œuvre dans le système CanTest", thèse de l'Université de Bordeaux I, 1993.
- [Bal96] A. Balivada, J. Chin, J. A. Abraham
"Analog Testing with Time Response Parameters", *IEEE Design & Test*, Vol. 13, 1996.
- [Bur83] Burr-Brown
"Dynamic Tests for A/D Converter Performance", *Application Bulletin Burr-Brown Corporation AB-072*, 1983.
- [Cap95] P. D. Capofreddi and B. A. Wooley
"The Use of Linear Models for the Efficient and Accurate Testing of A/D Converters", *Proc. International Test Conference*, pp. 54-60, 1995.
- [Cha96] A. Chatterjee, B. C. Kim, N. Nagi
"DC Built-In-Self-Test for Analog Circuits", *IEEE Design & Test*, Vol.13, No. 2, pp. 26-33, 1996.

- [Che00] S. Cherubal and A. Chatterjee
"Optimal INL/DNL Testing of A/D Converts Using a Linear Model", *Proc. International Mixed Signal Testing Workshop*, pp.319-324, 2000.
- [Csi99] N. Csizmadia and A.J.E.M. Janssen
"Estimating the Integral Non-Linearity of AD-Converters via the Frequency Domain", *Proc. International Test Conference*, pp.757-761, 1999.
- [Dam95] K. Damm and W. Anheier
"HBIST of Nonlinear Analog Building Blocks In Mixed-Signal Circuits", *Proc. International Mixed-Signal Testing Workshop*, pp.257-262,1995.
- [Dew92] M. Dewit, G. Gross and R. Ramachandran
"Built -In Self Test for ADC", *US Patent #5132685*, 1992.
- [Doe84] J. Doernberg, H.S. Lee and D.A. Hodges
"Full-Speed Testing of A/D Converters", *IEEE Journal Solid-State Circuits*, Vol. SC-19, No. 6, pp. 820-827, Dec. 1984.
- [Ehs96] M. Ehsanian, B. Kaminska and K. Arabi
"A New Digital Test Approach for Analog-to-Digital Converter Testing", *Proc. VLSI Test Symposium*, pp. 60-65, 1996.
- [Fre82] J. Frech and L. Au
"Statistical Method of Measuring the Differential NL of an CAN using a Pseudo-Random Triangle Wave Stimulus", *US Patent #4352160*, 1982.
- [Fri97] A. Frish and T. Almy
"HABIST: Histogram-based Analog Built-In-Self-Test", *Proc. International Test Conference*, pp. 760-767, 1997.
- [Fri98] A. Frish and T. Almy
"Histogram Based Testing of Analog Signals", *US Patent #5793642*,1998.
- [Gia92] J.D. Giacomini
"High-Performance ADCs Require Dynamic Testing", *Application Note AD-02 National Semiconductor*, 1992.
-

- [Gin91] B. Ginetti and P. Jespers
"Reliability of Code Density Test for High Resolution ADCs", *Electronics Letters*, Vol. 27, No. 24, 2231-2233, 1991.
- [Guz94] M. Guzinski, J. Lewandowski and V. Veasco
"Method and Apparatus for All Code Testing", *US Patent #5332996*, 1994.
- [Hau95] X. Haurie and G. W Roberts
"Arbitrary-Precision Signal Generation for Bandlimited Mixed-Signal Testing", *Proc. International Test Conference*, pp. 78-86, 1995.
- [Haw96] E. M. Hawrysh and G. W. Roberts
"An Integration of memory-based Analog Signal Generation Into Current DFT Architectures", *Proc. International Test Conference*, pp. 528-537, 1996.
- [Hor90] M.N. Horenstein
"Microelctronic Circuit & Devices", *Prentice-Hall International Editions*, ISBN 0-13-584673-0, 1990.
- [Joh96] D. A. Johns and K. Martin
"Analog Integrated Circuit Design", *John Wiley & Sons publisher*, ISBN 0-471-14448-7, 1996.
- [Jon89] A. Jongepeir
"Fast Test Method for Serial A/D and D/A Converters", *Proc. European Test Conference*, pp. 262–267, 1989.
- [Kha95] S. Khaled, B. Kaminska, B. Courtois, M.Lubaszewski
"Frequency-Based BIST for Analog Circuit Testing", *Proc. VLSI Test Symposium*, pp.54-59, 1995.
- [Ker99] H. G. Kerkhoff
"The Future of Mixed-Signal Testing in the Digital Millenium", *Proc. International Mixed Signal Testing Workshop*, pp.115-116, 1999.

- [Kuy99] T. Kuyel
"Linearity Testing Issues of Analog to Digital Converters", *Proc International Test Conference*, pp. 747-755, 1999.
- [Kuy99] T. Kuyel and H. Bilhan
"Relating Linearity Test Results to Design Flaws of Piplined Analog to Digital Converters", *Proc. International Test Conference*, pp. 772-779, 1999.
- [Lak94] K.R. Laker, W.M.C. Sansen
"Design of Analog Integrated Circuits and Systems", *Mc Graw-Hill International Editions*, ISBN 0-07-113458-1, 1994.
- [Lec99] A. Lechner, A. Richardson and B. Hernes
"Reconfiguration-Based Built-In Self-Test for Analogue Front-End Circuits", *Proc. International Mixed Signal Testing Workshop*, pp.243-247, 1999.
- [Lib96] V. Liberali, F. Maloberti and M. Stramesi
"ADC Characterisation Using the Code Density Test Method With Deterministic Sampling", *Proc. International Mixed Signal Testing Workshop*, pp.113-18, 1996.
- [Lu94] A. K. Lu and G. W. Roberts
"An Analog Multi-Tone Signal generator for Built-In-Self-Test Applications", *Proc. International Test Conference*, pp. 650-659, 1994.
- [Mah87] M. Mahoney
"DSP-based Testing of Analog and Mixed-Signal Integrated Circuits", *IEEE Computer Society Press*, ISBN 0-8186-0785-8, 1987.
- [Max89] S. Max
"Fast Accurate and Complete ADC Testing", *Proc International Test Conference*, pp. 111-117, 1989.
- [Max99] S. Max
"Testing High Speed High Accuracy Analog to Digital Converters Embedded in Systems On a Chip", *Proc. International Test Conference*, pp.763-771, 1999.

