



HAL
open science

Représentation et traitement des signaux analogiques dans le domaine temporel, pour répondre aux défis des technologies CMOS très avancées

David Buffeteau

► **To cite this version:**

David Buffeteau. Représentation et traitement des signaux analogiques dans le domaine temporel, pour répondre aux défis des technologies CMOS très avancées. Micro et nanotechnologies/Microélectronique. Université Grenoble Alpes, 2018. Français. NNT : 2018GREAT105 . tel-02138956

HAL Id: tel-02138956

<https://theses.hal.science/tel-02138956>

Submitted on 24 May 2019

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.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **NANO ELECTRONIQUE ET NANO TECHNOLOGIES**

Arrêté ministériel : 25 Mai 2016

Présentée par

David BUFFETEAU

Thèse dirigée par **Dominique MORCHE**

préparée au sein du **Laboratoire CEA/LETI/DACLE/SCCI/LAIR**
dans l'**École Doctorale Electronique, Electrotechnique, Automatique,**
Traitement du Signal (EEATS)

Représentation et traitement des signaux analogiques dans le domaine temporel, pour répondre aux défis des technologies CMOS très avancées.

Thèse soutenue publiquement le **24 Octobre 2018**,
devant le jury composé de :

Emil NOVAKOV

Professeur, Université Grenoble Alpes, Président

Pascal NOUET

Professeur, Université de Montpellier, Rapporteur

Nicolas DEHAESE

Maître de conférence, Université d'Aix-Marseille, Rapporteur

José-Luis GONZÁLEZ-JIMÉNEZ

Ingénieur de recherche, CEA LETI (encadrant), Examinateur

Dominique MORCHE

Directeur de Recherche CEA, CEA LETI, HDR (directeur de thèse), Invité



“Suave, mari magno turbantibus aequora ventis e terra magnum alterius spectare laborem ; non quia vexari quemquamst iucunda voluptas, sed quibus ipse malis careas quia cernere suavest.”, Lucrèce, *De natura rerum*

Je savais que j'aurais dû être plus attentif pendant les cours de latin ... Si le moi du futur invente la machine à voyager dans le temps, qu'il revienne à cet instant précis pour m'interrompre et m'empêcher d

Remerciements

Qui eût cru que la page la plus difficile à écrire de ce manuscrit serait la page la moins technique ? En arrivant au bout de la rédaction de ce manuscrit, il est bon de jeter un oeil derrière soi pour voir le chemin parcouru.

Et parce qu'on ne peut pas faire une thèse tout seul, il est temps de remercier ceux qui ont contribué, de près ou de loin, à la finalisation de ce travail.

Je commencerai bien évidemment par ceux qui ont directement contribué à ce manuscrit. Dominique, bien sûr pour ses relectures, ses conseils et certaines des tournures de phrases les plus élégantes de ce manuscrit. Marc Renaudin, pour ses lignes directrices avisées et ses remarques pertinentes ainsi que pour les déjeuners partagés qui permettent de prendre du recul sur la thèse. Ma moitié, Amandine, qui a épargné aux lecteurs un grand nombre de fautes de français et de tournures de phrases alambiquées (il en reste peut-être, hélas). Enfin, je tenais à remercier mon frère, Guyhème, et son ami Jordan Rous (qui est devenu le mien également), pour avoir été les garants de l'aspect esthétique de ce manuscrit. J'espère que leurs illustrations de début de chapitre, à la fois belles et pleines d'humour, égayeront le cheminement du lecteur.

Mais le travail de thèse ne se résume pas à ce manuscrit et il serait terriblement injuste de ne pas remercier ceux dont la contribution n'est pas directement visible dans ce manuscrit.

Parmi ces personnes j'aimerais commencer par Jumana Boussey, ma responsable de filière auprès de l'école doctorale. Elle a été à la fois un soutien moral et logistique dans cette thèse et je la remercie pour son implication.

Et puis, il y a ceux dont la contribution n'est pas quantifiable tant elle est étalée sur le long terme. Je parle bien sûr des collègues qui au fil du temps se sont transformés en amis. Chacun ayant une astuce pour augmenter la productivité, de

la nourriture à la boisson en passant par des bonbons. Et parce qu'il est impossible d'être au top tous les jours et qu'on a parfois besoin d'un coup de pouce qui peut aller du calcul de l'équation d'une droite (il y a des jours où l'on est vraiment pas au top ...) à l'émergence d'une piste de recherche suite à une discussion où l'idée est jetée en l'air, je les remercie pour leur aide.

Enfin, s'il on en croit les travaux de Edward N. Lorenz, tout ce travail est le résultat d'une multitude d'évènements. Un professeur de mathématiques qui donne envie d'aller plus loin, un professeur de physique qui donne envie d'expliquer son environnement, des parents, toujours dans l'effort, pour contenter une curiosité qui semble sans borne. Et bien sûr le soutien, à la fois affectif et financier (dont l'un a permis de financer l'ordinateur sur lequel je suis en train de taper ces lignes, l'autre m'ayant lâchement abandonné en pleine rédaction) de mes parents, mes grands-parents, mon frère et ma chérie.

À tous, je vous dis merci.

Table des matières

Table des matières	v
Liste des Acronymes et Glossaire	vii
Introduction et contexte	ix
Problématique	xiii
La conversion dans le domaine de l'amplitude : problème du Power wall	xiv
Évolution des besoins applicatifs : ADC à performance variable	xvi
Contributions	xviii
Organisation du document	xix
I Etat de l'art	1
I.1 Représentation dans le domaine temporel	2
I.1.A Définitions et présentation	2
I.1.B Synchrone et asynchrone	8
I.1.C Conversion dans le domaine temporel : avantages et limitations	10
I.1.D Bilan et positionnement des différentes méthodes vis-à-vis de l'état de l'art	19
I.2 Traitement dans le domaine temporel	22
I.2.A Calcul temporel sur les créneaux de G. Roberts	22
I.2.B Traitement des signaux pulsés	25
I.2.C Bilan	27
I.3 Systèmes à performance variable	28
I.3.A État de l'art	28
I.3.B Étude de la dynamique de reconfigurabilité des ADC dans le domaine temporel	29
I.3.C Bilan	35
I.4 Bilan de l'état de l'art	36

II	Méthode de traitement dans le domaine temporel : calcul par le biais de créneaux	37
II.1	Présentation de la méthode	38
II.2	Intérêt de la méthode	42
II.3	Caractérisation théorique de la méthode	43
II.4	Limitations	46
II.5	Conclusion	48
III	Le convertisseur à deux modes de fonctionnement : l'ADC hybride	51
III.1	Caractère asynchrone du VCO-based ADC	52
III.2	Deux méthodes de traitement de la sortie d'un VCO	54
III.2.A	Comme un train d'impulsions asynchrones	55
III.2.B	De manière synchrone avec compteur	57
III.3	L'exploitation des modes : l'ADC hybride à deux modes de fonctionnement	59
III.3.A	Bloc de décision	60
III.3.B	Bilan	62
IV	Un mode performant amélioré : l'ADC avec extraction du résidu	67
IV.1	Présentation de l'idée de résidu	68
IV.2	Modélisation de la solution	72
IV.2.A	Modéliser le VCO	72
IV.2.B	Ajout des non idéalités	78
IV.2.C	Plage de validité du modèle tout numérique	81
IV.3	Comment extraire le résidu en pratique	83
IV.4	Faisabilité de l'extraction de créneaux	89
IV.4.A	Idée 1 : La méthode de calcul avec des créneaux du Chapitre II	89
IV.4.B	Idée 2 : Quantifier les créneaux avec une horloge	90
IV.4.C	Idée 3 : La méthode par charge de capacités	91
IV.5	Résultats et limitations	93
V	Conclusion et Perspectives	97
V.1	Calcul dans le domaine temporel	98
V.2	L'ADC hybride répondant aux systèmes à deux modes de fonctionnement	98
V.3	Le VCO-based ADC avec résidu pour améliorer la résolution	99
V.4	Perspectives	99
	Bibliographie	131
	Table des figures	135

Liste des Acronymes et Glossaire

Acronymes

CAN ou ADC	Convertisseur analogique numérique
CNA ou DAC	Convertisseur numérique analogique
DSP	Unité de traitement numérique
IoT	Internet of things ou Internet des objets
ITRS	International Technology Roadmap for Semiconductors
PWM	Modulation sur la largeur de créneaux
SNDR	Rapport signal à bruit plus distorsion
SNR	Rapport signal à bruit
VCO	Voltage controlled oscillator ou Oscillateur contrôlé en tension

Glossaire

Conversion	Désigne dans ce manuscrit la conversion analogique numérique ; c'est-à-dire l'action de transformer une information analogique en une information numérique compréhensible par une unité de traitement
Cortana	L'outil de reconnaissance vocale Microsoft qui répond à la phrase "Hey, Cortana"
Event-driven	Terme anglais désignant la capacité d'un système à évoluer en fonction de l'activité de son entrée
Google Now	L'outil de reconnaissance vocale Google qui répond à la phrase "OK, Google"

Grafcet	Langage graphique représentant le fonctionnement d'un automatisme
ITRS	Feuilles de route correspondant à l'évolution des technologies rédigées par des représentants d'organisations de l'industrie des semi-conducteurs
Level-crossing	Terme anglais désignant la conversion par franchissement de seuils
Loi de Moore	Conjecture émise par Gordon E. Moore disant que la densité de transistors des microprocesseurs double tous les deux ans
Power penalty factor	Surcoût d'énergie à payer pour conserver les performances d'une conversion malgré la réduction de la dynamique
Power wall	Limite théorique de la réduction de la dynamique des tensions d'alimentation
Système asynchrone	Système dont la sortie est produite à des instants ne pouvant être prédits a priori
Système synchrone	Système dont la sortie est produite à des instants prédictibles, souvent des fronts d'horloge
Vdd ou VDD	Tension d'alimentation maximale
$V_{DS_{sat}}$	Marge de tension drain-source nécessaire pour garder le transistor dans la zone linéaire
Vss ou VSS	Tension d'alimentation minimale
Wake-up radio	Système radio qui reste actif et qui réveille (wake-up) le récepteur principal afin de recevoir des données

Introduction et contexte



« *L'électronique au quotidien.* » Crédits : Guyhème Buffeteau



LES systèmes électroniques s'ouvrent de plus en plus au grand public et ne cessent de se complexifier avec des ordinateurs de plus en plus performants ou des téléphones de plus en plus polyvalents. Ces produits qui sont des ambassadeurs de ce que l'on appelle communément *l'ère du numérique* tirent leur puissance de leur capacité de calcul croissante et de leur interaction avec le monde qui les entoure. Ainsi ces petits bijoux technologiques sont équipés de gyroscopes, d'accéléromètres et d'une multitude d'autres capteurs. Les informations recueillies par ces capteurs ne peuvent cependant pas directement être traitées par les processeurs toujours plus puissants de ces appareils car elles appartiennent au monde dit *analogique*.

Contrairement au domaine du *numérique* composé de valeurs discrètes que les processeurs arrivent toujours plus efficacement à traiter, le monde physique qui nous entoure est composé lui, tout du moins à notre échelle, de grandeurs à valeurs continues. Il n'existe ainsi pas un jeu de valeurs pour l'intensité lumineuse de l'environnement qui nous entoure pas plus que de valeurs discrètes d'intensité sonore ou de tension.

Pour pouvoir faire interagir ces deux mondes que sont le monde analogique et le monde numérique, nous avons recours à des convertisseurs. Lorsqu'ils servent à faire la conversion du domaine analogique vers le domaine numérique ils sont appelés *Convertisseurs Analogique Numérique* (abrégié en CAN ou ADC) et lorsqu'ils sont chargés de faire la conversion inverse ils sont appelés *Convertisseurs Numérique Analogique* (abrégié en CNA ou DAC). Ces convertisseurs jouent donc un rôle primordial dans les systèmes électroniques modernes puisqu'ils leur permettent de comprendre les informations sur leur environnement pour pouvoir interagir avec ce dernier.

Les évolutions de ces dernières années ont amené à développer davantage les appareils portables. En effet, smartphones, montres connectées ou plus généralement objets connectés ont envahi les maisons des particuliers. La contrainte de ces périphériques mobiles (outre les contraintes de performance de n'importe quel appareil) réside dans la capacité à accorder une autonomie suffisante pour son utilisation. Personne ne voudrait d'un smartphone qu'il faut recharger toutes les heures.

Dans un souci d'optimisation de la consommation pour ces appareils autonomes (au cœur de ce que l'on appelle *l'Internet des objets* ou *IoT* dans la langue de Shakespeare), un enthousiasme grandissant s'est développé autour des systèmes qui attendent d'être sollicités avant de fonctionner et qui restent en veille autrement. Le cas des "Wake-up radios" ou des systèmes de reconnaissance vocale comme Cortana ou Google Now en sont de bons exemples.

Tous ces circuits électroniques destinés au grand public ont pour point commun (en tous cas pour l'écrasante majorité) d'être réalisés sur du silicium, matériau

servant de substrat à ces circuits. Dans cette atmosphère de compétitivité accrue et dans ce contexte de réduction des coûts tout en augmentant les performances des systèmes électroniques, les chercheurs, motivés par les industriels, se sont lancés depuis plusieurs années dans une course à la miniaturisation des systèmes électroniques (voir l'évolution décrite en figure 1) car qui dit miniaturisation dit réduction des coûts de production. Cette course à la miniaturisation des transistors ; brique de base des circuits électroniques ; a donné naissance à ce que l'on appelle la *Loi de Moore*.

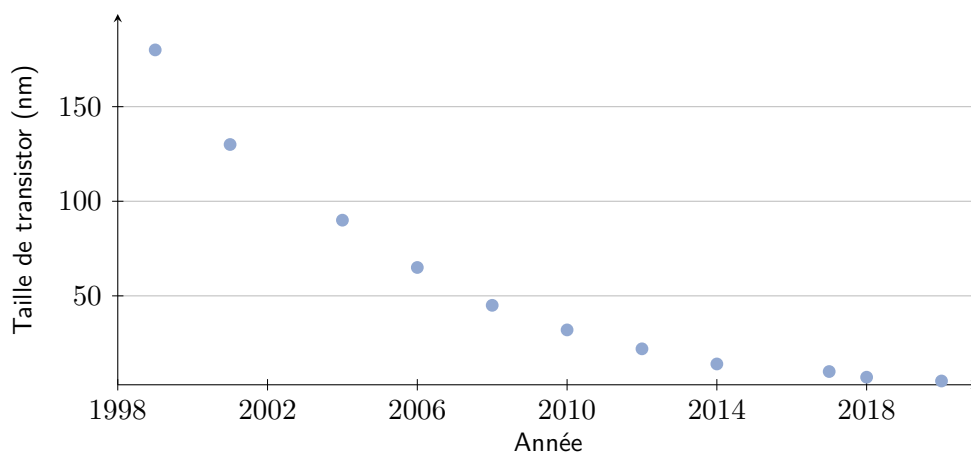


FIGURE 1 – Évolution des nœuds technologiques au cours du temps selon l'ITRS.

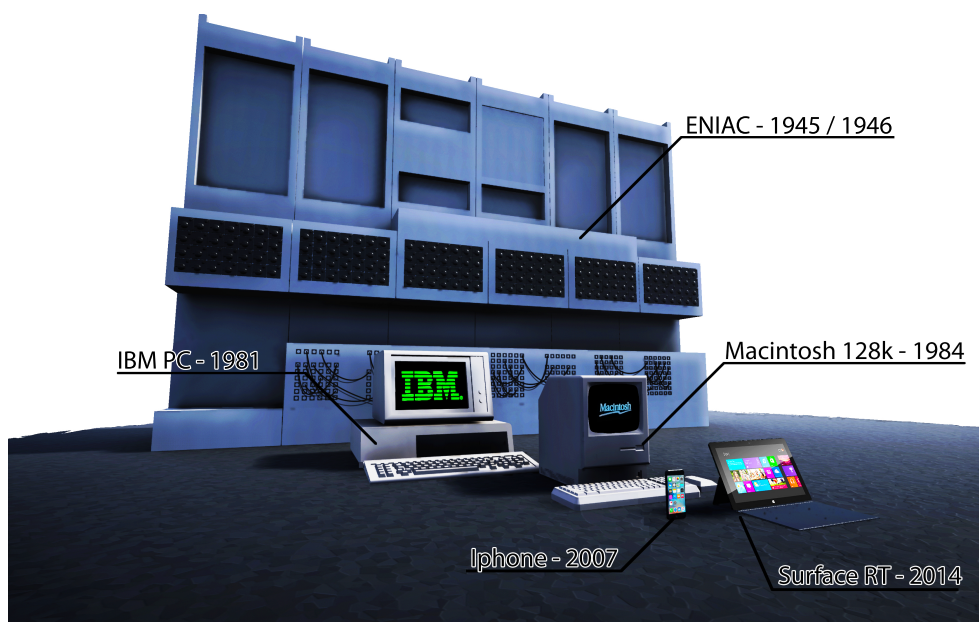
La réduction de la taille des transistors offrant davantage de performances aux circuits numériques, de nouvelles possibilités ont été offertes, qui elles-mêmes ont permis de répondre à de nouveaux besoins ou d'en créer d'autres qui nécessitent davantage de performances et ainsi de suite. La tendance décrite ainsi par la loi de Moore a pendant de nombreuses années collé à la réalité et en un sens a guidé le développement industriel des circuits électroniques et notamment des microprocesseurs. Cependant, face à de nouvelles contraintes physiques (taille des transistors ne pouvant descendre indéfiniment, apparition de nouveaux phénomènes quantiques dans les transistors) et face à des contraintes d'ordre économique (chaînes de production dont le prix augmente de manière exponentielle à chaque nouveau nœud technologique) cette loi de Moore ne pourra pas être suivie indéfiniment. Cela sort cependant du spectre de ce manuscrit et ne sera pas abordé davantage ici.

Mais si la réduction des tailles des transistors a des effets extrêmement bénéfiques vis-à-vis de l'électronique numérique (c'est-à-dire celle qui compose les processeurs utilisés par le grand public par exemple), il n'en est pas de même pour les circuits analogiques (circuits qui doivent fonctionner avec des grandeurs continues comme le monde qui nous entoure et dont font partie les convertisseurs).

En effet, la miniaturisation a pour effet d'augmenter la densité des transistors (nombre de transistors par unité de surface) mais d'autres paramètres sont également améliorés. Dans un premier temps, la vitesse de commutation des transistors (passage d'un transistors de la valeur "0" à la valeur "1" et vice-versa) augmente, ce qui conduit à une vitesse de fonctionnement accrue pour un système numérique. Dans un second temps, les tensions d'alimentation nécessaires au fonctionnement des circuits numériques sont réduites. Bien que bénéfique pour les circuits numériques, cette dernière ne l'est absolument pas pour les circuits faisant intervenir des signaux analogiques comme les convertisseurs introduits plus tôt. Cela impose de trouver un compromis entre profiter pleinement des avantages apportés à la partie numérique par l'évolution de la technologie et les problèmes que cette dernière apporte aux circuits analogiques si l'on veut pouvoir regrouper ces circuits de différentes natures au sein d'une même puce (toujours pour des intérêts économiques).


C'est dans ce contexte de réduction des tailles de transistors que vient s'inscrire ce travail de thèse. La partie suivante va servir à mettre l'emphase sur les conséquences de la réduction des tailles des transistors pour les circuits analogiques d'une part et de l'évolution des besoins d'autre part.

Problématique



« Réduction des tailles de transistors. » Crédits : Jordan Rous

La conversion dans le domaine de l'amplitude : problème du Power wall

OUS nous intéressons particulièrement à la partie conversion analogique numérique qui permet de donner une représentation compréhensible par les systèmes de traitement numérique d'un signal analogique. Traditionnellement, un signal analogique est représenté par un signal continu d'amplitude variable et la conversion analogique numérique consiste à évaluer cette amplitude pour la coder en un mot numérique. Seulement, la réduction des tailles entraînant une réduction des tensions d'alimentation (se reporter à la figure 2) on se retrouve confronté à un problème en ce qui concerne la représentation des signaux analogiques.

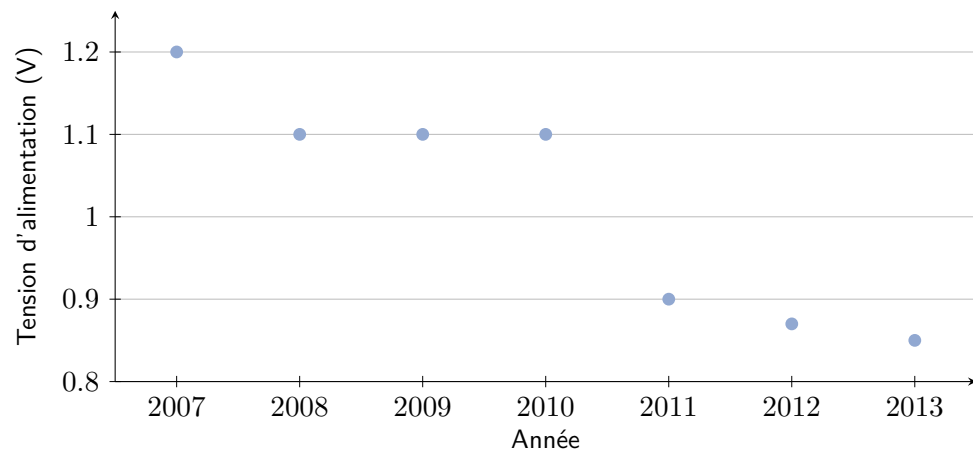


FIGURE 2 – Évolution de la tension d'alimentation des transistors au cours du temps selon l'ITRS.

En effet, la dynamique en tension (c'est-à-dire la plage de valeurs disponibles en tension) du signal pour garder le transistor dans sa zone de saturation comme il est fait dans les circuits analogiques dépend directement de la tension d'alimentation (que l'on note couramment V_{dd}) des transistors dans le système comme illustré dans la figure 3. Ici, un signal analogique classique (sous la forme d'une sinusoïde) est représenté dans la dynamique qui lui est allouée. Afin de rester dans la zone de fonctionnement linéaire, le signal ne peut pas occuper toute la dynamique jusqu'à V_{DD} mais doit rester entre $V_{SS} + V_{DS_{sat}}$ et $V_{DD} - V_{DS_{sat}}$ (V_{SS} étant la valeur minimale de l'alimentation). Ainsi, la réduction des tensions d'alimentation entraîne la réduction de cette dynamique disponible en amplitude. La figure 4 permet de visualiser l'impact de la réduction de la tension d'alimentation sur la dynamique disponible pour représenter le signal analogique.

Une réduction d'un facteur 2 de la dynamique V_{SS} à V_{DD} ne se traduit pas par une réduction d'un facteur 2 de la dynamique réelle disponible pour le signal

(une fois les marges de $V_{DS_{sat}}$ retirées). En effet, dans les circuits analogiques à basse tension, il est couramment admis de poser l'hypothèse que $V_{DS_{sat}}$ est constant indépendamment de la valeur de V_{DD} .

Dans l'exemple choisit figure 4, une réduction de la tension d'alimentation d'un facteur 2 entraîne une réduction de la dynamique réelle disponible pour les signaux d'un facteur 2,5.

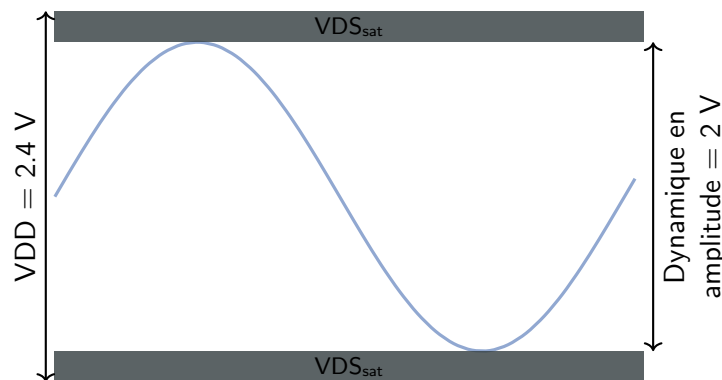


FIGURE 3 – Lien entre tension d'alimentation et dynamique du signal.

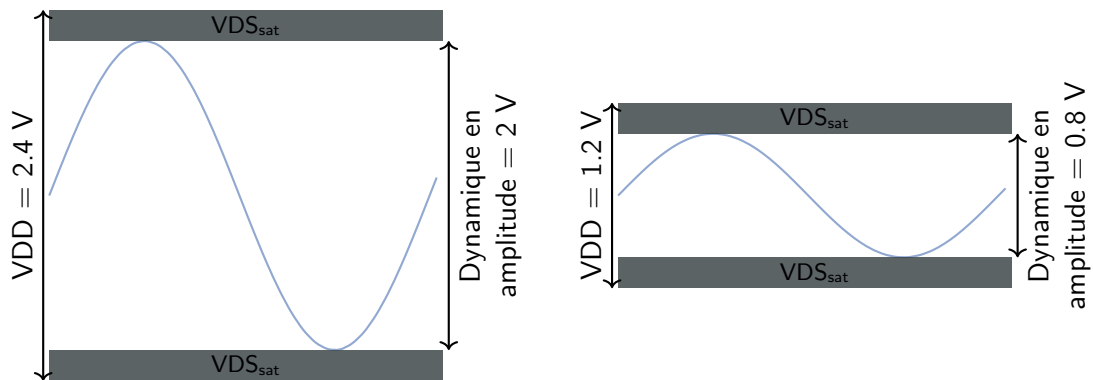


FIGURE 4 – Effet de la réduction des tensions d'alimentation sur la dynamique du signal.

Dans ses travaux [Kinget, 2015], P. Kinget a mis en évidence que la réduction de la dynamique disponible pour représenter les signaux en amplitude engendre un surcoût d'énergie pour maintenir un niveau de performance donné dans le convertisseur. Ce surcoût que P. Kinget nomme *Power Penalty Factor*, soit facteur de pénalité en puissance, augmente exponentiellement quand la dynamique disponible en amplitude et donc la tension d'alimentation diminuent. Cet effet est présenté dans la figure 5 tirée de [Kinget, 2015] et dans le contexte évoqué plus haut de la

réduction des tensions d'alimentation, les conséquences prédites par cet effet ne sont pas optimistes.

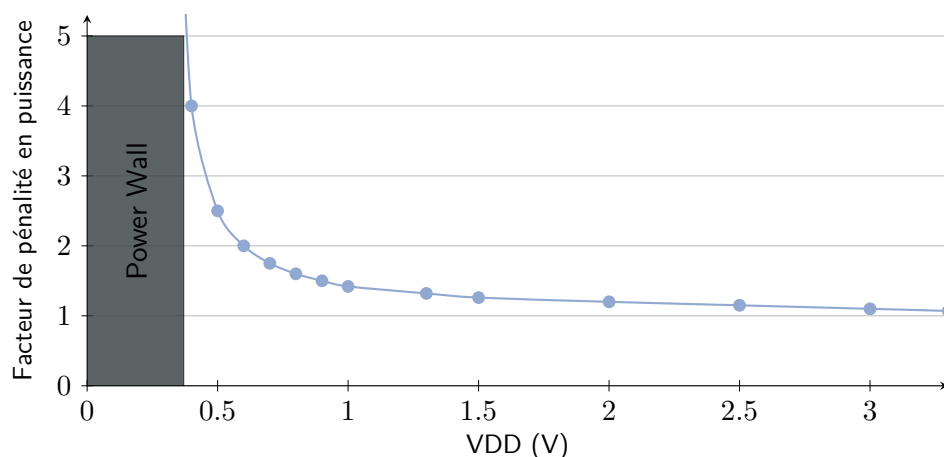


FIGURE 5 – Évolution du facteur de pénalité en puissance en fonction de la tension d'alimentation.

Dans le cas limite, la dynamique est réduite à zéro ; c'est-à-dire qu'il n'y a plus de place de disponible pour coder l'information sur l'amplitude en restant dans la zone de régime linéaire des transistors. Le circuit ne peut alors pas fonctionner. On se heurte à ce que P. Kinget nomme le « Power wall » : un mur infranchissable quelle que soit la puissance.

L'objectif de notre travail va donc être de proposer une alternative à cette représentation traditionnelle des données dans le domaine de l'amplitude afin d'offrir un moyen de contourner la limitation du « Power wall » précédemment énoncée. Nous verrons par la suite que la voie sur laquelle nous avons décidé d'avancer est celle de la représentation dans le domaine temporel.

Ceci constitue le premier axe de recherche de ce travail.

Évolution des besoins applicatifs : ADC à performance variable



Autre part, en rapport avec ce qui a été présenté plus tôt, nous avons souhaité nous intéresser à un autre axe de recherche lié à celui évoqué dans la partie précédente pendant ce travail de thèse : l'impact de l'évolution des besoins applicatifs évoquée précédemment sur les ADC.

Les contraintes (de consommation principalement) évoquées plus tôt ont conduit au développement des systèmes adaptatifs. Ces systèmes ont pour but d'adapter leur performance (et par conséquent leur consommation) afin d'ajuster leur fonctionnement à ce qui est nécessaire sans en faire davantage.

Le cas particulier de fonctionnement variable auquel nous proposons de nous intéresser est le cas des systèmes à deux modes de fonctionnement dont on a également parlé plus tôt. Ces systèmes requièrent à la fois de pouvoir opérer dans un mode performant pour traiter l'information mais également dans un mode dégradé dans lequel la consommation prime sur la performance (état de veille) comme décrit dans le grafctet figure 6.

Pour ces systèmes qui requièrent un convertisseur analogique numérique, il apparaît donc le besoin d'avoir un ADC dont le fonctionnement est compatible avec ce principe de fonctionnement en deux modes.

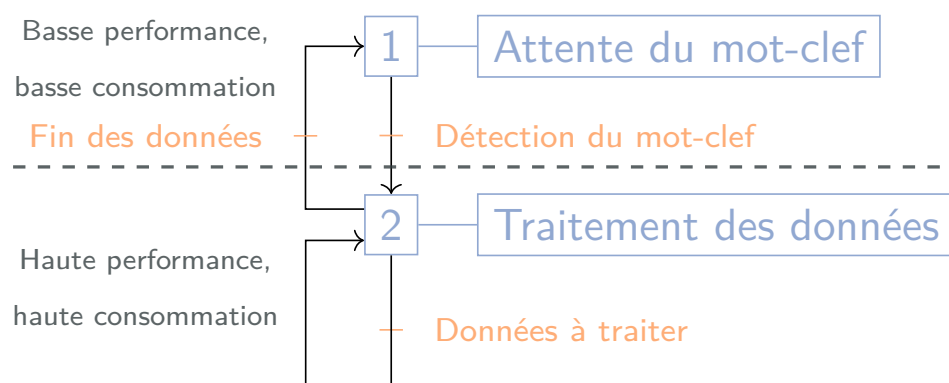


FIGURE 6 – Grafctet de fonctionnement d'un système à deux modes de fonctionnement.

En partant de l'observation de la figure 7 qui représente des signaux numériques servant à coder l'information dans le domaine temporel, on constate que la représentation temporelle permet de manière indépendante de la tension d'alimentation (donc indépendante du V_{dd} et de la dynamique en tension pour peu qu'on ne soit pas dans le cas limite où la dynamique est réduite à zéro) de décider de la dynamique (en temps donc) que l'on souhaite allouer pour représenter le signal. En revanche, la dynamique temporelle dont on décide pour le système va directement impacter sa vitesse de fonctionnement.

Cette observation nous permet de postuler que la représentation d'un signal dans le domaine temporel offre plus de flexibilité vis-à-vis du système (postulat que nous étudierons donc et dont nous détaillerons les limites par la suite), et nous émettons l'hypothèse qu'un convertisseur analogique numérique à perfor-

mances variables qui opère dans le domaine temporel pourrait être réalisé afin de proposer une solution à ce besoin d'un ADC qui fonctionne selon deux modes distincts (un axé sur la performance et un second dégradé axé sur la consommation).

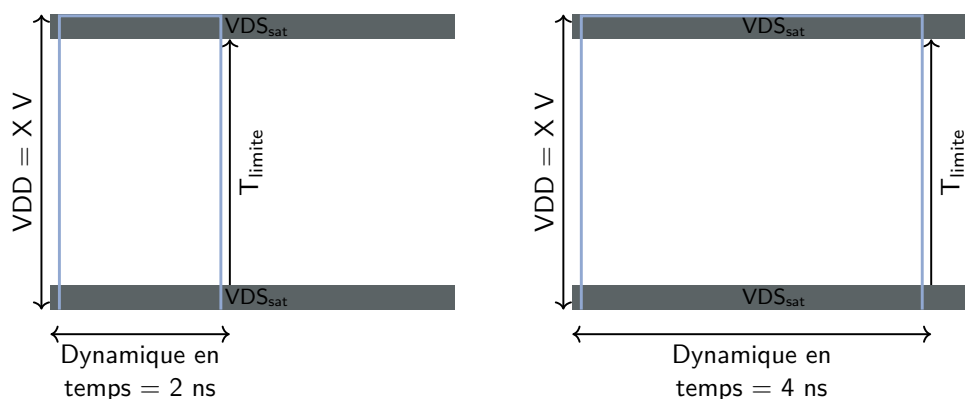


FIGURE 7 – Dynamique arbitrairement grande en temps.

Ce sont donc là les deux axes que nous nous sommes proposés d'étudier et que nous viendrons discuter dans la suite de ce manuscrit :

- D'une part, l'étude de la conversion analogique numérique dans le domaine temporel afin de palier aux problèmes engendrés par la réduction de la dynamique en amplitude des nœuds technologiques avancés ;
- D'autre part, la mise en pratique de la conversion dans le domaine temporel pour répondre au besoin d'un ADC à deux modes de fonctionnement dans les systèmes qui fonctionnent sur ce principe.

Contributions



U regard de la problématique précédemment énoncée, voici les éléments de solution apportés par ce travail et qui seront développés dans ce manuscrit.

Pour adresser la partie représentation de la problématique de conversion analogique numérique dans le domaine temporel, une amélioration d'un convertisseur analogique numérique à oscillateur contrôlé en tension est proposée qui a pour but d'augmenter la résolution de ce dernier pour une bande passante donnée. En ce qui concerne la partie calcul dans le domaine temporel de cette même problématique, une méthode permettant de manipuler des créneaux par le biais d'oscillateurs contrôlés en tension a été développée.

Quant à la partie sur le sujet d'un ADC à deux modes de fonctionnement, une proposition d'architecture répondant à cette fonctionnalité a été proposée et sa

faisabilité a été étudiée.

Ces travaux ont donné lieu à la publication suivante :

- D. Buffeteau, D. Morche, J-L. González-Jiménez, « VCO VerilogAMS model for fast simulation in VCO-based ADC », 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), 2018.

Les travaux ont été réalisés au sein du Laboratoire des Architectures Intégrées Radiofréquences (LAIR) au Département Architecture, Conception et Logiciels Embarqués (DACLE) du Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA) de Grenoble.

Organisation du document



A suite du présent mémoire se divise en cinq chapitres permettant de parcourir le déroulement du travail et de la réflexion menés autour des deux parties de la problématique de conversion dans le domaine temporel d'une part (à la fois la partie représentation de l'information et sa partie traitement) et celle de conversion à deux modes de fonctionnement d'autre part. Les chapitres sont organisés comme suit.

Dans le Chapitre I, un état de l'art permet de balayer les différentes solutions existantes qui répondent à la problématique. Dans un premier temps en s'intéressant aux solutions de conversion dans le domaine temporel et dans un deuxième temps en parcourant les solutions existantes pour calculer dans le domaine temporel. Pour finir, un panorama des convertisseurs à performance variable sera dressé afin de clarifier les pistes d'amélioration à explorer pour répondre à la problématique de convertisseur à deux modes de fonctionnement.

Dans le Chapitre II, une méthode de calcul dans le domaine temporel basée sur des créneaux temporels sera présentée puis discutée. Les limitations de cette méthode seront enfin abordées afin de conclure sur les performances théoriques de cette méthode.

Le Chapitre III quant à lui, servira à exposer la solution proposée pour répondre au besoin d'avoir un ADC à deux modes de fonctionnement. L'idée sous-jacente au concept sera discutée, puis une architecture sera proposée. Enfin, la faisabilité de la solution sera démontrée pour prouver la validité de l'approche.

En ne s'intéressant qu'à une partie de l'architecture présentée Chapitre III, une amélioration d'un convertisseur existant (un convertisseur analogique numérique basé sur un oscillateur contrôlé en tension) sera proposée et étudiée dans le Chapitre IV. Le principe de l'amélioration ainsi que sa faisabilité seront ensuite

discutés. Pour finir, les performances attendues (en simulation) de ce convertisseur seront présentées ainsi que ses limitations.

Enfin, nous concluons ce manuscrit en offrant des perspectives de recherche et d'amélioration des travaux proposés pour le lecteur curieux de pousser davantage les travaux effectués.

Chapitre I

Etat de l'art



« Collection des connaissances. » Crédits : Jordan Rous

AVANT d'avancer davantage dans ce manuscrit, commençons par poser les concepts qui vont servir de fondations aux deux axes principaux de notre travail ; à savoir la conversion dans le domaine temporel et les systèmes (et en particulier les convertisseurs analogique numérique) à performances variables.

I.1 Représentation dans le domaine temporel

I.1.A Définitions et présentation

RÉPRÉSENTER un signal dans le domaine temporel c'est le représenter de sorte que l'information qu'il contient soit, au moins en partie, codée via une grandeur temporelle (la notion de grandeur temporelle sera explicitée par la suite). Afin de clarifier le plus possible ces notions, commençons par décrire une conversion analogique numérique conventionnelle telle qu'elle est présentée par exemple dans [Kar *et al.*, 2015] et illustrée figure I.1.

L'analyse de cette conversion standard et qui ne repose pas sur un codage dans le domaine temporel permettra de mettre en avant les leviers clés de la conversion dans le domaine temporel.

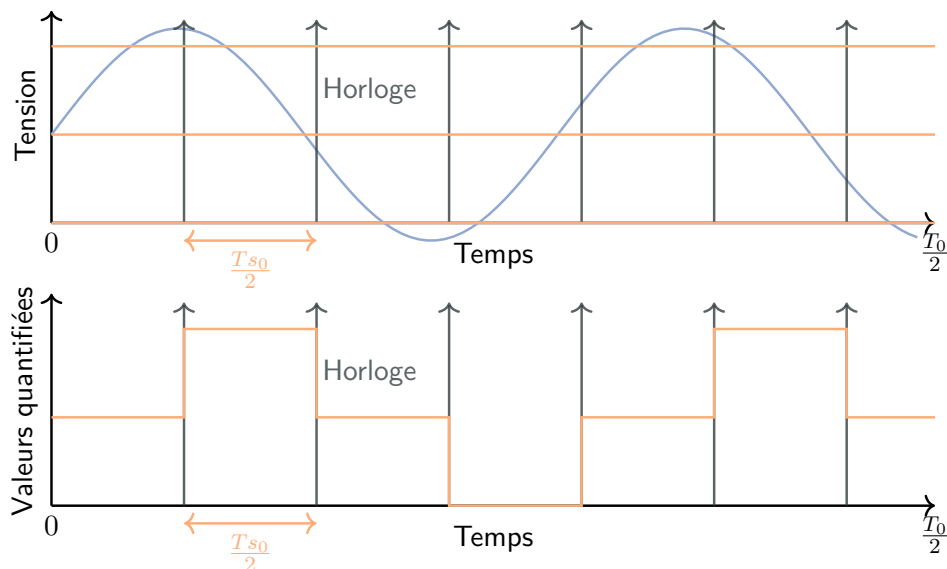


FIGURE I.1 – Une conversion analogique numérique traditionnelle.

Analysons l'exemple de conversion analogique numérique classique décrite dans cette figure. Dans un premier temps, le signal analogique (ici en bleu) est échantillonné aux instants d'échantillonnage (représentés par les flèches en gris

anthracite). Les points ainsi échantillonnés sont ensuite quantifiés ; c'est-à-dire comparés et arrondis aux valeurs des niveaux de quantification. Le signal obtenu est alors le signal composé de ces points échantillonnés qui peuvent être interprétés de la manière la plus simple comme le signal continu par morceaux (représenté en orange) sur le graphe du bas de la figure I.1.

L'information contenue dans le signal analogique de départ se retrouve donc, une fois convertie, sous la forme d'une suite de couples (*instant d'échantillonnage ; valeur quantifiée*) que l'on nommera couples échantillons. Cependant dans une conversion échantillonnée de manière uniforme comme c'est le cas ici (c'est-à-dire que la fréquence d'échantillonnage est fixe) les instants d'échantillonnage présents dans les couples échantillons n'apportent pas en eux-mêmes une information sur le signal. On pourrait très bien se contenter d'un numéro d'échantillon. Seule la fréquence d'échantillonnage et la valeur de l'échantillon contiennent l'information du signal.

Nous ne sommes donc pas dans le cas d'une conversion dans le domaine temporel car la donnée temporelle a priori présente dans les couples échantillons ne transporte pas l'information contenue dans le signal de base.

Il est également intéressant de noter que cette conversion est synchrone ; c'est-à-dire qu'elle produit une sortie (en l'occurrence la valeur numérique correspondant au signal d'entrée) à des instants prédictibles et définis par avance (en l'occurrence les instants d'horloge). Mais ne brûlons pas les étapes, nous reviendrons plus tard sur cette notion.

Procédons maintenant au même exercice que précédemment sur un exemple de conversion dans le domaine temporel afin de bien pointer les différences entre ces deux approches. Etudions pour cela l'exemple présenté dans la figure I.2. Il s'agit de ce que l'on appelle une conversion par détection de seuil. Contrairement à l'exemple précédent dans lequel le signal analogique d'entrée est échantillonné à un intervalle de temps donné, c'est ici la valeur du signal qui va décider de si le signal doit être échantillonné ou non.

Ainsi, la valeur du signal est comparée à des seuils (en orange sur la figure) et le franchissement de ces seuils va déclencher l'échantillonnage du signal. Les points ainsi échantillonnés pourront être interprétés par exemple de la façon retenue dans le graphe du bas.

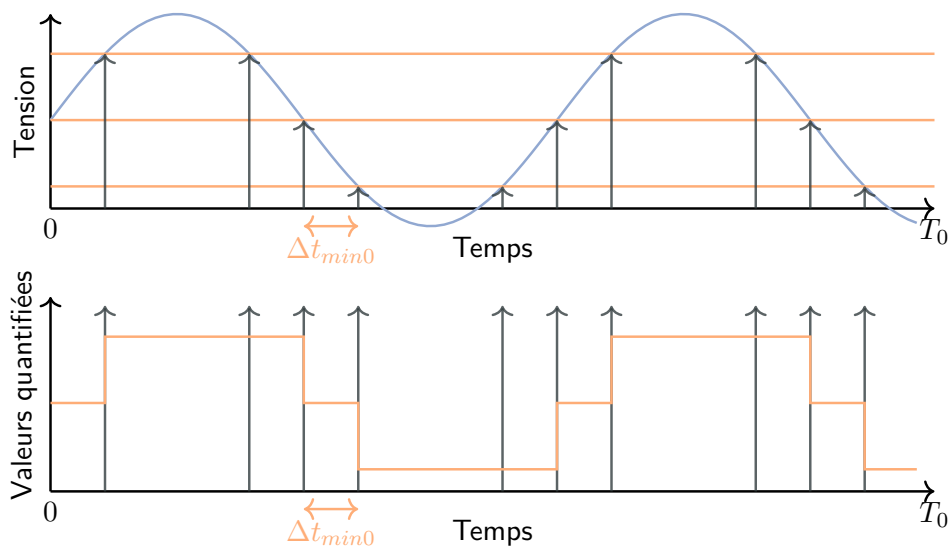


FIGURE 1.2 – Une conversion analogique numérique dans le domaine temporel.

