

Microsystèmes inertiels vibrants pour applications spatiales : apport des fonctions numériques

Baptiste Maréchal

▶ **To cite this version:**

Baptiste Maréchal. Microsystèmes inertiels vibrants pour applications spatiales : apport des fonctions numériques. Informatique. Université Montpellier, 2016. Français. NNT : 2016MONTT265 . tel-01809008

HAL Id: tel-01809008
https://theses.hal.science/tel-01809008

Submitted on 6 Jun 2018

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

Délivré par **l'Université de Montpellier**

Préparée au sein de l'école doctorale
Information, Structures et Systèmes (I2S)

Et de l'unité de recherche
Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM)

Spécialité : **Systemes Automatiques et Microelectroniques (SyAM)**

Présentée par **Baptiste MARÉCHAL**

Microsystèmes inertiels vibrants pour applications spatiales : apport des fonctions numériques

Soutenue le **19 Décembre 2016** devant le jury composé de

Laurent LATORRE, Professeur des Universités, Université de Montpellier	Président
Jean-Yves FOURNIOLS, Professeur des Universités, INSA Toulouse	Rapporteur
Fabien PARRAIN, Maître de Conférences HDR, Université Paris-Sud	Rapporteur
Frédérick MAILLY, Maître de Conférences, Université de Montpellier	Examinateur
Jean GUÉRARD, Ingénieur, Onera	Examinateur
Guillaume DELAVOIPIÈRE, Ingénieur, CNES	Invité
Pascal NOUET, Professeur des Universités, Université de Montpellier	Directeur de thèse

Table des matières

Introduction **1**

1 Introduction aux capteurs inertiels et à l'électronique numérique **5**
1.1 La mesure accélérométrique 8
1.1.1 Mesurer l'accélération 8
1.1.2 Les accéléromètres pendulaires 9
1.1.3 Les accéléromètres à lames vibrantes 11
1.2 La mesure gyrométrique 15
1.2.1 Rappels physiques sur la force de Coriolis 16
1.2.2 Gyromètres vibrants à effet Coriolis 17
1.3 Introduction aux oscillateurs électroniques 21
1.3.1 Bref rappel du fonctionnement d'un oscillateur 22
1.3.2 Présentation d'oscillateurs numériques à résonateurs analogiques . . 24
1.3.3 Pilotage du résonateur : comparateur de phase, filtre. 28
1.4 Choix d'une architecture pour l'électronique des capteurs 29

2 Conception et dimensionnement du compteur de fréquence et du synthétiseur de fréquence **31**
2.1 La mesure de fréquence en électronique numérique 32
2.1.1 Mesure par «gates» 32
2.1.2 Comptage réciproque 33
2.1.3 Mesure de fréquence par marquage temporel 37
2.1.4 Résolution et stabilité de la mesure de fréquence : la variance d'Allan 39
2.2 Exploitation des données du marquage temporel pour la mesure d'accélération 41
2.3 Formation d'un signal sinusoïdal en électronique numérique 45
2.3.1 Paramètres de conception d'une DDS 45
2.3.2 Contraintes entre les paramètres 46
2.3.3 Influence de la fréquence d'échantillonnage sur le spectre de sortie . 49
2.3.4 Influence de l'accumulateur de phase sur le spectre de sortie 50
2.3.5 Influence de la LUT 55
2.3.6 La conversion numérique-analogique 66

2.3.7 Conclusion sur le dimensionnement d'une DDS 75

3 Réalisation et évaluation du compteur et du synthétiseur de fréquence 79
3.1 Conception d'un compteur sur Spartan 6 81
3.1.1 Introduction à la programmation du Spartan 6 81
3.1.2 Les DSP48 82
3.1.3 Cas particulier d'additionneur 83
3.1.4 Les registres de capture et les monostables 84
3.1.5 Architecture du périphérique 85
3.2 Caractérisations 85
3.2.1 Mesure simple 88
3.2.2 Mesure différentielle 89
3.3 Analyse des résultats 91
3.3.1 Identification de l'origine de la perturbation 92
3.3.2 Modification du mode de création des horloges 95
3.4 Estimation du déphasage 98
3.4.1 Référence de phase du signal pilote du résonateur 98
3.4.2 Mesure du déphasage à l'aide du compteur seul 99
3.4.3 Utilisation des données de la DDS en boucle fermée 102
3.5 Correction du déphasage 103
3.6 Conception d'une DDS 105
3.6.1 Architecture du périphérique 105
3.6.2 Mémoires dans le FPGA 106
3.6.3 Dimensionnement de la DDS 110
3.7 Caractérisation du synthétiseur 113
3.7.1 Architecture et paramètres 113
3.7.2 Caractérisation spectrale 116

4 Validations expérimentales sur capteurs inertiels 119
4.1 Accéléromètre vibrant 121
4.1.1 Assemblage de la cellule 121
4.1.2 Réalisation de mesures 122
4.1.3 Analyse des résultats 124
4.2 Gyromètre vibrant 125
4.2.1 Assemblage de la cellule 125
4.2.2 Architecture électronique 125
4.2.3 Réalisation de mesures 128
4.2.4 Analyse des résultats 129

Conclusion 131

Annexes **137**

A Simulateur de spectres de sortie de DDS **137**
A.1 Pourquoi Python ? . . . 137
A.2 Précisions sur le code . . . 138
A.2.1 Spécificités de Python . . . 138
A.2.2 Méthode `__calc_pts` . . . 138
A.2.3 Méthode `__dophase` . . . 139
A.2.4 Méthode `dofft` . . . 139
A.3 Code source . . . 140

B Détails sur les périphériques développés **147**
B.1 Périphérique sur bus AXI . . . 147
B.2 Macros et composants spécifiques au Spartan 6 . . . 148

C Variance d'Allan **151**
C.1 Estimation de la variance vraie . . . 151
C.2 Calcul pratique de la variance d'Allan . . . 152

D Codes logiciels **155**
D.1 Estimation du déphasage . . . 155

E Considérations numériciennes **159**
E.1 Transmission des données de comptage par liaison série . . . 159
E.2 Processeur, pipeline et puissance de calcul . . . 161

Bibliographie **169**

Table des figures **174**

Liste des tableaux **176**

Introduction

Les véhicules en mouvement tels que les avions, les lanceurs, les satellites... ont besoin de connaître leur position géographique et souvent leur orientation. La triangulation est une méthode classique de positionnement. Par exemple, les marins au bord des côtes mesurent l'azimut d'amers afin de se situer sur une carte. Plus modernes, les systèmes absolus de positionnement par satellites (GPS, Glonass ou Galileo) fonctionnent également sur le principe de triangulation de la constellation de satellites (les positions des satellites étant connues en tout instant). Pour l'orientation, ou l'attitude, sur Terre, une simple boussole peut, par exemple, pointer vers le nord magnétique, ou un fil à plomb indiquer la verticale locale. Pour mesurer très précisément l'attitude d'un véhicule, d'autres dispositifs existent, comme les viseurs d'étoiles, mais tous ces systèmes ne suffisent pas toujours à fournir les informations nécessaires. Par exemple, les systèmes d'un avion qui traverse une anomalie magnétique peuvent être suffisamment perturbés pour ne plus recevoir assez clairement le signal de positionnement. De même, un sous-marin en immersion ne voit pas le ciel et ne peut pas déterminer sa position sur le globe grâce aux étoiles.

Il existe cependant d'autres systèmes de positionnement, basés sur des données disponibles dans tous les cas, les données inertielles, c'est-à-dire accélérations et vitesse angulaire de l'objet. Le traitement de ces données fournit alors la position et l'attitude du véhicule et le rend autonome et indépendant de toute assistance extérieure. Ces capteurs sont des accéléromètres et des gyromètres. Des performances de ces capteurs dépendent directement l'exactitude et/ou la rapidité du positionnement.

Ces capteurs sont regroupés en classes de performance croissantes et leurs prix augmentent avec leurs classes. Aujourd'hui, un nouveau type de capteurs remplace de plus en plus souvent les capteurs traditionnels grâce aux évolutions technologiques, notamment de la filière silicium : les micro-capteurs ou capteurs MEMS[1]. L'intérêt de ces capteurs réside non seulement dans leur petite taille, mais aussi dans leur coût, réduit par la production de masse. L'électronique associée doit donc rester de dimension comparable au capteur sans pour autant en dégrader les performances. Il n'existe en effet aucun intérêt en un capteur micrométrique si son électronique occupe un volume très supérieur.

Dans le cas des accéléromètres, il existe plusieurs façons de détecter le déplacement d'une masse d'épreuve sous l'effet d'une accélération. L'une d'elles repose sur la varia-

1. MEMS : Micro-ElectroMechanical Systems, micro systèmes électromécaniques.

tion de fréquence de résonance d'une poutre vibrante sous l'effet d'une modification de contrainte, causée par le déplacement de la masse d'épreuve. En Anglais, on les appelle les Vibrating Beam Accelerometers, VBA. L'Onera a développé un micro-accéléromètre en quartz de ce type, le VIA (Vibrating Inertial Accelerometer) et se concentrait essentiellement sur le développement de la cellule sans chercher à développer une électronique associée. Aujourd'hui, l'Onera cherche à obtenir une qualification spatiale du composant final et s'intéresse donc à l'électronique embarquée. Cette volonté intervient à une période où l'électronique numérique tend à s'imposer de plus en plus, laissant à l'électronique analogique des applications spécifiques.

En effet, les progrès d'intégration incessants depuis l'apparition du transistor autorisent la conception de systèmes numériques toujours plus puissants et moins gourmands en énergie. C'est la célèbre loi de Moore : le nombre de transistors intégrés sur une surface donnée double tous les dix-huit mois. Un cas particulier attire l'attention dans ce domaine. Initialement conçus comme systèmes de prototypage, les FPGA[2] sont aujourd'hui suffisamment puissants pour constituer l'électronique finale de produits de consommation courants : télévisions, machines à laver, «box» multimédia... En effet, la facilité de programmation des FPGA modernes autorise une mise en place rapide de solutions performantes et évolutives.

Aujourd'hui, la puissance de calcul disponible dépasse largement les besoins des microcapteurs de l'Onera, tant et si bien que des modèles d'entrée de gamme suffisent à répondre au besoin. En outre, leurs dimension et masse sont comparables à celles des capteurs, ce qui implique des coûts de lancement très raisonnables dans le cadre d'une utilisation spatiale. Par ailleurs, ils peuvent être reprogrammés en cours de mission, aussi bien en ce qui concerne l'implantation matérielle que le logiciel embarqué. Ainsi, une mise à jour logicielle peut, par exemple, améliorer les performances des capteurs ou les adapter à l'application visée. Une reconfiguration matérielle peut éviter l'emploi de cellules endommagées, par des radiations notamment, afin d'assurer la continuité du fonctionnement.

Le travail exposé ici étudie donc les apports d'une électronique numérique au conditionnement des capteurs inertiels vibrants développés par l'Onera. L'amélioration des performances par rapport à une architecture analogique sous tend l'ensemble des développements et études conduits au cours de cette thèse.

Le premier chapitre de ce mémoire commence par rappeler les lois physiques fondamentales qui régissent les mesures inertielles. Il présente ainsi les principes de fonctionnement de quelques capteurs ainsi que des exemples de réalisation avant d'introduire plus en détails les capteurs de l'Onera. Des rappels sur les oscillateurs, analogiques puis numériques sont ensuite effectués en vue de fournir une architecture générale d'électronique numérique de conditionnement adaptée aux capteurs de l'Onera.

2. FPGA : Field-Programmable Gate Array, circuit logique programmable en Français. Il s'agit de circuits intégrés logiques reprogrammables.

Un second chapitre présente l'aspect théorique des fonctions numériques retenues dans l'architecture. En particulier, ces études fournissent les équations et les outils nécessaires à la conception des fonctions électroniques réelles adaptées aux contraintes de performances des capteurs. Il ne s'agit en aucun cas de réaliser une étude exhaustive, le cas particulier des capteurs de l'Onera et les contraintes spécifiées le temps venu orientent cette étude.

Suite à ces travaux théoriques, le chapitre 3 présente les mises en applications des fonctions numériques. L'observation de performances inférieures aux prévisions sur une des deux fonctions mène à rechercher des défauts et leurs origines dans l'électronique conçue. Une fois identifiée la source du problème, des considérations pratiques apportent encore des propositions d'optimisation de l'électronique, afin de s'adapter aux ressources disponibles.

Dans le dernier chapitre, les performances des électroniques développées sont évaluées en conditions inertielles, associées à des structures inertielles vibrantes.

Chapitre 1

Introduction aux capteurs inertiels et à l'électronique numérique

Sommaire

1.1 La mesure accélérométrique 8
1.1.1 Mesurer l'accélération 8
1.1.2 Les accéléromètres pendulaires 9
1.1.3 Les accéléromètres à lames vibrantes 11
1.2 La mesure gyrométrique 15
1.2.1 Rappels physiques sur la force de Coriolis 16
1.2.2 Gyromètres vibrants à effet Coriolis 17
1.3 Introduction aux oscillateurs électroniques 21
1.3.1 Bref rappel du fonctionnement d'un oscillateur 22
1.3.2 Présentation d'oscillateurs numériques à résonateurs analogiques 24
1.3.3 Pilotage du résonateur : comparateur de phase, filtre... 28
1.4 Choix d'une architecture pour l'électronique des capteurs . . 29

Un véhicule peut se repérer par rapport à des objets de positions connues autour de lui. Les navigateurs utilisent depuis longtemps cette méthode. Au bord des côtes d'une part, ils notent l'azimut de plusieurs amers puis en déduisent la position du navire par triangulation. En pleine mer d'autre part, ils utilisent un sextant, qui mesure un angle vertical, pour se positionner par rapport aux étoiles. Dans l'espace, les repères de choix pour l'orientation d'une plate-forme spatiale sont les étoiles, lointaines, fixes et ponctuelles, qui permettent de restituer l'attitude à quelques secondes d'arc.

Mais dans un certain nombre de cas où les repères extérieurs ne sont pas accessibles (éblouissement du viseur d'étoiles par le Soleil ou la Lune, masquage par la Terre, ou modes de fonctionnements dégradés), la navigation repose sur les données inertielles.

Les données inertielles, proviennent du mouvement même du véhicule et suivent les lois fondamentales de la dynamique énoncées par I. Newton [NLJ60]. En particulier, la

Seconde Loi établit une relation entre la quantité de mouvement d'un mobile et les forces extérieures appliquées.

La quantité de mouvement se définit ainsi :

$$\overrightarrow{p} = m\overrightarrow{v}$$

Avec :

— m, la masse du mobile ;

— $\overrightarrow{v}$, sa vitesse de déplacement.

Si $\overrightarrow{F_{ext}}$ désigne les forces extérieures appliquées au véhicule, alors le principe fondamental (ou deuxième loi) de la dynamique stipule que :

$$\frac{d\overrightarrow{p}}{dt} = \Sigma\overrightarrow{F_{ext}} \tag{1.1}$$

Dans le cas où la masse demeure constante, cette équation se simplifie :

$$m\overrightarrow{a} = \Sigma\overrightarrow{F_{ext}} \tag{1.2}$$

Par définition, et avec les notations du tableau 1.1 :

$$\overrightarrow{v} = \frac{d\overrightarrow{OM}}{dt}$$
$$\overrightarrow{a} = \frac{d\overrightarrow{v}}{dt} = \frac{d^2\overrightarrow{OM}}{dt^2}$$

Notation	Signification
O	Origine du repère dans lequel le mobile se déplace.
M	Centre de masse du mobile.
$\overrightarrow{OM}$	Vecteur position du mobile.
$\overrightarrow{v}$	Vitesse de déplacement du mobile.
$\overrightarrow{a}$	Accélération du mobile.

TABLE 1.1 – Notations employées dans les équations mécaniques

Compte-tenu du caractère fini des forces appliquées à l'appareil, l'accélération est intégrable. Par conséquent, une double intégration du signal d'accélération, couplée à la connaissance de la position et de la vitesse initiales du véhicule[1], fournit la position de l'appareil à tout instant.

Les accéléromètres visent justement à fournir une mesure de l'accélération, afin de positionner les appareils qu'ils équipent.

1. Les instants pour la vitesse et la position peuvent différer et ne pas nécessairement correspondre à l'instant initial. De même, la connaissance de deux positions à deux instants distincts convient également.

Cependant, cette mesure et les calculs qui en découlent ne précisent pas l'attitude de l'appareil. Impossible par exemple de savoir quelle face d'un satellite pointe vers la Terre. Ce qui signifie que tels quels, les accéléromètres ne mentionnent pas si le véhicule a tourné.

Le cas des rotations pose alors deux problèmes. D'une part, les axes de la mesure accélérométrique changent (à moins de disposer d'une plate-forme gyro-stabilisée), changement à prendre en compte dans le calcul de position. D'autre part, le référentiel local du véhicule n'est plus galiléen et le principe fondamental de la dynamique s'en voit modifié. G. G. Coriolis a montré [Cor31] [Cor35] qu'en présence d'une rotation, cette équation s'écrivait alors :

$$m\,\overrightarrow{a}(M)_{/R} = m\,\overrightarrow{a}(M)_{/R'} + m\,\overrightarrow{a}(M)_e + m\,\overrightarrow{a}(M)_c$$

Où R désigne un référentiel galiléen et R', le référentiel non galiléen en rotation par rapport à R. $\overrightarrow{a}(M)_e$ et $\overrightarrow{a}(M)_c$ forment respectivement l'accélération d'inertie d'entraînement et l'accélération d'inertie de Coriolis.

Afin de connaître les rotations des axes de mesure accélérométrique et de prendre en compte les forces d'inertie dans le calcul de la position du véhicule, il faut alors mesurer les rotations subies par l'appareil.

Les gyromètres se chargent de fournir ces mesures.

Ainsi, trois accéléromètres et trois gyromètres constituent une plate-forme inertielle capable d'alimenter un ordinateur de bord avec les données nécessaires au calcul du positionnement de l'appareil qu'il équipe, sans recours à une assistance extérieure. Des performances des capteurs dépendra la qualité du positionnement (trajectoire et attitude).

1.1 La mesure accélérométrique

1.1.1 Mesurer l'accélération

Bien que les accéléromètres exploitent le principe fondamental de la dynamique (équation (1.2)) pour fournir une mesure de l'accélération, ils ne donnent pas une mesure de l'accélération $\overrightarrow{a}$ directement, mais de toutes les forces autres que les interactions gravitationnelles. Autrement dit, avec g les interactions gravitationnelles locales, l'accélération que mesurent de tels capteurs peut s'exprimer par :

$$\overrightarrow{a_{mes}} = \frac{\partial^2 \overrightarrow{OM}}{\partial t^2} - \overrightarrow{g(M)} \tag{1.3}$$

Il s'agit donc d'une mesure des forces non gravitationnelles exercées sur la masse du capteur, un accéléromètre, appelée la masse d'épreuve. Par exemple, un accéléromètre disposé pour mesurer les accélérations verticales et placé dans un ascenseur à l'arrêt fournira sur sa sortie une mesure de g (et non 0, accélération d'un ascenseur à l'arrêt). En effet, le véhicule qui supporte l'accéléromètre est soumis à deux forces :

— la pesanteur, non mesurée par l'accéléromètre ;
— la réaction du plan sur lequel repose le mobile qui contient le capteur.

Et c'est cette dernière que l'accéléromètre mesure. L'accéléromètre affichera d'ailleurs la même mesure dans le cas (fictif) où l'ascenseur se propulserait vers le haut en l'absence de champ de gravité. Impossible de distinguer ces deux cas sans une connaissance à priori du champs de gravité local. Ce qui signifie qu'en vue de déterminer la position depuis les mesures accélérométriques en tout instant, le système doit connaître le champ de gravité local.

Si maintenant l'ascenseur tombe en chute libre, l'accéléromètre affiche une mesure nulle, alors qu'il est soumis à la seule pesanteur. En fait, la masse d'épreuve et l'ascenseur subissent la même accélération de pesanteur et n'exerce aucune force l'un sur l'autre (absence de toute réaction). Selon le principe d'équivalence, ils suivent donc rigoureusement la même trajectoire.

Il existe plusieurs types d'accéléromètres, qui exploitent différemment les effets d'une accélération sur un corps. Deux grandes familles d'accéléromètres se dégagent :

— les accéléromètres pendulaires (non vibrants) ;
— les accéléromètres vibrants.

Il existe également un troisième type de capteur, moins répandu que les deux précédents : les accéléromètres à fil chauffant.

Dans ces derniers, un dispositif, souvent un fil chauffant, chauffe un gaz prisonnier dans une cavité. Sous l'effet d'une accélération, les parties chaudes de ce gaz, moins denses, vont se déplacer dans le sens du vecteur d'accélération, alors que les parties plus

froides, plus denses, vont migrer dans l'autre sens. Des sondes de température mesurent alors les températures sur le(s) axe(s) sensible(s) du capteur et fournissent une mesure de l'accélération.

1.1.2 Les accéléromètres pendulaires

Les accéléromètres pendulaires s'appuient généralement sur un asservissement en position d'un dispositif mobile. Sous l'effet d'une accélération, le mobile tend à s'écarter de sa position d'équilibre. Un système électromécanique, électrostatique ou électromagnétique applique alors une force contraire pour maintenir le mobile à l'équilibre, c'est-à-dire l'asservir en position. La commande de cet asservissement fournit alors une mesure de l'accélération.

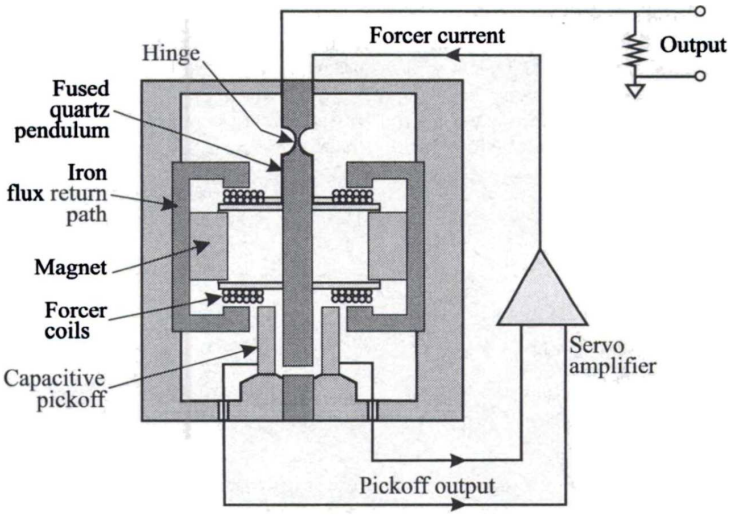


FIGURE 1.1 – Schéma de principe des accéléromètre à technologie Q-Flex® (d'après [Jac72])

La technologie d'accéléromètres Q-Flex® d'Honeywell fonctionne ainsi [Jac72] [RAD12] [2]. La figure 1.1 présente le schéma de principe de cette technologie. Un support mobile en quartz et articulé autour d'une charnière porte deux enroulements. Ces deux enroulements se déplacent dans des champs magnétiques générés par des aimants permanents et bouclés par des guides magnétiques. Sur la partie opposée à la charnière du mobile en quartz, des dépôts métalliques forment deux condensateurs avec des armatures conductrices fixes situées de part et d'autre de la partie mobile (voir la figure). Sous l'effet d'une accélération, le mobile en quartz se déplace, ce qui modifie la capacité des condensateurs. Un système de rétroaction électronique se charge alors d'appliquer un courant à travers les enroulements du mobile afin de le faire revenir à sa position initiale. Ce courant fournit alors une mesure de l'accélération appliquée au système.

2. Allied Signal appartient aujourd'hui à Honeywell

Cette technologie équipe les accéléromètres à sortie analogique d'Honeywell. Il existe plusieurs catégories de performances de ces accéléromètres. La classe inertielle, adaptée à la navigation, comporte deux accéléromètres, le QA-2000 et le QA-3000. Ce dernier présente les meilleures performances, présentées dans le tableau 1.2. Ces accéléromètres pendulaires, malgré leur conception assez ancienne, bénéficient d'une technologie très bien maîtrisée et font aujourd'hui référence en la matière.

Analog Devices dispose aussi d'accéléromètre à déplacement, basé sur la technologie iMems que la figure 1.2 présente. Le tableau 1.2 reprend les caractéristiques d'un accéléromètre basé sur cette technologie, l'ADXL278, un accéléromètre bi-axe et l'un des nombreux accéléromètres proposés qui exploitent cette technologie. Il s'agit là d'un capteur de classe de performance moindre, mais très bon marché (une dizaine de dollars) particulièrement bien adapté à des utilisations grand public (smartphones par exemple).

Caractéristique	QA3000-030	ADXL278
Intervalle de mesure	±60 g	±70 g en X et ±37 g en Y
Résolution	1 µg	< 4 mg
Biais	4 mg	≈ 1 mg
Facteur d'échelle	1,20 à 1,46 mA/g	27 mV/g en X, 55 mV/g en Y
Sensibilité thermique du biais	15 µg/°C	
Sensibilité thermique du facteur d'échelle	120 ppm/°C	
Bande passante	300 Hz	400 Hz

TABLE 1.2 – Performances des QA3000-030 d'Honeywell et ADXL278 d'Analog Devices

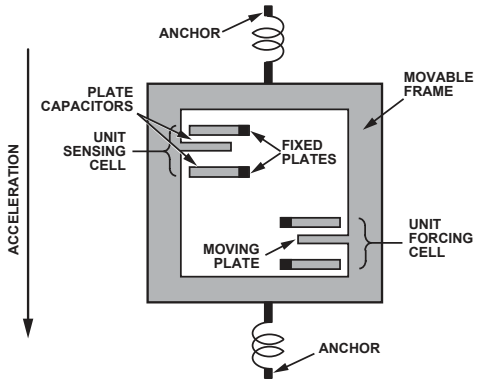


FIGURE 1.2 – Technologie iMEMS pour les accéléromètres [Dev]

En France, d'autres équipementiers ont également construit des accéléromètres de ce type [RAD12]. SEXTANT avait développé deux accéléromètres pendulaires, l'un davantage adapté au guidage de missile (forte résistance aux accélérations brèves et violentes – chocs et vibrations par exemple), l'autre, résolument tourné vers la navigation.

La SFIM (Société de Fabrication d'Instruments de Mesure) avait développé un accéléromètre pendulaire en silicium. Aujourd'hui, par le biais des rachats, fusions et ac-

quisitions, SFIM fait partie de SAGEM, qui proposait déjà une solution d'accéléromètre pendulaire à détection capacitive. Aujourd'hui, l'ensemble de cette activité inertielle est regroupé au sein de SAGEM Défense et Sécurité. En février 2013, Sagem a racheté un fabricant important de capteurs inertiels MEMS [3], la société suisse Colybris. Cette société se spécialise dans la conception et la fabrication de capteurs inertiels, sismiques ou inclinomètres par détection capacitive. Cette dernière a notamment développé un accéléromètre pendulaire MEMS en silicium dont le tableau 1.3 introduit les performances.

Caractéristique	**RS9010**
Intervalle de mesure	$\pm 10\,\mathrm{g}$
Résolution	$< 50\,\mu\mathrm{g}$
Biais	$50\,\mathrm{mg}$
Facteur d'échelle	$200\,\mathrm{mV/g}$
Sensibilité thermique du biais	$1000\,\mu\mathrm{g/°C}$
Sensibilité thermique du facteur d'échelle	$75\,\mathrm{ppm/°C} \pm 100$
Bande passante	$200\,\mathrm{Hz}$

TABLE 1.3 – Performances du RS9000 de Colybris [SDD10] [Colc] [Colb] [Cola]

Point intéressant, les accéléromètres les plus précis au monde reposent sur des accéléromètres pendulaires. La mission spatiale Microscope, qui vise à déterminer le rapport entre masse inerte et masse grave à la quinzième décimale, embarquera un accéléromètre pendulaire d'encombrement très supérieur (plusieurs litres) à celui des capteurs habituels (quelques cm^3), développé et construit à l'Onera. Dans cet instrument, une masse mobile, maintenue en lévitation dans une cage métallisée par des faces électrostatiques, fait varier une capacité ; il s'agit donc d'une détection capacitive. Aujourd'hui, seuls des accéléromètres pendulaires permettent d'atteindre cette résolution.

1.1.3 Les accéléromètres à lames vibrantes

Un autre moyen employé pour mesurer une accélération consiste à exploiter une variation de fréquence de résonance d'un système prévu à cet effet. Ce procédé mis au point depuis longtemps maintenant [Erd66] a très rapidement fait l'objet de développement en vue d'obtenir des accéléromètres à sortie fréquentielle [Ser68]. Typiquement, ce genre d'accéléromètre comporte une poutre dont la contrainte axiale varie sous l'effet d'une accélération, ce qui provoque une variation de la fréquence de résonance en flexion de la poutre. Le principe de fonctionnement se montre analogue à celui de l'accord d'une corde de guitare par exemple. Ce qui signifie que l'accélération mesurée se déduit d'une variation de fréquence.

3. MEMS : Microelectromechanical Systems

En règle générale, cette contrainte axiale s'obtient par l'intermédiaire d'une masse d'épreuve rattachée à la poutre : sous l'effet d'une accélération, la masse d'épreuve tend à se déplacer, ce qui modifie la contrainte appliquée sur le système vibrant. Ainsi, la fréquence de résonance varie selon un facteur d'échelle déterminé. L'équation (1.4) [MI68] [LF08] exprime cette variation de fréquence en fonction de l'accélération.

$$F = F_0 + K \times a \tag{1.4}$$

a représente la projection de l'accélération sur l'axe sensible de l'accéléromètre. La connaissance de la fréquence de résonance au repos F_0 et du facteur d'échelle K permettent alors de déterminer la valeur de l'accélération appliquée sur l'axe sensible.

Comparés aux accéléromètres pendulaires, les accéléromètres à lames vibrantes présentent l'avantage d'un facteur d'échelle très stable, notamment en température (quelques ppm/°C).

En outre, ces accéléromètres à lames vibrantes se prêtent bien à une mesure différentielle : deux systèmes résonants montés tête-bêche répondent de façon opposée à une accélération. L'équation (1.5) exprime l'intérêt d'une telle mesure dans un cas idéal. Si les deux systèmes vibrants disposent de la même fréquence de résonance (F_0 identiques) et du même facteur d'échelle (K identiques), alors mesurer l'écart de fréquence entre les deux systèmes résonants revient à mesurer le double de l'accélération.

$$\begin{aligned} F_1 &= F_0 + K \times a \\ F_2 &= F_0 - K \times a \\ \Delta F &= F_1 - F_2 \\ \Delta F &= 2K \times a \end{aligned} \tag{1.5}$$

Caractéristique	**RBA500**	**Thales**
Intervalle de mesure	±70 g	±80 g
Résolution	1 µg	1 µg
Biais	4 mg	0,2 mg après compensations thermiques
Facteur d'échelle	80 Hz g^{-1}	1 Hz g^{-1}
Bande passante	400 Hz	100 Hz

TABLE 1.4 – Exemples de caractéristiques d'accéléromètres VBA

Le tableau 1.4 présente les performances du RBA500 d'Honeywell [Hon05]. Il s'agit d'un VBA [4] monolithique en silicium. Le tableau le compare à un accéléromètre présenté

4. VBA : Vibrating Beam Accelerometer, accéléromètre à lame vibrante.

par Thales, vibrant et en silicium également. Ce dernier s'appuie sur une structure originale en forme de cric.

Le Draper Laboratory, propose aussi des accéléromètres oscillants en silicium : la technologie Silicon Oscillating Accelerometer (SOA) présentée sur la figure 1.3 [HMS+06]. Le tableau 1.5 introduit les performances de cet accéléromètre ainsi que celui d'une autre société, Kearfott, également en silicium : le MVBA [RJDdC95].

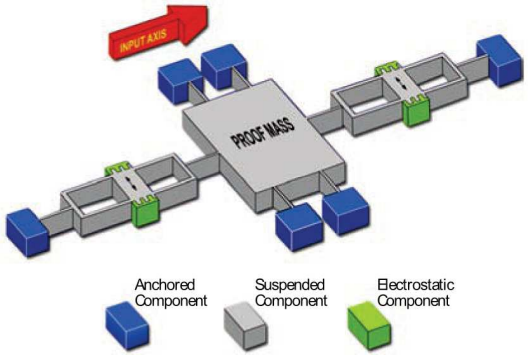


FIGURE 1.3 – Schéma de principe de l'accéléromètre SOA du Draper Laboratory (d'apres [HMS+06])

Caractéristique	SOA Navigation Draper	MVBA Kearfott
Intervalle de mesure	±2 g	±40 g
Résolution	0,08 µg	1 µg
Stabilité de biais	1 µg (90 jours)	2 µg (7 jours)
Stabilité du facteur d'échelle	10 ppm (90 jours)	50 ppm (7 jours)

TABLE 1.5 – Performances des accéléromètres SOA et MVBA

Dans leurs versions haut de gamme (réservées aux applications militaire stratégiques comme le guidage de missiles intercontinentaux), les accéléromètres à lames vibrantes surclassent les versions pendulaires, notamment pour les missions où l'accélération atteint fréquemment de fortes valeurs.

Pour les moyennes performances, les VBA se montrent intéressants pour leur grandeur de sortie quasiment numérisée, une fréquence (une électronique numérique relativement simple peut rapidement fournir la mesure sous une forme exploitable par un ordinateur).

Honeywell propose un autre accéléromètre, le SiMMA SA500, qui repose sur une structure monolithique en quartz [LF98]. Le tableau 1.6 reprend ses performances.

Mais il ne s'agit pas du premier VBA monolithique en quartz puisque l'Onera avait déjà conçu auparavant le VIA, premier VBA monolithique en quartz qui concilie performances, miniaturisation et faible coût de fabrication (par usinage collectif sur wafer).

Caractéristique	SiMMA SA500
Intervalle de mesure	±80 g
Résolution	50 µg
Stabilité du biais	2,5 mg (1 an)
Stabilité du facteur d'échelle	360 ppm (1 an)
Bande passante	1000 Hz

TABLE 1.6 – Performances de l'accéléromètre en quartz d'Honeywell

Le VIA de l'Onera

À l'évocation de capteurs vibrants, il vient rapidement à l'esprit d'utiliser le quartz comme matériau pour le capteur [Nis80], ce matériau étant réputé pour ses qualités mécaniques et son caractère piézoélectrique (notamment exploité en horlogerie depuis la fin des années 1960).

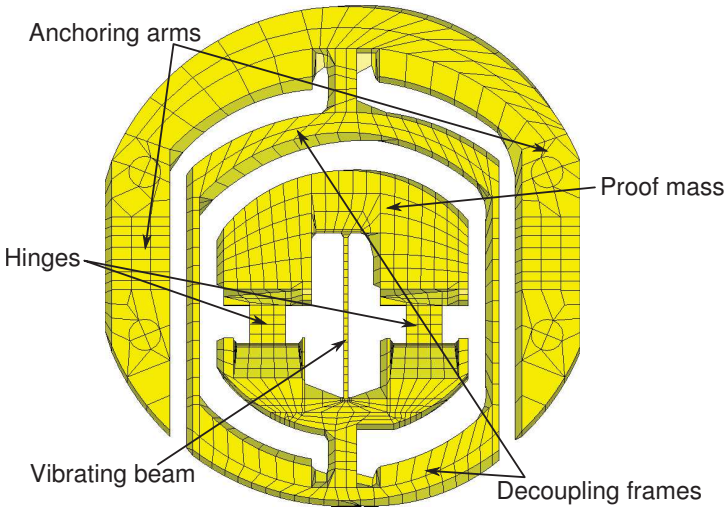
Il y a une vingtaine d'années, l'Onera a développé un accéléromètre MEMS : le Vibrating Inertial Accelerometer ou VIA [LTJMB98]. Il s'agit d'un accéléromètre à lame vibrante comme ceux évoqués précédemment, mais fabriqué en quartz, en structure monolithique qui intègre la lame vibrante, la masse d'épreuve et la partie fixe (le support). La figure 1.4 présente une cellule VIA. Un accéléromètre en comporte deux, montées tête-bêche, de telle sorte que lorsqu'une cellule mesure une accélération positive, l'autre mesure cette même accélération, mais négative.

Cette cellule en quartz a fait l'objet d'un brevet [JLTM97], notamment pour le cadre de découplage qui préserve la partie sensible du capteur (la poutre vibrante) d'influences indésirables. Le capteur repose sur une embase sur ses bras de fixation («anchoring arms»).

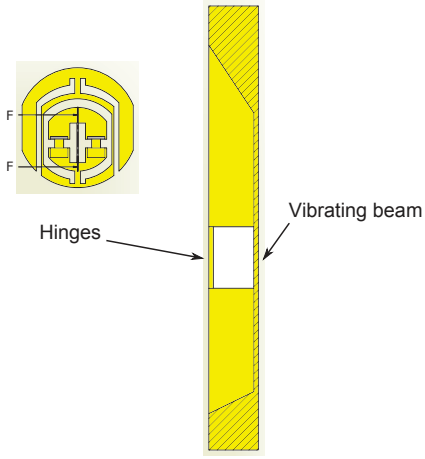
La figure 1.4 montre une vue de face et une vue de profil d'une cellule. Le lecteur notera sur la vue de profil la différence d'élévation entre la poutre, élément central de cette structure, et les charnières. Cette cellule monolithique nécessite ainsi une gravure asymétrique du wafer en quartz en vue d'obtenir cette différence de niveau.

Ainsi assemblé et monté sur embase, un accéléromètre à cellule VIA, pourvu qu'il soit accompagné d'une électronique adaptée, présente les caractéristiques présentées dans le tableau 1.7 [LG10]. Le graphique de la figure 1.5 présente quant à lui la variance d'ALLAN de l'accéléromètre, ce qui permet de lire la résolution atteignable en fonction de la bande passante de la mesure. La section 2.1.4 présentera la variance d'ALLAN, son intérêt et son allure générale. Sur cette courbe, le lecteur remarquera qu'à l'abscisse 10^1 s, la courbe rouge atteint un palier qui constitue la résolution ultime de l'accéléromètre présentée dans le tableau 1.7. Autrement dit, le VIA atteint sa résolution maximale après 10 s de mesure.

Compte-tenu des caractéristiques présentées au tableau 1.7, la résolution fréquentielle atteinte pour un accéléromètre vaut donc 20 µHz, soit relativement à la fréquence de fonctionnement (60 kHz) $3{,}3 \times 10^{-10}$. Cela donne un aperçu du potentiel métrologique de ce type de capteurs.



(a) Vue de face



(b) Vue en coupe

FIGURE 1.4 – Une cellule VIA

1.2 La mesure gyrométrique

La section précédente a présenté les accéléromètres, capteurs qui permettent de déterminer la position d'un mobile en déplacement. L'introduction de cette section a également rappelé que pour y parvenir, il faut donc aussi mesurer la rotation des axes accélérométriques, fonction assurée par les gyromètres.

En outre, cette mesure permet aussi de déterminer l'attitude d'un appareil, pour s'assurer, par exemple, qu'une face précise d'un satellite pointe vers la Terre.

Aujourd'hui, les gyromètres les plus répandus lorsqu'il s'agit de mesurer précisément une vitesse de rotation reposent sur l'effet SAGNAC. Il s'agit d'un effet découvert en 1913 par le physicien Français Georges SAGNAC : la séparation d'un faisceau lumineux en deux à la surface d'un objet en rotation (une sphère par exemple) provoque une différence de marche proportionnelle à la vitesse de rotation et donc, des interférences là où les faisceaux se rejoignent à nouveau. La figure 1.6 illustre ce phénomène.

Caractéristique	VIA
Intervalle de mesure	$\pm 100\,\mathrm{g}$
Résolution	$1\,\mu\mathrm{g}$
Temps d'intégration pour résolution maximale	$10\,\mathrm{s}$
Facteur d'échelle	$2 \times 10\,\mathrm{Hz/g}$
Fréquence de résonance de la poutre	$\approx 60\,\mathrm{kHz}$

TABLE 1.7 – Caractéristique d'une cellule VIA

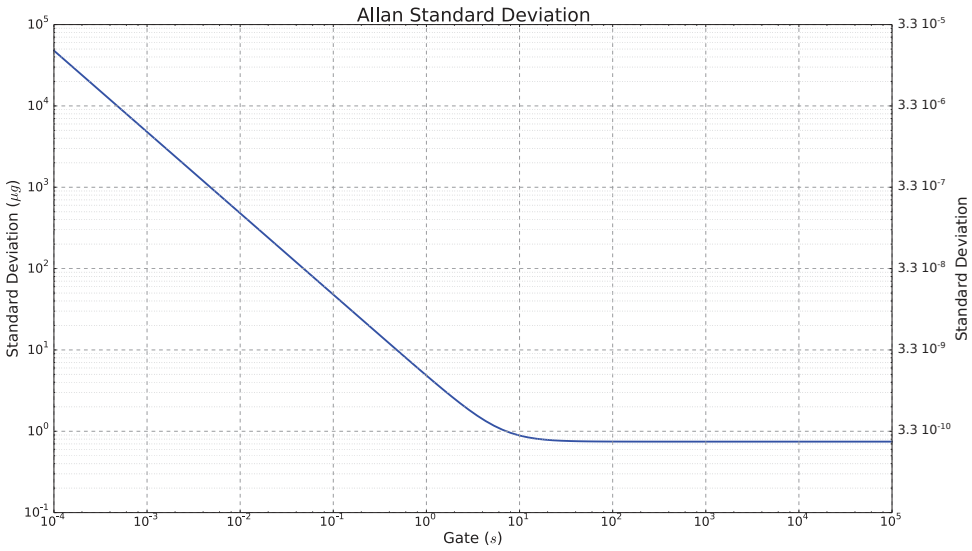


FIGURE 1.5 – Variance d'ALLAN de l'accéléromètre VIA (normalisée par rapport à la fréquence de résonance au repos)

Le problème de ces gyromètres se situe dans le volume qu'ils occupent. En effet, plus la distance à parcourir est grande, meilleure est la précision. Aussi, de tels gyromètres disposent-ils de kilomètres de fibres optiques enroulées. En vue de réduire l'espace occupé, les recherches se portent sur des gyromètres mécaniques, plus propices à une réduction d'encombrement. Ces derniers présentent l'avantage de pouvoir se miniaturiser jusqu'à des dimensions microscopiques. Le principe de fonctionnement repose cette fois-ci sur les effets de la force d'inertie de CORIOLIS.

1.2.1 Rappels physiques sur la force de Coriolis

Pour rappel, en présence d'une rotation, le principe fondamental de la dynamique s'exprime selon l'équation (1.6).

$$m\,\vec{a}(M)_{/R} = m\,\vec{a}(M)_{/R'} + m\,\vec{a}(M)_e + m\,\vec{a}(M)_c \tag{1.6}$$

Ainsi qu'énoncé en introduction, $\vec{a}(M)_e$ constitue l'accélération d'inertie d'entraînement et $\vec{a}(M)_c$, l'accélération d'inertie de CORIOLIS, définies aux équations (1.7) et (1.8).

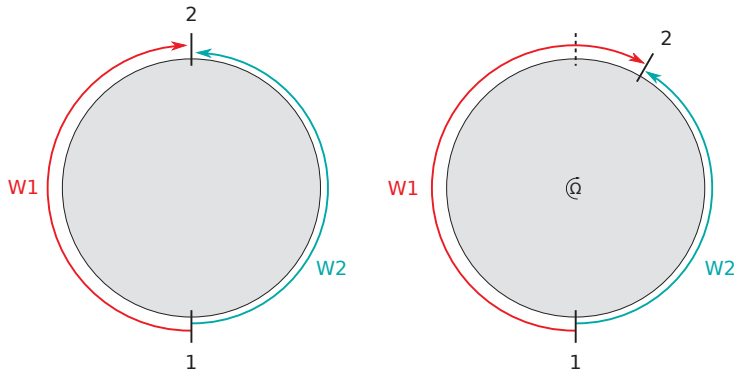


FIGURE 1.6 – Illustration de l'effet SAGNAC : deux signaux lumineux qui partent dans des directions opposées présentent une différence de marche lorsqu'ils se rejoignent si le mobile en leur centre entre en rotation

$m\,\overrightarrow{a}(M)_e$ et $m\,\overrightarrow{a}(M)_c$ constituent des forces fictives, la force d'inertie d'entraînement et la force d'inertie de CORIOLIS respectivement.

$$\overrightarrow{a}(M)_e = \overrightarrow{a}(O')_{/R} + \overrightarrow{\omega}_{R'/R} \wedge \left(\overrightarrow{\omega}_{R'/R} \wedge \overrightarrow{O'M}\right) + \frac{d\overrightarrow{\omega}_{R'/R}}{dt} \wedge \overrightarrow{O'M} \tag{1.7}$$

$$\overrightarrow{a}(M)_c = 2\,\overrightarrow{\omega}_{R'/R} \wedge \overrightarrow{v}(M)_{R'} \tag{1.8}$$

Ces forces fictives permettent de concevoir des gyromètres qui exploitent leurs effets en vue de donner une mesure des rotations subies.

1.2.2 Gyromètres vibrants à effet Coriolis

Avec l'essor des micro et maintenant nano-technologies et la capacité à concevoir des micro assemblages ou des structures mécaniques monolithiques aux dimensions réduites, les études se sont penchées sur la conception de gyromètres dits mécaniques, qui présentent des dimensions très réduites comparées aux gyromètres à effet SAGNAC. Ces micro-capteurs exploitent les forces d'inertie en vue de fournir une mesure de la rotation subie par le mobile qu'ils équipent. En première approximation, dans les cas habituels d'utilisation de ces capteurs, la force d'inertie de CORIOLIS (de l'ordre de $10^{-1}\,\mathrm{m/s^2}$) prédomine devant la force d'inertie d'entraînement (de l'ordre de $10^{-3}\,\mathrm{m/s^2}$). En vue de conserver des dimensions réduites, les oscillations mécaniques d'un résonateur s'imposent comme solution de conception, d'où l'appellation de gyromètres vibrants à efffet CORIOLIS.