- [Nag94] N. Nagi, A. Chatterjee and J. Abraham
"A Signature Analyser for Analog and Mixed-Signal Circuits", *Proc. International Conference on Computer Design*, pp.284-87, 1994.
- [Nag99] N. Nagi
"System-on-chip Mixed-signal Test: Issues, current industry practices and future trends", *Proc. International Mixed Signal Testing Workshop*, pp.201-211, 1999.
- [Nat76] National Semiconductor
"Specifying A/D and D/A Converters", *Application Note 156 National Semiconductor*, 1976.
- [Ohl91] M.J. Ohletz
"Hybrid Built-In Self-Test (HBIST) for Mixed Analog/Digital Integrated Circuits", *Proc. European Test Conference*, pp.307-16, 1991.
- [Per97] E.J. Peralias, A. Rueda and J.L. Huertas
"A DFT Technique for Analog-to-Digital Converters with Digital Correction", *Proc. VLSI Test Symposium*, pp. 302-307, 1997.
- [Per98] E.J. Peralias, A. Rueda, J.A. Prieto and J.L. Huertas
"DFT & On-line Test of High Performance Data Converters", *Proc. International Test Conference*, pp. 534-540, 1998.
- [Per00] E.J. Peralias, A. Rueda and J.L. Huertas
"Two BIST Schemes for Pipelined Analog to Digital Converters", *Proc. International Mixed-Signal Testing Workshop*, pp. 36-39, 2000.
- [Pla94] R. van de Plassche
"Integrated Analog-to-Digital and Digital-to-Analog Converters", *Kluwer Academic Publishers*, ISBN 0-7923-9436-4, 1994.
- [Pro98] B. Provost, S. Palermo and E. Sanchez-Sinencio
"Built-In Self Test for Pipeline ADC's", *Proc. International Workshop on Design of Mixed-Mode Integrated Circuits and Applications*, pp.5-8, 1998.

- [Pro99a] B. Provost and E. Sanchez-Sinencio
"Adaptive Analog Timer for On-Chip Testing", *Proc. International Workshop on Design of Mixed-Mode Integrated Circuits and Applications*, pp 29-32, 1999.
- [Pro99b] B. Provost and E. Sanchez-Sinencio
"Auto-Calibrating Analog Timer for On-Chip Testing", *Proc. International Test Conference*, pp. 686-695, 1999.
- [Ram97] R. Ramadoss and M.L. Bushnell
"Flash A/D Converters – Design For Testability", *Proc. International Mixed-Signal Testing Workshop*, pp. 62–67, 1997.
- [Ren90] S. RENAUD
"Contribution à la Caractérisation des Circuits de Conversion Analogique-Numérique. Conception et Réalisation d'un Système d'Evaluation Dynamique de ces Dispositifs", thèse de l'Université de Bordeaux I, 1990.
- [Rob95] G. W. Roberts and A. K. Lu
"Analog Signal Generation for Built-In-Self Test of Mixed-Signal Integrated Circuits", *Kluwer Academic Publishers*, ISBN 0-7923-9564-6, 1995.
- [Sho90] J. Shott, E Stokes
"Self-Characterizing ADC", *US Patent #4897650*, 1990.
- [Sla93] M. Slamani, B. Kamiska
"T-BIST : A Built-In-Self-Test for Analog Circuits Based on Parameter Translation", *Proc. Asian Test Symposium*, pp. 172-177, 1993.
- [Slo82] E. Sloane
"Method and Apparatus for Calibrating an ADC for DAC Test System", *US Patent #4354177*, 1982.
- [Sou90] T. M. Souders and G. N. Stenbakken
"A Comprehensive Approach For Modeling And Testing Analog and Mixed-Signal Devices", *Proc. International Test Conference*, pp. 169-176, 1990.

- [Ste99] M. A. Stegawski, B. Kaminska and A. Frisch
"Fast and Accurate BIST for D/A and A/D Converters : The Sigma-Delta Approach", *Proc. International Mixed Signal Testing Workshop*, pp.319-324, 1999.
- [Sun97a] S. Sunter and N.Nagi
"A Simplified Polynomial-Fitting Algorithm for DAC and ADC BIST", *Proc. International Test Conference*, pp. 389-395, 1997.
- [Sun97b] S. Sunter and N. Nagi
"Method and apparatus for testing DAC and ADC", *US Patent #5659312*, 1997.
- [Ter93] E. Teraoca, T. Kengaku, I.Yasui, K. Ishikawa and T. Matsuo
"A Built-In Self-Test for ADC and DAC in a Single-Chip Speech CODEC", *Proc. International Test Conference*, pp. 791-796, 1993.
- [Tex95] Texas Instruments
"Understanding Data Converters", *Application Note Texas Instruments SLAA013*, 1995.
- [Ton92] M.F. Toner and G.W. Roberts
"Histogram-based Test for Distortion and Gain Tracking of a Mixed-Signal 8-bit PCM Chip", *Proc. 6th Workshop on New Directions for Testing*, pp. 97-112, 1992.
- [Ton93] M.F. Toner and G.W. Roberts
"A BIST Scheme for an SNR Test of a Sigma-Delta ADC", *Proc. International Test Conference*, pp. 805-814, 1993.
- [Ton95] M.F. Toner and G.W. Roberts
"A BIST Scheme for a SNR, Gain Tracking and Frequency Response Test of a Sigma-Delta ADC", *IEEE Trans. Circuits & Systems II*, Vol. 42, pp. 1-15, 1995.
- [Ton96] M.F. Toner and G.W. Roberts
"A Frequency Response, Harmonic Distortion, and Intermodulation Distortion Test for BIST of a Sigma-Delta ADC", *IEEE Trans. Circuits & Systems II*, Vol. 43, No. 8, pp. 608-613, 1996.

- [Vri97] R. de Vries, T. Zwemstra, E. Bruls and P. Regtien
"Built-In Self-Test Methodology for A/D Converters", *Proc. European Design & Test Conference*, pp. 353–358, 1997.
- [Vri98] R. de Vries and B. Atzma
"Method of Testing an Analog-to-Digital Converter", *US Patent #5854598*, 1998.