La conversion ainsi effectuée nous donnera une série de couples échantillons (*valeur de l'échantillon ; instant de franchissement*). L'information contenue dans le signal analogique de départ est donc ici convertie en des couples de valeurs dont au moins une des valeurs contenant l'information appartient au domaine temporel. Contrairement à l'exemple traité avant celui-ci, la valeur de l'instant du franchissement contient l'information et dans le cas extrême où les seuils de quantifications sont placés à intervalles réguliers, la suite des instants de franchissement et des sens de franchissement de seuil suffisent à contenir l'information analogique de départ. L'information ici est donc bien codée dans le domaine temporel.

Il est également intéressant de noter (comme il a été fait remarquer pour l'exemple précédent) que cette conversion est asynchrone ; c'est-à-dire qu'elle produit une sortie (en l'occurrence la valeur numérique correspondant au seuil franchi par le signal d'entrée) à des instants non prédictibles et qui dépendent de l'évolution du signal d'entrée. Cela paraît tout à fait cohérent puisque ce sont ces instants qui contiennent l'information du signal analogique de départ.

Si l'on s'accorde pour définir la représentation dans le domaine temporelle comme le fait de représenter tout ou partie de l'information contenue dans un signal analogique par une grandeur appartenant au domaine temporel, une question légitime émerge alors. Qu'est-ce qu'une grandeur dans le domaine temporel ?

Pour apporter des éléments de réponse à cette question nous allons tenter de balayer ce qui se fait dans la littérature en termes de conversion dans le domaine temporel puis plus généralement nous essaierons de proposer un classement des différents types de conversion que nous avons rencontrés afin d'offrir un panorama

du spectre des possibles. Pour cela commençons par répertorier les différentes grandeurs appartenant au domaine temporel qui ont été utilisées pour effectuer une conversion dans le domaine temporel dans la littérature. Nous nous contenterons ici de présenter les méthodes associées aux différentes grandeurs temporelles. Nous détaillerons par la suite ces méthodes afin d'en extraire les points forts et les principales pistes d'amélioration.

La plus intuitive des valeurs temporelles via laquelle on peut représenter une information, c'est celle que l'on a évoquée dans l'exemple précédent ; à savoir l'instant. Intéressons-nous donc à cette représentation pour coder une information qu'est la détection de seuils également appelée level-crossing.

Dans le cas général, les seuils de détection ne sont pas répartis de manière uniforme et les couples échantillons sont donc formés de (*valeur du seuil ; instant de franchissement*). Des exemples de ce type de convertisseurs se trouvent dans la littérature comme [Allier *et al.*, 2003, Guan et Singer, 2007, Kar *et al.*, 2015, Roa *et al.*, 2014, Vezyrtzis et Tsividis, 2009, Sayiner *et al.*, 1996] et la figure I.3 donne une représentation schématique de ce type de conversion.

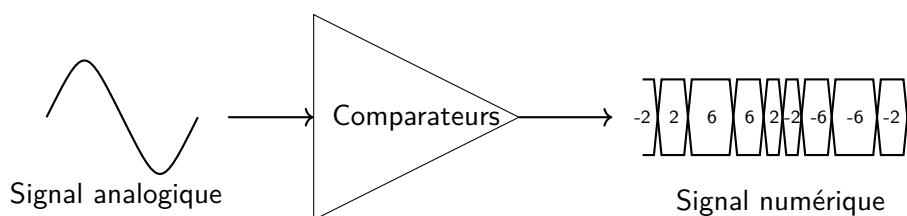


FIGURE I.3 – Schéma de fonctionnement de la détection de seuils.

Nous avons présenté ici un élément phare de la conversion dans le domaine temporel mais un instant n'est pas la seule grandeur temporelle pouvant être utilisée, on peut par exemple utiliser une durée. Une durée ; c'est-à-dire une différence entre deux instants, est par définition une grandeur temporelle. Coder une information sur une durée comme dans le cas d'une modulation de largeur de créneaux (Pulse Width Modulation ou PWM en anglais) est une autre forme de conversion dans le domaine temporel. Le principe de fonctionnement de base d'une telle modulation est schématisé dans la figure I.4.

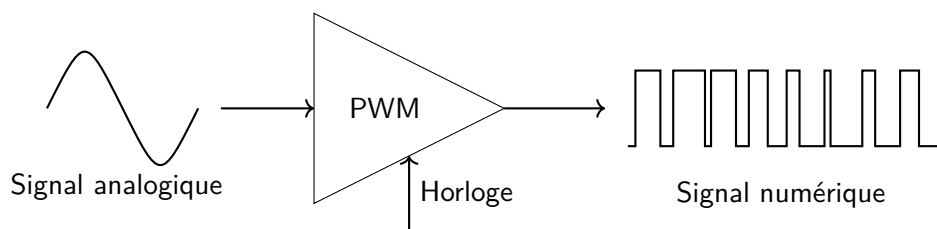


FIGURE 1.4 – Schéma de fonctionnement de la PWM.

Pour illustrer les choses simplement, le signal analogique à convertir est échantillonné par une horloge puis sa valeur sert ensuite à générer un signal crête dont la largeur des crêtes est proportionnelle à la valeur des points échantillonnés. L'information du signal se retrouve donc dans la largeur des crêtes.

Après avoir vu qu'il était possible de représenter un signal analogique par une durée, penchons-nous sur la représentation inverse qui est elle aussi répandue dans la littérature : la représentation sur la fréquence (une fréquence étant l'inverse d'une durée). Une fréquence est également une grandeur appartenant au domaine temporel. On trouve dans la littérature plusieurs travaux qui visent à convertir une donnée analogique par le biais de la fréquence [Unnikrishnan et Vesterbacka, 2016, Kim *et al.*, 2010, Straayer et Perrott, 2008]. Ces exemples se servent d'un oscillateur contrôlé en tension (voltage controlled oscillator en anglais que l'on abrègera dans le reste de ce document en VCO) afin de convertir le signal (en l'occurrence une tension) en un signal de fréquence variable (et c'est cette fréquence qui contient l'information du signal de départ) comme l'illustre le schéma de la figure 1.5.

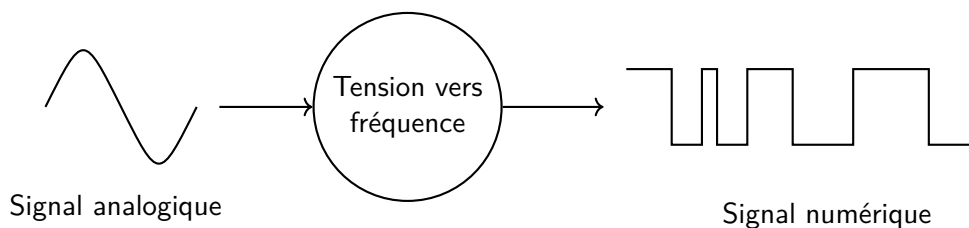


FIGURE 1.5 – Schéma de fonctionnement de la conversion via la fréquence.

En lien direct avec la représentation précédente sur la fréquence, la dernière grandeur temporelle à laquelle nous nous intéresserons dans ce manuscrit sera la phase. La phase peut se définir comme étant l'intégrale de la fréquence. Ce signal dont la définition est encore plus abstraite que la notion de fréquence peut être représenté (et par la même occasion peut être un peu plus aisément conceptualisé) comme la grandeur qui se trouve à l'intérieur de la fonction sinus dans un signal

sinusoïdal. Par exemple, dans le cas d'un sinus de fréquence constante comme dans la figure I.6 la phase (rappelons le, l'intégrale de la fréquence) est donc un signal de pente constante.

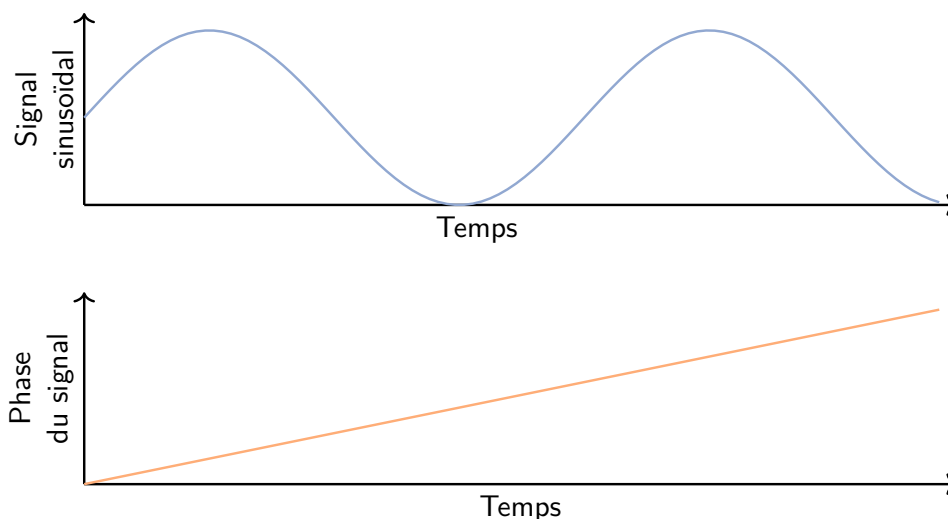


FIGURE I.6 – Représentation de la phase dans le cas d'un signal sinusoïdal de fréquence constante.

La phase est donc une vision de l'esprit et par conséquent n'est pas évidente à visualiser. C'est une grandeur mathématique, tout comme la fréquence et qui est même directement reliée à celle-ci. Une conversion qui se sert de la phase pour représenter un signal analogique se comportera globalement de la même manière qu'une représentation par le biais de la fréquence. Le schéma de la conversion est d'ailleurs le même. Seulement, le fait d'être l'intégrale de la fréquence va tout de même permettre quelques subtilités qui seront évoquées par la suite. On peut notamment se représenter la conversion sur la phase comme une conversion sur la fréquence via un VCO (cf. figure I.5) dont on viendrait quantifier la phase du signal en sortie du VCO comme illustré figure I.7.

Il faut bien comprendre qu'ici le signal représenté dans le graphe du bas de la figure I.7 n'est pas la phase du signal du graphe du haut. Le signal du graphe du bas est la phase d'un signal image en l'occurrence celui issu de la « conversion » par un VCO. L'information contenue dans le signal analogique de départ se retrouve donc dans la phase du signal image.

Toutes les méthodes de représentation évoquées ci-dessus et sur lesquelles nous reviendrons en détails par la suite ont pour but de mettre en évidence le fait que le domaine temporel est vaste et que plusieurs grandeurs, certaines avec une teneur physique très prononcée et d'autres avec une essence plus mathématique, peuvent être choisies comme support de l'information.

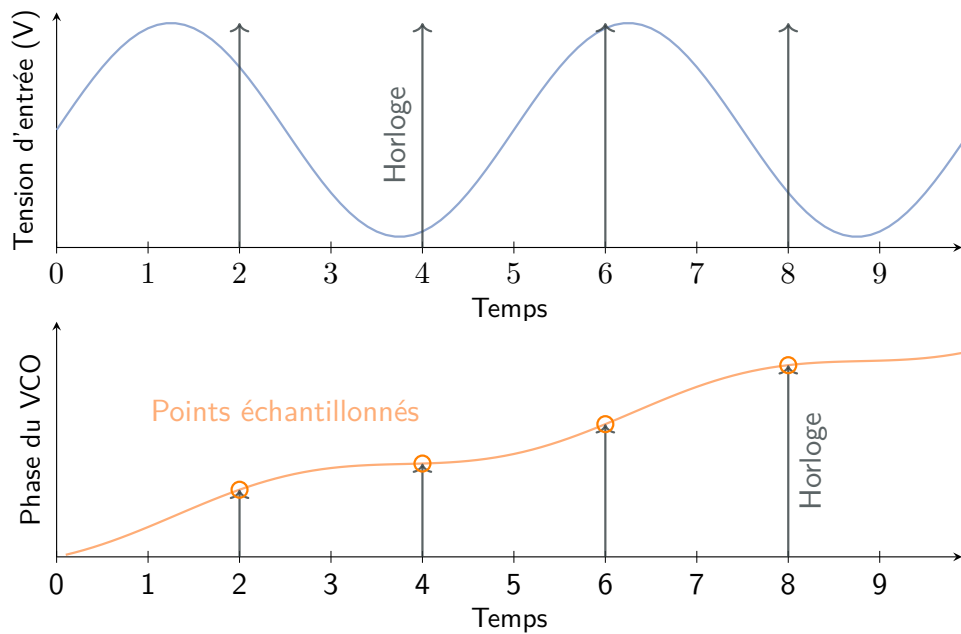


FIGURE 1.7 – Conversion d'un signal analogique en grandeur numérique en passant par une représentation sur la phase via un VCO.

I.1.B Synchronisme et asynchrone



NE autre notion importante relative à la conversion analogique numérique et plus particulièrement à la notion de représentation dans le domaine temporel que l'on peut entrevoir dans ces différents exemples, c'est la nature synchronisme ou asynchrone du procédé de conversion. Avant de détailler davantage les architectures évoquées plus haut, attardons nous un peu plus sur la notion de système synchronisme et asynchrone.

La notion de conversion dans le domaine temporel est étroitement liée à la notion de synchronisme d'un système. Naturellement, comme l'information va être codée sur une grandeur appartenant au domaine temporel, on peut légitimement supposer que la nature de la représentation du temps dans le système (discrète ou continue) va avoir un impact sur la conversion.

Intéressons-nous tout d'abord à définir ce qu'est un système synchronisme et ce qu'est un système asynchrone. Au sens strict, un système synchronisme est un système qui produit des sorties à des instants déterministes pouvant être prédit a priori sans connaissance du signal d'entrée. En d'autres termes, pour un système synchronisme il est possible a priori de connaître exactement les instants auxquels

seront produites les sorties. C'est le cas par exemple d'un système qui produit des sorties selon une horloge. A contrario, un système asynchrone produit des sorties à des instants arbitraires, pouvant intervenir à n'importe quel moment sans qu'il soit possible de les prédire a priori. Le cas de la détection de franchissements de seuils par exemple est un cas de système asynchrone car les franchissements de seuils peuvent intervenir à n'importe quel moment.

De manière moins formelle, on peut étendre le terme de processus asynchrone à tout processus qui se produit indépendamment de toute horloge. Ainsi, si l'on reprend l'exemple de la conversion analogique numérique par le biais de la fréquence dont le schéma de fonctionnement est illustré dans la figure I.5 et dont une application est présentée dans la figure I.10 page 15, bien que le système au sens strict soit synchrone (il produit des sorties de manière déterministe sur front d'horloge comme illustré dans la figure I.10), la conversion du domaine de l'amplitude vers le domaine des fréquence se fait, elle, conformément à ce qui est représenté dans la figure I.5, indépendamment de l'horloge et peut donc être considérée comme asynchrone. Nous reviendrons plus tard sur cette caractéristique et sur les possibilités que nous avons de l'exploiter.

Aussitôt que l'on commence à aborder la notion de synchronicité des différents systèmes évoqués, on vient à faire la distinction entre les systèmes à échantillonnage uniforme (systèmes dont l'échantillonnage est synchrone et régulier) tels qu'utilisés dans les convertisseurs classiques et les systèmes à échantillonnage non uniforme (dont l'échantillonnage est asynchrone et non régulier). Les principales caractéristiques qui permettent de différencier ces deux grandes familles de systèmes sont présentées dans le tableau 1 de [Allier *et al.*, 2003].

On retiendra globalement que les architectures à échantillonnage non-uniforme sont particulièrement adaptées à la conversion de signaux avec une activité faible (des signaux qui globalement bougent peu ou qui se manifestent de façon sporadique). Ces signaux (dont les signaux biologiques sont un exemple très représentatif) mettent en valeur la caractéristique phare de ces architectures ; à savoir la capacité à être déclenchés sur des événements du signal d'entrée (event-driven en anglais). En d'autres termes, la consommation de ces architectures dépend directement de l'activité de l'entrée. Si les signaux d'entrée ont une activité faible, la consommation du bloc sera faible et on trouvera un net avantage à ce genre de choix architecturale comparé à une conversion traditionnelle avec échantillonnage régulier.

À l'inverse, les architectures à échantillonnage uniforme sont elles beaucoup plus adaptées à des signaux rapides et avec une forte activité. Dans ces situations, la présence de l'horloge permet de relâcher les contraintes de conception (flot de conception mieux maîtrisé par les outils) mais elle permet surtout de réguler la fréquence de production des sorties en évitant de produire des sorties plus vite que ce qu'elles pourraient être traitées.

Le tableau I.1 ci-dessous résume parmi les différentes grandeurs temporelles présentées plus tôt, lesquelles se prêtent à la conversion avec échantillonnage non-uniforme.

Grandeur temporelle	Se prête à l'échantillonnage non-uniforme
Instant (level-crossing)	OUI
Durée (PWM)	NON
Fréquence (VCO-based ADC)	EN PARTIE
Phase (VCO-based ADC)	EN PARTIE

TABLEAU I.1 – Échantillonnage non-uniforme pour les systèmes de conversion étudiés

Les deux méthodes qui se prêtent « en partie » à l'échantillonnage non-uniforme ; à savoir celles qui sont utilisées dans les convertisseurs analogique numérique basés sur un oscillateur contrôlé en tension, bien que synchrones dans leurs applications traditionnelles, peuvent potentiellement se décliner en une version échantillonnée non-uniformément (asynchrone) et c'est ce qui va faire l'objet de l'étude présentée dans la partie III.

I.1.C Conversion dans le domaine temporel : avantages et limitations



DANS cette partie, nous nous efforcerons de préciser les différentes méthodes de conversions illustrées dans la partie I.1.A. Pour cela, nous descendrons d'un niveau d'abstraction afin de visualiser les différents signaux et leur évolution dans chacune de ces conversions. À cette occasion nous mettrons en exergue les points forts de chacune de ces méthodes ainsi que les axes d'amélioration inhérents à chaque type de conversion.

I.1.C.1 Le codage sur les instants : le level-crossing

Les grandes idées de la conversion par détection de seuils (level-crossing en anglais) avec échantillonnage non uniforme sont développées dans [Inose *et al.*, 1966] et [Allier *et al.*, 2003]. Nous allons essayer de synthétiser ici ces principes afin de lister les principaux avantages de cette conversion et les points qui nécessitent des améliorations.

Commençons par le premier des points forts de cette représentation. Après la problématique introduite au début de ce manuscrit ; à savoir la contrainte apportée par la réduction des tailles de transistors et la diminution des tensions

d'alimentation, la possibilité de représenter un signal analogique uniquement sous la forme de nombres entiers (correspondant aux valeurs des seuils) et d'instant (correspondant aux instants de franchissement) permet de limiter les contraintes de conception.

De plus, comme l'information se retrouve stockée dans la valeur des instants de franchissements de seuils, c'est la précision atteignable sur la valeur de ces instants qui va définir la précision atteignable de la conversion. Il se trouve là encore que la réduction des tailles de transistors apporte une amélioration. En effet, comme on le constate dans les circuits numériques, la réduction des tailles de transistors entraîne une augmentation de la vitesse de fonctionnement des transistors et par la même une augmentation de la fréquence maximale de fonctionnement des circuits. En d'autres termes, la résolution temporelle des circuits électroniques augmente avec la réduction des tailles de transistors rendant ainsi la représentation temporelle de plus en plus précise.

La figure I.8 permet d'illustrer de manière générale comment se déroule l'opération de conversion par franchissements de seuils. Les instants d'échantillonnage sont ainsi définis à chaque fois que le signal analogique franchit un seuil et la valeur retenue pour l'échantillon est la valeur du seuil franchi. On notera que ce système est bien à échantillonnage non-uniforme et bénéficie donc des avantages présentés dans la partie I.1.B.

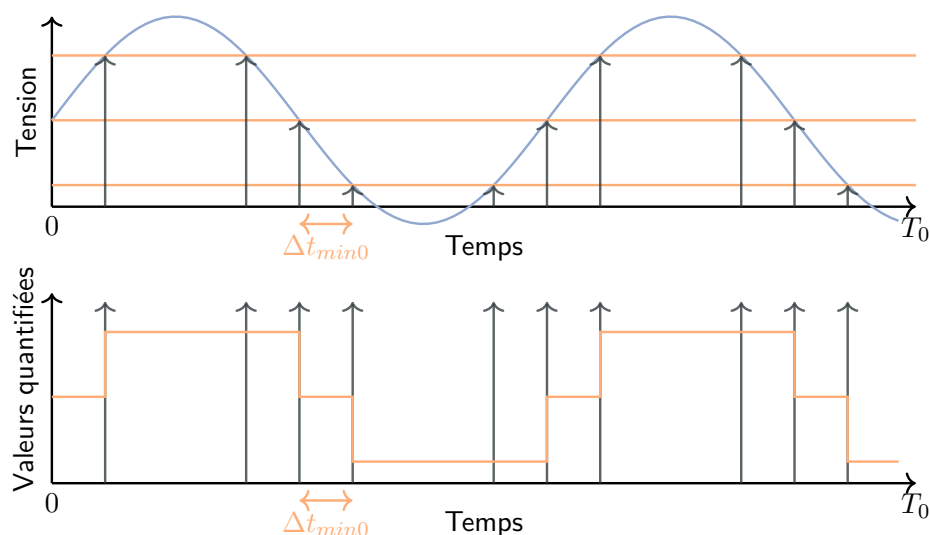


FIGURE I.8 – Visualisation des signaux lors d'une conversion par level-crossing.

Enfin, ce type de conversion bénéficie comme évoqué au début de ce manuscrit d'une dynamique pour représenter les signaux virtuellement illimitée. Si l'on ajuste le nombre de seuils ainsi que leurs positions, on peut récupérer un signal

arbitrairement grand.

Nous employons l'expression *virtuellement illimitée* car la dynamique illimitée reste théorique. Dans ce cas précis, il n'est pas réalisable de disposer d'une quantité arbitrairement grande de seuils de comparaison pour pouvoir détecter les franchissements d'un signal arbitrairement grand.

De surcroît, comme ce sont les franchissements de seuils qui génèrent des évènements, plus l'activité du signal sera grande plus les évènements seront générés en nombre. Ainsi, les signaux avec une variation très rapides génèreront une grandes quantités de données ce qui non seulement génèrera une forte consommation du circuit mais en plus génèrera une grande quantité d'évènements très rapides qui nécessiteront un bloc de traitement sur-dimensionné en aval du convertisseur.

Pour pallier à ces phénomènes, certains travaux comme par exemple [Patil *et al.*, 2015] proposent des modifications par rapport à une architecture simpliste : les deux idées proposées sont d'une part de maintenir le signal analogique entre deux seuils pour n'avoir que deux seuils indépendamment de la dynamique du signal en soustrayant la valeur du seuil au signal quand celui-ci le franchit. La seconde idée est de filtrer les composantes haute fréquence du signal analogique d'entrée afin de ne pas se retrouver dans le cas d'un signal de forte activité.

Ces améliorations permettent d'obtenir un convertisseur avec une consommation adaptée à l'activité du signal (avantage de l'échantillonnage non-uniforme) entre 8 et 24 μW et d'offrir un rapport signal à bruit plus distorsion (SNDR) d'environ 40 dB sur une bande passante de 40 MHz.

Ces aspects mènent à la conclusion que les convertisseurs analogique numérique qui codent l'information dans le domaine temporel sous la forme d'instantants sont particulièrement adaptés aux signaux de faibles activités et de dérivée faible ; c'est-à-dire plus généralement à des signaux relativement lents et avec peu d'évènements. Dans ces conditions favorables, ils permettent d'atteindre des consommations particulièrement basses qui sont extrêmement bien adaptées à des applications très basse consommation dans les domaines biologiques ou dans le traitement de la voix.

Dans le cas du travail réalisé dans [Roa *et al.*, 2014], qui vise à convertir un signal biologique de très faible activité (quelques occurrences par jour) et dont la forme caractéristique est connue a priori, le gain apporté en termes de consommation par rapport à une conversion à échantillonnage régulier est rapporté de l'ordre de 99%.

I.1.C.2 Le codage sur les durées : la PWM

Procédons de la même manière avec le codage sur la durée des impulsions. Il existe énormément de façons de faire une modulation sur la largeur d'impulsions. Le principe sous-jacent de ces différentes techniques reste le même ; à savoir la modification de la largeur d'un créneau en fonction de la valeur d'un signal.

En effet, s'il est de plus en plus difficile et de plus en plus coûteux de déterminer la valeur de l'amplitude d'un signal (cf. figure 5), il reste beaucoup plus facile de déterminer si cette valeur vaut « 0 » ou « 1 ». Rappelons-le, les circuits numériques tirent énormément d'avantages du contexte de réduction des tailles de transistors. Se retrouver avec les mêmes contraintes que celles de la conception numérique est un gros avantage pour de la conception analogique.

Regardons de plus près ce qu'il se passe au niveau des signaux lors d'une conversion par modulation de largeur de créneaux. La figure I.9 illustre un exemple de conversion. À chaque front d'horloge, un créneau est généré dont la largeur est l'image de l'amplitude du signal analogique à convertir. L'information utile se retrouve ainsi dans la largeur des créneaux produits.

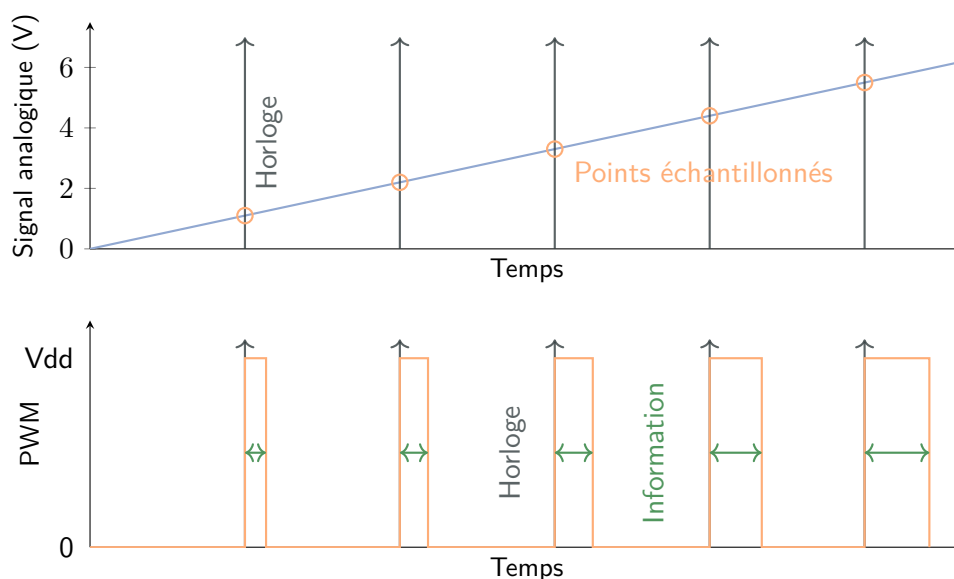


FIGURE I.9 – Visualisation des signaux lors d'une conversion par PWM.

Outre le fait de ne représenter le signal que par deux valeurs distinctes (0 et Vdd) comme c'est le but recherché, cette représentation présente l'avantage de pouvoir choisir la dynamique allouée pour représenter le signal analogique.

Là encore, on constate qu'il n'est cependant pas possible de convertir un signal analogique arbitrairement grand puisque cela conduirait à pouvoir représenter des créneaux arbitrairement longs et donc à avoir une fréquence d'horloge arbitrairement faible. Rendant ainsi le système infiniment lent.

Ces systèmes, à l'instar des systèmes de conversion par détection de seuils, bénéficient eux aussi de l'augmentation des fréquences de fonctionnement des nœuds technologiques avancés. Ils peuvent ainsi représenter des créneaux dont la largeur est d'autant plus précise.

Ces systèmes sont cependant connus pour leur bande passante limitée et la présence d'une horloge ne permet pas de bénéficier des avantages offerts par les circuits à échantillonnage non-uniforme en termes d'adaptation de la consommation du système en fonction de l'activité du signal d'entrée.

I.1.C.3 Le codage sur la fréquence : le VCO-based ADC

Avant de s'intéresser aux signaux lors de la conversion analogique numérique par le biais de la fréquence, penchons-nous d'abord sur le bloc au cœur de cette conversion : l'oscillateur contrôlé en tension.

Ce bloc électronique remplit exactement le rôle désiré pour la conversion tension vers fréquence ; à savoir qu'il produit un signal (créneau dans le cas que l'on étudie) dont la fréquence d'oscillation varie en fonction d'une tension analogique qui lui est fournie en consigne. Ce bloc génère ainsi un signal dont la fréquence est l'image d'une tension analogique. C'est bien ce que l'on recherche.

Ce type de conversion est très prometteur car il permet de s'affranchir complètement du codage sur l'amplitude (l'information est codée entièrement sur la fréquence) et contrairement aux méthodes précédemment présentées plus aucune grandeur appartenant au domaine de l'amplitude n'est requise pour représenter l'information codée dans le signal de départ. De plus, la possibilité de faire un VCO composé uniquement de cellule standards comme on le trouve par exemple dans [Unnikrishnan et Vesterbacka, 2014] permet de tirer profit des avancées technologiques vis-à-vis des circuits numériques (augmentation de l'intégration, amélioration de la vitesse de fonctionnement ou encore amélioration de la consommation).

Pour récapituler, la conversion va consister à produire un signal en créneau dont la fréquence instantanée est l'image du signal analogique de départ et ensuite à venir quantifier la valeur moyenne de la fréquence de ce signal créneau sur une période d'horloge. On récupère ainsi la valeur moyenne du signal analogique de départ sur la période d'horloge. Ce comportement est illustré dans la figure I.10

(pour essayer de clarifier le déroulement de la conversion). On notera ici qu'un compteur est utilisé pour estimer la fréquence du signal de sortie du VCO comme c'est le cas dans beaucoup d'articles de la littérature. Dans l'exemple, le compteur enregistre les fronts montants et les fronts descendants.

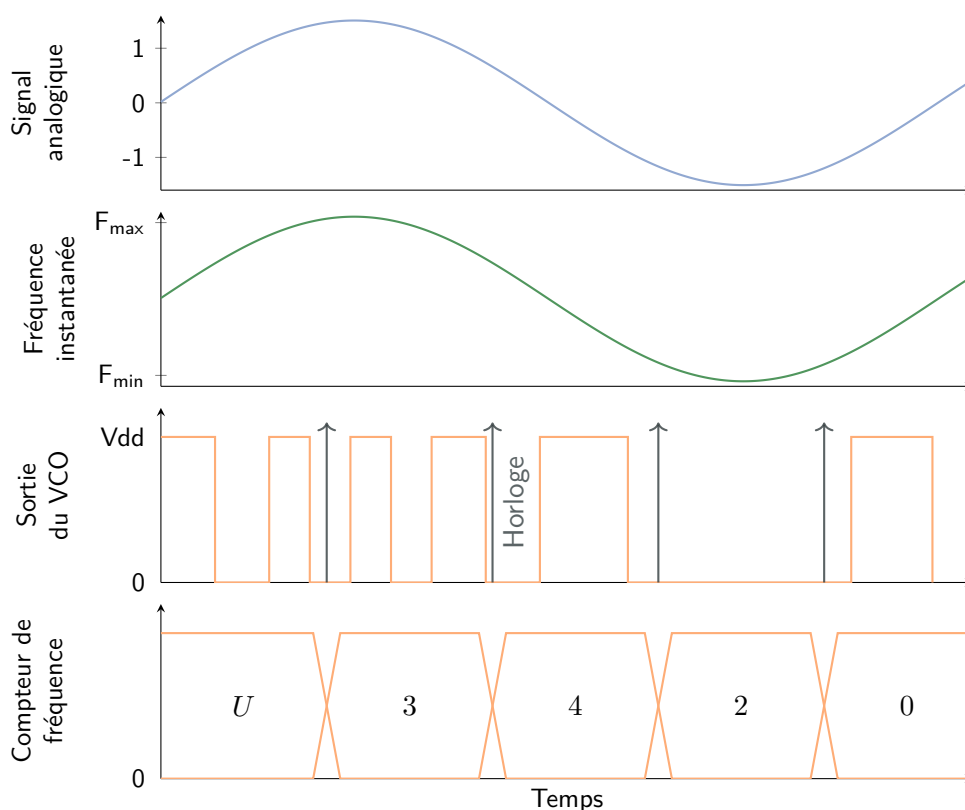


FIGURE 1.10 – Visualisation des signaux lors d'une conversion par un ADC basé sur un VCO.

Comme expliqué plus tôt, le VCO va transformer le signal analogique (en bleu sur le graphe du haut) en un signal dont la fréquence instantanée est l'image de ce signal analogique (le créneau en orange dans le troisième graphe). Afin de mieux visualiser la conversion, la fréquence instantanée du signal de sortie du VCO qui représente le support de l'information est tracée en vert dans le deuxième graphe. Pour finir, un compteur (en l'occurrence sensible à la fois aux fronts montant et aux front descendant du signal de sortie du VCO) va compter les fronts du signal de sortie du VCO pendant une période d'horloge avec remise à zéro à chaque nouveau front d'horloge. Ce compteur estime ainsi la fréquence du signal de sortie du VCO sur une période d'horloge en une valeur numérique (visible sur le quatrième graphe).

Tout le processus de conversion repose cependant sur la capacité à transposer fidèlement l'information contenue dans le signal analogique dans le domaine temporel sous la forme d'une fréquence. Malheureusement, les VCO, et plus particulièrement ceux composés de cellules standard rebouclées (appelés oscillateurs en anneau du fait de leur construction) présentent un défaut majeur qui va directement impacter cette conversion tension vers fréquence. Leur caractéristique de conversion tension vers fréquence est fortement non-linéaire. Il suffit de regarder la courbe caractéristique d'un VCO composé de 31 cellules standard NAND en technologie FDSOI 28 *nm* fournie en figure I.11 pour se rendre immédiatement compte de cette non-linéarité.

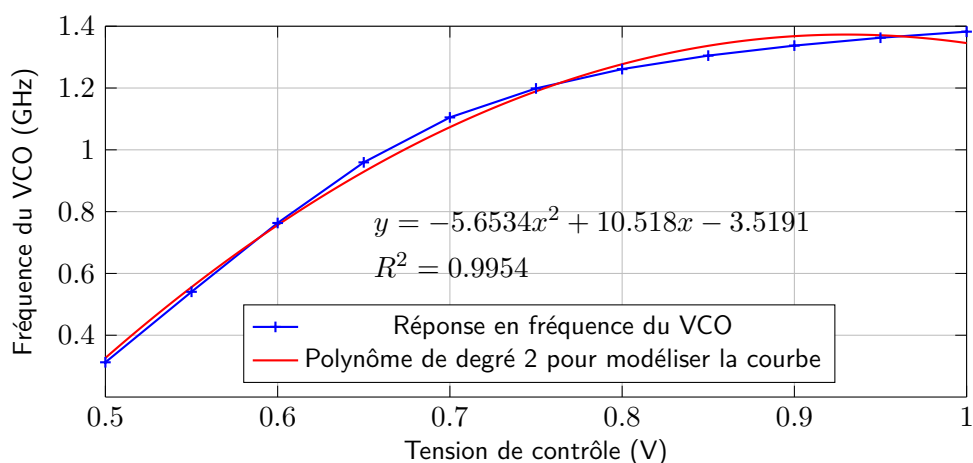


FIGURE I.11 – Caractéristique tension fréquence d'un VCO composé de 31 cellules NAND en FDSOI 28 nm.

Ce phénomène implique que la fréquence du VCO n'est pas une image fidèle du signal analogique et entraîne donc une erreur de représentation. Ce défaut rend ce processus bien moins idéal qu'il n'y paraît et fait de ce type d'architectures l'objet du travail de plusieurs équipes ayant donné lieu à plusieurs publications telles que [Ghosh et Pamarti, 2015, K. Lee *et al.*, 2015, El-Halwagy *et al.*, 2013, Gao *et al.*, 2012, Hernández *et al.*, 2011, Dhanasekaran *et al.*, 2009, Taylor et Galton, 2013, Taylor et Galton, 2010]. Chacune présente différentes techniques pour pallier à la non-linéarité des VCO telles que (de manière désordonnée et non exhaustive) :

- la correction de la non-linéarité par un bloc qui vient rajouter du bruit sur le signal d'entrée (dithering) [Ghosh et Pamarti, 2015] ;
- la suppression de la non-linéarité par ajout d'une modulation PWM en amont du VCO [Gao *et al.*, 2012][Hernández *et al.*, 2011] ;
- la correction de la non-linéarité par un bloc numérique qui va venir compenser la caractéristique tension-fréquence du VCO [Taylor et Galton, 2013][Taylor et Galton, 2010] ;

- un montage différentiel pour diviser par deux la dynamique des tensions de contrôle d'entrée du VCO afin de bénéficier d'une meilleure linéarité sur une dynamique inférieure [K. Lee *et al.*, 2015] ;
- un rebouclage de la sortie du convertisseur pour donner une architecture de convertisseur $\Sigma\Delta$ [El-Halwagy *et al.*, 2013][Hernández *et al.*, 2011][Dhanasekaran *et al.*, 2009][K. Lee *et al.*, 2015].

Ce qu'il y a à retenir de la littérature concernant les convertisseurs analogique numérique basés sur des oscillateurs contrôlés en tension, c'est que les meilleurs résultats obtenus le sont avec des convertisseurs pseudo-différentiels qui permettent d'annuler les non-linéarités d'ordre pair mis au sein d'une architecture de convertisseur $\Sigma\Delta$ qui a pour effet de réduire drastiquement la dynamique du signal qui entre dans le VCO, le faisant ainsi travailler sur une plus petite plage et augmentant par conséquent sa linéarité.

Ces convertisseurs peuvent atteindre des résolutions relativement élevées (de l'ordre de 70 dB) mais sont limités en bande passante par le facteur de sur-échantillonnage propre aux architectures $\Sigma\Delta$ qui impose la fréquence d'échantillonnage.

Ce sont donc pour obtenir des conversions avec une résolution moyenne sur des bandes passantes modestes (de quelques MHz à 50 MHz) que ces convertisseurs présentent les meilleures performances. Ils ont de surcroît la capacité de travailler à basse tension d'alimentation de par le fait qu'ils opèrent dans le domaine temporel.

1.1.C.4 Le codage sur la phase : le VCO-based ADC

Dans cette partie nous allons nous efforcer de démontrer qu'un convertisseur analogique numérique basé sur un oscillateur contrôlé en tension peut également servir à convertir l'information analogique en passant par le domaine de la phase.

Nous allons donc reprendre le même circuit que celui illustré dans la partie 1.1.C.3 auquel nous appliquerons le même signal d'entrée et qui en toute logique produira le même signal de sortie. La différence cependant va être de comprendre le mécanisme qui se passe à l'intérieur du VCO en l'analysant dans le domaine de la phase.

La figure 1.12 va permettre d'appuyer ce raisonnement en offrant une vue sur les signaux dans le domaine de la phase. Rappelons, si besoin est, que ces signaux sont une vue de l'esprit.

Repartons donc du même point de départ ; à savoir le signal analogique en bleu sur le premier graphe que l'on met en entrée d'un VCO. En faisant correspondre les oscillations du VCO (signal créneau orange sur le troisième graphe) avec ce

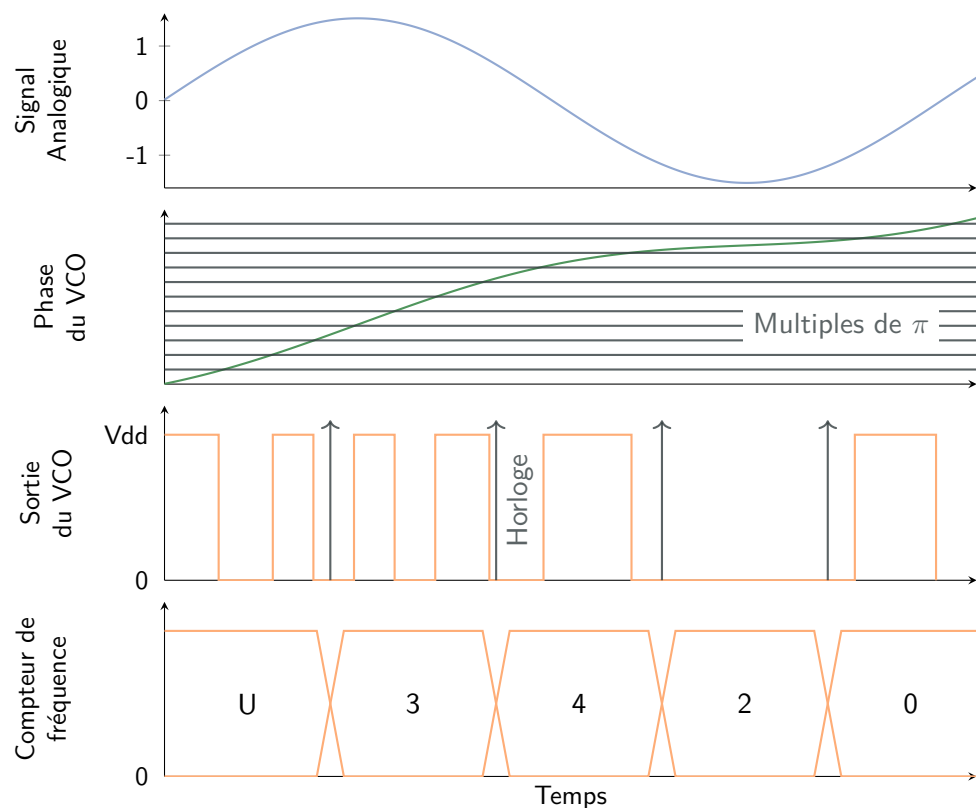


FIGURE 1.12 – Visualisation des signaux lors d’une conversion par un VCO-based ADC (vue dans le domaine de la phase).

qu’il se passe dans le domaine de la phase comme illustré par le deuxième graphe, on montre que les fronts du signal de sortie du VCO (fronts montants et fronts descendants) correspondent aux instants où la phase (en vert sur le deuxième graphe) franchit un des seuils (en gris sur le deuxième graphe). Ces seuils sont placés tous les multiples de π ce qui revient à dire qu’une période du VCO est équivalente à une rotation complète (2π) de la phase.