Il existe plusieurs types de capteurs qui correspondent à plusieurs types de résonateurs, chacun avec leurs avantages et inconvénients. Dans tous les cas, ces capteurs offrent des possibilités de miniaturisation et d'intégration qui vont de paire avec des coûts de production réduits par une production en masse.

Le nombre d'axes sensibles, les moyens d'actionnement et de détection permettent de répartir ces capteurs en catégories (mais d'autres classifications existent). Les capacités

technologiques de la filière silicium ont favorisé le développement des capteurs à actionnement et détection électrostatiques (capteurs dits capacitifs). Mais le silicium n'est pas le seul matériau disponible : le quartz permet de concevoir des capteurs à actionnement et détection piézoélectriques, détection qui se montre moins sensible à l'environnement qu'une détection capacitive (sensible à n'importe quelle variation de capacité).

Le gyromètre du Drapper Laboratory [GBKN91] constitue l'un des premiers gyromètres vibrants et repose sur un cadre oscillant. Un actionnement électrostatique excite le cadre à sa fréquence de résonance. L'excitation de ce cadre entraîne l'oscillation d'une plaque reliée mécaniquement au cadre et qui constitue l'élément sensible du capteur. En présence d'une rotation, cette plaque va induire un mouvement sur un axe perpendiculaire à l'axe d'excitation du cadre, l'axe de détection. Un système de détection capacitive mesure alors les mouvements autour de cet axe en vue de fournir une mesure de la vitesse de rotation.

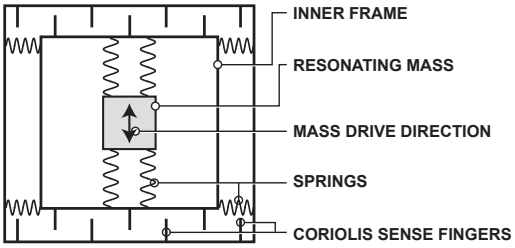
Depuis le début des années 2000, Analog Devices a développé une série de gyromètres basés sur leur technologie iMEMS® [GK03]. La figure 1.7 présente la structure et le principe de fonctionnement des capteurs basés sur cette technologie. Un système se charge de maintenir en résonance la masse située au centre. Elle présente ainsi un mouvement selon l'axe des ressorts par laquelle elle est rattachée au cadre intérieur. En présence d'une rotation, les oscillations de la masse se font également perpendiculairement à cet axe, entraînant avec elle une oscillation du cadre intérieur, qui se trouve, lui, rattaché au cadre extérieur par des ressorts. Des détecteurs capacitifs placés entre les deux cadres détectent alors ce mouvement et délivrent une mesure de la vitesse de rotation subie. Le gyromètre ADIS16136 forme le fer de lance de leur gamme et offre les meilleures performances de la gamme (voir le tableau 1.8).

Sur un autre principe, la société Silicon Sensing propose un gyromètre en silicium basé sur une structure en anneau, le CRS39. En 2001, l'université du Michigan avait exposé une telle structure [AN01], dans la continuité des travaux de Putty et Najafi. En France, Thales a présenté un micro-gyromètre en 2009 que le tableau 1.8 reprend.

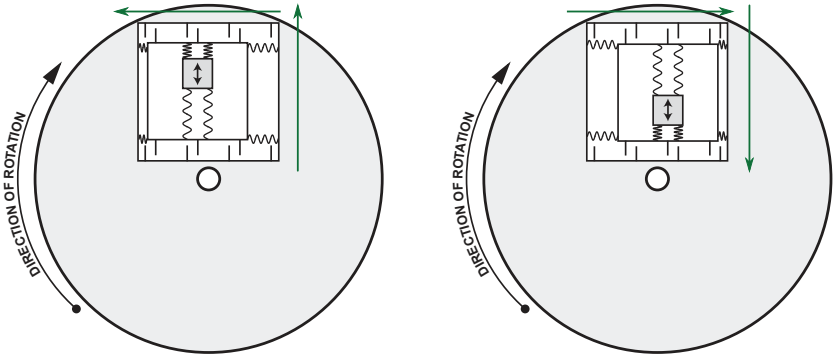
Enfin, une société norvégienne, Sensonor, développe un gyromètres avec une structure en papillon (présentée dans la figure 1.8), le SAR500. Le tableau 1.8 présente les performances escomptées pour ce capteur que Sensonor ne propose pas encore à la vente [LBHK].

Sur son site internet, le JPL [5] annonce disposer du gyromètre MEMS le plus performant au monde [Lab]. Il reposerait également sur une structure en anneau, gravée en silicium. Une autre gravure d'une structure légèrement modifiée sur substrat de quartz cette fois-ci, permettrait même d'atteindre de meilleures performances. Toutefois, ces travaux semblent faire l'objet de restrictions de publication puisque ce gyromètre aurait une utilisation militaire. Difficile donc de connaître ses performances réelles.

5. Jet Propulsion Laboratory ; laboratoire de recherche commun entre la NASA et Caltech.



(a) Schéma de principe de la technologie iMEMS®



(b) Illustration du fonctionnement d'un gyromètre iMEMS®

FIGURE 1.7 – La technologie iMEMS®

Cependant, le choix du quartz comme matériau n'en reste pas moins intéressant. En effet, tous ces gyromètres reposent sur des systèmes oscillants. Or le quartz demeure le matériau de référence dans ce domaine grâce à sa stabilité et sert par exemple de base de temps dans des horloges.

La société Systron Donner exploite depuis une vingtaine d'années maintenant un gyromètre en quartz avec une structure en double diapason telle que la présente la figure 1.9 [MC01]. Un système électronique maintient un premier diapason en vibration à sa fréquence de résonance, dans le plan du diapason comme indiqué sur la figure 1.9 ; il s'agit de la vibration pilote. Sous l'effet d'une rotation, la force d'inertie de CORIOLIS va provoquer un mouvement hors plan du deuxième diapason. Des électrodes se chargent ensuite de dé-

	Étendue de mesure	**Bande passante**	**Stabilité de biais**	**Bruit en sortie**	**Angular Random Walk**
ADIS16136	±450 °/s	380 Hz	4 °/h	0,11 °/s rms	0,167°/$\sqrt{\text{h}}$
CRS239	±25 °/s	25 Hz	0,2 °/h	0,05 °/s rms	0,03°/$\sqrt{\text{h}}$
Thales	±1000 °/s	50 Hz	8 °/h	0,007 °/h rms	0,01°/$\sqrt{\text{h}}$
QRS116	±100 °/s	60 Hz	3 °/h		0,035°/$\sqrt{\text{h}}$
SAR500	±500 °/s		0,04 °/h (in-run)		0,002°/$\sqrt{\text{h}}$

TABLE 1.8 – Performances de quelques gyromètres MEMS du marché

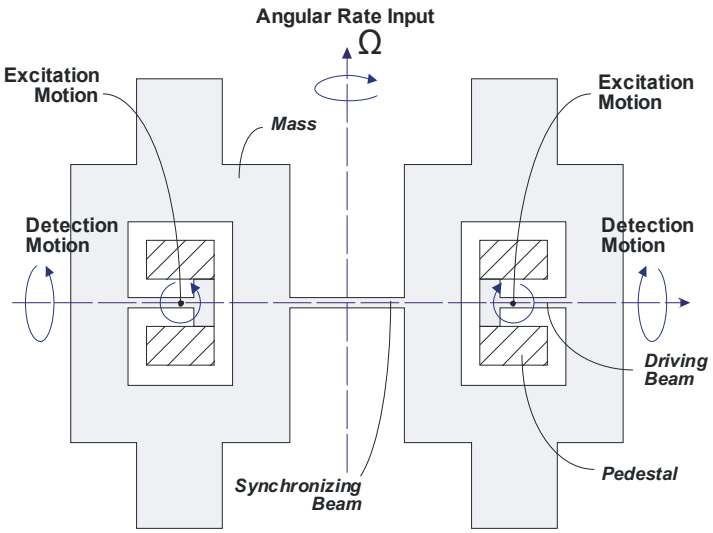


FIGURE 1.8 – Structure et principe du gyromètre de Sensonor

tecter ce mouvement. Cette structure en double diapason permet d'isoler les électrodes dédiées à l'excitation d'une part, et celles dédiées à la détection d'autre part. Le QRS116 offrent les meilleures performances de la gamme avec cette structure, performances présentées dans le tableau 1.8.

Le VIG de l'Onera

La problématique pour supprimer le diapason qui sert à la détection se pose au niveau des électrodes. Difficile en effet de placer sur un même diapason les électrodes d'excitation et celles de détection ; mais pas impossible puisque l'Onera a déposé le brevet d'un système d'électrodes imbriquées qui permet de placer efficacement sur un même diapason les électrodes d'excitation et de détection. L'Onera développe ainsi depuis une dizaine d'années un micro-gyromètre vibrant en quartz, le Vibrating Integrated Gyro (VIG).

La figure 1.10 présente ce micro-gyromètre en quartz. Cette fois-ci, un seul diapason suffit, ce qui réduit de moitié l'encombrement par rapport au système à double diapason. La figure 1.11 présente les modes de vibrations du VIG : les électrodes excitent le mode pilote par effet piézoélectrique indirect alors que la détection des vibrations du mode détecteur se fait par effet piézoélectrique direct. Comme toujours, la force d'inertie de CORIOLIS provoque le mouvement du mode détecteur en présence d'une rotation. Le rapport entre l'amplitude du mouvement du mode détecteur et celui du mode pilote définit le facteur d'échelle.

Enfin, le tableau 1.9 présente les performances actuelles du VIG.

À noter également que l'Onera a financé une thèse sur la conception d'une structure de gyromètre tri-axe [Rol12]. L'étude a démontré la faisabilité mécanique d'un unique mode pilote couplé par effet Coriolis selon les trois axes à trois modes de détection distincts.

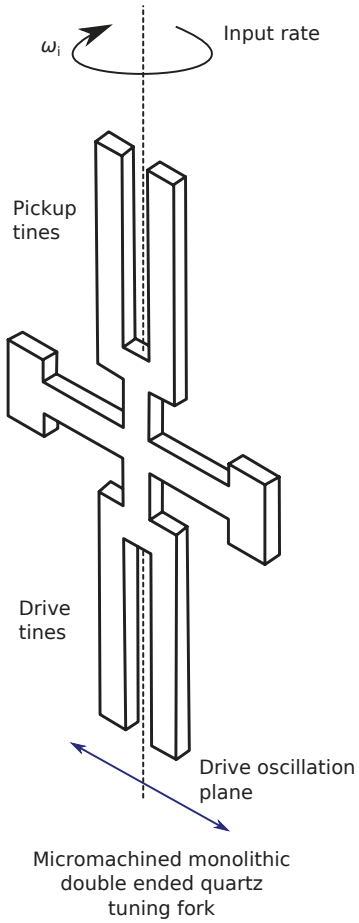


FIGURE 1.9 – Structure du QRS de Systron Doner

	Étendue de mesure	**Bande passante**	**Densité spectrale de bruit**	**Stabilité de biais**
VIG	100 Hz	$\pm 500\,°/s$	$3\,°/h\sqrt{Hz}$	$20\,°/h$

TABLE 1.9 – Performances du VIG de l'Onera

1.3 Introduction aux oscillateurs électroniques

Le travail de thèse rapporté ici vise à concevoir une électronique numérique pour les capteurs vibrants développés à l'Onera, le VIA et le VIG. Aussi, puisqu'il s'agit de capteurs vibrants, cette électronique devra avant tout maintenir les capteurs en résonance. Autrement dit, cette électronique devra constituer un oscillateur dans lequel un capteur formera le résonateur. Dans un second temps, elle se chargera d'exploiter les données inertielles issues des capteurs pour fournir une mesure de la grandeur d'entrée, accélération ou rotation.

Avant de présenter les moyens nécessaires à ces objectifs, cette section présente un rapide rappel sur les oscillateurs en général ainsi qu'un bref rappel sur l'oscillateur auto-entretenu, solution également étudiée pour ces capteurs dans le cadre d'une autre thèse.

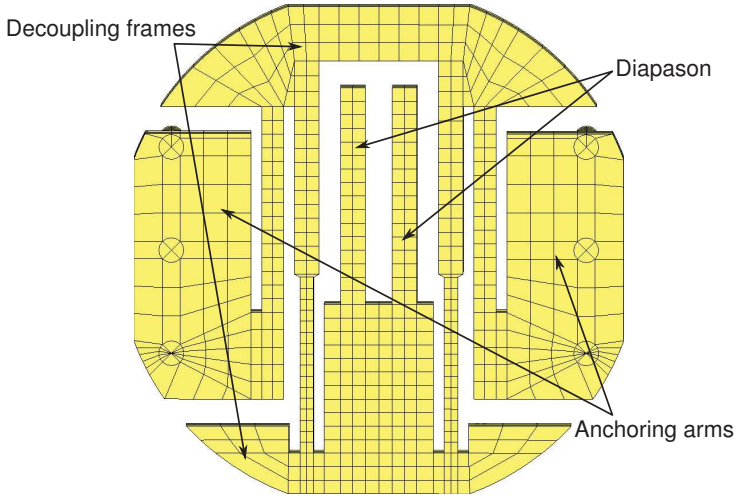


FIGURE 1.10 – Une cellule VIG

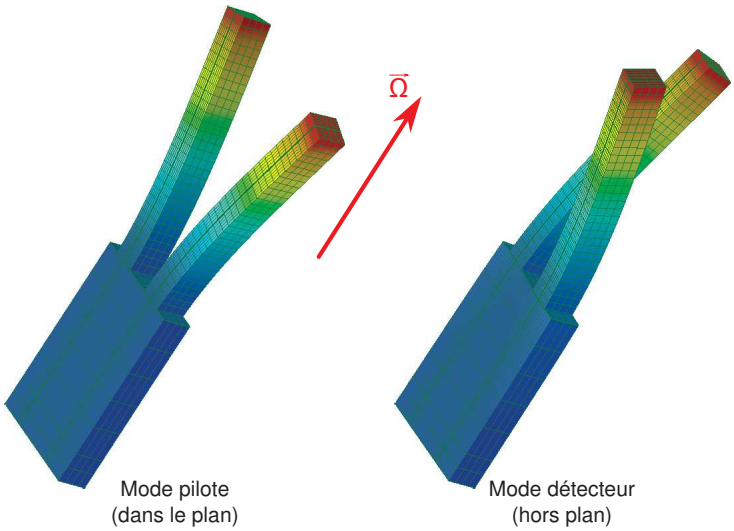


FIGURE 1.11 – Mode de vibration pilote et détecteur du diapason du VIG

1.3.1 Bref rappel du fonctionnement d'un oscillateur

Un oscillateur forcé, cas de cette étude, peut se décomposer en deux parties :

— le résonateur d'une part ;
— un dispositif chargé d'exciter le résonateur d'autre part.

Dans le cas général, le résonateur peut être de n'importe quel type : mécanique, électrique, électro-magnétique, chimique... Sa particularité provient de la non linéarité de sa réponse à une excitation harmonique. Pour un système à un degré de liberté, comme par exemple la tension pour un résonateur électrique, l'équation de la fonction de transfert du résonateur peut s'écrire sous la forme de l'équation (1.9). Dans cette équation, Q représente le facteur de qualité du résonateur, ω_0 sa pulsation propre et ω la pulsation courante.

$$H(\omega) = \frac{1}{1 - \left(\frac{\omega}{\omega_0}\right)^2 + jQ\frac{\omega}{\omega_0}} \tag{1.9}$$

Une fois le résonateur chargé avec une énergie, il va naturellement osciller. Ces oscillations correspondent à un échange régulier entre énergie potentielle (énergie élastique dans la poutre $\frac{1}{2}kx^2$) et énergie cinétique (mouvement $\frac{1}{2}mv^2$). Sans système d'excitation forcée, les oscillations dans un résonateur réel finissent toujours par s'arrêter à cause de l'amortissement (traduit par le facteur de qualité Q dans l'équation (1.9)). Dans un oscillateur forcé, le système d'excitation dicte la fréquence mais l'objectif reste d'exciter le résonateur à sa fréquence de résonance, le système d'excitation apportant alors l'énergie nécessaire pour compenser les pertes provoquées par les frottements, les résistances...

La solution la plus simple pour former un oscillateur à partir d'un résonateur consiste à boucler le système avec un gain et un déphaseur : le résonateur fait office de filtre puisque les excitations à sa fréquence de résonance seront bien plus amplifiées que celles aux autres fréquences. L'amplificateur se charge ensuite de compenser les pertes et le déphaseur, de réinjecter l'excitation à la phase correcte (la réponse du résonateur est déphasée par rapport à l'excitation). Avec les notations de la figure 1.12, une condition s'avère toutefois nécessaire afin que l'oscillateur puisse démarrer : c'est la condition de Barkhausen [GIO13] présentée aux équations (1.10).

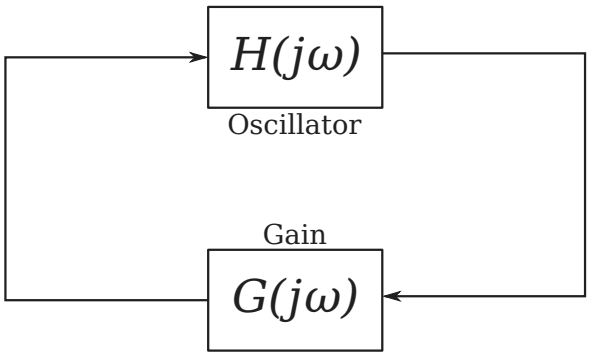


Figure 1.12 – Schéma bloc d'un oscillateur auto-entretenu

$$\left.\begin{aligned} |G(j\omega)\,H(j\omega)| &= 1 \\ Arg\,(G(j\omega)\,H(j\omega)) &= 0 \end{aligned}\right\} \tag{1.10}$$

Dans le cas d'un oscillateur forcé, la condition de phase n'existe plus puisque le système d'excitation impose la phase. En revanche, la condition sur l'amplitude demeure et stipule que le système d'excitation doit injecter suffisamment d'énergie pour compenser les pertes.

Sinon, les oscillations s'arrêtent.

1.3.2 Présentation d'oscillateurs numériques à résonateurs analogiques

Tout oscillateur vise à fournir une fréquence de référence. En électronique analogique, le signal de sortie se présente la plupart du temps sous forme sinusoïdale. En électronique numérique, deux cas se distinguent :

— les oscillateurs entièrement numériques qui délivrent un signal en créneaux ;
— les oscillateurs qui utilisent un résonateur analogique piloté par une électronique numérique.

Pour le premier cas, les oscillateurs en anneaux, enchaînement d'un nombre impair d'inverseurs, un des composants les plus simples de l'électronique numérique, génèrent justement des signaux en créneaux à la fréquence souhaitée. Une électronique supplémentaire permet de sélectionner le nombre d'inverseurs actifs dans la boucle et on obtient un oscillateur entièrement numérique dont la fréquence de sortie peut être ajustée : un DCO, Digitally Controlled Oscillator [EML11] (plus il y a d'inverseurs, plus la fréquence est faible).

Pour le second cas, si le contenu spectral du signal de sortie importe peu, un simple filtrage d'un signal carré peut suffire. Dans le cas contraire, il existe des systèmes entièrement numériques capables de créer des signaux sinusoïdaux à partir d'une horloge de référence, les synthétiseurs de fréquences. Ainsi, des architectures entièrement numériques peuvent exciter des systèmes analogiques, comme des résonateurs en quartz par exemple. Dans l'optique de former un oscillateur numérique à partir d'un résonateur analogique externe, il manque encore une boucle de retour, capable de mesurer la phase de la sortie du résonateur. Les compteurs numériques peuvent jouer ce rôle et fournir en plus les outils nécessaires à une mesure de fréquence (utile pour mesurer une accélération avec un VIA). Enfin, un système traite les données reçues du (des) compteur(s) et calcule l'excitation (domaine d'excellence de l'électronique numérique) en vue de maintenir le résonateur en résonance.

Au final, cet ensemble excitation, mesure de phase/fréquence, système de traitement reprend les blocs constitutifs d'une PLL analogique, mais entièrement numérique : une Digital Phased Locked Loop, DPLL. La figure 1.13 reprend ce schéma de principe.

La synthèse de fréquence

Les mixeurs de fréquences Les premières techniques de synthèse de fréquence utilisaient les harmoniques d'un signal ou de deux signaux de référence [Noo69]. En fait, il s'agissait d'adaptations ou d'améliorations de systèmes analogiques. L'électronique numérique, par la mise en place de boucles et de diviseurs, améliorait grandement les capacités

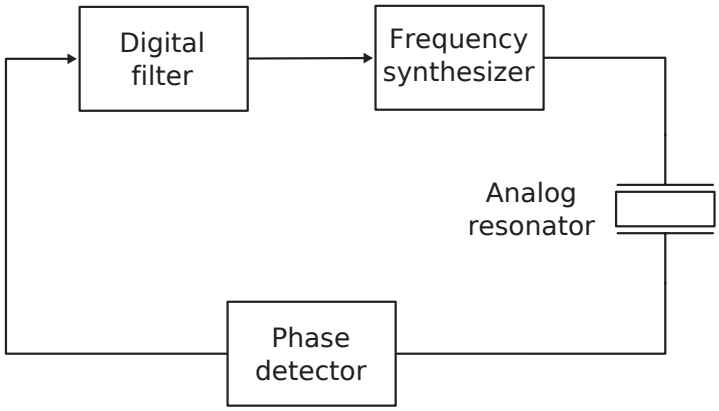


FIGURE 1.13 – Schéma blocs d'un oscillateur numérique avec un résonateur analogique

de ces systèmes. Elle rendait par exemple possible la formation de sous-multiples entiers des fréquences de sorties des systèmes purement analogiques (1/2, 1/3, 1/4 ...).

Un premier cas, dit de multiplication de fréquence, appliquait un fenêtrage à un signal de référence de sorte à le déformer très fortement (transformé en signal carré par écrêtage). Le signal ainsi obtenu présentait un contenu spectral riche en harmoniques du signal initial. Un jeu de filtres analogiques passe-bandes sélectionnait ensuite l'harmonique à la fréquence désirée. L'utilisation d'un diviseur fractionnaire dans une boucle (voir [Noo69, pp 261–262]) rendait accessibles des fréquences presque quelconques autour des harmoniques. Le lecteur comprendra rapidement tout de même les limitations de ce système, qui réside dans le compromis à faire entre fréquence de référence et fréquences de sorties accessibles. En effet, puisque le signal de sortie résulte d'un filtrage des harmoniques, qu'il demeure impossible de disposer d'une infinité de filtres, et que le rapport signal sur bruit des harmoniques diminue avec la fréquence (ce qui rend inutilisables les harmoniques d'ordres élevés), la fréquence maximale de sortie est limitée.

Une variation de ce système évitait la distorsion par fenêtrage et mixait deux fréquences de référence avec une électronique analogique. Un mixeur de fréquence, en plus de délivrer des signaux aux fréquences somme et différence des signaux d'entrée, fournit aussi tous les harmoniques $nf_1 \pm mf_2$, ou n et m sont des entiers. Puis une fois de plus, un ensemble de filtres analogiques permettait de sélectionner la fréquence de sortie.

Dans tous les cas, ces systèmes restaient des systèmes hybrides analogique/numérique dans lesquels le numérique ne faisait qu'améliorer les performances de l'électronique analogique.

La synthèse numérique directe La véritable évolution qui s'affranchit de toute électronique analogique consiste à utiliser le signal d'horloge comme référence de temps pour calculer la phase courante et en déduire l'amplitude analogique à chaque front montant d'horloge. Ce signal d'horloge ne passe plus au travers d'une série de filtres pour obtenir la sortie, il marque simplement les instants d'échantillonage. C'est la synthèse de fréquence

numérique directe, ou Direct Digital frequency Synthesis (souvent raccourcie en Direct Digital Synthesis), DDS, en Anglais [TRG71].

Pour comprendre le fonctionnement de ces systèmes, il faut revenir à la définition mathématique des sinus/cosinus. Il s'agit de fonctions qui, à une phase entre 0 et 2π (modulo 2π), associent un nombre réel entre -1 et 1. Pour une fréquence donnée, la phase évolue linéairement en fonction du temps et se définit par $2\pi ft$, où t est le temps et f la fréquence souhaitée. Ainsi, une DDS utilise sa référence de temps pour calculer la phase courante, puis convertie cette phase en une amplitude numérique, elle même enfin convertie en une tension de sortie analogique.

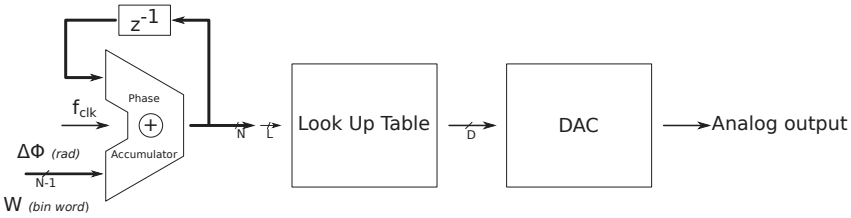


FIGURE 1.14 – Principe de fonctionnement d'une DDS

La figure 1.14 expose ce principe et reprend les constituants d'une DDS en schéma bloc. La commande de la DDS, $\Delta\phi$, représente un incrément de phase et va décider de la fréquence du signal de sortie. L'accumulateur de phase se charge de calculer la phase courante qui va adresser une Look Up Table (LUT), LUT qui convertit ensuite cette phase en une amplitude numérique. Enfin, le convertisseur numérique/analogique se charge de convertir l'amplitude numérique en une tension analogique.

La LUT a longtemps posé problème. Il s'agit d'une mémoire, aujourd'hui composant de petite taille, qui consomme peu et peu cher. Jusqu'à la fin du XX^{e}siècle, tant que les mémoires restaient des composants chers, les DDS n'ont pas véritablement percé. Les ingénieurs et chercheurs ont proposé divers systèmes en vue de réduire [Van97] ou remplacer [LSWD08] cette LUT et de convertir une phase en une amplitude de sinus. Aujourd'hui, les avancées technologiques ont balayé ce problème.

La section 2.3 s'attachera à décrire plus en détail ces systèmes et proposera une étude spectrale de la sortie en fonction des divers paramètres qui définissent une DDS.

Introduction au comptage de fréquence et à la détection de phase

Afin de piloter le synthétiseur de fréquence et d'ajuster la fréquence d'excitation du résonateur, la logique de décision doit connaître la fréquence et la phase (afin de détecter la résonance) de son signal de sortie.

Mesurer des fréquences constitue l'un des besoins fondamentaux de l'électronique, aussi bien analogique que numérique. Il existe de multiples techniques pour mesurer une

fréquence en électronique numérique. L'immense majorité d'entre elles reposent sur des compteurs, qui n'ont pas attendu l'essor de l'électronique numérique pour se développer. Des compteurs électriques numériques sont apparus au début du XX^{ème}, bien souvent basés sur des condensateurs et des triodes [Bli49].

Dans les années 1930, l'idée d'un composant qui pourrait remplacer les triodes voit le jour [Lil30] [Lil33]. Ce composant utiliserait des semi-conducteurs et présenterait des dimensions réduites et une consommation électrique plus faible. Mais ce n'est qu'en 1946 que le transistor voit le jour au sein des laboratoires Bell [Sho51] et offre à l'électronique numérique l'outil dont elle avait besoin.

Outre leur utilisation dans la conception de circuits de logique combinatoire, les transistors servent également à concevoir des bascules [IK77], base de l'électronique synchrone et donc des compteurs, dont l'utilisation en comptage de fréquence s'avère aujourd'hui extrêmement répandue. Dans un premier temps, les techniques de comptage demeurent simples : le compteur compte le nombre de périodes du signal à mesurer en un temps donné, 1 seconde par exemple. Cette technique rudimentaire se révèle assez limitée et d'autres techniques, toujours basées sur des compteurs vont progressivement voir le jour. La section 2.1 reviendra plus en détails sur les limitations de cette première méthode, ainsi que sur les évolutions de la mesure de fréquence par compteur.

Ces évolutions, qui résident essentiellement dans la manière d'utiliser le (ou les) compteurs utilisent et motivent en parallèle les évolutions technologiques. Les compteurs les plus simples reposent sur des bascules T et demeurent asynchrones [Sin06, pp 67–73]. Une bascule T offre relativement peu de fonctionnalités. C'est pourquoi d'autres architectures basées notamment sur des bascules JK ont vu le jour, elles permettent de choisir le mode (comptage ou décomptage), offrent des possibilités de pré-chargement et/ou encore de remise à 0. Toutefois, le caractère asynchrone de ces compteurs empêche de compter des grandes valeurs : le temps de propagation total dépend du nombre de bascules, nombre qui détermine aussi la valeur maximale du compteur. Si un compteur dispose de N bascules, avec chacune un temps de propagation de t secondes, alors le compteur peut compter jusqu'à $2^N - 1$ avant de reboucler à 0 et le temps de propagation total vaut $N \times t$. Ce qui signifie que la plus haute fréquence mesurable avec un tel compteur vaut $\frac{1}{N \times t}$.

Avec un système de comptage simple où le compteur compte le nombre de périodes du signal à mesurer pendant un temps donné, la précision de la mesure dépend de la fréquence du signal à mesurer et donc du nombre de périodes mesurées (la section 2.1 présente le détail des calculs). Or, pour mesurer des hautes fréquences, il faut un temps de propagation faible et donc peu de bascules. Mais dans ce cas, la résolution devient limitée du fait du petit nombre de périodes comptées.

Ces limitations des compteurs asynchrones ont poussé vers l'emploi de compteurs synchrones. Ces dispositifs plus complexes réduisent le temps de propagation total au temps

de propagation d'une seule bascule [Sin06, pp 75–77]. Il devient ainsi possible de concevoir des compteurs avec beaucoup de bascules qui offrent une bonne résolution et demeurent capables de mesurer des fréquences élevées. L'annexe B.2 présente une architecture de compteur binaire synchrone moderne.

En outre, les compteurs interviennent aussi dans les systèmes numériques de détection de phase. En effet, dans le cas où un compteur mesure la fréquence d'un signal sinusoïdal, les évènements comptés correspondent bien souvent au passage par 0 du signal depuis une tension négative, c'est-à-dire au passage par 0 de la phase du signal. Autrement dit, le compteur peut aussi servir de détecteur de phase.

Il s'agit là d'une approche simplifiée. Les télécommunications numériques mais aussi les recherches en vue de concevoir des PLL entièrement numériques [LC81] ont grandement contribué au développement de détecteurs numériques de phase, notamment par l'utilisation simultanée de plusieurs compteurs.

1.3.3 Pilotage du résonateur : comparateur de phase, filtre...

Afin de boucler le système pour réaliser l'oscillateur, un détecteur de phase doit mettre en cohérence le signal d'excitation et la réponse du résonateur : la boucle à verrouillage de phase.

Lorsque le signal d'erreur est analogique, un régulateur Proportionnel-Intégral-Dérivé (PID) détermine la commande (c'est-à-dire la fréquence du synthétiseur) qui a pour effet de maintenir à zéro l'erreur de phase.

Le même schéma s'applique lorsque le signal d'erreur est numérique, issu de compteurs de phase par exemple. Un régulateur PID numérique n'est autre qu'un filtre (au sens large), et peut s'implémenter de plusieurs façons :

— en arithmétique câblée – le filtrage numérique met en œuvre des additionneurs et des multiplieurs ;
— en logiciel – solution plus souple car l'architecture de filtrage n'est pas figée, mais nécessite un processeur.

La deuxième solution s'avère évidemment préférable tant que le processeur supporte la quantité de traitements de données dont le flux est à la fréquence de la porteuse. Jusqu'à 1 MHz ou plus selon le composant, un processeur supportera un traitement logiciel, mais au delà, seule de la logique câblée sera efficace, jusqu'à plusieurs centaines de mégahertz.

Dans tous les cas, qu'il s'agisse d'une PLL matérielle ou logicielle, elle nécessite un détecteur de phase, pour lequel l'utilisation d'un compteur tel que présenté auparavant constitue une des options possibles mais pas la seule [LC81]. La différence entre les deux solutions se situe en aval, dans le traitement de l'information issue du détecteur de phase. Dans le cas d'une implantation matérielle, elle passera dans un ensemble d'additionneurs, multiplieurs, diviseurs... instanciation matérielle du filtre désiré. Dans le cas d'une solu-

tion logicielle, le processeur traitera cette information par l'algorithme logiciel avec lequel il aura été programmé.

Toutes choses étant égales par ailleurs, une DPLL[6] matérielle fonctionnera toujours plus rapidement qu'une PLL logicielle ou SPLL (Software PLL) [EAM88]. Toutefois, pour les applications considérées dans le cadre de cette thèse et les technologies actuelles, une SPLL constitue une solution tout à fait envisageable [XWS10].

1.4 Choix d'une architecture pour l'électronique des capteurs

La section précédente a présenté l'ensemble des éléments nécessaires à la réalisation d'une électronique pour les capteurs vibrants de l'Onera. Il reste alors à choisir une architecture. Pour cela, le contexte de l'étude va apporter quelques contraintes qui feront préférer une solution plutôt qu'une autre.

L'Onera souhaite faire de ses capteurs vibrants des capteurs de classe inertielle, c'est-à-dire qui rendent possible la navigation d'un mobile sans système de positionnement global (comme le GPS, Galiléo ou Glonass) ou avec un usage limité. Cela peut par exemple être le cas de satellites, de lanceurs ou d'autres engins spatiaux. De plus, dans le cas d'applications militaires ou spatiales, une électronique dédiée doit suivre un cycle de qualification en vue de s'assurer qu'elle pourra résister à des environnements sévères (radiations solaires, vents solaires... dans le cas du spatial).

Une DPLL logicielle présente l'avantage de reposer sur une architecture électronique figée (qui peut d'ailleurs comporter plusieurs circuits entre lesquels le logiciel bascule) et de permettre des mises à jour en vue d'améliorer les performances des capteurs ou de corriger les défauts des capteurs ou de l'électronique elle même, défauts qui pourraient apparaître avec le temps. Ceci favorise donc la conception d'une électronique centrée autour d'un microprocesseur.

En outre, les capteurs et leurs systèmes de fixation présentent de multiples modes de vibration dont les fréquences exactes varient d'un capteur à l'autre. Par conséquent, il s'avère préférable de maîtriser le contenu spectral des signaux d'excitation envoyés aux capteurs afin de ne pas exciter un mode non souhaité. Une DDS permet justement ceci, par la maîtrise de son spectre de sortie obtenu par l'ajustement de ses paramètres de conception (section 3.6).

Nous avons donc choisi une architecture numérique telle que présentée à la figure 1.15. Le processeur fournira la commande de la DDS à l'aide des informations fournies par le compteur. Ces informations serviront également à calculer l'accélération mesurée par

6. Digital Phase Locked Looped

l'accéléromètre. Le processeur se chargera aussi de mettre en trames les mesures, c'est-à-dire de distribuer ces données à un ordinateur de bord au format attendu.

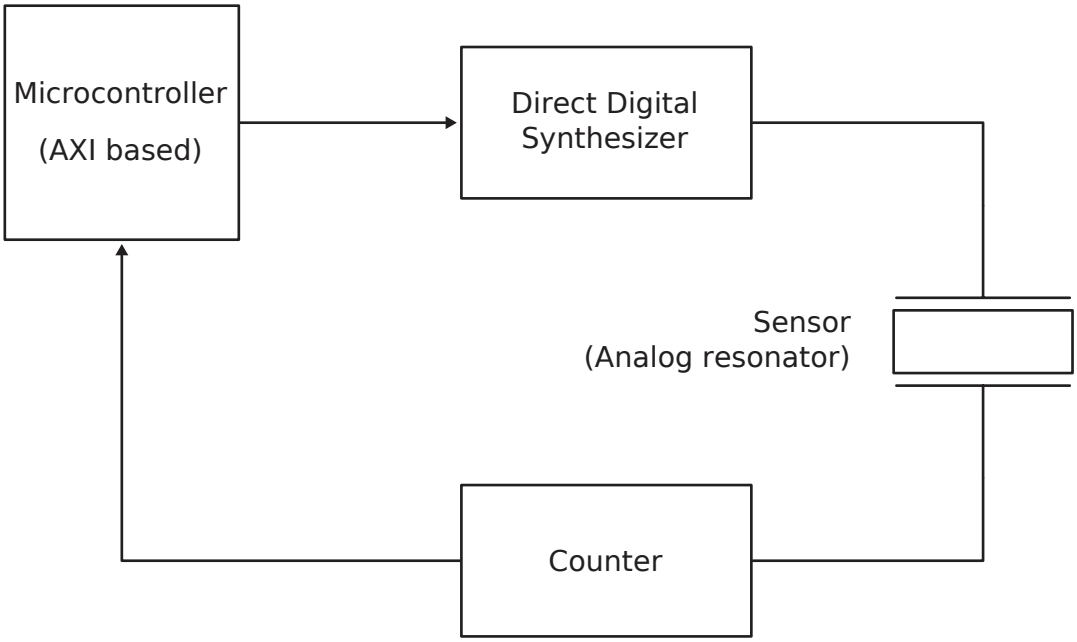


FIGURE 1.15 – Architecture numérique retenue pour les capteurs vibrants de l'Onera (les amplificateurs de charges ne sont pas représentés, ils se situent juste avant et juste après le capteur)

Chapitre 2

Conception et dimensionnement du compteur de fréquence et du synthétiseur de fréquence

Sommaire

2.1 La mesure de fréquence en électronique numérique 32

2.1.1 Mesure par «gates» 32

2.1.2 Comptage réciproque 33

2.1.3 Mesure de fréquence par marquage temporel 37

2.1.4 Résolution et stabilité de la mesure de fréquence : la variance d'Allan 39

2.2 Exploitation des données du marquage temporel pour la mesure d'accélération 41

2.3 Formation d'un signal sinusoïdal en électronique numérique . 45

2.3.1 Paramètres de conception d'une DDS 45

2.3.2 Contraintes entre les paramètres 46

2.3.3 Influence de la fréquence d'échantillonnage sur le spectre de sortie 49

2.3.4 Influence de l'accumulateur de phase sur le spectre de sortie . . 50

2.3.5 Influence de la LUT 55

2.3.6 La conversion numérique-analogique 66

2.3.7 Conclusion sur le dimensionnement d'une DDS 75

2.1 La mesure de fréquence en électronique numérique

La section 1.3.2 a rapidement évoqué les méthodes de comptage de fréquence. Cette partie explique plus en détail différentes méthodes avec leurs avantages et inconvénients en vue de choisir laquelle appliquer aux VBA.

En outre, cette partie considère le comptage d'une fréquence d'un signal carré ou d'une sinusoïde. Un comparateur logique peut toujours être utiliser pour transformer un signal sinusoïdal en un carré, du moment que le retard entre le passage à phase nulle de la sinusoïde et le front montant du créneau demeure fixe dans le temps (sinon, le comparateur introduit un jitter qui n'est pas lié au signal mesuré ni au système qui effectue la mesure).

Subséquemment, un évènement désignera soit le front montant d'un signal carré, soit le passage par zéro d'une sinusoïde croissante (on suppose ici une alimentation symétrique pour les portes logiques).

2.1.1 Mesure par «gates»

La méthode canonique de détermination d'une fréquence par comptage, appelée comptage simple, consiste à compter le nombre d'évènements dans un intervalle de temps donné [Joh05]. Cet intervalle de temps s'appelle «fenêtre temporelle», ou *gate*. Quelle que soit la fréquence du signal à mesurer, l'incertitude sur le nombre de périodes comptées vaut toujours ± 1. Pour un gate de durée T_G et M évènements comptés, ainsi que le représente la figure 2.1, l'estimateur de la fréquence mesurée vaut alors :

$$\hat{F} = \frac{M}{T_G} \tag{2.1}$$

L'incertitude de l'ordre de 1 sur le nombre de périodes dans l'intervalle de mesure fournit directement la résolution de cette technique de mesure : $\frac{1}{T_G}$, exprimée en Hz, et indépendante de la fréquence mesurée. Ce qui conduit à l'erreur relative et à son estimateur :

$$\frac{\delta F}{F} = \frac{1}{T_G \times F} \tag{2.2}$$

$$\frac{\delta \hat{F}}{F} = \frac{1}{T_G \times \hat{F}} \tag{2.3}$$

Évidemment, plus le temps de gate augmente, plus l'erreur relative diminue.

Bien que très simple, cette technique ne permet donc pas de mesurer toutes les fréquences avec la même précision.

Un autre paramètre intervient également dans la mesure : la bande passante, qui

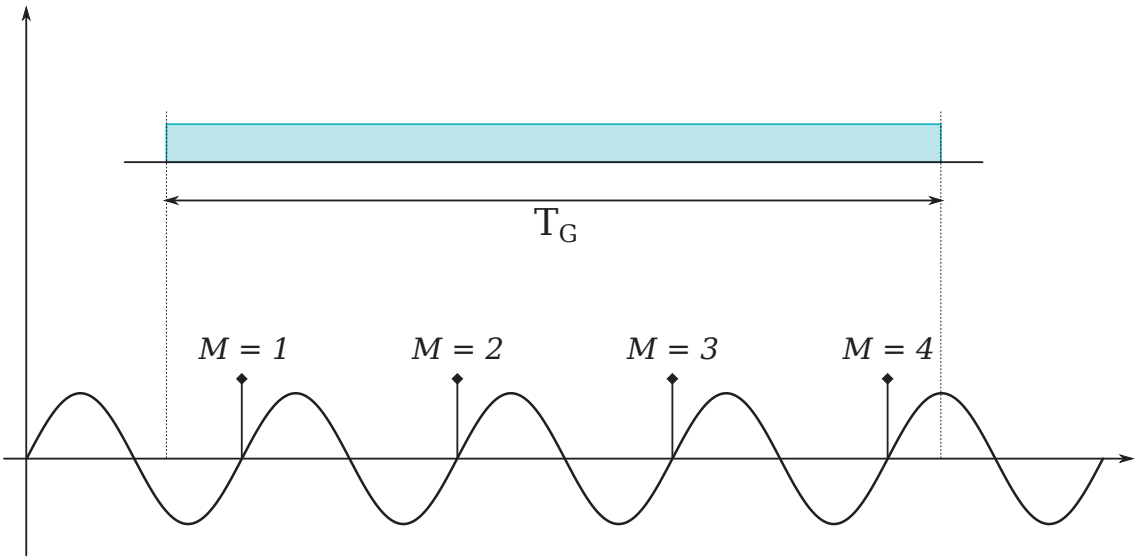


FIGURE 2.1 – Comptage simple d'un signal sinusoïdal

détermine la fréquence maximale à laquelle le système peut mesurer les variations du mesurande, accélérations ou rotations. Or, dans le cas des systèmes de comptage simple, la bande passante de la mesure dépend directement du gate et vaut tout simplement $\frac{1}{2T_G}$. Un gate petit permettra de fournir une bande passante élevée mais dégradera la résolution de la mesure d'autant. Au final, cette méthode naïve de comptage de fréquence s'avère beaucoup trop limitée pour nos besoins. Par exemple, pour une erreur relative de mesure de 10^{-9} (valeur typique des erreurs attendues pour des mesures inertielles) et une fréquence mesurée à 50 kHz, selon l'équation (2.2), il faudrait un gate de :

$$T_G = \frac{1}{10^{-9} \times 50 \times 10^6}$$
$$T_G = 20 \times 10^3 \,\mathrm{s}$$

Soit près de 5 heures et demie. Ce qui ne convient pas à une mesure inertielle.

2.1.2 Comptage réciproque

Le lecteur comprendra qu'il s'avère préférable, avec la méthode présentée dans la section précédente, de mesurer des fréquences élevées.

Afin de rendre la résolution de la mesure indépendante de la fréquence mesurée, le comptage réciproque compte les évènements d'une autre fréquence de référence F_{clk}, la fréquence d'horloge, plus rapide que le signal à mesurer [RS13b]. Un tel compteur mesure un nombre entier de périodes du signal étudié ; le gate ne sert plus à déterminer un intervalle de mesure précis mais plutôt le début et la fin souhaités de la mesure. La figure 2.2 illustre le fonctionnement : lorsque le gate s'ouvre (1), le système attend un évènement du signal à mesurer, puis compte le nombre d'évènements ainsi que le nombre

de fronts de l'horloge de référence. Lorsque le gate se referme (2), le système attend de nouveau un évènement sur le signal d'entrée avant d'arrêter les deux compteurs. À ce moment, les compteurs indiquent le nombre de périodes du signal mesuré, N, et le nombre de fronts de l'horloge de référence, M.