Travaux publiés durant la thèse

- [Aza00a] F. Azaïs, S. Bernard, Y. Bertrand and M. Renovell
"Towards an ADC BIST Scheme Using the Histogram Test Technique", *Proc. IEEE European Test Workshop*, pp. 129-134, 2000.
- [Aza00b] F. Azaïs, S. Bernard, Y. Bertrand and M. Renovell
"Sinusoidal Histogram-based BIST for ADC Testing", *Proc. Design of Integrated Circuits and Systems*, pp. 21-24, 2000.
- [Aza01a] F. Azaïs, S. Bernard, Y. Bertrand, M. Renovell
"Optimizing Sinusoidal Histogram Test for Low Cost ADC BIST", Accepté pour publication à *Journal of Electronic Testing : Theory and Application*, Kluwer Academic publishers, 2001.
- [Aza01b] F. Azaïs, S. Bernard, Y. Bertrand, M. Renovell
"A Low-Cost BIST Architecture for Linear Histogram Testing of ADCs", Accepté pour publication à *Journal of Electronic Testing : Theory and Application*, Kluwer Academic publishers, 2001.
- [Aza01c] F. Azaïs, S. Bernard, Y. Bertrand, X. Michel and M. Renovell
"A low-cost adaptive ramp generator for analog BIST applications", *IEEE VLSI Test Symposium*, 2001.
- [Aza01d] F. Azaïs, S. Bernard, Y. Bertrand and M. Renovell
"Implementation of a Linear Histogram BIST for ADCs", *Design Automation and Test in Europe*, 2001.
- [Aza01e] F. Azaïs, S. Bernard, Y. Bertrand, X. Michel and M. Renovell
"On-chip Generation of High-Quality Ramp Stimulus with Minimal Silicon Area", *IEEE Latin American Test Workshop*, 2001.
- [Ber00a] S. Bernard, F. Azaïs, Y. Bertrand and M. Renovell
"Linear Histogram Test for ADCs: a BIST Implementation", *Proc. IEEE International Mixed-Signal Testing Workshop*, pp. 40-45, 2000.

- [Ber00b] S. Bernard, F. Azaïs, Y. Bertrand and M. Renovell
"Minimizing the Hardware Overhead of a Histogram-Based BIST Scheme for Analog-to-Digital Converters", *Proc. 1st IEEE Latin American Test Workshop*, pp. 118-122, 2000.
- [Ren99] M. Renovell, F. Azaïs, S. Bernard and Y. Bertrand
"Procédé et dispositif de test intégré pour un convertisseur analogique-numérique et convertisseur muni d'un tel dispositif", *Brevet CNRS France #9911304*, extension internationale #PCT/FR00/02474, 1999.
- [Ren00] M. Renovell, F. Azaïs, S. Bernard and Y. Bertrand
"Hardware Resource Minimization for an Histogram-Based ADC BIST", *Proc. IEEE VLSI Test Symposium*, pp. 247-252, 2000.

Liste des figures et tableaux

Liste des figures

Figures du Chapitre 1

Figure 1.1 : Conversion Analogique/Numérique	12
Figure 1.2 : Fonction de transfert	14
Figure 1.3 : Variation de l'erreur de quantification	15
Figure 1.4 : Bruit de quantification	16
Figure 1.5 : Erreur d'offset	17
Figure 1.6 : Erreur de gain	17
Figure 1.7 : Non-linéarité Différentielle et Intégrale	18
Figure 1.8 : jitter	20
Figure 1.9 : Structure du test par histogramme	22
Figure 1.10 : Signal linéaire aléatoire	24
Figure 1.11 : Histogramme de référence pour un signal linéaire	25
Figure 1.12 : Signal sinusoïdal aléatoire	26
Figure 1.13 : Densité de probabilité d'une sinusoïde	27
Figure 1.14 : Histogramme de référence pour un signal d'entrée sinusoïdal	28
Figure 1.15 : Histogramme verrouillé	29
Figure 1.16 : Signal sinusoïdal aléatoire	30
Figure 1.17 : Analyse spectrale "single Tone"	33
Figure 1.18 : Analyse spectrale "dual Tone"	34
Figure 1.19 : Construction du gabarit	35
Figure 1.20 : Structure MADBIST	37
Figure 1.21 : Calcul de l'intervalle de temps entre deux commutations du LSB	39
Figure 1.22 : Procédure de test des non-linéarités	39
Figure 1.23 : Structure BIST pour CAN pipeline	41
Figure 1.24 : Technique basée sur l'oscillation du CAN sous test	42

Figures du Chapitre 2

Figure 2.1 : Structure BIST dédiée au test d'un CAN	47
Figure 2.2 : Histogramme idéal pour un signal d'entrée triangulaire	51
Figure 2.3 : Influence de l'erreur d'offset	52
Figure 2.4 : Influence de l'erreur de gain	54
Figure 2.5 : Influence des erreurs de non-linéarités	55
Figure 2.6 : Histogramme de référence pour un signal sinusoïdal	56
Figure 2.7 : Influence de l'erreur de d'offset	57
Figure 2.8 : Influence de l'erreur de gain	60
Figure 2.9 : Optimisation des ressources opératives	62
Figure 2.10 : Décomposition de test en différentes phases	63
Figure 2.11 : Approximation par morceaux de l'histogramme de référence	69
Figure 2.12 : Architecture optimale du module d'exploitation	70

Figures du Chapitre 3

Figure 3.1 : Détecteur de code	76
Figure 3.2 : Module d'exploitation	82
Figure 3.3 : Phases du test	84
Figure 3.4 : Schéma niveau porte du contrôleur	84
Figure 3.5 : Début de la phase de détermination de l'erreur d'offset	85
Figure 3.6 : Fin de la phase de détermination de l'erreur d'offset	85
Figure 3.7 : Implantation physique de l'analyseur de réponses de test	87
Figure 3.8 : Simulation après fabrication	89
Figure 3.9 : Programme d'évaluation	91
Figure 3.10 : Interface utilisateur	92
Figure 3.11 : Diviseur série	97
Figure 3.12 : Additionneur série	99
Figure 3.13 : Automatisation de construction de la structure	105