En adoptant ce point de vue, le compteur qui sert à produire l’information numérique ne va pas avoir pour rôle cette fois d’estimer la valeur moyenne de la fréquence sur une période d’horloge mais il va servir à estimer l’accroissement de la phase pendant une période de l’horloge. Les sorties produites étant identiques au convertisseur précédent (puisque’il ne s’agit que d’une vision de l’esprit avec un autre point de vue sur ce qu’est le VCO-based ADC) il va de soi que les avantages et les limitations de cette conversion sont partagés avec la conversion dans le domaine de la fréquence.

Cependant, on met surtout en évidence ici le fait qu'un VCO-based ADC comprend de manière intrinsèque à la fois une conversion à échantillonnage uniforme représentée par l'échantillonnage par l'horloge de la valeur de la phase et dans le même temps une conversion à échantillonnage non-uniforme (en l'occurrence par franchissement de seuils) puisque la sortie du VCO est complètement asynchrone et correspond à une conversion par franchissement de seuils de la phase du VCO. C'est cette observation très importante qui guidera notre choix ultérieurement pour proposer un début de solution pour réaliser un convertisseur à deux modes de fonctionnement.

I.1.D Bilan et positionnement des différentes méthodes vis-à-vis de l'état de l'art



PRÈS avoir présenté et illustré les différentes méthodes qui serviront de socle de base à ce travail, cette partie va servir à récapituler les différents points clés évoqués plus haut et mettre ces méthodes en perspectives par rapport à l'état de l'art pour permettre d'y voir plus clair sur les pistes de travail à envisager et les points sur lesquels une amélioration peut être proposée. Pour finir nous présenterons les choix qui ont été faits pour ce travail sur la problématique de représentation dans le domaine temporel.

Comme il a été discuté, les convertisseurs à franchissement de seuils sont particulièrement adaptés pour un fonctionnement asynchrone avec un échantillonnage non-uniforme. Ainsi ils ne possèdent pas d'horloge (qui représente une source de forte consommation d'énergie) et leur consommation dépend de l'activité du signal d'entrée. Cette propriété fait d'eux d'excellents candidats pour les circuits basse consommation et cela peut être mesuré par une grandeur que l'on appelle la figure de mérite (en l'occurrence nous parlons de la figure de mérite de Walden qui représente l'énergie dépensée par le convertisseur par pas de conversion). Dans la grande étude comparative sur les convertisseurs menée par Boris Murman [Murmann, 2016] cette figure de mérite est tracée dans un graphe qui permet de mesurer l'énergie nécessaire à une conversion en fonction de la vitesse du convertisseur. Nous la reproduisons en figure I.13. Les convertisseurs de ce type s'illustrent par leur positionnement dans ce graphe comme par exemple [Patil *et al.*, 2015].

Mais s'ils sont excellents en ce qui concerne la consommation énergétique, ces convertisseurs ne permettent pas d'atteindre de hautes performances en termes de résolution ou de bande passante. Cette limitation impose de se restreindre à des applications peu gourmandes en termes de vitesse et de résolution. Si par exemple nous décidons de nous intéresser aux nouveaux besoins apparus avec l'évolution des technologies dont on a parlé en introduction, ces convertisseurs ne présentent pas les performances suffisantes. Ce ne sont donc pas de bons candidats pour

adresser des modes de fonctionnement demandant de hautes performances.

Les convertisseurs par modulation de largeur de créneaux, même s'ils permettent d'atteindre de meilleures résolutions que la famille des convertisseurs à détection de seuils, sont limités en bande passante. De plus, ils nécessitent une horloge et ne bénéficient donc pas des avantages apportés par l'échantillonnage non-uniforme.

A l'inverse, l'architecture qui repose sur des VCO présentée plus tôt, et qui suscite de plus en plus l'engouement de la communauté, et ses variantes permettent d'obtenir de bien meilleures performances en termes de résolution que les convertisseurs à détection de seuils, sans pour autant faire partie des meilleurs ADC. Dans le graphe I.14 dont les données proviennent de [Murmman, 2016], on constate que les VCO-based ADC restent dans la moyenne en termes de performance. Leur limitation en bande passante vient du fait de leur fonctionnement qui nécessite un facteur de sur-échantillonnage élevé. On peut raisonnablement émettre l'hypothèse que leurs performances moyennes sont en partie dues à la relative nouveauté de ces architectures (qui possèdent donc une plage d'amélioration conséquente) ainsi qu'au fait que seul des circuits dans des technologies relativement anciennes ont été présentés. Leur conception très fortement orientée numérique est également un plus qui présage un bien meilleur futur que des convertisseurs qui se basent sur des comparateurs pour fonctionner et permet de tirer profit des évolutions technologiques notamment en termes de fréquence.

Du point de vue de la consommation cette fois (ou plutôt de la figure de mérite pour avoir une comparaison normée) les VCO-based ADC ne sont clairement pas aussi efficaces que les convertisseurs à franchissement de seuils. Cela peut être observé sur la figure I.13. On peut faire l'hypothèse que cette consommation plus élevée est liée au fait que contrairement aux convertisseurs précédents qui peuvent tirer avantage de leur qualité event-driven et donc adapter leur activité (et par conséquent leur consommation) en fonction de l'activité du signal d'entrée à convertir, les VCO-based ADC nécessitent en permanence à la fois l'activité de l'horloge d'échantillonnage mais également le fonctionnement encore plus rapide de l'oscillateur contrôlé en tension. Ceci dit, leur positionnement en termes de figure de mérite reste pourtant tout à fait correct. Ces convertisseurs restent donc de bons candidats pour des applications basse consommation.

Pour conclure cette partie, les observations et les conclusions développées précédemment sont consignées dans le tableau récapitulatif ci-après (tableau I.2) pour offrir une vue plus synthétique de ce qui a été abordé.

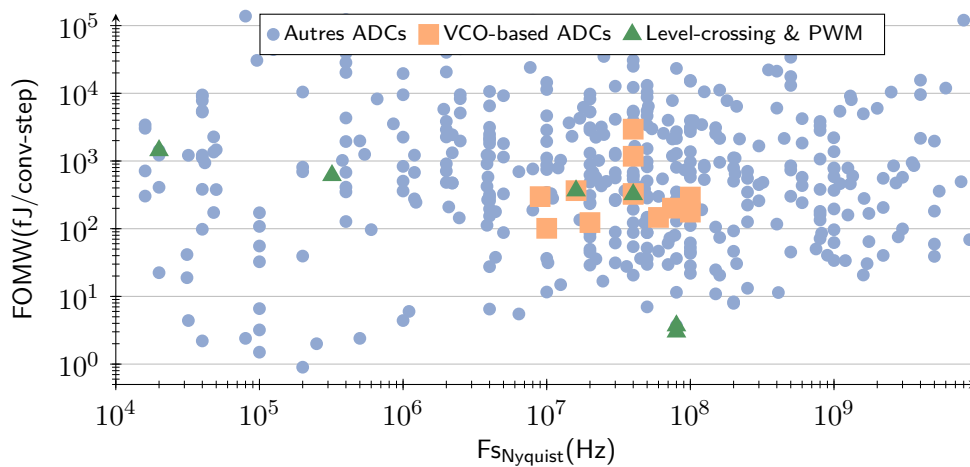


FIGURE I.13 – Positionnement des ADCs en termes d'efficacité énergétique.

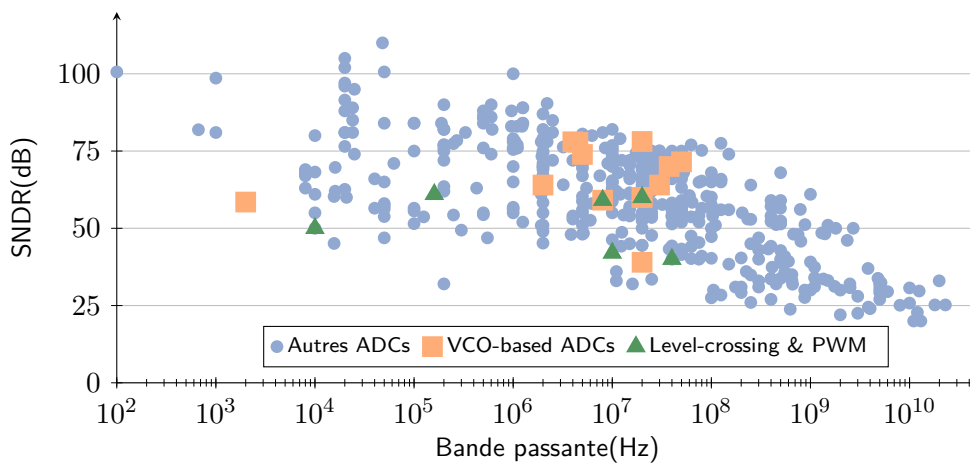


FIGURE I.14 – Positionnement des ADCs en termes de résolution.

Méthode	Asyn- chrone	BW >= 50 MHz	Résolu- tion élevée	Principal avantage	Principal défaut
Franchisse- ment des seuils	😊	😊	😊	Event-driven	Génération d'évènements en BW élevée
PWM	😞	😞	😊	Dynamique facilement variable	Si dynamique élevée alors vitesse lente
VCO-based ADC fréquence	😊	😊	😊 21	Profite du design numérique	Non-linéarité et bruit de phase
VCO-based ADC phase	😊	😊	😊	Profite du design numérique	Non-linéarité et bruit de phase

TABEAU I.2 – Tableau récapitulatif de la conversion dans le domaine temporel

Il est donc proposé d'explorer une architecture de convertisseur analogique numérique basé sur des oscillateurs contrôlés en tension qui améliorerait le positionnement de ce type de convertisseur en termes de résolution et de bande passante.

I.2 Traitement dans le domaine temporel



ONVERTIR une grandeur analogique en une grandeur numérique par le biais du domaine temporel est une chose. Mais y a-t-il obligatoirement besoin de convertir la donnée dans le domaine numérique pour pouvoir la traiter ? On peut naturellement se demander s'il n'y a pas la possibilité de traiter la donnée directement dans le domaine temporel plutôt que d'avoir recours à une numérisation pour permettre le traitement de la donnée de manière plus conventionnelle (avec une unité de traitement numérique ou DSP en anglais).

Dans cette partie nous allons parcourir ce qui existe dans la littérature en termes de calculs dans le domaine temporel. Il sera également entrepris d'analyser ces différentes solutions afin de mettre en exergue les goulots d'étranglement et d'apporter notre propre contribution à la résolution de cette problématique.

I.2.A Calcul temporel sur les créneaux de G. Roberts



CE qui se rapproche le plus de la notion intuitive qu'il est possible d'avoir à propos du calcul dans le domaine temporel se trouve dans les travaux de Gordon W. Roberts dont les bases sont présentées dans [Roberts et Ali-Bakhshian, 2010].

Il y est proposé de faire des calculs directement sur la durée de créneaux. Les figures I.15, I.16 et I.17 présentent ce qui est retenu pour réaliser les différentes opérations mathématiques de base. On peut ainsi se représenter comment fonctionnent les opérations d'addition, de soustraction ou encore de multiplication par un nombre entier.

Ainsi pour additionner deux valeurs A et B, il « suffit » de mettre un créneau de longueur B à la suite d'un créneau de longueur A.

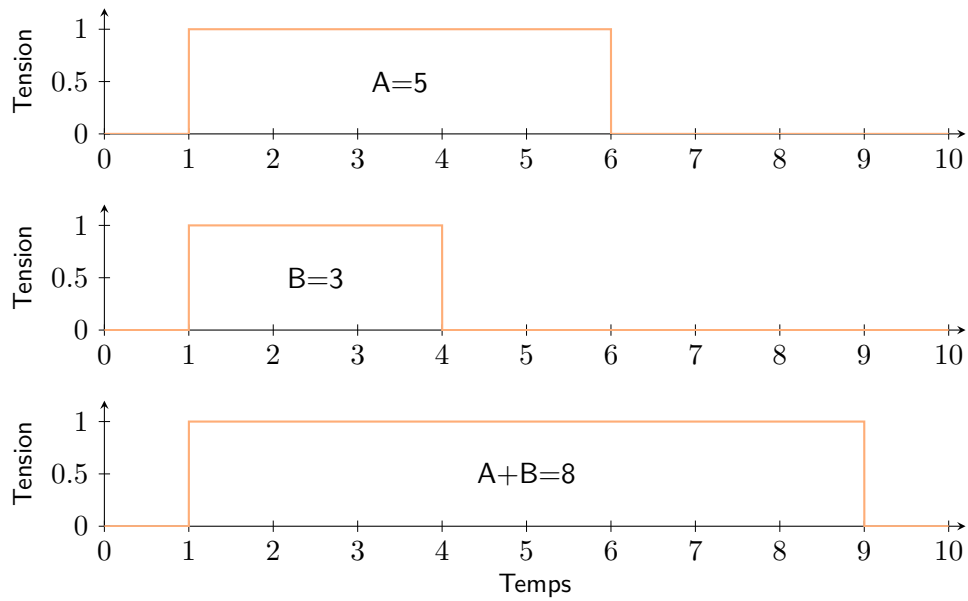


FIGURE I.15 – Principe d'addition sur les créneaux de G. Roberts.

De la même façon, une soustraction consiste à créer un créneau dont la longueur est la différence de longueurs entre le créneaux le plus long et le créneau le plus court.

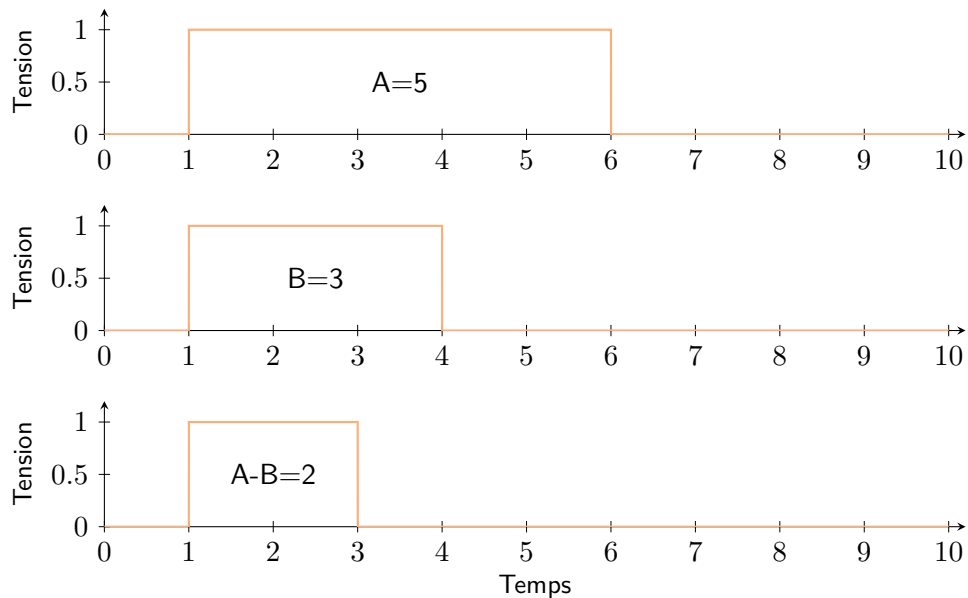


FIGURE I.16 – Principe de soustraction sur les créneaux de G. Roberts.

Et pour finir, multiplier par un nombre entier revient à faire une addition avec le même créneau plusieurs fois de suite.

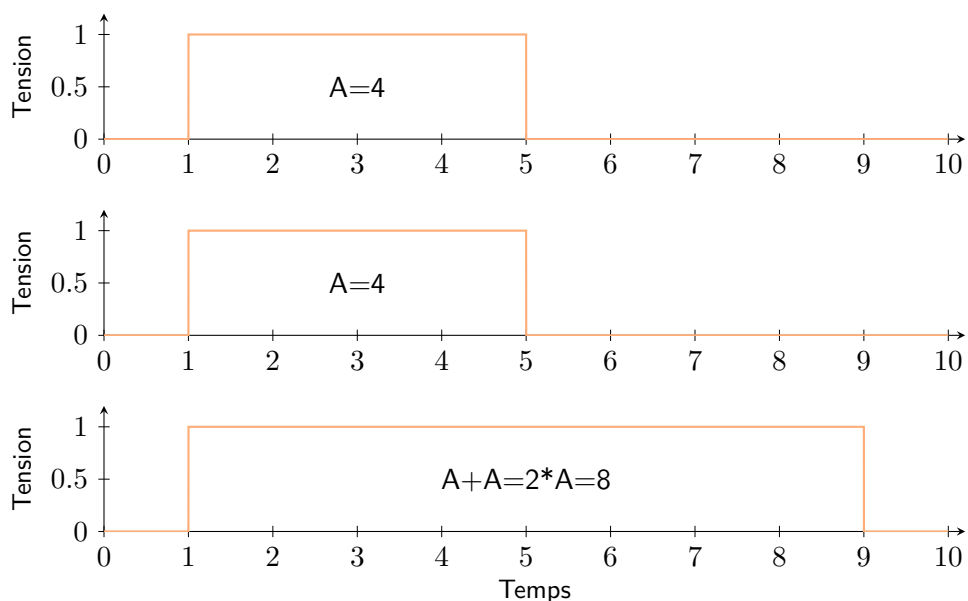


FIGURE I.17 – Principe de multiplication par un nombre entier sur les créneaux de G. Roberts.

La pierre de voûte pour réaliser chacune de ces opérations est de pouvoir manipuler les créneaux à souhait. En effet, pour additionner deux créneaux, il faut pouvoir enregistrer les créneaux puis être en mesure de démarrer le second créneau dès que le premier se termine. Pour la soustraction une simple cellule OU exclusif (XOR) suffit à effectuer l'opération à la condition non triviale de pouvoir décider de l'instant où démarrent les créneaux et de pouvoir les faire commencer au même instant. La réussite de cette méthode repose donc sur la capacité à pouvoir mémoriser un créneau.


C'est donc un dispositif qui permet de réaliser cette fonction qui est présenté dans [Roberts, 2013] sous le nom de « Time latch ». Ce bloc est en réalité un convertisseur qui va transformer la longueur du créneau en une quantité de charges stockée dans un condensateur. Permettant ainsi de retenir la valeur du créneau et de le reproduire au moment voulu en déchargeant cette capacité.

Cette méthode en apparence très attractive puisqu'elle permet effectivement d'effectuer des calculs par le biais de créneaux ne présente malheureusement pas que des aspects positifs. Premièrement, la réalisation du « Time latch » implique un codage du signal dans le domaine des tensions et est donc assujetti aux contraintes de ce domaine. En effet, même si les opérations s'effectuent sur des créneaux

temporels, le fait de stocker ces créneaux sous la forme de charges au sein d'un condensateur ramène sur le devant de la scène les problèmes liés à la représentation dans le domaine de l'amplitude. La capacité aura entre autres une dynamique limitée pour représenter les données et les inconvénients liés à la précision de la représentation de la donnée avec des dynamiques réduites referont leur apparition.

De plus, même en supposant une alternative au stockage sous forme de charge qui resterait uniquement dans le domaine temporel (solution hypothétique dont nous n'avons pas connaissance), il resterait à adresser le problème des opérations plus complexes que sont la multiplication entre créneaux et la division. Ces opérations sont loin d'être triviales et ne semblent pas pouvoir être réalisées uniquement en mémorisant et recopiant des créneaux.

1.2.B Traitement des signaux pulsés

NE autre approche qui connaît un succès grandissant dans la littérature concernant le traitement de l'information dans le domaine temporel est le traitement de signaux impulsionnels. Le but est de représenter l'information sous la forme d'un train d'impulsions dont la densité varie en fonction de l'information. Cela revient implicitement à numériser la donnée mais les calculs peuvent se faire à l'intérieur de processeurs dédiés à cet usage. On peut distinguer deux grands modes de fonctionnements : un mode qui va utiliser un compteur et une horloge pour estimer la densité d'impulsions au cours du temps, et un mode de traitement sans horloge où un filtre passe-bas est utilisé pour générer un signal qui est l'image de la densité d'impulsions.

La figure I.18 présente le schéma d'un circuit du premier mode avec un compteur et une horloge. Cette méthode a le gros avantage de permettre des calculs complexes une fois l'information mise sous la forme de mots numériques de la façon présentée dans le graphe du bas de la figure I.19. Ces mots numériques peuvent alors être utilisés par des unités de traitement numériques avec les capacités qu'on leur connaît ou si l'on veut rester dans le domaine temporel, on peut fournir cette série d'impulsion à un processeur dédié au calcul sur les trains d'impulsions.

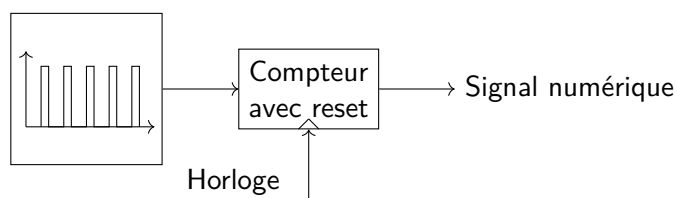


FIGURE I.18 – Exemple de calcul avec des trains de pulses et une horloge.

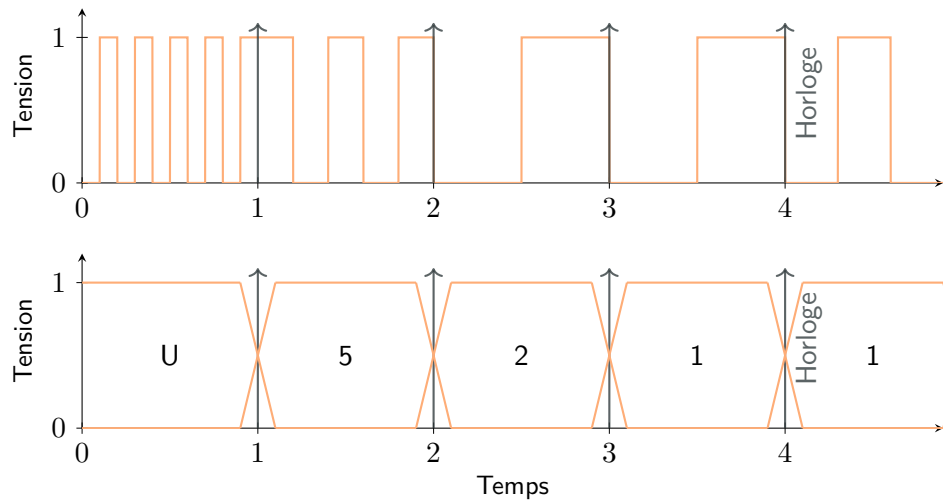


FIGURE I.19 – Exemple de calcul avec des trains de pulses et une horloge.

En contrepartie, les performances de calcul (la résolution et la sensibilité) sont limitées par le rapport entre la largeur d'une impulsion, l'espace minimal entre deux impulsions et la période d'horloge.

A l'opposé, le second type de calcul, présenté en figure I.20 dans lequel la densité d'impulsions est estimée avec un filtre passe-bas ne nécessite pas d'horloge. La fenêtre de temps sur laquelle sera estimée la densité des impulsions étant fixée par la constante de temps du filtre utilisé. La consommation est donc impactée favorablement par l'absence d'horloge.

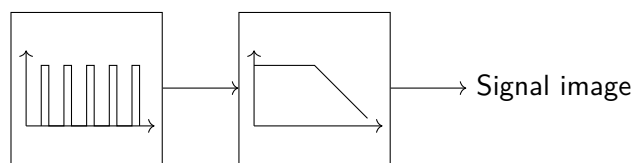


FIGURE I.20 – Exemple de calcul avec des trains de pulses et un filtre.

En revanche, la mise en forme de l'information n'est pas adaptée pour les calculs complexes. La figure I.21 permet de voir les signaux en jeu dans ces calculs et on constate que le signal image de la densité d'impulsions est un signal analogique. On se retrouve de nouveau confronté à tous les problèmes découlant de la représentation dans le domaine de l'amplitude qu'on se donne pour mission ici de contourner.

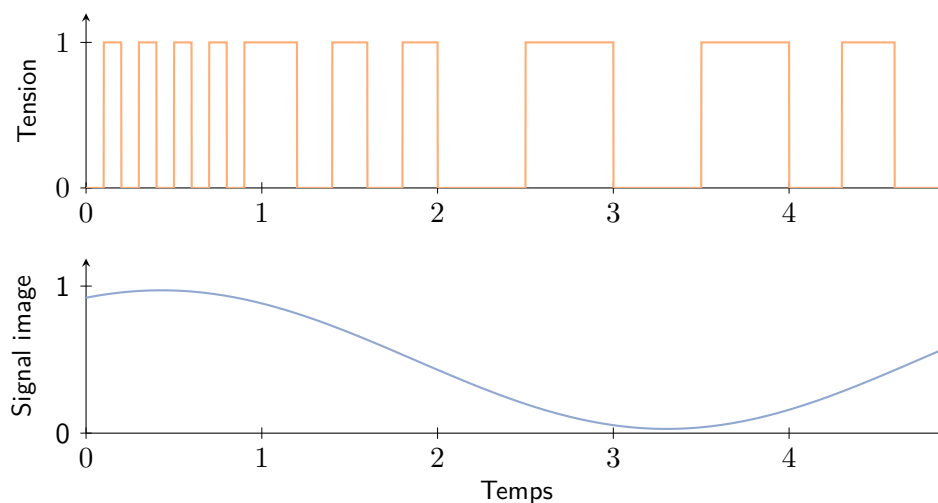


FIGURE 1.21 – Exemple de calcul avec des trains de pulses et un filtre.

1.2.C Bilan

LARMI les deux méthodes proposées, aucune ne permet de calculer uniquement dans le domaine temporel tout en permettant des opérations complexes sur les créneaux. La première méthode de calcul, même si elle offre l'avantage de ne pas avoir à numériser la donnée, ne permet pas de faire des opérations complexes en plus de continuer de se heurter à la problématique de la représentation dans le domaine de l'amplitude.


La méthode de calculs via la densité d'impulsions estimée par un filtre passe-bas n'offre pas de perspective attractive puisqu'elle ne s'affranchit pas de la représentation dans le domaine de l'amplitude. Sa version plus numérique en revanche offre l'avantage de ne pas passer par le domaine de l'amplitude et tire profit des avantages de la conception numérique (grâce au compteur). Les inconvénients liés à l'horloge peuvent être amoindris dans l'optique d'un traitement à faible performances qui par conséquent nécessite une faible fréquence d'horloge.

Il faut cependant garder à l'esprit en parcourant les solutions proposées pour traiter l'information que la solution adoptée pour traiter la donnée dépend très fortement du format de donnée qui est généré par le convertisseur situé en amont de la chaîne. Ainsi si l'on prend un peu de recul pour considérer la chaîne de traitement depuis le signal analogique de départ jusqu'au bloc de traitement, il faut prendre en considération les interactions entre les différentes parties de la chaîne.


On propose d'étudier la faisabilité d'un bloc de calcul dans le domaine temporel dont la conception serait très fortement orientée numérique pour tous les avantages

déjà évoqués et qui s'affranchit complètement du domaine de l'amplitude et de discuter de son impact vis-à-vis des éléments qui l'entourent. Ce bloc de calcul sera l'objet du chapitre II du présent manuscrit.

I.3 Systèmes à performance variable

ANS cette partie, nous allons aborder le sujet des systèmes à performance variable et plus précisément le cas des convertisseurs analogique numérique à performance variable. Comme évoqué plus tôt, la chaîne de traitement dépend beaucoup de ce que le convertisseur analogique numérique va lui fournir. Ainsi, si l'on veut un système à performance variable, il faut commencer par avoir un ADC à performance variable.

I.3.A État de l'art

OMMENÇONS par déterminer ce que signifie l'expression « performance variable » pour un convertisseur analogique numérique. Il y a beaucoup de critères de performance quand on s'intéresse à un ADC. Ici, nous allons principalement nous focaliser sur deux d'entre eux. D'une part la résolution du convertisseur ; c'est-à-dire sa capacité à produire une conversion précise. Pour cela nous nous servons du SNDR comme outil de mesure. D'autre part nous considérerons la bande passante de l'ADC ; à savoir la plage de fréquence dans laquelle un signal d'entrée sera effectivement converti.

Commençons par parcourir ce qui se trouve dans la littérature concernant les convertisseurs dans le domaine de l'amplitude dont on peut faire varier la performance. Démarrons avec le cas du convertisseur flash à résolution variable. Les travaux sur cette architecture, présentés par exemple dans [Varshney *et al.*, 2013] consistent à déterminer l'amplitude du signal d'entrée afin d'adapter la résolution du convertisseur entre 4 et 6 bits. Cette reconfigurabilité permet de n'activer que les comparateurs utiles.

Dans le même esprit, on retrouve les convertisseurs pipelines dont une version reconfigurable en résolution est proposée dans [Kumar *et al.*, 2009] et une version reconfigurable en bande passante est proposée dans [Zhang *et al.*, 2011]. La solution proposée pour adapter la résolution du convertisseur est très semblable à la méthode retenue pour le convertisseur flash ; à savoir une détection de l'amplitude maximale du signal afin de déterminer combien d'étages doivent être activés. Le convertisseur peut ainsi passer d'une résolution de 8 bits à une résolution de 12 bits.

Pour la variabilité en bande passante en revanche, un système d'interrupteurs permet d'activer ou de désactiver les étages du pipeline afin de déterminer la

longueur du pipeline pour la conversion et une machine à états finis permet de régler la fréquence d'horloge sur F_s , $F_s/2$ ou $F_s/4$. La fréquence d'échantillonnage est ainsi réglée ce qui fixe la bande passante du convertisseur.

Dernier exemple de convertisseur à performance variable, le convertisseur à approximations successives (également appelé SAR) dont la résolution peut varier de 6 à 10 bits présenté dans [Zhu *et al.*, 2015]. Ici, ce sont les capacités qui sont désactivées par des interrupteurs pour obtenir des résolutions plus basses.

Pour finir il est important de noter que des architectures beaucoup plus complexes comme ce convertisseur flash rebouclé dans une architecture $\Sigma\Delta$ permettant une grande variabilité de fréquence [Shibata *et al.*, 2012] ont été proposées. Mais aussi performantes que soient les solutions proposées, elles ne permettent qu'une dynamique de reconfigurabilité assez faible et elles ne permettent pas réellement d'opérer en deux modes de fonctionnement distincts. Par dessus tout, elles reposent sur une représentation dans le domaine de l'amplitude qui n'est pas souhaitable au regard du point de vue adopté dans le travail de ce manuscrit.

Attardons-nous donc aux différents mécanismes et aux différents leviers sur lesquels il est possible d'intervenir pour rendre reconfigurable un convertisseur qui opère dans le domaine temporel.

1.3.B Étude de la dynamique de reconfigurabilité des ADC dans le domaine temporel



N s'applique ici à regarder les paramètres qui influent sur les performances des convertisseurs opérant dans le domaine temporel ainsi que les conséquences d'une augmentation de résolution ou de bande passante. Pour cela on va comparer pour chaque type de convertisseur son fonctionnement dans une situation dite « nominale » et son fonctionnement si on veut doubler sa bande passante ou sa résolution.

Commençons par le cas du convertisseur analogique numérique à franchissement de seuils. Nous définissons la situation représentée en figure 1.22 comme étant la situation en fonctionnement nominal. Ce convertisseur va pouvoir générer des événements rapprochés d'au minimum Δt_{min0} ; c'est ce qui va définir sa bande passante. Ce temps minimal correspond à la capacité des comparateurs à détecter le franchissement d'un seuil.

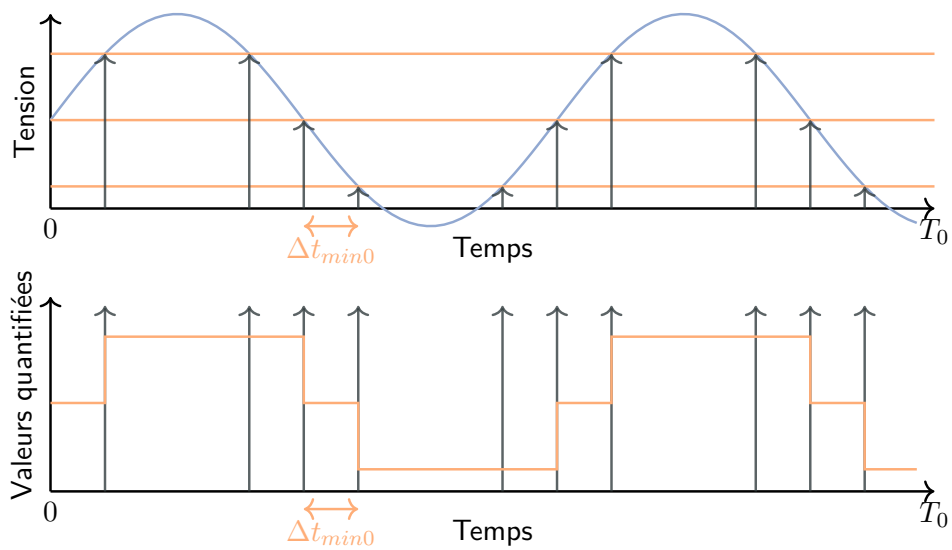


FIGURE 1.22 – Fonctionnement d'un ADC à franchissement de seuils dans une situation donnée.

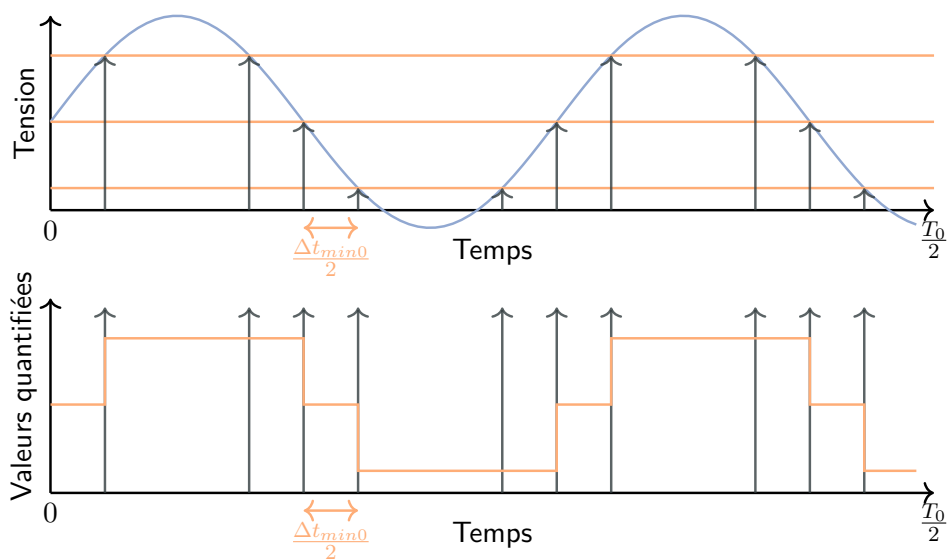


FIGURE 1.23 – Fonctionnement d'un ADC à franchissement de seuils avec une bande passante deux fois plus grande.

Pour doubler la bande passante de ce convertisseur, toutes choses égales par ailleurs, il faut se retrouver dans la situation décrite en figure 1.23. On notera que l'axe des abscisses représente une durée deux fois plus courte. Ainsi, il faut pouvoir générer des évènements deux fois plus rapidement donc avoir un temps de traitement minimal (Δt_{min0}) deux fois plus petit que celui nécessaire pour le

fonctionnement nominal donc des comparateurs deux fois plus rapides. Si l'on veut pouvoir passer d'une bande passante à l'autre, il faut donc prévoir le Δt_{min0} nécessaire à la bande passante la plus élevée puis ensuite décider de générer des signaux plus lentement.

Procédons de la même façon mais en se focalisant cette fois-ci sur la résolution du convertisseur. Là encore, reprenons une situation nominale donnée par la figure I.24.

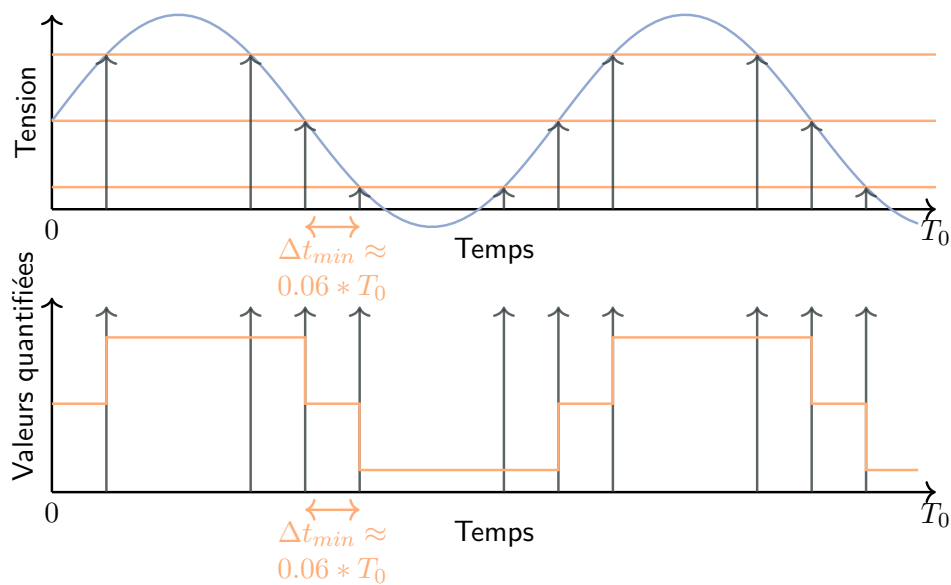


FIGURE I.24 – Fonctionnement d'un ADC à franchissement de seuils dans une situation donnée.

Pour une résolution deux fois plus fine, il faut se retrouver dans la situation décrite en figure I.25. On a donc dans cette situation des seuils deux fois plus proches que dans la situation nominale. Cela entraîne donc deux fois plus de seuils de comparaison et pour un même signal des événements seront donc en moyenne générés deux fois plus souvent. On a donc à la fois besoin de plus de matériel (plus de comparateurs) mais également besoin d'un Δt minimal deux fois plus petit ainsi que de traiter deux fois plus de données. Pour pouvoir changer dynamiquement de résolution, il faut donc avoir sur le circuit le nombre maximal de comparateurs et ensuite décider si oui ou non ils doivent tous être utilisés.

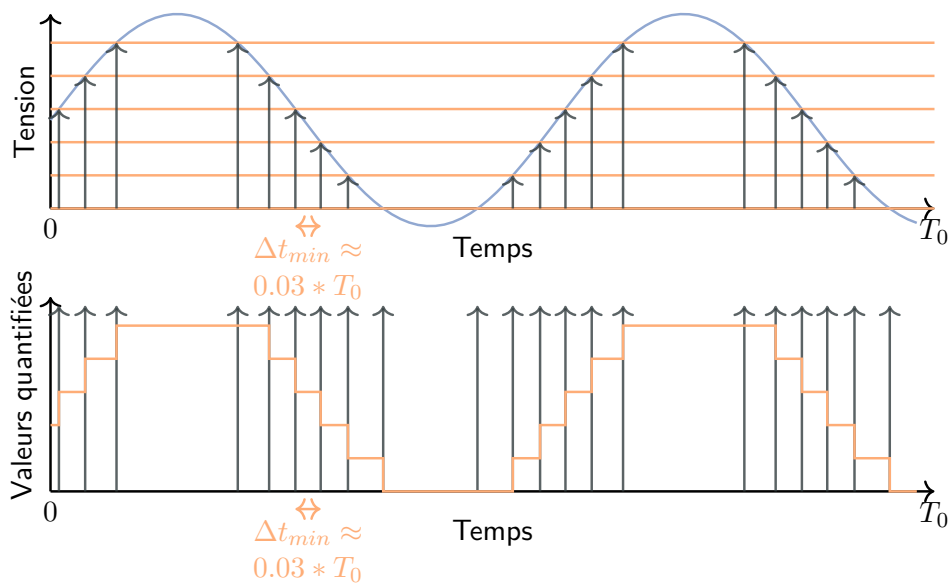


FIGURE 1.25 – Fonctionnement d'un ADC à franchissement de seuils avec une résolution deux fois plus fine.

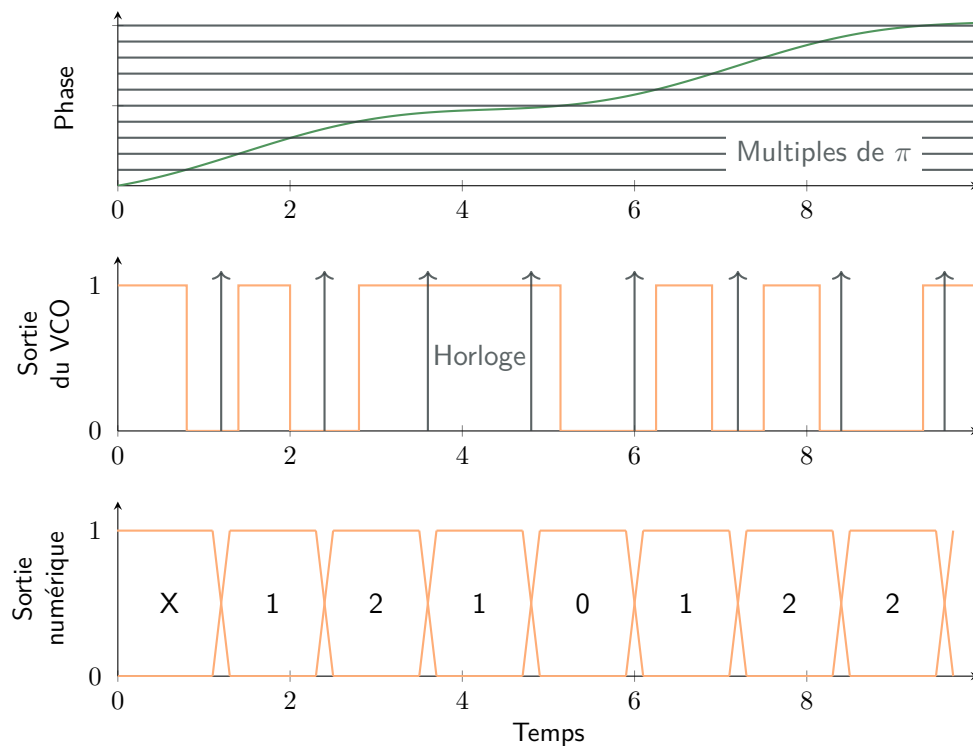


FIGURE 1.26 – Fonctionnement d'un VCO-based ADC dans une situation donnée.

Regardons maintenant ce qu'il en est en termes de variabilité en ce qui concerne les convertisseurs analogique numérique basés sur un VCO. Là encore, procédons de la même manière. Considérons une situation nominale telle que présentée en figure I.26. Il est à noter qu'il a été choisi de représenter la phase afin d'avoir une meilleure idée de ce qu'il se passe au sein du VCO.