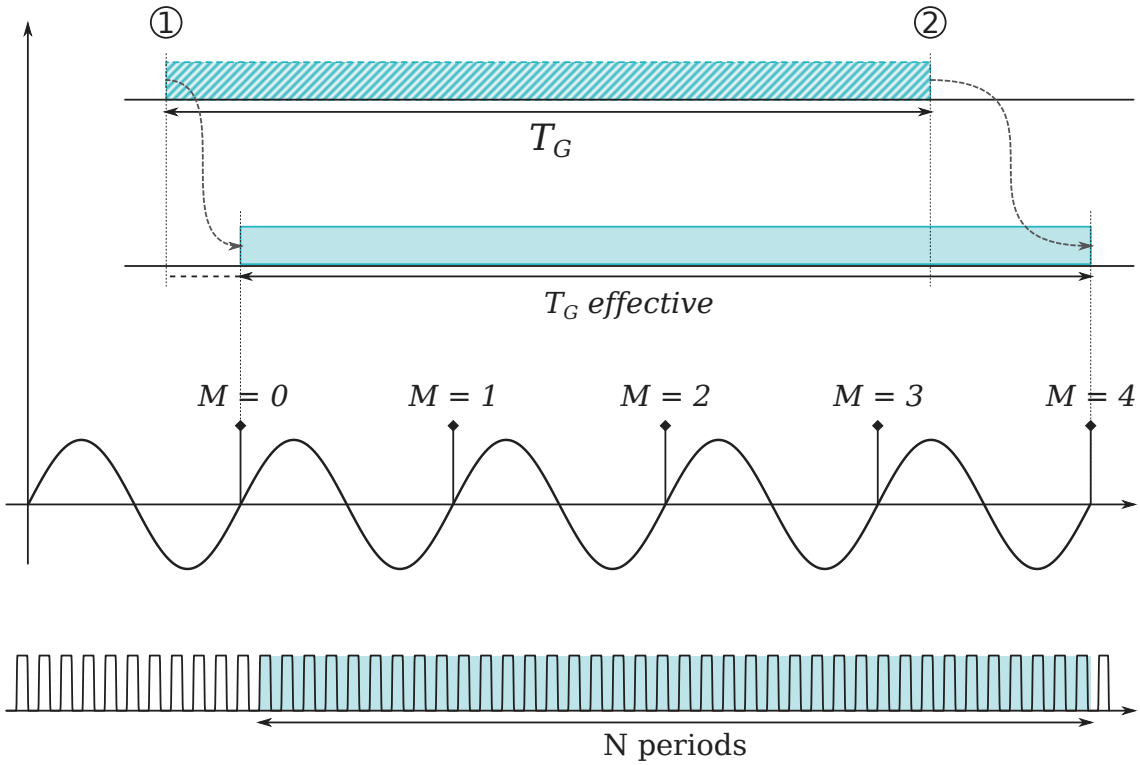


FIGURE 2.2 – Comptage réciproque d'un signal sinusoïdal

L'estimateur de la fréquence mesurée et sa variance s'expriment alors par [RS13b] [Joh05] :

$$\hat{F} = \frac{M}{N} \times F_{clk} \tag{2.4}$$

$$\sigma = \frac{\sqrt{2}}{T_G}\frac{M}{N} \tag{2.5}$$

L'incertitude sur la mesure provient maintenant du nombre de périodes de l'horloge de référence comptées, soit une incertitude de 2 sur N, 1 période au début et 1 à la fin. De plus, le gate effectif dépend de la fréquence mesurée puisqu'il s'ajuste pour obtenir un nombre entier de périodes. Toutefois, pour des fréquences élevées ($T_G \gg T$ où T désigne la période du signal d'entrée), la différence de gate d'une fréquence mesurée à une autre peut-être négligée et l'erreur relative s'exprime alors par l'équation (2.6). Cette erreur relative devient indépendante de la fréquence mesurée et ne dépend plus que du temps de mesure et de la fréquence de référence. Une augmentation du gate ou une augmentation

de la fréquence de référence fournira une mesure avec une meilleure résolution (une erreur relative plus basse).

$$\frac{\delta F}{F} = \frac{2 \times T_{clk}}{T_G} \tag{2.6}$$

En revanche, un tel système nécessite une phase de réinitialisation des compteurs. Tout comme avec la méthode précédente, le gate souhaité peut s'ajuster et ce temps fixe la bande passante de la mesure, ce qui conduit au même inconvénient relatif à la bande passante de la mesure (impossible d'effectuer une mesure à 10 Hz avec un gate de 1 s par exemple).

Cette méthode a fait l'objet d'améliorations en vue d'obtenir des résolutions inférieures à une période d'horloge et de fournir une mesure d'une fraction de période d'horloge [Kal04]. En effet, les évènements qui causent l'ouverture et la fermeture du gate ne sont pas synchrones avec l'horloge de comptage. Par conséquent, il existe un petit délai avant la réaction du système de mesure (l'ouverture ou la fermeture du gate). La mesure de ce délai améliore la résolution qui devient alors inférieure à une période d'horloge.

Il existe des versions analogiques et des versions numériques de ces améliorations. Les versions analogiques reposent sur des rampes de chargement et/ou déchargement de condensateurs : lorsqu'un évènement sur le signal d'entrée survient, un condensateur se charge ou se décharge jusqu'au front d'horloge du signal de référence suivant. La lecture de la tension fournit une mesure d'une fraction de période d'horloge, ce qui améliore la résolution. Cette amélioration dépend de la précision de la conversion analogique-numérique. Si, par exemple, un convertisseur analogique-numérique fournit la lecture du niveau de charge de la rampe sur 10 bits, il améliorerait l'incertitude de la mesure d'un facteur 1024 ($2^{10} = 1024$). Dans la pratique, les erreurs de linéarité de la rampe limitent l'amélioration à un facteur 100. Soit une erreur relative de :

$$\frac{\delta F}{F} = \frac{2 \times \frac{T_{clk}}{100}}{T_G}$$

En règle générale, la conversion analogique-numérique pose un problème de dualité entre résolution et fréquence d'échantillonnage. Toutefois dans le cas présent, puisqu'à la fin du gate, le décompte s'arrête, le système dispose de plus de temps pour échantillonner la tension de la rampe. Un bloqueur d'ordre 0 de bonne qualité (c'est-à-dire qui maintient correctement sa charge le temps de la conversion) laissera suffisamment de temps au convertisseur. Il faudra juste s'assurer que le convertisseur aura le temps de finir sa conversion avant la prochaine mesure.

Les versions numériques se séparent en deux catégories : les systèmes qui utilisent aussi

des horloges et ceux qui reposent sur les temps de propagation des portes ou bascules. Dans le premier cas, le système de comptage utilise en fait trois compteurs : un pour compter les évènements du signal d'entrée, un pour ceux d'un premier signal de référence et un dernier pour ceux d'un second signal de référence, cadencé à 99/100 du premier par exemple. La partie fractionnaire de l'horloge se déduit alors des mesures des deux signaux de référence par le principe du vernier. Cela descend ainsi l'incertitude à 1/100 de T_{clk} dans le cas d'une horloge à 99/100 de l'autre. La difficulté réside cependant dans la fabrication et la stabilité de cette seconde référence à 99/100 de la première fréquence F_{clk}.

Les systèmes qui utilisent les temps de propagation des portes ou bascules permettent de s'affranchir de ce troisième compteur. Pour une technologie, une architecture et un wafer donnés, les temps de propagation sont stables (peu de variations au cours du temps sur une exécution) et répétables (même temps de propagation à chaque exécution sous condition d'environnement identique) dans le temps. Il existe plusieurs variantes de ce principe, mais dans tous les cas, un front lance la propagation d'un évènement dans une chaîne de portes ou bascules. Lors du front de l'horloge de référence suivant, il suffit de lire l'avancement de la propagation. La figure 2.3 présente un tel système. Il s'agit alors d'une mesure de temps absolu. L'erreur relative de la mesure devient indépendante de T_{clk} et s'exprime par :

$$\frac{\delta F}{F} = \frac{2 \times \tau}{T_G} \tag{2.7}$$

Où τ désigne le temps de propagation d'une porte (ou d'un groupe de portes). Bien entendu, ces systèmes disposent de suffisamment de portes pour que le temps de propagation total soit supérieur à T_{clk}.

Aujourd'hui, les technologies modernes permettent de s'affranchir de l'un des plus gros inconvénients des deux méthodes de comptage présentées précédemment, l'arrêt de la mesure entre la fin d'un gate et le début du suivant. La miniaturisation des transistors rend possible la conception de circuits intégrés, réduit ainsi les temps de propagation et autorise la conception de compteurs très rapides sur un grand nombre de bits. Il est ainsi devenu possible d'enchaîner les gates de façon continue. Malgré tout, ces méthodes n'exploitent qu'une partie des informations. En effet, seuls le premier et le dernier évènement servent à déterminer la fréquence du signal mesuré mais il apparaît évident que l'exploitation des mesures des périodes intermédiaires non exploitées pourrait fournir davantage d'information.

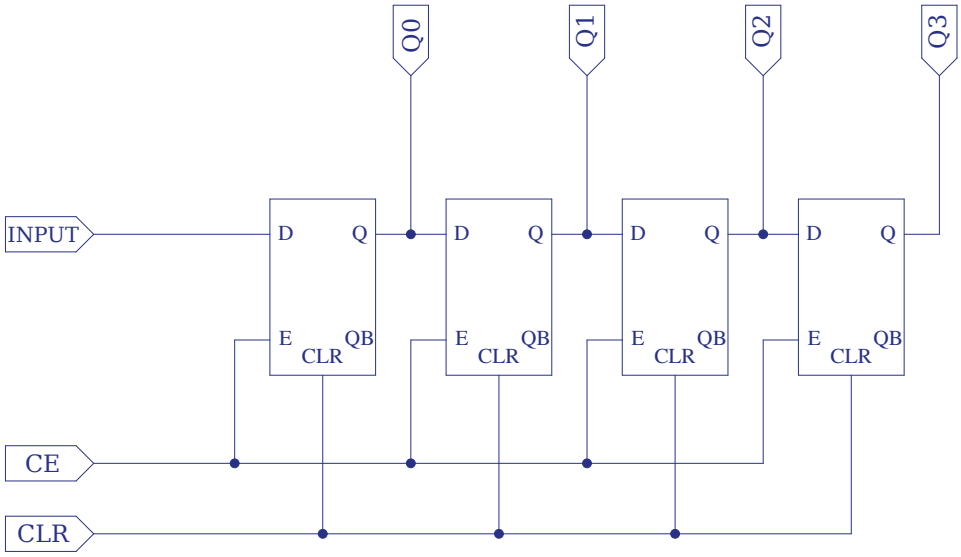


FIGURE 2.3 – Exemple d'une chaîne à retard à bascules D.
Plutôt que d'avoir une sortie toutes les portes, il est possible d'en avoir toutes les n portes.

2.1.3 Mesure de fréquence par marquage temporel

Amélioration du comptage réciproque

Avec l'amélioration des technologies, ce raisonnement a conduit au développement d'une autre technique dite de marquage temporel. Elle repose toujours sur l'emploi d'un compteur cadencé par une horloge rapide, mais le système enregistre la valeur courante du compteur à chaque évènement. Le compteur fonctionne de façon ininterrompue et se remet à 0 une fois sa valeur maximale atteinte. La valeur enregistrée lors d'un évènement sur le signal d'entrée, qui désigne le nombre de périodes du signal d'horloge depuis le premier évènement, prend alors l'appellation de date. D'où l'appellation de marquage temporel.

Une première utilisation de ces dates consiste à appliquer la méthode de comptage réciproque présentée précédemment : la différence date à date fournit le temps écoulé et le nombre de dates capturées le nombre de périodes dans cet intervalle de temps. Si le système a enregistré une date t_k et une date t_{k+m} m périodes du signal d'entrée plus tard, alors avec une fréquence d'horloge F_{clk}, la fréquence du signal s'exprime par l'équation (2.8), similaire à la formule de l'équation (2.4) [RS13a, section 2.1].

$$\hat{F} = \frac{m}{t_{k+m} - t_k} \times F_{clk} \tag{2.8}$$

En comptage réciproque, le marquage temporel permet de fournir davantage de mesures. En effet, puisque le système stocke les dates (dans les limites de sa capacité mémoire), à chaque nouvel évènement sur le signal d'entrée, il devient possible de fournir une nouvelle mesure de fréquence pour un gate donné. Le gate devient glissant ainsi que

le montre la figure 2.4.

Attention toutefois, la bande passante de la mesure reste déterminée par le gate. Cette utilisation en comptage réciproque du marquage temporelle permet juste d'accéder à une mesure plus rapidement puisqu'il n'est plus nécessaire d'attendre la fin d'un gate : à chaque front d'horloge, une mesure est disponible.

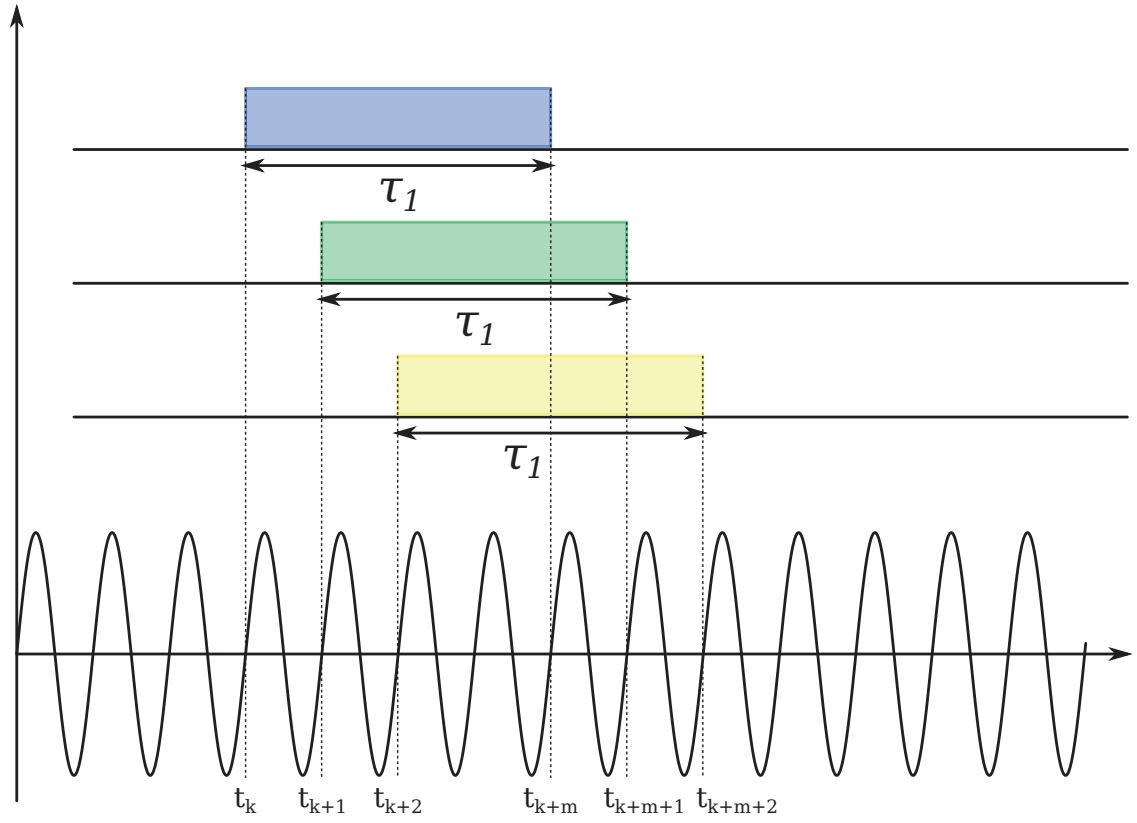


FIGURE 2.4 – Glissement du gate de comptage réciproque en marquage temporel

Le marquage temporel incite ainsi à effectuer des mesures de fréquence non pas toutes les x secondes (définition originelle du gate T_G), mais plutôt à chaque période du signal à mesurer en calculant la fréquence sur les M périodes les plus proches. En outre, puisque le système mémorise les dates des périodes, il peut mesurer la fréquence avec un nombre M de périodes variable, c'est-à-dire avec une bande passante de la mesure variable : à 1 période (mesure instantanée), à 10 périodes, à 1000 périodes... ainsi que l'illustre la figure 2.5. Le lecteur comprendra alors que le gate perd de son sens par rapport aux méthodes présentées précédemment.

Il ne s'agit là que d'une évolution mineure de la technique de comptage réciproque présentée plus tôt, une sorte de comptage réciproque «jointif»rendu possible par l'évolution des technologies. En effet, un tel système nécessite d'effectuer une soustraction afin de déterminer le nombre de fronts d'horloge entre deux évènements du signal mesuré. Or, au début du comptage réciproque, soustraire deux nombres binaires coûtait cher puisque

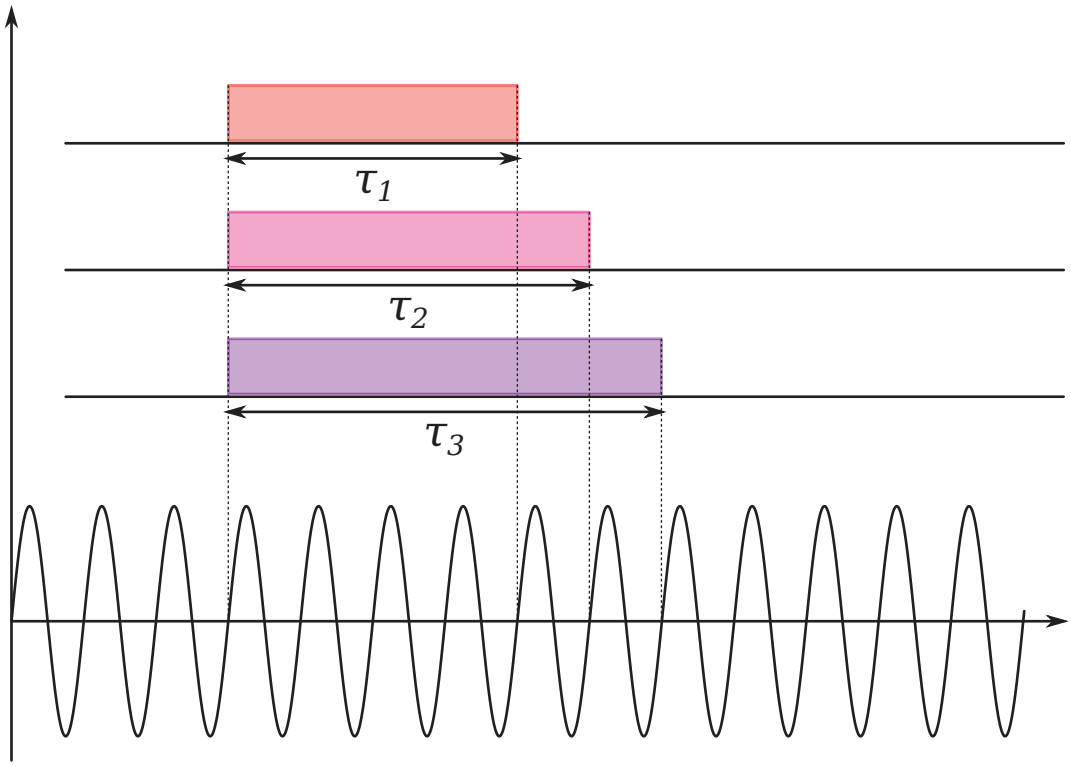


FIGURE 2.5 – Différents intervalles de mesure à l'aide des mêmes données issues du marquage temporel

cela nécessitait une électronique dédiée.

Bien que cette méthode utilise davantage les données disponibles, elle ne les exploite toujours pas comme un tout. La section 2.2 présentera une méthode qui exploitera au maximum ces informations.

2.1.4 Résolution et stabilité de la mesure de fréquence : la variance d'Allan

Les sections précédentes n'ont porté aucune attention au bruit. Or il existe systématiquement, causé soit par l'oscillateur à l'origine, soit par des parasites qui modifient la fréquence instantanée (comme par exemple le signal à 50 Hz de l'alimentation secteur ou bien encore les divers signaux radios). Les architectures numériques considérées ici, en logique séquentielle synchrone, réagissent sur les fronts montants du signal d'horloge, ou plus exactement lorsque la tension de celui-ci dépasse un certain seuil. Par conséquent, tous les bruits qui modifient l'instant t de franchissement de ce seuil induisent une modification de la période du signal et de sa fréquence instantanée. Ceci peut également se voir comme un bruit de phase, ou *jitter* (gigue en Français).

Ainsi, le jitter d'horloge détériore la résolution effective du système de mesure conçu. Afin de qualifier cette imperfection et tenter de quantifier la stabilité du système de

mesure, une première approche consiste à se tourner vers un calcul de la variance de la fréquence d'horloge de référence (celle du compteur). Cependant, cette méthode pose le problème du nombre d'échantillons, limités, alors qu'il en faudrait une infinité pour calculer effectivement la valeur vraie de la variance. [VER12]

Dans le cas d'un nombre fini d'échantillons, d'autres estimateurs existent et l'IEEE a normalisé l'utilisation d'une variance à deux échantillons, ou variance d'Allan [All87], pour l'étude fréquentielle des oscillateurs. L'équation (2.9) fournit l'expression de cet estimateur, noté σ_A. y_2 et y_1 représentent deux échantillons mesurés à un intervalle de temps τ. À noter que lorsque τ tend vers 0, cet estimateur tend vers la variance vraie (c'est-à-dire la variance au sens mathématique, avec un nombre infini d'échantillons). Par ailleurs, cet estimateur ne présente pas de biais pour le bruit blanc dans le cas d'une étude fréquentielle. Il n'en va pas de même pour d'autres bruits ou bien pour des études autres que fréquentielle, en phase par exemple.

$$\sigma_A^2(\tau) = \frac{1}{2}\langle (y_2 - y_1)^2 \rangle \tag{2.9}$$

Cette variance présente d'autres propriétés, comme une convergence pour les principaux types de bruit. Le tracé de cette variance présente une allure caractéristique en «cuvette», qui s'interprète par l'influence des différents types de bruit du modèle en lois de puissance. La figure 2.6 présente la forme caractéristique de l'écart type, soit la racine de cette variance, et identifie les bruits responsable de l'allure de la courbe. Au final, cet estimateur constitue un excellent outil pour l'étude de la stabilité des oscillateurs à long terme.

Le lecteur prendra garde à l'abus de langage courant qui consiste à parler de variance d'Allan mais à observer et tracer en réalité l'écart type, cas de la figure 2.6.

À noter que dans le cas des gyromètres, les source de bruit s'identifient plus spécifiquement, ainsi que le précise le tableau 2.1.

Pente	**Cas standard**	**Cas d'un gyromètre**
-1	Bruit blanc de phase	Bruit de quantification
-1	Bruit de phase de scintillation	
$-\frac{1}{2}$	Bruit blanc	Marche aléatoire angulaire
0	Bruit de scintillation	Instabilité de biais
$\frac{1}{2}$	Marche aléatoire	Marche aléatoire de rotation

Table 2.1 – Dénomination des bruits pour un gyromètre

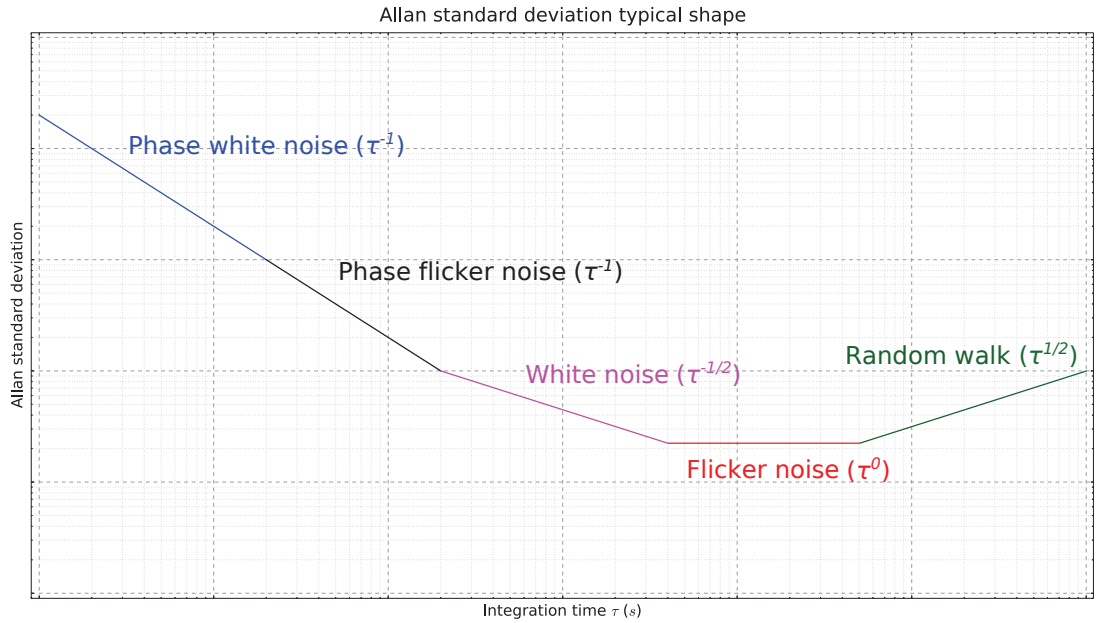


FIGURE 2.6 – Forme caractéristique de la variance d'ALLAN et identification des contributeurs majoritaires

2.2 Exploitation des données du marquage temporel pour la mesure d'accélération

Comme énoncé plus tôt, le marquage temporel n'exploite pas encore la totalité des informations disponibles. La méthode qui exploite au maximum ces données repose sur une régression linéaire. En effet, la pente de la droite qui représente le nombre de périodes d'horloge en fonction du nombre de périodes du signal d'entrée (figure 2.7) fournit directement la période du signal et son inverse, la fréquence [Joh05].

L'erreur relative de cette méthode est déterminée par analyse statistique et l'équation (2.10) présente le résultat. Pour n suffisamment élevé ($n > 8$), une régression linéaire améliore la résolution d'un facteur $\sqrt{\frac{6}{n-2}}$ [Joh05] par rapport au comptage réciproque classique puisqu'elle utilise $n-2$ points supplémentaires.

$$\frac{\sigma\left(\hat{T}\right)}{\hat{T}} = \frac{\sigma\left(\hat{F}\right)}{\hat{F}} = \frac{2\sqrt{3}T_{clk}}{T\sqrt{n-2}} \tag{2.10}$$

Cependant, ce calcul nécessite davantage d'opérations que le simple comptage réciproque. En effet, l'équation (2.8) implique une multiplication et une division [1], opérations déjà lourdes pour des systèmes binaires. Au contraire, une régression linéaire implique un calcul de déterminant de matrice et une division. Soit n multiplications-additions [2] et une

1. En fonction du fonctionnement du compteur, la soustraction peut déjà être faite.
2. Multiply-accumulate en Anglais, terme souvent utilisé même en Français tellement cette opération

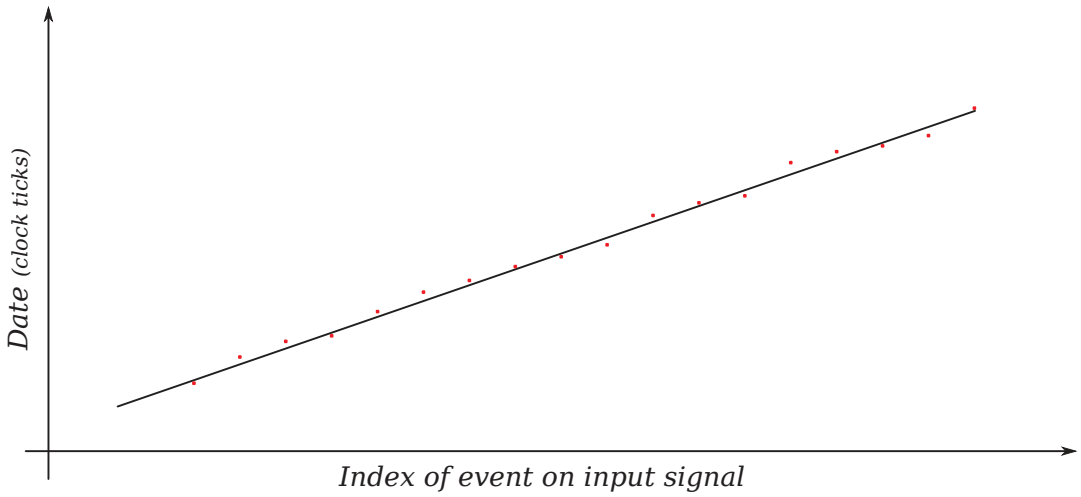


FIGURE 2.7 – Comptage de fréquence par marquage temporel. On relève les dates, exprimées en nombre de périodes d'horloge, des fronts montants du signal d'entrée. La fréquence du signal est donnée par la pente de la droite de régression.

Notation	Signification
T	Période vraie du signal d'excitation (en nombre de fronts d'horloge du compteur)
k	Instant, numéro du front d'horloge sur le compteur
C_k	Valeur de la capture (date) du signal d'excitation à l'instant k
R_k	Valeur de la capture (date) de la réponse du capteur à l'instant k
T_k	Période instantanée du signal d'excitation ($T_k = C_k - C_{k-1}$)
N	Nombre de points pour calculer l'estimateur de T
$\hat{T}_k$	Estimateur de T à l'instant k avec n valeurs

TABLE 2.2 – Signification des variables employées dans ce chapitre

division. À l'apparition du comptage réciproque, une électronique avec la puissance de calcul nécessaire aurait coûté trop cher, voilà pourquoi cette solution ne s'est pas imposée immédiatement.

Avec les notations du tableau 2.2, et avec b quelconque, la régression recherche T tel que :

$$C_k = k \times T + b \tag{2.11}$$

b n'est en fait pas utile dans le calcul : il suffit de soustraire la première date de la fenêtre de mesure à toutes les dates de la fenêtre pour faire disparaître b dans ce qui suit (ainsi $C_0 = 0$).

La méthode des moindres carrés est la méthode la plus commune pour réaliser une régression linéaire. Cette méthode minimise les distances entre les points de mesure et

est commune en traitement de signal, et donc souvent implémentée dans les systèmes de calculs.

la droite du modèle. Autrement dit, avec les notations de l'équation (2.11), la méthode cherche un minimum de la somme :

$$D(\hat{T}) = \sum_{k=0}^{N-1} \left(C_k - k \times \hat{T} \right)^2$$

Il s'agit d'une fonction de $\hat{T}$. Un minimum se caractérise par une dérivée nulle, soit :

$$\frac{dD}{dT} = 0$$

Il vient :

$$\sum_{k=0}^{N-1} kC_k - k^2\hat{T} = 0$$
$$\sum_{k=0}^{N-1} kC_k - \hat{T} \sum_{k=0}^{N-1} k^2 = 0$$

La somme des carrés est égale à :

$$\sum_{k=0}^{N} k^2 = \frac{N(N-1)(2N-1)}{6}$$

D'où :

$$\hat{T} = 6 \frac{\sum_{k=0}^{N} kC_k}{N(N-1)(2N-1)} \tag{2.12}$$

Il est plus judicieux de mémoriser directement $\frac{N(N-1)(2N-1)}{6}\hat{T}$ et de ne pas diviser. Dans ce cas précis, il n'y a pas de risque de dépassement de ce produit puisque l'équation (2.10) a montré qu'il ne sert à rien de prendre un grand nombre de points et que la résolution tend vers celle d'un marquage temporel classique (sans régression linéaire) lorsque ce nombre croit. Néanmoins, la méthode des moindres carrés demande un minimum de points pour fournir un résultat précis.

Un calcul sur un nombre entre 16 et 32 captures paraît un bon compromis entre utilisation mémoire, puissance de calcul nécessaire, rapidité d'exécution et validité du résultat.

L'équation (2.12) ne permet pas d'optimisation particulière : le produit kC_k empêche de mémoriser les résultats précédents. Bien que lorsque les MSB des C_k ne varient pas ils puissent être ignorés (ils sont assimilés à une constante que l'on peut regrouper avec b) et le calcul de $\sum_{k=0}^{N} kC_k$ effectué sur 32 bits uniquement (exécution plus rapide), la logique pour traiter les cas où ces MSB ne s'annulent pas fait perdre tout le gain de cette optimisation. Le code qui calcule $(\hat{T})$ suit donc exactement l'équation (2.12). Il faut alors :

— $2N$ chargements mémoire (entiers sur 64 bits et non 32) ;
— $2N$ multiplications ;
— $3N$ additions (il faut propager la retenue de la multiplication des LSB) ;
— N additions pour l'incrément de l'indice de boucle ;
— N additions pour le calcul de l'indice du tableau contenant les C_k à partir de l'indice de boucle.

Soit $9N$ instructions pour le cœur de la fonction de calcul. Pour $N = 25$ et un calcul tous les N points et un VIA à 65 kHz, le système doit effectuer ce calcul 2600 ($65 \times 10^3 \div 25$) fois par secondes. Une puissance surestimée à $0,6$ MIPS (million d'instructions par seconde) garantit que le système aura le temps d'effectuer correctement ce calcul. Le MicroBlaze en est largement capable puisqu'il dispose d'une puissance de calcul de $1,03$ DMIPS.MHz^{-1} [3], soit 103 DMIPS à 100 MHz. Dans le cas d'un autre processeur, une puissance de $0,7$ DMIPS devrait convenir.

Enfin, F la fréquence du capteur se déduit très rapidement de la formule suivante :

$$\hat{F} = \frac{300 \times 10^6}{\hat{T}} \frac{6}{N(N-1)(2N-1)} \tag{2.13}$$

Ici encore, la division peut être évitée en utilisant directement la valeur mémorisée $\frac{N(N-1)(2N-1)}{6}\hat{T}$. Toutefois, en vue d'obtenir la mesure de l'accélération, il faut inévitablement diviser par le facteur d'échelle des accéléromètres. Cette opération ne sera pas effectuée dans l'électronique du capteur, afin d'éviter l'implantation d'une unité de division. La documentation technique du capteur fournira simplement la valeur d'un LSB de la sortie du capteur.

En outre, cette valeur peut dépendre d'une ou plusieurs variables configurables, comme le nombre de points de calcul de la régression linéaire N. La documentation technique précisera comment varie le LSB en fonction des paramètres choisis.

En conclusion, la méthode de calcul par régression linéaire peut se contenter de calculs entier uniquement.

3. Voir l'annexe E.2 pour la différence entre MIPS et DMIPS.

2.3 Formation d'un signal sinusoïdal en électronique numérique

2.3.1 Paramètres de conception d'un synthétiseur numérique direct

Ainsi que l'a montré la section 1.3.2, les numériciens ont mis en œuvre plusieurs méthodes en vue de créer des signaux analogiques périodiques à partir de systèmes entièrement numériques, en particulier dans la gamme de fréquence audible. Aujourd'hui, avec l'amélioration générale de l'électronique numérique dans tous les domaines, notamment en terme de consommation, de fréquence de fonctionnement et de coût, la méthode de synthèse numérique directe avec LUT[4] tend à s'imposer. La figure 2.8 rappelle son principe de fonctionnement. La simplicité et la flexibilité qu'offrent les DDS de ce type contribuent aujourd'hui à leur essor.

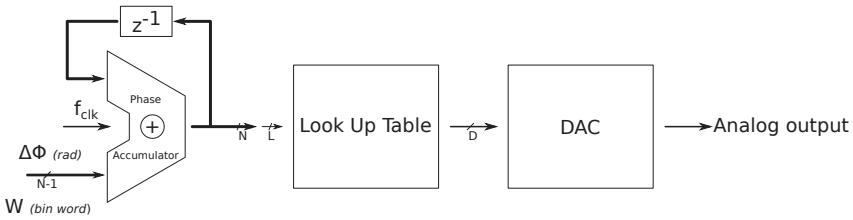


FIGURE 2.8 – Schéma de principe d'une DDS

Simplicité parce qu'un simple additionneur binaire suffit à jouer le rôle de l'accumulateur de phase. Flexibilité parce que le signal de sortie peut prendre n'importe quelle forme puisqu'une programmation de la LUT avec la forme d'onde souhaitée suffit (créneau, dent de scie...). De plus, le signal mémorisé peut même être beaucoup plus complexe, en vue par exemple, de corriger des défauts sur le système excité par la DDS.

La suite de cette étude utilise les notations listées dans le tableau 2.3

Notation du paramètre	Signification
f_{clk}	Fréquence d'horloge du système, aussi appelée fréquence d'échantillonnage
f_{out}	Fréquence de sortie souhaitée ou fréquence cible
N	Nombre de bits de l'accumulateur de phase
L	Nombre de bits d'adresses de la LUT
D	Nombre de bits du DAC

TABLE 2.3 – Notation des paramètres d'une DDS

4. Look Up table : abaque.

Pour obtenir la fréquence de sortie désirée, l'utilisateur programme un incrément de phase $\Delta\phi$ sur l'accumulateur de phase sous forme d'un mot binaire W, choisi selon la formule suivante :

$$f_{out} = \frac{W \times f_{clk}}{2^N} \tag{2.14}$$

L'équation (2.15) donne ensuite la relation entre le mot binaire W (aussi appelé tuning word en Anglais) et l'incrément de phase $\Delta\phi$, en radians.

$$\Delta\phi = W \times \delta\phi \tag{2.15}$$

Dans cette dernière équation, $\delta\phi$ représente le pas de phase de la DDS et vaut :

$$\delta\phi = \frac{2\pi}{2^N} \tag{2.16}$$

Les N bits de sortie de l'accumulateur de phase représentent alors la phase courante. Les bits de poids forts de cette phase servent ensuite de bits d'adresse à la LUT qui fournit en sortie l'amplitude courante correspondante, sous forme d'un mot numérique sur D bits, le nombre de bits du DAC. La LUT agit véritablement comme une fonction au sens mathématique :

$$f(\phi(t)) = y$$

La LUT fournit donc l'amplitude à un instant t_k du signal souhaité (puisqu'il s'agit d'un système numérique, le temps est discret). Cette amplitude demeure encore sous forme numérique et il reste à la convertir sous forme analogique, un niveau de tension par exemple, fonction remplie par un convertisseur numérique-analogique, Digital/Analog Converter (DAC) en Anglais. Aujourd'hui, il existe plusieurs systèmes de DAC, chacun avec leurs avantages et leurs inconvénients.

Dans une approche détaillée, il paraît évident que chaque paramètre de chaque composant qui intervient dans une DDS influe sur l'allure du signal de sortie obtenu, notamment sur sa pureté spectrale.

2.3.2 Contraintes entre les paramètres

La figure 2.8 et le tableau 2.3 montrent qu'une DDS, dans sa conception, présente 4 degrés de liberté :

— la fréquence d'échantillonnage/de fonctionnement du système, f_{clk} ;
— le nombre de bits de l'accumulateur de phase N ;
— le nombre de bits d'adresse de la LUT L ;

— le nombre de bits du DAC D, qui correspond au nombre de bits de sortie de la LUT.

La première chose à faire en vue d'étudier l'influence de chacun de ces paramètres sur le spectre de sortie consiste à réduire, si possible, le nombre de degrés de liberté.

L'équation (2.14) présente la première contrainte qui lie entre elles deux dimensions de l'espace d'étude. Cette équation définit en fait le pas de fréquence de la DDS (et de ce fait, la plus petite fréquence synthétisable) :

$$\delta f = \frac{f_{clk}}{2^N} \tag{2.17}$$

Autrement dit, si la largeur N de l'accumulateur de phase et la fréquence d'échantillonnage f_{clk} s'avèrent fixées, alors cette équation (2.17) détermine le pas de fréquence.

L'équation (2.18) explicite en outre la relation entre le pas de phase $\delta\phi$ défini à l'équation (2.16) et le pas de fréquence.

$$\delta\phi = 2\pi\frac{\delta f}{f_{clk}} \tag{2.18}$$

Si, pour une fréquence d'horloge f_{clk} donnée, l'utilisation finale de la DDS impose un pas de fréquence minimal δf_{min}, alors l'équation (2.17) conduit à définir une valeur minimale pour le nombre de bits de l'accumulateur de phase, N. L'équation (2.19) présente l'inégalité qui en découle, avec $\log_2$ le logarithme en base 2.

$$\begin{aligned} \delta f_{min} \geqslant \delta f &= \frac{f_{clk}}{2^N} \\ N &\geqslant \log_2(f_{clk}) - \log_2(\delta f_{min}) \end{aligned} \tag{2.19}$$

L'équation (2.17) peut aussi s'écrire selon l'équation (2.20) pour respecter une valeur minimale de pas de fréquence pour un N donné par exemple.

$$f_{clk} \leqslant \delta f_{min} \times 2^N \tag{2.20}$$

L'équation (2.17) limite donc l'espace des solutions possibles lorsque certaines contraintes sur la fréquence d'échantillonnage f_{clk}, la largeur de l'accumulateur de phase N ou le pas de phase minimum nécessaire $\delta\phi_{min}$ apparaissent.

En sortie de la DDS, c'est-à-dire du côté de la LUT et du DAC, le fonctionnement du système ne dicte aucune relation entre le nombre de bits d'adresse de la LUT L et le nombre de bits du DAC D, qui définit aussi la taille des mots stockés dans la LUT, c'est-à-dire sa largeur de sortie. Cependant, la recherche de l'optimisation conduit à définir une contrainte entre L et D.

Une DDS optimale stockerait un minimum de mots de largeur D et présenterait sur le

DAC un incrément minimal là où la pente de la sinusoïde est la plus forte, soit aux passages par 0 (mod π). Autrement dit, autour de 0 et π, un incrément de 1 sur l'adresse de la LUT correspondrait à un incrément ou un décrément de 1 sur le DAC [5]. L'équation (2.21) traduit cette optimisation. La LUT prend en entrée une phase sur L bits. Puisque cette phase varie de 0 à 2π en radian, soit de 0 à $2^L - 1$ en binaire, un incrément d'adresse de 1 correspond à une variation de $\frac{2\pi}{2^L}$ radian. À sa sortie, la LUT présente une amplitude entre 0 et 2, et les mots qu'elle contient varient de 0 à $2^D - 1$. Ceci conduit alors à l'équation (2.21).

$$\frac{2\pi}{2^L} = \frac{2}{2^D} \tag{2.21}$$

Un rapide calcul à partir de l'équation (2.21) mène alors à une relation entre L et D.

$$\begin{aligned} 2^L &= \pi\, 2^D \\ L &= D + \log_2(\pi) \end{aligned}$$

Puisque D et L sont entiers, il vient, avec $\lceil x \rceil$ la partie entière par excès de x :

$$\begin{aligned} L &= D + \lceil \log_2(\pi) \rceil \\ L &= D + 2 \end{aligned} \tag{2.22}$$

Cette égalité (2.22) est une recommandation pour le dimensionnement de la table d'onde, qui optimise la quantité d'information : en dessous de $D + 2$, la table est petite et saute des valeurs du DAC dans le parcours de la sinusoïde (aux passages par zéro) ; au dessus de $D + 2$, la table est grande et des valeurs du DAC apparaissent en double.

Il reste à établir une relation entre les deux parties de la DDS, l'accumulateur de phase et le couple LUT/DAC. En réalité, cette relation s'impose par des considérations pratiques, technologiques et/ou économiques. En effet, la section 1.3.2 a mis en avant que la LUT a longtemps freiné le développement de ces systèmes. Malgré les progrès technologiques, considérer des LUT de plusieurs gigabits demeure absurde : trop grand, trop cher, trop énergivore. Autrement dit, le produit $2^L \times D$, le nombre de bits stockés dans la LUT, doit rester de l'ordre de quelques dizaines de milliers pour un système embarqué.

Par ailleurs, les DAC haute résolution, à 24 ou 32 bits par exemple, ne fonctionnent pas à fréquence élevée. Ceci conduit à utiliser des DAC d'une dizaine de bits et donc des LUT avec une dizaine de bits d'adresse.

La suite de cette partie présente l'influence des 3 degrés de liberté de conception d'une DDS sur le spectre de sortie, à savoir :

5. La phase qui sert d'adresse sur la LUT ne peut qu'augmenter ; la sortie de la LUT donc l'entrée du DAC peut, elle, augmenter ou diminuer.

— la fréquence de fonctionnement de la DDS, ou fréquence d'échantillonnage f_{clk} ;
— la largeur de l'accumulateur de phase N ;
— le nombre de bits du DAC D qui fixe également le nombre de bits d'adresse de la LUT L en vertu de l'égalité (2.22), et définit la troncature de phase, c'est-à-dire le nombre de bits de poids fort conservés en sortie de l'accumulateur pour adresser la LUT.

2.3.3 Influence de la fréquence d'échantillonnage sur le spectre de sortie

La fréquence de fonctionnement de l'ensemble du périphérique s'avère évidemment le premier paramètre influant sur le spectre de sortie, puisqu'elle fixe la fréquence d'échantillonnage du signal délivré. Cette fréquence définit le nombre de points par période du signal de sortie, d'après les équations (2.23) et (2.24).

$$n_{points} = \frac{f_{clk}}{f_{out}} \tag{2.23}$$

$$n_{points} = \frac{2^N}{W} = \frac{2\pi}{\Delta\phi} \tag{2.24}$$

L'équation (2.24) pourrait laisser penser que le nombre de bits N de l'accumulateur détermine le nombre de points dans le signal de sortie. En réalité, si N augmente, W augmente aussi de sorte à conserver le ratio des équations (2.23) et (2.24). C'est donc bien l'équation (2.23) qui dicte le nombre de points par période du signal de sortie. Puisque le théorème de Nyquist-Shannon impose un minimum de deux points par période pour reconstituer le signal échantillonné, la fréquence d'échantillonnage doit être au moins deux fois supérieure à celle de la fréquence à reproduire.

La figure 2.9 montre la sortie d'une DDS avec un compteur, une LUT et un DAC chacun avec un nombre de bits infini dans des cas extrêmes du théorème de Nyquist-Shannon, à $f_{out} = \frac{f_{clk}}{2}$. La sortie est normalisée entre 0 et 1 car, en général, les DAC utilisent une seule tension de référence, positive, et ne délivrent donc pas de tension négative[6]. La courbe en pointillés représente la sinusoïde théorique et les points bleus, les points fournis par la DDS. Dans le cas de la figure 2.9a, tout se passe comme si il n'y avait aucun signal. En revanche, dans le cas de la figure 2.9b, l'amplitude est maximale mais pourrait aussi ressembler à un signal carré. Difficile d'imaginer alors que ceci constitue une sinusoïde... Pourtant, la figure 2.10[7] montre que le spectre de sortie ne comporte qu'une valeur moyenne (puisque la moyenne de la sortie vaut $0,5$), la fréquence souhaitée, et une répétition de celle-ci par symétrie autour de la fréquence d'échantillonnage et de ses harmoniques ($k \times 120\,\text{kHz} \pm 60\,\text{kHz}$). Attention toutefois, le spectre présenté ici s'arrête

6. Des DAC avec alimentations symétriques peuvent fournir une sortie entre -1 V et 1 V par exemple.
7. Le spectre est normalisé pour former une grandeur sans dimension.

à 600 kHz mais en réalité, les harmoniques se répètent à l'infini (jusqu'à atténuation complète).

L'annexe A présente le cœur du code à l'origine de ce graphique et de ceux à venir. Brièvement, le simulateur calcule l'amplitude des valeurs de sortie de la sinusoïde aux fronts d'horloge en fonction des paramètres choisis de la DDS. Il s'appuie ensuite sur la bibliothèque FFT West pour calculer les spectres sans utiliser de fenêtrage.