Figures du Chapitre 4

Figure 4.1 : Influence de l'amplitude du signal d'entrée	109
Figure 4.2 : Influence des non-linéarités du signal d'entrée	110
Figure 4.3 : Génération de rampe basée sur la charge d'une capacité à courant constant	111
Figure 4.4 : Miroir de courant cascode	112
Figure 4.5 : Miroir de courant grande excursion	114
Figure 4.6 : Générateur de rampe	115
Figure 4.7 : Signal de sortie du générateur	116
Figure 4.8 : Non-linéarité de la rampe de sortie	117
Figure 4.9 : Variation de la pente de la rampe	118
Figure 4.10 : Définition de la pente de la rampe	119
Figure 4.11 : Schéma de principe du système de compensation	120
Figure 4.12 : Système de compensation par asservissement numérique	122
Figure 4.13 : Chronogramme du système de calibration par asservissement numérique	123
Figure 4.14 : Convertisseur Numérique/Analogique série	123
Figure 4.15 : Système de compensation par asservissement analogique	125
Figure 4.16 : Incrémenteur/décrémenteur analogique	126
Figure 4.17 : Implantation du générateur de rampe auto-calibré	128
Figure 4.18 : Simulation du générateur de rampe auto-calibré	129
Figure 4.19 : Layout du générateur de rampe auto-calibré	129
Figure 4.20 : Non-linéarité de la rampe générée par la structure auto-calibrée	131
Figure 4.21 : Structure BIST globale	133
Figure 4.22 : Trame du signal utilisé comme stimulus de test	135
Figure 4.23 : Linéarité du signal d'entrée estimée par la technique de test par histogramme	136
Figure 4.24 : Générateur de signaux triangulaires	137
Figure 4.25 : Générateur de rampe auto-calibré	138
Figure 4.26 : Fonctionnement du système de calibration	140
Figure 4.27 : Implantation du générateur de signaux triangulaires auto-calibré	141
Figure 4.28 : Simulation du générateur de triangle auto-calibré	142
Figure 4.29 : Linéarité du signal d'entrée estimée par la technique de test par histogramme	143

Liste des tableaux

Tableaux du Chapitre 1

Tableaux du Chapitre 2

Tableau 2.1 : Nombre d'échantillons pour le test par histogramme _____ 46

Tableaux du Chapitre 3

Tableau 3.1 : Résultats de simulation _____ 82

Tableau 3.2 : Surface de l'analyseur de réponses de test _____ 84

Tableau 3.3 : Comparaison entre notre technique et la technique classique de test par
histogramme _____ 89

Tableau 3.5 : Temps de test _____ 97

Tableaux du Chapitre 4

Tableau 4.1 : Tailles des transistors utilisés _____ 116

Tableau 4.2 : Surface des éléments constituant le générateur de rampe _____ 130

Tableau 4.3 : Surface de la structure BIST _____ 134

Annexe

Annexe 1 : Exemple de spécifications de CAN

AD9034—SPECIFICATIONS

ELECTRICAL CHARACTERISTICS (+V_S = +5 V; -V_S = -5.2 V; Encode = 20.48 MSPS, unless otherwise noted)

Parameter (Conditions)	Temp	Test Level	AD9034AD/AZ			AD9034BD/BZ			AD9034TD/TZ			Units
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
			12			12			12			Bits
DC ACCURACY												
Differential Nonlinearity	+25°C	I	0.65	1.25		0.5	1.0		0.5	1.0		LSB
	Full	VI		1.5			1.25			1.25		LSB
Integral Nonlinearity	+25°C	V	1.0			1.0			1.0			LSB
	Full	V	2.0			2.0			2.0			LSB
No Missing Codes	Full	VI	Guaranteed			Guaranteed			Guaranteed			
Offset Error	+25°C	I	5	15		5	15		5	15		
	Full	VI		25			25			30		
Gain Error	+25°C	I	±0.5	±1.0		±0.5	±1.0		±0.5	±1.0		
	Full	VI		±2.5			±2.5			±2.0		
ANALOG INPUT												
Input Voltage Range	+25°C	V	±1.024			±1.024			±1.024			V
Input Bias Current ¹	+25°C	I		100			100			100		μA
	Full	VI		200			200			200		μA
Input Resistance	+25°C	VI	50	350		50	350		50	350		kΩ
Input Capacitance	+25°C	IV	7	10		7	10		7	10		pF
Analog Bandwidth	+25°C	IV	110	150		110	150		110	150		MHz
SWITCHING PERFORMANCE²												
Conversion Rate	Full	VI	dc		20.48	dc		20.48	dc		20.48	MSPS
Aperture Delay (t _A)	Full	IV	1	3	5	1	3	5	1	4	7	ns
Aperture Uncertainty (Jitter)	Full	IV		5	10		5	10		5	10	ps, rms
Output Delay (t _{OD})	Full	IV	7	10	13	7	10	13	7	10	13	ns
Data Ready Delay (t _{DR})	Full	IV	3.5	7.5	10.5	3.5	7.5	10.5	3.5	7.5	10.5	ns
Output Time Skew	Full	IV		1	2		1	2		1	2	ns
ENCODE INPUT												
Logic "1" Voltage	Full	IV	-1.1			-1.1			-1.1			V
Logic "0" Voltage	Full	IV			-1.5			-1.5			-1.5	V
Logic "1" Current	Full	VI		150	300		150	300		150	300	μA
Logic "0" Current	Full	VI		150	300		150	300		150	300	μA
Input Capacitance	+25°C	V		10			10			10		pF
Pulse Width (High)	+25°C	IV	13			13			13			ns
Pulse Width (Low)	+25°C	IV	13			13			13			ns
DYNAMIC PERFORMANCE												
Transient Response	+25°C	IV		15	30		15	30		15	30	ns
Overvoltage Recovery Time	+25°C	IV		25	40		25	40		25	40	ns
Harmonic Distortion												
Analog Input												
@ 1.2 MHz	+25°C	I	70	80		74	80		74	80		dBc
@ 1.2 MHz	Full	VI	68			70			70			dBc
@ 4.3 MHz	+25°C	V		75			77			77		dBc
@ 9.6 MHz	+25°C	I	67	74		70	76		70	76		dBc
@ 9.6 MHz	Full	VI	64			67			67			dBc
Signal-to-Noise Ratio³												
Analog Input												
@ 1.2 MHz	+25°C	I	65	67		65	67		65	67		dB
@ 1.2 MHz	Full	VI	64			64			64			dB
@ 4.3 MHz	+25°C	V		66			66			66		dB
@ 9.6 MHz	+25°C	I	63	66		63	66		63	66		dB
@ 9.6 MHz	Full	VI	62			62			62			dB
Two-Tone Intermodulation Distortion Rejection ⁴	+25°C	V		68			70			70		dBc

```

use IEEE.std_logic_1164.all;

entity control is
  port(ck,debut,fin,In0,outA5,outA7,outB8,Ini_A,outB3,outB6,clear
  : in std_logic;
       c1,c2,clr_A,clr_B,set_B,ctr,code_suivant : out
  std_logic);
end control;

----- ARCHITECTURE -----

architecture Archcontrol of control is

  type state is
  (EtatIni,s1,s3,s2,s5,s6,s4,s34,s30,s31,s11,s10,s8,s7,s9,s9b,s33,s
  19,s16,s20,s32,s18,s16b,s17,s13,s14,s12,s15,s35,s36,s37,s21);
  signal etatfut, etatpres: state;
begin

controleur: process
(etatpres,debut,fin,outA5,outA7,outB8,In0,outB3,outB6,Ini_A,clear
)

  begin
    etatfut <= etatpres;
    case (etatpres) is

--initialisation
      when EtatIni =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--clear compteur A
      sortie <= "0000101";

      c1 <= '0';
      c2 <= '0';
      clr_A <='0';
      clr_B <='0';
      set_B <='1';
      code_suivant <= '0';
      ctr <='1';

--clear compteur A
      etatfut <= s1;

      when s1 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--
      sortie <= "0011101";
      c1 <= '0';
      c2 <= '0';
      clr_A <='1';
      clr_B <='1';
      set_B <='1';
    end case;
  end process;
end Archcontrol;