Pour doubler la bande passante de ce convertisseur, il faut se retrouver dans la même situation que la figure I.27. On notera que l'axe des abscisses représente une durée deux fois plus petite que dans la figure précédente. On constate que pour arriver à cette situation, il faut non seulement doubler la fréquence d'horloge, mais il faut en plus doubler les caractéristiques du VCO ; à savoir son gain et sa fréquence centrale, pour que la phase augmente deux fois plus rapidement. Pour pouvoir alterner entre les deux bandes passantes il faut pouvoir modifier la fréquence d'horloge à la volée (ce qui est relativement simple) mais également pouvoir changer les caractéristiques du VCO.

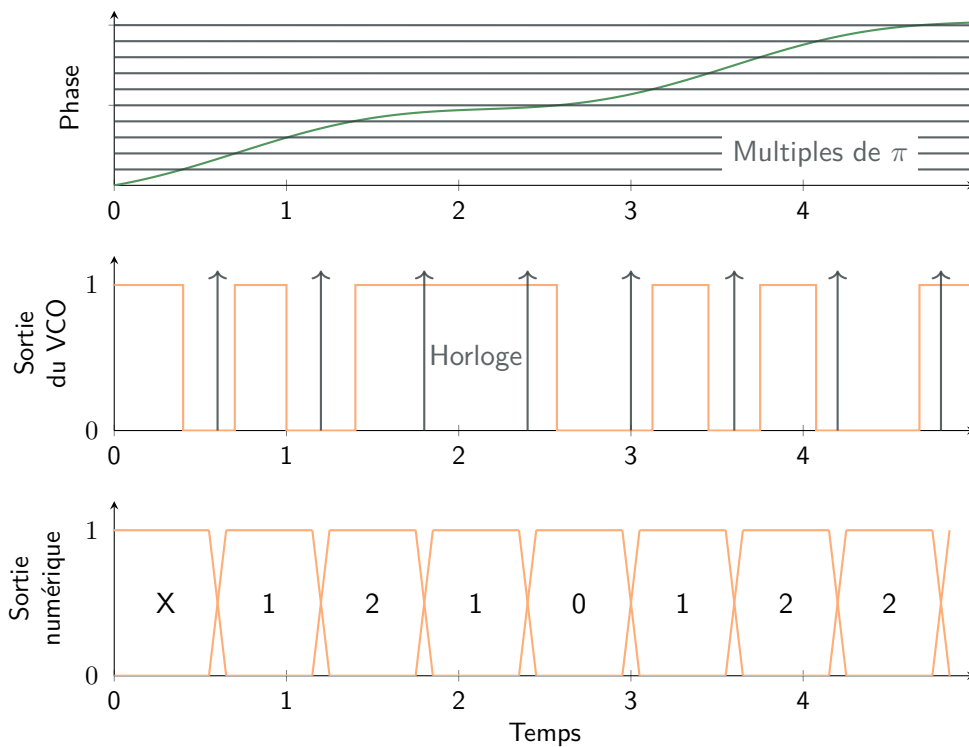


FIGURE I.27 – Fonctionnement d'un VCO-based ADC avec une bande passante deux fois plus grande.

Finissons ce petit exercice en se penchant maintenant sur la question de la résolution du convertisseur. Là encore, la même procédure est employée. On repart

d'une situation nominale figure I.28.

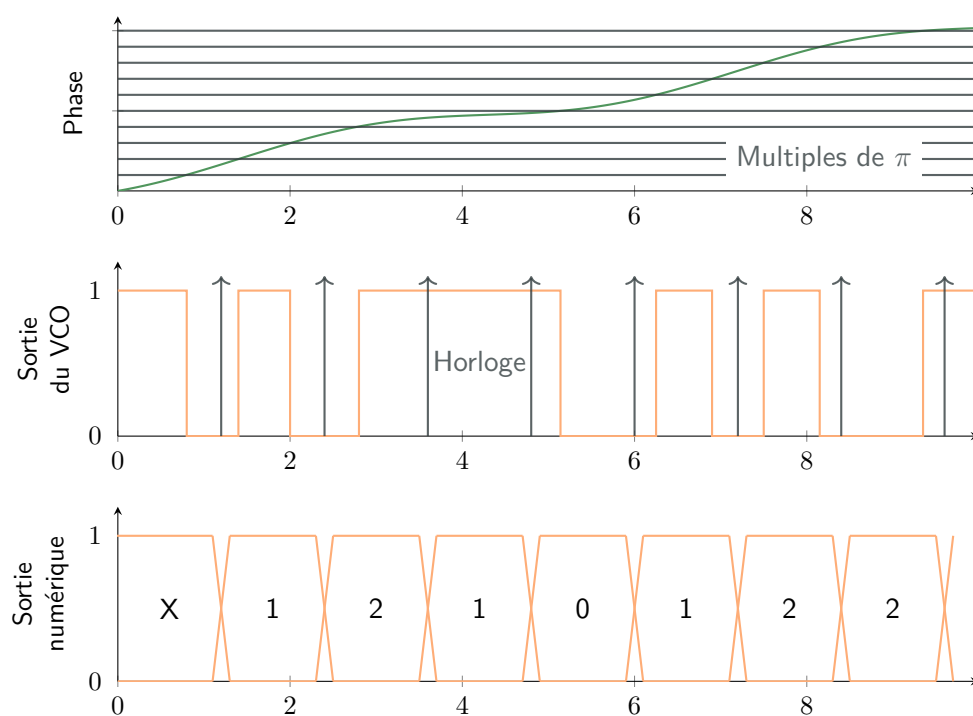


FIGURE I.28 – Fonctionnement d'un VCO-based ADC dans une situation donnée.

On cherche à se placer dans la situation de la figure I.29 pour obtenir une résolution de conversion deux fois plus fine. On constate que pour y arriver il faut doubler les caractéristiques du VCO de sorte que la phase croît deux fois plus rapidement et que deux fois plus de fronts se produisent sur la sortie du VCO. Il faut donc un compteur capable de compter deux fois plus et deux fois plus vite.

Pour faire varier les performances des convertisseurs qui opèrent dans le domaine temporel, on a mis en évidence ici des leviers qui permettaient, plus ou moins facilement, d'amener de la reconfigurabilité.

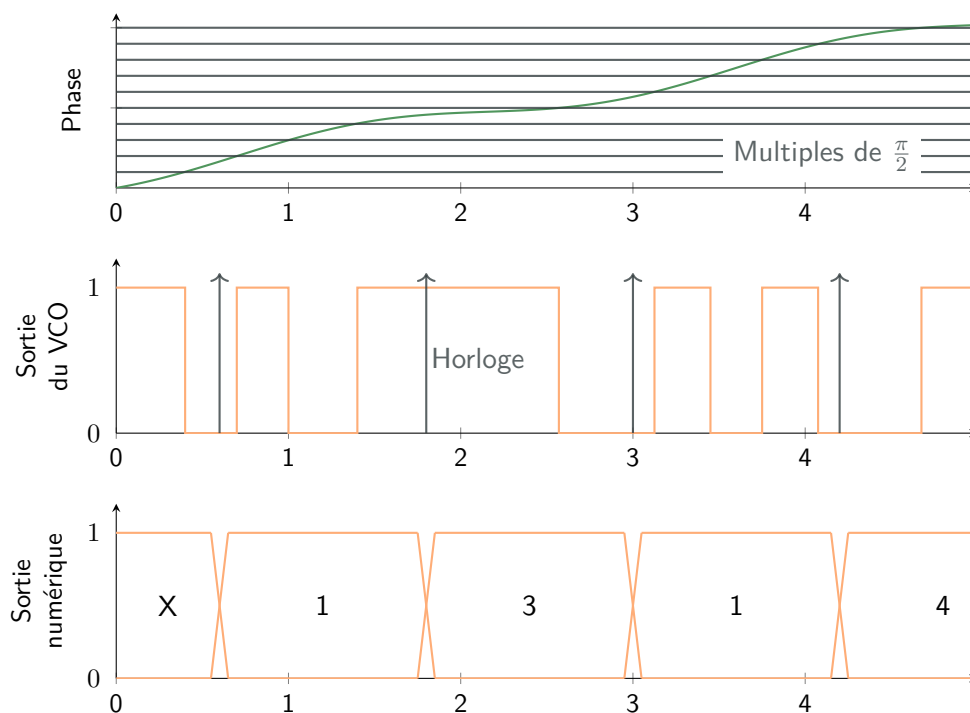


FIGURE 1.29 – Fonctionnement d'un VCO-based ADC avec une résolution deux fois plus fine.

Cependant il est bien question ici de reconfigurabilité donc d'une modification de faible dynamique des paramètres du convertisseur et pas un changement dans le mode de fonctionnement de celui-ci.

1.3.C Bilan



A partie précédente nous a permis de constater qu'il n'existait pas réellement de convertisseur fonctionnant selon deux modes, dont l'un serait orienté vers la performance et l'autre, dégradé, serait optimisé pour la faible consommation. Bien que les convertisseurs opérant dans le domaine temporel présentent plusieurs leviers qui permettent de jouer sur les deux performances que nous nous sommes fixées ; à savoir la résolution et la bande passante, ils ne présentent pas non plus de fonctionnement selon deux modes distincts.

La nature ambivalente des VCO-based ADCs (à la fois une partie échantillonnée par une horloge et à la fois une partie franchissement de seuils) devrait permettre de fonctionner dans deux modes différents. Nous allons donc étudier par la suite

le potentiel de cette architecture.

I.4 Bilan de l'état de l'art



La problématique étudiée a permis de définir trois axes de recherche :

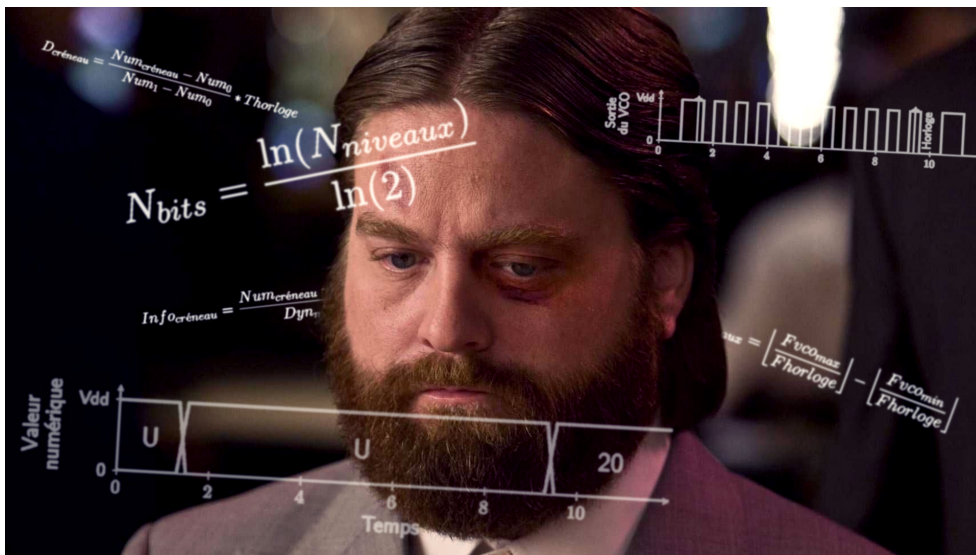
- Un premier axe sur le calcul dans le domaine temporel indépendant de la représentation dans le domaine de l'amplitude ;
- Un deuxième axe sur le développement d'un convertisseur opérant dans le domaine temporel pouvant fonctionner selon deux modes différents ;
- Et enfin un axe sur la représentation dans le domaine temporel et la volonté d'améliorer le ratio résolution/bande passante des convertisseurs basés sur un VCO.

L'étude de l'état de l'art a permis d'orienter ces axes de recherches en pointant du doigt les pistes d'amélioration disponibles. Nous proposons donc de pousser les axes de recherches précédents respectivement dans les directions suivantes :

- Proposer une méthode de calcul par le biais de créneaux grâce à l'utilisation d'un oscillateur contrôlé en tension. Ceci sera développé dans le chapitre II de ce manuscrit ;
- Proposer un convertisseur qui tire profit de la nature ambivalente des VCO afin de pouvoir fonctionner dans deux modes différents. Ce sujet sera abordé dans le chapitre III de ce document ;
- Et pour finir, en se focalisant sur le mode performant du convertisseur proposé au chapitre III, étudier la faisabilité de l'augmentation des performances de ces structures. Ceci fera l'objet du chapitre IV de ce travail.

Chapitre II

Méthode de traitement dans le domaine tempore : calcul par le biais de créneaux



« Calculs avec les créneaux. » Crédits : Jordan Rous

II.1 Présentation de la méthode



DANS cette partie, nous allons nous focaliser sur le développement de la théorie d'une méthode de calcul sur des signaux créneaux. Cette méthode de calcul permettra ainsi de prendre en entrée une information temporelle (la largeur de signaux en créneaux) afin de pouvoir calculer avec cette information.

L'objectif de cette méthode est de proposer une solution (ou a minima un début de solution) qui permet d'effectuer des calculs complexes sur des données représentées par des signaux en créneaux et qui opère dans le domaine temporel, toujours au regard de la problématique de réduction des tensions d'alimentation.

Comme il a été vu dans la partie I.2, la méthode de traitement proposée par G. Roberts offre l'avantage de ne pas numériser les signaux mais elle reste tout de même soumise aux contraintes de la représentation dans le domaine de l'amplitude tout en limitant la complexité des calculs possibles.

En reprenant, en outre, les éléments de conclusion de la partie I.2.C on peut poser les piliers de départ de notre réflexion ; à savoir :

- la solution de numériser l'information offre la possibilité d'effectuer des calculs complexes grâce à la puissance des blocs de traitement numériques ;
- l'utilisation d'une architecture fortement orientée numérique permet de tirer profit des nouveaux nœuds technologiques ;
- le fonctionnement avant le bloc de traitement numérique doit opérer dans le domaine temporel.

Le premier des points ci-dessus va nous mener vers un bloc de traitement numérique. N'ayant pas la prétention de pouvoir faire mieux qu'un expert dans le domaine de la conception de ces blocs (DSP) dans le temps alloué à ce travail, nous allons donc nous orienter vers la conception d'un convertisseur qui transformera la donnée temporelle contenue dans la largeur des créneaux en mots numériques compréhensibles par l'unité de traitement numérique. L'enjeu étant que ce convertisseur opère dans le domaine temporel.

Il est proposé ici de coder l'information sur la durée des créneaux comme il est fait dans la méthode de calcul présentée en partie I.2.A mais d'utiliser une conversion des créneaux qui ne repose pas sur le domaine de l'amplitude et qui permet de numériser les signaux ; donc de pouvoir faire des calculs complexes avec par la suite.

Ce qui est proposé est d'utiliser un convertisseur analogique numérique basé sur un oscillateur contrôlé en tension comme celui présenté dans la figure I.5 sur les signaux créneaux. Ce chapitre consistera à étudier le comportement de ce convertisseur ainsi qu'à dériver ses propriétés élémentaires.

L'idée sous-jacente de cette solution est la suivante. Sur une période d'horloge, l'oscillateur contrôlé en tension oscillera à sa fréquence maximale pendant la durée où le créneau vaut « 1 » et à sa fréquence minimale quand le créneau vaut « 0 » permettant ainsi de retrouver la durée du créneau.

Le système utilisé pour l'opération est schématisé dans la figure II.1 dans lequel on peut voir les différents éléments en jeu. Le signal créneau d'entrée est envoyé au VCO qui va produire un signal créneau oscillant dont la fréquence variera en fonction de la valeur du créneau (0 ou 1) et le compteur va venir compter le nombre de fronts (montants et descendants) de la sortie du VCO pendant une période d'horloge.

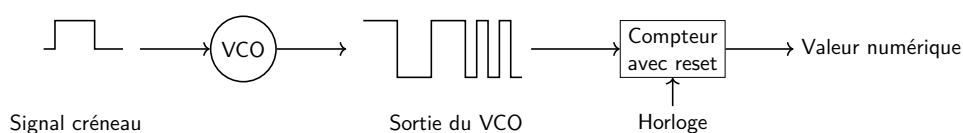


FIGURE II.1 – Schéma de la méthode de numérisation des créneaux.

Dans la méthode avancée, les créneaux présentés en entrée du convertisseur ont donc une durée bornée par la période de l'horloge. Pour simplifier, un créneau de durée nulle vaudra 0 tandis qu'un créneau de même durée que la période d'horloge vaudra 1. Ainsi, toutes les données seront comprises entre 0 et 1.

À l'aide des figures II.2 et II.3, il va être plus facile de se représenter le déroulement de la conversion. Pour cela, commençons par regarder le résultat de la conversion d'un créneau de durée nulle qui représente donc la valeur 0 pour le calcul ainsi que le résultat de la conversion du signal représentant la valeur 1 ; c'est-à-dire un signal dont la durée vaut une période d'horloge.

On notera que dans l'exemple de l'illustration, le rapport entre la fréquence d'échantillonnage et la plage de fréquence de fonctionnement de l'oscillateur a volontairement été choisi de la sorte pour faciliter la représentation.

Le fonctionnement de la numérisation de créneaux est assez simple. Un signal de durée nulle, donc codant pour la valeur 0, aura pour valeur numérisée 10 (avec les valeurs de l'exemple) alors qu'un signal de durée maximale ; c'est-à-dire d'une période d'horloge (donc qui code pour la valeur 1 au sens du calcul) aura une valeur numérisée de 20. Donc un signal qui vaudrait une fois numérisé 13 par exemple, serait en fait un signal d'une durée de 0.3 période d'horloge (donc qui code pour la valeur 0.3).

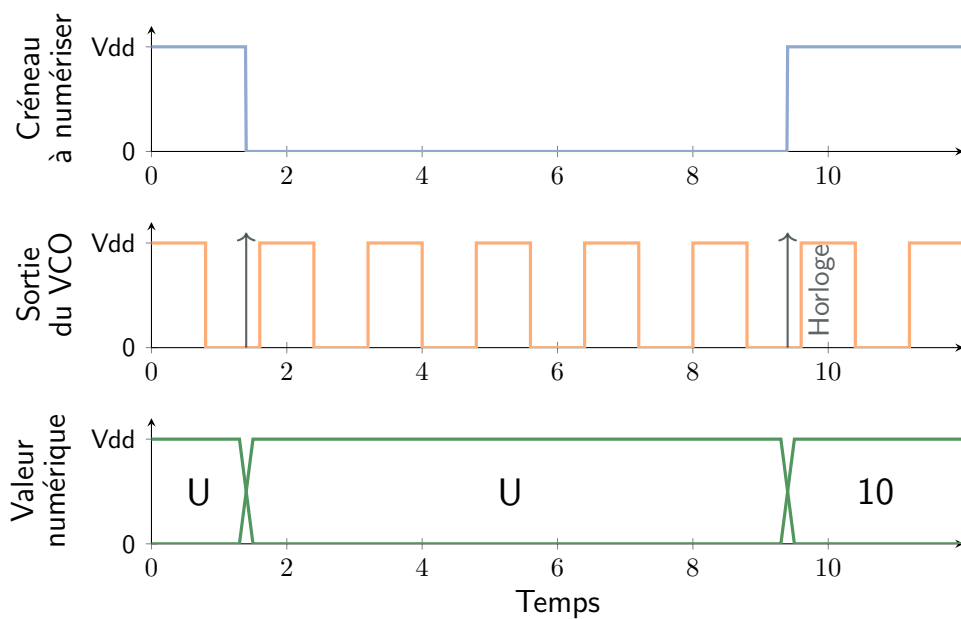


FIGURE II.2 – Exemple de conversion d'un signal codant pour « 0 ».

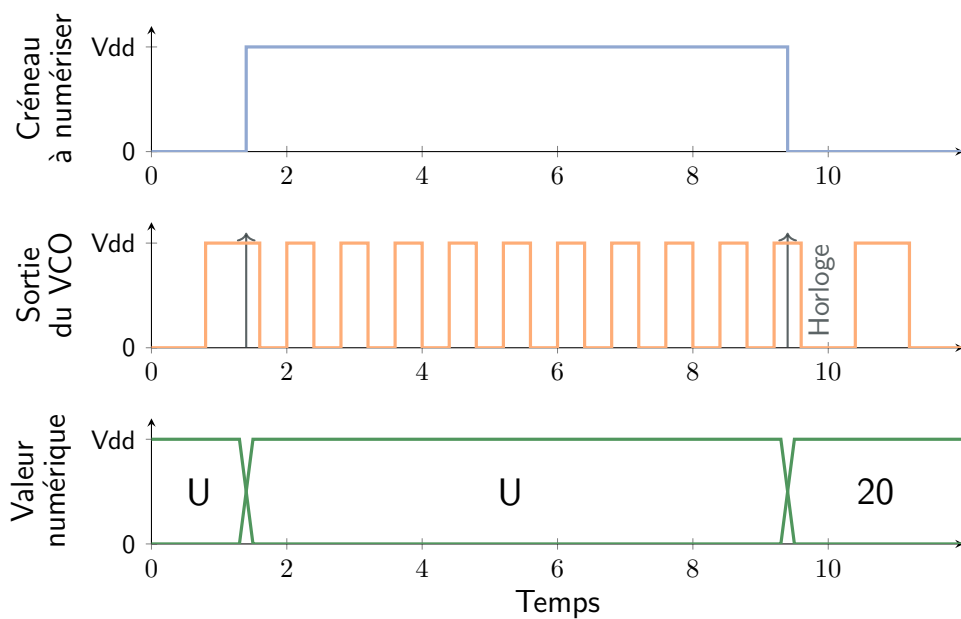


FIGURE II.3 – Exemple de conversion d'un signal codant pour « 1 ».

De manière un peu plus formelle, nous allons préciser le fonctionnement de ce convertisseur en déroulant les équations qui régissent son fonctionnement. Dans un premier temps, nous considérerons le convertisseur sans aucune non-idéalités afin

de le simplifier puis nous amènerons au fur et à mesure les imperfections présentes dans ce type de convertisseurs afin de dessiner les limites d'une telle application.

Commençons par exprimer la valeur numérique qui code pour la valeur 0 ; c'est-à-dire la valeur numérique correspondant à un signal créneau de durée nulle.

Ainsi :

$$Num_0 = \left\lfloor \frac{2 \times Fvco_{min}}{Fhorloge} \right\rfloor \quad (II.1)$$

avec :

Num_0 la valeur numérique codant pour un signal de durée nulle ;
 $Fvco_{min}$ la fréquence minimale du VCO ;
 $Fhorloge$ la fréquence de l'horloge.

La présence du 2 au numérateur vient du fait que dans l'exemple, à la fois les fronts montants et les fronts descendants sont comptabilisés.

De même, la valeur numérique correspondant à un signal créneau dont la durée serait la même que la période d'horloge est régit par l'équation suivante :

$$Num_1 = \left\lfloor \frac{2 \times Fvco_{max}}{Fhorloge} \right\rfloor \quad (II.2)$$

avec :

Num_1 la valeur numérique codant pour un signal de la même durée que la période d'horloge ;
 $Fvco_{max}$ la fréquence maximale du VCO.

Une fois que l'on connaît la valeur numérique minimale Num_0 que peut produire le convertisseur ainsi que la valeur numérique maximale Num_1 qu'il peut produire, il est possible, à partir de la valeur numérique produite par le convertisseur, de retrouver la durée initiale du créneau présenté en entrée.

Pour ce faire, il est nécessaire de comprendre comment fonctionne le codage de ce convertisseur. Reprenons l'exemple avec les valeurs données par les figures II.2 et II.3 donc avec $Num_0 = 10$ et $Num_1 = 20$. Dans cet exemple, on dispose donc d'une dynamique sur la représentation numérique de 10 (20-10) pour représenter un signal allant de 0 à 1. On pose donc :

$$Dyn_{num} = Num_1 - Num_0 \quad (II.3)$$

avec :

Dyn_{num} la dynamique servant à représenter les signaux de manière numérique.

En suivant une règle de proportionnalité, un signal qui, toujours avec les valeurs choisies dans l'exemple utilisé, aurait une valeur numérique de 13, représenterait donc un signal qui vaut 3 de plus que la valeur minimale et, vis-à-vis de la dynamique disponible pour coder l'information, un signal de valeur 0.3 ($\frac{3}{10}$). Plus généralement, on peut poser :

$$Info_{créneau} = \frac{Num_{créneau} - Num_0}{Dyn_{num}} \quad (II.4)$$

avec :

$Info_{créneau}$ l'information contenue dans la durée du créneau, comprise entre 0 et 1 ;

$Num_{créneau}$ la valeur numérisée retournée par le compteur.

Il convient alors pour retrouver la durée réelle du créneau présenté à l'entrée du convertisseur de rappeler que la représentation (entre 0 et 1) est définie en fonction de la période d'horloge et donc qu'un signal qui code pour 0.3 (i.e. $Info_{créneau} = 0.3$) est un signal d'une durée de 0.3 période d'horloge.

On peut ainsi définir la valeur d'un créneau de n'importe quelle durée en fonction de la valeur retournée par le compteur et de la période d'horloge. Avec la remarque du paragraphe précédent et en remplaçant dans l'équation (II.4) Dyn_{num} par son expression dans l'équation (II.3), on obtient qu'un signal créneau, dont la valeur numérisée est $Num_{créneau}$, est un signal dont la durée $D_{créneau}$ est telle que :

$$D_{créneau} = \frac{Num_{créneau} - Num_0}{Num_1 - Num_0} * Thorloge \quad (II.5)$$

avec :

$Thorloge$ la période de l'horloge ;

$D_{créneau}$ la durée du créneau.

II.2 Intérêt de la méthode



DANS cette partie, les différents avantages de la méthode vont être mis en avant afin de justifier de son intérêt. Il va s'agir simplement de passer en revue ces différents points d'intérêt.

Le principal intérêt de la méthode proposée est de s'affranchir complètement du domaine de l'amplitude pour réaliser les calculs. C'était un des objectifs de ce chapitre. Avec la solution proposée, plus question de charges stockées (comme c'est le cas dans les travaux de G. Roberts) ou d'amplitudes pour manipuler les données (comme on le trouve dans les convertisseurs analogique numérique classiques) mais uniquement des créneaux d'amplitude fixe valant soit « 0 » soit « Vdd » et des mots numériques.

Le fait de procéder à une numérisation des données permet de surcroît de pouvoir effectuer des calculs complexes. Bien que l'intégralité du processus de traitement ne se déroule pas dans le domaine temporel, grâce à la numérisation de l'information (donc sa transformation en mots numériques) il est ainsi possible de faire toutes les opérations que l'on peut réaliser avec une unité de traitement numérique ce qui inclut (de manière non exhaustive) les additions, soustractions, multiplications et même des opérations plus complexes telles que les divisions.

Ces calculs peuvent être réalisés par le biais d'unités de calculs numériques classiques puisque le résultat de la conversion est sous la forme d'un mot numérique synchronisé sur l'horloge. Cela implique que n'importe quelle architecture d'unité de traitement numérique existante dans la littérature peut être utilisée avec le convertisseur présenté ici. Ce qui contribue à justifier le fait qu'on n'aborde pas le bloc de traitement ici.

Le problème en cas de signaux créneaux non synchronisés évoqué dans la partie I.2 ne se pose plus puisque le convertisseur analogique numérique basé sur un oscillateur contrôlé en tension utilise une horloge et que par conséquent, le début des créneaux est généré sur les fronts de cette horloge comme illustré sur la figure II.3. Bien que plus facile à concevoir, le système ne sera cependant pas apte à adapter son activité en fonction de l'activité du signal d'entrée.

Pour finir, le fait de n'utiliser l'oscillateur contrôlé en tension qu'avec des signaux en créneaux ne pouvant prendre que deux valeurs distinctes (soit 0 soit Vdd) faisant ainsi fonctionner le VCO soit à sa fréquence minimale, soit à sa fréquence maximale, élimine de manière intrinsèque la non-linéarité de la conversion tension/fréquence réalisée par l'oscillateur contrôlé en tension. En ne fonctionnant qu'en deux points, la fonction de transfert qui en résulte est forcément linéaire rendant ainsi la conversion totalement linéaire.

II.3 Caractérisation théorique de la méthode



MAINTENANT que la méthode a été présentée et que ses avantages ont été mis en avant, nous allons dans cette partie caractériser la méthode ; à

savoir déterminer la résolution de la conversion réalisée ainsi que son pas de quantification.

Par le terme résolution, il est désigné le nombre de bits sur lesquels la conversion est réalisée. Le terme de pas de quantification quant à lui représente la valeur minimale au-delà de laquelle deux données seront discriminées.

Comme annoncé dans la partie II.1, nous considérerons dans un premier temps le convertisseur comme idéal afin de dériver les équations qui le caractérisent. Nous aborderons les non-idéalités de l'architecture dans un second temps.

Pour cela, commençons par nous intéresser à la valeur minimale détectable pour la numérisation. En d'autres termes, la durée minimale du créneau qui fera passer la sortie de la conversion de Num_0 à $Num_0 + 1$.

En se servant de l'équation (II.5) on peut poser :

$$D_{créneau\min} = \frac{Num_0 + 1 - Num_0}{Num_1 - Num_0} * Thorloge \quad (II.6)$$

avec :

$D_{créneau\min}$ la durée minimale d'un créneau pour qu'il soit détecté par la conversion.

Au final, on se retrouve avec l'équation suivante :

$$D_{créneau\min} = \frac{1}{Num_1 - Num_0} * Thorloge \quad (II.7)$$

Si l'on remplace dans l'équation (II.7) les valeurs utilisées dans l'exemple de la figure II.3, on obtient que la valeur minimale détectable dans cet exemple est un créneau de durée $0.1 * Thorloge$. Cela correspond ainsi au pas de quantification de l'exemple puisqu'il faut deux créneaux dont les durées sont au moins différentes de $0.1 * Thorloge$ (en l'occurrence un créneau de durée 0 et un créneau de durée $0.1 * Thorloge$) pour faire changer la valeur numérisée en sortie du convertisseur.

Considérons maintenant la dynamique de la conversion. Il a été montré que la durée des créneaux utilisés par cette méthode de calcul peut varier entre une durée nulle (codant pour un « 0 ») et une durée d'une période d'horloge entière (codant pour « 1 »).

On peut donc définir la dynamique de la conversion comme étant :

$$Dyn = D_{max} - D_{min} \quad (II.8)$$

avec :

D_{max} la durée maximale qu'un créneau peut avoir ;

D_{min} la durée minimale qu'un créneau peut avoir ;

Dyn la dynamique de la conversion.

Ce qui, en remplaçant par les valeurs de chaque grandeurs nous permet de déduire :

$$Dyn = Thorloge - 0 = Thorloge \quad (II.9)$$

Or, en se référant à ce qui a été démontré précédemment, on a :

$$q_{step} = D_{créneau}min \quad (II.10)$$

avec :

q_{step} le pas de quantification de la conversion.

En se servant des équations précédentes, on peut alors déterminer le nombre de niveaux sur lesquels peut être codée le résultat de la conversion qui est défini comme étant :

$$N_{niveaux} = \frac{Dyn}{q_{step}} \quad (II.11)$$

avec :

$N_{niveaux}$ le nombre de niveaux disponibles pour la conversion.

Ce qui donne si l'on remplace les grandeurs par leur expressions dans (II.11) :

$$N_{niveaux} = \frac{Thorloge}{\frac{Thorloge}{Num_1 - Num_0}} \quad (II.12)$$

L'équation (II.12) peut être réécrite et se simplifie ainsi en :

$$N_{niveaux} = Num_1 - Num_0 \quad (II.13)$$

On peut alors exprimer la résolution du convertisseur en nombre de bits en utilisant la formule suivante :

$$N_{niveaux} = 2^{N_{bits}} \quad (II.14)$$

avec :

N_{bits} la résolution de la conversion.

Ce qui permet d'exprimer la résolution de la conversion comme le rapport de logarithmes suivant :

$$N_{bits} = \frac{\ln(N_{niveaux})}{\ln(2)} \quad (II.15)$$

Il est préférable d'exprimer le nombre de bits de résolution de la conversion en fonction des paramètres du convertisseur sur lesquels il est possible d'avoir un impact.

Ainsi, en reprenant les expressions établies par les équations (II.1) et (II.2), l'équation (II.13) devient :

$$N_{niveaux} = \left\lfloor \frac{F_{vco_{max}}}{F_{horloge}} \right\rfloor - \left\lfloor \frac{F_{vco_{min}}}{F_{horloge}} \right\rfloor \quad (II.16)$$

En reprenant les valeurs de l'exemple servant à illustrer ce chapitre ; à savoir avec $N_{niveaux} = 10$, on obtient une résolution d'environ 3 bits. Si l'on considère une fréquence d'horloge de 20 MHz, cela donne un oscillateur contrôlé en tension qui doit avoir une plage de fréquence allant de 100 MHz à 200 MHz.

Grâce aux équations développées dans cette partie, la méthode de conversion dans le but de calculer par le biais de créneaux a été caractérisée. Mais ces équations ne sont vraies que si la méthode avancée ici est idéale (c'est-à-dire si elle ne présente aucune imperfection).

Analysons maintenant un peu plus en détails les facettes moins idéales de la méthode proposée.

II.4 Limitations



BIEN qu'intéressante, la méthode proposée n'est pas exempte de non-idéalité et possède quelques limitations que nous allons tâcher d'aborder dans cette partie.

Évoquons d'abord une contrainte liée à la construction même de la méthode de conversion utilisée pour permettre le calcul : la nécessité d'une référence de temps. En effet, puisqu'un convertisseur analogique numérique basé sur un oscillateur contrôlé en tension est utilisé pour numériser le signal créneau afin de permettre d'effectuer des calculs complexes avec, il est nécessaire d'avoir une horloge dans le système. La présence de cette horloge apporte indubitablement une consommation non négligeable.

Plus contraignant cette fois, la mise en forme de l'information utilisée est un point noir pour cette méthode de calcul. Il a été implicitement considéré

que les données arrivaient à l'entrée du convertisseur sous la forme de signaux créneaux synchronisés sur le signal d'horloge. Seulement, imposer une telle mise en forme des données implique que la méthode ne peut pas être utilisée telle quelle. Il faut que toute la chaîne de transmission fonctionne avec cet encodage, ce protocole.

Il faut donc, soit utiliser un convertisseur en amont de ce dispositif qui permet de mettre la donnée brute initiale sous la forme désirée pour être traitée par la méthode proposée. Auquel cas le système perd énormément d'intérêt puisqu'il nécessite d'introduire deux convertisseurs successifs dans la chaîne de traitement. Soit que le protocole initial de transfert de donnée soit déjà compatible avec la méthode de numérisation proposée ici. Ainsi l'émetteur à l'origine du signal émet la donnée sous la forme désirée pour le système de traitement ce qui est assez contraignant.

Pour finir, en regardant d'un peu plus près le fonctionnement de la numérisation des créneaux, il apparaît que la méthode énoncée possède un défaut intrinsèque. Au moment de la transition entre deux conversions, le VCO ne se remet pas à zéro. La figure II.4 illustre bien ce comportement.

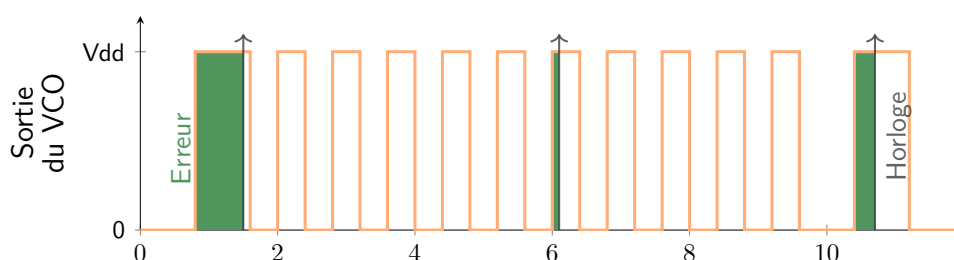


FIGURE II.4 – Erreur systématique introduite par la méthode de conversion.

La précision de la conversion n'est alors plus celle qui est théoriquement atteignable et qui a été établie dans la partie II.3. À la place, chaque conversion va contenir une erreur qui va se cumuler au fil du temps et qui dépend donc des conversions précédentes. Cette erreur (l'erreur de quantification d'un VCO-based ADC), ou plutôt une proposition pour s'en affranchir fera l'objet du travail présenté dans la partie IV.

Pour estimer l'impact de l'erreur ainsi mise en évidence, un modèle de simulation MATLAB™, disponible en Annexe V.4 page 102, permet de comparer avec les valeurs de l'exemple choisi au début de ce chapitre les conversions avec et sans non-idéalités. Comme l'erreur mise en évidence dans la figure II.4 dépend du signal d'entrée, la figure II.5 permet de mettre en évidence l'effet de l'erreur pour des signaux d'entrée à différentes fréquences en calculant le SNDR de la conversion (avec les paramètres de l'exemple du début du chapitre) sur une bande passante

de 9 MHz.

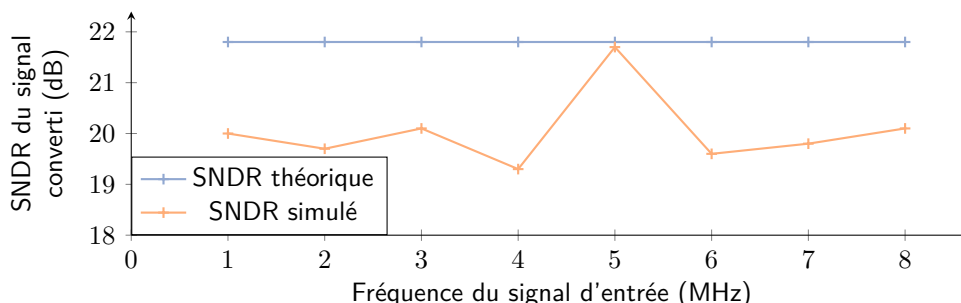



FIGURE II.5 – Comparaison entre les conversions avec et sans non-idéalités.

Une dernière limitation touche à la difficulté de pouvoir traiter des données de tailles très différentes. Si l'on veut d'une part pouvoir traiter des données petites (donc des créneaux très courts) et d'autre part des données de grandes valeurs (donc de longs créneaux), il va falloir choisir les paramètres du convertisseur de sorte à avoir un pas de quantification suffisamment faible pour détecter les premières et une période d'horloge suffisamment grande pour pouvoir représenter des créneaux de grande durée correspondant aux grandes valeurs.

Une méthode pour contourner ce problème consiste à coder l'information non plus à l'aide d'uniquement un créneau mais à l'aide d'une grandeur numérique en plus d'un créneau. Cette grandeur numérique peut être considérée comme un registre d'échelle ou encore une partie entière là où le créneau représente une information entre 0 et 1. Ainsi pour pouvoir manipuler à la fois une donnée valant 0.5 et une donnée valant 25.2, on pourra dans un cas avoir une valeur numérique de 0 et un créneau d'une durée d'une demie période d'horloge (représentant la valeur 0.5) et d'un autre côté avoir un mot numérique valant 25 et un créneau de 0.2 période d'horloge (représentant la valeur 25.2).

II.5 Conclusion

 A méthode de calcul proposée qui repose sur un convertisseur dans le domaine temporel et une unité de calcul numérique, bien que conceptuellement intéressante, se révèle relativement contraignante. Les restrictions présentées dans la partie précédente limitent sa faisabilité ainsi que la capacité à l'utiliser dans n'importe quel système traditionnel principalement à cause de la nécessité de formater les données (sous la forme de créneaux et éventuellement sous la forme avancée d'un registre plus un créneau). De ce fait, elle ne sera pas

investiguée davantage sous cette forme dans le reste de ce manuscrit.

Elle a cependant le mérite d'être un cas d'étude intéressant d'un convertisseur analogique numérique basé sur un oscillateur contrôlé en tension. Elle permet de se familiariser avec ce type de convertisseurs sur des signaux très simples (des signaux en créneaux) et permet sans doute de mieux appréhender la suite du travail qui est exposé dans ce manuscrit en plus d'offrir une réponse, même limitée dans son utilisation, à la problématique de calcul par une méthode qui s'affranchit des contraintes du domaine de l'amplitude et servira de socle de départ aux travaux présentés dans la suite de ce mémoire.

Chapitre III

Le convertisseur à deux modes de fonctionnement : l'ADC hybride



« Hybridation entre synchrone et asynchrone. » Crédits : Jordan Rous



DANS ce chapitre clef, le travail réalisé sur la problématique d'un convertisseur analogique numérique pouvant fonctionner selon deux modes, en fonction de ce que la situation exige, va être présenté.

Le cheminement qui a mené à l'architecture proposée va être exposé. S'en suivra une discussion sur la faisabilité de ladite architecture en pointant du doigt les goulots d'étranglement susceptibles d'entraver son développement.

La solution retenue est celle qui consiste à exploiter la nature duale de ce qui se déroule au sein d'un oscillateur contrôlé en tension afin de pouvoir basculer d'un convertisseur synchrone à haute performance à un convertisseur asynchrone dont les performances sont dégradées mais qui est mieux adapté à un budget énergétique strict.

Après un bref rappel permettant de fixer correctement les idées sur la nature duale du convertisseur analogique numérique basé sur un oscillateur contrôlé en tension, les deux aspects de cette nature seront exploités pour obtenir deux fonctions de conversion adaptées à deux types de besoins différents. Une méthode pour combiner ces deux conversions sera finalement apportée et les limitations qui lui sont imputées seront alors discutées.

III.1 Caractère asynchrone du VCO-based ADC



La partie I.1.C.3 explique déjà le fonctionnement d'un convertisseur analogique numérique basé sur un oscillateur contrôlé en tension sous sa forme la plus élémentaire. Il y est exposé qu'un tel convertisseur (VCO-based ADC pour le rappeler) est un convertisseur synchrone. Il dispose d'une horloge et produit ses sorties uniquement aux fronts d'horloge.

Ce qui va être abordé ici, bien que la partie I.1.C.4 l'ait déjà annoncée, c'est la nature asynchrone intrinsèque à l'oscillateur contrôlé en tension qui se trouve au cœur du VCO-based ADC. Comme cette caractéristique est plus visible dans le domaine de la phase, c'est ce point de vue qui sera utilisé ici.

La première chose à clarifier à propos du comportement asynchrone de l'oscillateur contrôlé en tension, c'est la transition du signal d'entrée vers le signal de phase de l'oscillateur qui correspond au passage de la première courbe de la figure III.1 (en bleu) à la seconde courbe de la figure III.1 (en vert).