Toute fréquence f_{out} telle que $f_{out} < 2f_{clk}$, c'est-à-dire qui respecte le théorème de Nyquist-Shannon, présente un spectre similaire à celui de la figure 2.10 : deux raies, l'une à la fréquence nulle, l'autre à la fréquence de sortie désirée, puis des raies à $\pm f_{clk}$ par rapport aux harmoniques de la fréquence d'échantillonnage. La figure 2.12 montre la répétition de ces dernières et la figure 2.13 présente un zoom autour d'un harmonique de la fréquence d'horloge.

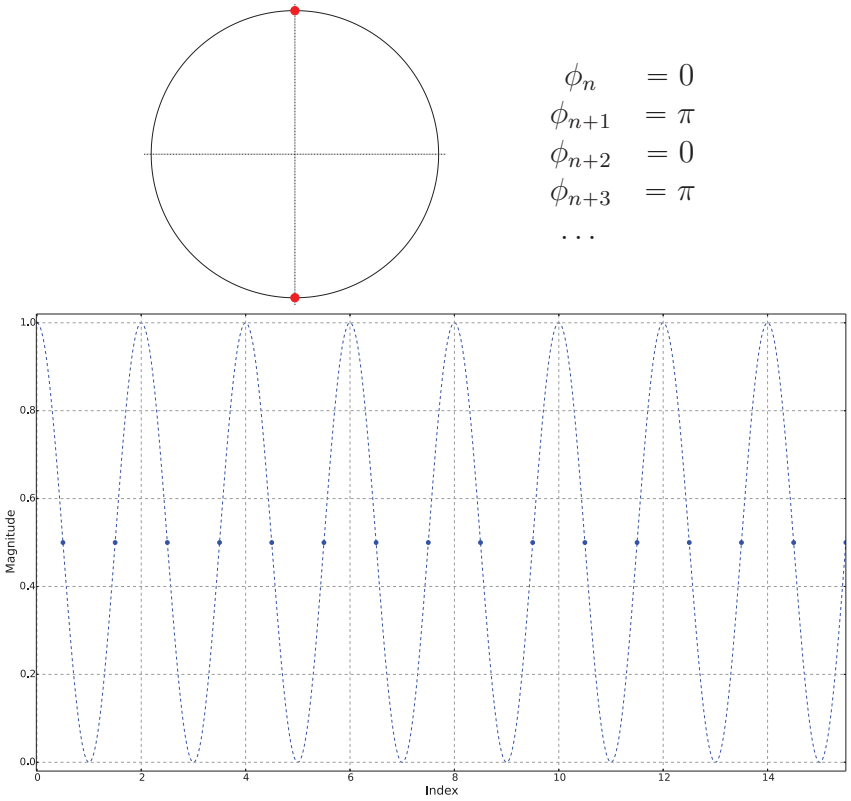
Plusieurs éléments peuvent détériorer ces spectres. Si le rapport $\frac{f_{clk}}{f_{out}}$ n'est pas un entier, alors il faut attendre un grand nombre de périodes du signal de sortie avant d'avoir un nombre entier de périodes dans la fenêtre d'échantillonnage. Plus exactement, le temps nécessaire pour atteindre un nombre entier de périodes s'exprime par $\mathrm{lcm}(f_{out}, f_{clk})$, avec lcm le plus petit commun multiple. De même, les dimensions finies de la DDS limitent les fréquences de sortie possibles. En boucle fermée, cela signifie que la DDS oscille entre deux fréquences proches de la cible afin d'obtenir en moyenne la fréquence voulue. Cependant, si le système bascule trop rapidement d'une fréquence à l'autre, alors la DDS n'a pas le temps de générer un nombre entier de périodes. Ceci se traduit sur les spectres par un élargissement du pic à la fréquence programmée et de ses harmoniques et une dégradation du rapport signal à bruit. Les figures 2.11 et 2.12 mettent en évidence ces effets.

La figure 2.12 montre aussi qu'une fréquence d'échantillonnage plus élevée diminue le bruit sur l'ensemble du spectre. Il s'agit là d'une propriété importante pour la suite de l'étude, qui montre que les autres paramètres de dimensionnement affectent grandement le bruit en sortie de la DDS. Ainsi, une fréquence d'échantillonnage suffisamment élevée ne contribue pas au bruit général sur le signal de sortie et rejette aussi les harmoniques (symétriques autour des multiples de la fréquence d'horloge) suffisamment loin pour ne pas exciter de mode non désiré des capteurs.

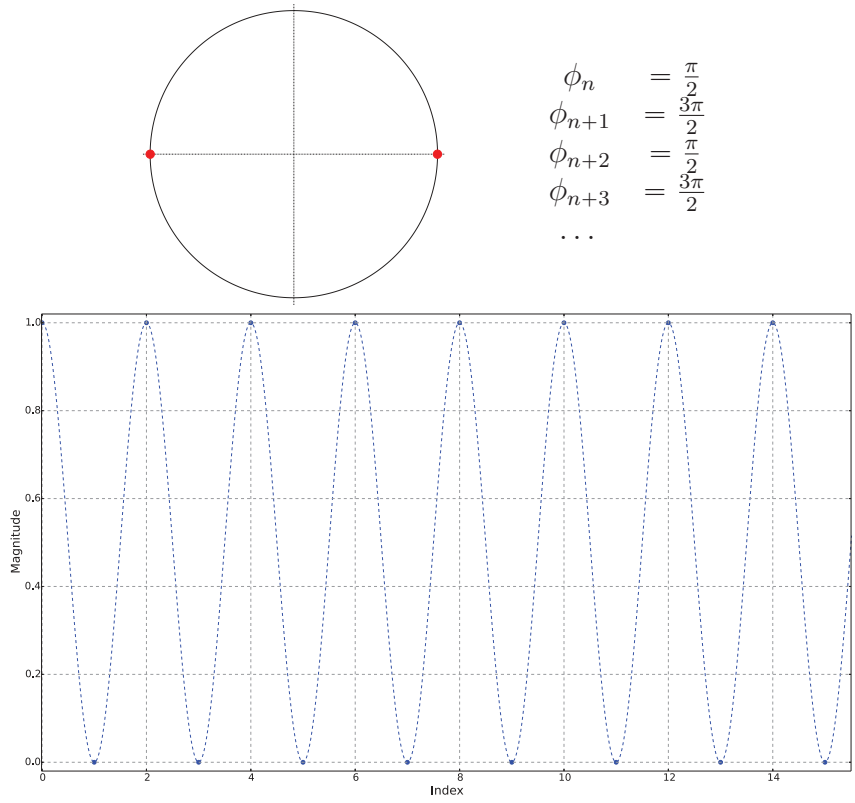
Dans toute la suite de cette étude, la fréquence d'horloge du système est fixée à 50 MHz. La section 3.6 justifiera ce choix qui répond à des contraintes technologiques.

2.3.4 Influence de l'accumulateur de phase sur le spectre de sortie

Le nombre de bits de l'accumulateur de phase constitue le second paramètre qui intervient dans la conception d'une DDS.



(a) Cas où le signal de sortie est nul



(b) Cas où le signal oscille avec une amplitude maximum

FIGURE 2.9 – Cas limite du théorème de NYQUIST-SHANNON où $f_{clk} = 2f_{out}$. Selon la phase de l'échantillonnage, la sortie peut demeurer nulle (figure 2.9a) ou former un créneau (figure 2.9b).

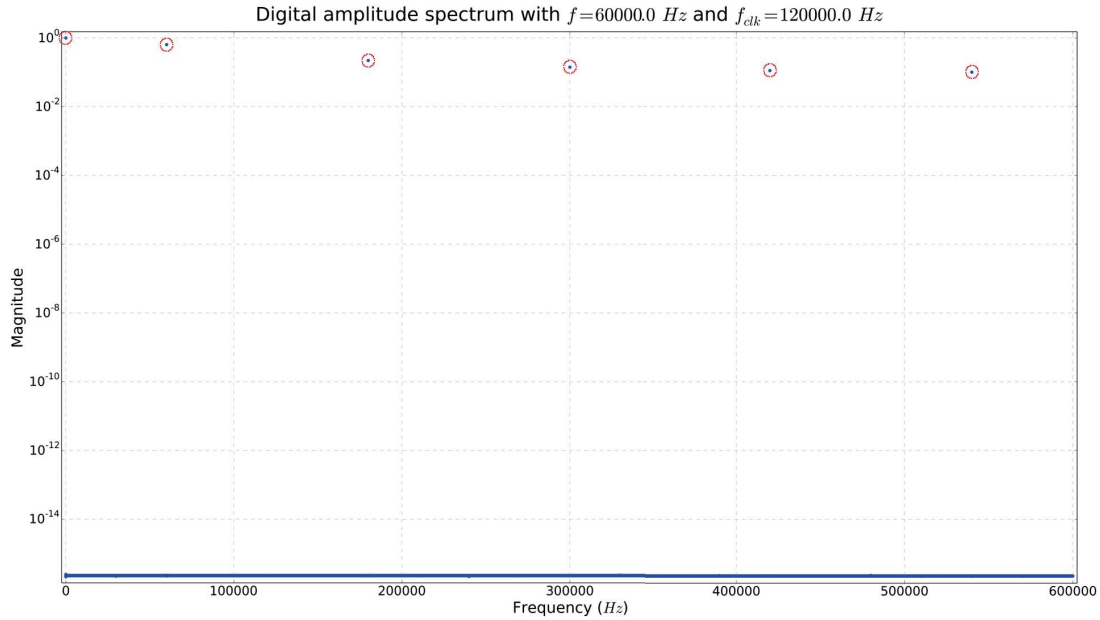


FIGURE 2.10 – Spectre de sortie d'un signal à 60 kHz avec une fréquence d'échantillonnage de 120 kHz

Les équations (2.14) et (2.17) ont déjà montré comment N, le nombre de bits de l'accumulateur de phase, intervenait dans le fonctionnement général de la DDS : il fixe le pas de phase ($\frac{2\pi}{2^N}$) et donc le pas de fréquence en fonction de f_{clk} ($\frac{f_{clk}}{2^N}$). Il détermine ainsi la plus petite fréquence synthétisable mais aussi l'incrément de fréquence possible : la DDS ne synthétise alors que des fréquences multiples de cette valeur.

Il s'agit là de la restriction des fréquences accessibles mentionnée plus tôt. La DDS ne peut délivrer que les seules fréquences multiples du pas de fréquence.

Ce paramètre N n'influe en rien sur le spectre de sortie. Deux DDS qui fonctionnent à la même fréquence, avec un nombre de bits différent pour leurs accumulateurs de phase, et des LUT et DAC idéaux (nombres de bits infinis) qui génèrent deux signaux à la même fréquence (c'est-à-dire une fréquence synthétisables pour les deux DDS), fournissent les mêmes spectres car les sorties de leurs accumulateurs de phase sont identiques. En fait, les bits de poids faible de l'accumulateur de phase avec la plus grande résolution restent en permanence à 0.

Les figures 2.14 et 2.15 montrent cette correspondance, pour des DDS avec une fréquence d'horloge de 50 MHz et des accumulateurs de phase à 24 (bleue) et 32 bits (rouge). Ces figures présentent des spectres pour une fréquence autour de 60 kHz. Les fréquences générées sur ces figures sont bien synthétisables par les deux DDS en vertu de l'équations (2.25) et comme le montre le tableau 2.4.

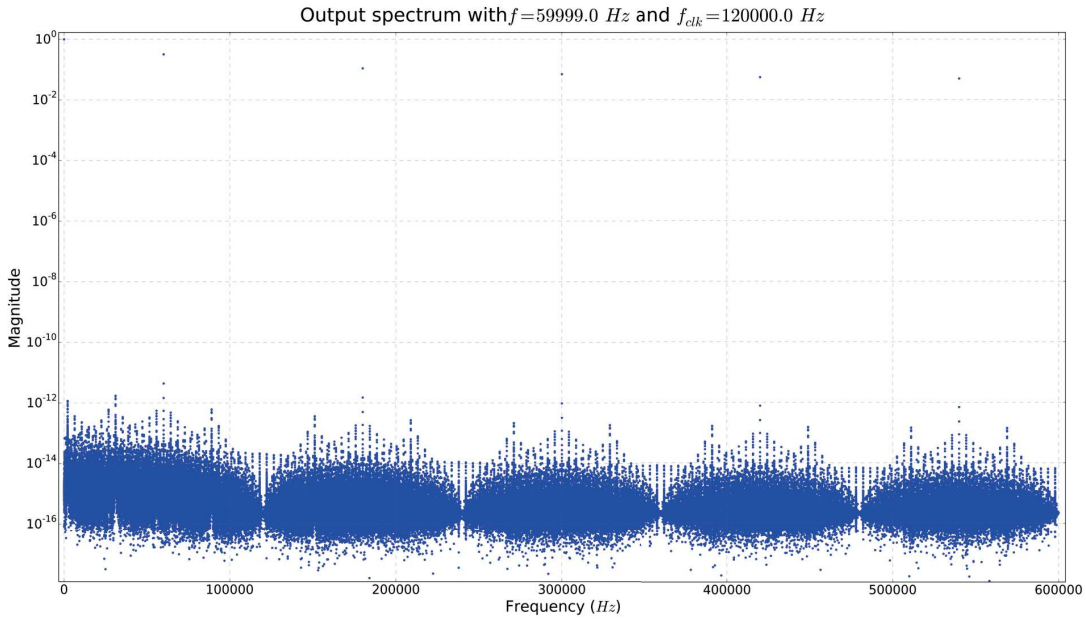


FIGURE 2.11 – Sortie d'une DDS dans un cas non idéal avec un taux de sur-échantillonnage faible (2 ici)

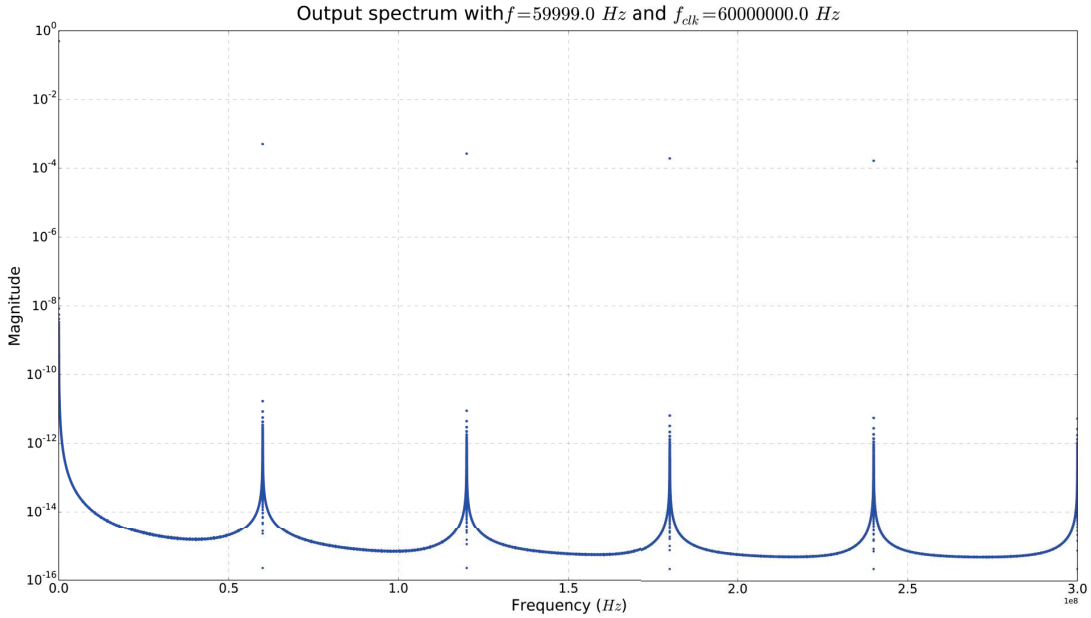


FIGURE 2.12 – Sortie d'une DDS dans un cas non idéal avec un taux de sur-échantillonnage élevé

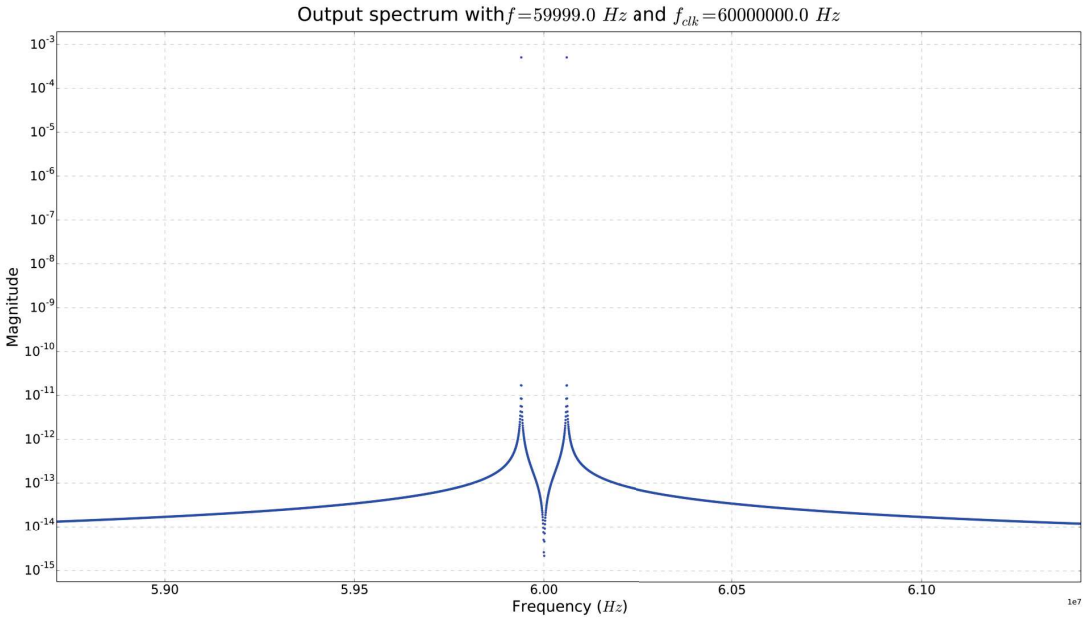


FIGURE 2.13 – Mise en évidence des symétries (zoom de la figure 2.12 autour de f = 60 MHz)

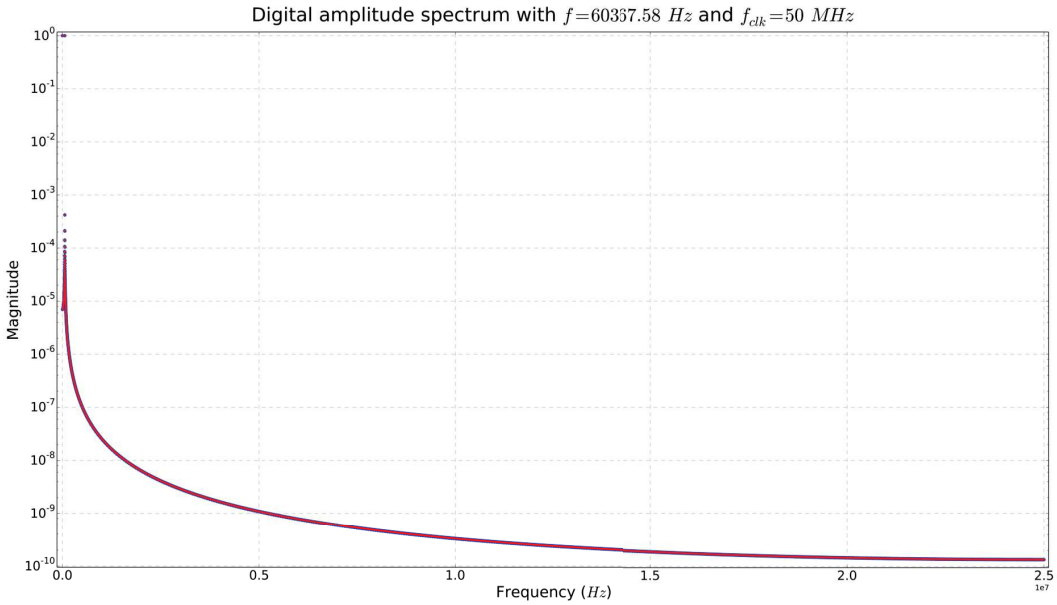


FIGURE 2.14 – Spectres de sorties de deux DDS qui ne diffèrent que par leur nombre de bits sur l'accumulateur de phase

$$
\begin{aligned}
f_{out} &= \frac{50 \times 10^6}{2^{24}} \times W \\
f_{out} &= \frac{50 \times 10^6}{2^{32}} \times 2^{32-24} \times W \\
f_{out} &= \frac{50 \times 10^6}{2^{32}} \times W \times 2^8
\end{aligned} \tag{2.25}
$$

DDS	N	δf (Hz)	W
1	24	$2,98$	20256
2	32	$0,012$	$20256 \times 2^8 = 5185536$

TABLE 2.4 – Formation d'un signal à la même fréquence à partir de deux DDS différentes

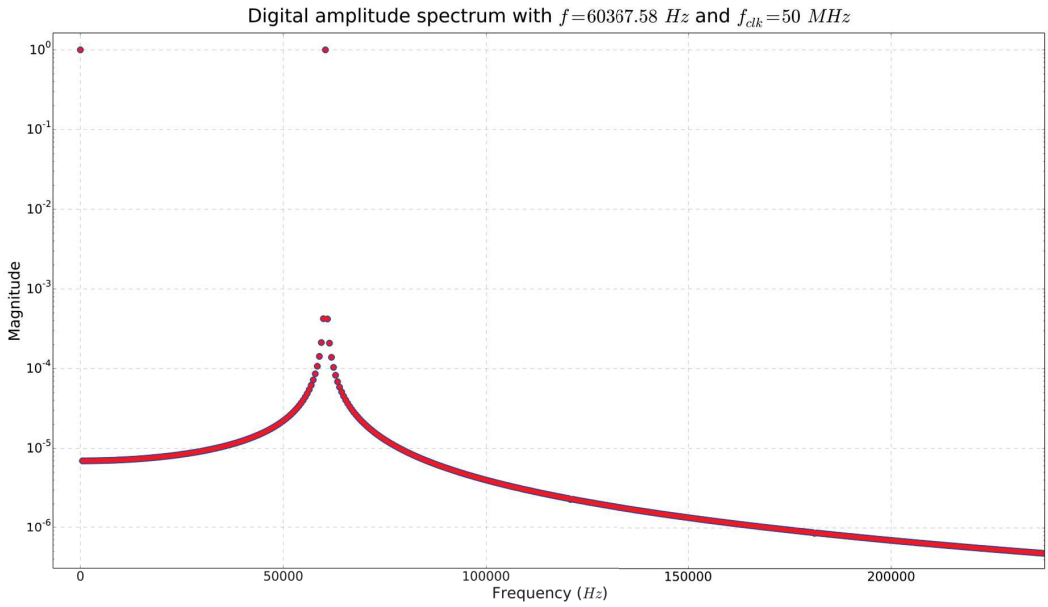


FIGURE 2.15 – Zoom sur les spectres de la figure 2.14

En conclusion, le nombre de bits de l'accumulateur de phase N n'influence que le pas de fréquence et nullement le spectre de sortie. Ce paramètre est donc fixé pour répondre aux besoins de l'application en terme de pas de fréquence.

Dans la suite de l'étude, il est fixé à 48 ; la section 3.1 motivera ce choix.

2.3.5 Influence de la LUT

L'équation (2.22) lie le nombre de bits de la LUT et celui du DAC. Les études de l'influence de chacun de ces deux paramètres sur le spectre de sortie doivent pourtant rester

distinctes. La nature des phénomènes qui interviennent sur le spectre final s'avère bien différente. Aussi, cette section n'étudie que l'influence de L, le nombre de bits d'adresse de la LUT, sur le spectre de sortie. La section suivante étudiera l'influence du nombre de bits du DAC.

Conséquences de la troncature de phase

Choisir L tel que $L = N$, avec N la largeur de l'accumulateur de phase, semble le premier choix à faire, le plus simple. Malheureusement, si N égal 32, cela représente $2^{32} = 4\,294\,967\,296$ mots à stocker. Selon l'équation (2.22), chacune de ces valeurs comporteraient 30 bits ; soit au total, 15 GB, une valeur bien trop grande pour un dispositif qui se veut petit, économique et peu gourmand en énergie. D'autant plus que dans le cadre d'une utilisation spatiale, la mémoire s'avère particulièrement critique en terme de coût et de fiabilité. Sa réduction réduit les problèmes potentiels ainsi que les coûts.

Ce nombre L doit donc demeurer petit. Pour $L = 12$ par exemple, 5 kB suffisent à stocker toutes les valeurs sur 10 bits.

Ainsi, si L devient plus petit que N, seule une partie des N bits de l'accumulateur de phase est utilisée : il y a troncature de phase. Seuls les L bits de poids forts (du bit N au bit $N - L + 1$), en sortie de l'accumulateur de phase déterminent l'amplitude de sortie.

Une option simpliste consisterait à choisir $N = L$. Cependant, N et L influencent différemment les performances finales de la DDS et demeurent indépendants. En effet, N fixe le pas de phase indépendamment de L, et donc retenir un N petit (une dizaine de bits) impose un grand pas de phase, pas nécessairement adapté à l'application.

Cette troncature de phase entraîne alors une erreur de phase. La constellation des phases de la figure 2.16 représente sous forme graphique l'erreur de phase qui peut apparaître à chaque front d'horloge. Elle permet de mieux comprendre en quoi et comment cette erreur va affecter le spectre du signal de sortie. Cette figure considère une DDS avec les paramètres du tableau 2.5. L'équation (2.26) exprime le pas de phase $\delta\phi$ de cette DDS (voir équation (2.16).

$$\delta\phi = \frac{2\pi}{2^9} = 12\,\text{mrad} \tag{2.26}$$

Paramètre	Valeur
N	9
L	6
D	∞

Table 2.5 – Paramètres de la DDS de la figure 2.16

Les points bleus du cercle de phase représentent l'ensemble des phases accessibles avec

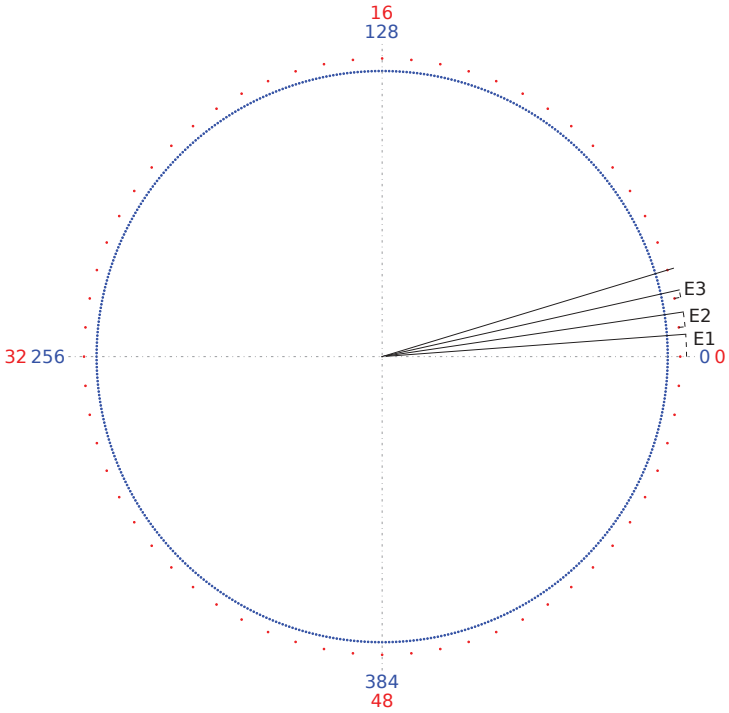


FIGURE 2.16 – Représentation de l'erreur de phase sur le cercle de phase

un tel pas. Les points rouges représentent les phases pour lesquelles la LUT contient une valeur, soit un pas de phase, avec $L = 6$, exprimé par :

$$\frac{2\pi}{2^6} = 98\,\text{mrad}$$

Considérons maintenant un cas où l'utilisateur programme un pas de phase de 6 ($W = 6$) sur la DDS. Le tableau 2.6 présente les sorties de l'accumulateur de phase et de la LUT, l'erreur de phase engendrée et l'arc d'erreur représenté sur la figure 2.16. L'erreur de phase forme, dans le cas général, un signal en dent de scie. La figure 2.17 présente les valeurs du tableau 2.6 et met en évidence la forme en dent de scie de ce signal.

La poursuite des calculs du tableau 2.6 fait alors apparaître la périodicité des erreurs de phase.

Étude de cas particuliers

Irrémédiablement, ces erreurs périodiques vont causer des spurs, c'est-à-dire des fréquences non désirées à des niveaux significatifs, sur le spectre de sortie. De plus, deux consignes de pas de phase W différentes mais proches (distance de HAMMING de 1 par exemple) engendrent des spectres totalement différents. En fait, l'erreur de phase est toujours un signal périodique en dent de scie sauf dans deux cas particuliers : une erreur nulle (absence de spur) et une erreur en créneaux dont l'amplitude maximale est égale à la moitié du pas de phase (amplitude maximale des spurs).

Front d'horloge	Sortie de l'accumulateur	Sortie de la LUT	Erreur de phase	Arc d'erreur
0	0	0	0	
1	6	0	$6 \times \frac{2\times\pi}{2^9} - 0 = 74\,\text{mrad}$	E1
2	12	1	$2 \times 6 \times \frac{2\times\pi}{2^9} - \frac{2\times\pi}{2^6} = 49\,\text{mrad}$	E2
3	18	1	...	E3
4	24	2	0	
5	...	...	74 mrad	
6	...	...	...	

TABLE 2.6 – Erreur de phase au cours du fonctionnement d'une DDS

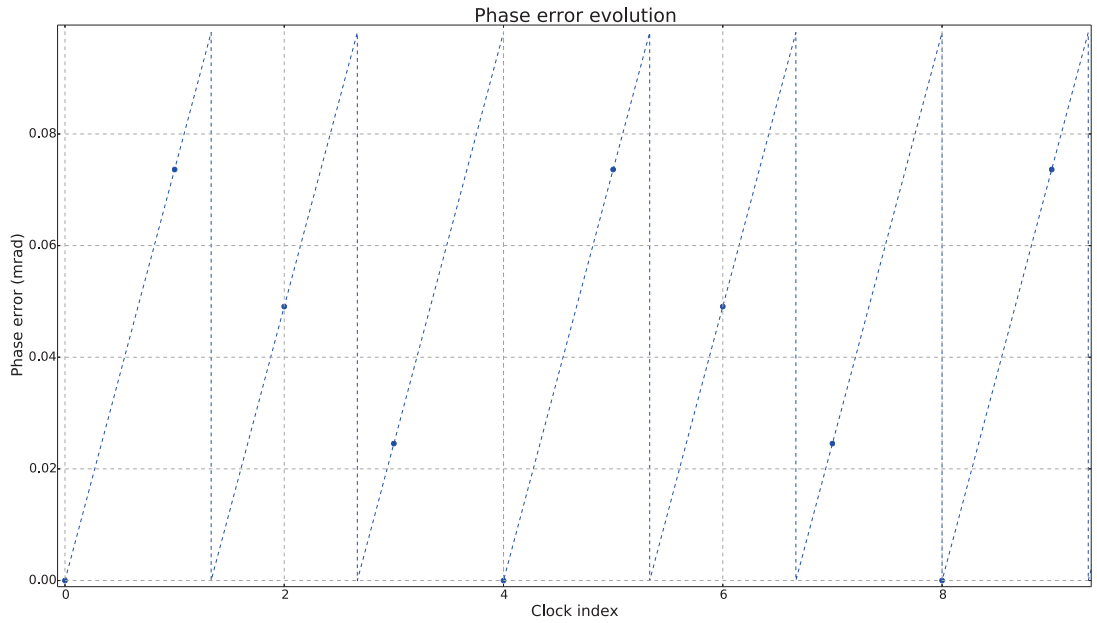


FIGURE 2.17 – Allure du signal d'erreur de phase causé par la troncature

Absence de spur Quelle que soit la DDS (ou plus exactement, la LUT), il existe toujours des fréquences qui ne causent aucun spurs liés à la troncature de phase. Pour N plus grand que L, si la valeur entrée pour $\Delta\phi$ est telle que celle présentée à la figure 2.18, il n'y a en fait aucune troncature puisque les LSB [8] ignorés, les bits 1 à $N-L$ sur le motif considéré, valent systématiquement 0 en sortie de l'accumulateur de phase.

Le bit à la position N vaut systématiquement 0 sinon le système ne respecte pas le théorème de NYQUIST-SHANNON évoqué à la section 2.3.3 [9].

Dans ce cas, le spectre obtenu en sortie de LUT ne comporte que la fréquence voulue (figure 2.19). Ce spectre ressemble alors à ceux présentés dans la section 2.3.3 pour une LUT idéale : en effet, puisqu'il n'y a pas de troncature de phase, tout se passe comme si la LUT était idéale.

8. LSB : Least Significant Bit
9. Dans la pratique, le mot de commande est câblé sur $N-1$ bit.

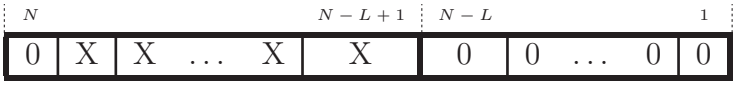


FIGURE 2.18 – Motif de mot de commande W qui ne donne pas lieu à troncature de phase

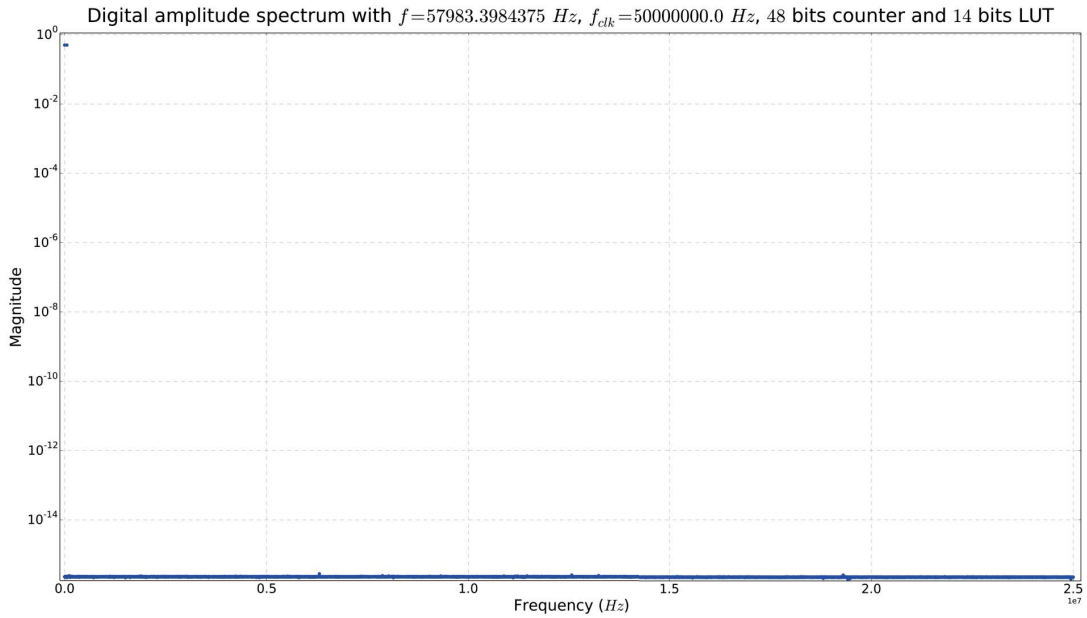


FIGURE 2.19 – Spectre de sortie en l'absence de troncature de phase : fondamental parfait (en haut, à gauche, amplitude $\frac{1}{2}$), bruit spectral confondu avec le bruit de calcul, et raies parasites à la fréquence de NYQUIST-SHANNON.

La section 2.3.4 avait d'ailleurs mentionné que certaines fréquences de sortie présentent une meilleure pureté spectrale que d'autres, sans chercher précisément ces fréquences. Il s'agit en fait de l'ensemble des fréquences qui ne donnent pas lieu à troncature de phase, soit les fréquences qui s'expriment par, avec K entier :

$$f_{out} = \frac{f_{clk}}{2^N} \times 2^{N-L} \times K$$
$$f_{out} = \frac{f_{clk}}{2^L} \times K$$

Pour ces fréquences, le paramètre L n'a aucune influence sur la pureté du spectre de sortie.

Dernier point : lorsque le bloc DDS sert à piloter un résonateur dont la fréquence évolue dans le temps (à cause de la température, de la variation de la grandeur d'entrée mesurée...), ces cas particuliers ne se présentent que ponctuellement. La relation (2.27) caractérise les entiers K qui donnent de tels spectres, pour un taux de suréchantillonnage de $400 \pm 10\%$. Si la LUT comporte moins de 10 bits d'adresse ($\frac{2^9}{880} \leqslant \frac{2^{10}}{720} < 1$), aucune fréquence non nulle ne respecte ce critère. Pour $L = 14$ par exemple, avec $f_{clk} = 50\,\text{MHz}$, la seule fréquence qui respecte ce critère autour de 60 kHz (fréquence de résonance typique

du VIA) se trouve pour $K = 20$, à 60,035 kHz.

$$\begin{aligned} 720 &\leqslant \frac{f_{clk}}{f_{out}} \leqslant 880 \\ 720 &\leqslant \frac{2^L}{K} \leqslant 880 \\ \frac{2^L}{720} &\geqslant K \geqslant \frac{2^L}{880},\ K \in \mathbb{N} \end{aligned} \tag{2.27}$$

Plus généralement, l'équation (2.28) donne la probabilité de tomber sur une telle fréquence.

$$\begin{aligned} f_{out} &\leqslant \frac{f_{clk}}{2} \\ \frac{f_{clk}}{2^L} \times K &\leqslant \frac{f_{clk}}{2} \\ K &\leqslant 2^{L-1} \\ p(K) &= \frac{2^{L-1}}{2^{N-1}} = 2^{N-L} \end{aligned} \tag{2.28}$$

Amplitude maximale À l'inverse, il existe des mots de commande W qui génèrent des spurs d'amplitude maximale.

La figure 2.16 montre que l'erreur de phase présente une périodicité quelle que soit la valeur de W. Il existe donc des cas pour lesquels l'erreur ne prend que deux valeurs par alternance : 0 et la moitié du pas de phase de la LUT. Ces cas se produisent lorsque les mots de commande présentent le motif de la figure 2.20.



FIGURE 2.20 – Motif de mot de commande qui génère les pires spurs

Lorsque le bit à la position $N - L - 1$ vaut 1 et tous les suivants 0 (le reste des LSB, Least Significant Bits), c'est-à-dire lorsqu'il s'agit d'un multiple impair de 2^{N-L-2}, alors, peu importe la valeur des MSB (Most Significant Bits), l'erreur de phase alterne entre 0 et $\frac{1}{2}\frac{2\pi}{2^L} = \frac{\pi}{2^L}$ avec une périodicité de 2 : l'erreur vaut une fois 0, une fois $\frac{\pi}{2^L}$, une fois 0, une fois $\frac{\pi}{2^L}$... Il s'agit de la plus petite périodicité envisageable avec la plus grande erreur de phase possible[10]. Plus exactement, cette erreur varie à une fréquence de $\frac{f_{clk}}{2}$.

La figure 2.21 montre des spectres obtenus dans ce cas pour deux DDS avec les paramètres de la table 2.7. Les fréquences cibles se situent autour de 61 kHz pour les deux

10. La plus grande erreur de phase possible se produit lorsque tous les LSB sont à 1. Mais dans ce cas, la période de répétition du signal d'erreur est plus grande, ce qui conduit à plus de spurs mais d'amplitude plus faibles.

Paramètre	Valeur
f_{clk}	50 MHz
N	48
L	rouge : 14 bleu : 18
D	∞

TABLE 2.7 – Paramètre de la DDS de la figure 2.21

spectres afin de conserver des taux de suréchantillonnage comparables.

La figure montre bien les spurs à 25 MHz qui correspondent à la fréquence de l'erreur de phase causée par la troncature de phase. En outre, il apparaît que la plus grande LUT (en bleu) permet de réduire l'amplitude de ce spur maximal. Plus exactement, le spur bleu à 25 MHz se trouve à un niveau 16 fois plus faible que le rouge, dû au fait que la LUT de la courbe bleue dispose de 16 (2^4) fois plus d'adresses et stocke 16 fois plus de données. L'amplitude de l'erreur de phase est donc 16 fois plus faible.

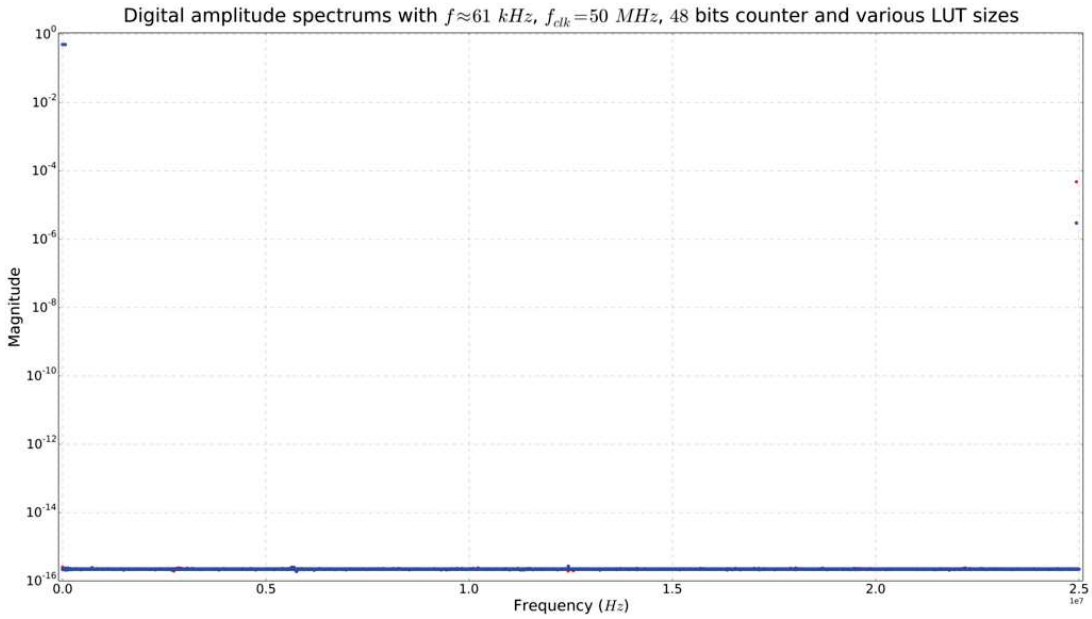


FIGURE 2.21 – Spectres de sortie dans un pire cas (LUT de 14 et 18 bits en rouge et bleu respectivement)

Étude du cas général

Les deux cas présentés jusqu'ici constituent des cas particuliers. Déterminer exactement la position et l'amplitude des spurs demande une étude analytique complète, effectuée dans [NS87]. Cette analyse fournit en particulier un algorithme pour calculer la position et l'amplitude des spurs. En résumé, la méthode commence par donner les

fréquence possibles où peuvent se trouver des spurs, puis par déterminer si un spur apparaît à cette fréquence ou non, avant de calculer l'amplitude maximale possible. Enfin, l'algorithme ne garantit pas l'unicité du maximum et ne donne pas sa (ou ses) positions.

En fonction de la consigne de phase $\Delta\phi$, ce maximum est donné par :

$$\begin{aligned}\zeta_{worst} &= \frac{\frac{\pi \gcd(\Delta\phi, 2^{N-L})}{2^{N-L}}}{2^L \sin\left(\frac{\pi \gcd(\Delta\phi, 2^{N-L})}{2^{N-L}}\right)} \\ \zeta_{worst} &= \frac{1}{2^L sinc\left(\frac{\pi \gcd(\Delta\phi, 2^{N-L})}{2^{N-L}}\right)}\end{aligned} \tag{2.29}$$

où $\gcd(a, b)$ désigne le plus grand commun diviseur de a et b.

Dans le cadre des capteurs vibrants que la DDS excitera et compte-tenu qu'au cours de son fonctionnement sur un capteur, la fréquence de sortie changera, il devient rapidement fastidieux de calculer tous les spurs pour toutes les fréquences de sortie envisagées.

En réalité, l'équation (2.29) suffit à dimensionner la DDS à réaliser. Elle permet de borner l'amplitude du pire spur et conduit à une expression beaucoup plus simple pour fixer L. Tout d'abord, il convient de noter que le plus grand commun diviseur entre 2^{N-L} et $\Delta\phi$ peut valoir au maximum 2^{N-L-1}. En effet, ce nombre ne peut s'exprimer que par des puissances de deux (et pas une somme de puissance de 2) et s'il vient à dépasser la valeur de 2^{N-L-1}, alors il n'y a plus de troncature de phase, donc plus de spurs. Ainsi, il vient :

$$\begin{aligned}0 &\leqslant \gcd(\Delta\phi, 2^{N-L}) \leqslant 2^{N-L-1} \\ 0 &\leqslant \frac{\gcd(\Delta\phi, 2^{N-L})}{2^{N-L}} \leqslant \frac{1}{2}\end{aligned}$$

Aussi, la fonction sinus cardinal est monotone décroissante sur l'intervalle $\left[0; \frac{1}{2}\right]$. D'où :

$$\begin{aligned}sinc(0) &\geqslant sinc\left(\frac{\gcd(\Delta\phi, 2^{N-L})}{2^{N-L}}\right) \geqslant sinc\left(\frac{1}{2}\right) \\ 1 &\leqslant \frac{1}{sinc\left(\frac{\gcd(\Delta\phi, 2^{N-L})}{2^{N-L}}\right)} \leqslant \frac{1}{sinc\left(\frac{1}{2}\right)} \\ \frac{1}{2^L} &\leqslant \zeta_{worst} \leqslant \frac{1}{2^L}\frac{1}{sinc\left(\frac{1}{2}\right)}\end{aligned}$$

En outre, puisque $\frac{1}{sinc\left(\frac{1}{2}\right)} \approx 1,5$:

$$\zeta_{worst} \approx \frac{1}{2^L} \tag{2.30}$$

Ainsi, L, le nombre de bits d'adresse de la LUT, détermine directement le niveau maximal des spurs en sortie de DDS. La figure 2.22 présentent des spectres typiques obtenus avec les paramètres du tableau 2.7. Les lignes en pointillés marquent l'amplitude maximale possible déterminée par la formule (2.30) : comme attendu, aucun point ne se situe au dessus, à l'exception de la fréquence cible. Le lecteur remarquera bien l'atténuation des pires spurs d'un facteur 16 entre les deux lignes en pointillés, tout comme à la figure 2.21.