```

```

      ctr <='1';
      if(debut ='1') then
        etatfut <= s2;
      else
        etatfut <=s1;
      end if;

-----CALCUL DE L'OFFSET-----
---

      when s2 =>

        clr_A <='1';
        code_suivant <= '0';
        ctr <='1';

--test du bit 0
--pour comptage
-- decomptage

      if(In0='1') then
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr

--      sortie <= "0111001";
        if (Ini_A='0') then
          c1 <= '0';
          c2 <= '1';
        else
          c1 <= '0';
          c2 <= '0';
        end if;
        --clr_A <='1';
        clr_B <='1';
        set_B <='0';
        --code_suivant <= '0';
        --ctr <='1';

      else
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr

--      sortie <= "1010101";
        if (Ini_A='0') then
          c1 <= '1';
          c2 <= '0';
        else
          c1 <= '0';
          c2 <= '0';
        end if;

        --clr_A <='1';
        clr_B <='0';
      end if;
    end when;
  end process;
end Archcontrol;

```

```

        set_B <='1';
        --code_suivant <= '0';
        --ctr <='1';

    end if;

    if(fin='1') then
        etatfut <= s3;
    else
        etatfut <=s2;
    end if;

-- cas ou le bit=0

--mise a 0 du compteur
--B et lecture de la valeur
--du compteur A
        when s3 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0010101";

        --c1 <= '1';
        --c2 <= '0';
        clr_A <='1';
        clr_B <='0';
        set_B <='1';
        code_suivant <= '0';
        ctr <='1';

        if (outA7='1') then
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "1110101";
            c2 <= '1';
            c1 <= '1';

        else
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0010101";
            c2 <= '0';
            c1 <= '0';

        end if;

        etatfut <=s4;

----- CALCUL DU GAIN -----

```

```

        when s4 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0001100";

        c1 <= '0';
        c2 <= '0';
        clr_A <='0';
        clr_B <='1';
        set_B <='1';
        code_suivant <= '0';
        ctr <='0';

        etatfut <=s5;

-- comptage su B si B < 8
-- sans comparaison
--de la sortie du CAN

        when s5 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0011110";

        c1 <= '0';
        c2 <= '0';
        clr_A <='1';
        clr_B <='1';
        set_B <='1';
        code_suivant <= '1';
        ctr <='0';

        etatfut <=s6;

        when s6 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0011100";

        c1 <= '0';
        c2 <= '0';
        clr_A <='1';
        clr_B <='1';
        set_B <='1';
        code_suivant <= '0';
        ctr <='0';

        if(outB3='1') then
            etatfut <=s7;
        else
            etatfut <=s5;
        end if;

```

```

                                end if;
--des que B =8
--comptage avec comparaison

--code suivant
    when s7 =>

-- vecteur de sortie  c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0011110";

    c1 <= '0';
    c2 <= '0';
    clr_A <='1';
    clr_B <='1';
    set_B <='1';
    code_suivant <= '1';
    ctr <='0';

    if(debut='1') then
        etatfut <= s8;
    else
        etatfut <=s7;
    end if;

    when s8 =>

-- vecteur de sortie  c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "1011101";

    --c1 <= '  ';
    --c2 <= '  ';
    clr_A <='1';
    clr_B <='1';
    set_B <='1';
    code_suivant <= '0';
    ctr <='1';
    if(Ini_A='1') then
        c1 <= '0';
        c2 <= '0';
    else
        c1 <= '1';
        c2 <= '0';
    end if;

    if(fin='1') then
        etatfut <=s9;
    else
        etatfut <=s8;
    end if;

```

```

--test si fin=0 => on reste
-- dans le meme etat

                                when s9 =>

-- vecteur de sortie  c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0011100";

    c1 <= '0';
    c2 <= '0';
    clr_A <='1';
    clr_B <='1';
    set_B <='1';
    code_suivant <= '0';
    ctr <='0';

                                etatfut <=s9b;

--sinon si B < 32 => on incremente
--B sinon on va a suite
                                when s9b =>

-- vecteur de sortie  c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0011100";

    c1 <= '0';
    c2 <= '0';
    clr_A <='1';
    clr_B <='1';
    set_B <='1';
    code_suivant <= '0';
    ctr <='0';

    if (outB6='1') then
        etatfut <=s10;
    else
        etatfut <=s7;
    end if;

-- mise a zero de B et lecture
--de A
                                when s10 =>
-- vecteur de sortie  c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0010100";

    clr_A <='1';
    clr_B <='0';
    set B <='1';