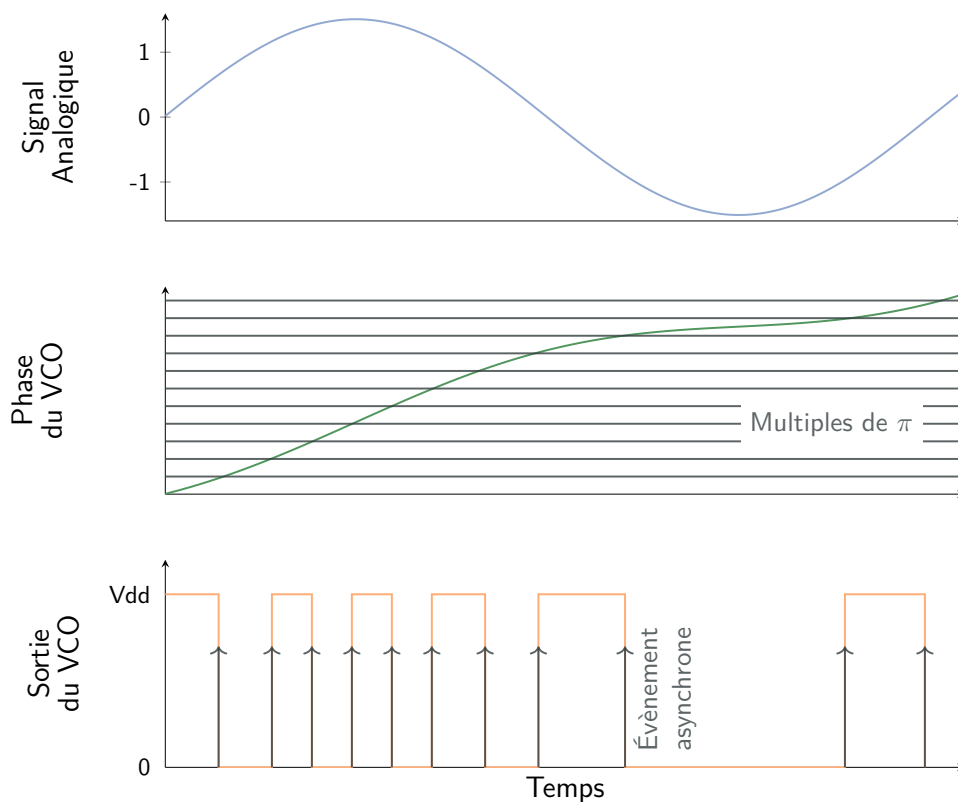


FIGURE III.1 – Fonctionnement d'un VCO vu comme un générateur d'évènements asynchrones.

Le plus simple pour comprendre cette étape est de considérer l'expression analytique de la phase de l'oscillateur en fonction de la tension de contrôle qui lui est appliquée. Ainsi, la phase de l'oscillateur obéit à l'équation suivante :

$$\phi(t) = \int_0^t [2 \times \pi (K_{vco} \times V_{in}(u) + F_{centrale})] du \quad (III.1)$$

avec :

- ϕ la phase de l'oscillateur ;
- K_{vco} le gain du VCO en $\text{Hz} \cdot \text{V}^{-1}$;
- V_{in} la tension de contrôle appliquée en entrée de l'oscillateur en V ;
- $F_{centrale}$ la fréquence centrale de l'oscillateur en Hz.

Si besoin est, rappelons que la fréquence centrale de l'oscillateur est la fréquence à laquelle l'oscillateur oscille lorsque sa tension d'entrée est nulle et que la tension de contrôle appliquée à l'entrée de l'oscillateur est ici le signal analogique à convertir.

L'équation (III.1) permet de montrer que lorsque l'oscillateur contrôlé en tension transpose la tension analogique qui lui est appliquée en entrée en un signal dans le domaine de la phase (signal qui, rappelons-le, est une vision mathématique et non un signal physique), il procède en réalité à une intégration continue. Petite subtilité cependant, à cause du terme $F_{centrale}$ dans l'équation (III.1), la phase est un signal strictement croissant.

Le passage du domaine de la phase à la sortie de l'oscillateur (de la deuxième courbe à la troisième de la figure III.1) peut être vu comme une quantification du signal de phase qui, par détection de franchissement de seuils tous les multiples de π , génère des événements asynchrones sous la forme de commutations du signal créneau de sortie.

On a donc en réalité montré ici que l'oscillateur contrôlé en tension peut en fait être vu comme un bloc qui réalise une intégration continue d'un signal analogique suivi d'un échantillonnage et d'une quantification non-uniformes (à des intervalles de temps non constants) et asynchrone de ce signal intégré par le biais d'une détection de franchissement de seuils placés tous les multiples de π . Ces événements peuvent être représentés et manipulés sous forme d'impulsions en choisissant, par exemple, de ne conserver que les instants de transitions de la sortie de l'oscillateur contrôlé en tension tel que représenté dans le troisième graphique de la figure III.1. Chaque événement asynchrone représenté par une flèche de couleur grise dans ce graphe sera alors une impulsion d'amplitude et de largeur fixées.

C'est ce fonctionnement asynchrone propre à l'oscillateur contrôlé en tension qui va être utilisé dans la partie suivante dans le but de réaliser une conversion analogique numérique asynchrone à faible performance.

III.2 Deux méthodes de traitement de la sortie d'un VCO



COMME annoncé plus tôt, nous allons commencer par présenter deux fonctions de conversion analogique numérique qui correspondent aux deux cas de figure du fonctionnement hybride que nous souhaitons réaliser avant de les regrouper en un seul convertisseur. Le but étant d'utiliser le même matériel de base pour les deux modes de fonctionnement afin d'optimiser l'architecture globale.

III.2.A Comme un train d'impulsions asynchrones

EN reprenant le raisonnement où il s'était arrêté dans la partie III.1 ; à savoir en récupérant les événements asynchrones générés par l'oscillateur contrôlé en tension (c'est-à-dire les instants de commutations de la sortie) en envoyant la sortie de l'oscillateur dans un générateur d'impulsions (un bloc qui va dériver le signal créneau pour générer des impulsions de durée constante à chaque commutation du signal de l'oscillateur), le signal récupéré devient alors un train d'impulsions asynchrones.

L'information du signal de départ est alors contenue dans la densité des impulsions du train d'impulsions. Ce signal numérique peut ensuite être traité par des unités de traitement numérique adaptées aux signaux impulsifs. La solution la plus simple restant d'utiliser un filtre passe-bas pour réaliser le traitement du train d'impulsions comme illustré en figure III.2 .

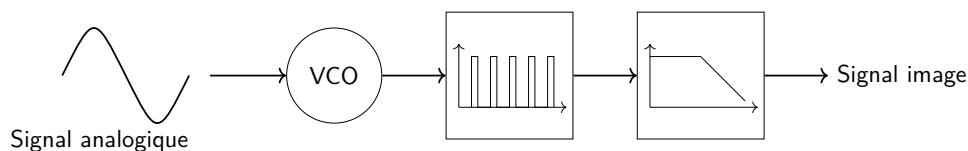


FIGURE III.2 – Utilisation d'un VCO pour une conversion asynchrone.

Le système complet se compose donc d'un oscillateur (en l'occurrence un oscillateur en anneau qui produit un signal créneau) contrôlé en tension (le bloc VCO dans la figure III.2) dans lequel le signal analogique à convertir est envoyé. La sortie de cet oscillateur est ensuite transmise à un bloc de générateur d'impulsions qui a pour but de transformer le signal créneau en un train d'impulsions correspondant à chaque transition du signal créneau. Cet ensemble constitue le convertisseur analogique numérique asynchrone.

Le bloc filtre passe-bas qui est ajouté en dernier tient lieu de « bloc de traitement » afin de vérifier que le train d'impulsions généré contient bien l'information du signal de départ dans la densité de ses impulsions.

Là encore, dans le but de mieux comprendre le détail des opérations dont on discute ici, une vue sur les signaux lors de la conversion asynchrone est proposée en figure III.3.

On retrouve dans cette figure les différents signaux mis en jeu à commencer par le signal analogique à convertir. On prend l'exemple classique d'un signal sinusoïdal. La conversion asynchrone est elle réalisée par le VCO. Le train d'impulsions (appelé signal pulsé dans la figure III.3) représente la sortie de la conversion. Cette sortie est bien asynchrone et on a bien réalisé une fonction de conversion analogique

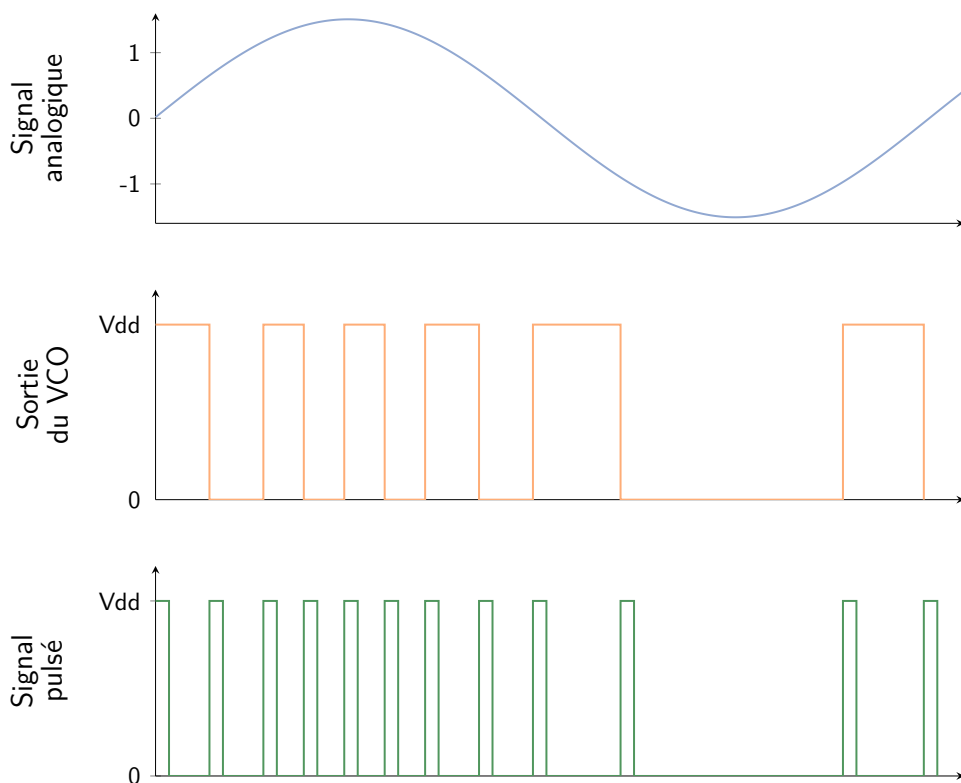


FIGURE III.3 – Vue des signaux dans le système ci-dessus.

numérique.

Le traitement présenté ici porte le nom de codage *PDM* (pour *Pulse Density Modulation*). L'avantage de cette réalisation pour notre application réside dans le fait qu'elle repose principalement sur un oscillateur contrôlé en tension. C'est cette caractéristique qui permettra, *in fine*, d'obtenir une architecture globale qui ne se contente pas d'être deux convertisseurs analogique numérique juxtaposés mais bien un seul convertisseur avec des composants communs aux deux modes de fonctionnement.

On entreprend alors de réaliser un modèle MATLAB de ce convertisseur. Les fichiers ayant servi à modéliser sous Matlab les différents blocs mis en jeu dans la figure III.2 au cours de l'expérimentation sur ce sujet sont disponibles en annexe de ce manuscrit en page 105.

À titre d'information, les résultats obtenus en simulation donnent de très faibles performances. On calcule des SNDR de l'ordre d'une dizaine de dB. Par exemple, pour un signal sinusoïdal en entrée à une fréquence de 9 MHz, en utilisant une fois de plus le modèle d'un VCO composé de 31 cellules NAND en 28 nm FDSOI et

un filtre passe-bas de type Butterworth d'ordre 2, on obtient un SNDR de 6.6 dB sur une bande passante de 20 MHz.

Il ne faut cependant pas oublier que cette architecture correspond à un mode de fonctionnement dégradé dont la mission, si l'on s'en réfère aux exemples utilisés en introduction de ce manuscrit, est uniquement de détecter la présence ou non d'un signal au milieu d'un bruit.

Ce qu'il est important de noter concernant cette conversion, c'est qu'elle ne requiert que peu d'éléments (un oscillateur contrôlé en tension et un bloc générateur d'impulsions qui revient à dériver le signal) et qu'elle ne nécessite pas d'horloge. Elle est donc particulièrement adaptée à une conversion analogique numérique dégradée avec un budget énergétique strict.

III.2.B De manière synchrone avec compteur



présent envisageons l'exercice inverse : essayer de concevoir un convertisseur analogique numérique basé sur un oscillateur contrôlé en tension (pour utiliser le plus possible le matériel présenté précédemment) avec de bonnes performances en termes de résolution.

Il a déjà été montré dans la figure I.14 de la section I.1.D que les VCO-based ADCs permettent de relativement bonnes performances en termes de résolution vis-à-vis de la bande passante (de l'ordre de 75 dB pour une bande passante de 20 MHz).

L'architecture la plus simple que l'on puisse trouver d'un tel convertisseur est présentée dans la figure III.4. Pour rappel, le principe de fonctionnement de cette architecture est développé plus en détails dans la partie I.1.C.3.

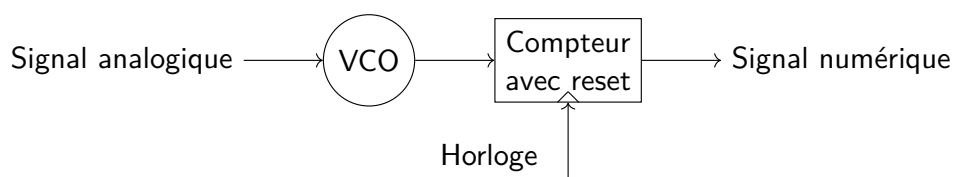


FIGURE III.4 – Utilisation d'un VCO pour une conversion synchrone.

Le but ici étant d'obtenir un convertisseur adéquat pour le mode de fonctionnement performant, il est judicieux de choisir une des architectures disponibles dans la littérature qui présente les meilleurs résultats.

L'architecture choisie est un convertisseur pseudo-différentiel au sein d'une boucle de rétroaction (architecture $\Sigma\Delta$) comme il peut être trouvé dans [K. Lee *et al.*, 2015] et dont une représentation schématique est proposée dans la figure III.5.

Cette architecture a le même comportement global que l'architecture présentée dans la figure III.4 ci-dessus. Cependant, la complexification du montage permet d'ajouter quelques optimisations au fonctionnement du convertisseur.

D'une part, le montage pseudo-différentiel a pour effet de supprimer les non-linéarités d'ordres pairs augmentant ainsi la linéarité de la conversion. D'autre part, la conversion va être améliorée par le montage en $\Sigma\Delta$ puisque la boucle de rétroaction va contribuer à réduire la dynamique en entrée des oscillateurs contrôlés en tension et va permettre de les faire travailler dans une zone plus linéaire de leur caractéristique de fonctionnement. Ce convertisseur en revanche requiert une horloge pour fonctionner.

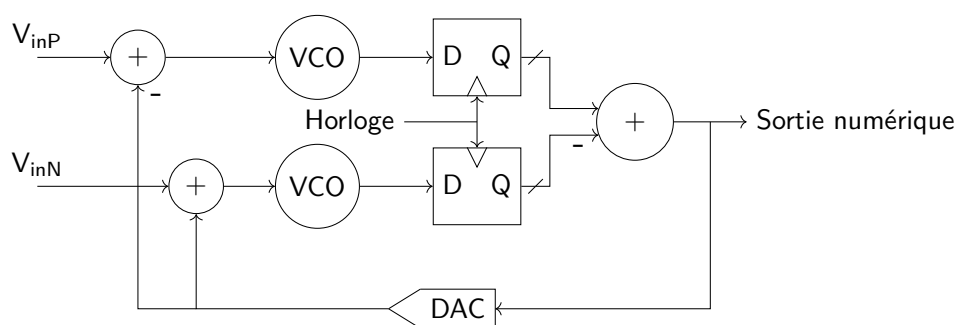


FIGURE III.5 – Architecture plus optimisée de VCO-based ADC.

A titre d'information, le convertisseur présenté dans la figure III.5 a été modélisé en prenant comme oscillateur en anneau une boucle de 31 cellules standards NAND en technologie FDSOI 28 nm. Les simulations permettent d'obtenir un rapport signal à bruit et distorsion (SNDR) de 64 dB sur une bande passante de 50 MHz et un SNDR de 56 dB sur une bande passante de 100 MHz.

Là encore, les modèles Matlab et VerilogAMS utilisés pour ces simulations sont disponibles sous une forme plus évoluée (cf. Chapitre IV) en annexe page 108 de ce manuscrit. Nous rentrerons dans le détail des modèles de simulation ultérieurement dans la partie IV.2 qui y sera dédiée.

III.3 L'exploitation des modes : l'ADC hybride à deux modes de fonctionnement

ÉCAPITULONS ce qui a été vu dans les deux parties précédentes. D'une part, un oscillateur en anneau contrôlé en tension peut être utilisé pour effectuer une conversion analogique numérique asynchrone et de faible performance en termes de résolution. D'autre part, le même bloc de base ; à savoir le VCO en anneau, peut être utilisé au sein d'une architecture plus complexe afin de produire une conversion analogique numérique plus performante mais qui requiert une horloge.

Il faut donc arriver à combiner les deux architectures pour parvenir à proposer une solution au besoin évoqué dans la problématique concernant un convertisseur analogique numérique à deux modes de fonctionnement. De la sorte, lorsque le système est dans son mode « dégradé » (se référer à la figure 6 page xvii à propos du fonctionnement en deux modes) le convertisseur analogique numérique est équivalent au schéma de la figure III.6 et lorsque le système est en mode « performant », le circuit du convertisseur est équivalent au circuit de la figure III.7.

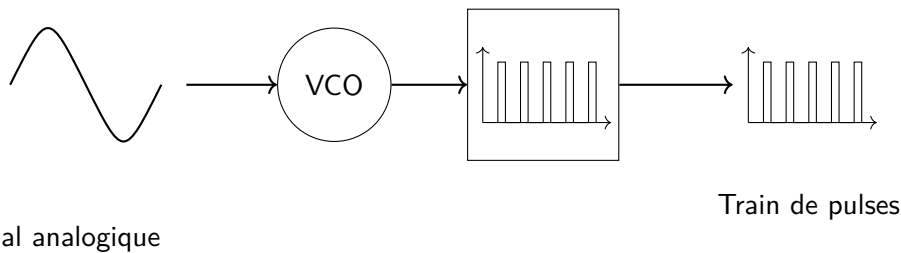


FIGURE III.6 – Schéma équivalent de l'ADC hybride en mode dégradé.

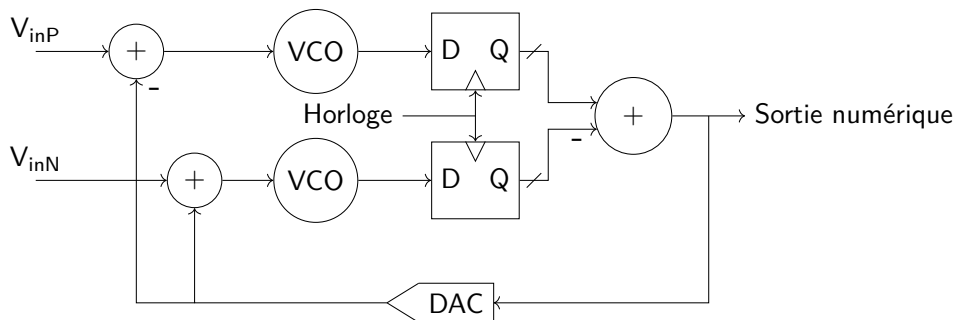


FIGURE III.7 – Schéma équivalent de l'ADC hybride en mode performant.

Il va donc falloir apporter au convertisseur une sorte d'« intelligence » pour lui permettre de décider dans quel mode il doit fonctionner. Cette partie vise à

présenter la solution retenue à la fois pour l'architecture système du convertisseur et pour le bloc de décision.

III.3.A Bloc de décision



OUS allons dans un premier temps nous focaliser sur le fonctionnement du bloc de décision. Dans un second temps, nous présenterons l'architecture à laquelle nous sommes parvenus.

Afin de rester sur l'exemple d'un circuit de détection de parole (type Google Now) qui a été avancé au début de ce manuscrit, il est décidé de se baser sur l'activité du signal afin de déterminer le mode de fonctionnement adapté. Un signal de faible activité correspondant à un mode « dégradé » du circuit (on considère attendre le signal et ne pas avoir besoin d'une grande performance) alors qu'un signal de forte activité correspond au mode « performant » (on considère qu'il y a de l'information qui arrive et qu'il faut donc la traiter avec précision).

On aboutit alors à un bloc de décision dont le comportement doit traduire la machine à états présentée dans la figure III.8. Dans son état initial, le système se met en mode dégradé (et produit donc un train d'impulsions) puis le bloc de décision va venir surveiller l'évolution de la densité des impulsions au cours du temps. Si la densité d'impulsions varie peu, cela revient à dire que la fréquence de l'oscillateur contrôlé en tension varie peu donc que le signal en entrée de l'oscillateur varie peu. L'activité du signal d'entrée étant faible, il convient de rester en mode « dégradé ».

A l'inverse, si la densité des impulsions varie beaucoup (le seuil à partir duquel on considère une variation importante est à fixer en fonction des besoins de l'application), cela indique que la fréquence de l'oscillateur varie et donc que le signal en entrée du convertisseur présente des variations. Il y a un signal à traiter donc il faut que le convertisseur passe en mode performant.

Une fois en mode performant, il devient plus facile de mesurer l'évolution du système. En effet, le convertisseur produit dans ce mode de fonctionnement des mots numériques de manière synchrone. Mesurer l'activité du signal d'entrée revient alors à mesurer l'évolution de la valeur du mot numérique d'une période d'horloge à une autre. Si les valeurs sont différentes d'un cycle à l'autre, c'est que le signal d'entrée continue d'avoir une activité élevée et donc qu'il y a une information à traiter. Il convient de rester en mode performant. Là encore, le seuil qui permet de considérer que les valeurs sont réellement différentes est à adapter en fonction de l'application. Typiquement considérer que deux valeurs sont différentes dès qu'elles diffèrent de ± 1 n'a pas de sens puisque cela revient à considérer les erreurs d'arrondi (erreur de quantification du convertisseur analogique numérique)

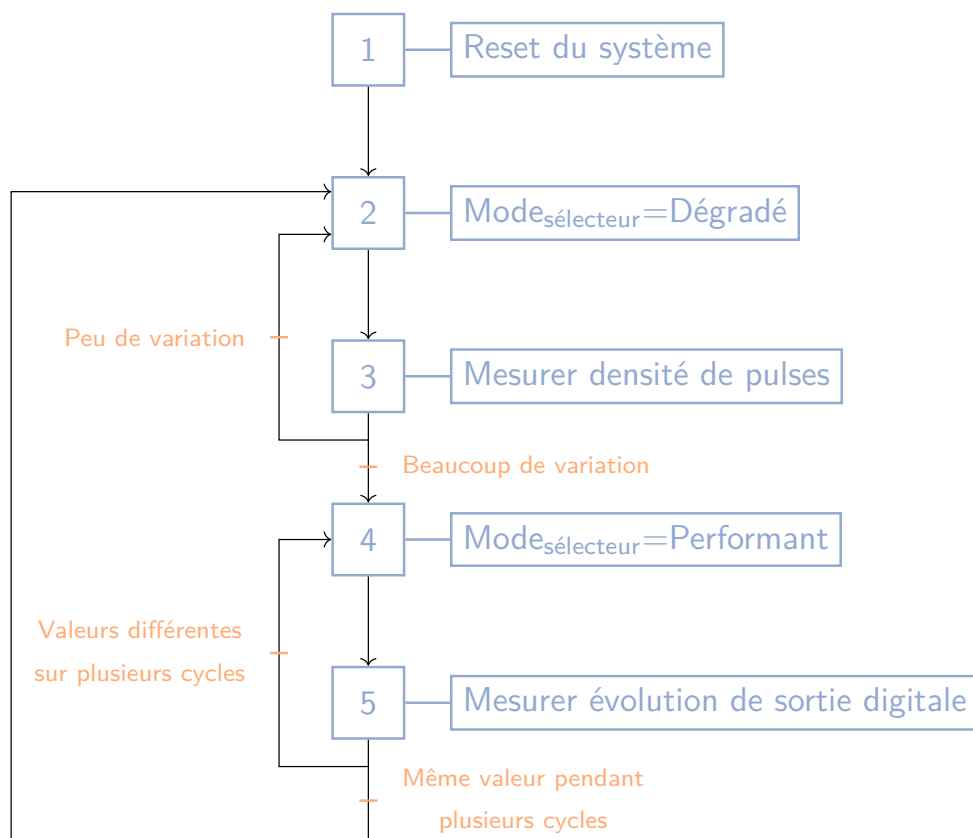


FIGURE III.8 – Machine à états du bloc de décision.

comme du signal utile.

Là encore, dans le cas où la valeur numérique varierait peu pendant plusieurs cycles d'horloge (valeur toujours à définir en regard de l'application), cela traduirait le fait que le signal d'entrée varie peu et donc qu'il n'y a plus lieu d'être en mode « performant ». Le bloc de décision coupe alors l'horloge et remet le système dans son état « dégradé ».

Le principal obstacle rencontré pour réaliser l'ensemble de ces opérations est donc la capacité d'évaluer la densité des impulsions lorsque le convertisseur est en mode « dégradé ».

Une méthode, bien que non optimale, consiste à réutiliser ce qui a été vu dans la partie III.2.A ; c'est-à-dire à récupérer un signal image à partir du train d'impulsion en ayant recours à un intégrateur. Le signal image ainsi obtenu (c.f figure III.3) reflète alors la densité des impulsions et un comparateur (sur le principe d'une détection de seuils) permet alors de savoir si la densité d'impulsions varie.

En résumé, dans sa version « dégradée », le convertisseur effectue une modulation sur la densité d'impulsions (PDM en anglais) d'une manière similaire à ce qui est utilisé dans [Yamada et Toshiyoshi, 2017]. Pour évaluer la densité des impulsions, la méthode utilisée revient à faire le décodage de la PDM.

Cette méthode n'est sûrement pas la méthode optimale pour exécuter la fonction requise. Cependant, elle est très simple à mettre en œuvre, ne nécessite que des éléments de circuiterie élémentaires (un intégrateur qui peut être réalisé avec un filtre passe-bas et un ou deux comparateurs en fonction de la finesse désirée pour estimer les variations de la densité d'impulsions). De plus, cette technique a le bon goût de ne pas faire intervenir d'horloge ce qui est dans l'intérêt du mode « dégradé ».

III.3.B Bilan



MAINTENANT que toutes les briques de l'architecture étudiée ont été posées, il est temps de rassembler tous les morceaux et de présenter l'architecture globale à laquelle a mené cette étude.

L'architecture dans son ensemble est donc composée d'une part du bloc de décision dont la fonctionnalité est détaillée dans la partie III.3.A et dont le schéma bloc (pour permettre de situer les entrées et sorties dans le reste de l'architecture) est dessiné en haut de la figure III.9 et d'autre part du convertisseur lui-même qui est capable de fonctionner dans les deux modes « dégradé » et « performant » en se basant en partie sur le même matériel et dont le schéma bloc est représenté dans la partie basse de la figure III.9.

Lorsque le bloc de décision détermine (de la façon expliquée dans la partie III.3.A) que le convertisseur analogique numérique doit fonctionner en mode « dégradé », il met le signal *Mode sel* à la valeur « Dégradé » ce qui a pour effet de couper l'un des oscillateurs contrôlés en tension (celui du bas sur la figure III.9) ainsi que de couper la génération du signal d'horloge. Le démultiplexeur commute également vers la sortie appropriée ; c'est-à-dire la sortie correspondant aux trains d'impulsions. La seule partie active de la partie conversion est donc celle colorée en bleu dans la figure III.10.

Dans le cas contraire, lorsque le bloc de décision détermine qu'il faut activer le mode « performant » du convertisseur analogique numérique, il met le signal *Mode sel* sur la valeur « performant » ce qui réactive l'horloge et le second oscillateur contrôlé en tension en plus de faire commuter le démultiplexeur pour envoyer le signal dans la chaîne qui mène à la sortie correspondant aux mots numériques. C'est alors uniquement la partie en orange sur la figure III.11 qui est active dans

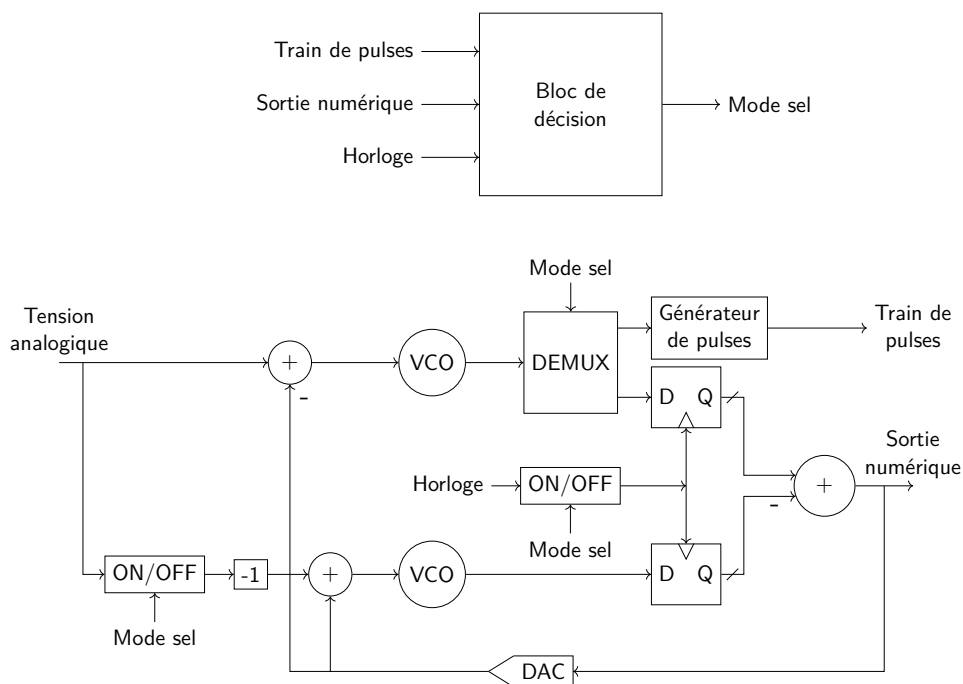


FIGURE III.9 – Schéma complet de l'ADC hybride à deux modes de fonctionnement.

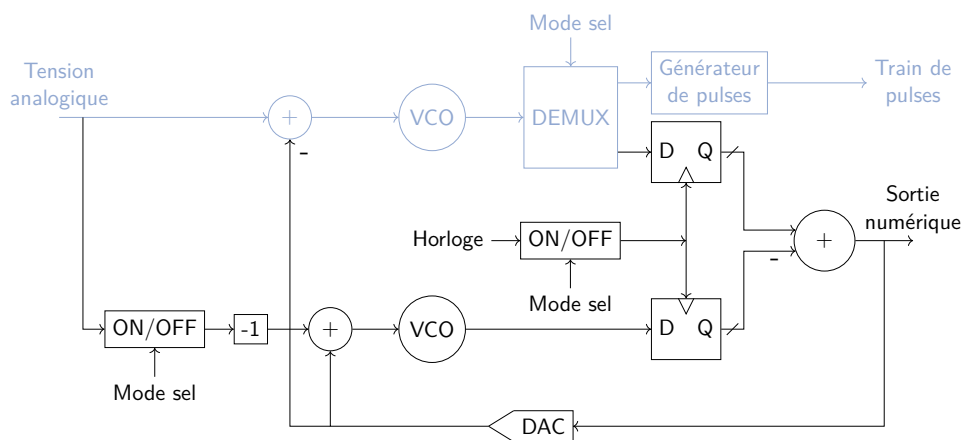


FIGURE III.10 – Partie active lors du fonctionnement en mode dégradé.

le convertisseur hybride.

L'architecture proposée remplit bien la fonction de convertisseur analogique numérique hybride à deux modes de fonctionnement qui répond au besoin présenté dans la problématique de la partie Introduction de ce manuscrit tout en opérant dans le domaine temporel afin d'éviter les contraintes croissantes amenées par les nouveaux nœuds technologiques en ce qui concerne les convertisseurs analogique

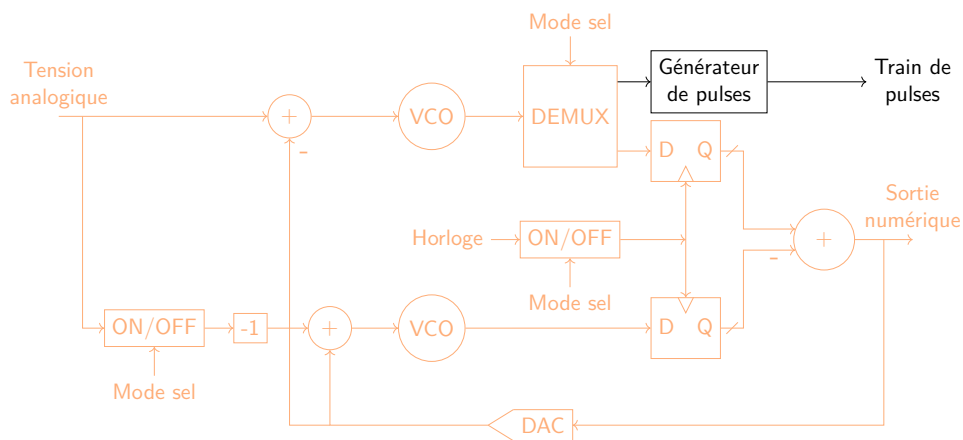


FIGURE III.11 – Partie active lors du fonctionnement en mode performant.

numérique qui opèrent dans le domaine de l'amplitude.

On se retrouve alors avec un convertisseur qui va pouvoir adapter son fonctionnement. D'une part, un mode peu performant et peu consommant sous la forme d'un convertisseur asynchrone qui génère un train d'impulsions et dont la résolution n'est que de quelques dB. D'autre part, un convertisseur synchrone, composé principalement de deux VCO, plus gourmand en énergie, mais qui permet d'atteindre des résolutions de l'ordre de 60 dB.

Mais l'on pourrait pousser davantage le raisonnement précédent pour améliorer la granularité des modes de fonctionnement. Pourquoi se limiter au deux modes présentés ici alors qu'on pourrait, avec le même matériel mais un bloc de décision différent, fonctionner selon 3 modes. On pourrait en effet envisager un mode « intermédiaire » où le convertisseur fonctionnerait comme un VCO-based ADC (donc avec horloge) mais où l'on activerait qu'un seul VCO. Cela donnerait alors une moins bonne linéarité de conversion mais réduirait la consommation du système puisqu'un seul VCO consommerait.

Mieux encore, on pourrait envisager une utilisation encore plus fine du convertisseur dans son mode « performant ». Comme l'architecture repose sur deux VCO au sein d'un montage différentiel, le principe d'adaptation poussé à son paroxysme consisterait à pouvoir régler les fréquences centrales des deux VCO. De la sorte, en théorie tout du moins, on pourrait venir adapter la consommation du convertisseur en mode performant à la fréquence du signal d'entrée. La résolution ne changerait pas mais la consommation pourrait évoluer en fonction de la bande-passante nécessaire à la conversion. Il faudrait alors changer le bloc de décision pour qu'il prenne en considération d'autres critères de décision et qu'il permettent de générer les signaux nécessaires au pilotage des VCO.

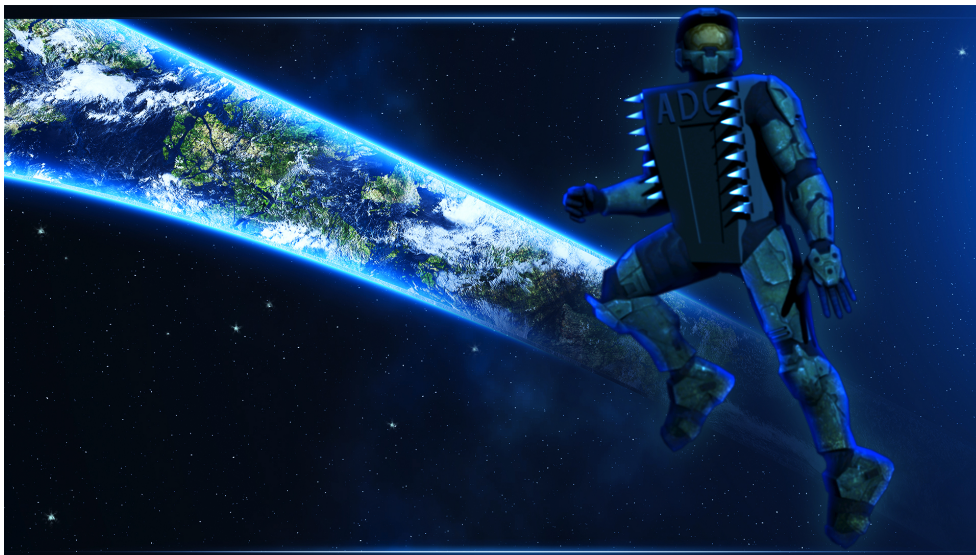
Malheureusement, nous n'avons pas eu le loisir, faute de temps mais aussi de compétences en conception de VCO, d'approfondir les recherches autour de ces idées. Nos travaux se sont orientés sur le mode « performant » de l'architecture présentée dans ce chapitre et sur une possibilité d'amélioration de ce mode.

Rappelons que le but du mode « performant » est de présenter un convertisseur analogique numérique (basé sur des oscillateurs contrôlés en tension dans ce cas) avec le meilleur ratio SNDR/Bande passante. L'étape naturelle est donc de regarder les limitations (vis-à-vis de ce ratio) de ce convertisseur en mode « performant » et d'essayer d'en proposer une version améliorée qui pousserait un peu plus la barrière SNDR/Bande passante de l'état de l'art tel que présenté dans la figure I.14 de la partie I.1.D.

Toute cette étude ainsi qu'une proposition d'amélioration de la résolution du convertisseur présenté ici dans son mode performant font l'objet de la partie suivante de ce manuscrit et composent le cœur de ce travail de thèse.

Chapitre IV

***Un mode performant
amélioré : l'ADC avec
extraction du résidu***



« Allégorie d'un ADC amélioré. » Crédits : Guyhème Buffeteau



DANS ce chapitre, nous allons nous intéresser au mode performant du convertisseur présenté dans le chapitre précédent. Le but de ce travail est d'améliorer le VCO-based ADC existant pour lui permettre d'atteindre une meilleure résolution.

Dans un premier temps, l'origine de l'erreur de quantification du VCO-based ADC va être discutée ainsi qu'une méthode pour extraire cette erreur dans le but de la compenser. L'idée va ensuite être testée au travers de plusieurs simulations et sa faisabilité va être étudiée afin de conclure quant à la pertinence de son application.

Pour finir, un modèle plus proche de la réalité sera mis au point qui permettra de simuler et d'extraire des performances permettant de comparer la solution envisagée avec les convertisseurs de l'état de l'art (en gardant à l'esprit que ce ne sont que des résultats de simulation).

IV.1 Présentation de l'idée de résidu



Le point de départ de ce travail est le convertisseur analogique numérique basé sur un oscillateur contrôlé en tension correspondant au mode de fonctionnement « performant » de l'architecture présentée dans le Chapitre III. On considère donc le convertisseur correspondant au schéma de la figure III.11 page 64.

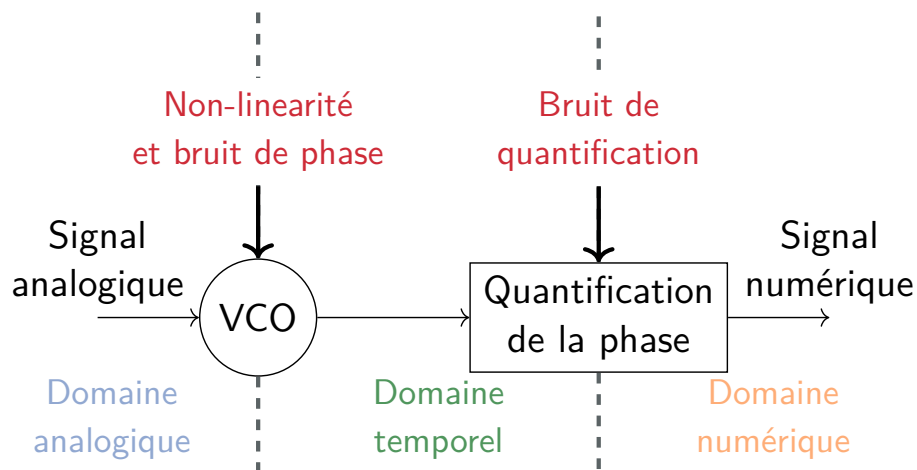


FIGURE IV.1 – Principales pistes d'amélioration des VCO-based ADCs.

Avant de pouvoir proposer une amélioration de la résolution d'un VCO-based ADC, il convient de comprendre où apparaissent les non-idéalités au cours de

la conversion. La figure IV.1 recense ces non-idéalités ainsi que l'endroit de la conversion où elles entrent en jeu.

Le bruit de phase dépend de l'oscillateur contrôlé en tension et les oscillateurs en anneaux, dont la conception est plus facile et qui tirent profit de leur architecture très fortement teintée numérique, ne sont malheureusement pas très performants vis-à-vis de cette caractéristique.

La non linéarité, elle aussi propre au VCO choisi, a déjà été adressée de plusieurs façons (cf. le chapitre I page 14). La méthode retenue pour l'architecture du convertisseur (un convertisseur rebouclé $\Sigma\Delta$ avec un montage pseudo-différentiel) permet déjà d'apporter une amélioration concernant ce point.

C'est donc le bruit de quantification que nous allons essayer de réduire pour améliorer la résolution du convertisseur. Pour y arriver, nous allons nous projeter de nouveau le fonctionnement du VCO-based ADC dans le domaine de la phase comme illustré dans la figure I.12 page 18. Le but étant de remonter à l'origine de l'erreur de quantification.

La conversion, rappelons-le, consiste à intégrer le signal d'entrée (ce qui le transpose dans le domaine de la phase) avant de le quantifier. Ainsi, au front montant d'horloge, la valeur de la phase va être échantillonnée et quantifiée (idéalement sans commettre d'erreur) et il suffit de lui soustraire la valeur de la phase échantillonnée au coup d'horloge précédent pour connaître l'accroissement de la phase au cours de la période d'horloge.

Seulement, la phase du VCO (vue mathématique de l'esprit) ne nous est accessible que via les changements de fronts de la sortie du VCO ; c'est-à-dire à chaque fois que la phase du VCO franchit un seuil multiple de π . La valeur qui est récupérée au front montant de l'horloge n'est donc pas la réelle valeur de la phase à cet instant mais la dernière valeur connue de la phase ; à savoir la valeur du dernier seuil franchi avant le front d'horloge.

C'est cette différence entre la valeur de la phase à l'instant d'échantillonnage et la valeur à laquelle on a accès (celle du dernier multiple de π franchi) que l'on appelle l'erreur de quantification.

La figure IV.2 illustre l'explication précédente et clarifie l'origine de l'erreur de quantification dans le domaine de la phase pour conceptualiser plus facilement cette notion.