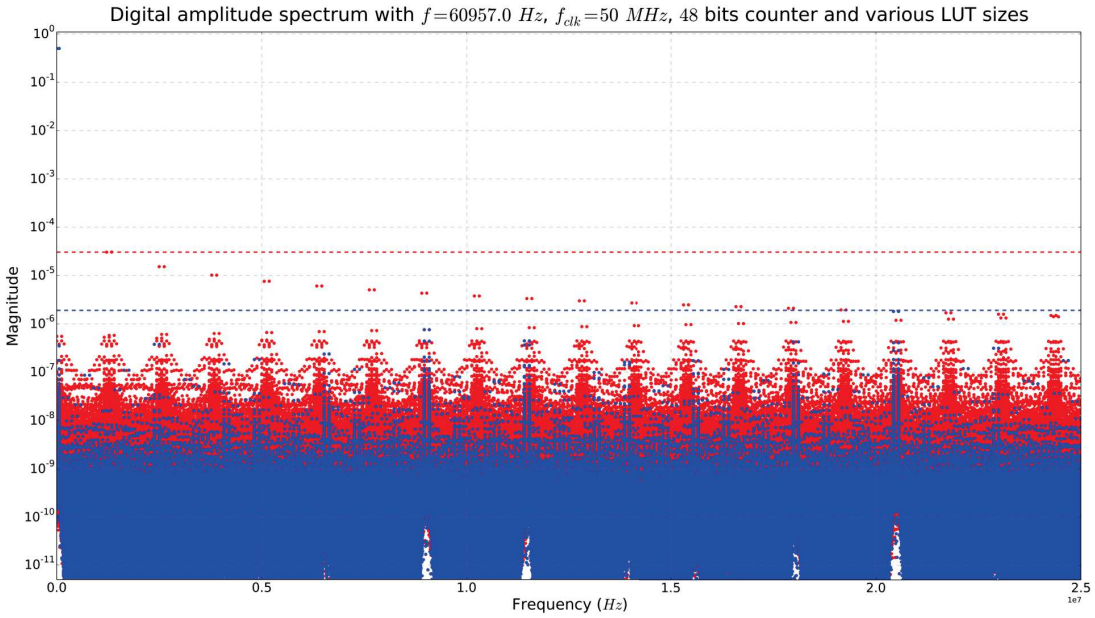


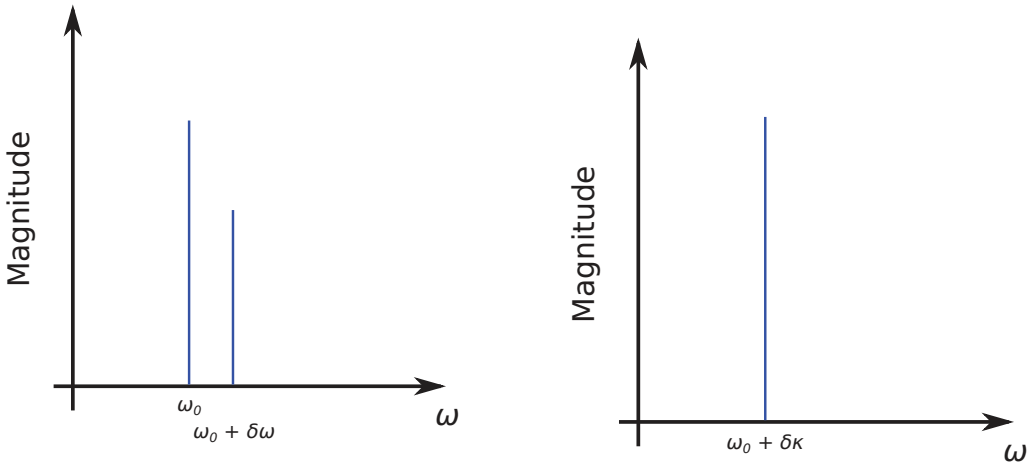
FIGURE 2.22 – Spectres de sortie dans des cas quelconques (LUT de 14 et 18 bits en rouge et bleu respectivement)

La figure montre aussi que dans les cas considérés ici, les spurs d'amplitude maximale se situent sur des harmoniques. La courbe rouge montre d'ailleurs un contenu spectral riche en harmonique alors que la courbe bleue en contient moins. Ceci provient d'un changement de taille de la LUT qui modifie la troncature de phase. Enfin, le lecteur notera que les harmoniques sur la courbe rouge décroissent en allant vers $\frac{f_{clk}}{2}$ et se retrouvent repliés par la suite. Ils se montrent légèrement décalés par rapport aux harmoniques non repliés, à des niveaux inférieurs et décroissent entre $\frac{f_{clk}}{2}$ et 0.

Relation entre le niveau des spurs et l'erreur de mesure sur la fréquence

Il reste alors à déterminer le lien entre l'amplitude b d'un spur à la pulsation $\omega_0 + \delta\omega$ et l'erreur de mesure de pulsation $\delta\kappa$ commise par le système de mesure (quel qu'il soit).

Cela conduit à une relation supplémentaire sur les paramètres de dimensionnement d'une DDS en fonction de l'erreur maximale acceptable du système de mesure.



(a) Pulsation réelle (fondamental à ω_0 et spur à $\omega_0 + \delta\omega$)

(b) Pulsation mesurée par le compteur

FIGURE 2.23 – Représentations spectrales de l'équation (2.31)

Mathématiquement, cela se traduit par l'équation (2.31), avec les paramètres suivants :

- A l'amplitude du fondamental délivré par la DDS (la fréquence désirée) ;
- ω_0 la pulsation qui correspond à ce fondamental ($\omega_0 = 2\pi f_{out}$) ;
- b l'amplitude du spur considéré ;
- $\delta\omega$ le décalage de pulsation par rapport à ω_0 de ce spur ;
- A' l'amplitude du signal mesuré par le dispositif de comptage de fréquence ;
- $\delta\kappa$ le décalage de pulsation mesuré par rapport à ω_0.
- si besoin, la fréquence mesurée sera notée $\hat{f}$ et vérifiera $\hat{f} = f_0 + \delta\nu$.

Et bien entendu, t désigne le temps.

$$A\ \cos(\omega_0\, t) + b\ \cos\left((\omega_0 + \delta\omega)\, t\right) = A'\ \cos\left((\omega_0 + \delta\kappa)\, t\right) \tag{2.31}$$

La figure 2.23 présente les spectres correspondants à chaque membre de cette équation : à gauche, le spectre réel, à droite, le spectre déduit de la mesure par le système.

Le développement des cosinus conduit à deux équations par identification des termes en cos $\omega_0 t$ et sin $\omega_0 t$.

$$A\ \cos(\omega_0\, t) + b\ \cos(\omega_0\, t)\ \cos(\delta\omega\, t) - b\ \sin(\omega_0\, t)\ \sin(\delta\omega\, t) = A'\ \cos(\omega_0\, t)\ \cos(\delta\kappa\, t) - A'\ \sin(\omega_0\, t)\ \sin(\delta\kappa\, t)$$

$$A + b\ \cos(\delta\omega\, t) = A'\ \cos(\delta\kappa\, t) \tag{2.32}$$

$$b\ \sin(\delta\omega\, t) = A'\ \sin(\delta\kappa\, t) \tag{2.33}$$

Dans le cadre des applications envisagées ici, à savoir la mesure d'accélérations à l'aide de cellules VIA, la section 1.1.3 a présenté les performances ultimes du capteur. Aussi, sur une période de mesure de 10 secondes, le système doit mesurer une variation de fréquence de 10 µHz pour conserver la résolution ultime de l'accéléromètre. Ceci fixe donc la valeur maximale acceptable pour $\delta\kappa$ à $63\,\mu\mathrm{rad\,s^{-1}}$.

Par conséquent, sur un intervalle de mesure de 10 s, il devient possible d'écrire :

$$\cos(\delta\kappa\, t) \approx 1$$

L'équation (2.32) conduit alors à :

$$A + b\ \cos(\delta\omega\, t) = A'$$

Ce qui implique, pour obtenir $A \approx A'$:

$$b\cos(\delta\omega\, t) \ll Ab \ll A$$

Des égalités (2.32) et (2.33), il vient finalement :

$$A = A'$$

$$\delta\kappa\, t = \frac{b}{A}\delta\omega\, t$$

En reprenant le modèle défini par l'équation (2.31), la présence de ce spur vient modifier les temps des passages par 0 du signal pur à la pulsation ω_0, d'un délai qui varie entre $-\frac{b}{A}\frac{1}{\omega_0}$ et $\frac{b}{A}\frac{1}{\omega_0}$.

Dans le cas d'un système par marquage temporel qui compte m périodes du signal à mesurer sur un intervalle de temps Δt, la mesure de la période mesurée T s'exprime par (équation (2.8)) :

$$T = \frac{\Delta t}{m}$$

La présence du spur modifie alors la mesure et fournit la période $\hat{T}$, dans le pire cas :

$$\hat{T} = \frac{\Delta t + 2\dfrac{b}{A}\dfrac{1}{\omega_0}}{m}$$

$$\hat{T} = \frac{\Delta t + \dfrac{b}{A}\dfrac{1}{\pi\, f_0}}{m}$$

Cette dernière équation conduit à l'erreur relative suivante sur la mesure :

$$\frac{\delta T}{T} = \frac{\delta F}{F} = \frac{\frac{b}{A}\frac{1}{\pi f_0}}{m\frac{1}{f_0}}$$

$$\frac{\delta T}{T} = \frac{\delta F}{F} = \frac{\frac{b}{A}}{m\,\pi} \tag{2.34}$$

Si r représente l'erreur relative sur la mesure à atteindre, il vient :

$$\begin{aligned} \frac{\delta F}{F} &\leqslant r \\ \frac{\frac{b}{A}}{m\,\pi} &\leqslant r \\ \frac{b}{A} &\leqslant m\,\pi\,r \end{aligned}$$

Ce rapport $\frac{b}{A}$ représente alors l'amplitude du spur normalisé par l'amplitude de la fréquence de sortie souhaitée. Autrement dit, pour que la DDS ne dégrade pas la résolution de la mesure au delà de la résolution ultime du capteur, il faut :

$$\begin{aligned} \zeta_{worst} &\leqslant m\,\pi\,r \\ \frac{1}{2^L} &\leqslant m\,\pi\,r \\ 2^L &\geqslant \frac{1}{m\,\pi\,r} \\ L &\geqslant -\log_2(m\,\pi\,r) \end{aligned} \tag{2.35}$$

L'équation (2.35) fixe ainsi le nombre de bits d'adresse L et la taille de la LUT avec l'équation (2.22) ; tous les paramètres de conception d'une DDS optimisée pour les mesures inertielles sont donc maintenant déterminés.

2.3.6 La conversion numérique-analogique

Il ne reste alors plus qu'à choisir un DAC à D bits tel que :

$$D = L - 2$$

En théorie, un DAC idéal n'affecte en rien les spectres de sortie de la DDS, qui doivent rester similaires à ceux de la figure 2.22 par exemple.

En pratique, quelle que soit la technologie du DAC retenu, ses imperfections modifient le spectre de la sortie. Il s'avère assez difficile de prévoir quantitativement l'impact de chaque défaut, mais il est parfois possible de prévoir leurs effets de façon qualitative.

Sans entrer dans les détails, les défauts possibles d'un DAC peuvent se diviser en trois catégories :

— les erreurs de quantification ;
— les non linéarités ;
— les erreurs de monotonie.

L'erreur de quantification provient du fait qu'un DAC ne délivre que des valeurs discrètes. Même avec un DAC parfait, dès que la valeur analogique souhaitée n'est pas un multiple du pas de quantification, il existe une différence entre la valeur souhaitée et la valeur délivrée par le DAC : c'est l'erreur de quantification. Il s'agit donc d'une erreur inhérente à la conversion et pas d'un défaut du convertisseur : même un DAC parfait présente ce défaut.

Si q représente le pas de quantification du DAC, ΔX l'étendue des valeurs analogiques possibles, et D le nombre de bits du DAC, alors :

$$q = \frac{\Delta X}{2^D - 1}$$
$$q = \frac{\Delta X}{2^D}, D \geqslant 7 \tag{2.36}$$

Par exemple, un DAC en tension de 10 bits avec une pleine échelle de 3,3 V dispose d'un pas de 3,2 mV.

Pour chaque valeur possible du DAC, l'erreur de quantification possible varie alors entre $-\frac{q}{2}$ et $\frac{q}{2}$ et reste centrée sur 0.

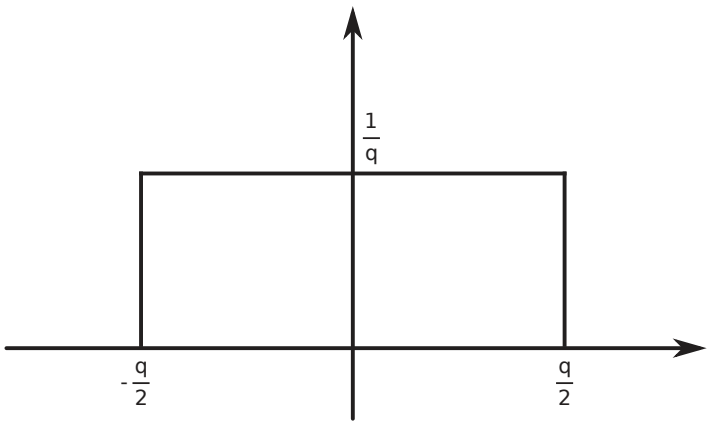


FIGURE 2.24 – Probabilité d'erreur de quantification d'un DAC

Dans le cas d'un signal quelconque, l'erreur de quantification présente une densité de

probabilité constante entre $-\frac{q}{2}$ et $\frac{q}{2}$, et nulle partout ailleurs. La somme des probabilités étant de 1, la densité de probabilité est donc de $\frac{1}{q}$. Cela conduit à un bruit additif indépendant de la fréquence (bruit blanc) sur le signal converti de puissance RMS $\frac{q^2}{12}$ (équation (2.37), avec x le signal à convertir, $\hat{x}$, le signal fourni par le DAC et k, l'instant de l'échantillonnage).

$$\begin{aligned} e(k) &= x(\hat{k}) - x(k) \\ \sigma_e^2 &= \int_{-\frac{q}{2}}^{\frac{q}{2}} e^2 \frac{1}{q}\, de \\ \sigma_e &= \frac{q}{\sqrt{12}} \end{aligned} \tag{2.37}$$

Dans le cas d'un DAC réel, les erreurs de quantification correspondent à l'écart entre les valeurs de sortie et la droite théorique en pointillés noirs sur la figure 2.25. Pour un DAC idéal, cette erreur prend la forme d'une dent de scie centrée sur 0 et d'amplitude $\frac{q}{2}$. Cependant, les valeurs discrètes de sortie peuvent s'éloigner de cette droite et suivre, par exemple, une loi hyperbolique (en vert sur la figure) ou polynomiale de puissance impaire (en rouge). Il s'agit alors d'erreur de non linéarité puisque la progression du DAC ne suit plus une droite.

L'effet de ces non linéarités va dépendre de la répartition des erreurs vis à vis de la valeur médiane du DAC (2^{D-1}, centre du repère sur la figure 2.25). Dans le cas où les erreurs se situent toujours du même côté de la droite théorique (en pointillés) et conservent une symétrie par rapport à cette valeur médiane (cas de la courbe verte), c'est-à-dire que l'erreur en $2^{D-1} + x$ égale celle en $2^{D-1} - x$, alors la sinusoïde en sortie perd sa symétrie (entre les lobes inférieurs et supérieurs), ce qui engendre des harmoniques paires sur le spectre final.

À l'inverse, si les erreurs se répartissent d'un côté de la droite théorique pour les valeurs inférieures à la valeur médiane du DAC, et de l'autre côté pour les valeurs supérieures, tout en formant une antisymétrie (cas de la courbe rouge), c'est-à-dire que l'erreur en $2^{D-1} - x$ vaut l'opposé exact de celle en $2^{D-1} + x$, alors la sinusoïde de sortie conserve sa symétrie (entre les parties supérieures et inférieures), mais se trouve déformée. Dans ce cas, les harmoniques impairs se dégagent sur le spectre final obtenu.

Le dernier défaut important d'un DAC, et peut-être le plus critique, est l'erreur de monotonie. En principe, un DAC doit fournir des valeurs strictement croissantes entre sa commande la plus basse et sa commande la plus élevée [11].

11. Les DAC signés (c'est-à-dire qui accepte des entiers relatifs à leur entrée, disposent en réalité d'un système de translation qui va assigner les commande supérieures à la valeur médiane aux entiers positifs, et inférieure à la médiane aux entiers négatifs. Un DAC ne peut pas délivrer de tension, courant... négatif

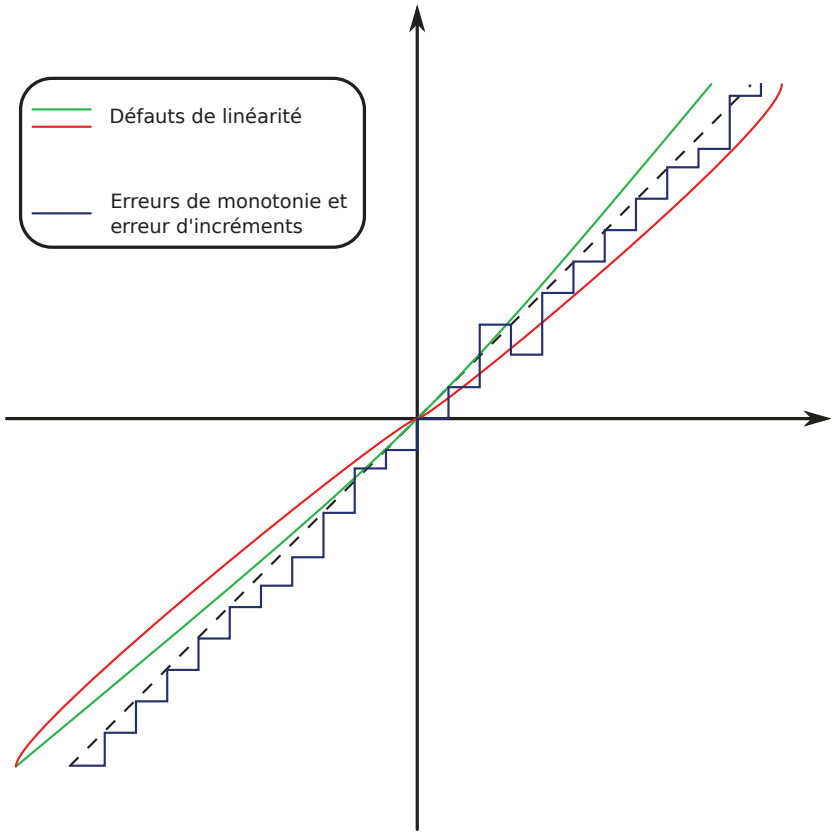


FIGURE 2.25 – Représentation des erreurs de non-linéarité d'un DAC

Deux cas de figure se présentent alors :

— soit l'erreur demeure ponctuelle ;

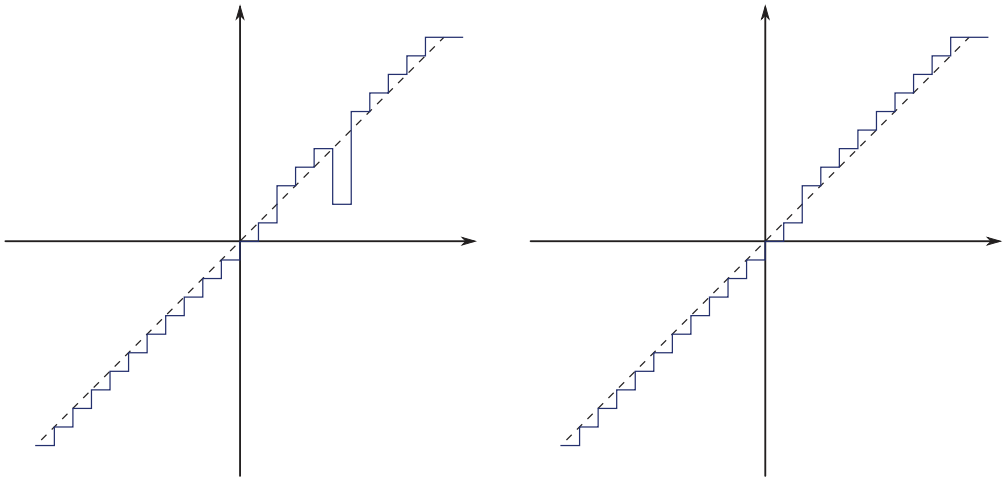
— soit l'erreur provoque un décalage.

Dans le premier cas (courbe en bleu sur la figure 2.25), cela cause une erreur de monotonie car la valeur erronée s'avère inférieure ou égale à la valeur précédente, ou bien égale ou supérieure à la suivante. Dans le second cas, la valeur en n délivre en réalité la valeur que le DAC devrait fournir en $n \pm x$. Si cette valeur en n se révèle inférieure à la précédente, la rupture de monotonie apparaît évidente (cas de la figure 2.26a). Si maintenant, cette valeur en n se montre supérieure à la valeur précédente, le DAC ne pourra pas poursuivre une monotonie stricte jusqu'à sa pleine échelle (cas de la figure 2.26b ; la dernière et l'avant-dernière valeurs sont identiques).

Dans tous les cas, ces imperfections sur la monotonie du DAC dégradent notablement le spectre de sortie de la DDS en provoquant des ruptures de symétrie de la sinusoïde de sortie.

Si le DAC présente des erreurs de monotonie, on pourra réarranger les mots stockés dans la LUT de façon à rétablir la monotonie stricte en sortie du DAC. Par exemple, si

de toutes façons. Donc tout se rapporte au cas où la commande 0 fournit une valeur nulle et la commande la plus élevée, la pleine échelle du DAC.



(a) Erreur locale de monotonie (b) Non linéarité différentielle

FIGURE 2.26 – Erreurs de linéarité d'un DAC

le DAC inverse les valeurs de sortie pour n et m, alors il suffit d'inverser les mots stockés dans la LUT aux adresses qui alimentent normalement le DAC avec ces valeurs n et m. Ainsi, la sortie de la LUT ne correspond plus à une sinusoïde parfaite, mais compense les défauts du DAC de telle sorte que la sortie demeure bien sinusoïdale.

Afin de ne pas dégrader les performances de la DDS, le DAC doit donc disposer d'un nombre de bits assez important.

La conversion numérique-analogique par modulation Σ/Δ

Les DAC sont des circuits intégrés mixtes analogiques/numériques. Une fonction DAC ne peut pas être entièrement synthétisée sur composant numérique programmable comme un accumulateur de phase et une table d'onde précédemment. En particulier, un tel composant fait l'objet d'un choix de dimensionnement chez un constructeur, qui le fabrique et le propose à son catalogue. Dans le domaine spatial, les convertisseurs qualifiés sont peu nombreux et même sujets à obsolescence. Pour contourner les dépendances (vis à vis de références, de fabricants ou de technologies), ce chapitre explore un procédé de conversion plus intégré du point de vue numérique, qui réduit au minimum les étages analogiques : la conversion Σ/Δ.

Dans cette technique, la sortie varie entre 0 et 1 (où 1 représente la pleine échelle du DAC, 3,3 V par exemple), tel que sur la courbe cyan de la figure 2.27. Cette figure présente aussi en rouge la sortie d'un DAC conventionnel et en bleu la sinusoïde théorique recherchée. La rapidité de l'échantillonnage (donc le suréchantillonnage) vient compenser le manque de résolution sur le pas de quantification.

D'un point de vue algorithmique, un modulateur Σ/Δ fonctionne ainsi. Le DAC accepte en entrée des mots de D bits. Au front d'horloge, le DAC lit le mot en entrée et

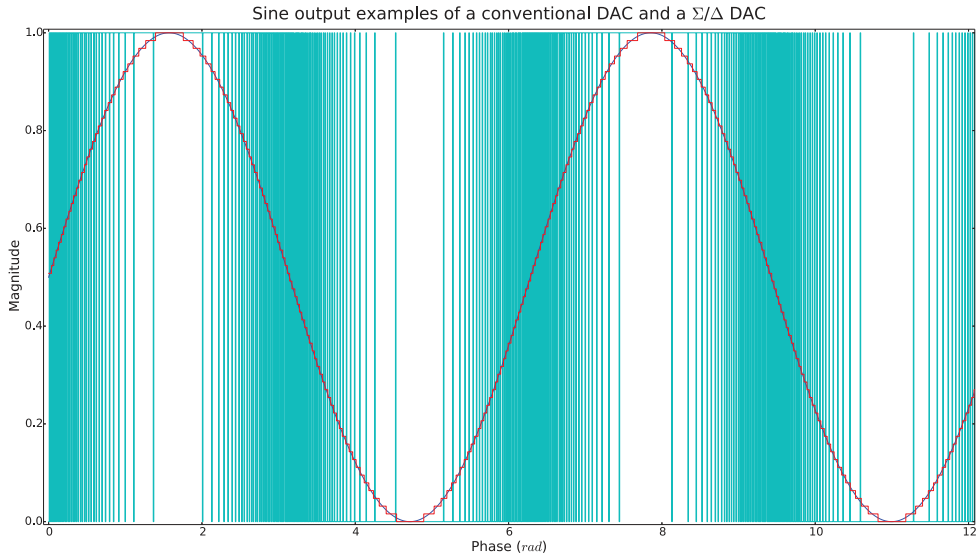


FIGURE 2.27 – Comparaison des sorties d'un DAC conventionnel et d'un convertisseur Σ/Δ par rapport à une sinusoïde de référence

l'additionne au résidu (la suite va expliquer le calcul de ce résidu). Si cette somme dépasse 2^{D-1}, alors le DAC place sa sortie à 1 et retranche à cette somme 2^{D-1} ; cela forme le résidu pour la prochaine opération. Si cette somme ne dépasse pas 2^{D-1}, alors le DAC place sa sortie à 0 et la somme courante devient le résidu pour l'opération suivante.

Au front d'horloge suivant, le DAC réitère l'algorithme avec le nouveau résidu obtenu.

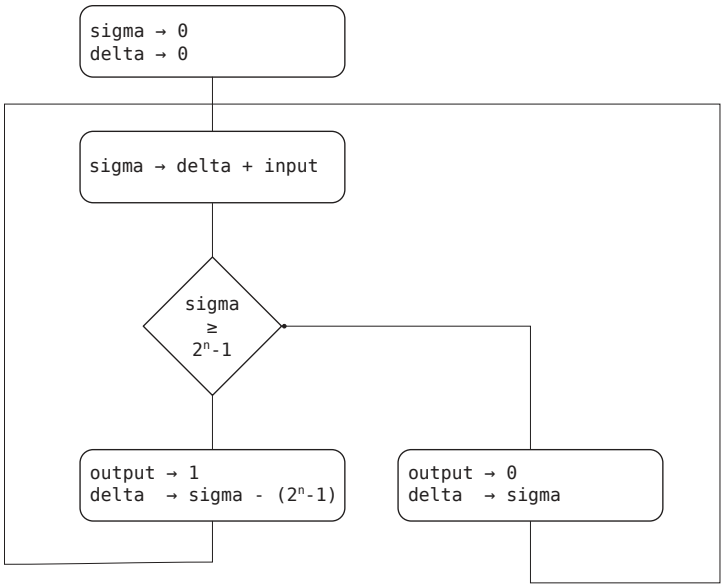


FIGURE 2.28 – Algorithme d'un DAC Σ/Δ

La figure 2.28 décrit sous forme schématique cet algorithme, dit du premier ordre, avec Y la sortie du DAC et X, le signal en entrée. La figure 2.29 présente un schéma bloc d'un modulateur Σ/Δ où Q désigne le bruit de quantification. Avec $I(z)$ la fonction de

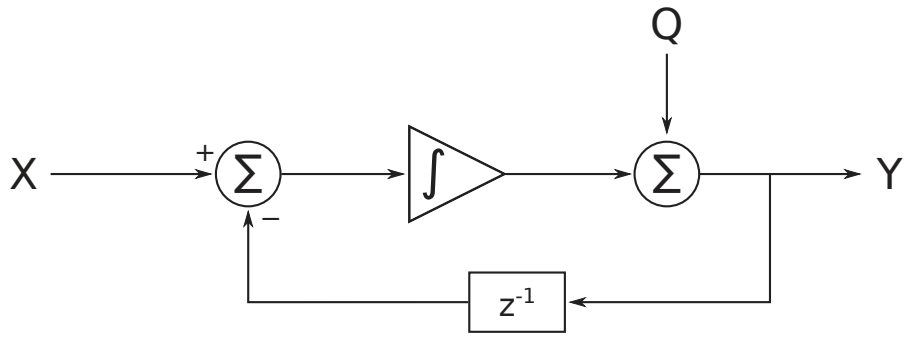


FIGURE 2.29 – Schéma bloc d'un modulateur Σ/Δ

transfert d'un intégrateur, ce schéma mène à écrire :

$$Y(z) = \left[X(z) - z^{-1}Y(z)\right] I(z) + Q(z)$$

Il en découle l'équation (2.38), avec $I(z) = \frac{1}{1-z^{-1}}$.

$$\begin{aligned} Y(z)\left[1 + z^{-1}I(z)\right] &= X(z)I(z) + Q(z) \\ Y(z) &= \frac{I(z)}{1 + z^{-1}I(z)}X(z) + \frac{1}{1 + z^{-1}I(z)}Q(z) \\ Y(z) &= X(z) + (1 - z^{-1})Q(z) \end{aligned} \tag{2.38}$$

Cette équation montre que la fonction de transfert du signal (Signal Transfer Function $STF(z)$) est égale à 1.

Pour un signal de sortie binaire Y entre $-\frac{X_{max}}{2}$ et $\frac{X_{max}}{2}$, le signal d'erreur Q prend ses valeurs entre $-X_{max}$ et X_{max}, avec une densité spectrale à priori uniforme (bruit blanc). Cette densité spectrale s'exprime [VE] par :

$$n_Q(f) = \frac{X_{max}}{\sqrt{f_{clk}}\sqrt{6}} \tag{2.39}$$

Il s'agit bien d'une densité spectrale en unité de X par $\sqrt{\text{Hz}}$.

La fonction de transfert $(1 - z^{-1})$ s'écrit (passage de la transformée en z à la transformée de FOURIER discrète en posant $z = e^{2j\pi\frac{f}{f_{clk}}}$) :

$$\begin{aligned} 1 - z^{-1} &= 1 - e^{-2j\pi\frac{f}{f_{clk}}} \\ 1 - z^{-1} &= e^{-j\pi\frac{f}{f_{clk}}}\left(e^{j\pi\frac{f}{f_{clk}}} - e^{-j\pi\frac{f}{f_{clk}}}\right) \\ 1 - z^{-1} &= e^{-j\pi\frac{f}{f_{clk}}} 2j \sin\left(\pi\frac{f}{f_{clk}}\right) \end{aligned} \tag{2.40}$$

Le premier terme $e^{-j\pi\frac{f}{f_{clk}}}$ est un retard pur de module 1. Le deuxième terme $2j\sin\left(\pi\frac{f}{f_{clk}}\right)$ s'assimile à un dérivateur $2\pi\frac{f}{f_{clk}}$ en basses fréquences, lorsque $f \ll f_{clk}$ (le modulateur Σ/Δ est conçu pour travailler avec un large sur-échantillonnage).

Compte-tenu de l'expression de la fonction de transfert (2.38), la densité spectrale de bruit d'un modulateur Σ/Δ s'exprime alors par :

$$n_Y(f) = 2\pi\frac{f}{f_{clk}}\frac{X_{max}}{\sqrt{f_{clk}}\sqrt{6}} \tag{2.41}$$

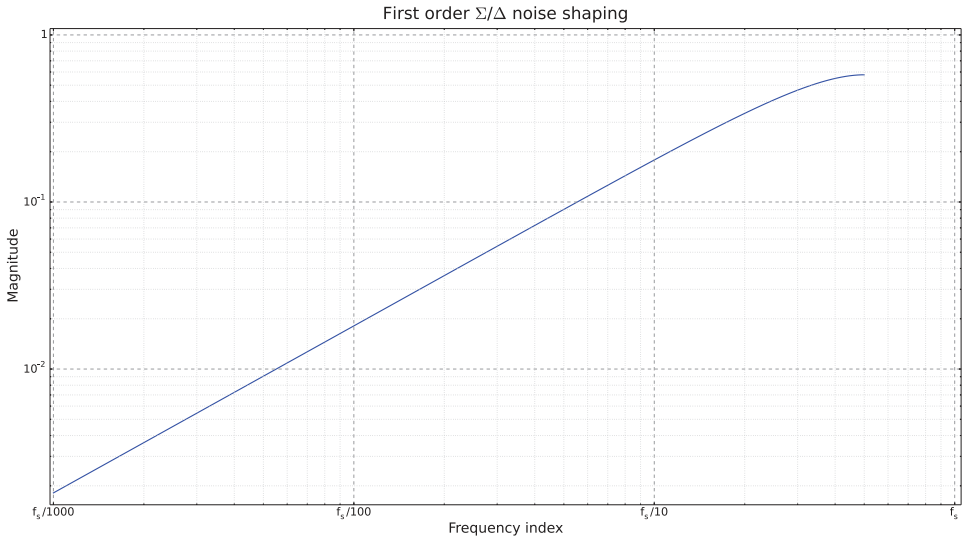


FIGURE 2.30 – Densité spectrale de bruit d'un DAC Σ/Δ du premier ordre (normalisée par rapport à la fréquence d'échantillonnage et au pas de quantification.

La figure 2.30 met en évidence l'effet dérivateur sur le bruit de la modulation Σ/Δ. Pour augmenter la bande passante utile, c'est-à-dire la bande de fréquence où le bruit du convertisseur demeure suffisamment faible, une option consiste à augmenter la fréquence d'échantillonnage. Cependant, si une seule horloge est utilisée dans le système, cela implique d'augmenter le pas de fréquence de la DDS.

Il demeure possible d'utiliser deux horloges, une pour l'accumulateur de phase et une autre pour le convertisseur. Cela complique alors l'architecture et pose des problèmes de couplage parasite de par leur proximité dans le circuit : une horloge affecte l'autre et réciproquement, ce qui conduit à une augmentation de leurs bruits de phase.

L'algorithme décrit ci-dessus est celui d'un convertisseur du premier ordre. Il existe aussi des convertisseurs d'ordre supérieur qui améliorent la mise en forme du bruit («noise shaping») pour augmenter la bande passante utile [NST97]. Ces convertisseurs mettent en jeu plusieurs additionneurs et plusieurs boucles de rétroaction. Le nombre d'additionneurs (qui font office d'intégrateurs numériques) définit alors l'ordre du convertisseur.

Dans le cadre de cette thèse (environnement spatial), ce type de convertisseurs convient parfaitement puisqu'un condensateur et une résistance suffisent pour réaliser un filtre satisfaisant pour rejeter les fréquences au delà de 2 à 3 fois la fréquence souhaitée (afin de ne pas exciter de mode de résonance non souhaité sur les capteurs). Ces composants sont facilement remplaçables et ne présente pas de problèmes de disponibilité ou d'obsolescence. Par ailleurs, cela maintient une électronique aussi numérique que possible et réduit la partie analogique au minimum.

Bien entendu, l'emploi d'un DAC multi-bits, couplé à une modulation Σ/Δ, réduit encore plus le bruit de quantification sans changer la fréquence d'échantillonnage ; la densité spectrale de bruit est celle de l'équation (2.41) avec un pas de quantification plus fin : $X_{max}/2^B$, B nombre de bits du DAC.

Paramètre	**Valeur**
f_{clk}	50 MHz
N	48
L	14
D, à l'entrée du système modulateur Σ/Δ-DAC	16
D', après modulateur Σ/Δ	4

TABLE 2.8 – Paramètres de la DDS des figures 2.31, 2.32 et 2.33

La figure 2.31 montre la sortie d'une DDS avec un DAC 4 bits associé à une modulation Σ/Δ. Le tableau 2.8 précise les valeurs de tous les paramètres de cette simulation. Dans ce cas, le modulateur Σ/Δ accepte des mots de 16 bits en entrée pour respecter l'équation (2.22) ($16 = 14 + 2$) et délivre en sortie des mots de 4 bits pour alimenter le DAC analogique. Les valeurs en sortie s'échelonnent donc entre 0 et $2^{16} - 1 = 65535$; 65535 représente donc la pleine échelle de la valeur analogique de sortie du DAC analogique, 3,3 V par exemple.

La figure 2.32 présente le spectre du signal de la figure 2.31 avec une échelle linéaire pour les fréquences (attention, la valeur maximale n'est plus $0,5$ cette fois-ci, mais $\frac{65535}{2}$ en $f = 0$ et $\frac{65535}{\sqrt{2}}$ en f_{out}), alors que la figure 2.33 dispose d'une échelle logarithmique pour les fréquences.

Ces figures montrent qu'autour de la fréquence souhaitée, les fréquences parasites se situent environ 5 ordres de grandeur plus bas, du fait de la LUT de 14 bits (équation (2.30)) et de la mise en forme du bruit par le modulateur $\Sigma\Delta$. Aux fréquences plus élevées, elles sont plus présentes mais ceci a peu d'importance puisque le filtre placé en sortie de la DDS atténuera ces bruits hautes fréquences.

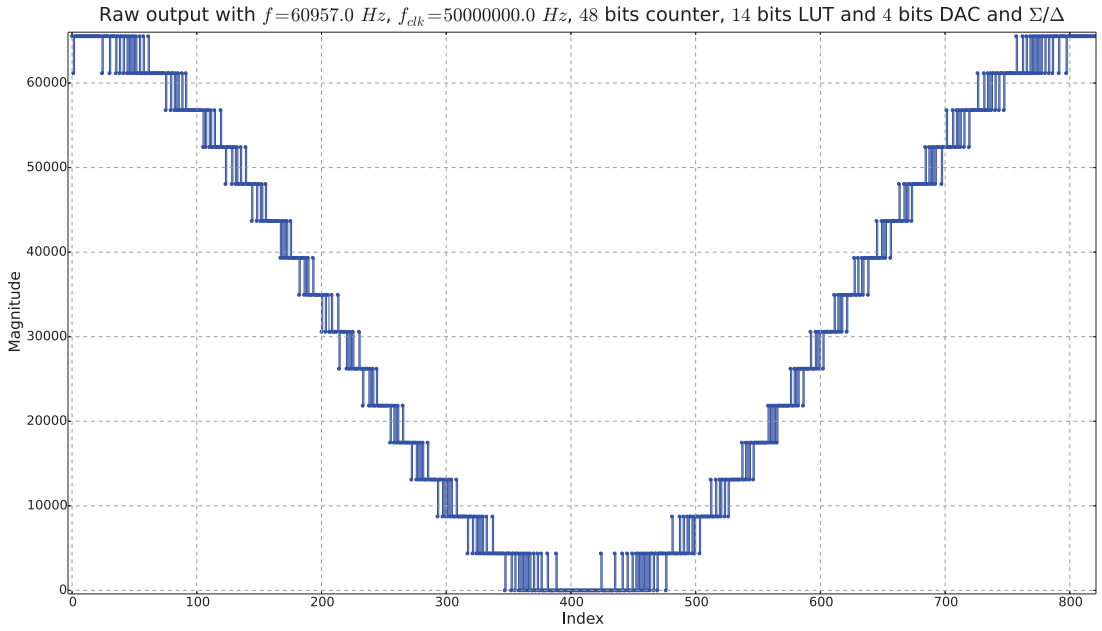


FIGURE 2.31 – Sortie d'une DDS avec un DAC de 4 bits et un convertisseur Σ/Δ

2.3.7 Conclusion sur le dimensionnement d'un synthétiseur numérique direct

Le tableau 2.3 a présenté les paramètres qui définissent une DDS et ses performances, à savoir :

— f_{clk} la fréquence d'horloge du système ;
— N le nombre de bits de l'accumulateur de phase ;
— L le nombre de bits d'adresse de la LUT ;
— D le nombre de bits du DAC si il y a lieu.

Le début de la section a ensuite établi des liens ou des contraintes entre ces paramètres.

Tout d'abord, f_{clk} et N fixent le pas de fréquence, selon l'équation (2.17), rappelée ci-après.

$$\delta\phi = \frac{f_{clk}}{2^N}$$

Ces paramètres n'influent en rien sur l'amplitude maximum des spurs observés sur le spectre de sortie (c.f formule (2.30)).

La section 2.3.4 a montré que la DDS peut engendrer un ensemble restreint de fréquences. Toutefois, il demeure possible d'obtenir en moyenne une fréquence quelconque. En boucle fermée, avec par exemple une DDS à pas de fréquence de 1 Hz, si l'utilisation visée nécessite une fréquence de 3,75 Hz, alors la DDS génère une sinusoïde à 3 Hz un quart du temps, et 4 Hz le temps restant. En moyenne, la DDS fournit ainsi une sinusoïde à 3,75 Hz.

En outre, une recherche d'optimisation de l'usage d'un DAC multi-bits conduit à lier

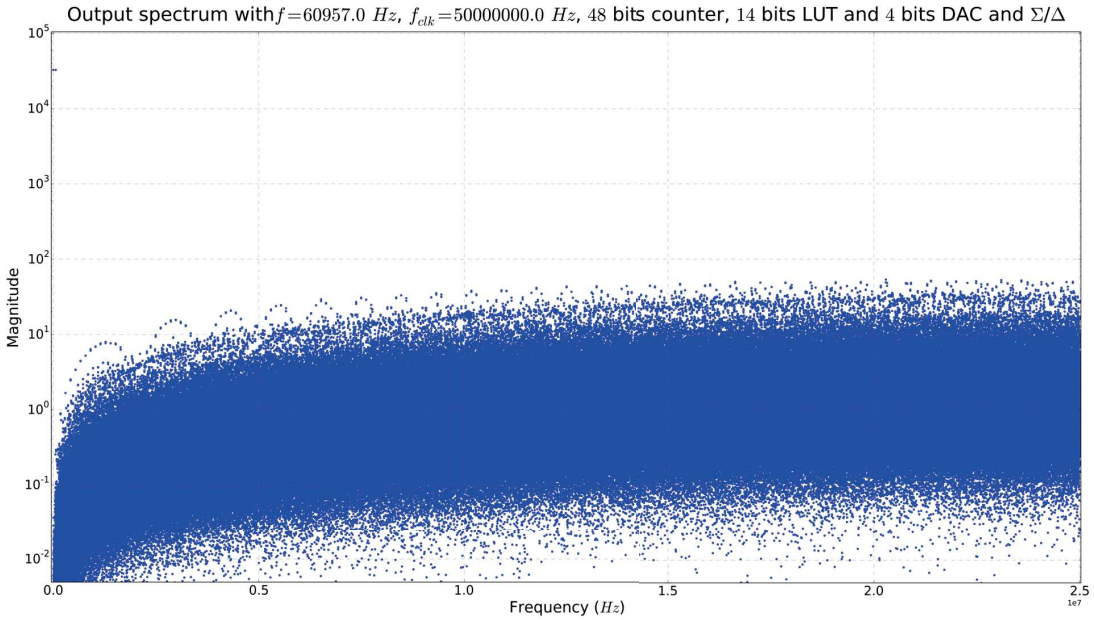


FIGURE 2.32 – Spectre du signal de la figure 2.31

la taille de la LUT, c'est-à-dire son nombre de bits d'adresse (donc le nombre de mots stockés), et le nombre de bits du DAC. L'équation (2.22), ci-après exprime cette contrainte.

$$L = D + 2$$

L'étude des spectres de sortie par simulation révèle plusieurs contraintes importantes à prendre en considération lors de la conception de la DDS.

Tout d'abord, l'amplitude des spurs, ou du moins celle du maximun, ne dépend que de la taille de la LUT. L'équation (2.29) donne l'amplitude de ce maximum. L'étude de l'influence de ces spurs sur la résolution d'une mesure inertielle conduit à l'équation (2.35) :

$$L \geqslant -\log_2 (m\,\pi\,r)$$

Avec les notations suivantes :

— $\log_2$ désigne le logarithme en base 2 ;

— m représente le nombre de périodes comptées sur un intervalle de mesure ;

— r note l'erreur relative recherchée.

Bien que seul L donne une borne supérieure à l'amplitude des spurs, la fréquence d'horloge du système, f_{clk}, la fréquence de sortie souhaitée f_{out} et le nombre de bits de l'accumulateur de phase N déterminent le nombre et les fréquences des spurs observés. Le rapport $\frac{f_{clk}}{f_{out}}$, ou taux de suréchantillonnage, à un facteur 2 près, joue notamment un rôle prépondérant sur le nombre de ces spurs.

Dans la réalité cependant, le bruit de phase de f_{clk} influe sur les spurs observés, bien que nous ayons considéré f_{clk} comme une horloge de référence pure, sans défaut.