```

```

        code_suivant <= '0';
        ctr <='0';

        if (outA5='1') then
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "1110100";
            c2 <= '1';
            c1 <= '1';

        else
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0010100";
            c2 <= '0';
            c1 <= '0';

        end if;

        etatfut <=s11;

----- CALCUL DES NLD -----

        when s11 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0000100";

            c1 <= '0';
            c2 <= '0';
            clr_A <='0';
            clr_B <='0';
            set_B <='1';
            code_suivant <= '0';
            ctr <='0';

            etatfut <=s12;

        when s12 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0011110";

            c1 <= '0';
            c2 <= '0';
            clr_A <='1';
            clr_B <='1';
            set_B <='1';
            code_suivant <= '1';
            ctr <='0';
            if(outB8='1') then

```

```

        etatfut <= s16;
        elsif (debut ='1') then
            etatfut <= s13;
        else
            etatfut <=s12;
        end if;

        when s13 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0111101";

            --c1 <= ' ';
            --c2 <= ' ';
            clr_A <='1';
            clr_B <='1';
            set_B <='1';
            code_suivant <= '0';
            ctr <='1';
            if(Ini_A='1') then
                --sortie <= "0011101";
                c1 <= '0';
                c2 <= '0';
            else
                --sortie <= "0111101";
                c1 <= '1';
                c2 <= '0';
            end if;
            if(fin='1') then
                etatfut <=s14;
            else
                etatfut <=s13;
            end if;

        when s14 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "1011100";

            --c1 <= '1';
            --c2 <= '0';
            clr_A <='1';
            clr_B <='1';
            set_B <='1';
            code_suivant <= '0';
            ctr <='0';

            if (outA5='1') then
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "1111100";
                c2 <= '1';
                c1 <= '1';

```

```

else
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0011100";
      c2 <= '0';
      c1 <= '0';

      end if;
      etatfut <=s15;

      when s15 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0001100";

      c1 <= '0';
      c2 <= '0';
      clr_A <='0';
      clr_B <='1';
      set_B <='1';
      code_suivant <= '0';
      ctr <='0';

      etatfut <=s12;

----- CALCUL DES NLI -----

      when s16 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0000100";

      c1 <= '0';
      c2 <= '0';
      clr_A <='0';
      clr_B <='0';
      set_B <='1';
      code_suivant <= '0';
      ctr <='0';

      etatfut <=s16b;

      when s16b =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0000100";

      c1 <= '0';
      c2 <= '0';
      clr_A <='0';

```

```

      clr_B <='1';
      set_B <='1';
      code_suivant <= '0';
      ctr <='0';

      etatfut <=s17;

      when s17 =>
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0011110";

      c1 <= '0';
      c2 <= '0';
      clr_A <='1';
      clr_B <='1';
      set_B <='1';
      code_suivant <= '1';
      ctr <='0';
      if(outB8='1') then
      etatfut <= s21;
      elsif (debut ='1') then
          etatfut <= s18;
          else
              etatfut <=s17;
          end if;

      when s18 =>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0111101";
--c1 <= ' ';
--c2 <= ' ';
      clr_A <='1';
      clr_B <='1';
      set_B <='1';
      code_suivant <= '0';
      ctr <='1';
      if(Ini_A='1') then
          --sortie <= "0011101";
          c1 <= '0';
          c2 <= '0';
      else
          --sortie <= "0111101";
          c1 <= '1';
          c2 <= '0';
      end if;
      if(fin='1') then
          etatfut <=s19;
      else
          etatfut <=s18;
      end if;

```

```

        when s19 =>
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "1011100";

        --c1 <= '1';
        --c2 <= '0';
        clr_A <='1';
        clr_B <='1';
        set_B <='1';
        code_suivant <= '0';
        ctr <='0';

        if (outA5='1') then
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "1110100";
                c2 <= '1';
                c1 <= '1';

        else
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0010100";
                c2 <= '0';
                c1 <= '0';

        end if;
        etatfut <=s20;

        when s20 =>
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0011100";

        c1 <= '0';
        c2 <= '0';
        clr_A <='1';
        clr_B <='1';
        set_B <='1';
        code_suivant <= '0';
        ctr <='0';

        etatfut <=s17;

        when s21 =>
-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "0000100";

        c1 <= '0';
        c2 <= '1';

```

```

        clr_A <='0';
        clr_B <='0';
        set_B <='1';
        code_suivant <= '0';
        ctr <='0';

        etatfut <=EtatIni;

        when others=>

-- vecteur de sortie c1,c2,clr_A,clr_B,set_B,code_suivant,ctr
--sortie <= "1000101";

        c1 <= '1';
        c2 <= '0';
        clr_A <='0';
        clr_B <='0';
        set_B <='1';
        code_suivant <= '0';
        ctr <='1';

        etatfut <= EtatIni;

        end case;

end process controleur;

synchro: process (clear, ck, etatfut)
begin
    if ( clear='0')
        then etatpres <= etatIni ;
        elsif (ck'event and ck='1') then
            etatpres <= etatfut ;

        end if;
end process synchro;

end Archcontrol;

configuration conf_control of control is
for Archcontrol
end for;
end conf_control;

```

Annexe 3 : Dimensionnement du générateur de rampe

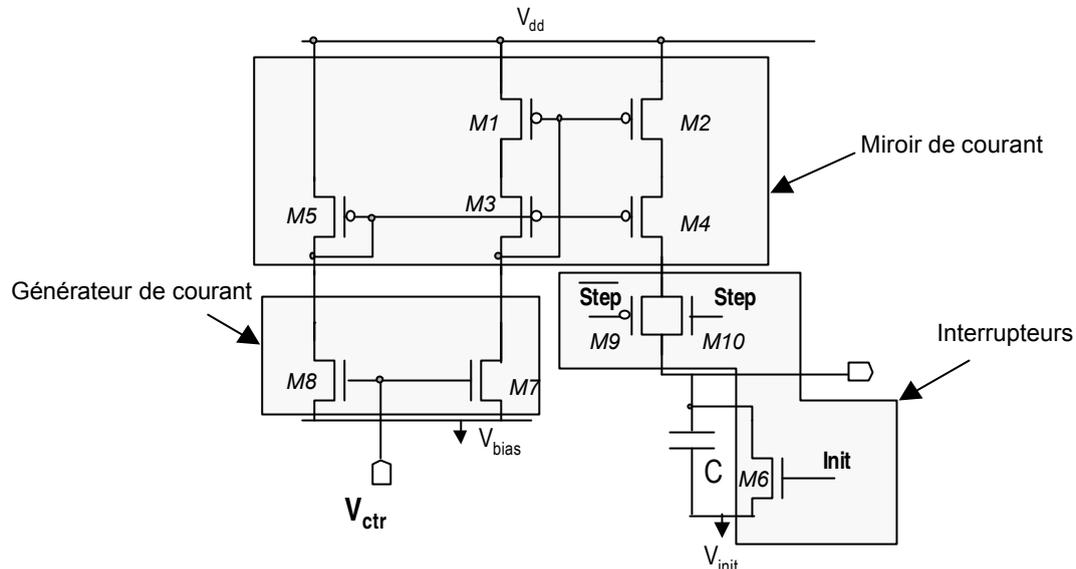


Figure A.1 : Générateur de rampe

Le générateur de rampe (figure A.1) est composé de trois modules distincts :