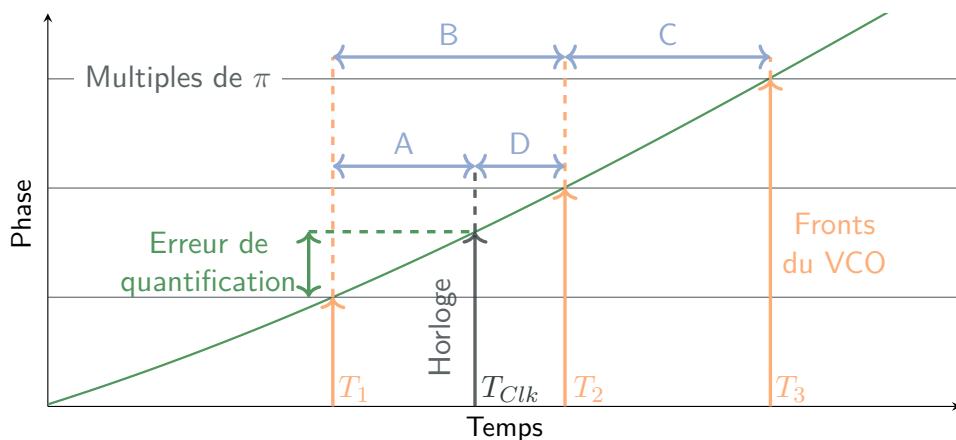


FIGURE IV.2 – Cause et apparition de l'erreur de quantification de la phase.

Maintenant que l'origine de l'erreur de quantification est connue, l'approche choisie est très simple : il faut récupérer la valeur de cette erreur de quantification afin de pouvoir la compenser.

La méthode qui est proposée pour remplir cet objectif est la suivante :

- On fait l'hypothèse que la phase est linéaire entre deux franchissements successifs de multiples de π . NB : ces franchissements correspondant à une commutation de la sortie du VCO, on parlera indifféremment de franchissements de seuils de la phase ou de fronts du VCO à partir de maintenant (l'un prenant place dans le domaine de la phase et l'autre pouvant être visible sur la sortie du VCO).
- On peut ensuite connaître la valeur de l'erreur de quantification en connaissant l'instant d'horloge et les instants de franchissement de seuils de la phase.

De manière plus formelle, si l'on considère la situation décrite dans la figure IV.2, l'hypothèse de linéarité que l'on fait est de supposer que la phase est linéaire entre les instants T_1 et T_2 . De la sorte, toujours en prenant les notations de la figure IV.2, une simple règle de proportionnalité nous permet de déduire que l'erreur de quantification peut être exprimée par le ratio suivant :

$$Err_{quant} = \frac{A}{B} \times \pi \quad (IV.1)$$

avec :

Err_{quant} l'erreur de quantification commise sur la valeur de la phase.

Cette hypothèse de linéarité ne sort pas de nulle part. En effet, la durée séparant les instants T_1 et T_2 correspond à la période instantanée du VCO. L'aspect

non-linéaire de la phase, lui, est l'image de la fréquence du signal en entrée du VCO. Or la fréquence du VCO, même à sa valeur minimale, reste bien supérieure à la fréquence du signal puisqu'il a déjà été établi que les VCO-based ADC fonctionnaient avec un rapport de sur-échantillonnage élevé (une fréquence d'échantillonnage nettement supérieure à deux fois la fréquence d'entrée) et que la fréquence du VCO est elle-même supérieure à la fréquence d'horloge. En définitive, la fréquence du VCO étant très nettement supérieure à la fréquence du signal d'entrée, le signal de phase au cours d'une période de VCO ne changera pas significativement de pente.

Comme la phase est une grandeur abstraite et que le signal auquel on peut physiquement avoir accès dans le convertisseur est la sortie du VCO, il peut être intéressant d'exprimer l'erreur de quantification de la phase dans le domaine temporel via les signaux qui sont physiquement accessibles. Ainsi, la figure IV.3 permet cette représentation. D'un côté, on a la représentation du mécanisme d'extraction du résidu dans le domaine de la phase (figure IV.2) pour lier le processus d'extraction du résidu à son origine. De l'autre côté, figure IV.3, on a le même processus mais cette fois avec des signaux réellement accessibles dans le circuit (et non plus la phase qui est une vue de l'esprit) ; à savoir la sortie du VCO et des signaux créneaux correspondant aux différents instants importants (T_{ii}) de la conversion. Les notations utilisées dans la figure IV.3 sont d'ailleurs les mêmes que celles de la figure IV.2 afin de bien mettre en exergue le fait qu'il s'agit exactement de la même situation qui est étudiée de deux points de vue différents.

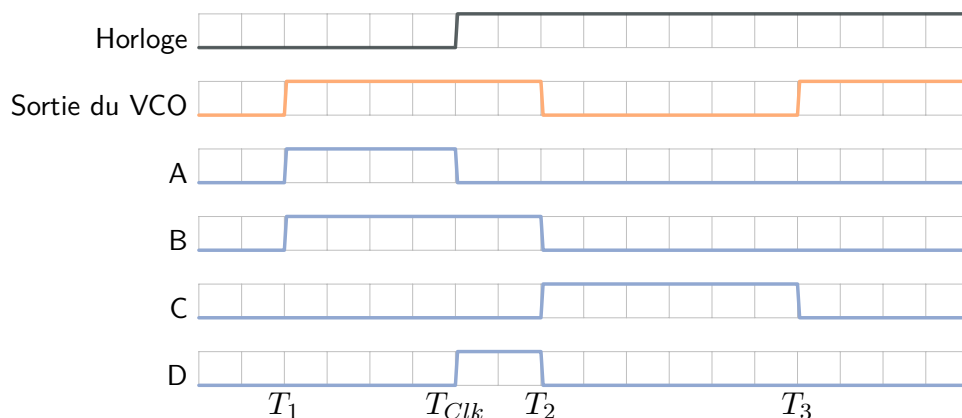


FIGURE IV.3 – Extraction du résidu vue dans le domaine temporel.

L'idée de base étant posée, il convient maintenant de montrer sa validité et son intérêt. Il faut également étudier sa faisabilité pour déterminer si cette proposition a un réel intérêt ou si ce n'est qu'une méthode théorique qui n'a pas de cas d'utilisation.

Tout ceci nécessite de pouvoir simuler le système et de pouvoir modifier les différents paramètres des blocs impliqués dans l'architecture. En substance, il faut pouvoir modéliser le convertisseur étudié. C'est l'objet de la partie suivante.

IV.2 Modélisation de la solution



DANS cette partie dédiée à la modélisation du convertisseur analogique numérique basé sur un oscillateur contrôlé en tension, nous allons aborder à la fois les contraintes liées à la modélisation de cette famille d'architectures et les solutions pour s'affranchir de ces contraintes mais nous allons aussi décrire les modèles qui seront utilisés dans tout le reste de ce travail. Les modèles en question sont disponibles en annexe à partir de la page 108.

IV.2.A Modéliser le VCO



LE bloc au cœur d'un VCO-based ADC (et ce qui se révélera le plus contraignant à simuler), c'est bien évidemment l'oscillateur contrôlé en tension. La solution la plus intuitive pour simuler un VCO c'est d'utiliser un VCO déjà existant dans une bibliothèque de composants et de lancer une simulation électrique de ce composant à l'aide d'un simulateur de type EldoTM ou SpiceTM.

Cette solution naïve n'est absolument pas réalisable en pratique. Dans un contexte de développement constitué de multiples prototypages et tests successifs (succession de cycles en V), les simulations électriques sont beaucoup trop consommantes en temps de simulation et sont donc à proscrire.

C'est une nécessité de s'orienter vers une modélisation dite *comportementale* du bloc à simuler. Ce modèle, comme son nom l'indique, va se focaliser sur le comportement global d'un bloc plutôt que sur les équations électriques des éléments qui le composent. Cette approche est très communément employée quand il s'agit de concevoir un système.

Pour élaborer un tel modèle comportemental de l'oscillateur contrôlé en tension, on revient au fonctionnement mathématique du VCO que l'on a découpé en trois étapes comme expliqué dans la partie I.1.C.4 page 17 et qui est illustré dans la figure IV.4.

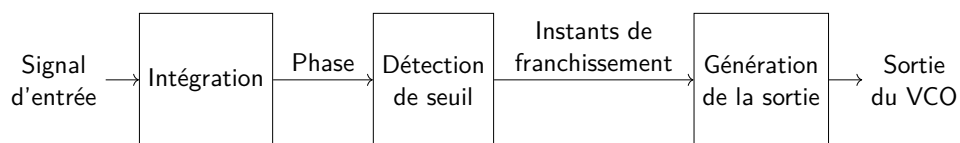


FIGURE IV.4 – Modèle comportemental d'un VCO.

Dans ce modèle comportemental, l'étape la plus critique est celle de détection des instants de franchissement de seuils. En effet, le but étant de récupérer l'instant exact de cette transition afin de pouvoir générer le front de sortie de l'oscillateur contrôlé en tension, les calculs réalisés au cours de la simulation doivent donc à la fois avoir une grande précision sur la valeur des signaux simulés (pour pouvoir détecter le moment où la valeur franchit le seuil) mais également être très précis dans le domaine temporel (pour pouvoir générer un événement au moment du franchissement et ne pas introduire d'erreur sur la génération du front du VCO).

Cette contrainte forte quant à la détection et la génération précise des fronts du VCO est d'autant plus critique si la sortie du VCO est utilisée dans un système asynchrone (dont les fronts du VCO sont les signaux qui vont commander les actions en aval du bloc).

On se propose alors d'utiliser MATLABTM qui est un outil bien connu de la communauté en matière de logiciel de calcul et qui se prête très bien à la modélisation de systèmes grâce à ses performances en termes de calculs matriciels.

En calculant la phase grâce à l'équation mathématique qui la relie au signal d'entrée (pour rappel l'équation (III.1) page 53) puis en détectant les instants où cette phase franchit des multiples de π , on peut réaliser un modèle de VCO. C'est d'ailleurs le modèle qui a été utilisé dans le Chapitre II et que l'on peut retrouver en annexes à la page 102.

Il y a cependant deux ombres au tableau d'un tel modèle. La première, c'est que lorsqu'un signal (en l'occurrence la phase) atteint de très grandes valeurs (ce qui est le cas dans un VCO puisque la phase est sans cesse croissante étant donné que c'est l'intégrale de la fréquence et que cette dernière ne peut être négative) les fonctions MATLAB qui peuvent permettre aisément de détecter les franchissements de seuils (*floor*, *rem*, ...) ne fonctionnent plus correctement.

Ce comportement, (documenté dans le manuel MATLAB) impose de conserver la phase dans un intervalle donné pour éviter une telle situation. C'est d'ailleurs ce qui est fait dans le modèle présenté en annexes. Cela nous contraint à passer d'un mode de fonctionnement matriciel (l'intégralité du signal est traité en parallèle) dans lequel MATLAB excelle à un mode de fonctionnement séquentiel dans lequel MATLAB montre des faiblesses.

On pourrait penser qu'il faudrait alors trouver un moyen de conserver la phase bornée tout en restant dans le domaine du calcul matriciel pour parvenir à régler ce problème. La réalité est que nous n'avons même pas cherché à le faire. C'est là que la seconde ombre au tableau entre en jeu.

Comme il a été annoncé, l'architecture que nous cherchons à simuler *in fine* est un VCO-based ADC rebouclé dans une architecture de convertisseur $\Sigma\Delta$. Or, pour pouvoir simuler une architecture rebouclée, il est nécessaire de connaître l'état et la sortie du système à l'instant t avant de pouvoir calculer son état à l'instant $t+1$.

Cela signifie qu'il est obligatoire de recourir à du calcul séquentiel et donc d'abandonner l'espoir de trouver une solution permettant le calcul matriciel.

Maintenant que cette possibilité est définitivement écartée, concentrons-nous sur les choix que nous avons pour modéliser le comportement mathématique du VCO. On retrouve communément deux approches pour traiter ce problème. D'un côté, une simulation à échantillonnage constant, où les signaux sont échantillonnés à fréquence uniforme (le pas de la simulation) et les calculs sont faits sur chacun de ces points. C'est le cas d'une simulation MATLAB. D'un autre côté une approche qui consiste à échantillonner de manière adaptative les signaux. De la sorte, seuls les points utiles pour atteindre la précision de simulation demandée sont échantillonnés et traités. C'est la méthode utilisée pas les moteurs de simulation de type Spice ou Eldo par exemple.

Néanmoins, ces deux approches restent très gourmandes en temps de simulation. La première à cause de la quantité astronomique de points à traiter dans notre cas (à cause du besoin critique de précision évoqué plus tôt). La seconde à cause du nombre de points qui reste conséquent mais surtout car les algorithmes de convergence utilisés par les simulateurs demeurent gourmands en temps.

A l'inverse, les simulateurs purement numériques eux, sont réputés pour leur vitesse d'exécution très élevée. C'est principalement dû au fait qu'ils fonctionnent sur évènements.

On se propose alors de réaliser un modèle de VCO qui serait compatible avec les simulateurs numériques (en VerilogAMS en l'occurrence) et qui pourrait nous permettre de simuler rapidement notre architecture cible pour pouvoir la prototyper efficacement.

On propose alors un modèle entièrement compatible avec un simulateur numérique qui se comporte comme un bloc de délais variable. Le modèle fonctionne de la manière suivante. Il commence par échantillonner la tension d'entrée à l'instant initial. En fonction de cette tension, il calcule la durée d'une demi-période que le

VCO qu'on cherche à simuler aurait. Au bout de cette durée, la sortie du modèle de simulation est commutée et le signal d'entrée est de nouveau échantillonné. Là encore, la durée de la demi-période qu'aurait le VCO est calculée et le processus se répète ainsi.

Un pseudo-code correspondant à ce processus est fourni en figure IV.5 afin de permettre de clarifier les choses. On y retrouve les différentes étapes décrites précédemment.

On est en droit de se demander si le modèle ainsi proposé reflète bien le comportement mathématique initialement présenté. En effet, ce modèle ne « met pas à jour » la valeur de la période instantanée du VCO pendant l'attente d'une demi-période. Il ne reproduit pas le comportement d'intégration continue du signal d'entrée. Ce modèle se comporte en réalité comme si la fréquence instantanée du VCO ne variait pas pendant une demi-période du VCO.

```
LIRE Tension(Signal d'entrée);  
  
CALCULER la Demi-période correspondante;  
  
ATTENDRE pendant Demi-période;  
  
COMMUTER la Sortie du VCO;  
  
RECOMMENCER l'algorithme;
```

FIGURE IV.5 – Pseudo-code du modèle event-driven.

On discutera par la suite de cette hypothèse, de son caractère acceptable, de son impact sur la simulation et de sa plage de validité. Mais dans un premier temps nous allons commencer par vérifier que le modèle mis au point remplit bien sa mission première ; à savoir modéliser le comportement d'un VCO de manière rapide.

On va pour cela comparer le nombre de points nécessaires à la simulation du VCO sur le même intervalle de temps pour les différents modèles évoqués plus tôt, puis comparer le temps nécessaire à la simulation du même VCO, avec le même signal d'entrée et sur la même durée modélisé selon les trois manières étudiées.

Les figures IV.6, IV.7 et IV.8 correspondent à la comparaisons du nombre de points nécessaires aux différentes méthodes de simulation étudiées (l'échantillonnage constant, l'échantillonnage adaptatif et le modèle numérique event-driven).

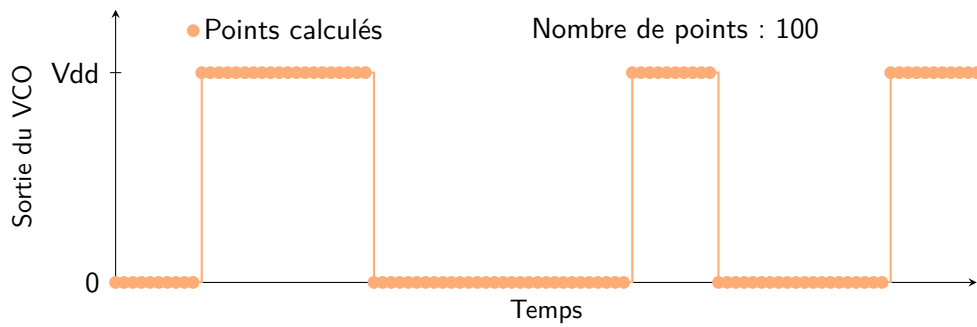


FIGURE IV.6 – Points calculés avec le modèle à échantillonnage constant (Matlab).

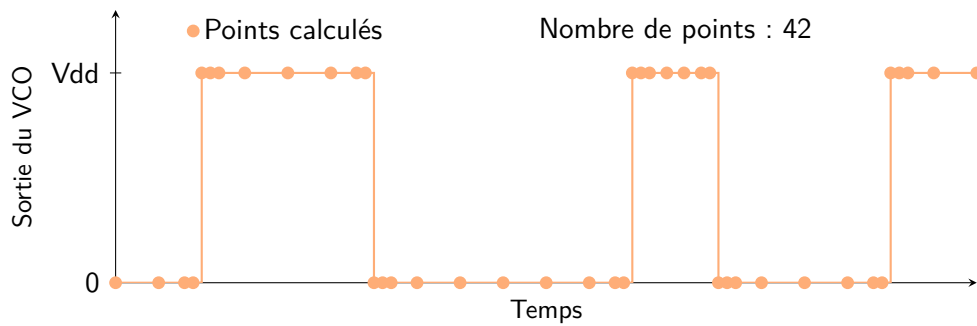


FIGURE IV.7 – Points calculés avec le modèle à échantillonnage adaptatif (Spectre).

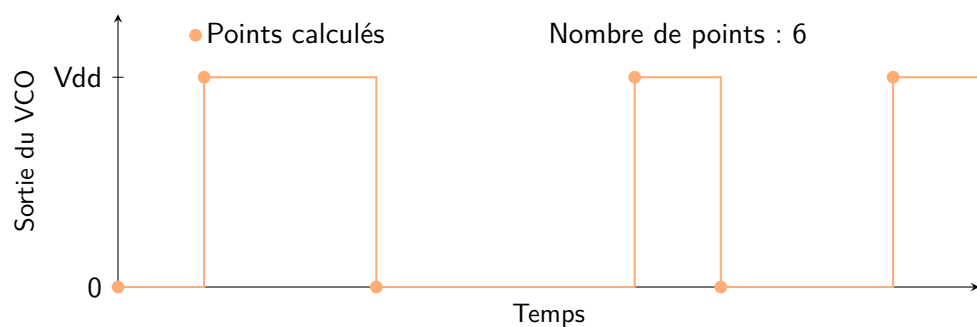


FIGURE IV.8 – Points calculés avec le modèle à échantillonnage event-driven (VerilogAMS).

Sur ces figures, on peut clairement constater l'avantage apporté par le modèle complètement numérique. Le nombre de points nécessaires à la simulation du modèle en VerilogAMS est nettement inférieur à celui nécessaire dans les autres modèles.

Le tableau suivant récapitule le temps de simulation nécessaire à la même situation avec les différents modèles étudiés. Il est important de noter que les simulations ont été réalisées sur le même ordinateur avec la même charge de travail sur la machine. Cependant, comme le temps de simulation dépend très fortement de l'environnement et de la machine utilisée, les valeurs absolues des temps de simulation ne sont pas pertinentes. Il convient donc de considérer les différentes valeurs les unes par rapport aux autres. Enfin, à titre purement indicatif, la même simulation est effectuée avec un modèle de simulation électrique afin de montrer la différence colossale qu'il y a entre les modèles électriques et comportementaux.

Élec- trique	Échantillonnage constant	Échantillonnage adaptatif	Event- driven
45720 s	602 s	391 s	23 s

TABLEAU IV.1 – Tableau comparatif des temps de simulation

Les modèles VerilogAMS et MATLAB sont disponibles en annexes de ce manuscrit aux pages 104 et 109.

On en profite ici pour introduire une nouvelle notion qui sera prise en compte dans les modèles de la suite de ce manuscrit : la notion de VCO multi-phase. Dans l'exemple que nous étudions ; à savoir un VCO composé de 31 cellules NAND en 28 nm FDSOI, on peut considérer ce VCO soit comme un seul VCO dont la sortie est la sortie de la dernière porte NAND, soit comme un VCO qui produit 31 sorties (la sortie de chacune des cellules NAND) et qui, une fois concaténées toutes ensemble, reproduisent le comportement d'un VCO dont la vitesse serait 31 fois supérieure.

Pour résumer, un VCO à N phases vient quantifier la phase non plus tous les π mais tous les $\frac{2\pi}{N}$. On pourra donc par la suite considérer des modèles de VCO à N phase en multipliant dans les modèles la fréquence de fonctionnement des VCO par N. En pratique pour modéliser un VCO à 31 phases, plutôt que de générer 31 sorties, on pourra se permettre de modéliser un VCO avec une seule sortie 31 fois plus rapide.

Avant de s'attarder aux effets et au domaine de validité de l'hypothèse faite dans le fonctionnement du modèle tout numérique (fréquence de VCO constante sur une demi-période) il faut étudier la possibilité d'ajouter au modèle proposé (qui est pour l'instant purement idéal) les non-idéalités qu'il est nécessaire de prendre en compte lors d'une simulation de VCO-based ADC.

IV.2.B Ajout des non idéalités



ETTE partie est consacrée à la modification du modèle simpliste développé dans la partie précédente pour lui permettre de prendre en compte les deux non-idéalités majeures qui sont critiques dans les convertisseurs analogique numérique basés sur des VCO que nous voulons étudier. Il s'agit de la non-linéarité du VCO et son bruit de phase.

IV.2.B.1 La non-linéarité

Afin de pouvoir simuler la non-linéarité de la caractéristique tension/fréquence d'un VCO, on décide de modifier à la fois dans le modèle MATLAB et dans le modèle VerilogAMS la façon dont sont calculées respectivement la phase et la durée de la demi-période.

Ainsi, à la place d'une fonction linéaire pour calculer ces grandeurs, on opte pour une fonction non linéaire en fonction de la tension d'entrée du VCO.

Pour déterminer la fonction non linéaire à utiliser ainsi que ses coefficients, on décide de reproduire le comportement d'un VCO en anneaux qui serait composé de 31 cellules standards NAND en technologie 28 nm FDSOI (étant donné que c'est ce VCO qui sera utilisé dans nos architectures).

On procède donc à une simulation électrique du VCO pour plusieurs valeurs de tension d'entrée afin d'en établir la caractéristique tension/fréquence qui est tracée en bleu dans la figure IV.9.

On s'emploie ensuite à venir extraire une équation de courbe qui se rapproche suffisamment de cette caractéristique électrique. On décide, comme illustré par la courbe orange de la figure IV.9, d'utiliser une fonction polynomiale de degré 2 qui nous offre une approximation plutôt correcte de la caractéristique électrique avec un coefficient de détermination de 0.9954 tout en restant sur un modèle mathématique extrêmement simple.

Si le besoin s'était fait ressentir d'une plus grande précision d'approximation, ou si l'on avait eu besoin de points particuliers de la caractéristique électrique, il aurait été tout à fait envisageable d'utiliser un modèle mathématique plus complexe comme une fonction puissance ou une série de Taylor pour affiner le modèle mathématique. Dans la suite de ce travail, nous nous contenterons du polynôme de degré 2 très facile à manipuler.

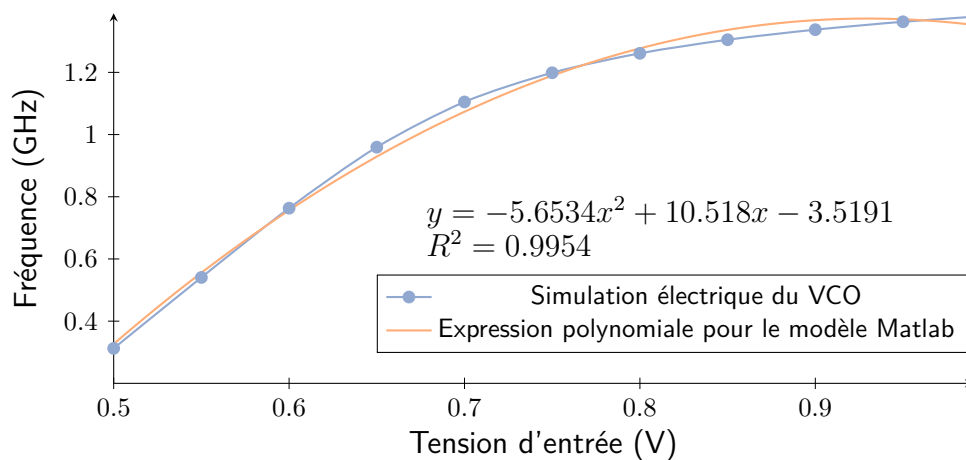


FIGURE IV.9 – Caractéristique électrique du VCO choisi et modèle mathématique associé.

Une fois de plus, les modèles de VCO qui prennent en compte la non-linéarité sont disponibles en annexes en page 109 pour le modèle MATLAB et en page 111 pour le modèle VerilogAMS.

IV.2.B.2 Ajout du bruit de phase

Dans cette partie, nous allons expliquer comment la modélisation du bruit de phase peut être ajoutée à notre modèle tout numérique. Nous allons commencer par rappeler brièvement ce qu'est le bruit de phase et nous verrons ensuite de quelle manière nous l'avons ajouté au modèle VerilogAMS.

Commençons par préciser ce qu'on entend ici par bruit de phase. Lorsque l'oscillateur contrôlé en tension se voit appliquer une tension de contrôle constante, il est supposé osciller à une fréquence donnée. Or en pratique, si l'on visualise le spectre de la sortie de l'oscillateur, il va osciller autour de la fréquence théorique définie par la tension de contrôle (au lieu de n'osciller qu'à cette fréquence). Plus précisément, ce à quoi nous allons nous intéresser, c'est la composante en $\frac{1}{f^2}$ de ce bruit.

Pour prendre en compte ce bruit dans notre modèle, nous allons nous servir de la relation qu'il y a entre ce dernier et une gigue sur la valeur de la période d'oscillation du VCO. Cette relation est bien décrite dans [Staszewski *et al.*, 2005].

Il apparaît que le bruit de phase qui nous intéresse (en $\frac{1}{f^2}$) peut être ramené à une gigue sur la période du VCO qui suit une loi normale de moyenne 0 et

d'écart-type σ^2 telle que l'on a la relation suivante :

$$\mathcal{L}(f) = \frac{F_{oscillation}^3 \times \sigma^2}{f^2} \quad (IV.2)$$

avec :

- $\mathcal{L}(f)$ la densité spectrale du bruit de phase à la fréquence f ;
- $F_{oscillation}$ la fréquence à laquelle l'oscillateur doit osciller sans le bruit de phase ;
- σ^2 l'écart-type de la loi normale de la distribution de la gigue sur la période.

On peut donc alors modifier notre modèle VerilogAMS pour qu'il prenne en compte ce comportement. On aboutit alors au modèle de VCO disponible en annexes à la page 112 dont l'algorithme peut être condensé dans le pseudo-code présenté dans la figure IV.10.

```
//LIRE Tension(Signal d'entrée);
V_in= Tension(Signal d'entrée);

//CALCULER la Demi-période correspondante (non-linéarité);
Demi-période=-5.6534e9*(V_in)^2+10.518e9*V_in-3.5191e9;

//CALCULER le Jitter une fois par période (bruit de phase);
Jitter=$dist_normal(seed,0,Sigma^2);

//ATTENDRE pendant Demi-période;
#(Demi-période+Jitter);

//COMMUTER la Sortie du VCO;
Sortie de VCO <= ~Sortie de VCO;

//RECOMMENCER l'algorithme;
```

FIGURE IV.10 – Pseudo-code prenant en compte le bruit de phase.

On retrouve les étapes du modèle précédent auxquelles ont rajouté l'étape de prise en compte du bruit de phase. Cela consiste à générer la gigue sur la période (variable jitter) avec une loi normale dont les paramètres correspondent à ce qui est amené dans l'équation (IV.2) puis à rajouter cette gigue une fois toutes les deux actualisations de la sortie du VCO (la gigue s'applique sur une période non sur une demi-période).

Pour une rapide vérification du comportement du modèle on trace la densité spectrale de puissance de la sortie du VCO simulé avec le modèle VerilogAMS pour une tension de contrôle constante mais avec un bruit de phase.

Le résultat de cette simulation est reporté dans la figure IV.11. On y voit bien l'étalement spectral autour de la fréquence d'oscillation de l'oscillateur qui est caractéristique de la présence du bruit de phase.

Pour être totalement rigoureux, il faudrait préciser le comportement de ce modèle dans le cas d'une simulation d'un oscillateur contrôlé en tension multi-phase. Pour ces VCO composés de plusieurs entités (plusieurs blocs de délais, plusieurs cellules NAND, ...) il faudrait en toute rigueur considérer le bruit de phase de chacune des entités du VCO pour reproduire le plus fidèlement possible le bruit de phase du VCO. Cela complexifie cependant le modèle et pour optimiser le temps disponible nous avons préféré ne considérer qu'un seul bruit de phase global sur l'oscillateur.

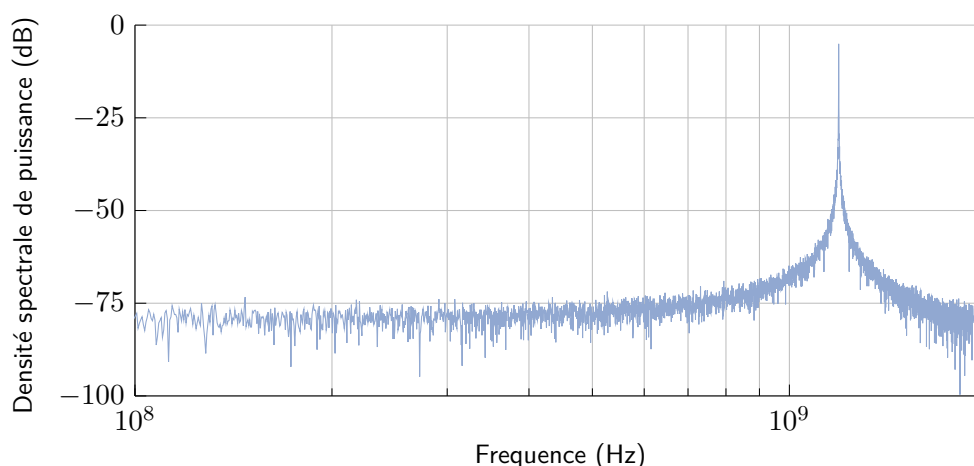



FIGURE IV.11 – Visualisation du bruit de phase avec le modèle VerilogAMS amélioré.

IV.2.C Plage de validité du modèle tout numérique

 EST dans cette partie que nous allons aborder les conséquences de l'approximation faite dans le modèle tout numérique. Pour rappel, le modèle tout numérique part de l'hypothèse que la fréquence instantanée de l'oscillateur contrôlé en tension ne varie pas au cours d'une demi-période. Cette hypothèse permet de rendre le modèle *event-driven* et de pouvoir le simuler aussi rapidement. En contrepartie, la fonction d'intégration continue du signal d'entrée du VCO est perdue. Il convient donc de caractériser l'impact de cette approximation et de

définir une plage de fonctionnement du modèle.

Pour cela, on va venir exacerber l'erreur commise par l'approximation réalisée dans le modèle tout numérique. En effet, l'erreur commise par l'approximation sera d'autant plus grande que le signal d'entrée varie vite par rapport à la période du VCO (vu que dans le modèle l'hypothèse faite revient à supposer que le signal d'entrée ne varie pas pendant une demi-période du VCO).

On va donc s'intéresser au ratio de fréquences $\frac{F_{min}}{F_{in}}$ où F_{min} représente la fréquence minimale d'oscillation du VCO et F_{in} représente la fréquence du signal d'entrée.

Pour mesurer l'erreur commise par l'approximation, on va alors simuler un VCO-based ADC le plus simple possible dans lequel le modèle de VCO utilisé est le modèle tout numérique. On va alors mesurer le rapport signal à bruit (SNR) de la conversion et on va comparer cette mesure à la même mesure mais pour un convertisseur dont le VCO est modélisé par un modèle traditionnel sans l'approximation faite dans le modèle tout numérique.

On pourra alors venir quantifier la différence entre les mesures des SNR des deux conversions et définir à partir de quel ratio $\frac{F_{min}}{F_{in}}$ le modèle tout numérique a un comportement qui diverge du comportement théorique d'un VCO.

Ces mesures sont reportées sur la figure IV.12. On y discerne assez distinctement une limite à partir de laquelle le comportement du modèle tout numérique diverge de celui du VCO modélisé selon ses équations mathématiques (avec intégration continue du signal d'entrée).

Pour garder une certaine marge de sécurité, on considèrera que le modèle tout numérique suit le comportement théorique d'un VCO (donc que l'approximation du modèle n'a qu'un faible impact sur la simulation) lorsque le ratio caractéristique $\frac{F_{min}}{F_{in}}$ est supérieur à 2. Pour un ratio caractéristique supérieur à 30, les deux modèles sont équivalents avec, là encore, le même gain en temps de simulation.

On peut donc conclure de cette étude que le modèle que l'on a mis au point, lorsqu'il est utilisé dans sa plage de fonctionnement ; à savoir lorsque la simulation est telle que $\frac{F_{min}}{F_{in}}$ est supérieur à 2, permet de simuler correctement le comportement d'un VCO tout en faisant gagner jusqu'à 92% de temps de simulation.

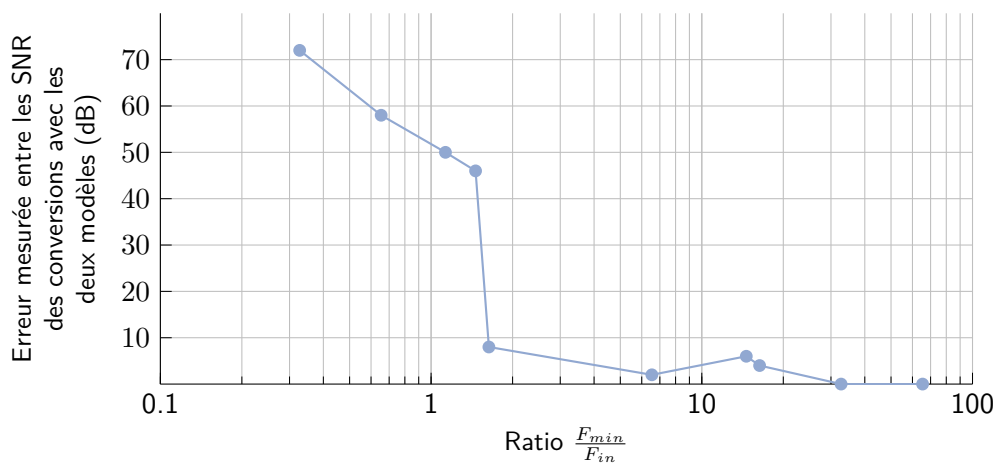


FIGURE IV.12 – Erreur entre le modèle tout numérique et le modèle standard.

Maintenant que l'on dispose d'un outil qui permet efficacement de simuler un VCO avec les contraintes requises pour un convertisseur numérique analogique, on va pouvoir expérimenter et développer la méthode d'extraction du résidu qui a été proposée plus haut dans ce chapitre tout en considérant des aspects plus concrets sur la faisabilité d'une telle méthode.

IV.3 Comment extraire le résidu en pratique

PRENONS la proposition expliquée dans la partie IV.1 concernant l'extraction du bruit de quantification (bruit qu'à partir de maintenant nous appellerons résidu). En nous aidant de la figure IV.2 que nous rappelons ici (sous le nom de figure IV.13) pour avoir les notations à portée, reprenons la conclusion du raisonnement qui a été mené dans la partie IV.1 ; à savoir que le résidu pouvait être extrait via l'équation suivante :

$$Res = \frac{A}{B} \times \pi \quad (IV.3)$$

avec :

Res le résidu que l'on cherche à extraire.

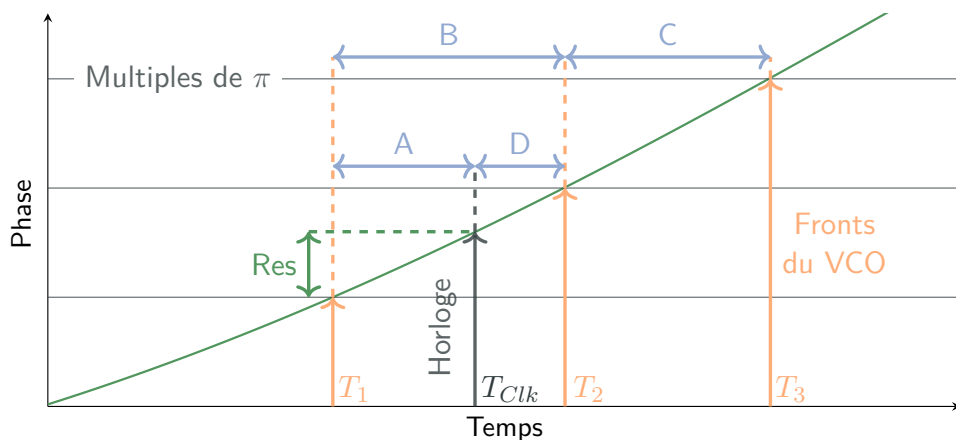


FIGURE IV.13 – Cause et apparition de l'erreur de quantification de la phase.

L'équation (IV.3) n'est vraie que si l'hypothèse qu'on a faite sur la phase ; c'est-à-dire qu'on peut la considérer linéaire entre deux franchissements de seuil, est vraie.

En se reportant à la figure IV.13, on constate que $A = B - D$. En remplaçant A par cette expression dans l'équation (IV.3), cette dernière devient :

$$Res = \left(1 - \frac{D}{B}\right) \times \pi \quad (IV.4)$$

On constate cependant que pour pouvoir extraire le résidu avec l'équation (IV.4), il est nécessaire de connaître l'instant T_1 qui correspond au dernier front du VCO (franchissement de seuil de la phase) avant le front d'horloge. Comme il est impossible de savoir quel front sera celui qui se produit juste avant le front d'horloge (on ne peut pas prédire le futur), cela implique de devoir enregistrer tous les fronts du VCO et de ne garder que celui qui se trouve effectivement être celui qui nous intéresse une fois que le front d'horloge se produit.

Autrement dit, cela force le système à récupérer tous les fronts du VCO (qui se produisent en grande quantité) pour en réalité ne garder qu'une très petite partie de cette information (qui est l'information utile). Cette solution n'est pas envisageable en pratique.

On propose alors de faire une seconde approximation en partant de l'observation suivante : dans les architectures de VCO-based ADC que l'on trouve dans l'état de l'art, le facteur de sur-échantillonnage est relativement élevé (25 dans [Ghosh et Pamarti, 2015] ou 22.5 dans [Park et Perrott, 2009] par exemple).

On fait alors l'approximation que la durée d'une période de VCO ne varie pas (ou très peu) d'une période à la période suivante. Pour reprendre les notations de

la figure IV.13, cette hypothèse revient à dire que :

$$B \approx C \quad (IV.5)$$

En réinjectant ce résultat dans l'équation (IV.4), on obtient alors :

$$Res \approx \left(1 - \frac{D}{C}\right) \times \pi \quad (IV.6)$$

Le gros avantage de cette nouvelle équation, c'est justement de ne plus avoir à enregistrer de manière systématique tous les fronts du VCO. En effet, dans l'équation (IV.6), on a besoin de connaître uniquement les créneaux C et D . Il est alors nécessaire d'enregistrer uniquement deux fronts du VCO (pour obtenir les deux créneaux) qui sont ceux qui se produisent après le front d'horloge.

Pour se référer à la méthode où l'on utilise l'équation (IV.6) pour extraire le résidu, on utilisera l'appellation *approximation « coup d'après »* par opposition à *l'extraction théorique du résidu* qui se sert de l'équation (IV.4).

Pour résumer, en mettant de côté les difficultés techniques liées à l'extraction (et la quantification) des durées des créneaux ainsi que celles liées à la division de ces durées, l'équation (IV.6) nous permet d'extraire le résidu pour venir compenser l'erreur de quantification commise sur la phase.

Cependant, ce résidu, pour compenser l'erreur de phase, doit être rajouté à la valeur quantifiée de la phase et subir les mêmes opérations que cette dernière. Or rappelons-le, le convertisseur étudié, issu du mode performant du convertisseur étudié au Chapitre III est un convertisseur rebouclé (une architecture de convertisseur $\Sigma\Delta$). Il faut conserver le *temps de boucle* (temps entre l'entrée de la donnée et la mise à jour de la boucle de rétroaction) le plus faible possible pour cette famille d'architectures. On ne peut donc pas attendre une période de VCO supplémentaire plus le temps de traitement pour calculer la valeur du résidu avant de permettre à la donnée de continuer son chemin.

Pour pallier à ce problème, on décide alors de traiter le résidu complètement indépendamment de la conversion qui a lieu dans la boucle du convertisseur. Pour ce faire on va, une fois la valeur du résidu calculée, l'ajouter au signal numérique directement issu du convertisseur en lui faisant subir un traitement numérique pour émuler le traitement que le résidu aurait subi s'il avait été ajouté à la phase et s'il avait été dans la boucle du convertisseur $\Sigma\Delta$. L'équation à laquelle ce bloc de traitement (qui prend en entrée le signal numérique issu du convertisseur et le résidu) obéit est la suivante :

$$\begin{aligned} Signal_{total} = & Signal_{num}(z^{-1}) + (2 \times Gain \times DAC_{step} - 1) \times Signal_{num}(z^{-2}) \\ & + Res(z) - Res(z^{-1}) \end{aligned} \quad (IV.7)$$

avec :

- $Signal_{total}$ le signal reconstruit supposé être le signal original numérisé ;
- $Signal_{num}$ le signal issu du convertisseur sans le résidu ;
- $Gain$ le gain de la chaîne direct du convertisseur soit $\frac{K_{VCO}}{F_s}$;
- DAC_{step} le pas du convertisseur numérique analogique de la boucle de rétroaction ;
- Res la valeur du résidu calculé comme expliqué précédemment.

Pour faciliter la visualisation des signaux évoqués ici et de leurs interactions, la figure IV.14 présente l'architecture du convertisseur qui résulte du raisonnement développé dans ce chapitre. Cette digression est nécessaire pour clarifier l'explication en cours.

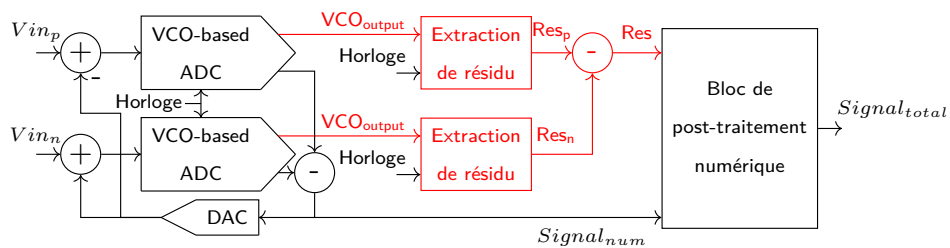


FIGURE IV.14 – Architecture du convertisseur résultant du raisonnement de ce chapitre.