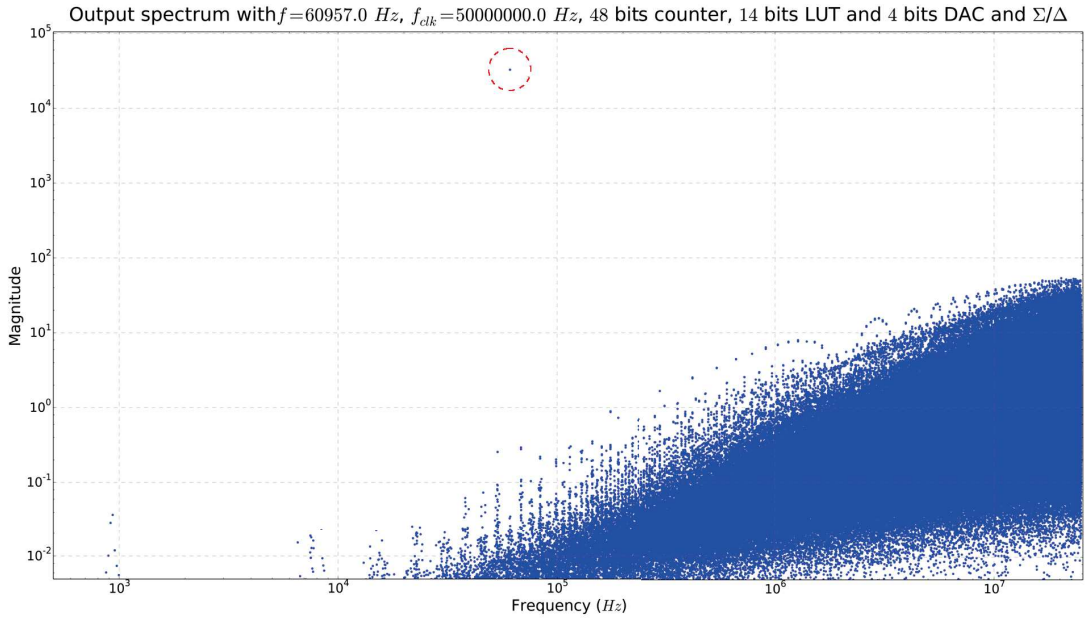


FIGURE 2.33 – Spectre du signal de la figure 2.31 en échelle logarithmique. Le fondamental de la sinusoïde est présent à l'amplitude 3×10^4, bien au dessus du bruit et des spurs.

Enfin, le DAC idéal n'existe pas et ses divers défauts viennent enrichir le contenu spectral. En particulier, certains défauts relativement courants peuvent rompre la symétrie de la sinusoïde de sortie obtenue et faire apparaître des harmoniques pairs ou impairs.

Par ailleurs, dans le cadre d'une utilisation spatiale de ce synthétiseur, les DAC constituent des éléments sensibles. L'utilisation d'une modulation Σ/Δ réduit leur caractère critique, et peut même permettre de s'en dispenser complètement. Cette modulation rejetant le bruit de quantification en haute fréquence, il faut veiller à conserver un large taux de suréchantillonnage lors de la réalisation.

Ce chapitre a donné les clés qui serviront à dimensionner les blocs numériques de l'architecture électronique retenue, le compteur de fréquence pour le marquage temporel et le bloc DDS pour le pilotage des capteurs [MGM+14], ainsi que la puissance nécessaire au calcul de la fréquence mesurée par régression linéaire, méthode qui utilise au maximum les donnée du marquage temporel. Il ne reste qu'à appliquer ces formules au dimensionnement de ces blocs pour une utilisation avec les capteurs inertiels de l'Onera, le VIA et le VIG notamment.

Chapitre 3

Réalisation et évaluation du compteur et du synthétiseur de fréquence

Sommaire

3.1 Conception d'un compteur sur Spartan 6 81
3.1.1 Introduction à la programmation du Spartan 6 81
3.1.2 Les DSP48 82
3.1.3 Cas particulier d'additionneur 83
3.1.4 Les registres de capture et les monostables 84
3.1.5 Architecture du périphérique 85
3.2 Caractérisations 85
3.2.1 Mesure simple 88
3.2.2 Mesure différentielle 89
3.3 Analyse des résultats 91
3.3.1 Identification de l'origine de la perturbation 92
3.3.2 Modification du mode de création des horloges 95
3.4 Estimation du déphasage 98
3.4.1 Référence de phase du signal pilote du résonateur 98
3.4.2 Mesure du déphasage à l'aide du compteur seul 99
3.4.3 Utilisation des données de la DDS en boucle fermée 102
3.5 Correction du déphasage 103
3.6 Conception d'une DDS 105
3.6.1 Architecture du périphérique 105
3.6.2 Mémoires dans le FPGA 106
3.6.3 Dimensionnement de la DDS 110
3.7 Caractérisation du synthétiseur 113

3.7.1 Architecture et paramètres 113
3.7.2 Caractérisation spectrale 116

Le FPGA retenu pour les expérimentations est un Spartan 6 de Xilinx. Il a été choisi en début de thèse pour son coût modeste et la disponibilité de ses outils de développements. La famille Spartan constitue l'entrée de gamme chez Xilinx, la série Spartan 6 apparaissant alors en remplacement de la série Spartan 3 vieillissante. Les familles Virtex constituent quant à elles le haut de gamme, et sont du reste les seules à proposer des références qualifiées spatiales. Mais il n'était pas nécessaire d'acquérir un composant qualifié pour développer nos fonctions numériques ; écrites en VHDL, leur portage de Spartan vers des versions qualifiées et même vers des FPGA d'autres constructeurs est garanti par le langage.

Une autre raison du choix de Xilinx était l'environnement de développement ISE (désormais remplacé par VIVADO) réputé assez convivial et utilisé par une communauté d'utilisateurs assez large. Un niveau de licence assez économique permettait de synthétiser un microcontroleur complet (processeur, mémoire et bus) pour le pilotage des fonctions développées ci-après, et leur interfaçage avec l'ordinateur hôte.

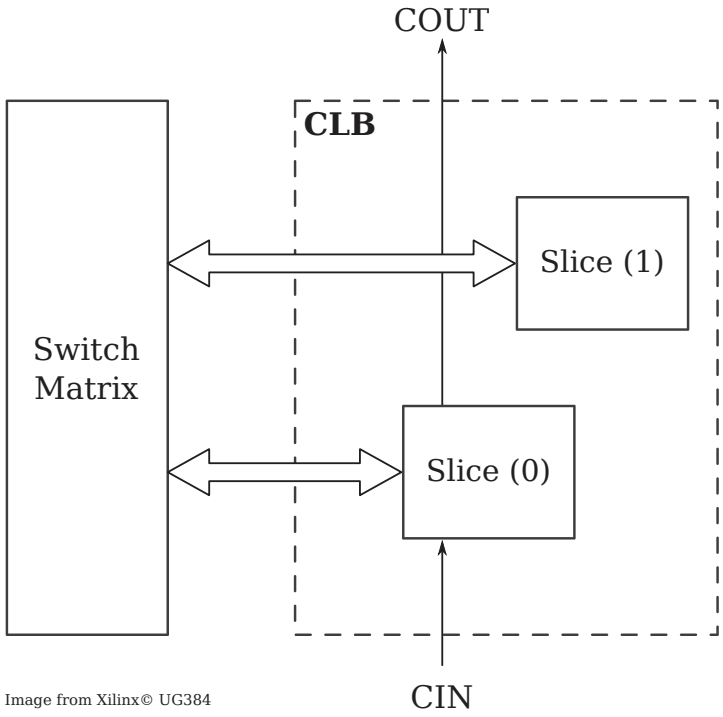


FIGURE 3.1 – Organisation d'un *Configurable Logic Block*

3.1 Conception d'un compteur sur Spartan 6

La section 2.1 n'a pas fait d'hypothèse sur l'architecture du compteur. Les architectures les plus simples mettent en oeuvre des bascules câblées entre elles pour incrémenter de 1 le mot binaire qu'elles représentent à chaque front d'horloge. Avec un peu de généralisation, un compteur est aussi un accumulateur avec comme donnée d'entrée la constante 1. Il peut enfin être avantageusement réalisé sur des macro-blocs spécifiques du FPGA, les DSP48.

3.1.1 Introduction à la programmation du Spartan 6

Le *Configurable Logic Block* ou CLB présenté à la figure 3.1, constitue l'élément de base des FPGA Xilinx, inventeur des FPGA. Un FPGA se compose d'une multitude de CLB, reliés entre eux par une matrice d'interconnexion. Un CLB comporte lui-même deux *slices*, des unités logiques de base, terme qui n'est presque jamais traduit en Français. Il est important de noter sur la figure 3.1 qu'elles ne sont pas reliées entre elles autrement que par la matrice d'interconnexion. De plus, seule une des deux *slices* dispose d'un système de propagation de retenue (CIN et COUT), ce qui implique que la moitié seulement du FPGA est optimisée pour des opérations arithmétiques sur des nombres plus grands que ceux que peut traiter un CLB de façon autonome (sans requérir à ses voisins).

Une *slice* de FPGA implémente effectivement la fonctionnalité logique voulue par l'utilisateur ; c'est la brique de base pour construire toutes les fonctions logiques. Dans

un Spartan 6, Xilinx propose trois types différents de *slices*, qui permettent de réaliser de simples opérations de logique combinatoire, jusqu'à des registres à décalage de 32 bits ou des blocs mémoire de 256 bits. Le tableau 3.1 détaille les caractéristiques des différentes *slices*. Dans tous les cas, les slices s'appuient sur des LUT et des bascules D pour réaliser la fonction voulue par l'utilisateur.

Caractéristiques	**slicex**	**slicel**	**slicem**
LUT 6 entrées	X	X	X
8 bascules	X	X	X
Multiplexeurs larges [a]		X	X
Logique de retenue		X	X
RAM distribuée			X
Registres à décalage			X

a. Les multiplexeurs larges offrent plus de choix pour connecter les entrées des LUT, notamment les retenues par exemple.

TABLE 3.1 – Caractéristiques des slices en fonction de leur type.

Les LUT contenues dans les *slices* stockent les tables de vérités des fonctions logiques programmées afin de les exécuter, contrairement à un ASIC [1] dans lequel les portes logiques optimisent l'encombrement, la consommation, la vitesse d'une fonction spécifique. Par conséquent, une implémentation sur FPGA est toujours plus lente qu'un circuit spécifique réalisé dans une technologie équivalente. Cependant, les technologies actuelles permettent la conception de FPGA suffisamment rapides pour satisfaire les besoins de beaucoup d'applications.

3.1.2 Les DSP48

Pour les opérations arithmétiques complexes, le Spartan 6 possède des unités spécialisées : les DSP48. En effet, même si l'utilisation de quelques CLB permet d'atteindre localement des fréquences de fonctionnement élevées, leur utilisation en nombre (sur une surface étendue dans le FPGA donc) limite la fréquence de fonctionnement d'opérations plus complexes. Ainsi, plus une fonction occupe de l'espace (et donc des CLB), plus les fréquences de fonctionnement atteignables diminuent en raison des limitations de l'arbre d'horloge interne (le décalage du temps d'arrivée entre deux fronts d'horloge devient trop important). Or, dans le cas d'une mesure de fréquence, plus le compteur de référence compte vite plus il devient possible de mesurer des fréquences élevées ou d'obtenir une meilleure précision à long terme (voir la section 2.1).

1. Application-Specific Integrated Circuit, circuit intégré spécifique à une application.

Un des buts principaux de l'électronique numérique demeure le traitement de données, soit bien souvent additions, multiplications et décalages. C'est pourquoi le Spartan 6 intègre des DSP [2] dédiés et optimisés pour réaliser ces opérations. Il s'agit de composants à l'architecture figée gravés sur le silicium dans le FPGA. Si une fonction numérique peut être implantée au choix à l'aide de CLB ou sur des DSP48, elle fonctionnera à des fréquences plus élevées sur DSP48.

Le Spartan 6 disposent de blocs DSP48A1 dont le nombre et l'organisation varient en fonction du FPGA utilisé. Le FPGA retenu en dispose de 58, organisés en deux rangs (de 30 et 28) qui permettent la propagation de retenues (par rang mais pas entre les rangs). La figure 3.2 présente l'architecture générale simplifiée de ce composant, dont le multiplieur 18×18 constitue le cœur.

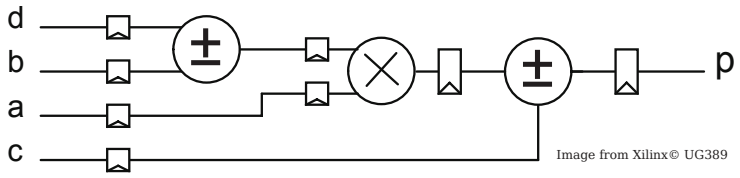


FIGURE 3.2 – Architecture simplifiée d'un bloc DSP48A1

3.1.3 Cas particulier d'additionneur

Un compteur incrémente une valeur courante de 1 en 1 à chaque front d'horloge, puis revient à 0 une fois la valeur maximum atteinte. Ceci peut être réalisé à l'aide d'un bloc plus général, un additionneur, dont la sortie est rebouclée sur une entrée, c'est à dire configuré en accumulateur, et l'autre entrée est fixée à 1. La figure 3.3 présente cette implantation.

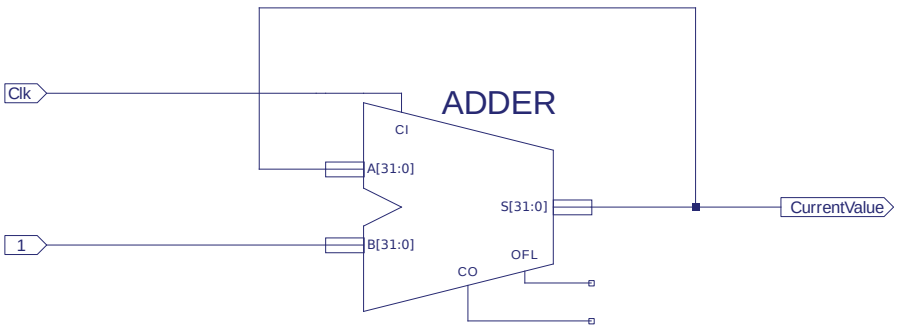


FIGURE 3.3 – Câblage d'un additionneur en compteur

Le périphérique que nous avons développé s'appuie un bloc encore plus général, le DSP48, qui inclut, comme montré plus haut, des multiplieurs. Ce choix relève davantage

2. DSP : Digital Signal Processor, processeur de signal numérique

de la facilité d'usage dans l'environnement de développement que d'une nécessité technologique, du fait de facilités de configurations. Précisons que ceci ne limite en rien la portabilité du composant développé. Lors du portage sur une autre plateforme que le FPGA considéré, il suffira de revenir à un accumulateur ou un compteur décrit directement en portes logiques élémentaires. Il sera a fortiori capable de fonctionner aux mêmes fréquences tout en consommant moins et pour une empreinte sur le silicium moindre.

3.1.4 Les registres de capture et les monostables

En fonctionnement normal, le compteur compte sans interruption. Lorsqu'un évènement se produit, comme le passage par 0 de la tension de sortie d'un signal d'un capteur, la valeur courante du compteur doit être mémorisée. Puisque le compteur ne s'arrête pas et que le processeur ne peut accéder instantanément à la valeur du compteur (le temps pour le processeur de mettre en suspens son opération courante puis d'accéder au périphérique à travers le bus), un registre de capture doit mémoriser la valeur du compteur au moment de l'évènement.

Dans le fonctionnement prévu, un comparateur de tension génère un signal carré à partir du signal sinusoïdal de sortie du capteur (sortie d'un accéléromètre, pilote ou détecteur d'un gyromètre). En vue de jouer leur rôle, les registres de capture ne doivent capturer la valeur du compteur qu'au moment de l'évènement et non en continu. Ainsi, la sortie du comparateur de tension ne peut pas commander directement ces registres.

Fréquence du signal	**Horloge**	**Nombre de périodes d'horloge**
50 kHz	100 MHz	2000
60 kHz	300 MHz	5000

TABLE 3.2 – Nombre de périodes d'horloge par période du signal à mesurer

Le tableau 3.2 présente des exemples du nombre de périodes d'horloge par période du signal à mesurer. En l'absence de dispositif adapté, cela signifie que pendant la moitié d'une période du signal à mesurer (celle qui correspond à l'état haut en sortie du comparateur), le registre de capture associé capture la valeur du compteur à chaque front d'horloge. Ceci s'avère parfaitement réalisable techniquement, mais pose d'autres problèmes, notamment sur le fonctionnement logiciel du système complet final (un processeur cadencé à 100 MHz ne va pas assez vite pour traiter ces données ; et un processeur capable de traiter ces données consommerait beaucoup plus d'énergie).

Des monostables non réenclenchables (*Non retriggerable monostable* en Anglais, NRM) pilotent donc les registres de capture. Ces dispositifs créent des impulsions lors d'un évènement et provoquent la mémorisation de la valeur courante du compteur par les registres de capture au moment de l'évènement. La figure 3.4 illustre le fonctionnement

de ces monostables, un positif (lorsque le signal d'entrée passe de 0 à 1) et un négatif (lorsque le signal d'entrée passe de 1 à 0). Tant que la sortie du comparateur ne redescend pas à 0, le monostable ne crée pas de nouvelle impulsion.

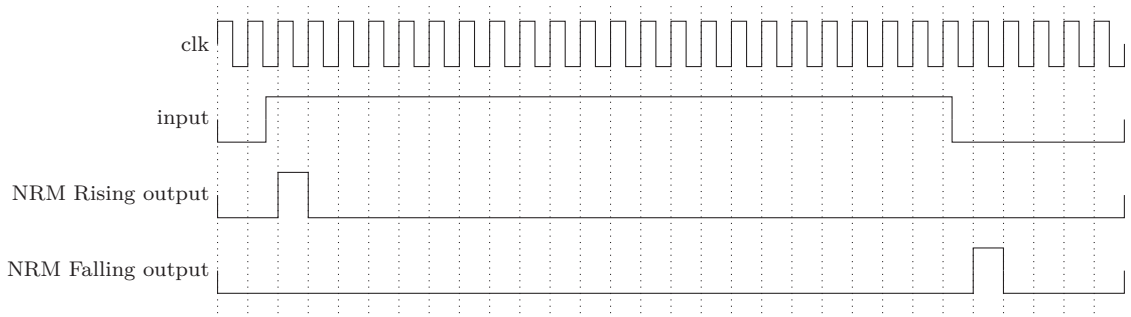


FIGURE 3.4 – Fonctionnement d'un monostable non réenclenchable

3.1.5 Présentation de l'architecture du périphérique

La figure 3.5 présente le périphérique de mesure de fréquence développé. Les entrées `Input N` permettent de signaler les moments où les registres `Capture N Rising edge` et `Capture N Falling edge` doivent capturer la valeur courante du compteur, respectivement sur les fronts montants et sur les fronts descendants des signaux d'entrées. Elles reçoivent les sorties des comparateurs mentionnés plus tôt.

L'implantation matérielle du compteur ne comporte que deux entrées, `Input 1` et `Input 2` sur la figure 3.5. Cependant, l'architecture peut supporter un nombre quelconque d'entrées.

La figure 3.5 affiche aussi la logique de bus, non détaillée plus haut. Elle se charge notamment de signaler au processeur qu'une nouvelle capture vient d'avoir lieu (par le biais d'interruptions). Elle contient également un registre de configuration afin de paramétrer les fonctionnalités du périphérique : activation ou désactivation du compteur, remise à zéro de la valeur courante...

Les fonctionnalités de cette logique de bus ne feront pas l'objet d'une explication.

3.2 Caractérisations

L'architecture du périphérique de comptage établie, il reste à implanter ce périphérique dans un FPGA et à déterminer sa fréquence de fonctionnement. Les performances des capteurs dictent le choix de cette fréquence, afin que l'électronique de mesure ne dégrade pas leur résolution intrinsèque.

Dans un montage différentiel typique tel que présenté à la section 1.1.3, l'accéléromètre présente les performances du tableau 3.3 qui conduisent à une erreur relative de $3{,}3 \times 10^{-10}$

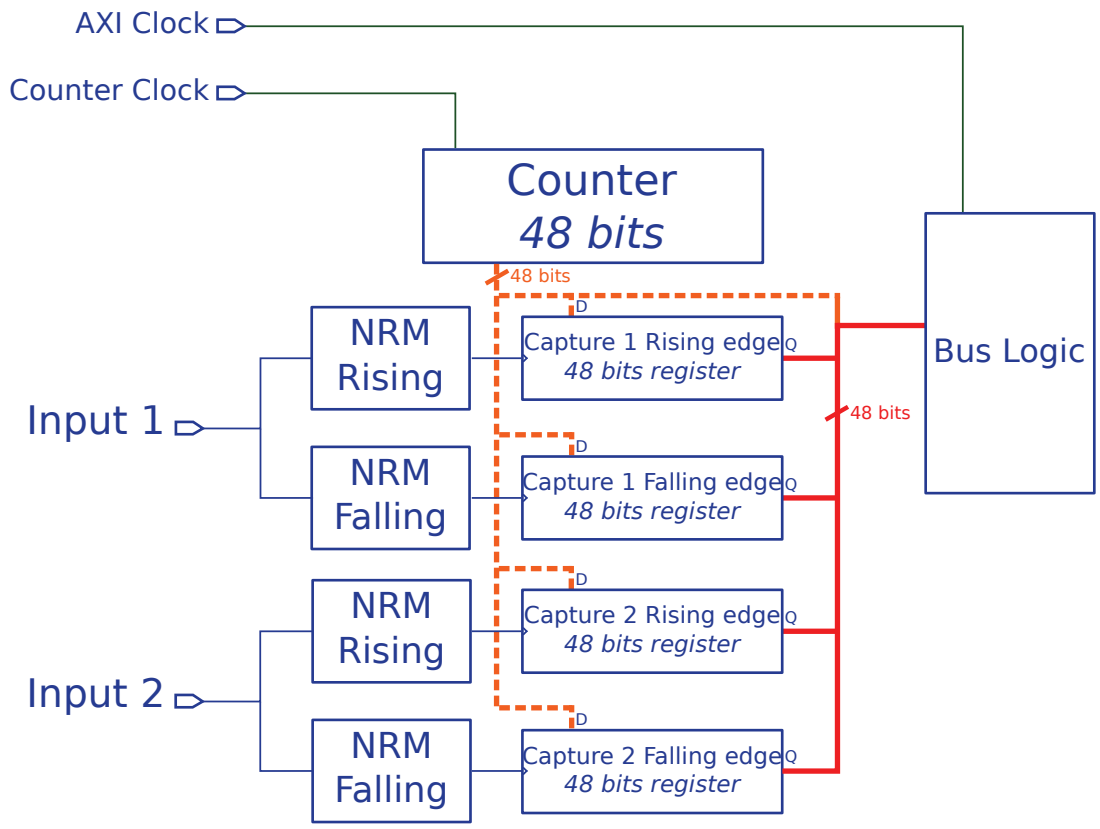


FIGURE 3.5 – Architecture du périphérique de comptage de fréquence développé

Facteur d'échelle	20 Hz/g
Résolution	1 µg ≡ 20 µHz
Temps d'intégration	10 s
Fréquence de résonance	≈ 60 kHz
Erreur relative	$\frac{20\,\mu\text{Hz}}{60\,\text{kHz}} = 3 \times 10^{-10}$

TABLE 3.3 – Performances d'un accéléromètre à cellules VIA (sources [LG10])

au bout de 10 s, soit une fréquence de comptage de[3] :

$$\frac{1}{3{,}3 \times 10^{-10}} \times \frac{1}{10} = 300\,\text{MHz}$$

Dans l'ensemble des expériences présentées ci-après et sauf mention contraire, le compteur compte donc à une fréquence de 300 MHz.

La fréquence de fonctionnement calculée, il reste à tester notre fréquencemètre avec un oscillateur ultra stable avant de le raccorder à un accéléromètre résonnant réel.

L'oscillateur ultra stable est en fait la sortie à 1 kHz d'un compteur précis SR-620 (Stanford Research Systems)[4] régulièrement étalonné (tous les 2 ans) par un laboratoire externe afin de garantir la précision de la mesure, autrement dit, la précision de son horloge de référence. En pratique, ce compteur permet de mesurer des fréquences jusqu'à 1,3 GHz avec une erreur relative de l'ordre de 1×10^{-10} sur 1 s [Sys].

Mesurer la fréquence de sortie du SR-620 avec le FPGA fournit donc à priori la résolution des mesures de fréquence effectuées par le FPGA, en supposant que la précision du FPGA est moins bonne que celle de la référence (ce qui semble assez probable). En effet, cette mesure révélera une erreur relative supérieure à celle du SR-620 (une variance d'Allan au dessus des 5×10^{-12} en 10 s du SR-620). Mesurer la fréquence de référence du SR-620 et tracer sa variance d'Allan à l'aide du compteur synthétisé sur le FPGA équivaut à déterminer la résolution de la mesure du compteur embarqué.

Cependant, cette information présente peu d'intérêt puisque la mesure de l'accélération se déduit par mesure différentielle de fréquence : l'erreur sur la valeur exacte mesurée n'a aucun impact puisque la mesure différentielle la compense.

Par la suite, afin de simuler un accéléromètre à sortie différentielle, un générateur Agilent 33522A fournit deux signaux en créneaux à environ 60 kHz. Les deux fréquences diffèrent toutefois très légèrement afin de reproduire l'écart de fréquence au repos qui peut exister entre deux cellules VIA, outre l'écart causé par une accélération (l'objet de la mesure). Les deux sorties du générateur sont donc raccordées aux entrées de capture du compteur du FPGA. Le code embarqué stocke les «dates» des fronts montants du signal mesuré et les transmet à un ordinateur par un port série. Une autre solution consiste à faire calculer directement la variance d'Allan par le processeur, puis à récupérer le résultat. Mais cette deuxième solution requiert davantage de développement logiciel, notamment pour écrire sur une mémoire de type flash (SD Card), afin de pouvoir transmettre les résultats et les données brutes. À ce stade du projet, ces développements n'étaient pas indispensables et la thèse s'est donc concentrée sur les problèmes plus fondamentaux.

3. Sur le FPGA retenu, seul un compteur basé sur un DSP48 (donc un compteur à additionneur) peut fonctionner à cette fréquence.

4. Ce compteur mesure des fréquences par le même moyen que le compteur en cours d'étude. En revanche, celui-ci n'est pas embarqué, pèse plusieurs kilos, occupe un espace assez important et consomme plusieurs watts.

Le volume de données intervient également dans le dimensionnement de la liaison série. Pour obtenir une variance aux temps courts (c'est-à-dire à haute fréquence), le système doit transmettre un maximum de dates. Dans ce cas, le maximum correspond à la fréquence mesurée, soit 1 kHz pour la mesure de référence avec le SR-620, puis 60 kHz pour la mesure avec les GBF. Ceci nécessite un débit d'au moins 28,8 Mbit s^{-1}, non atteignable par la liaison série utilisée[5]. Autre problème dans le cas de mesures longues (pour les temps longs sur la variance d'Allan), les fichiers deviennent extrêmement volumineux et compliqués à manipuler.

Par conséquent, le microcontrôleur réalise un prescaler programmable pour ne transmettre qu'une valeur sur P. La différence entre deux dates successives conduit alors à la période moyenne calculée sur P périodes. Pour les hautes fréquences, P prend une valeur assez basse (100 par exemple, pour une fréquence de 60 kHz) et les acquisitions durent une dizaine de minutes. Pour les basses fréquences, les mesures durent de quelques heures à plusieurs jours, avec un prescaler relativement élevé, de l'ordre de la fréquence à mesurer (ce qui fournit 1 point par seconde).

3.2.1 Variance de la mesure simple

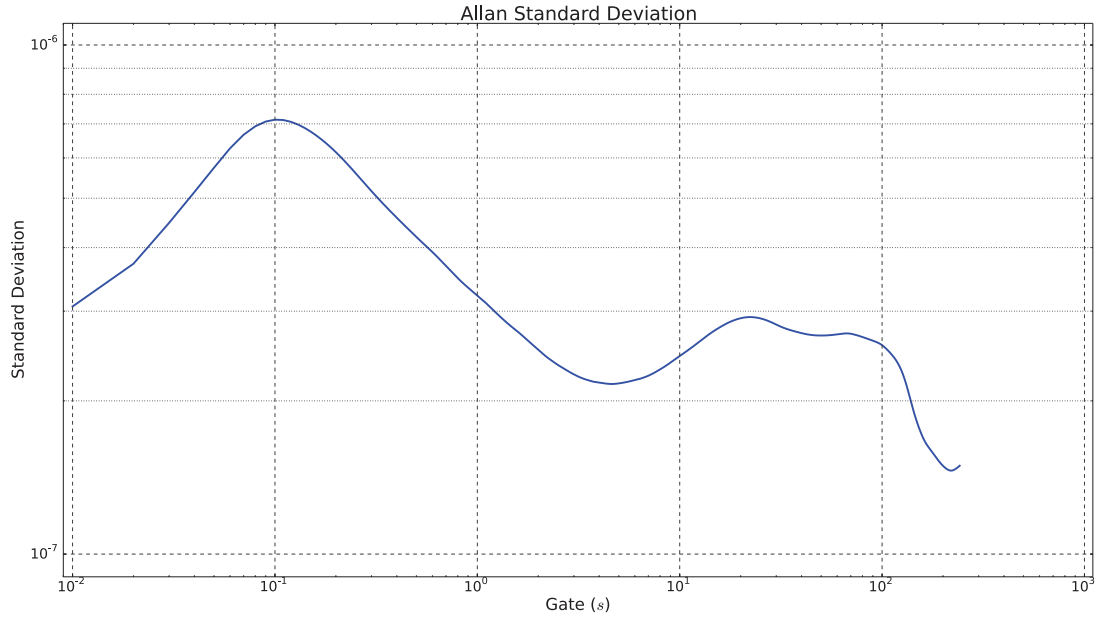


Figure 3.6 – Variance d'Allan obtenue lors de la mesure d'un signal à 1 kHz (normalisée par rapport à F_0)

La figure 3.6 montre la variance d'Allan obtenue lors de la mesure par le FPGA de

5. Il aurait été possible d'établir une connexion par liaison Fast Ethernet, beaucoup plus rapide (jusqu'à cent fois). Mais cela aurait demandé un travail supplémentaire assez fastidieux, aussi bien du côté FPGA (hardware et software) que de du PC, ce qui sort totalement du cadre de la thèse.

la sortie de fréquence de référence du SR-620. Cette mesure ne présente pas les caractéristiques attendues d'une variance d'Allan : un segment descendant révélant le moyennage du bruit blanc lorsque la durée d'intégration *tau* augmente, puis un palier limitant la résolution quelle que soit la durée d'intégration, dû au bruit Flicker.

Cette figure 3.6 présente le résultat d'une mesure sur une seule voie. Une autre mesure réalisée avec l'autre entrée de capture du compteur fournit la même courbe.

Le SR-620 étant un appareil de laboratoire de très haute résolution, il est plus vraissemblable d'attribuer la limitation de résolution au FPGA et en particulier la bosse observée. Si le problème se montre interne au FPGA, il restera encore à déterminer quel élément ou configuration en particulier est/sont à incriminer.

D'autres mesures effectuées avec un générateur de signaux Agilent 33522A confirment que le FPGA provoque les défauts observés. La figure 3.7 présente les variances d'Allan de ces deux signaux de sortie mesurées par le FPGA, l'un en bleu avec marqueurs, l'autre en vert (respectivement Output 1 et Output 2). L'allure rappelle celle de la figure 3.6, les courbes se superposent. Pour aller plus loin dans l'investigation et déterminer la source de bruit, la figure 3.8 montre la variance d'Allan de mesures simples (en vert et bleu) obtenues avec deux générateurs de signaux Agilent différents (même modèle, mais deux appareils distincts). Les horloges des deux mesures sont décorrélées, et pourtant les variances sont superposées. C'est donc que la source est commune et provient du FPGA lui-même. L'ensemble de ces expériences prouve ainsi que le FPGA cause la bosse observée et limite la précision de la mesure.

Puisque la perturbation affecte pareillement les deux entrées de capture, une mesure différentielle devrait réduire significativement voire annuler cette «bosse»et peut-être améliorer la résolution si des défauts identiques affectent les deux entrées de captures (comme par exemple un défaut sur l'horloge).

3.2.2 Variance de la mesure différentielle

La figure 3.8 présente la variance d'Allan d'une mesure différentielle (en rouge) des signaux verts et bleus (obtenus avec deux mêmes modèles de générateurs Agilent). En particulier, la mesure différentielle réduit efficacement les défauts communs aux deux mesures de variance : la «bosse» à 1×10^{-1} s disparaît et la mesure différentielle atteint bien l'objectif de résolution recherché de $3{,}3 \times 10^{-10}$ sur 10 s. Bien que le périphérique n'atteigne pas cette résolution en mesure simple, il répond tout de même aux attentes de performances pour un accéléromètre à deux cellules VIA et à mesure différentielle.

Enfin, cette mesure montre que le palier de résolution (plancher dû au bruit flicker) se situe à environ 3×10^{-10} et que les effets de marches aléatoires apparaissent dès une vingtaine de secondes de mesure (dus au FPGA ou aux générateurs, mais peu importe puisque cela se produit au delà des 10 s de mesure).

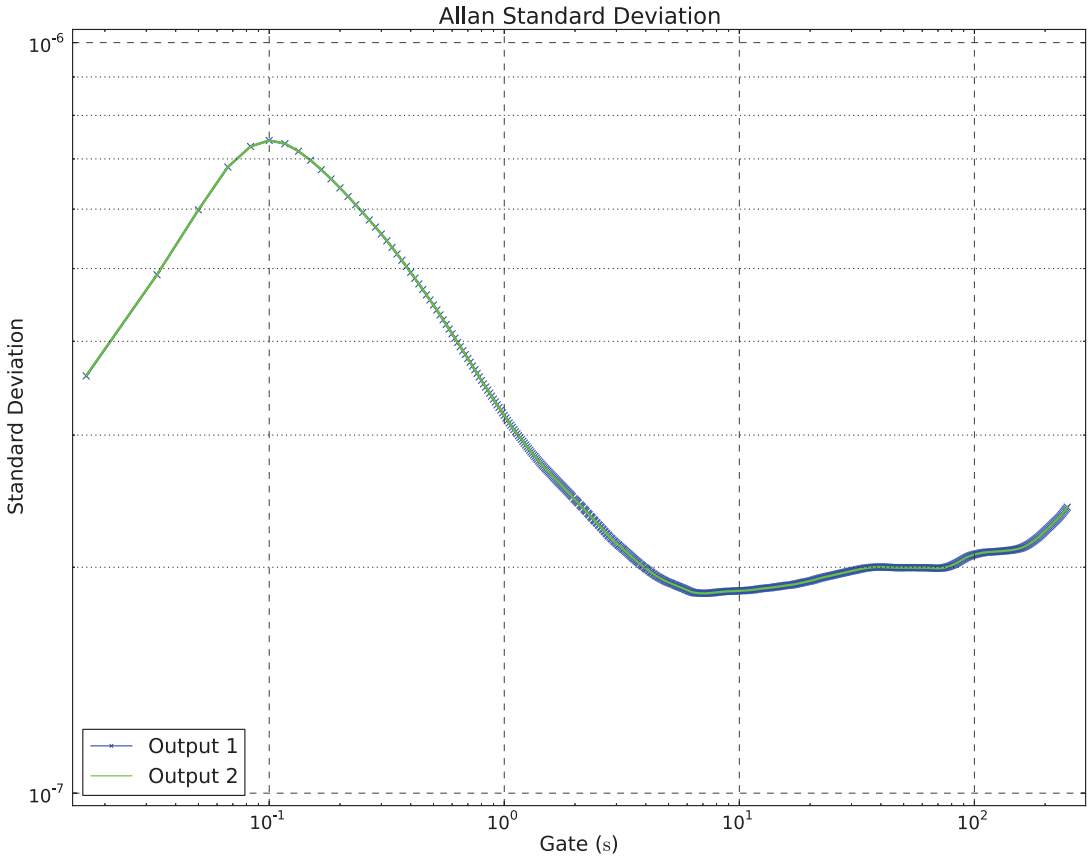


FIGURE 3.7 – Variance d'ALLAN d'un Agilent 33522A obtenue avec le FPGA pour une fréquence de comptage de 300 MHz

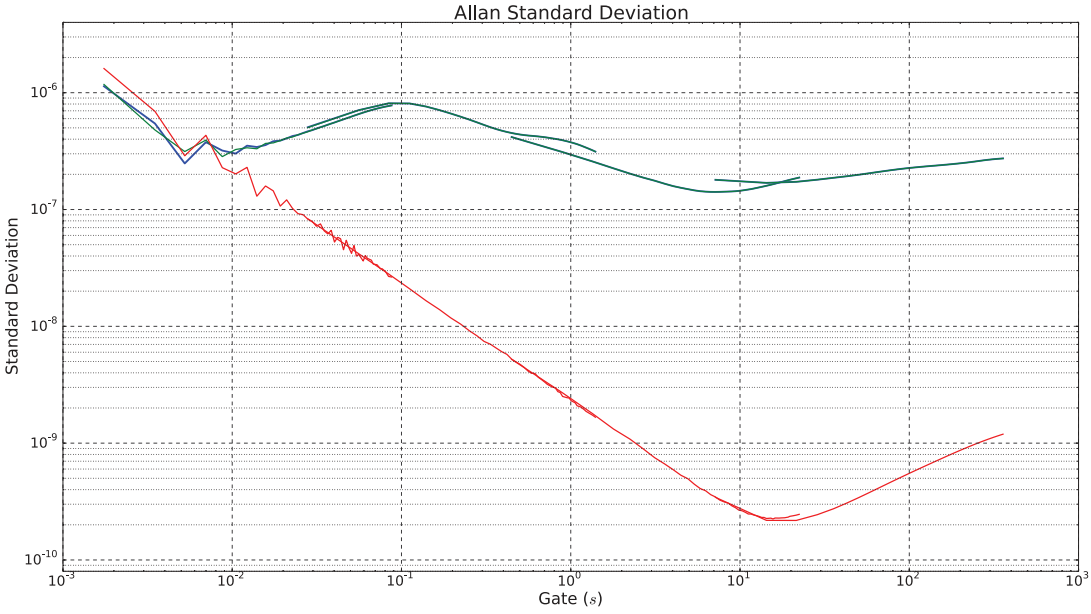


FIGURE 3.8 – Variance d'ALLAN d'une mesure différentielle en rouge, à partir des mesures simples en bleu et vert.

3.3 Analyse des résultats

Bien que la mesure différentielle atteigne l'objectif de résolution, il est essentiel de comprendre l'origine des défauts du périphérique, notamment à 10^{-1} secondes (soit 10 Hz). L'étude des données brutes donne accès à une mesure discrète de la fréquence des signaux mesurés :

$$f_k, k \in \mathbb{N}$$

Puisque le système transmet une mesure toutes les P périodes, f_k est en réalité la moyenne de la fréquence sur P périodes. Il s'avère donc intéressant d'étudier l'écart, en nombre de périodes d'horloge, des différences de dates relativement à la différence moyenne. Si δ_k représente la différence des dates entre la date k et la date $k-1$, alors la moyenne se définit par :

$$\delta_m = \langle \delta_k \rangle = \sum_{k=0}^{N} \frac{\delta_k}{N}$$

avec N le nombre de points de la mesure.

Le graphique 3.9 présente la valeur de la différence $\delta_k - \delta_m$ au cours du temps. En bleu figure les données de la capture 1, en rouge, celles de la capture 2 (les sorties des deux générateurs de fréquences Agilent). Globalement, sur P périodes, les périodes mesurées varient entre +15 et −15 périodes d'horloge autour de la valeur moyenne.

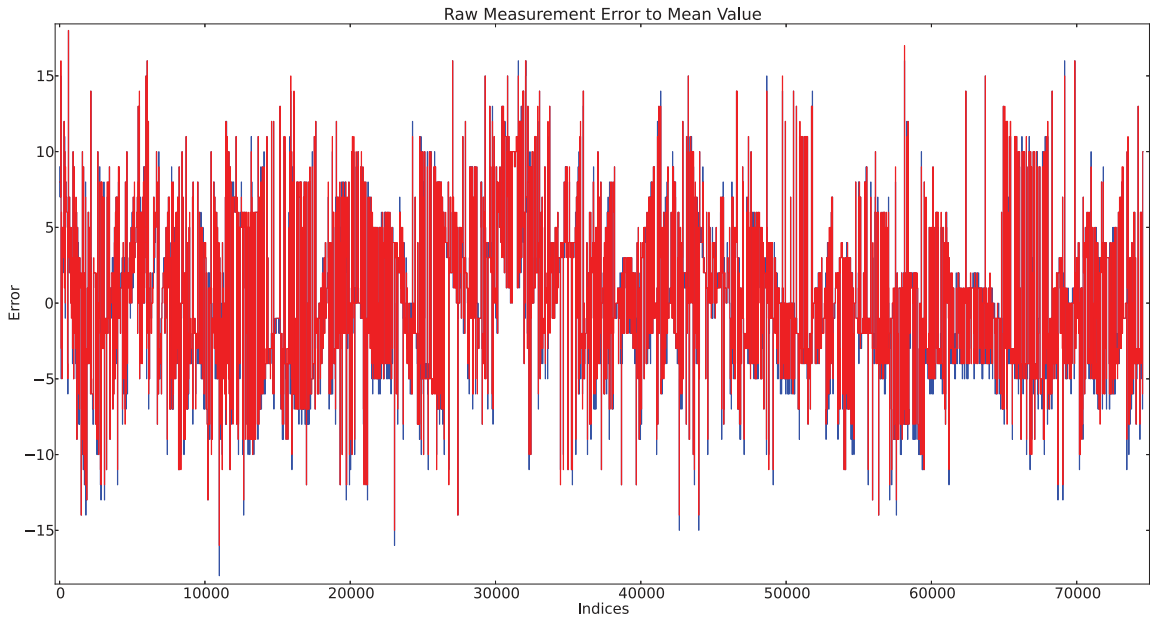


FIGURE 3.9 – Erreur des différences de date à la moyenne ($P = 1000$)

La figure 3.10a présente un zoom de la figure 3.9 sur 10 000 points environ. Il devient

possible de distinguer plusieurs types de sauts sur cette figure : des grands sauts (de 10 à 20 voire plus) et des petits sauts (de moins de 10, typiquement de 5). Les petits sauts proviennent d'une conception incomplète de la DDS sur FPGA. En effet, les registres de capture des fronts fonctionnent à deux vitesses : d'un côté à 300 MHz pour les captures, de l'autre à 50 MHz pour le bus AXI. Or des études menées après la thèse ont révélé qu'il n'était pas possible de mélanger ainsi des «domaines d'horloge». Il faut une double bufferisation, soit deux registres l'un derrière l'autre : l'un à 300 MHz uniquement et l'autre à 50 MHz uniquement.

Abstraction faite de ces petits sauts, les grands sauts ressemblent à une marche aléatoire dont le minimum se situerait vers le point d'abscisse 31 000 environ, compensée à la hausse ou à la baisse tous les 10 fronts d'horloge en vue de rester aussi près que possible de la valeur nulle. La figure 3.10b propose une reconstruction de cette marche aléatoire en supprimant les sauts. Tout se passe comme si une source extérieure venait perturber la fréquence de référence de la PLL qui pilote le compteur et que cette PLL agissait pour contrer cette influence. Les pointillés sur les figures 3.10a et 3.10b distinguent des domaines sur laquelle la tendance lente ne semble pas compensée par la PLL. Ce phénomène expliquerait pourquoi les deux signaux (bleu et rouge) semblent impactés de façon identique puisque l'horloge en sortie de PLL en serait la cause, horloge commune aux deux signaux.

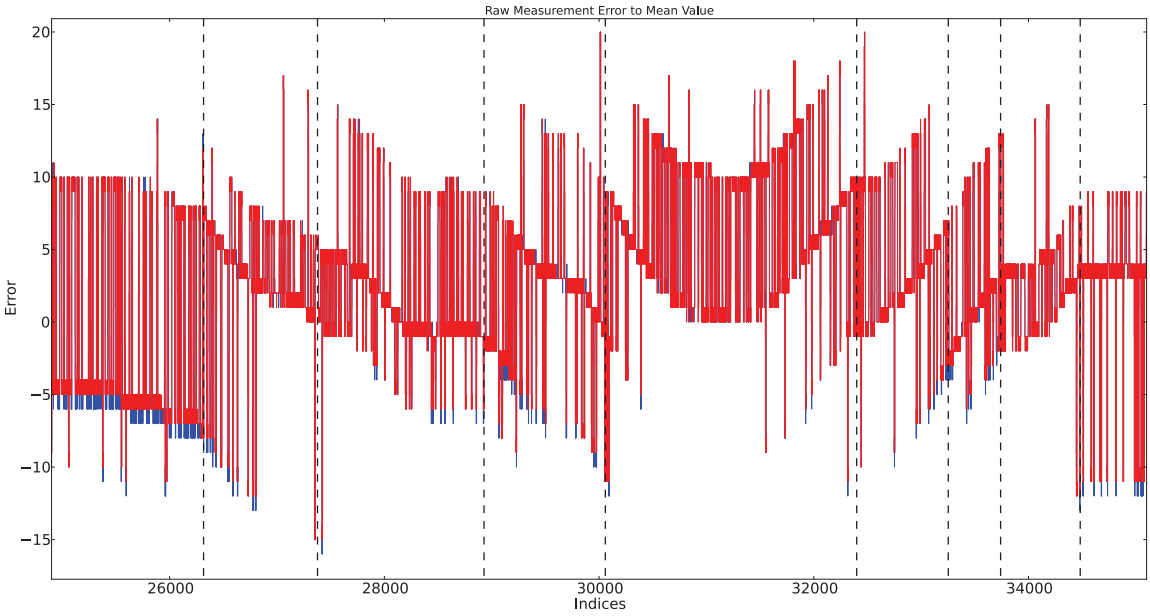
La figure 3.11 présente un nouveau zoom, avec un horizon encore plus restreint, quelques centaines de points seulement. Le comportement commun des deux captures, des deux signaux mesurés, apparaît clairement : les deux mesures présentent des sauts de même amplitude aux mêmes instants. La courbe rouge ne diffère jamais de la courbe bleue de plus d'une période d'horloge. Pour rappel, ces deux courbes proviennent de la mesure de deux signaux avec des fréquences très proches, autour de 60 kHz. Cela signifie que la différence des deux mesures tend vers une constante. Ces variations d'une période d'horloge uniquement sont normales et correspondent à la gigue naturelle de la méthode de mesure (section 2.1.2).