- le miroir de courant *grande excursion*,
- le module comportant les interrupteurs de commande,
- le générateur de courant.

Dans cette annexe, nous décrivons brièvement les orientations que nous avons prises afin de dimensionner les transistors de ces trois modules.

➤ Le générateur de courant

Les transistors du générateur de courant doivent respecter deux conditions de fonctionnement. D'une part, pour une variation donnée du signal de contrôle V_{ctr} , la variation du courant généré doit être minimale pour permettre une auto-calibration optimale du générateur. D'autre part le courant généré doit être le plus possible indépendant des variations de la tension drain-source du transistor pour assurer une linéarité maximale de la rampe générée.

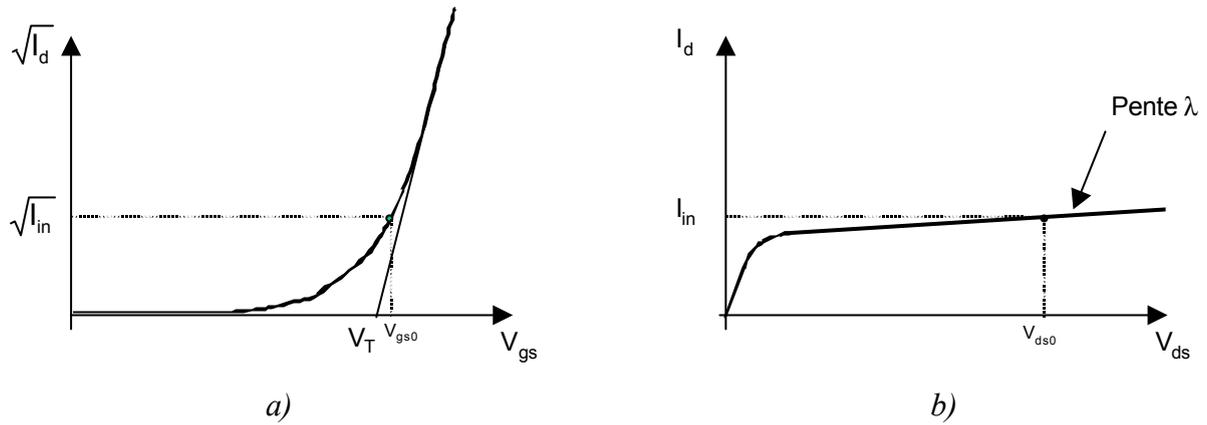


Figure A.2: Caractéristiques d'un transistor NMOS

La première condition impose d'augmenter le rapport $\frac{W}{L}$ du transistor pour diminuer le rapport $\frac{\Delta I_d}{\Delta V_{gs}}$ tout en imposant une valeur de polarisation supérieure à V_T pour éviter les effets non-linéaires d'un fonctionnement en *subthreshold* du transistor (figure A2.a). Dans ces conditions, l'équation suivante permet d'estimer la valeur du courant I_d en fonction de la tension effective ($V_{eff} = V_{gs} - V_T$) :

$$I_d = K_n \frac{W}{L} V_{eff}^2 [1 + \lambda (V_{ds} - V_{eff})] \quad (\text{A1})$$

K_n est une constante qui dépend de la technologie utilisée.

La deuxième condition impose le choix d'une valeur minimale pour la constante λ définie dans l'équation (A.1). Cette constante étant inversement proportionnelle à la largeur du transistor, il est nécessaire d'augmenter cette largeur pour diminuer la valeur de λ (figure A2.b).

Finalement, en prenant en compte ces deux critères et en utilisant le module d'optimisation d'Analog Artiste (SPECTRE), nous obtenons les tailles suivantes pour les transistors :

- M7 : $L = 42,4 \mu\text{m}$ $W = 2 \mu\text{m}$,
- M8 : $L = 11,8 \mu\text{m}$ $W = 2 \mu\text{m}$.

➤ **Le module comportant les interrupteurs**

Les transistors utilisés sont choisis de taille minimale pour minimiser les capacités parasites et les effets d'injection de charges. Ainsi, pour les transistors M6, M9 et M10, nous imposons $W = 0,8 \mu\text{m}$ et $L = 0,6 \mu\text{m}$.

➤ **Le miroir de courant**

La dynamique maximale de la rampe générée dépend directement de la tension de polarisation des transistors du miroir de courant. Ainsi, nous avons prédéterminé les tailles des transistors pour avoir une tension grille-source $|V_{gs0}|$ minimale tout en restant supérieure à $|V_T|$ pour le courant considéré ($i = 300 \text{ nA}$). Après avoir optimisé les valeurs obtenues grâce à un ensemble de simulations, nous avons imposé $W = 5 \mu\text{m}$ et $L = 5 \mu\text{m}$ pour les transistors M1 à M4 impliquant $W = 1,25 \mu\text{m}$ et $L = 5 \mu\text{m}$ pour le transistor M5.