Cette équation permet uniquement de pouvoir retrouver le signal de départ à partir des éléments dont nous disposons ; à savoir la valeur numérique issue du convertisseur et le résidu que l'on extrait selon les équations (IV.4) ou (IV.6) selon la méthode utilisée. C'est la seule fonction du bloc de traitement numérique que l'on vient d'évoquer et que l'on nommera par la suite (sans grande originalité) *Bloc de post-traitement numérique*.

On va donc pouvoir tester si l'extraction du résidu apporte bel et bien une amélioration de la résolution de la conversion comme escompté en simulant le convertisseur qui fait l'objet de notre étude ; à savoir le convertisseur du Chapitre III en mode performant dont le schéma-bloc est, rappelons-le, celui de la figure III.11 page 64.

On utilise pour cette simulation les modèles non-linéaires des VCO puis on vient artificiellement (avec MATLAB) récupérer les valeurs des durées des créneaux qui nous intéressent. On peut ensuite calculer la division de ces créneaux et faire

varier la résolution avec laquelle on obtient le résultat de cette division pour mettre en évidence l'impact que cela apporte sur la conversion. On fait enfin subir à la fois au signal numérique issu du convertisseur et au résidu extrait la fonction décrite par l'équation (IV.7) qui permet de retrouver le signal en entrée du convertisseur.

On calcule alors le SNDR de la conversion (à la fois avec la méthode d' *extraction théorique du résidu* et avec l' *approximation « coup d'après »*) sur différentes bandes-passantes avec différentes résolution de résidu. Les résultats de cette études sont consignés dans les figures IV.15a, IV.15b et IV.15c. NB : Les conversions avec un résidu de 0 bit de résolution correspondent à une conversion sans résidu.

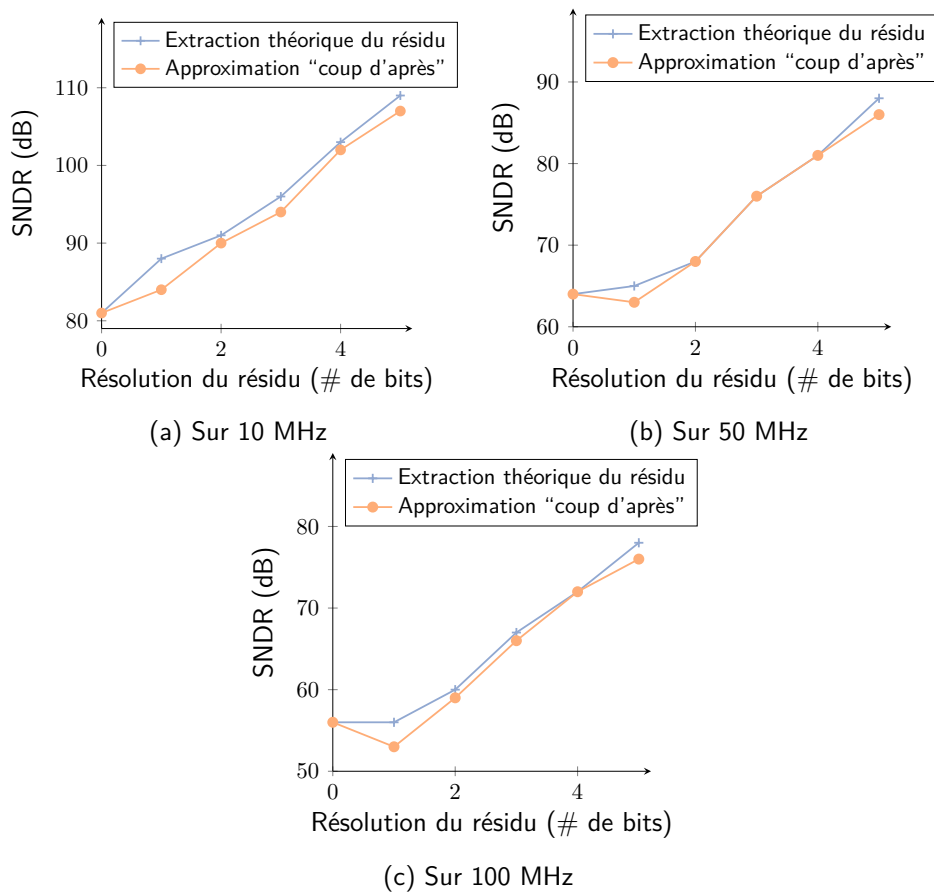


FIGURE IV.15 – Impact des approximations sur différentes bandes passantes.

Les simulations sont réalisées avec les paramètres suivants :

- Les VCO sont paramétrés pour simuler des anneaux de 31 cellules NAND standards en 28 nm FDSOI ;
- Le DAC de la boucle de rétroaction est considéré sur 5 bits ;
- Le signal d'entrée est une sinusoïde de fréquence 8 MHz ;
- La fréquence d'échantillonnage est de 3 GHz ;
- Le pas de simulation est de 1 ps.

Analysons ces figures pour définir ce que l'on peut retirer de cette étude. Il y a trois choses importantes à retenir de ces figures.

Tout d'abord, l'idée proposée d'extraire le résidu est une idée qui fonctionne. On peut en effet constater que peu importe la résolution choisie pour le résidu, l'extraction théorique du résidu offre une amélioration par rapport à la conversion sans résidu. En toute logique, plus le résidu est extrait précisément (avec une résolution élevée) plus la résolution de la conversion augmente. On peut noter par exemple qu'une extraction de résidu avec une résolution de 3 bits (ce qui reste raisonnable) permet un gain de 10 dB environ sur le SNDR de la conversion.

Le deuxième fait important que ces simulations mettent en évidence est que l'approximation du « coup d'après » qui rend conceptuellement beaucoup plus réalisable l'extraction du résidu est une approximation qui est valide. Les courbes entre le SNDR d'une conversion avec extraction théorique du résidu et celui d'une conversion avec l'approximation du « coup d'après » ne varient, au plus, que de quelques dB (3 dB dans les courbes présentées et au plus 5 dB dans les simulations effectuées au cours de ces travaux même si aucune vérification exhaustive n'a pu être effectuée). On retient donc pour la suite l'extraction du résidu avec l'approximation du « coup d'après » telle que décrite dans l'équation (IV.6).

Enfin, et c'est peut-être la moins attendue des observations, on note un point particulièrement étrange sur les courbes correspondant à l'extraction de résidu par l'approximation du « coup d'après » pour un résidu avec une résolution de 1 bit. Et ce point est tout particulièrement étrange sur les courbes des figures IV.15b et IV.15c. En effet, sur ces courbes, l'extraction du résidu par cette méthode empire la qualité de la conversion (SNDR inférieur à celui de la conversion sans résidu).

L'hypothèse la plus probable est que l'extraction du résidu via l'approximation du « coup d'après » ramenée sur 1 bit (qui vaut donc soit 0 soit 1) vient rajouter une erreur sur le signal. On se retrouve donc dans la situation où l'erreur liée à l'approximation faite pour le calcul du résidu, lorsqu'elle est quantifiée avec une si faible résolution, est plus néfaste que bénéfique à la conversion. Il va sans dire qu'il faut éviter de se retrouver dans cette situation.

On vient de montrer que la méthode d'extraction du résidu est une méthode intéressante pour améliorer la résolution du convertisseur analogique numérique dans son mode performant ; le rendant ainsi encore plus performant.

On a également montré que la méthode d'extraction de résidu via l'approximation du « coup d'après » permet de rendre conceptuellement plus réalisable l'extraction du résidu et que l'erreur amenée par cette approximation reste marginale.

Il reste donc maintenant un point critique à discuter : la faisabilité d'une telle extraction. Il faut trouver un moyen pratique (par opposition à une méthode mathématique via MATLAB) d'extraire la durée des créneaux et de réaliser la division de ces durées pour pouvoir mettre en oeuvre tout ce qui a été développé précédemment.

IV.4 Faisabilité de l'extraction de créneaux



N va donc dans cette partie venir présenter et discuter différentes techniques qui répondent au besoin d'extraction des durées des créneaux ainsi qu'à la problématique de division de ces durées.

IV.4.A Idée 1 : La méthode de calcul avec des créneaux du Chapitre II



La première idée qui vient à l'esprit pour calculer avec des données contenues dans la durée des créneaux est en toute logique celle d'utiliser la méthode proposée dans le Chapitre II qu'on nommera *méthode par numérisation de créneaux*. En effet, la méthode présentée dans ledit chapitre répond très exactement à la problématique rencontrée ici.

On entreprend alors d'étudier la possibilité d'utiliser cette méthode de calcul sur la durée des créneaux pour le besoin énoncé ici. Il s'agit en substance de numériser les différents créneaux qui interviennent dans le calcul du résidu à l'aide d'un VCO que l'on appellera *VCO res* pour le distinguer et de faire leur division.

En se référant au Chapitre II et plus particulièrement à l'équation (II.7) page 44 qui traite de la durée minimale détectable par la méthode de calcul sur les durées des créneaux, on va essayer de dériver les caractéristiques que le système d'extraction de créneaux devrait avoir pour pouvoir fonctionner.

On commence par exprimer la durée minimale détectable par la méthode en fonction des paramètres du circuit de calcul grâce aux équations (II.1), (II.2) et

(II.7). On a donc avec les notations du Chapitre II :

$$D_{cr\acute{e}neau}^{min} = \frac{1}{\left\lfloor \frac{2 \times F_{vco} \text{ res}_{max}}{F_{horloge}} \right\rfloor - \left\lfloor \frac{2 \times F_{vco} \text{ res}_{min}}{F_{horloge}} \right\rfloor} \times T_{horloge} \quad (IV.8)$$

Ce que l'on peut très grossièrement estimer (pour avoir un ordre de grandeur) par :

$$D_{cr\acute{e}neau}^{min} \approx \frac{1}{2 \times (F_{vco} \text{ res}_{max} - F_{vco} \text{ res}_{min})} \quad (IV.9)$$

D'autre part, considérons la durée des créneaux que nous avons à récupérer. En admettant qu'on utilise le VCO composé de 31 cellules NAND en technologie 28 nm FDSOI et que l'on considère « toutes les phases » du VCO ; c'est-à-dire les sorties de chacune des portes NAND, on cherche à estimer la durée des créneaux que l'ont veut extraire.

Ces créneaux ont une durée pouvant varier de 0 à la période instantanée du VCO (qui est virtuellement multipliée par 31 puisque l'on considère les sorties de chaque cellules standards NAND). Pour avoir une extraction qui dans le pire cas estime à zéro les créneaux dont la durée est inférieure à la période minimale du VCO, il faut pouvoir détecter des créneaux d'une durée supérieure à $\frac{T_{vco}^{min}}{31}$.

En reprenant les valeurs obtenues via la simulation électrique de la figure IV.9 page 79 soit une fréquence maximale de 1.38 GHZ, on obtient donc qu'on doit être en mesure de détecter un créneau d'une durée $D_{cr\acute{e}neau}^{min}{}_{application}$ telle que :

$$D_{cr\acute{e}neau}^{min}{}_{application} = \frac{1}{31 * 1.38 \times 10^9} \approx 23 \text{ ps} \quad (IV.10)$$

En réinjectant ce résultat dans l'équation (IV.9), on obtient :

$$F_{vco} \text{ res}_{max} - F_{vco} \text{ res}_{min} \approx \frac{1}{46 \times 10^{-12}} \quad (IV.11)$$

$$\approx 21.7 \text{ GHz} \quad (IV.12)$$

On obtient alors des spécifications complètement irréalisables (tout du moins à l'heure actuelle). La *méthode par numérisation de créneaux* du Chapitre II n'est donc pas la solution à envisager pour répondre à cette problématique.

IV.4.B Idée 2 : Quantifier les créneaux avec une horloge



A première proposition s'étant révélée infructueuse, on propose une deuxième idée pour quantifier la durée des créneaux.

On envisage de venir quantifier la longueur du créneau avec une horloge. L'idée est très simple, une horloge est activée au début du créneau et désactivée une fois

que le créneau repasse à son niveau logique bas. Il suffit alors de venir compter le nombre de fronts d'horloge obtenu pour avoir une grandeur numérique image de la durée du créneau.

Après l'étude de la partie IV.4.A, il n'est nul besoin de pousser davantage cette solution pour une raison évidente. En effet, si l'on veut pouvoir détecter un signal créneau dont la durée est de 23 ps, il faut une horloge dont la fréquence soit environ 43 GHz.

Inutile de dire que là encore la solution n'est absolument pas envisageable. Il nous faut donc trouver une autre idée.

IV.4.C Idée 3 : La méthode par charge de capacités



A troisième idée pour résoudre le problème de quantification et division des créneaux se démarque des deux précédentes puisqu'elle renonce à venir quantifier la durée de chaque créneau avant d'en faire la division.

À l'inverse, on propose ici de procéder à la division avant de procéder à la quantification du résultat de cette division. On propose pour cela de procéder par une méthode quelque peu détournée qui repose sur un convertisseur analogique numérique de type flash.

L'idée sous-jacente est la suivante. Un convertisseur analogique numérique (de type flash en l'occurrence) fonctionne en comparant une tension d'entrée à une multitude de valeurs qui sont des subdivisions d'une tension de référence. Si l'on parvient à placer en entrée V_{in} d'un tel ADC une tension image de la durée de l'un des créneaux et en tension de référence V_{ref} une tension image de la durée du second créneau puis que l'on se débrouille pour présenter, en lieu des différentes valeurs de tension des fractions de la tension de référence (V_{ref} , $0.75 \times V_{ref}$, $0.5 \times V_{ref}$ et $0.25 \times V_{ref}$ par exemple). On peut, en théorie, obtenir comme résultat de conversion un code thermomètre qui correspond à la numérisation du rapport $\frac{V_{in}}{V_{ref}}$.

On propose alors l'architecture décrite dans la figure IV.16 pour mettre en œuvre cette idée.

Dans un premier temps, les créneaux (plus précisément leurs durées) sont convertis en tension. On utilise pour cela une méthode de conversion temps vers tension semblable à celle utilisée par G.W. Roberts dans sa proposition de calculs avec les créneaux détaillée dans la partie I.2.A page 22. On se sert donc des créneaux comme de signaux de contrôles pour charger deux capacités (une pour chaque créneau) à courant constant. On obtient ainsi des tensions images de la

durée des créneaux.

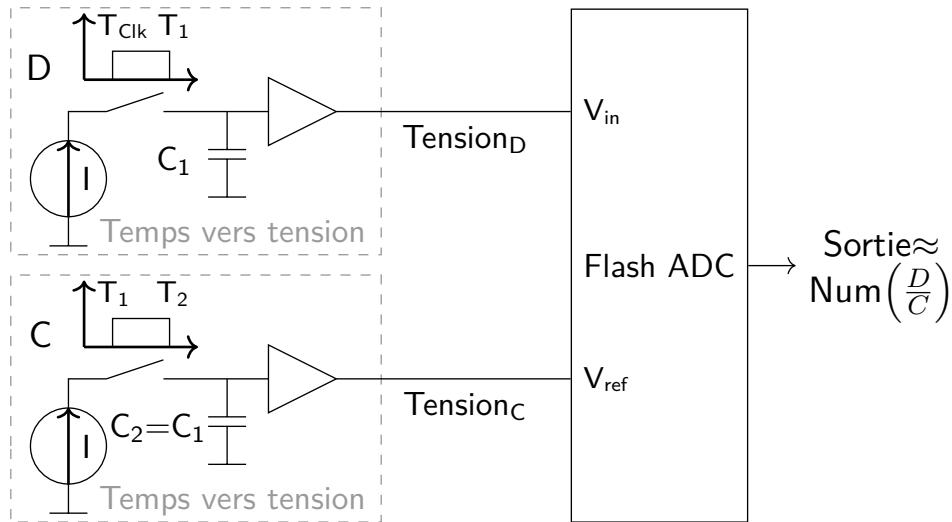


FIGURE IV.16 – Méthode de calcul de résidu par capacités.

On place ensuite ces tensions aux bornes du convertisseur analogique numérique de type flash avec $Tension_D$ et $Tension_C$ (les images des créneaux D et C) respectivement sur V_{in} et V_{ref} .

Le signal en sortie de ce circuit de conversion sera une valeur numérique image du rapport $\frac{D}{C}$.

Cette méthode présente de nombreux avantages. Premièrement, les créneaux ne sont pas quantifiés indépendamment ce qui a deux effets :

- on ne rencontre plus la difficulté précédente de devoir être plus rapide que le créneau que l'on vient quantifier puisqu'on vient quantifier un signal image du rapport des durées ;
- on peut choisir directement la résolution avec laquelle on vient quantifier le résidu. Rappelons-le, le résidu que l'on cherche à quantifier est le rapport des deux durées. Cette méthode permet donc de définir la résolution avec laquelle on quantifie le résidu directement au travers de la résolution du convertisseur flash.

En outre, cette méthode offre aux comparateurs du convertisseur flash quasiment une période d'horloge entière pour prendre une décision. Très exactement, le temps disponible pour décider de la valeur de la comparaison est d'une période d'horloge moins le plus long des deux créneaux en jeu. Or, rappelons-le, les VCO-based ADC fonctionnent avec un taux de sur-échantillonnage relativement élevé ce qui implique qu'une période d'horloge est beaucoup plus grande que la

période du VCO (qui est la borne supérieure de la durée que peut avoir un créneau).


Cette méthode n'est malheureusement pas exempte de défaut. L'intégralité du processus repose sur la création d'un signal de tension image de la durée du créneau par le biais d'une capacité. Les déviations du processus de fabrication des circuits sur silicium rendant quasiment impossible la réalisation de deux capacités exactement de même valeur (c'est ce que l'on appelle le *mismatch*), la méthode précédente va venir introduire une erreur systématique dans la pente d'intégration entre les différents créneaux.

On pourrait bien sûr envisager de mesurer ce *mismatch* en phase de calibration puis de venir le corriger avec un bloc adapté à cette fonction mais cela complexifierait considérablement le circuit d'extraction qui ne l'oublions pas n'est là que pour extraire un résidu.

Nous aurions voulu pousser davantage le dimensionnement de ce dispositif. Dans un premier temps régler la valeur des capacités en fonction de la période du VCO pour rendre la dynamique des signaux $Tension_D$ et $Tension_C$ compatibles avec la dynamique des tensions d'alimentation. Puis une fois la valeur nominale des capacités déterminée, essayer de prédire, par le biais d'outils statistiques, quel serait le *matching* et l'effet qu'il induit sur l'extraction du résidu. Malheureusement, les contraintes de temps nous font parfois dévier de nos objectifs.

Nous restons cependant optimistes étant données les faibles résolutions que nous aspirons à atteindre sur le résidu (3 ou 4 bits de résolution).

IV.5 Résultats et limitations

 OUR résumer, nous avons dans ce chapitre parcouru le raisonnement qui nous a mené à proposer une amélioration de l'architecture d'un VCO-based ADC différentiel rebouclé en $\Sigma\Delta$. En combinant toutes les étapes précédentes, on aboutit à l'architecture présentée dans la figure IV.17. On reconnaît le convertisseur du mode performant du Chapitre III. Et on peut visualiser plus aisément où se situe la partie extraction du résidu.

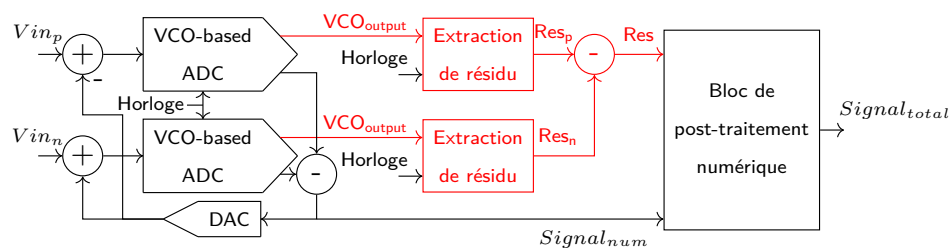


FIGURE IV.17 – Architecture de l'ADC avec extraction du résidu.

Quelques ombres restent cependant au tableau à commencer par le retour dans le domaine de l'amplitude au moment de quantifier le résidu. L'idéal aurait bien sûr été de pouvoir s'affranchir complètement de toute représentation dans le domaine de l'amplitude. Mais nous n'y sommes hélas pas parvenus. C'est d'ailleurs la principale piste d'amélioration de ce travail.

En revanche, avec cette architecture, les contraintes du domaines de l'amplitude sont reportées sur le traitement d'un résidu (et non plus du signal en lui-même) ce qui réduit l'effet des erreurs commises.

Il reste également en suspens la question du *matching* des capacités dans le circuit d'extraction du résidu. C'est un problème complexe qui mériterait d'être investigué. Une approche a d'ailleurs été entamée dans le modèle Matlab donné en annexes page 117 mais n'a pas pu être complètement abouti faute de temps.

Cela dit, le schéma proposé dans la figure IV.17 a tout de même été simulé (à la fois en MATLAB et en VerilogAMS) avec des capacités de même valeurs pour l'extraction du résidu avec une résolution de 3 ou 4 bits.

Les résultats, bien que n'étant que des résultats de simulation, ont ensuite été reportés vis-à-vis de l'état de l'art. C'est ce qui est présenté dans la figure IV.18.

Ces résultats, bien qu'issus de simulations, sont très encourageants quant aux possibilités offertes par l'architecture en présentant un gain de 10 à 15 dB en fonction des paramètres choisis.

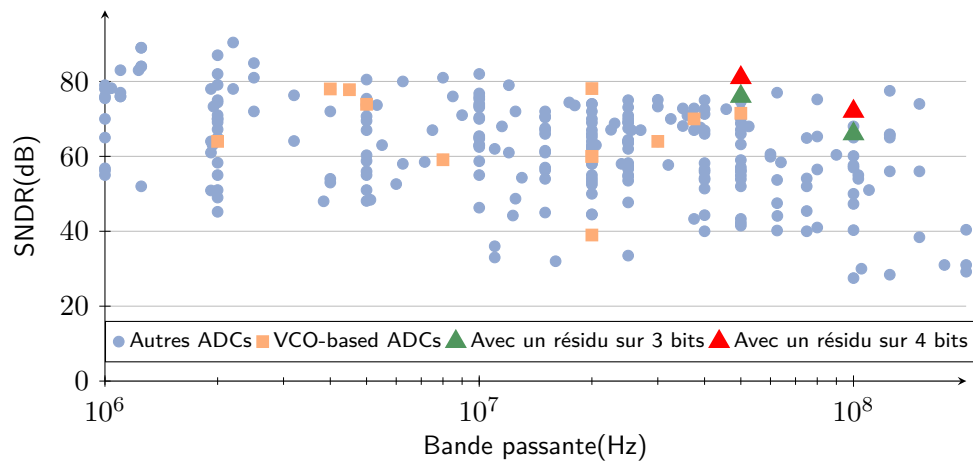


FIGURE IV.18 – Positionnement de notre simulation par rapport à l'état de l'art.


Les modèles VerilogAMS qui constituent le cœur de l'architecture du VCO-based ADC avec extraction de résidu sont fournis en annexes de ce manuscrit page 114 pour le lecteur curieux qui voudrait approfondir le travail présenté ici ainsi qu'un fichier MATLAB permettant de simuler l'architecture complète avec une multitude d'options. Ce dernier fichier étant le résultat de deux années d'itérations et d'expérimentations, il va sans dire qu'il n'est pas aussi propre que les fichiers de modèles qui composent le reste des annexes de cet ouvrage.

Chapitre V

Conclusion et Perspectives




« Un monde de perspectives. » Crédits : Guyhème Buffeteau

 OUR clôturer ce manuscrit, nous allons procéder à un récapitulatif de ce qui a été abordé et des aboutissements de chacun des points étudiés. Dans un second temps, nous ouvrirons l'éventail des possibilités pour permettre à quiconque souhaiterait poursuivre ce travail d'avoir des pistes d'améliorations à explorer.

Le premier chapitre de ce manuscrit peut déjà servir de base de travail en constituant un passage en revue des différentes options offertes en termes de représentation et traitement dans le domaine temporel mais aussi parce qu'il permet un aperçu de ce que l'on peut rencontrer en termes de convertisseurs reconfigurables.

Les annexes, quand à elles, offrent un bon point de départ en termes d'outils pour l'intrépide qui souhaiterait s'atteler à la poursuite de ce travail. Le modèle tout numérique de VCO pourra, lui, servir à tous ceux qui auraient besoin de simuler rapidement un tel bloc.


V.1 Calcul dans le domaine temporel

 UR le sujet du calcul dans le domaine temporel, il a été étudié comment un signal du domaine temporel pouvait être numérisé (et ainsi mis en forme pour permettre des calculs complexes) en s'affranchissant du domaine de l'amplitude et des contraintes apportées par ce domaine (la ligne directrice de ce travail de thèse).

Il a ainsi pu être montré qu'un oscillateur contrôlé en tension pouvait servir de brique de base pour numériser un signal en créneau dont la largeur servait à coder l'information.

Cette approche reste cependant bien trop contraignante à l'heure actuelle mais se révélera peut-être intéressante si une application appropriée fait son apparition.

V.2 L'ADC hybride répondant aux systèmes à deux modes de fonctionnement

 A partie qui met en place l'architecture hybride qui permet d'obtenir un convertisseur analogique numérique qui peut adapter sa consommation et sa performance en fonction de la nécessité; ou plus exactement l'idée d'utiliser un VCO-based ADC pour parvenir à cette fonction constitue vraisemblablement la contribution de ce manuscrit qui a le plus de potentiel futur.

La réalisation qui est effectuée ici n'est certes qu'un préliminaire à ce qui peut être réalisé par la suite mais le concept de pouvoir concevoir un convertisseur analogique numérique, qui en utilisant les mêmes blocs de base (des VCO) permet d'adresser une reconfigurabilité allant d'un fonctionnement asynchrone à un convertisseur $\Sigma\Delta$ de résolution relativement correcte, ouvre réellement le spectre des possibles.

L'idée proposée à la fin de cette étude qui consiste à pouvoir adapter quasiment de façon continue la bande passante du convertisseur et sa consommation en faisant varier la fréquence centrale des deux VCO offre elle aussi un éventails de possibilités à explorer.

V.3 Le VCO-based ADC avec résidu pour améliorer la résolution



A proposition d'extraction du résidu au sein du convertisseur analogique numérique basé sur des oscillateurs contrôlés en tension offre, elle, le travail le plus abouti de cette thèse.

Bien que l'étude n'ait pas été menée jusqu'à la phase de layout et de prototype silicium, elle reste tout de même prometteuse. Les résultats obtenus par simulations (qui prennent en compte un certain nombre de non-idéalités) laissent présager d'un réel intérêt pour ce type d'architectures et il y a fort à parier que de nombreux travaux d'améliorations continueront de voir le jour en plus de ceux existants (et dont ce manuscrit fait partie).

L'intérêt des ADC opérant dans le domaine temporel (pas uniquement les VCO-based ADC) étant renforcé par la nécessité de développer des convertisseurs qui ne sont plus tributaires du domaine de l'amplitude et des contraintes liées à la réduction des tensions d'alimentations.

Pour finir, si les tendances se confirment en termes d'évolution des technologies, il est fort probable que les oscillateurs contrôlés en tension (sur lesquels repose l'architecture étudiée dans ce travail ainsi que ses performances) offrent de meilleures caractéristiques et permettent la conception de VCO-based ADC encore meilleurs que ceux existants.

V.4 Perspectives



I, d'aventure, une personne ou un groupe de personnes voulaient reprendre ce travail là où je l'ai laissé, voici quelques pistes d'amélioration que je leur

suggère ou quelques point que je regrette de n'avoir pu traiter moi-même.

Premièrement, et cela semble relativement évident, il faudrait finaliser le développement du convertisseur basé sur les VCO qui utilise le principe d'extraction du résidu.

Pour cela je proposerais de prendre la valeur de capacité maximale qui permet de placer les signaux images des créneaux intervenant dans le calcul du résidu. Il faudrait alors procéder au layout pour avoir des mesures silicium des performances de l'architecture et pouvoir vérifier si l'état de l'art est bel et bien dépassé.

Une seconde piste d'amélioration serait de poursuivre le développement de l'architecture hybride de convertisseur pour permettre le réglage fin des fréquences centrales des VCO en fonction du signal d'entrée à convertir.

Dans cette veine on pourrait envisager de considérer différents paramètres au sein du bloc de décision pour un contrôle plus fin du convertisseur.

Mais de manière un peu plus globale, en prenant un peu de recul, ce sont toutes les interactions entre le monde synchrone et le monde asynchrone que j'encourage à explorer. L'électronique asynchrone, bien que plus délicate à concevoir et caractériser que son homologue synchrone, a fait ses preuves et dans ses domaines de prédilection son efficacité a été démontrée.

Dans toutes les applications mettant en jeu des signaux irréguliers et de faible activité, la solution asynchrone peut valoir le coup d'être étudiée et pour des applications qui doivent couvrir une large plage de fonctionnement, une solution hybride entre synchrone et asynchrone peut être souhaitable.

Il reste enfin la question du traitement de l'information à proprement parler dans le domaine temporel. Serait-il envisageable de réaliser une unité de traitement numérique (DSP) qui opère entièrement dans le domaine temporel? Un DSP qui n'aurait pas besoin de convertir les données mais qui traiterait directement l'information temporelle et qui permettrait de réaliser des calculs complexes.

Je pense que cette question pourrait légitimement constituer le sujet d'une à plusieurs autres thèses. Peut-être qu'elle trouvera une réponse à l'avenir ...

Annexes

Modèles MATLAB utilisés dans le Chapitre II

Modèle de simulation et de comparaison pour le convertisseur du chapitre II

```
%Le but est de simuler le calcul sur les créneaux du Chapitre II. Pour cela
%on va générer un sinus, le coder sous la forme de créneaux de largeur
%variable et ensuite le convertir en mots numériques avec la méthode
%proposée au Chapitre II.
% On analysera ensuite le SNDR de la onversion pour le comparer au SNDR
% théorique et ainsi mettre en évidence le bruit de quantification du
% VCO-based ADC.

%% Génération du sinal sinusoidal

F_in=9e6; % La fréquence du signal d'entrée
Fs=20e6; % La fréquence de l'horloge
Fstep=20e9; % La fréquence de simulation
Nb_pts_per=Fstep/Fs; % Nombre de points par période d'horloge
P2=15;
Nb_useful_pts=2^P2+1; % Le nombre de points sur lequel on étudie la conversion
k=floor((Nb_useful_pts-1)*F_in/Fs);
F_in=Fs/(Nb_useful_pts/k); % La fréquence d'entrée modifiée pour avoir un nombre
entier de périodes du signal

BW=9e6; % La bande passante sur laquelle on calcule le SNDR

T_step=1/Fstep;
t_simu=T_step:T_step:(((Nb_useful_pts+3)*Nb_pts_per)*T_step);

V_input=0.5+0.5*sin(2*pi*F_in*t_simu); %Le signal sinusoidal d'entrée ramené entre
0 et 1

%% Echantillonnage de l'entrée
V_input_sampled=V_input(1:Nb_pts_per:end);

figure;
plot(t_simu(1:Nb_pts_per:end),V_input_sampled);
title('Signal échantillonné');

%% Génération du signal en créneau
V_input_creneau_len=round(Nb_pts_per*V_input_sampled); %Calcul de la longueur des
créneaux arrondie au pas de simulation
V_input_creneau=zeros(1,Nb_pts_per*Nb_useful_pts);
```

```

for i=1:1:(length(V_input_creneau_len))
    V_input_creneau(((i-1)*Nb_pts_per+1):((i-1)*Nb_pts_per+1+V_input_creneau_len(i)))=1;
end

%% Passage dans le VCO
F_min=100e6;
F_max=200e6;
K_vco=F_max-F_min;

VCO_output=VCO_lin_calc_cren( V_input_creneau, F_min,K_vco, T_step);

figure;
plot(t_simu(1:length(VCO_output)),VCO_output);
title('Sortie du VCO');

%% Comptage des fronts montant et descendants
Num_output=zeros(1,Nb_useful_pts);
for i=1:1:(Nb_useful_pts)
    Num_output(i)=length(find(diff(VCO_output(((i-1)*Nb_pts_per+1):(i*Nb_pts_per+1)))));
end

figure;
plot(t_simu(1:Nb_pts_per:Nb_pts_per*(Nb_useful_pts)),Num_output);
title('Sortie du compteur');

%% Comparaison spectre du signal d'entrée et spectre du signal de sortie
plot_over_BW_base(V_input_sampled(1:Nb_useful_pts),Fs,0,BW,1);
title('DSP du signal en entrée');

plot_over_BW_base(Num_output(1:end),Fs,0,BW,1);
title('DSP du signal numérisé');

N_bit_theorique=3.322; % Correspond à l'exemple du manuscrit avec 10 niveaux disponibles

SNDR_theorique=6.02*N_bit_theorique+1.76; % Correspond dans l'exemple à 21.8 dB

SNDR=SNDR_over_BW(Num_output,Fs,F_in,[1e3 BW]);

```

Modèle de VCO utilisé aux Chapitres II et IV

```
function [ VCO_output ] = VCO_lin_calc_cren( V_input, F_min, K_vco, T_step )
% Un modèle de VCO simpliste et linéaire
% V_input est la tension de contrôle entre 0 et 1
% F_min est la fréquence minimale pour V_input = 0
% K_vco est le gain constant du VCO. F_max = F_min+K_vco
% T_step est le pas de simulation utilisé pour V_input

Phase=zeros(1,length(V_input));
VCO_output=ones(1,length(V_input));

Phase(1)=2*pi*(K_vco*V_input(1)+F_min)*T_step; % On calcule la valeur de la Phase
initiale

if (Phase(1) >= pi)
    VCO_output(1)=0; % On détecte ou non le franchissement du seuil par la phase
end

Phase(1)=mod(Phase(1),pi); % On ramène la Phase entre 0 et pi

for i=2:length(V_input)
    Phase(i)=Phase(i-1)+2*pi*(K_vco*V_input(i)+F_min)*T_step; % On calcule la nouvelle
Phase

    if (Phase(i) >= pi)
        VCO_output(i)= VCO_output(i-1); % On recommence la détection de seuil
    else
        VCO_output(i)=VCO_output(i-1);
    end

    Phase(i)=mod(Phase(i),pi); % On borne de nouveau la Phase
end
```

Modèles utilisés dans le Chapitre III

Modèle de conversion par traitement des trains d'impulsions

```
% Script ayant pour but de simuler le fonctionnement de l'ADC hybride en
% mode degrade. Cela correspond au VCO-based ADC puis au traitement de
% train d'impulsion par un filtre.

F_in=10e6; % Frequence de la sinusoide en entree
Fstep=50e11; % Pas de la simulation
P2=23;
Nb_useful_pts=2^P2+1;
k=floor((Nb_useful_pts-1)*F_in/Fstep);
F_in=Fstep/(Nb_useful_pts/k)% Modification de la frequence pour les calculs de DSP

BW=50e6; % Bande passante sur laquelle les calculs de SNDR sont faits
T_step=1/Fstep;

K_vco_a=-5.6534e9;
K_vco_b=10.518e9;
K_vco_c=-3.5191e9; % Caracteristiques du VCO

t_simu=T_step:T_step:((Nb_useful_pts+50000)*T_step);

V_input=sin(2*pi*F_in*t_simu); % Signal sinus en entree

figure;
plot(t_simu,V_input);
title('Voltage input');

VCO_output=VCO_manuscrit( V_input, K_vco_c,K_vco_a, K_vco_b, T_step); % Le signal
passe dans le VCO

figure;
plot(t_simu,VCO_output);
title('VCO output');

V_pulse=Pulse_generator( VCO_output,3e-10, T_step ); % On derive la sortie du VCO
pour generer les impulsions

figure;
plot(t_simu,V_pulse);
title('Pulse output');
```

```

%% Filtrage
Fc=BW;
omegab = 2*pi*Fc;
filter_choice=2; % On definit le filtre utilise. On peut essayer des filtres
d'ordres differents

switch filter_choice
case 1
    num=[omegab];
    den=[1 omegab];
    %H_Butterworth = (omegab)./(p + (omegab));
case 2
    num=[omegab^2];
    den=[1 (1.4142)*(omegab) (omegab^2)];
    %H_Butterworth = (omegab^2)./(p.^2 + (1.4142)*(omegab)*p + (omegab^2));
case 3
    num=[omegab^3];
    den=[1 (2)*(omegab) (2)*(omegab^2) (omegab^3)];
    %H_Butterworth = (omegab^3)./(p.^3 + (2)*(omegab)*p.^2 + (2)*(omegab^2)*p +
(omegab^3));
case 4
    a = 0.7654; b = 1.8478;
    num=[omegab^4];
    den=[1 (a+b)*(omegab) (a*b+2)*(omegab^2) (a+b)*(omegab^3) (omegab^4)];
    %H_Butterworth = (omegab^4)./(p.^4 + (a+b)*(omegab)*p.^3 + (a*b+2)
(omegab^2)*p.^2 + (a+b)*(omegab^3)*p + (omegab^4));
case 5
    a = 0.6180; b = 1.6180;
    num=[omegab^5];
    den=[1 (a+b+1)*(omegab) (a*b+2+a+b)*(omegab^2) (a+b+a*b+2)*(omegab^3) (a+b+1)*(omegab^4)
(omegab^5)];
    %H_Butterworth = (omegab^5)./(p.^5 + (a+b+1)*(omegab)*(p.^4)+ (a*b+2+a+b)
(omegab^2)*(p.^3) + (a+b+a*b+2)*(omegab^3)*(p.^2) +
%(a+b+1)*(omegab.^4)*p+ (omegab^5));

end

TL_FTBF_VCO= tf(num,den);
TZ_FTBF_VCO= c2d(TL_FTBF_VCO,T_step);
[num,den] = tfdata(TZ_FTBF_VCO); % On cree le filtre
filtered_signal=filter(num{1,1},den{1,1},V_pulse); % On procede au filtrage

```

```

figure; plot(filtered_signal);
title('Filter output');

start=find(diff(filtered_signal)<0,1,'first')
filtered_signal=filtered_signal(start:(start+Nb_useful_pts-1)); % On coupe le signal
pour supprimer les effets transitoires de debut de signal

plot_over_BW_base(filtered_signal,Fstep,0,BW,1);
title('Filtered output no Blackman');

SNDR_nB=SNDR_over_BW(filtered_signal,Fstep,F_in,[1e6 BW])
SNDR=SNDR_over_BW_extended5(blackman(length(filtered_signal),'periodic')'
.*filtered_signal,Fstep,F_in,[1e6 BW])
% On calcule le SNDR de la conversion sur la bande BW. Version avec et sans
% fenetre d'apodisation

```


Modèles utilisés dans le Chapitre IV

Modèle tout numérique de VCO linéaire VerilogAMS

```
'include "constants.vams"
'include "disciplines.vams"
'timescale 1s/1fs
module VCO_lin(V_ctrl, V_in,V_out);

    input V_in; // On reboucle V_out sur V_in
    input V_ctrl; // La tension de controle du VCO
    output V_out; // La sortie en creneau du VCO
    reg    temp; // Registre interne contenant la valeur de V_out

    electrical V_ctrl;

    real    timing; // Le temps d'attente d'une demi-periode
    real    F_instant; // La frequence instantanee du VCO

    real    used_V;

    parameter real K_vco=1e9;
    parameter real F_cent=2e9;

    parameter real V_min=0.5;
    parameter real V_max=1;
    // Min et max des tensions sont definis pour eviter des grandeurs qui dépassent
    // la plage de fonctionnement du VCO

    initial begin
        temp<=1;
    end

    always @(posedge V_in or negedge V_in)
        begin

            if (V(V_ctrl) < -V_max)
                begin
                    used_V=-V_max;
                end
            else if (V(V_ctrl) > V_max)
                begin
                    used_V=V_max;
                end
            else
                begin
                    used_V=V(V_ctrl);
                end

            F_instant=(K_vco*used_V+F_cent);

            timing=(1/(2*F_instant)); // La duree de la demi-periode

            #timing temp <= ~V_in;
        end

    assign V_out=temp;

endmodule
```

Modèle MATLAB de VCO non-linéaire

```
function [ VCO_output ] = VCO_manuscrit( V_input, K_vco_c, K_vco_a,K_vco_b, T_step
)
% Un modèle de VCO prenant en compte la non-linéarité
% V_input est la tension de contrôle entre 0.5 et 1
% K_vco_a, K_vco_b et K_vco_c sont les coefficients de la caractéristique tension/fréquence.

% On a donc Freq=K_vco_a*V_input2+K_vco_b*V_input+K_vco_c
% T_step est le pas de simulation utilisé pour V_input

Phase=zeros(1,length(V_input));
VCO_output=ones(1,length(V_input));

Phase(1)=2*pi*(K_vco_a*V_input(1)*V_input(1)+K_vco_b*V_input(1)+K_vco_c)*T_step;

if (Phase(1) >= pi)
    VCO_output(1)=0;
end

Phase(1)=mod(Phase(1),pi);

for i=2:length(V_input)
    Phase(i)=Phase(i-1)+2*pi*(K_vco_a*V_input(i)*V_input(i)+K_vco_b*V_input(i)+K_vco_c)*T_step;

    if (Phase(i) >= pi)
        VCO_output(i)= VCO_output(i-1);
    else
        VCO_output(i)=VCO_output(i-1);
    end

    Phase(i)=mod(Phase(i),pi);

end
```

Outil de test du VCO non-linéaire MATLAB

```
% Script de test permettant de vérifier le fonctionnement du modèle
% de VCO non-linéaire
% On observe le fonctionnement du modèle au pont de fonctionnement
% V_fonc_test au travers des tracés des différentes courbes et on peut
% vérifier aisément la fréquence de fonctionnement du modèle via le tracé
% de la PSD du signal de sortie.

V_fonc_test=0.7; % Le point de fonctionnement à tester

K_vco_a=-5.6534e9;
K_vco_b=10.518e9;
K_vco_c=-3.5191e9; % Les caractéristiques de la courbe du VCO

F_fonc_th=K_vco_a*V_fonc_test^2+K_vco_b*V_fonc_test+K_vco_c % La fréquence à laquelle
le VCO doit fonctionner d'après la courbe mathématique
Fstep=200e9; % La fréquence de simulation (très élevée pour atténuer le repliement
du signal crête en sortie)
P2=15;
Nb_useful_pts=2^P2+1; % Le nombre de points sur lequel on étudie l'oscillateur
k=floor((Nb_useful_pts-1)*F_fonc_th/Fstep);
Fstep=F_fonc_th*(Nb_useful_pts/k) % La fréquence de la simulation

BW=min(3*F_fonc_th,Fstep/2); % La bande sur laquelle on tracera la PSD du signal
de sortie

V_input=V_fonc_test*ones(1,Nb_useful_pts);
figure;
plot(V_input);
title('V input'); % On vérifie que le signal d'entrée est correct

V_output= VCO_manuscrit( V_input, K_vco_c, K_vco_a,K_vco_b, T_step );
figure;
plot(V_output);
title('V output'); % On observe le signal crête en sortie

plot_over_BW_base(V_output(1:end),Fstep,0,BW,1);
title('DSP de la sortie du VCO'); % La PSD permet de voir rapidement la fréquence
de fonctionnement du convertisseur
```