Autrement dit, puisque cette perturbation affecte les deux mesures, elle s'annule lors d'une mesure différentielle et la variance d'Allan tend vers la résolution intrinsèque du capteur. Voilà pourquoi sur les courbes de variances d'Allan, les mesures différentielles se montrent bien meilleures.

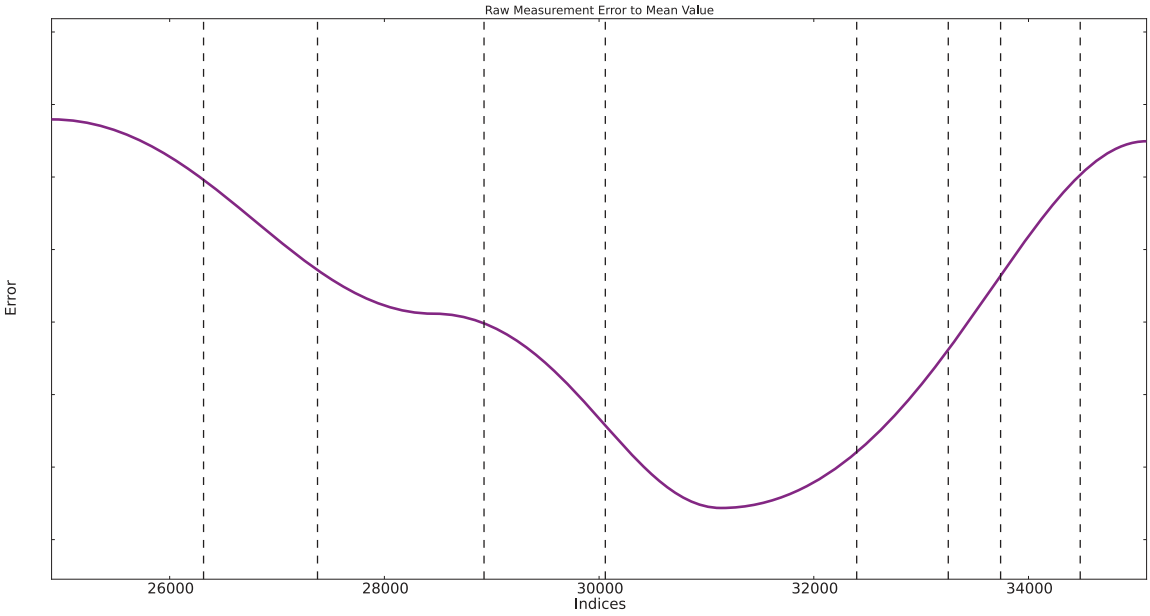
Le périphérique lui-même (le compteur) ou son horloge peuvent provoquer cette perturbation. Il reste donc à en identifier l'origine exacte.

3.3.1 Identification de l'origine de la perturbation

Une simple expérience permet de distinguer l'origine interne (le périphérique, c'est-à-dire le compteur lui-même) ou externe (l'horloge de référence) de cette imperfection :



(a) Erreur des différences de date à la moyenne sur 10 000 points environ ($P = 1000$)



(b) Reconstruction de la fluctuation lente décompensée

FIGURE 3.10 – Erreur des différences de date à la moyenne brute et extraction d'une fluctuation lente

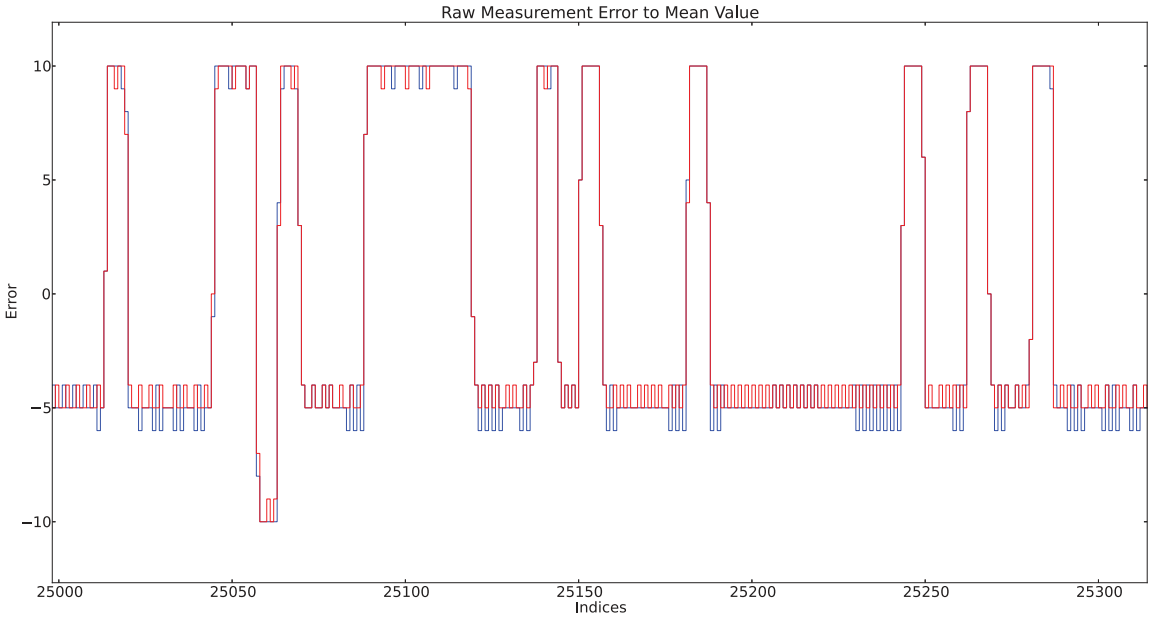


FIGURE 3.11 – Erreur des différences de date à la moyenne sur 300 points environ ($P = 1000$)

mesurer un sous multiple de l'horloge qui pilote le compteur.

Avec les notations de la figure 3.5, `Counter Clock` continue de fournir l'horloge de référence du compteur. Par ailleurs, une autre partie du FPGA reprend cette horloge et la divise pour fournir un signal de fréquence plus basse et de bruit de phase identique. Ce signal est extrait du FPGA puis re-connecté en externe successivement aux entrées `Input 1` et `Input 2`. La figure 3.12 présente un schéma cette expérience.

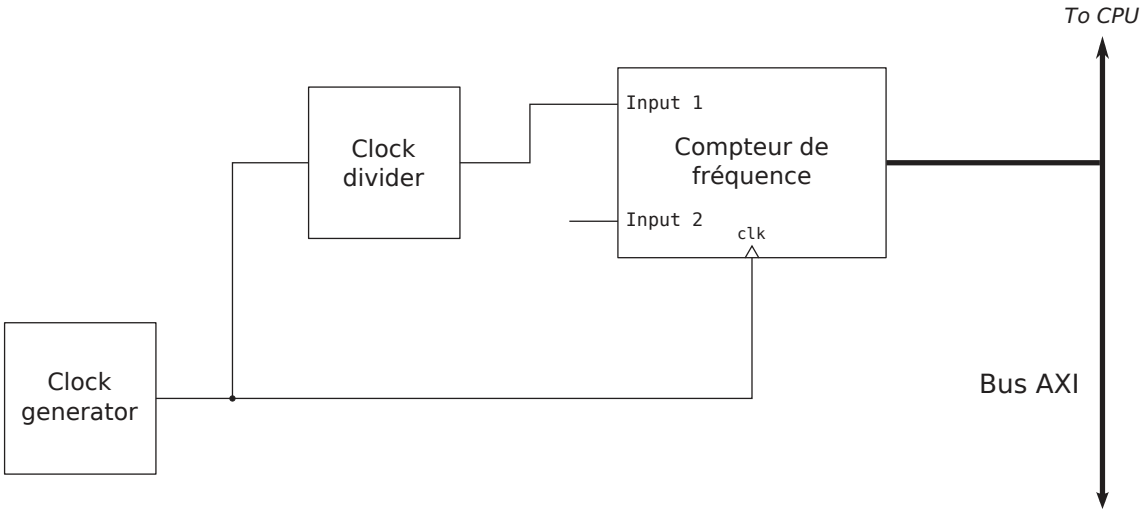


FIGURE 3.12 – Schéma de principe de l'expérience de discrimination de la source des imperfections

Le lecteur pourrait rétorquer que ce branchement s'avère inutile et que l'horloge divisée

pourrait rester à l'intérieur du FPGA. Mais dans ce cas, les outils de synthèse et de routage utiliseraient des composants spécifiques dans le FPGA qui modifieraient l'horloge : le signal observé serait différent de celui observé avec cette expérience. En outre, cela déplacerait les entrées et modifierait le périphérique : les entrées deviendraient internes et non plus externes. Modification mineure certes, mais modification tout de même.

Avec un branchement externe, les mesures de la fréquence du signal généré par le diviseur de fréquence par les deux entrées de capture permettent de discriminer l'origine du problème : si le défaut à 0,1 s (10 Hz) sur la variance d'Allan persiste, alors le périphérique lui-même est en cause. Si à l'inverse, il disparaît, cela signifie que l'horloge de référence est en cause puisque dans ce cas, la mesure annule la gigue car elle est identique pour la référence et le signal à mesurer.

Cette expérience vérifie également que le périphérique fonctionne correctement. En effet, les mesures doivent montrer un jitter de l'ordre de 1, c'est-à-dire le jitter théorique d'incertitude, puisque tout jitter sur l'horloge de référence `Counter Clock` se retrouve sur l'horloge divisée.

Finalement, cette expérience a fourni un résultat sans équivoque : la mesure de l'horloge divisée ne présente aucun jitter et demeure parfaitement stable dans le temps. Ceci se traduit par une variance d'Allan nulle qu'un tracé logarithmique ne peut représenter. Ce résultat apporte en outre la preuve que le périphérique fonctionne comme attendu et que l'horloge de référence qui cadence le compteur est à l'origine de la perturbation.

Il resterait à implanter la même architecture sur un autre Spartan 6 afin de vérifier si le défaut constaté se répète ou s'il est propre au composant de la carte de développement utilisé. Cette expérience telle quelle n'a pas eu lieu, mais l'évolution des FPGA a conduit à l'utilisation d'une nouvelle carte de développement basée sur un Artix 7, qui présente aussi une perturbation, mais à une fréquence légèrement supérieur.

3.3.2 Modification du mode de création des horloges

L'expérience décrite à la section précédente a mis en évidence la responsabilité de l'horloge dans la perturbation à 0,1 s sur les variances d'Allan des mesures.

Jusqu'à présent, l'étude n'a porté aucune attention particulière aux signaux d'horloge internes au FPGA. Les outils de développement se chargent automatiquement de générer les horloges nécessaires à partir d'un minimum d'information, à savoir :

— la fréquence voulue ;
— le déphasage souhaité ;
— le «groupe».

Les deux derniers paramètres s'avèrent optionnels. Et le «groupe» permet d'obtenir une série d'horloges à partir d'une même fréquence d'origine.

L'outil de synthèse utilise alors les ressources disponibles dans le FPGA pour créer les horloges demandées. Cependant, bien que les horloges fournies correspondent aux fréquences demandées, les outils n'exploitent pas au mieux les capacités du Spartan 6. Ils ne donnent pas accès aux blocs de création des horloges par exemple, et gèrent ces derniers automatiquement sans fournir aucun contrôle sur leurs paramètres, comme les diviseurs et multiplieurs de PLL. Pour contrôler finement ces paramètres et tirer le meilleur parti du FPGA, il faudrait développer un générateur d'horloge personnalisé.

D'une façon générale, les ressources d'horloge disponibles dans le Spartan 6 [Xilb], se distinguent en deux catégories :

— les composants chargés du routage des horloges ;

— les dispositifs de création/modification des horloges.

Les composants en charge du routage des horloges sont passifs, c'est-à-dire qu'ils ne modifient pas les caractéristiques (fréquence, déphasage) de l'horloge acheminée. Ils se distinguent des autres cellules du FPGA par leur sortance très élevée, ce qui autorise à raccorder leurs sorties à un grand nombre de portes. En outre, ils disposent d'emplacements et d'interconnexions spécialement étudiés au sein du FPGA, afin de garantir au mieux la propagation de l'horloge. Les outils de développement s'assurent d'ailleurs que le biais d'horloge, illustré sur la figure 3.13 (le délai d'arrivée d'un même front d'horloge sur deux éléments distincts du FPGA), demeure raisonnable[6].

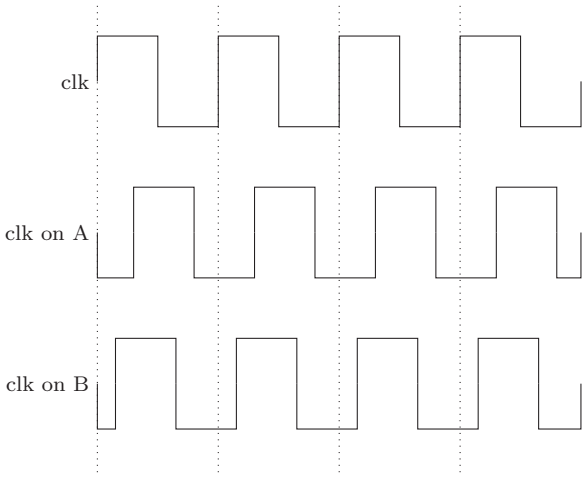


FIGURE 3.13 – Illustration du biais d'horloge, «clock skew» en Anglais : les temps d'arrivée sur les composants A et B diffèrent

La possibilité de transiter ou non par ces composants différencie un signal classique d'un signal d'horloge. Ceci signifie que lors de la création d'une horloge, comme par exemple pour la sortie d'une DDS, l'utilisateur reste libre de choisir ou non de typer le

6. Les outils permettent de fixer la tolérance à un pourcentage de la période d'horloge ou à un temps absolu.

signal comme horloge. Cependant, ne pas déclarer un signal comme horloge fournit parfois de meilleurs résultats, notamment en terme de déphasage, et facilite son emploi dans le FPGA. En effet, les outils de développement se montrent très contraignants vis-à-vis des horloges, ce qui oblige à des circonvolutions pour obtenir le résultat souhaité (utilisation de buffers de sortie de mémoire DDR2, cadencés par l'horloge à sortir du FPGA).

Les dispositifs actifs de création et/ou modification d'horloges se regroupent en blocs d'une PLL [7] pour deux DCM [8]. Le nombre de ces blocs varie ensuite en fonction du FPGA. Le Spartan 6 utilisé en contient 4.

L'utilisation des outils de développement et du générateur d'horloge automatique pour fournir les fréquences demandées conduit à l'emploi d'une PLL sur le FPGA. Mais, comme énoncé plus tôt, il n'y a aucun contrôle sur les paramètres du bloc PLL utilisé (voir [Xilb] pour plus de détails ; pas de contrôle sur les coefficients de la PLL, l'ordre de chainage des blocs PLL et DCM, etc. . .). La conception d'un générateur d'horloge personnalisé donne accès à tous les paramètres de ces blocs, leur enchaînement etc. . . . Notamment, l'emploi de DCM *et* de PLL peut peut-être supprimer cette bosse à 10^{-1} s sur la variance d'Allan.

En outre, tout bloc de gestion d'horloge doit être alimenté par une horloge de référence externe, qui rentre dans le FPGA par des broches spécifiques afin de suivre un chemin dédié ainsi qu'expliqué plus tôt. Dans la carte de développement utilisée, le nombre de sources possibles est de trois :

— un oscillateur à quartz soudé sur la carte avec une fréquence de 200 MHz, connecté à l'entrée principale/standard du FPGA ;
— un socle pour un oscillateur à quartz secondaire ;
— une entrée pour un signal différentiel sur connecteurs SMA.

Le développement d'un générateur d'horloges personnalisé permet aussi de déterminer si cette perturbation provient de la gestion des horloges par le FPGA (DCM et PLL), ou de l'horloge externe utilisée, ce que l'expérience présentée ne permet pas. En effet, si, malgré les expérimentations avec les PLL et DCM des blocs de gestion des horloges (par exemple, n'utiliser qu'un DCM, ou qu'une PLL, ou bien une PLL suivie d'un DCM. . .), la perturbation ne disparaît pas, alors seul le quartz peut encore expliquer le phénomène. Changer d'horloge de référence incrimine ou, au contraire, discrimine ce quartz.

Ce générateur d'horloge développé (en remplacement du générateur implanté automatiquement par les outils de développement) ouvre donc la voie à l'étude et possiblement à

7. PLL : Phased Locked Loop
8. DCM : Digital Clock Manager. Il s'agit d'un élément numérique capable de multiplier ou diviser une horloge de référence, de modifier son rapport cyclique, de créer des déphasages ou encore de supprimer le biais d'horloge (clock skew). Il emploie un DFS (Digital Frequency Synthesiser) pour ses opérations de multiplication/division d'horloge et une DLL (Delay-Locked Loop) pour le déphasage. Une DLL constitue le pendant numérique d'une PLL où une chaîne à retard remplace le VCO.

la suppression de la perturbation observée. En plus d'améliorer l'horloge de référence du compteur, il améliore aussi celle de la DDS.

Avant d'en venir à la mise en œuvre de la DDS, l'attention va se porter sur l'estimation et la correction du déphasage à l'aide du compteur étudié à l'instant.

3.4 Estimation du déphasage

Une fois en boucle fermée, la DDS excite les résonateurs et une DPLL les maintient à leur fréquence de résonance. Avant de concevoir cette boucle à verrouillage de phase (logicielle ou implantée sur une électronique dédiée), il faut mesurer le déphasage entre l'excitation et la réponse du capteur. L'algorithme asservira ensuite ce déphasage à la valeur appropriée.

À la résonance, le déphasage entre l'excitation et la réponse du capteur vaut $\frac{\pi}{2}$, entre potentiel appliqué (force d'excitation) et charges collectées (déplacements dans le matériau).

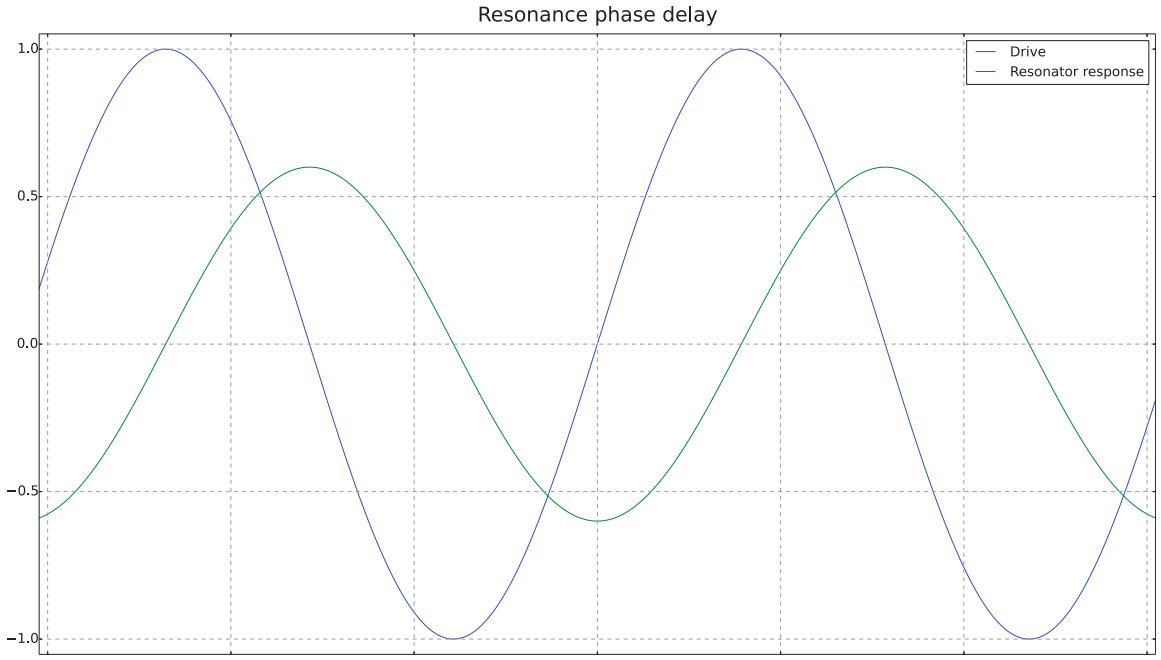


FIGURE 3.14 – Représentation du déphasage entre le signal d'excitation et la réponse du capteur

3.4.1 Référence de phase du signal pilote du résonateur

La section 1.3.2 a montré que le compteur développé pour la mesure de fréquence détecte la phase nulle du signal mesuré. En vue de réaliser une boucle à verrouillage de

Notation	Signification
T	Période vraie du signal d'excitation (en nombre de fronts d'horloge du compteur)
k	Instant, numéro du front d'horloge sur le compteur
C_k	Valeur de la capture (date) du signal d'excitation à l'instant k
R_k	Valeur de la capture (date) de la réponse du capteur à l'instant k
T_k	Période instantanée du signal d'excitation ($T_k = C_k - C_{k-1}$)
n	Nombre de points pour calculer l'estimateur de T
$\hat{T}_k$	Estimateur de T à l'instant k avec n valeurs

TABLE 3.4 – Signification des variables employées dans ce chapitre

phase numérique, il ne manque plus qu'une mesure de phase du signal d'excitation pour calculer le déphasage.

Le passage par la phase nulle de ce signal se déduit très facilement de l'accumulateur de phase de la DDS : son MSB passe de 1 à 0.

3.4.2 Mesure du déphasage à l'aide du compteur seul

Si la période du signal d'excitation vaut T (en nombre de fronts d'horloge du compteur), et si la prochaine date de passage du signal d'excitation à l'instant k (k-ième front d'horloge sur le compteur) vaut C_k, alors celle de la réponse du capteur doit avoir lieu en $R_k = C_k + \frac{T}{4}$. Maintenir le capteur à sa fréquence de résonance consiste donc à maintenir un intervalle de $\frac{T}{4}$ entre les dates de passages des deux signaux.

Dans un premier temps, une estimation de T est nécessaire. La moyenne des dernières différences date à date des captures T_j ($j \leqslant k$) forme un estimateur évident de T à l'instant k. Soit, avec n le nombre de captures sur lesquelles effectuer la moyenne :

$$\hat{T}_k = \frac{\sum_{j=k-n+1}^{j=k} T_j}{n} \tag{3.1}$$

Se pose alors la question du nombre de points sur lequel effectuer le calcul. L'estimation de T s'améliore avec le nombre de points de mesure. Toutefois, un calcul de $\hat{T}_k$ avec trop de points réduit la bande passante de la mesure inertielle (la moyenne est un filtre passe-bas).

Bien que $\hat{T}_k$ soit défini pour tout k, le calcul de cet estimateur ne s'effectue pas nécessairement à chaque instant k. En revanche, le calculer à chaque instant (calcul par moyenne glissante) permet à l'algorithme de verrouillage de phase d'adapter plus rapidement la commande.

Les ressources disponibles dans le système posent aussi certaines contraintes sur ce calcul. Tout d'abord, le processeur opère en général avec des opérandes de 32 bits. Cela ne signifie pas qu'il ne peut pas manipuler des mots de longueurs supérieures à 32 bits,

mais il faut plus de temps. Il est donc plus judicieux de stocker des différences date à date et non les dates complètes de captures.

En effet, bien que les dates de capture soient sur 48 bits, le capteur le plus lent, le VIG, oscille à environ 20 kHz : avec un compteur à 300 MHz, les T_k sont de l'ordre de 15 000. Un codage sur 16 bits suffirait donc à stocker ces valeurs en mémoire mais demanderait des manipulations supplémentaires (masquage et décalage pour chaque valeur) par rapport au stockage d'un entier de 32 bits, utilisable directement. Dans ce cas, l'utilisation d'entiers de 32 bits représente une économie mémoire de $\frac{1}{3}$ par rapport aux mots de 48 bits.

Une autre approche incite tout de même à stocker les captures sur 48 bits directement car le calcul de $\hat{T}_k$ peut se simplifier comme suit (voir le tableau 3.4 pour la signification des notations) :

$$\begin{aligned} \hat{T}_k &= \frac{\sum_{j=k-n+1}^{k} T_j}{n} \\ \hat{T}_k &= \frac{\sum_{j=k-n+1}^{k} C_j - C_{j-1}}{n} \\ \hat{T}_k &= \frac{C_k - C_{k-n+1}}{n} \end{aligned} \tag{3.2}$$

Bien que le MicroBlaze manipule plus difficilement ces entiers sur 48 bits, ce calcul de $\hat{T}_k$ ne requiert plus qu'une soustraction et une division, sauf lorsque le compteur revient à 0 puisque C_k est alors plus grand que C_{k+1}. Une simple comparaison des MSB des nombres à soustraire permet de traiter séparément ce cas de remise à 0 du compteur.

En comparaison, le calcul selon l'équation (3.1) requiert n additions et 1 division. En revanche, conserver en mémoire des nombres de 48 bits occupe deux fois plus de mémoire puisque le système stocke deux mots de 32 bits.

Un MicroBlaze avec un maximum de 64 KiB peut ainsi stocker 8192 ($64\,\text{KiB} \div (2 \times 4)$) valeurs en utilisant des entiers de 48 bits (codés en réalité sur 64 bits). En fonctionnement réel, une partie de la mémoire disponible sert à l'exécution du programme. Le nombre exact de valeurs dépend du programme et de la taille mémoire qu'il occupe mais sera inférieur à 8192.

Toutefois, il faut également raisonner en terme d'itération. En effet, le processeur calcule l'estimation $\hat{T}_k$ à chaque nouvelle capture ou à intervalle régulier de captures. Or, seules quelques valeurs changent entre deux calculs successifs. Deux seulement si le calcul est fait à chaque instant k : la nouvelle période instantanée T_k contribue au nouvel estimateur alors que la période instantanée la plus ancienne T_{k-n} disparaît. Par conséquent, si le système conserve en mémoire la dernière estimation, il suffit d'ajouter la nouvelle période et retrancher la plus ancienne (toutes deux divisées par n). L'estimation

$\hat{T_k}$ l'instant k est donc :

$$\hat{T_k} = \hat{T_{k-1}} + \frac{T_k}{n} - \frac{T_{k-n}}{n}$$

Cela complique quelque peu le programme mais permet de ne stocker que les périodes instantanées, sur 32 bits, et non les captures complètes. Cette approche semble le meilleur compromis entre performance et utilisation mémoire.

L'équation précédente nécessite toujours une division. Or, de par ses capacités mémoires assez petites, le processeur ne pourra pas stocker énormément de périodes instantanées T_k. Il n'y a donc pas de risque de dépassement (d'overflow) du produit $n\hat{T_k}$. Par conséquent, plutôt que de calculer et travailler avec l'estimateur $\hat{T_k}$, le système mémorise le produit $n\hat{T_k}$. À chaque nouvelle capture, il est alors modifié comme suit :

$$n\hat{T_k} = n\hat{T_{k-1}} + T_k - T_{k-n} \tag{3.3}$$

En revanche, cette méthode et les limitations de la mémoire ne permettent pas d'effectuer ce calcul sur un grand nombre de points. Une mémoire plus grande seule ne suffit pas : le produit $n\hat{T_k}$ dépasse 32 bits au bout d'un peu plus de 14 s. Si la mémoire le permet (il faut stocker tous les T_k), stocker le produit $n\hat{T_k}$ sur 64 bits supprime ce problème de dépassement.

Si la mémoire est trop petite pour stocker suffisamment de T_k pour calculer un estimateur sur un temps long, une solution intermédiaire consiste à stocker une date 1 C_k à intervalles réguliers et à utiliser l'équation (3.2) : par exemple, le système peut stocker 1 C_k toutes les 5 s.

Une fois l'estimation de la période connue, l'annexe D.1 présente un algorithme possible pour fournir l'erreur de phase de la réponse du capteur. Cet algorithme commence par s'assurer que l'écart entre excitation et réponse du capteur ne dépasse pas 1 sur les 32 bits de poids forts. Si c'est le cas, une erreur de phase maximale est retournée (la valeur est choisie de telle sorte qu'elle n'entraîne pas une réaction trop forte du correcteur ensuite). Sinon, il calcule le produit $n\hat{T_k}$ et prédit la date de la réponse. Enfin, il calcule la différence entre cette prédiction et la réalité et l'erreur de phase retournée est n fois supérieure à la valeur réelle (pour éviter des divisions).

Dans le cas standard, c'est-à-dire sans remise à 0 du compteur ou des LSB de la réponse du capteur, le processeur requiert 27 instructions par exécution de la fonction. Les durées d'exécution de certaines instructions varient en fonction des options de configuration du MicroBlaze. De même, la profondeur du pipeline passe de 3 à 5 (voir annexe E.2) en fonction de ces options. Au final, environ 35 périodes d'horloge sont nécessaires pour exécuter la fonction dans le pire cas.

Dans le cas extrême, soit un VIA à 65 kHz, le logiciel appelle cette fonction 65 000 fois par seconde, soit une puissance de calcul nécessaire de 2, 28 MIPS, puissance tout à fait atteignable par le MicroBlaze (103DMIPS à 100 MHz pour rappel).

Dans le cas d'un autre processeur, une puissance de 2, 5 DMIPS doit suffire à exécuter la fonction.

3.4.3 Utilisation des données de la DDS en boucle fermée

Les possibilités de mesure du déphasage présentées à la section précédente ne reposent que sur les données issues du compteur. Or, cet algorithme fonctionne en boucle fermée et alimente la DDS qui génère le signal pilote et dispose directement de l'information de phase de l'accumulateur de phase. Ainsi, en utilisant les signaux d'entrées des compteurs, il est possible de lire la valeur de la phase ϕ_k à l'instant k dans la DDS au lieu de capturer la valeur courante du compteur. Cela demande une nouvelle modification mineure de l'architecture de la DDS pour fournir la phase courante.

Si $\phi_{E,k}$ est la phase courante de la DDS du dernier passage par 0 du signal d'excitation et $\phi_{R,k}$ celle du signal de sortie du capteur, le déphasage $\Delta\phi_k$ à l'instant k est alors fourni par :

$$\Delta\phi_k = \phi_{R,k} - \phi_{E,k} \tag{3.4}$$

Une fois encore, la moyenne de cette valeur instantanée fournit un estimateur du déphasage, soit :

$$\Delta\hat{\phi}_k = \frac{\sum_{j=k-n+1}^{j=k} \Delta\phi_j}{n} \tag{3.5}$$

Il s'agit du pendant de l'équation (3.1) et la même optimisation de calcul s'applique :

$$n\hat{\phi}_k = n\Delta\hat{\phi}_{k-1} + \Delta\phi_k - \Delta\phi_{k-n}$$

Cela ne change pas la puissance de calcul requise et fournit directement le déphasage, plus naturel qu'un décalage temporel, pour alimenter le correcteur de phase qui fonctionne en aval dans l'architecture d'une boucle à verrouillage de phase.

Le déphasage estimé, le système modifie la fréquence de sortie de la DDS pour maintenir le capteur en résonance grâce à l'algorithme de la boucle à verrouillage de phase présenté dans la section suivante.

3.5 Présentation d'un algorithme de correction du déphasage

Le correcteur le plus simple n'exploite que le signe de l'erreur de phase. Il fonctionne comme suit :

— pour une erreur positive (le passage à 0 de la réponse du capteur arrive trop tard), le correcteur incrémente le mot de commande de la DDS de 1 ;
— en cas d'erreur négative, il décrémente le mot de 1 ;
— enfin pour une erreur nulle, la commande reste inchangée.

Un démonstrateur préliminaire fonctionnait ainsi, avec un calcul d'erreur sur 32 valeurs. Il reposait sur un ARM Cortex M3, un compteur à 100 MHz intégré et une puce DDS externe AD9837. La DDS utilisait un accumulateur de phase 32 bits et une horloge à 5 MHz, conduisant à un pas de fréquence de $\approx 1{,}2$ mHz.

Bien que rudimentaire, cet algorithme permettait de maintenir un VIG proche de sa fréquence de résonance. En effet, la fréquence d'oscillation variait de quelques Hertz autour de la fréquence de résonance et le verrouillage n'était pas stable. En outre, si la fréquence de départ était trop loin de la fréquence de résonance, le système passait la fréquence de résonance sans l'accrocher, en raison du facteur de qualité élevé des capteurs. L'algorithme utilisé ne prenait pas en compte ce retard et il aurait fallu modifier le calcul du signal d'erreur pour donner davantage d'importance aux valeurs anciennes. Une autre solution consiste à augmenter le nombre de valeurs stockées en mémoire pour déterminer l'erreur.

La solution pour corriger ce problème consiste à utiliser un correcteur proportionnel intégral dérivé (PID). L'équation (3.6) présente la formule d'un tel correcteur, avec ε_k l'erreur calculée par la fonction présentée à la section 3.4 à l'instant k. Ce correcteur fournit la variation à appliquer à la commande de la DDS (elle est toujours supérieure d'un facteur n à la vraie valeur).

$$\begin{aligned} u_k &= u_{p,k} + u_{i,k} + u_{d,k} \\ u_k &= K_p \times \varepsilon_k + K_i \sum_{l=0}^{k} \varepsilon_l + K_d \left(\varepsilon_k - \varepsilon_{k-1}\right) \end{aligned} \tag{3.6}$$

Dans ce correcteur, les corrections $u_{p,k}$, $u_{i,k}$ et $u_{d,k}$ peuvent être limitées par l'algorithme, notamment l'intégrateur, afin de ne pas causer de variations trop violentes, au lancement de la boucle par exemple. C'est le cas entre les lignes 28 et 33 du cadre suivant qui présente un code pour un tel correcteur.

Notation	Signification
ε_k	Erreur calculée par le système à l'instant k
K_p	Facteur de correction proportionnelle
K_d	Facteur de correction dérivée
K_i	Facteur de correction intégrale

TABLE 3.5 – Notation de l'équation la correction PID

```
// Correction factors.
#define Kp                XX
#define Ki                XX
/* Kd must divide the derivative term by 2, i.e. if Kd is to be 1, then
* it should be set to 1/2 here.
*/
#define Kd                XX

#define integral_max    XX
#define integral_min    XX

int32_t pid_corr (const int32_t phase_difference_error) {
// Derivative value
static int32_t derivative = 0;
// Intergral value
static int32_t integral = 0;
// Correction
float PID = 0;

// Add the proportional and derivative terms.
PID = Kp * phase_difference_error +\
Kd * (phase_difference_error -\
derivative);
// Remember the current error for the next derivative.
derivative = phase_difference_error;
// Add up the integral error if within given range.
integral = integral + phase_difference_error;
if (integral > integral_max) {
    integral = integral_max;
}
// Check minimum integral value.
if (integral_min > integral) {}
    integral = integral_min;
}
// Return the correction value.
PID += Ki * integral;
return PID;
}
```

Un tel correcteur conduit à une boucle à verrouillage de phase stable et rapide à corriger les fluctuations.

Cette fonction logicielle peut être facilement modifiée pour utiliser des facteurs de correction K_p, K_i et K_d variables, ou des extrema variables pour le correcteur intégral (`integral_min` et `integral_max`) ou des deux autres corrections. Au contraire, une réalisation matérielle ne peut apporter la même souplesse qu'au prix d'une complexité accrue et d'une occupation plus importante sur le FPGA.

La fonction proposée ici prévoit de travailler avec des entiers afin d'éviter le recours à une FPU. Si besoin, les facteurs K_x peuvent être multipliés par une valeur entière suffisamment grande, 1000 par exemple, par rapport à leur valeur décimale réelle. La valeur retournée `PID` le sera aussi.

Si la valeur de la correction intégrale ne dépasse pas les extrema `integral_min` et `integral_max`, le code de cette fonction nécessite 15 instructions (avec les chargements et sauvegardes en mémoire pour `derivative` et `integral`). Avec un VIA à 65 kHz, cela correspond à 1 MIPS pour maintenir le verrouillage.

Au final, la boucle à verrouillage de phase, qui exécute la fonction ci-dessus et le calcul du déphasage présenté à la section (3.4), utilise environ $3,5$ DMIPS par capteur.

Au total, pour exécuter le calcul de fréquence par régression linéaire (section 2.2) et réaliser la boucle à verrouillage de phase, le système doit offrir une puissance de calcul de $4,2$ DMIPS par capteur VIA. Un VIG demande moins de puissance en raison de sa fréquence de résonance plus basse. À cela s'ajoutent quelques instructions supplémentaires pour appeler les fonctions, gérer leur retour, traiter les captures sur le compteur et ajuster la DDS.

Les fonctions de mesure de fréquence, de calcul du déphasage et de sa correction abordées, il reste à mettre en place la dernière fonction numérique : la DDS.

3.6 Conception d'un synthétiseur numérique direct

La figure 1.15 a présenté l'architecture retenue pour l'électronique numérique des capteurs. Le compteur réalisé, il reste à implanter le synthétiseur numérique de fréquence.

3.6.1 Présentation de l'architecture du périphérique

Compte-tenu de la structure d'une DDS présentée sur la figure 2.8 et sans conversion Σ/Δ présentée à la section 2.3.6, le composant à concevoir dispose donc de trois blocs :

— un accumulateur de phase ;
— une mémoire pour stocker la forme du signal désiré ;
— la logique de bus pour communiquer avec le processeur.

Lors de l'utilisation en compteur de fréquence, le câblage adéquat de l'additionneur d'un bloc DSP48 permettait d'obtenir un compteur (figure 3.3). Pour obtenir un accumulateur, il suffit de remplacer l'entrée statique à 1 par la valeur à accumuler, pilotée par la logique de bus et donc, le processeur, comme le montre la figure 3.15. Cette valeur est l'incrément de phase à chaque période d'horloge.

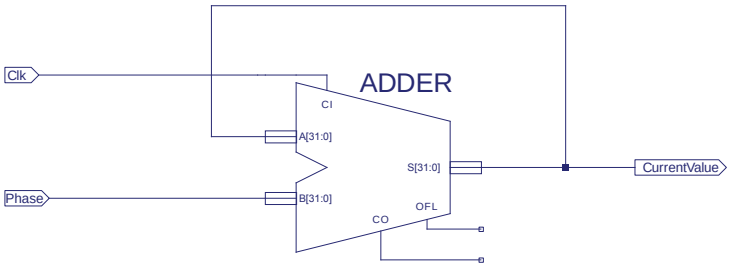


FIGURE 3.15 – Câblage d'un additionneur en accumulateur

Encore une fois, les mêmes remarques à propos de l'emploi du DSP48 et de la portabilité du périphérique s'appliquent. Par ailleurs, lors du portage des deux périphériques conçus, il peut s'avérer intéressant de décrire un composant additionneur générique qui servira pour le compteur de fréquence et pour l'accumulateur de phase de la DDS.

En vue de compléter la DDS, il ne manque plus qu'une LUT, c'est-à-dire une mémoire.

3.6.2 Introduction aux mémoires dans le FPGA

La section 3.1.1 a déjà expliqué qu'une *slice* pouvait jouer le rôle de bloc mémoire de 256 bits. Leur association crée des mémoires plus grandes parfaites pour une utilisation locale ou le stockage de petits volumes de données. Cependant, cette mémoire distribuée ne convient pas réellement lorsqu'il s'agit de concevoir des espaces de stockage plus volumineux pour lesquels il existe des blocs adaptés : les blocs RAM, BRAM en abrégé.

Ces blocs se répartissent régulièrement dans tous le FPGA et stockent chacun 18 KiB dont seuls 16 KiB stockent véritablement des données, les 2 derniers stockant des bits de parité si l'utilisateur le souhaite. La figure 3.16 montre comment ces blocs se répartissent dans le FPGA par rapport aux autres éléments déjà mentionnés (les horloges passent au travers des buffers BUFG et BUFH, deux des composants en charge du routage des horloges mentionnés à la section 3.3.2).

Ces blocs mémoire peuvent s'utiliser soit entier, soit en deux blocs de 9 KiB (8 de données plus 1 de parité). La dimension des données peut varier de 1 à 32 bits[9] par puissance de 2. Bien entendu, il demeure toujours possible de n'utiliser qu'une partie

9. L'utilisation en deux blocs de 9 KiB limite les fonctionnalités du bloc. Mais cette étude ne se voit pas concerner par ces limitations.

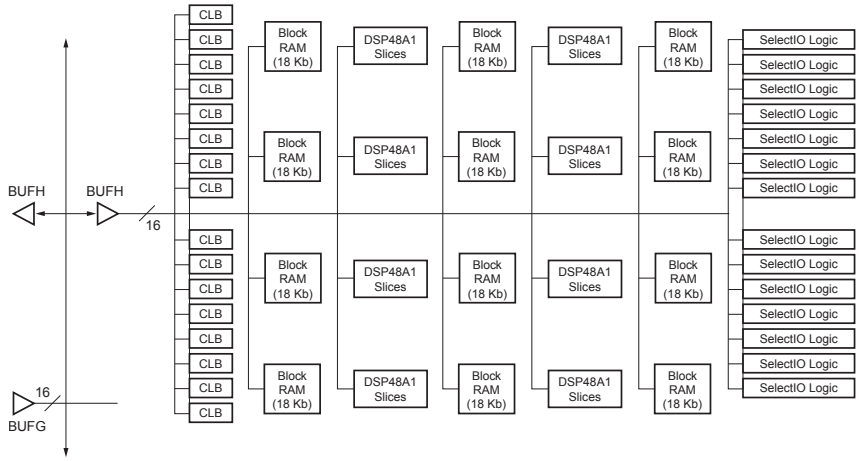


FIGURE 3.16 – Architecture simplifiée d'une rangée du FPGA. Image tirée de [Xila]

seulement des bits. Par exemple, pour des données sur 10 bits, l'utilisateur configurera le bloc pour fournir une donnée sur 16 bits, mais seuls les 10 bits de poids faibles (ou forts) porteront une information, les autres restant à 0.

Puisque les blocs stockent 2×8 ou 16 KiB de données utiles, le nombre d'emplacements mémoire et donc, le nombre de bits d'adresse, varie. Par exemple, pour une sortie sur 8 bits, il y aura 2048 emplacements mémoire dans un bloc de 16 KiB, soit 11 bits d'adresse, puisque $2048 \times 8 = 16$ KiB. Le tableau 3.6 reprend quelques paires de dimensions possibles pour les entrées/sorties de ces BRAM (les plus pertinentes pour la réalisation de la DDS). Un BRAM utilisé en deux blocs de 8 KiB dispose naturellement de deux fois moins de mots (deux fois moins d'adresses disponibles).

Nombre de mots	Largeur des mots
16384	1
8192	2
4096	4
2048	8
1024	16
512	32

TABLE 3.6 – Nombre d'adresses et tailles des données stockées dans un BRAM de 16 KiB

Évidemment, il est tout à fait possible de grouper plusieurs BRAM pour stocker plus de données, ce qui signifie augmenter le nombre de bits d'adresse et concevoir une logique de décodage d'adresse pour adresser le bon BRAM. Ainsi, avec un bit d'adresse supplémentaire il devient possible d'adresser 2 blocs RAM, le bit de poids fort servant à identifier sur quel bloc router le reste des bits d'adresse.

Le FPGA retenu pour cette thèse dispose de 116 blocs RAM pour un total de 1856 KiB. Le processeur utilise une partie de ces mémoires (l'espace alloué et donc le nombre de blocs

utilisés par le processeur est réglé à l'aide des outils de développement) mais il en reste suffisamment pour former une LUT, tant que le convertisseur ne dépasse pas 14 bits (il faut alors $2^{(14+2)} \times 14 \div 16384 = 56$ blocs RAM). Ceci dans le cas d'une LUT qui stockerait la totalité de la sinusoïde, soit sans optimisation aucune.

Présentation des techniques de réduction de la LUT

La section 2.3.5 a montré qu'une LUT et un accumulateur de phase de mêmes tailles conduisent à un volume de stockage irréaliste et a étudié l'impact de la troncature de phase engendrée. Toutefois, elle n'a montré aucune optimisation pour la LUT.

En effet, une sinusoïde présente beaucoup de symétries (figure 3.17). Tout d'abord, une symétrie par rapport à l'axe des abscisses (dans sa représentation habituelle entre -1 et 1). Puis, en ne considérant que la partie positive par exemple, une symétrie centrale, c'est-à-dire en 0 pour un cosinus et $\frac{\pi}{2}$ pour un sinus. Au final, un quart de sinusoïde suffit à reproduire une onde complète, soit un gain possible de 2 bits d'adresse ($2^2 = 4$) sur la LUT.

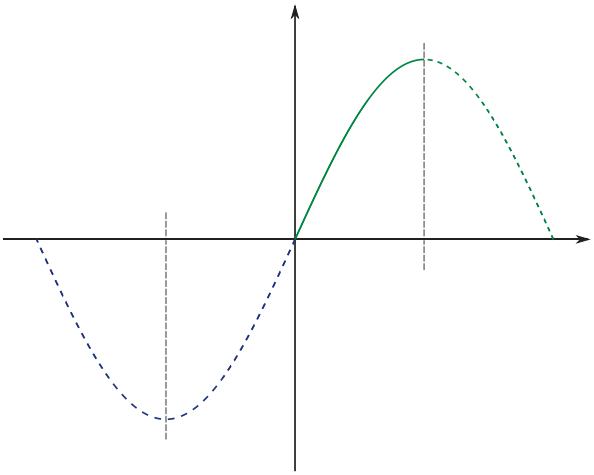


FIGURE 3.17 – Symétries dans une sinusoïde

Si une LUT stocke un quart de sinusoïde, entre $-\pi$ et $-\frac{\pi}{2}$ par exemple, alors le parcours de ses adresses en ordre inverse fournit la sinusoïde entre entre $-\frac{\pi}{2}$ et 0. Le bit de poids fort de la sortie de l'accumulateur de phase permet ensuite de situer la phase avant ou après 0 et donc de reconstituer la totalité de la sinusoïde.