Annexe 4 : Dimensionnement du générateur de triangles

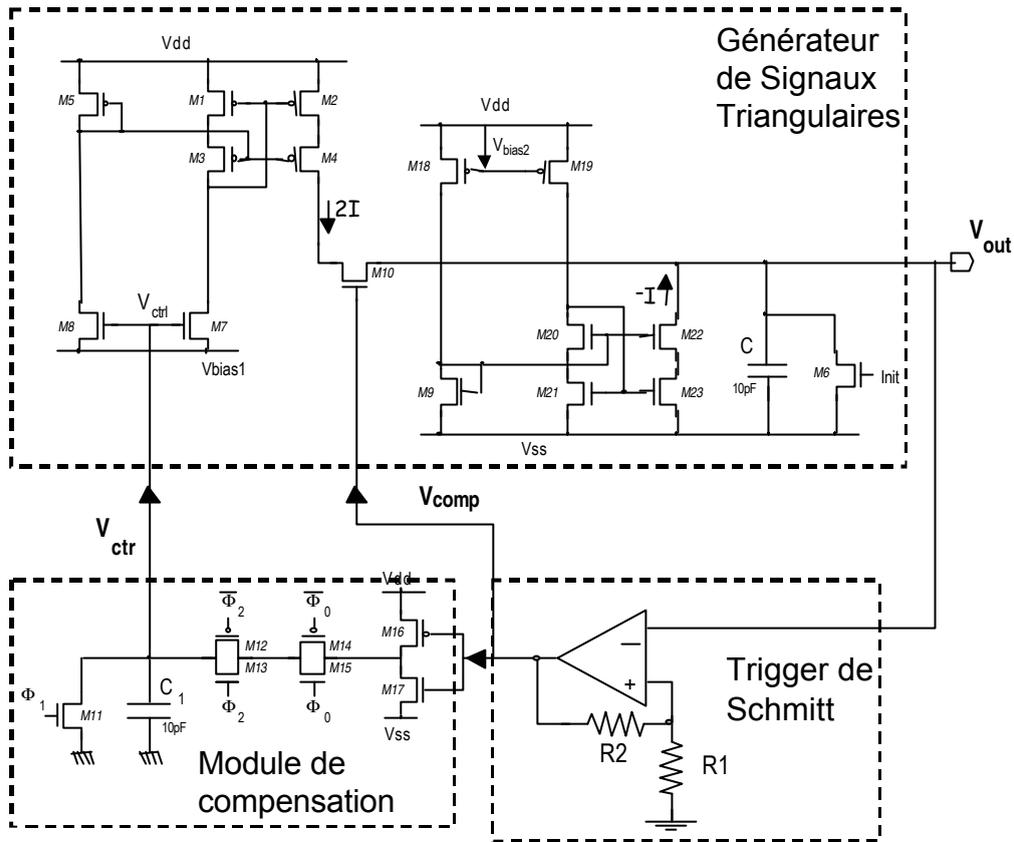


Figure A.3 : Générateur de signaux triangulaires

Partant des mêmes contraintes que pour le générateur de rampe nous avons prédéterminé la taille des transistors du générateur et optimisé les valeurs obtenues à l'aide du module d'optimisation d'Analog Artist. Ainsi, nous obtenons les tailles de transistors résumées dans le tableau suivant pour $V_{bias2} = 1,33$ V :

	M1⇒M4	M5	M6	M7	M8	M9	M10⇒M15	M16	M17	M18	M19	M20⇒M23
W	5	1,25	0,8	2	2	1,25	0,8	3,5	2	2	2	5
L	5	5	0,6	42,4	11,8	5	0,6	0,6	0,6	9	31	5

Ces résultats permettent de présenter une première solution de dimensionnement. A l'avenir, il sera intéressant d'optimiser cette solution en vue d'une implantation physique du circuit. En effet, même si les longueurs importantes de certains transistors ne causent pas de problème au niveau de la simulation, dans le cas d'une implantation physique elles risquent de perturber le bon fonctionnement du montage.

Résumé

Les circuits intégrés mixtes développés pour les nouvelles applications multimédias et télécommunications sont constitués de blocs analogiques et de blocs numériques. Le coût du test de ces circuits mixtes est un facteur critique pour leur prix de revient. En particulier, en production industrielle, les Convertisseurs Analogique/Numérique (CAN) sont testés en mode fonctionnel (histogramme, FFT) en utilisant des ressources de test externes extrêmement coûteuses. Dans ce contexte, une solution attractive pour réduire le coût du test consiste à intégrer directement sur la puce tout ou une partie des ressources nécessaires au test.

L'objectif des travaux présentés dans cette thèse est donc la conception et le développement de structures d'auto-test intégré (BIST) permettant le test par histogramme des CAN. L'implantation directe sur silicium de cette technique de test ne serait pas possible car elle nécessiterait un surcoût de silicium important. Pour rendre cette intégration viable nous avons donc été amenés à envisager des solutions originales basées sur la décomposition et l'analyse par histogramme. Cette approche, associée à la mise en place d'un certain nombre de simplifications des calculs d'extraction nous a permis de réduire considérablement les ressources matérielles (mémoires, module de calcul) à intégrer. Enfin, pour compléter cette structure BIST, nous avons conçu une architecture originale de générateur de rampe et de générateur de signaux triangulaires. Ces générateurs utilisent un système d'auto-calibration qui leur permet de générer un signal précis et insensible aux variations des paramètres technologiques tout en impliquant une surface de silicium minimale.

Mots clés :

Circuits Intégrés, Circuits Mixtes, Convertisseurs Analogique/Numérique, Générateur de Rampe, Générateur de Signaux Triangulaires, Test, Test Intégré.

Built-In-Self Test of Analog-to-Digital Converters

Abstract

The mixed-signal circuits developed for the new multimedia applications include analog blocks and digital blocks. In an industrial context, the Analog-to-Digital Converters (ADC) are tested with a functional approach (histogram, FFT) by using expensive external test equipment. An attractive solution to reduce the cost of the test consists in moving some or all the tester functions onto the chip itself. The objective of the work presented in this thesis is the design and the development of architectures for histogram-based ADC Built-In-Self-Test (BIST). The straightforward implementation of the histogram test technique requires a significant silicon area overhead. In order to reduce this area, we propose original solutions based on the time decomposition of the histogram-based test. With this approach and simplifications of calculations, we can reduce the required hardware resources (memories, calculation). Finally, to make this BIST structure complete, we design an original architecture of ramp and triangular-wave generator. These generators use a corrective scheme to generate an accurate signal which is insensitive to the process variations. Furthermore they exhibit a very low area overhead.

Key words :

Integrated Circuits, Mixed-Signal Circuits, Digital-to-Analog Converters, Ramp Generator, Triangular Wave, Test, Built-In-Self-Test.