Modèle de VCO non-linéaire VerilogAMS

```
'include "constants.vams"
'include "disciplines.vams"
'timescale 1s/1fs
module VCO_non_lin(V_ctrl, V_in,V_out);

    input V_in; // On reboucle V_out sur V_in
    input V_ctrl; // La tension de controle du VCO
    output V_out; // La sortie en creneau du VCO
    reg    temp; // Registre interne contenant la valeur de V_out

    electrical V_ctrl;

    real    timing; // Le temps d'attente d'une demi-periode
    real    F_instant; // La frequence instantanee du VCO

    real    used_V;

    parameter real K_vco_a=-5.6534e9;
    parameter real K_vco_b=10.518e9;
    parameter real K_vco_c=-3.5191e9;

    parameter real V_min=0.5;
    parameter real V_max=1;
    // Min et max des tensions sont definis pour eviter des grandeurs qui dépassent
    // la plage de fonctionnement du VCO

    initial begin
        temp<=1;
    end

    always @(posedge V_in or negedge V_in)
        begin

            if (V(V_ctrl) < -V_max)
                begin
                    used_V=-V_max;
                end
            else if (V(V_ctrl) > V_max)
                begin
                    used_V=V_max;
                end
            else
                begin
                    used_V=V(V_ctrl);
                end

            F_instant=(K_vco_a*used_V*used_V+K_vco_b*used_V+K_vco_c);

            timing=(1/(2*F_instant)); // La duree de la demi-periode

            #timing temp <= ~V_in;
        end

    assign V_out=temp;

endmodule
```

Modèle de VCO non-linéaire VerilogAMS avec bruit de phase

```
'include "constants.vams"
'include "disciplines.vams"
'timescale 1s/1fs
module VCO_nlin_bphase(V_ctrl, V_in,V_out);

    input V_in; // On reboucle V_out sur V_in
    input V_ctrl; // La tension de controle du VCO
    output V_out; // La sortie en creneau du VCO
    reg    temp; // Registre interne contenant la valeur de V_out

    electrical V_ctrl;

    real    timing; // Le temps d'attente d'une demi-periode
    real    F_instant; // La frequence instantanee du VCO

    real    used_V;
    integer jitter_fs; // Le jitter de periode en fs
    real    jitter_time;

    parameter real K_vco_a=-5.6534e9;
    parameter real K_vco_b=10.518e9;
    parameter real K_vco_c=-3.5191e9;

    parameter real V_min=0.5;
    parameter real V_max=1;
    // Min et max des tensions sont definis pour eviter des grandeurs qui dépassent
    // la plage de fonctionnement du VCO

    parameter real std_dev_fs=15000; // L'ecart-type de la loi normale pour la
    //generation du bruit de phase en fs

    integer    seed=286; // Initialisation pour la generation aleatoire
    reg        per; // registre permettant de definir le debut d'une periode

    initial begin
        temp<=1;
        per <=1;
    end

end
```

```

always @(posedge V_in or negedge V_in)
begin
    if (V(V_ctrl) < -V_max)
        used_V=-V_max;
    else if (V(V_ctrl) > V_max)
        used_V=V_max;
    else
        used_V=V(V_ctrl);

    F_instant=(K_vco_a*used_V*used_V+K_vco_b*used_V+K_vco_c);

    if (~per) // On ajoute le jitter une fois sur deux
    begin
        jitter_fs=$dist_normal(seed,0,std_dev_fs);
        jitter_time=jitter_fs*1e-15;
        per <=1;
    end
    else
    begin
        jitter_time=0;
        per<=0;
    end // else: !if(~per)

    timing=(1/(2*F_instant))+jitter_time; //La duree d'une demi-periode

    #timing temp <= ~V_in;

end

assign V_out=temp;

endmodule

```

Modèles VerilogAMS constituant le convertisseur du Chapitre IV

```
'include "constants.vams"
'include "disciplines.vams"
'define NB_PHASE 31

// Ce module prend en entree les sorties des VCO ainsi que l'horloge et
// calcule le resultat de la conversion

module Inc_dec_bloc_nb_phase(V_in_p,V_in_m,Clk,D_out);
  input [1:'NB_PHASE] V_in_p;
  input [1:'NB_PHASE] V_in_m;
  input Clk;
  output wreal D_out;

  real tmp;
  real pre_out;

  initial begin
    tmp=0;
    pre_out =0;
  end

  always @(V_in_p) begin
    tmp <= tmp+1;
  end

  always @(V_in_m) begin
    tmp <= tmp-1;
  end

  always @(posedge Clk) begin
    pre_out <= pre_out+tmp;
    tmp<=0;
  end

  assign D_out=pre_out;
endmodule // Inc_dec_bloc_nb_phase
```

```

'include "constants.vams"
'include "disciplines.vams"
'define NB_PHASE 31

// Ce module prend en entree la sortie d'un VCO ainsi que l'horloge et
// rise l'extraction du residu par l'approximation du coup d'apres

module Res_cd_nb_phase(VCO_in,Clk,Res_quant );

    input [1:'NB_PHASE] VCO_in;
    input Clk;
    output wreal Res_quant;
    reg Res_in_process;
    reg Res_needed;
    real Res_quant_tmp;

    real Res_start;
    real Res_stop;
    real Sel_stop;

    integer output_nul;

    initial
    begin
        output_nul=0;
        Res_in_process<=0;
        Res_quant_tmp<=0;
        Res_needed<=0;
    end

    always @(posedge Clk)
    begin
        if (output_nul==0)
            output_nul<=1;
        else
            begin
                Res_in_process<=1;
                Res_needed<=1;
                Res_start<= $abstime;
            end
        end
    end
end

```



```

always @(VCO_in)
begin
  if (Res_in_process)
  begin
    Res_in_process<=0;
    Res_stop<=$abstime;
  end
else
begin
  if (Res_needed)
  begin
    Sel_stop=$abstime;
    Res_quant_tmp<=1-((Res_stop-Res_start)/(Sel_stop-Res_stop));
    Res_needed<=0;
  end
  end // else: !if(Res_in_process)

end // always @ (VCO_in)

assign Res_quant=Res_quant_tmp;
endmodule

```

Script MATLAB de simulation du convertisseur avec extraction de résidu

```
%% Options de simulation
% Ci-dessous se trouvent les differentes options de la simulation

%% Parametrage de la generation des signaux d'entree
% GEN_VIN = 1 => Le signal sinusoidal V_in sera genere a la
% frequence Fin
% GEN_VIN = 0 => La simulation s'attend a ce qu'un signal V_in de
% la longueur appropriee soit deja present

GEN_VIN=1;

if GEN_VIN
    clear all; % Si on genere de nouveaux signaux, on fait le menage avant
    close all;
    GEN_VIN=1;
end

close all;

%% Tracage des courbes
% PLOTTING = 1 => On trace les courbes simulees. Ca cree beaucoup de graphes ...

PLOTTING=1;

%% Quantification du residu
% Le residu est un ratio A/B
% QUANT_ABS = 1 => On quantifie A puis on quantifie B
% QUANT_ABS = 0 => On quantifie le rapport A/B

QUANT_ABS=0;

%% Non linearite
% NL = 0 => On considere Kvco constant sur toute la plage de
% frequence
% NL = 1 => Kvco suit un polynome en
% coeff_a.x^2+coeff_b.x+coeff_c

NL=1;

% Generation de la caracteristique du VCO en fonction des parametres
```

```

if ( NL)
    Kvco=2.0382e9;
    Fcentrale=1.0464e9;
else
    Kvco=2.0378e9; %Valeur nominale
    coeff_a=-5.6534e9;
    coeff_b=10.518e9;
    coeff_c=-3.5191e9;
end

CAPA_MATCH=1; % Tentative pour prendre en compte l'impact du matching.

if CAPA_MATCH
    Matching_factor=1; % Si à 1, on a une capacite de valeur nominal_value et la
seconde de valeur Matching_factor*nominal_value
    Nominal_value_p=1; % Pour le chemin plus de l'architecture pseudo-differentielle

    Unmatched_value_p=Matching_factor*Nominal_value_p;
    Nominal_value_n=1; % Pour le chamin moins
    Unmatched_value_n=Matching_factor*Nominal_value_n;

end

%% Definition des parametres du systeme

Fin=10e6; % Frequence du signal d'entree desiree
Tin=1/Fin;
Fs_wished=3e9; % La frequence d'echantillonnage desiree
Ts_fs=round(((1/Fs_wished)*1e15)/2); % Expression en fs pour aller dans la simulation
Cadence si besoin
Ts=2*Ts_fs*1e-15;
Fs=1/Ts % Le vraie frequence d'echantillonnage (choisie pour avoir un nombre entier
de points dans une periode d'echantillonnage)
P2=15;
Nb_useful_pts=2^P2; % Nombre de points a considerer in fine. Utile pour tracer des
FFT
k=floor((Nb_useful_pts)*Fin/Fs);
Fin=Fs/(Nb_useful_pts/k);
Tin_fs=round((1/Fin)*1e15);
Fin=1/(Tin_fs*1e-15) % Vraie frequence d'entree (pour avoir un nombre entier de
periodes du signal pour la FFT)

```

```

Vfmin=0.5; % Correspond a la tension donnant Fvco_min
Vfmax=1; % Correspond a la tension donnant Fvco_max

Nphase=31; % Nombre de phases du VCO
Tstep_wished=1e-12; % Le pas de simulation desire

% Test. Tentative de prise en compte du bruit de phase sur l'horloge. pas verifie.

Purity_clock=-500; % Purete spectrale de l'horloge. Definie en dBc @ 1MHz de la
porteuse
Purity_VCO_msb=-500; % Purete spectrale du VCO. Definie en dBc @ 1MHz de la porteuse

V_amp=0.25; % Amplitude du signal d'entre
V_bias=0.75; % Tension mediane

DAC_step=0.028; % Gain du DAC
N_lv1=31; % Nombre de niveaux du DAC
BW=50e6; % Bande pour le calcul de SNDR

%% Quantification des residus

if QUANT_ABS
    % Version absolue
    N_bits=20; % Nombre de bits de quantification du residu
    T_max_phase=1/(Nphase*(coeff_c+Vfmin*Kvco)); % Dynamique du residu
    q_step=T_max_phase/(2^N_bits-1) % Pas de quantification

    % Version relative entre 0 et 1
else
    N_bits=4; % Nombre de bits de quantification du residu
    q_step=2/(2^N_bits-1) % Pas de quantification
end

%%

%% Definition de grandeurs utiles
if ( NL)
    F_max=coeff_c+Vfmax*Kvco;
else
    F_max=coeff_a*Vfmax^2+coeff_b*Vfmax+coeff_c;
end

```

```

V_ref_DAC=N_lvl/2*DAC_step; % La limite en tension du DAC

length_simu_min=(Nb_useful_pts+7)*Ts;

nb_point_in_Ts=round(Ts/Tstep_wished);
Tstep=Ts/nb_point_in_Ts
Fstep=1/Tstep
t_simu=Tstep:Tstep:(length_simu_min);
t_test=Ts:Ts:(length_simu_min);

Tmin=1/(Nphase*F_max);
if (Tstep >= Tmin/4) % Une vérification sommaire de fonctionnement
    fprintf(2,'Error !!! The chosen parameters do not respect the second required
relation in order the converter to work properly.\n');
    return
end

if GEN_VIN
    V_in=V_amp*sin(2*pi*Fin*t_simu+pi/2);% MODIFIE
    %V_in=0.1*ones(1,length(t_simu));
end

if PLOTTING
    figure;
    plot(V_in);
end

V_in_plus=V_bias+V_in;
V_in_minus=V_bias-V_in;

input_Clock_no_jitter=nb_point_in_Ts*[1:(Nb_useful_pts+3)];%+3

D_clock=(10^(Purity_clock/10))*2*pi^2*(1e6)^2;
f_pll=5e6; % En cours de test
w_pll=2*pi*f_pll;
jitter=PLL_jitter_generation(w_pll,1/sqrt(2),Fs, D_clock,length(input_Clock_no_jitter),2);
jitter_index=round(jitter/Tstep);
input_Clock=input_Clock_no_jitter+jitter_index; % Generation de l'horloge

```

```

if PLOTTING
    plot_MSB=figure;
    plot_res=figure;
    plot_res_cd=figure;
    drawnow; % Pour pas se faire preempter le focus quand la simulation sera finie

end

tic;

% Generation des fronts d'horloge

Clk_fall=round(diff(input_Clock)/2);
Clk_fall=[round(input_Clock(1)/2) Clk_fall+input_Clock(1:(end-1))];
Clk_rise_fall=sort([input_Clock Clk_fall]);
Clk_rise_fall=reshape(Clk_rise_fall,2,round(length(Clk_rise_fall)/2))';
Clk_zero_cell=num2cell(Clk_rise_fall,2);
Clk_zero_index=cellfun(@(x) x(1):(x(2)-1),Clk_zero_cell,'UniformOutput',0);
Clk_zero_index=cat(2,Clk_zero_index{:});
Clk_tempus=ones(1,length(t_simu));
Clk_tempus(Clk_zero_index)=0;

%% Initialisation

output_MSB_plus=zeros(1,length(input_Clock_no_jitter));
output_MSB_minus=zeros(1,length(input_Clock_no_jitter));

output_MSB=zeros(1,length(input_Clock_no_jitter));
MSB_front=zeros(1,length(input_Clock_no_jitter));

output_DAC=zeros(1,length(input_Clock_no_jitter));

vco_msb_minus=ones(1,length(Clk_tempus));
vco_msb_plus=ones(1,length(Clk_tempus));

V_in_msb_plus=zeros(1,length(Clk_tempus));
V_in_msb_minus=zeros(1,length(Clk_tempus));

Phase_msb_plus=zeros(1,length(Clk_tempus));

```

```

Phase_msb_minus=zeros(1,length(Clk_tempus));

Compteur_msb_plus=zeros(1,length(Clk_tempus));
Compteur_msb_plus_old=0;
Compteur_msb_minus=zeros(1,length(Clk_tempus));
Compteur_msb_minus_old=0;

Res_minus=zeros(1,length(Clk_tempus));
Res_plus=zeros(1,length(Clk_tempus));
Res_coup_dapres=zeros(1,length(Clk_tempus));

current_index=1;
res_to_get=0;
sel_inst_plus=0;
sel_inst_minus=0;
res_inst_plus=0;
res_inst_minus=0;

waiting_res_plus=0;
waiting_res_minus=0;

Compteur_de_front_plus=0;
Compteur_de_front_minus=0;

wait_cd_minus=0;
wait_cd_plus=0;

wait_cd_sel_plus=0;
wait_cd_sel_minus=0;

Res_cd_plus=0;
Res_cd_minus=0;

Compteur_tmp_plus=0;
Compteur_tmp_minus=0;

% Conversion

for i=1:length(Clk_tempus)

    if i==1 % Pour la valeur initiale on a pas d'instant n-1

        V_in_msb_plus(i)=V_in_plus(i);

```

```

if (V_in_msb_plus(i)>=Vfmax)
    V_in_msb_plus(i)=Vfmax;
end
if (V_in_msb_plus(i)<=Vfmin)
    V_in_msb_plus(i)=Vfmin;
end

V_in_msb_minus(i)=V_in_minus(i);
if (V_in_msb_minus(i)>=Vfmax)
    V_in_msb_minus(i)=Vfmax;
end
if (V_in_msb_minus(i)<=Vfmin)
    V_in_msb_minus(i)=Vfmin;
end

if ( NL)
    Phase_msb_plus(i)=2*pi*Tstep*(Kvco*V_in_msb_plus(i)+coeff_c);
    Phase_msb_minus(i)=2*pi*Tstep*(Kvco*V_in_msb_minus(i)+coeff_c);
else
    Phase_msb_plus(i)=2*pi*Tstep*(coeff_a*(V_in_msb_plus(i)^2)+coeff_b
        *V_in_msb_plus(i)+coeff_c);
    Phase_msb_minus(i)=2*pi*Tstep*(coeff_a*(V_in_msb_minus(i)^2)+coeff_b
        *V_in_msb_minus(i)+coeff_c);
end

Compteur_msb_plus(i)=floor(Phase_msb_plus(i)/(2*pi/Nphase));
Compteur_msb_minus(i)=floor(Phase_msb_minus(i)/(2*pi/Nphase));

else

V_in_msb_plus(i)=V_in_plus(i)-output_DAC(i-1);
V_in_msb_minus(i)=V_in_minus(i)+output_DAC(i-1);

if (V_in_msb_plus(i)>=Vfmax)
    V_in_msb_plus(i)=Vfmax;
end
if (V_in_msb_plus(i)<=Vfmin)
    V_in_msb_plus(i)=Vfmin;
end

if (V_in_msb_minus(i)>=Vfmax)
    V_in_msb_minus(i)=Vfmax;
end

```



```

if (V_in_msb_minus(i)<=Vfmin)
    V_in_msb_minus(i)=Vfmin;
end
% Calcul de la phase
if ( NL)
    Phase_msb_plus(i)=Phase_msb_plus(i-1)+2*pi*Tstep*(Kvco
        *V_in_msb_plus(i)+coeff_c);
    Phase_msb_minus(i)=Phase_msb_minus(i-1)+2*pi*Tstep*(Kvco*
        V_in_msb_minus(i)+coeff_c);
else
    Phase_msb_plus(i)=Phase_msb_plus(i-1)+2*pi*Tstep*(coeff_a
        *(V_in_msb_plus(i)^2)+coeff_b*V_in_msb_plus(i)+coeff_c);
    Phase_msb_minus(i)=Phase_msb_minus(i-1)+2*pi*Tstep*(coeff_a
        *(V_in_msb_minus(i)^2)+coeff_b*V_in_msb_minus(i)+coeff_c);
end

% Detection du nombre de fronts de VCO

Compteur_instant_plus=floor(Phase_msb_plus(i)/(pi/Nphase));
Compteur_instant_minus=floor(Phase_msb_minus(i)/(pi/Nphase));

Compteur_tmp_plus=Compteur_tmp_plus+Compteur_instant_plus;
Compteur_tmp_minus=Compteur_tmp_minus+Compteur_instant_minus;

% Gestion des compteurs

if (Compteur_tmp_plus==1)
    nope=0;
end

if (mod(Compteur_tmp_plus,2)==0 && Compteur_tmp_plus =0)
    Compteur_msb_plus(i)=Compteur_msb_plus(i-1)+1;
    Compteur_tmp_plus=0;
else
    Compteur_msb_plus(i)=Compteur_msb_plus(i-1);
end

if (mod(Compteur_tmp_minus,2)==0 && Compteur_tmp_minus =0)
    Compteur_msb_minus(i)=Compteur_msb_minus(i-1)+1;
    Compteur_tmp_minus=0;
else
    Compteur_msb_minus(i)=Compteur_msb_minus(i-1);
end

```

```

if (Compteur_msb_plus(i)==1)
    nope=0;
end

if (Compteur_msb_plus(i)==2)
    nope=0;
end

% On borne la phase

Phase_msb_plus(i)=mod(Phase_msb_plus(i),pi/Nphase);
Phase_msb_minus(i)=mod(Phase_msb_minus(i),pi/Nphase);

% On genere la sortie du VCO (duplique pour chemin negatif)

if (Compteur_instant_plus>0) % (floor(Phase_msb_plus(i)/(pi/Nphase_msb))
> floor(Phase_msb_plus(i-1)/(pi/Nphase_msb)))
    vco_msb_plus(i)=1-vco_msb_plus(i-1);

else
    vco_msb_plus(i)=vco_msb_plus(i-1);
end

% Extraction du residu (duplique pour le chemin negatif)

if (vco_msb_plus(i)==1 && vco_msb_plus(i-1)==0)
    sel_length_plus=i-sel_inst_plus;
    sel_inst_plus=i;
    res_inst_plus=i;
    Compteur_de_front_plus=Compteur_de_front_plus+1;
    if waiting_res_plus
        if QUANT_ABS
            Res_plus(current_index-1)=round(res_length_plus*Tstep/q_step)
            *q_step/(round(sel_length_plus*Tstep/q_step)*q_step);
        else
            Res_plus(current_index-1)=round((res_length_plus/sel_length_plus)
            /q_step)*q_step;
        end
        waiting_res_plus=0;
    end
    if (wait_cd_sel_plus==1)
        wait_cd_sel_plus=0;
        if QUANT_ABS

```

```

        if CAPA_MATCH
            Res_cd_plus=1-(round(Nominal_value_p*Res_cd_plus*Tstep/q_step)
                *q_step/(round((i-Start_sel_plus)*Unmatched_value_p
                    Tstep/q_step)*q_step));
        else
            Res_cd_plus=1-(round(Res_cd_plus*Tstep/q_step)*q_step
                /((round((i-Start_sel_plus)*Tstep/q_step)*q_step));
        end
    else
        if CAPA_MATCH
            Res_cd_plus=1-(round(((Res_cd_plus*Nominal_value_p)
                /((i-Start_sel_plus)*Unmatched_value_p))/q_step)*q_step);
        else
            Res_cd_plus=1-(round(((Res_cd_plus)/((i-Start_sel_plus)))
                /q_step)*q_step);
        end
    end

end

if (wait_cd_plus==1)
    Res_cd_plus=i-Start_res;
    Start_sel_plus=i;
    wait_cd_plus=0;
    wait_cd_sel_plus=1;
end

end

if (Compteur_instant_minus>0)%((floor(Phase_msb_minus(i)/(pi/Nphase_msb))
> floor(Phase_msb_minus(i-1)/(pi/Nphase_msb)))
    vco_msb_minus(i)=1-vco_msb_minus(i-1);

else
    vco_msb_minus(i)=vco_msb_minus(i-1);
end
if (vco_msb_minus(i)==1 && vco_msb_minus(i-1)==0)
    sel_length_minus=i-sel_inst_minus;
    sel_inst_minus=i;
    res_inst_minus=i;
    Compteur_de_front_minus=Compteur_de_front_minus+1;
    if waiting_res_minus
        if QUANT_ABS

```

```

    if CAPA_MATCH
        Res_minus(current_index-1)=round(res_length_minus*Nominal_value_n
            *Tstep/q_step)*q_step/(round(sel_length_minus*Unmatched_value_n
            *Tstep/q_step)*q_step);
    else
        Res_minus(current_index-1)=round(res_length_minus*Tstep/q_step)
            *q_step/(round(sel_length_minus*Tstep/q_step)*q_step);
    end
else
    if CAPA_MATCH
        Res_minus(current_index-1)=round(((res_length_minus*Nominal_value_n)
            /(sel_length_minus*Unmatched_value_n))/q_step)*q_step;
    else
        Res_minus(current_index-1)=round((res_length_minus/sel_length_minus)
            /q_step)*q_step;
    end
end

waiting_res_minus=0;
end
if (wait_cd_sel_minus==1)
    wait_cd_sel_minus=0;
    if QUANT_ABS
        if CAPA_MATCH
            Res_cd_minus=1-(round(Res_cd_minus*Nominal_value_n*Tstep/q_step)*q_step
                /(round(((i-Start_sel_minus)*Unmatched_value_n*Tstep)/q_step)*q_step));
        else
            Res_cd_minus=1-(round(Res_cd_minus*Tstep/q_step)*q_step
                /(round((i-Start_sel_minus)*Tstep/q_step)*q_step));
        end
    else
        if CAPA_MATCH
            Res_cd_minus=1-(round((Res_cd_minus*Nominal_value_n/((i-Start_sel_minus)
                *Unmatched_value_n))/q_step)*q_step);
        else
            Res_cd_minus=1-(round((Res_cd_minus/(i-Start_sel_minus))/q_step)*q_step);
        end
    end
end

end

if (wait_cd_minus==1)
    Res_cd_minus=i-Start_res;
    Start_sel_minus=i;

```

```

        wait_cd_minus=0;
        wait_cd_sel_minus=1;
    end

end

    if (Clk_tempus(i)==1 && Clk_tempus(i-1)==0) % Front d'horloge donc on met
a jour les sorties

        output_MSB_plus(current_index)=Compteur_msb_plus(i);
        output_MSB_minus(current_index)=Compteur_msb_minus(i);
        output_MSB(current_index)=output_MSB_plus(current_index)
            -output_MSB_minus(current_index);
        %output_DAC(i)=last_DAC;
        output_DAC(i)=output_MSB(current_index)*DAC_step;
        if (output_DAC(i)>=V_ref_DAC)
            output_DAC(i)=V_ref_DAC;
        end
        if (output_DAC(i)<=-V_ref_DAC)
            output_DAC(i)=-V_ref_DAC;
        end

        MSB_front(current_index)=Compteur_de_front_plus-Compteur_de_front_minus;
        Res_coup_dapres(current_index)=Res_cd_plus-Res_cd_minus;
        current_index=current_index+1;
        waiting_res_plus=1;
        waiting_res_minus=1;
        res_length_minus=i-res_inst_minus;
        res_length_plus=i-res_inst_plus;

        Start_res=i;
        wait_cd_minus=1;
        wait_cd_plus=1;
    else
        output_DAC(i)=output_DAC(i-1);
    end

end

end

if PLOTTING % On trace tout

```

```

figure;
plot(vco_msb_plus);
title('VCO MSB plus');
figure;
plot(output_MSB);
title('output MSB');
figure;
plot(output_DAC);
title('DAC output');
end

G=Nphase*Kvco/Fs;
Res_loop=Res_plus(1:Nb_useful_pts+1)-Res_minus(1:Nb_useful_pts+1);

% On applique la fonction du bloc de post-traitement numerique et on calcule le
SNDR

SNDR_msb=SNDR_over_BW(output_MSB(2:Nb_useful_pts+1),1/Ts,Fin,[0 BW])
SNDR_res_lin_black=SNDR_over_BW_extended5(my_blackman(Nb_useful_pts,'periodic')
.*(output_MSB(2:Nb_useful_pts+1)+(2*G*DAC_step-1)*output_MSB(1:Nb_useful_pts)
+Res_loop(2:Nb_useful_pts+1)-Res_loop(1:Nb_useful_pts)),1/Ts,Fin,[0 BW])
SNDR_res_lin_cd_black=SNDR_over_BW_extended5(my_blackman(Nb_useful_pts,'periodic')
.*(output_MSB(2:Nb_useful_pts+1)+(2*G*DAC_step-1)*output_MSB(1:Nb_useful_pts)
+Res_coup_dapres(3:Nb_useful_pts+2)-Res_coup_dapres(2:Nb_useful_pts+1)),1/Ts,Fin,[0
BW])

toc;

```

Bibliographie

- [Allier *et al.*, 2003] ALLIER, E., SICARD, G., FESQUET, L. et RENAUDIN, M. (2003). A new class of asynchronous A/D converters based on time quantization. *In Ninth International Symposium on Asynchronous Circuits and Systems, 2003. Proceedings.*, pages 196–205.
- [Dhanasekaran *et al.*, 2009] DHANASEKARAN, V., GAMBHIR, M., ELSAYED, M., SANCHEZ-SINENCIO, E., SILVA-MARTINEZ, J., MISHRA, C., CHEN, L. et PANKRATZ, E. (2009). A 20mhz BW 68db DR CT Sigma-Delta ADC based on a multi-bit time-domain quantizer and feedback element. *In Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pages 174–175,175a.
- [El-Halwagy *et al.*, 2013] EL-HALWAGY, W., DESSOUKY, M. et EL-GHITANI, H. (2013). A programmable 8-bit, 10mhz BW, 6.8mw, 200msample/sec, 70db SNDR VCO-based ADC using SC feedback for VCO linearization. *In 2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, pages 157–160.
- [Gao *et al.*, 2012] GAO, P., XING, X., CRANINCKX, J. et GIELEN, G. (2012). Design of an intrinsically-linear double-VCO-based ADC with 2nd-order noise shaping. *In Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, pages 1215–1220.
- [Ghosh et Pamarti, 2015] GHOSH, A. et PAMARTI, S. (2015). Linearization Through Dithering : A 50 MHz Bandwidth, 10-b ENOB, 8.2 mW VCO-Based ADC. *IEEE Journal of Solid-State Circuits*, 50(9):2012–2024.
- [Guan et Singer, 2007] GUAN, K. M. et SINGER, A. C. (2007). Opportunistic Sampling of Bursty Signals by Level-Crossing-an Information Theoretical Approach. *In Information Sciences and Systems, 2007. CISS'07. 41st Annual Conference on*, pages 701–707. IEEE.

- [Hernández et al., 2011] HERNÁNDEZ, L., PATÓN, S. et PREFASI, E. (2011). VCO-based sigma delta modulator with PWM precoding. *Electronics Letters*, 47(10):588–589.
- [Inose et al., 1966] INOSE, H., AOKI, T. et WATANABE, K. (1966). Asynchronous delta-modulation system. *Electronics Letters*, 2(3):95–96.
- [K. Lee et al., 2015] K. LEE, Y. YOON et N. SUN (2015). A Scaling-Friendly Low-Power Small-Area ADC With VCO-Based Integrator and Intrinsic Mismatch Shaping Capability. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 5(4):561–573.
- [Kar et al., 2015] KAR, A., MAJUMDER, A., MONDAL, A. et MISHRA, N. (2015). Design of ultra low power flash ADC using TMCC amp ; bit referenced encoder in 180nm technology. In *2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA)*, pages 1–6.
- [Kim et al., 2010] KIM, J., JANG, T.-K., YOON, Y.-G. et CHO, S. (2010). Analysis and Design of Voltage-Controlled Oscillator Based Analog-to-Digital Converter. *IEEE Transactions on Circuits and Systems I : Regular Papers*, 57(1):18–30.
- [Kington, 2015] KINGET, P. R. (2015). Scaling analog circuits into deep nanoscale CMOS : Obstacles and ways to overcome them. In *2015 IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–8.
- [Kumar et al., 2009] KUMAR, A. M., VEERAMACHANENI, S. et SRINIVAS, M. B. (2009). A novel low power, variable resolution pipelined ADC. In *2009 IEEE International SOC Conference (SOCC)*, pages 183–186.
- [Murmman, 2016] MURMANN, B. (2016). ADC Performance Survey 1997-2017.
- [Park et Perrott, 2009] PARK, M. et PERROTT, M. (2009). A 0.13 μm CMOS 78db SNDR 87mw 20mhz BW CT $\Delta\Sigma$ ADC with VCO-based integrator and quantizer. In *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International*, pages 170–171,171a.
- [Patil et al., 2015] PATIL, S., RATIU, A., MORCHE, D. et TSIVIDIS, Y. (2015). A 3-10fj/conv-step 0.0032mm² error-shaping alias-free asynchronous ADC. In *2015 Symposium on VLSI Circuits (VLSI Circuits)*, pages C160–C161.
- [Roa et al., 2014] ROA, G., PELLETER, T. L., BONVILAIN, A., CHAGOYA, A. et FESQUET, L. (2014). Designing ultra-low power systems with non-uniform sampling and event-driven logic. In *2014 27th Symposium on Integrated Circuits and Systems Design (SBCCI)*, pages 1–6.
- [Roberts, 2013] ROBERTS, G. (2013). Time-Domain Analog Signal Processing Techniques. *ITAC 2013*.
- [Roberts et Ali-Bakhshian, 2010] ROBERTS, G. W. et ALI-BAKHSIAN, M. (2010). A Brief Introduction to Time-to-Digital and Digital-to-Time Converters. *IEEE Transactions on Circuits and Systems II : Express Briefs*, 57(3):153–157.

- [Sayiner *et al.*, 1996] SAYINER, N., SORENSEN, H. V. et VISWANATHAN, T. R. (1996). A level-crossing sampling scheme for A/D conversion. *Circuits and Systems II : Analog and Digital Signal Processing, IEEE Transactions on*, 43(4):335–339.
- [Shibata *et al.*, 2012] SHIBATA, H., SCHREIER, R., YANG, W., SHAIKH, A., PATTERSON, D., CALDWELL, T. C., ALLDRED, D. et LAI, P. W. (2012). A DC-to-1 GHz Tunable RF Δ Sigma ADC Achieving DR = 74 dB and BW = 150 MHz at $f_0 = 450$ MHz Using 550 mW. *IEEE Journal of Solid-State Circuits*, 47(12):2888–2897.
- [Staszewski *et al.*, 2005] STASZEWSKI, R. B., FERNANDO, C. et BALSARA, P. T. (2005). Event-driven Simulation and modeling of phase noise of an RF oscillator. *IEEE Transactions on Circuits and Systems I : Regular Papers*, 52(4):723–733.
- [Straayer et Perrott, 2008] STRAAYER, M. et PERROTT, M. (2008). A 12-Bit, 10-MHz Bandwidth, Continuous-Time ADC With a 5-Bit, 950-MS/s VCO-Based Quantizer. *IEEE Journal of Solid-State Circuits*, 43(4):805–814.
- [Taylor et Galton, 2010] TAYLOR, G. et GALTON, I. (2010). A Mostly-Digital Variable-Rate Continuous-Time Delta-Sigma Modulator ADC. *IEEE Journal of Solid-State Circuits*, 45(12):2634–2646.
- [Taylor et Galton, 2013] TAYLOR, G. et GALTON, I. (2013). A Reconfigurable Mostly-Digital Delta-Sigma ADC With a Worst-Case FOM of 160 dB. *IEEE Journal of Solid-State Circuits*, 48(4):983–995.
- [Unnikrishnan et Vesterbacka, 2014] UNNIKRISHNAN, V. et VESTERBACKA, M. (2014). Time-Mode Analog-to-Digital Conversion Using Standard Cells. *IEEE Transactions on Circuits and Systems I : Regular Papers*, 61(12):3348–3357.
- [Unnikrishnan et Vesterbacka, 2016] UNNIKRISHNAN, V. et VESTERBACKA, M. (2016). Design of a VCO-based ADC in 28 nm CMOS. *In 2016 IEEE Nordic Circuits and Systems Conference (NORCAS)*, pages 1–4.
- [Varshney *et al.*, 2013] VARSHNEY, S., GOSWAMI, M. et SINGH, B. R. (2013). 4-6 Bit Variable Resolution ADC. *In 2013 International Symposium on Electronic System Design*, pages 72–76.
- [Vezyrtzis et Tsividis, 2009] VEZYRTZIS, C. et TSIVIDIS, Y. (2009). Processing of signals using level-crossing sampling. *In Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 2293–2296. IEEE.
- [Yamada et Toshiyoshi, 2017] YAMADA, S. et TOSHIYOSHI, H. (2017). An illuminance sensor integrated with analog digital converter using pulse density modulation. *In 2017 IEEE SENSORS*, pages 1–3.
- [Zhang *et al.*, 2011] ZHANG, H., TAN, J., ZHANG, C., CHEN, H. et SÁNCHEZ-SINENCIO, E. (2011). A 0.6-to-200msps speed reconfigurable and 1.9-to-27mw power scalable 10bit ADC. *In 2011 Proceedings of the ESSCIRC (ESSCIRC)*, pages 367–370.

[Zhu *et al.*, 2015] ZHU, Z., QIU, Z., LIU, M. et DING, R. (2015). A 6-to-10-Bit 0.5 V-to-0.9 V Reconfigurable 2 MS/s Power Scalable SAR ADC in 0.18 μm CMOS. *IEEE Transactions on Circuits and Systems I : Regular Papers*, 62(3):689–696.

Table des figures

1	Évolution des nœuds technologiques au cours du temps selon l'ITRS.	xi
2	Évolution de la tension d'alimentation des transistors au cours du temps selon l'ITRS.	xiv
3	Lien entre tension d'alimentation et dynamique du signal.	xv
4	Effet de la réduction des tensions d'alimentation sur la dynamique du signal.	xv
5	Évolution du facteur de pénalité en puissance en fonction de la tension d'alimentation.	xvi
6	Grafcet de fonctionnement d'un système à deux modes de fonctionnement.	xvii
7	Dynamique arbitrairement grande en temps.	xviii
I.1	Une conversion analogique numérique traditionnelle.	2
I.2	Une conversion analogique numérique dans le domaine temporel.	4
I.3	Schéma de fonctionnement de la détection de seuils.	5
I.4	Schéma de fonctionnement de la PWM.	6
I.5	Schéma de fonctionnement de la conversion via la fréquence.	6
I.6	Représentation de la phase dans le cas d'un signal sinusoïdal de fréquence constante.	7
I.7	Conversion d'un signal analogique en grandeur numérique en passant par une représentation sur la phase via un VCO.	8
I.8	Visualisation des signaux lors d'une conversion par level-crossing.	11
I.9	Visualisation des signaux lors d'une conversion par PWM.	13
I.10	Visualisation des signaux lors d'une conversion par un ADC basé sur un VCO.	15
I.11	Caractéristique tension fréquence d'un VCO composé de 31 cellules NAND en FDSOI 28 nm.	16
I.12	Visualisation des signaux lors d'une conversion par un VCO-based ADC (vue dans le domaine de la phase).	18
I.13	Positionnement des ADCs en termes d'efficacité énergétique.	20

I.14	Positionnement des ADCs en termes de résolution.	21
I.15	Principe d'addition sur les créneaux de G. Roberts.	23
I.16	Principe de soustraction sur les créneaux de G. Roberts.	23
I.17	Principe de multiplication par un nombre entier sur les créneaux de G. Roberts.	24
I.18	Exemple de calcul avec des trains de pulses et une horloge.	25
I.19	Exemple de calcul avec des trains de pulses et une horloge.	26
I.20	Exemple de calcul avec des trains de pulses et un filtre.	26
I.21	Exemple de calcul avec des trains de pulses et un filtre.	27
I.22	Fonctionnement d'un ADC à franchissement de seuils dans une situation donnée.	30
I.23	Fonctionnement d'un ADC à franchissement de seuils avec une bande passante deux fois plus grande.	30
I.24	Fonctionnement d'un ADC à franchissement de seuils dans une situation donnée.	31
I.25	Fonctionnement d'un ADC à franchissement de seuils avec une résolution deux fois plus fine.	32
I.26	Fonctionnement d'un VCO-based ADC dans une situation donnée. . .	32
I.27	Fonctionnement d'un VCO-based ADC avec une bande passante deux fois plus grande.	33
I.28	Fonctionnement d'un VCO-based ADC dans une situation donnée. . .	34
I.29	Fonctionnement d'un VCO-based ADC avec une résolution deux fois plus fine.	35
II.1	Schéma de la méthode de numérisation des créneaux.	39
II.2	Exemple de conversion d'un signal codant pour « 0 ».	40
II.3	Exemple de conversion d'un signal codant pour « 1 ».	40
II.4	Erreur systématique introduite par la méthode de conversion.	47
II.5	Comparaison entre les conversions avec et sans non-idéalités.	48
III.1	Fonctionnement d'un VCO vu comme un générateur d'évènements asynchrones.	53
III.2	Utilisation d'un VCO pour une conversion asynchrone.	55
III.3	Vue des signaux dans le système ci-dessus.	56
III.4	Utilisation d'un VCO pour une conversion synchrone.	57
III.5	Architecture plus optimisée de VCO-based ADC.	58
III.6	Schéma équivalent de l'ADC hybride en mode dégradé.	59
III.7	Schéma équivalent de l'ADC hybride en mode performant.	59
III.8	Machine à états du bloc de décision.	61
III.9	Schéma complet de l'ADC hybride à deux modes de fonctionnement.	63
III.10	Partie active lors du fonctionnement en mode dégradé.	63
III.11	Partie active lors du fonctionnement en mode performant.	64
IV.1	Principales pistes d'amélioration des VCO-based ADCs.	68

IV.2 Cause et apparition de l'erreur de quantification de la phase.	70
IV.3 Extraction du résidu vue dans le domaine temporel.	71
IV.4 Modèle comportemental d'un VCO.	73
IV.5 Pseudo-code du modèle event-driven.	75
IV.6 Points calculés avec le modèle à échantillonnage constant (Matlab). .	76
IV.7 Points calculés avec le modèle à échantillonnage adaptatif (Spectre). .	76
IV.8 Points calculés avec le modèle à échantillonnage event-driven (VerilogAMS).	76
IV.9 Caractéristique électrique du VCO choisi et modèle mathématique associé.	79
IV.10 Pseudo-code prenant en compte le bruit de phase.	80
IV.11 Visualisation du bruit de phase avec le modèle VerilogAMS amélioré. .	81
IV.12 Erreur entre le modèle tout numérique et le modèle standard.	83
IV.13 Cause et apparition de l'erreur de quantification de la phase.	84
IV.14 Architecture du convertisseur résultant du raisonnement de ce chapitre.	86
IV.15 Impact des approximations sur différentes bandes passantes.	87
IV.16 Méthode de calcul de résidu par capacités.	92
IV.17 Architecture de l'ADC avec extraction du résidu.	94
IV.18 Positionnement de notre simulation par rapport à l'état de l'art. . . .	95

Résumé

Dans un contexte de réduction des tailles de transistors dans les technologies CMOS très avancées entraînant la réduction des tensions d'alimentation et par conséquent des dynamiques disponibles pour la représentation des signaux analogiques, ce travail de thèse vise à proposer une alternative à la représentation des données dans le domaine de l'amplitude. La solution qui a été retenue est une représentation de la donnée dans le domaine temporel.

Dans ce manuscrit nous étudions à la fois la conversion d'une donnée analogique dans le domaine temporel via, notamment, un convertisseur analogique numérique basé sur un oscillateur contrôlé en tension mais aussi les possibilités de calculs sur des signaux supports d'une information déjà codée dans le domaine temporel.

Nous proposons à l'issue de ce travail à la fois une méthode pour numériser une information temporel afin de pouvoir effectuer des calculs complexes avec, une méthode « d'extraction du résidu » pour améliorer les performances d'un *VCO-based ADC* en termes de résolution par rapport à la bande passante et une architecture de « convertisseur hybride » permettant d'adapter son fonctionnement entre un *mode dégradé* asynchrone et peu consommant et un *mode performant* synchrone et plus gourmand en énergie tout en mettant en avant le potentiel de ces solutions au travers de simulations dont les modèles se basent sur la technologie CMOS FDSOI en 28 nm.

Mot clés Domaine temporel, CMOS, FDSOI, Oscillateur contrôlé en tension, Asynchrone

Summary

Advanced CMOS nodes trend to reduce the size of transistors hence reducing the power supply voltages and consequently available dynamics for the representation of analog signals. This work aims at proposing a data representation alternative which is usually done by an amplitude value. The chosen solution is to use a time-domain representation.

In this thesis, we study both the use of a VCO-based ADC to convert an analog data into a time-domain one and a calculating method using data already encoded into the time domain.

The three pillars of this thesis are a method to digitize a time-domain data so as to do more complex calculations, a method with a « residue extraction » allowing us to improve *VCO-based ADCs* performance in terms of resolution for a given bandwidth and an innovative architecture of a hybrid ADC which can adjust its operation switching between an asynchronous *low-performance mode* (which is a low power mode) and a synchronous *high-performance mode* (which is more energy consuming). The potential of these methods is pointed out by means of simulations that mimic the behavior of the 28 nm FDSOI CMOS technology.

Keywords Time-domain, CMOS, FDSOI, Voltage-controlled oscillator, Asynchronous