Concrètement, avec N le nombre de bits de la sortie de l'accumulateur de phase et L le nombre de bits d'adresse de la LUT (comme introduits à la section 2.3), si la LUT contient les valeurs pour le quadrant de phase entre 0 et $\frac{\pi}{2}$, les bits $N-2$ à $N-L$ de la sortie de l'accumulateur de phase permettent d'adresser la LUT. Lorsque le bit $N-1$ vaut 0, le système parcourt la LUT dans le sens des adresses croissantes. Lorsqu'il vaut 1, le périphérique opère un complément à 1 de l'adresse (une négation bit à bit). Enfin,

lorsque le bit en position N vaut 1, le système opère un complément à 1 sur la sortie de la LUT. La figure 3.18 présente cette implantation.

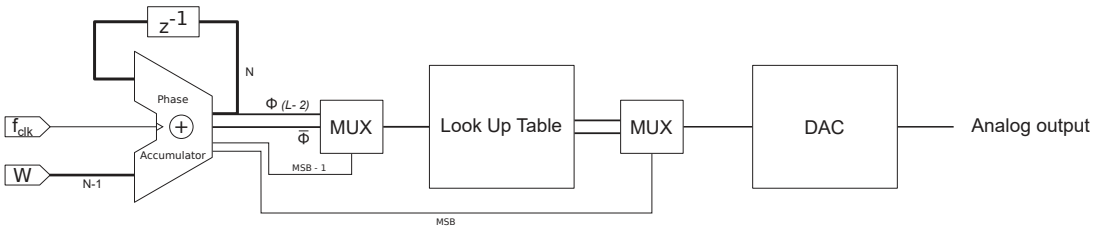


FIGURE 3.18 – DDS avec table demie-onde et utilisation du MSB de l'accumulateur de phase pour former le MSB du DAC

Pour fonctionner, cette technique nécessite cependant que les valeurs stockées dans la LUT présentent un déphasage d'un demi LSB [10], afin d'éviter que les sorties en 0 et π ne diffèrent lors des compléments à 1. Sans cela, l'amplitude en 0 et en $\frac{\pi}{2}$ disposeraient de la même adresse sur la LUT : il faudrait alors un bit supplémentaire pour les distinguer [Van97] [NSK88]. Cette solution réduirait la taille de la mémoire mais pas la taille du bus d'adresse, ce qui limite donc son intérêt.

En outre, un convertisseur numérique-analogique fournit un nombre pair de niveaux de tension de sortie. Par exemple, un convertisseur 10 bits alimenté en 0-1,8 V fournira 1024 tensions en sortie, par pas de $\approx 1{,}76\,\text{mV}$ [11]. Un convertisseur numérique analogique ne peut fournir la valeur médiane. Le convertisseur exemple considéré pourra délivrer une sortie à 899 mV et la suivante à 901 mV, mais jamais à 900 mV exactement. Le déphasage supplémentaire d'un demi LSB expliqué au paragraphe précédent évite ainsi le besoin d'une sortie à la valeur médiane exactement du convertisseur et ne provoque aucune rupture de symétrie et donc aucune dégradation supplémentaire sur le spectre de sortie final obtenu.

Cependant, la section 2.3.6 a expliqué les défauts des DAC réels et a évoqué le fait que la LUT pouvait corriger certains défauts, notamment les erreurs de monotonie si celles-ci correspondaient à des inversions de bits. Or ceci ne peut s'appliquer que si la LUT stocke les valeurs à corriger du DAC, autrement dit une table demie-onde qui stocke les valeurs de la fonction sinus entre $-\frac{\pi}{2}$ et $\frac{\pi}{2}$.

En outre, cela autorise aussi à programmer des formes d'onde un peu plus compliquées en vue de corriger des défauts sur les capteurs par exemple.

Il s'agit là d'un compromis qui dépend des contraintes : une table demi-onde pour corriger les défauts du DAC ou une table quart d'onde pour réduire la mémoire du périphérique final.

10. Si la LUT virtuelle dispose de 4 bits, et donc la LUT réelle de 2 bits, alors un demi LSB représentent une phase de $\frac{2\pi}{2^4} = \frac{\pi}{8}$.
11. Le pas s'obtient par $\frac{1{,}8}{1023}$ car la valeur numérique maximale est 1023.

Pour optimiser davantage le périphérique, le MSB de la sortie de l'accumulateur de phase peut commander le MSB du DAC (cela facilite le routage de ce bit). En effet, pour la fonction sinus, tant que le MSB de la phase reste à 0, l'amplitude reste positive, ce qui signifie que les valeurs sur le DAC doivent rester dans la partie supérieure, soit avec le MSB à 1. Par conséquent, la négation du MSB de la phase fournit directement le MSB du DAC et il devient inutile de le stocker dans la LUT.

Toutefois, l'utilisation de BRAM pour les LUT dans le Spartan 6 rend cette remarque non pertinente. En effet, les dimensions des entrées/sorties de ces mémoires ne peuvent prendre qu'un nombre fini de valeurs ainsi qu'énoncé plus tôt. Par exemple, dans le cas d'un convertisseur 12 bits, il faudra configurer un BRAM pour une sortie sur 16 bits. Aucun intérêt donc à rajouter des éléments logiques alors que la mémoire stocke même trop de données.

En revanche, cette optimisation reste valable pour un portage du périphérique sur une autre cible (un autre FPGA ou un ASIC).

L'utilisation d'un seul bloc RAM présente un intérêt par sa mise en œuvre relativement simple (pas de logique de décodage d'adresse pour les BRAM ni pour le MSB du DAC) mais limite la taille du convertisseur à 9 bits dans le cas d'une table demi onde (10 bits pour un table quart d'onde).

Dans ce cas, rajouter 1 bit d'adresse nécessite une logique relativement simple et autorise alors l'utilisation d'un DAC 10 bits pour une table demi onde (ou 11 bits pour une table quart d'onde).

Dernière remarque importante : en vue d'opérer la démodulation nécessaire à la mesure du gyromètre, le système a besoin de connaître la phase courante. Le périphérique DDS donne accès en lecture à la phase courante en sortie de l'accumulateur de phase : le processeur peut donc y accéder et la traiter.

Il pourrait s'avérer utile aussi de sortir cette donnée du périphérique afin qu'une autre partie du FPGA puisse accéder à cette phase et opérer la démodulation nécessaire sans intervention du processeur. Cette solution présenterait l'avantage de ne pas surcharger le processeur et fonctionnerait probablement plus rapidement qu'un traitement logiciel, mais requiert toutefois la création d'un autre périphérique dédié à la démodulation.

3.6.3 Dimensionnement de la DDS

Ces subtilités matérielles abordées, il reste à dimensionner effectivement la DDS. La section 2.3 a présenté les paramètres d'une DDS ainsi que leur influence sur le spectre de sortie.

La conception du système sur le FPGA fixe un premier paramètre : la fréquence de fonctionnement f_{clk}, fréquence du bus AXI cadencé à 50 MHz. Pour un accéléromètre

VIA, cela fournit un taux de suréchantillonnage de 400 environ, soit plus de 800 points dans la sinusoïde de sortie.

Il reste cependant possible de réduire cette fréquence à un autre sous-multiple de 600 MHz [12], tant que le spectre ne comporte pas d'harmoniques de niveaux élevés.

Le pas de fréquence souhaité fixe N, nombre de bits de l'accumulateur de phase. Des capteurs considérés dans cette thèse, le VIA se révèle le plus exigeant en matière de résolution fréquentielle. En effet, le VIG ne nécessite pas une excitation à sa résonance exacte puisque la mesure s'obtient par démodulation d'amplitude. Un positionnement précis facilite tout de même la démodulation en offrant un signal de mesure d'amplitude plus élevé. En revanche, la mesure du VIA se déduit de sa fréquence de résonance. Le système d'excitation doit donc s'ajuster au mieux afin de ne pas détériorer la résolution. Le tableau 3.7 rappelle les caractéristiques du VIA, présenté a la section 1.1.3.

Caractéristique	**VIA**
Résolution	1 µg
Temps d'intégration pour résolution maximale	10 s
Facteur d'échelle	10 Hz/g

TABLE 3.7 – Caractéristiques d'une cellule VIA

Pour une résolution de 1 µg, le pas de fréquence doit être inférieur à 10 µHz. Un pas δf de 1 µHz conviendra à ajuster au mieux la fréquence de sortie de la DDS à la fréquence de résonance du capteur. Soit, d'après l'équation (2.19), avec une largeur N de l'accumulateur de phase donnée par :

$$\begin{aligned} N &\geqslant \log_2(f_{clk}) - \log_2(\delta f) \\ N &\geqslant \log_2(50 \times 10^6) - \log_2(10 \times 10^{-6}) \\ N &\geqslant 42,2 \end{aligned} \tag{3.7}$$

Compte-tenu de l'accumulateur de phase basé sur un DSP48, l'accumulateur de phase fonctionnera sur 48 bits, soit un pas de fréquence de 0,18 µHz.

32 bits ne suffisent pas à fournir la résolution souhaitée mais les processeurs embarqués, optimisés en consommation, taille... fonctionnent aujourd'hui couramment sur 32 bits. Tous les échanges entre les périphériques se font alors sur des bus de 32 bits. Programmer l'accumulateur de phase demande donc 2 écritures sur la DDS afin de transmettre les bits du mot de commande, ici 47 bits ($48-1$ pour respecter le théorème de SHANNON) avec un accumulateur de phase basé sur un DSP48, mais au moins 42 bits ($43-1$) pour d'autres implantations.

12. La PLL dans le FPGA fournit une horloge maîtresse à cette fréquence.

L'optimisation de la programmation du mot de commande W de l'accumulateur de phase est donc importante. Un premier accès expose directement les bits de poids faibles et bits de poids fort avec un chevauchement des 16 bits intermédiaires : les 16 bits de poids fort des 32 bits de poids faible et les 16 bits de poids faible des 32 bits de poids fort. Ainsi dans certains cas, une seule écriture peut suffire, soit des bits de poids fort, soit des bits de poids faible.

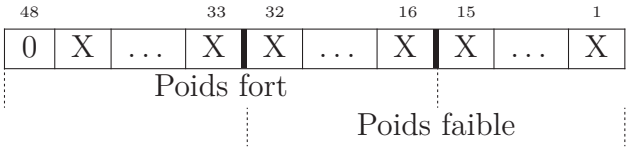


FIGURE 3.19 – Chevauchement des 16 bits intermédiaires pour la programmation du mot de commande

Une autre amélioration consiste à ajuster le mot de commande courant par addition d'un mot de 32 bits signés (puisqu'il faut aussi pouvoir retrancher une valeur). Ainsi, la plupart du temps, une seule écriture suffit à ajuster la fréquence de sortie. Cependant, ce système nécessite de rajouter un additionneur (signé pour ajouter des nombres négatifs) à la DDS. Il n'y a aucune difficulté en soit à cette solution, mais elle implique une consommation électrique légèrement accrue du périphérique puisqu'elle utilise quelques portes supplémentaires (même inactives, il existe toujours un petit courant de fuite).

Quant au VIG, avec une largeur de bande de résonance d'environ 20 mHz, un pas de fréquence de 0,18 µHz convient également.

Enfin, il reste à dimensionner la LUT à l'aide de l'équation (2.35). Le gyromètre ne présente aucune restriction particulière à ce sujet, contrairement à l'accéléromètre dont la précision de la mesure peut être affectée par un contenu spectral impropre. En effet, la mesure gyrométrique s'effectue par démodulation à des instants calculés à partir de la sortie de l'accumulateur de phase (à chaque passage de la phase en 0 et en π, plus ou moins une tolérance car la phase ne revient pas forcément à 0 exactement à chaque bouclage de l'accumulateur), donc en amont de la LUT.

La section 1.1.3 a fourni les informations nécessaires au calcul de la taille de la LUT, avec r l'erreur relative à atteindre et m le nombre de périodes du signal nécessaires pour obtenir l'erreur relative souhaitée (ici au bout de 10 s) :

$$r = 3{,}3 \times 10^{-10}$$
$$m = 60 \times 10^3 \times 10$$

La taille de la LUT vérifie donc :

$$L \geqslant -\log_2(3{,}3 \times 10^{-10} \times \pi \times 60 \times 10^4)$$
$$L \geqslant 10,6$$
$$L = 11$$

Avec une réduction à 9 bits réels dans le cas du stockage d'une LUT quart d'onde, ou 10 pour une demi onde.

L'équation (2.22) fournit la résolution du DAC :

$$D = L - 2$$
$$D = 9$$

Le tableau 3.8 reprend l'ensemble des paramètres de conception de la DDS adaptée aux caractéristiques des capteurs.

Paramètre	**Valeur**
f_{clk}	50 MHz
N	48
L	11 *(9 ou 10 réels)*
D	9

TABLE 3.8 – Paramètre de la DDS à concevoir pour satisfaire les performances des capteurs

3.7 Caractérisation du synthétiseur

La poursuite des travaux entamés au cours de la thèse a conduit à l'implantation d'un synthétiseur de fréquence fonctionnel sur FPGA. Du fait des évolutions technologiques, le FPGA est différent mais toutes les particularités mentionnées plus tôt, notamment sur les blocs mémoires, restent valables. Les travaux ont donc continué sur Artix 7, une famille FPGA gravée en 28 nm (contre 45 pour le Spartan 6), et ont apporté quelques modifications au dimensionnement mentionné plus haut.

3.7.1 Architecture et paramètres du synthétiseur de fréquence implanté

Ainsi que l'a montré l'équation (3.7), un accumulateur de phase sur 32 bits ne fournit pas un pas de fréquence suffisamment fin pour exploiter les capteurs considérés, alors qu'il

s'agit d'une longueur standard pour les entiers en programmation embarquée, facilitant ainsi leur manipulation dans le code.

L'utilisation de 48 bits facilite la conception matérielle de l'accumulateur de phase en s'appuyant sur un DSP48. D'un point de vue logiciel, cela nécessite cependant de coder soigneusement toutes les opérations effectuées sur les nombres destinés à piloter l'accumulateur de phase.

Au contraire, un accumulateur de phase sur 64 bits fournit un pas de fréquence bien assez fin (2,7 pHz avec une horloge à 50 MHz), sur-dimensionné par rapport aux dérives de l'horloge (c'est-à-dire que le pas de fréquence de l'accumulateur est petit devant les fluctuations de l'horloge) ou la jigue de la fréquence de résonance du capteur. En outre, il s'agit d'un type logiciel natif et beaucoup de processeurs 32 bits disposent d'instructions pour réaliser des opérations simples avec des entiers de 64 bits (addition, multiplication...), simplifiant ainsi leur utilisation.

Pour ces raisons, l'architecture mise en œuvre utilise un accumulateur de phase 64 bits programmé sur les cellules de base du FPGA et non plus sur un DSP48. Cela présente un double intérêt : d'une part, faciliter la gestion logicielle de l'accumulateur de phase, et d'autre part, porter l'accumulateur de phase sur n'importe quel FPGA ainsi qu'expliqué à la section 3.1.3.

Pour des raisons de simplicité, la LUT dispose de 10 bits d'adresse, correspondant aux blocs mémoires disponibles dans le FPGA, de 1024 adresses (soit 10 bits) contenant chacune 16 bits. La LUT contient une table pleine onde et ne suit pas l'équation (2.22). Avec une table quart d'onde et la mise en place de la logique de parcours et d'inversion de la table, elle est équivalente à une LUT à 12 bits d'adresse. Il s'agit donc ici d'une première version non optimisée mais simple et rapide à implanter.

Enfin, la conversion numérique-analogique se fait au moyen d'un DAC Σ/Δ à sortie différentielle. La paire binaire complémentaire de sortie alterne à haute cadence les valeurs 0/1 ou 1/0 pour réaliser la forme d'onde décrite par la table sur 16 bits.

Le tableau 3.9 reprend les paramètres et leurs valeurs.

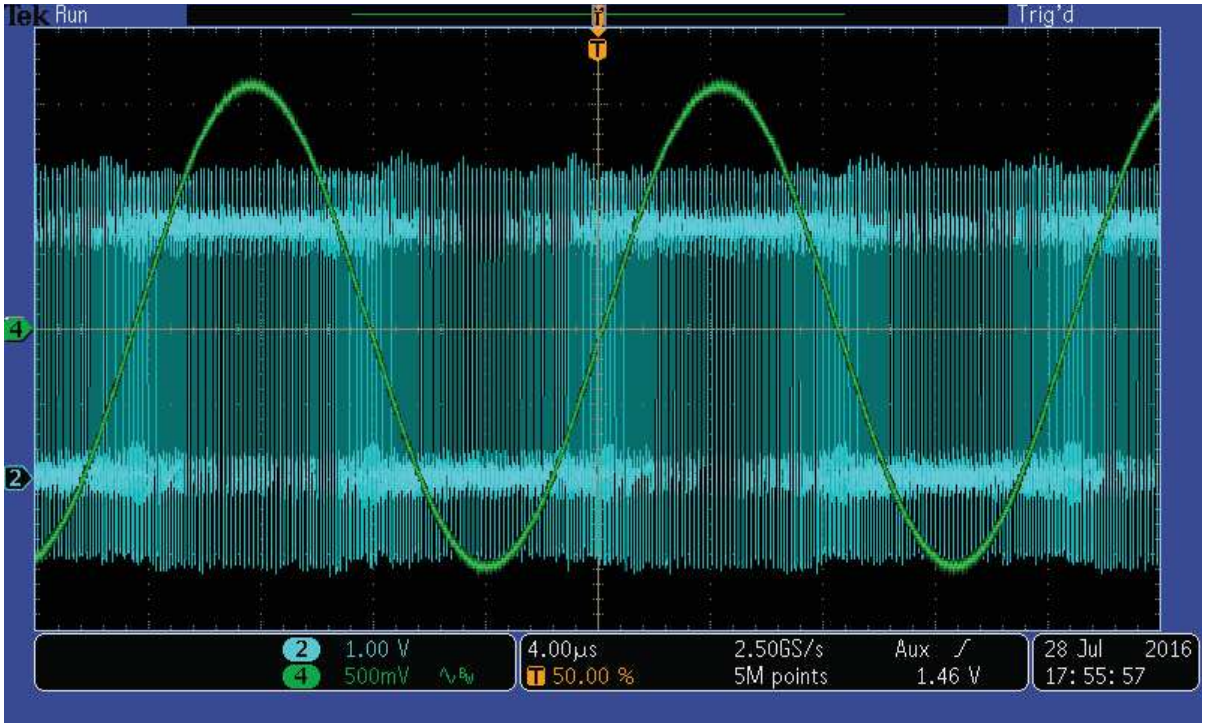
Paramètre	**Valeur**
f_{clk}	50 MHz
N	64
L	10
D	16

TABLE 3.9 – Paramètre de la DDS à concevoir pour satisfaire les performances des capteurs

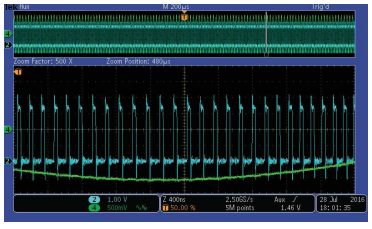
La sortie de la DDS n'excite pas directement le capteur mais passe d'abord par un filtre RC qui supprime les hautes fréquences afin d'éviter d'exciter des modes parasites sur

le capteur (modes de suspension de la structure vibrante ou de son cadre de découplage). La fréquence de coupure du filtre est choisie au delà du fondamental et atténue le bruit engendré par le processus Σ/Δ, soit une fréquence de coupure d'environ 600 kHz pour un accéléromètre VIA dont la résonance de lame est typiquement de 60 kHz.

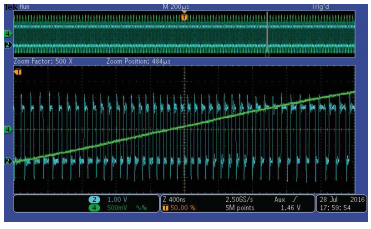
La figure 3.20 présente des captures d'écran de l'oscilloscope utilisé pour mesurer la sortie de la DDS (en bleu) et la sortie filtrée (en vert). Les outils de simulation de VIVADO permettent, pendant l'étape de mise au point, de visualiser le mot de phase (sortie d'accumulateur), le mot de forme d'onde (sortie de LUT), ainsi que les signaux de modulateur Σ/Δ.



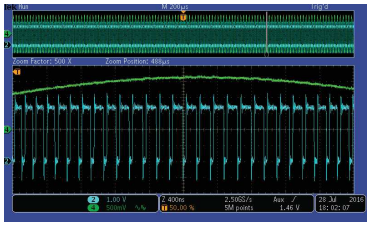
(a) Mesures en sortie de DDS et après le filtre.



(b) Zoom sur un creux de sinusoïde



(c) Zoom sur la pente la plus forte



(d) Zoom sur une crête de sinusoïde

FIGURE 3.20 – Capture d'écrans de mesures avec l'oscilloscope

La sinusoïde en sortie de filtre semble assez proche d'une sinusoïde pure et une analyse spectrale est donc réalisée pour caractériser le synthétiseur muni de son modulateur.

3.7.2 Caractérisation spectrale

La figure 3.21 présente donc les spectres de la sortie DDS avant filtrage pour trois fréquences d'horloge et pour une sinusoïde de sortie à 60 kHz.

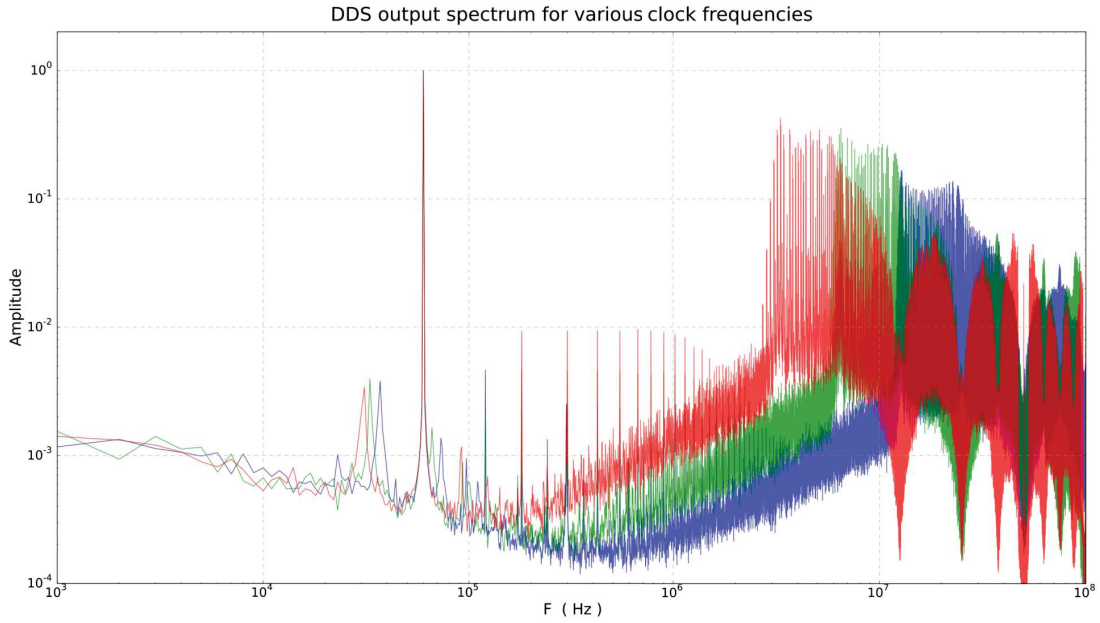


FIGURE 3.21 – Spectre de sortie mesuré de la DDS, avant filtrage, pour trois fréquences d'horloge : 25 MHz (rouge), 50 MHz (vert) et 100 MHz (bleu)

Dans les trois cas, le fondamental se distingue clairement à 60 kHz et la modulation Σ/Δ dérive bien le bruit de quantification et le repousse en hautes fréquences. Compte-tenu des paramètres de la DDS implantée, le pire spur observé devrait avoir une amplitude d'environ 10^{-3} selon l'équation (2.30) ($L = 1024$). On relève toutefois des harmoniques résiduels, à une amplitude inférieure à 10^{-2}, soit plus de 40 dB d'atténuation, probablement dus à une modulation Σ/Δ du premier ordre.

La figure 3.22 présente les amplitudes relatives de ces harmoniques, normalisées par rapport à l'amplitude attendue du fondamental en fonction de la fréquence d'horloge de la DDS. Les mesures normalisées du fondamental sont très proches de 1 et montrent le bon fonctionnement du modulateur Σ/Δ à toutes les fréquences.

En pratique, les figures 3.21 et 3.22 montrent qu'à partir d'une fréquence d'horloge de 20 MHz, soit un rapport 300 avec la fréquence cible, les harmoniques se situent à des niveaux acceptables de quelques millièmes par rapport au fondamental. Dans ces conditions, les capteurs utilisés sont réputés fonctionner correctement.

Des travaux en cours cherchent à réduire ces imperfections pour se rapprocher des performances spectrales théoriques. Ces travaux portent d'une part sur l'intégration électronique (circuit imprimé blindé, minimisation des routages de signaux et regroupement

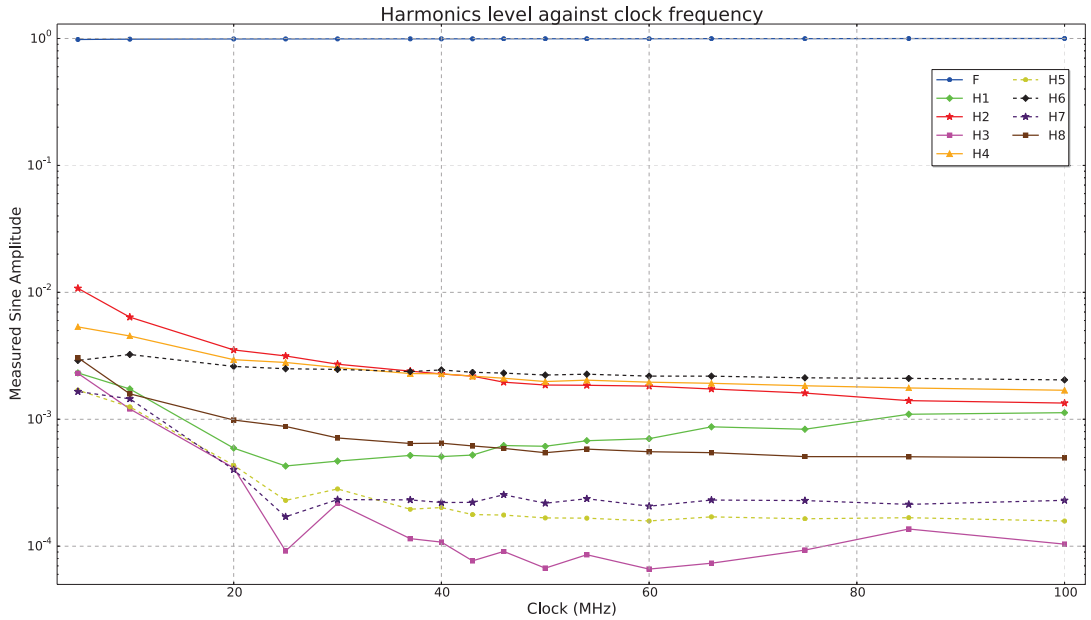


FIGURE 3.22 – Amplitudes relatives des harmoniques en fonction de la fréquence d'horloge de la DDS

par paires différentielles), et d'autre part sur le modulateur Σ/Δ : modulateur d'ordre supérieur, introduction de variations aléatoires dans les transitions pour casser les motifs répétitifs et réduire les harmoniques induits.

Ce chapitre a présenté l'ensemble des outils matériels et logiciels nécessaires à la mise en place d'une boucle fermée contenant le système de mesure de fréquence [MGM+12], de mesure du déphasage et de sa correction et le système d'excitation des capteurs. Les performances du compteur de fréquence et de la DDS n'atteignent pas encore les performances limites théoriques, mais la mise en évidence de l'origine des problèmes et les améliorations apportées et à venir offrent une base suffisante à la réalisation de la boucle fermée.

Chapitre 4

Validations expérimentales sur capteurs inertiels

Sommaire

4.1 Accéléromètre vibrant 121
4.1.1 Assemblage de la cellule 121
4.1.2 Réalisation de mesures 122
4.1.3 Analyse des résultats 124
4.2 Gyromètre vibrant 125
4.2.1 Assemblage de la cellule 125
4.2.2 Architecture électronique 125
4.2.3 Réalisation de mesures 128
4.2.4 Analyse des résultats 129

Les périphériques et l'algorithme d'estimation du déphasage étudiés précédemment ont conduit à l'élaboration d'un prototype en boucle fermée avec des capteurs réels. Suite aux difficultés rencontrées lors de la synthèse de la DDS sur FPGA, la chaîne numérique n'est pas pleinement opérationnelle dans les expériences menées : la modulation Σ/Δ est inactive et la sortie simplifiée en signal carré (c'est le bit de poids fort de l'accumulateur de phase). En contrepartie, elle fonctionne à 300 MHz. Cela ne pose pas de problèmes sur les ports d'entrée/sortie du FPGA car en l'absence de Σ/Δ le taux de transitions sur les ports n'est pas comparable à la fréquence d'échantillonnage, mais seulement à la fréquence synthétisée, de l'ordre de 10 000 fois inférieure.

Le microcontrôleur synthétisé sur le FPGA se charge de :

— maintenir les capteurs en résonance par une PLL ;

— collecter les dates de captures des différents évènements ;

— composer les trames de données et les envoyer à l'ordinateur hôte.

L'ordinateur hôte (un PC) se charge de :

— recevoir et dépaqueter les trames ;

— transformer ces données binaires brutes en grandeurs physiques ;
— afficher graphiquement et enregistrer sur disque les résultats.

La première partie de ce chapitre décrit les expériences réalisées et analyse les résultats et performances obtenus avec des capteurs accélérométriques et gyrométriques.

La suite du chapitre décrit une approche logicielle de la boucle à verrouillage de phase, qui exige des blocs spécifiques dans le processeur (diviseur, FPU [1]), et propose des modifications mineures des périphériques développés en vue d'optimiser les calculs ou le fonctionnement général de l'électronique.

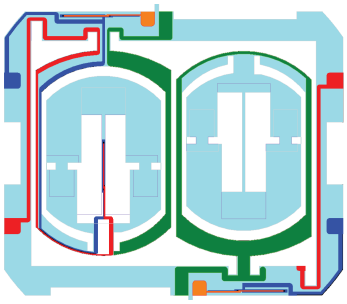
1. *Floating Point Unit*, unité de calcul en virgule flottante

À partir d'une même architecture sur FPGA (même compteur de fréquence et DDS, même microcontrôleur), deux séries d'expériences en boucle fermée ont été réalisées, avec des cellules accélérométriques puis gyrométriques.

4.1 Accéléromètre vibrant

4.1.1 Assemblage de la cellule

La tête accélérométrique utilisée sur ce prototype est une cellule DIVA développée au sein de l'unité de recherche CMT de l'Onera. Sa particularité est de grouper en une seule structure monolithique en quartz deux cellules accélérométriques VIA, décrites au chapitre 1 (paragraphe 1.1.3), en configuration différentielle (tête-bêche). Ainsi, les effets de mode commun comme la température sont minimisés par l'intimité des deux cellules. L'appariement des paramètres (facteurs d'échelle, facteurs de qualité, résistances motionnelles) est également favorisé (figure 4.1).



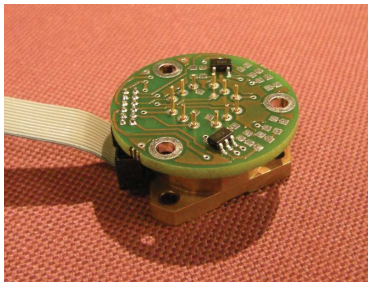
(a) Cellule accélérométrique à lame vibrante DIVA



(b) Montage sur embase



(c) Encapsulation sous vide en boîtier cuivre



(d) Tête accélérométrique complète avec son électronique de proximité

FIGURE 4.1 – Prototype d'un accéléromètre VIA

Sur la cellule utilisée, les paramètres sont nominaux (cf table 4.1) à l'exception du facteur d'échelle, d'un facteur 2 plus petit qu'attendu ($15\,\mathrm{Hz\,g^{-1}}$ d'après les modélisations par éléments finis de la structure vibrante). Ceci est la conséquence de problèmes de réalisation et notamment de maîtrise de l'épaisseur de la lame lors de la gravure chi-

Paramètre	Valeur
Taille de la cellule	∅ 10 mm
Fréquence propre de la lame	60 kHz
Facteur de qualité	10 000
Largeur de bande de la résonance	6 Hz
Sensibilité ou Facteur d'échelle	7 Hz g^{-1}

TABLE 4.1 – Paramètres des accéléromètres de la cellule DIVA utilisée

mique, mais n'empêche pas la campagne de mesures et l'évaluation des performances de l'électronique proposée. Il conviendra toutefois d'en tenir compte dans les perspectives de performance du capteur.

L'électronique de proximité est constituée d'un amplificateur de charge qui transforme les charges piézoélectriques collectées par les électrodes en tension sinusoïdale, et d'un comparateur pour transformer la sinusoïde en signal binaire destiné au FPGA. Chaque cellule de la paire est associée à une électronique dédiée.

Un exemple de signal accélérométrique mesuré au laboratoire est donné sur la figure 4.2. Ce signal est la variation de fréquence des lames, maintenues chacune à sa résonance par un circuit oscillateur. La différence de ces fréquences est proportionnelle à l'accélération recherchée, et permet au final de rejeter :

— les modes communs de variation fréquentielle dus par exemple à la température ;
— les fluctuations de l'horloge de référence du FPGA commune aux deux électroniques de mesure.

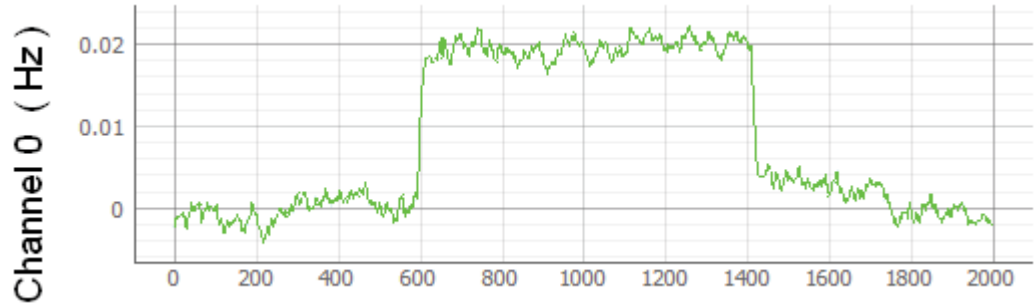
4.1.2 Réalisation de mesures

Initialement au repos (pesanteur 1 g dans le plan de la cellule de quartz, donc perpendiculaire à l'axe sensible), l'accéléromètre mesure la fréquence propre de référence de la lame vibrante, lorsque la structure est au repos, sans contrainte d'extension ni de compression due à une accélération extérieure. Les fréquences de référence des deux lames étant par construction identiques, leur différence résiduelle une fois les cellules gravées est un biais du capteur, de l'ordre de quelques centaines de Hz, équivalentes à quelques dizaines de g, mais heureusement très stable.

Dans l'exemple de la figure, une faible inclinaison de 0,17 ° (soit 3 mrad) est appliquée à la paire différentielle entre les échantillons 600 et 1400 (la fréquence d'échantillonnage est de 40 Hz). Cette inclinaison, équivalente à une variation d'accélération de 3 mg, engendre des variations symétriques de 20 mHz des fréquences de résonance des lames. Le signal différentiel, normalisé par rapport à la fréquence de résonance moyenne des lames est

tracé sans dimension sur la figure 4.2c.

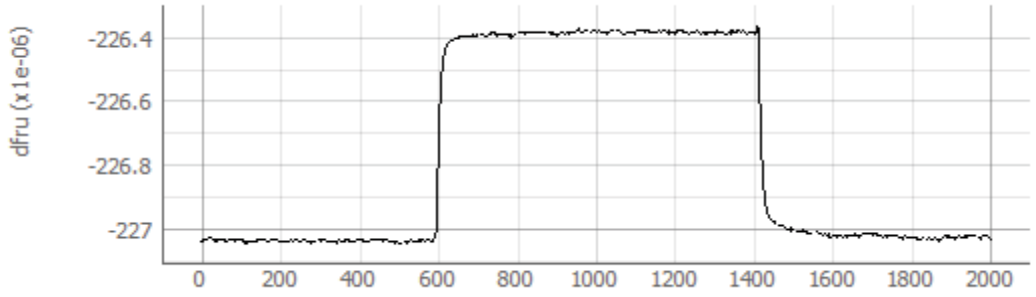
$$\frac{\delta f}{f} = \frac{F_1 - F_2}{F_0}$$



(a) Cellule de gauche



(b) Cellule de droite



(c) Signal différentiel normalisé. Écart de fréquence autour de la résonance en Hz.

FIGURE 4.2 – Signaux accélérométriques différentiels

Un important signal de bruit de mode commun (de même signe) est observé sur les deux mesures, qui révèle probablement les fluctuations de l'horloge du FPGA. Cette horloge étant commune aux deux électroniques, le chapitre 3 (section 3.3) a montré que ce bruit est rejeté par une mesure différentielle. La variance d'ALLAN peut alors être évaluée sur une série longue et dans un environnement stabilisé (en température).

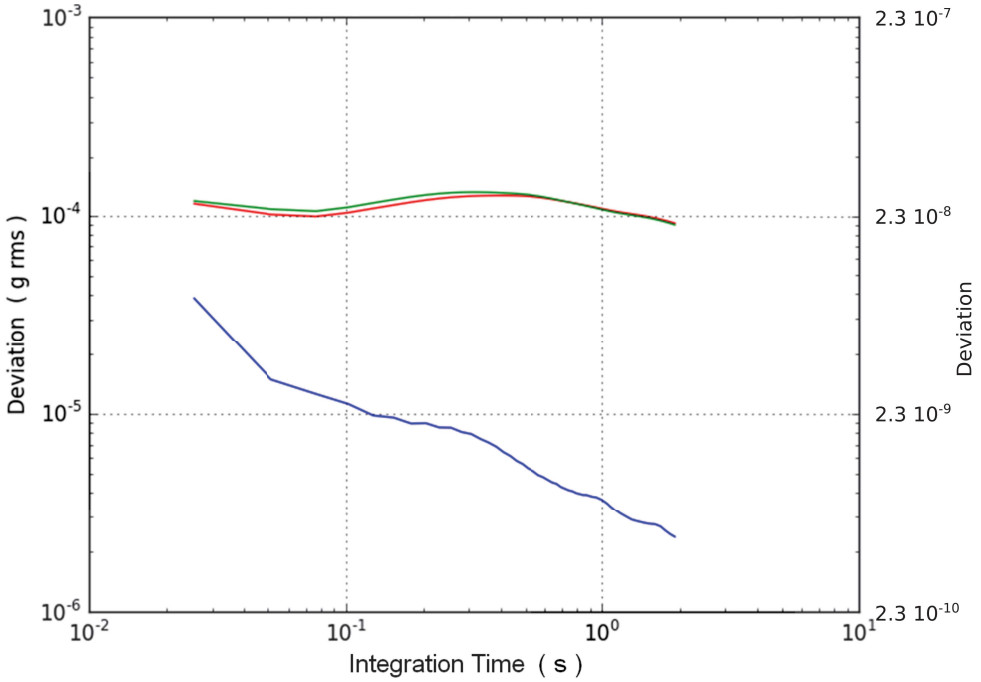


FIGURE 4.3 – Variance d'ALLAN pour la cellule accélérométrique. (rouge, vert) variance pour les cellules séparées. (bleu) variance du signal différentiel.

4.1.3 Analyse des résultats

Conformément au chapitre 3, l'horloge du compteur de fréquence du FPGA est fixée à 300 MHz, valeur tout à fait accessible aux FPGA modernes, qui sont capables du double même en entrée de gamme.

La fréquence de comptage brute est la fréquence de résonance (60 kHz), c'est à dire que le compteur fournit 60000 dates de fronts du comparateur d'entrée par seconde. Par contre la fréquence d'échantillonnage est de 40 Hz, c'est-à-dire que la fenêtre de mesure de la fréquence est de 25 ms (premier point à gauche de la figure 4.3). L'écart-type σ_F observé sur cette fenêtre est de 560 µHz en fréquence, soit 40 µg en accélération. Il est relié, d'après l'équation (2.6) à l'écart-type sur les datations :

$$\frac{\sigma_T}{T_G} = \frac{\sigma_F}{F_0} \tag{4.1}$$

D'où $\sigma_T = 4{,}7\,\text{ns} = 1{,}4\,T_C$ (T_C est la période d'horloge du compteur de fréquence, 3,3 ns). Les datations sont donc bien réalisées avec une résolution de l'ordre de la période d'horloge du compteur de fréquence.

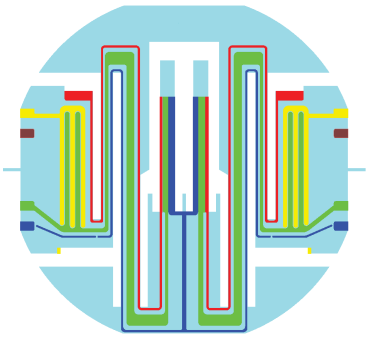
Le calcul de la variance d'ALLAN donne la performance sur des fenêtres de mesure plus longues. Une performance de 2,5 µg est atteinte à 2 s, mais une meilleure résolution pourrait vraisemblablement être obtenue sur une série plus longue (les séries effectuées avec cette expérience n'étaient pas assez longues pour prolonger la variance d'ALLAN jusqu'à 10 s). Le minimum prévu par l'étude théorique de la cellule étant de 1 µg à 10 s

([LG10] et figure 1.5), l'actuel relevé est au moins compatible avec cette performance.

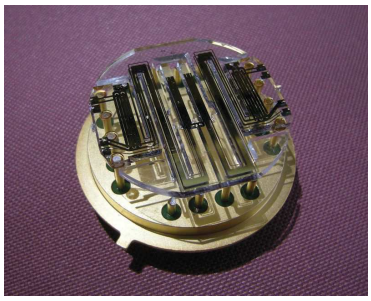
4.2 Gyromètre vibrant

4.2.1 Assemblage de la cellule

La tête gyrométrique utilisée est une cellule VIG, elle aussi développée dans l'unité. Également monolithique, cette structure intègre un diapason vibrant (figure 4.4b, au centre) et des cadres de découplage assurant les liaisons vers les zones de fixation latérales. Les modes pilote (drive), dans le plan, et détecteur (sense), hors plan, du diapason (chapitre 1, figure 1.11) sont parfaitement découplés vis à vis de l'extérieur, maximisant ainsi le facteur de qualité.



(a) Cellule gyrométrique vibrante VIG



(b) Montage sur embase



(c) Encapsulation sous vide en boîtier cuivre



(d) Tête gyrométrique complète avec son électronique de proximité

FIGURE 4.4 – Prototype d'un gyromètre VIG

4.2.2 Architecture électronique

Contrairement à l'accéléromètre, la sortie du gyromètre n'est pas une variation de fréquence, mais d'amplitude. Le rôle du synthétiseur est de maintenir le mode pilote à sa résonance, moyennant une boucle à verrouillage de phase. Puis, par couplage CORIOLIS en présence d'une rotation, le mode détecteur hors plan est plus ou moins sollicité, en

Paramètre	Valeur
Taille de la cellule	∅ 37 mm
Fréquence propre du mode pilote (plan)	5889 Hz
Fréquence propre du mode détecteur (hors plan)	5730 Hz
Facteur de qualité des modes	400 000
Facteur d'échelle	$1{,}2 \times 10^{-15}$ °C/°/s

TABLE 4.2 – Paramètres de la cellule gyrométrique VIG utilisée

régime forcé, à la fréquence du mode pilote. L'amplitude de ce mode hors plan donne alors la rotation du capteur.

L'architecture correspondante est présentée figure 4.5.

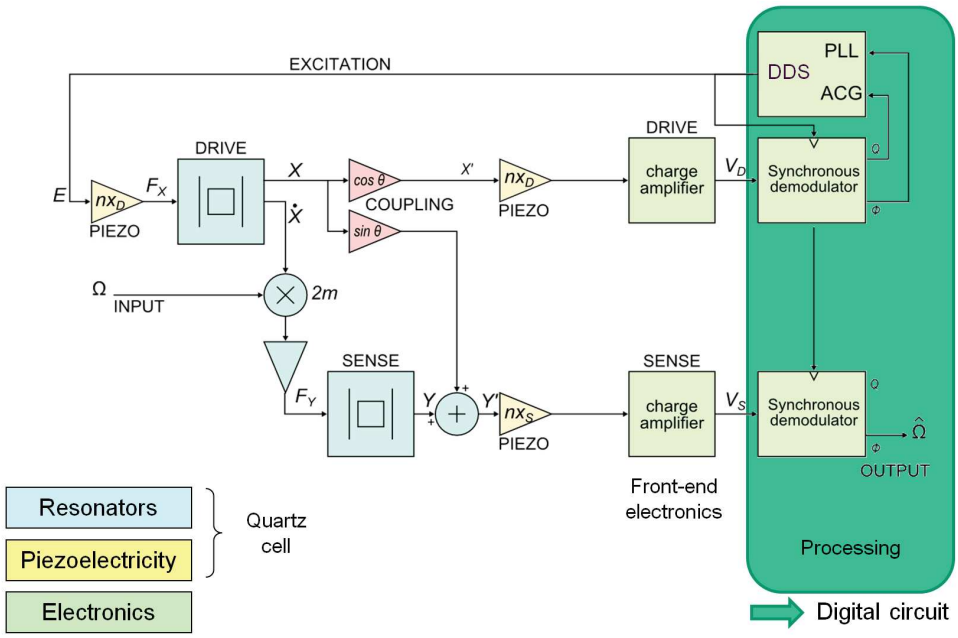


FIGURE 4.5 – Architecture électronique du gyromètre vibrant. Les propriétés piézoélectriques du quartz sont utilisées aussi bien pour l'excitation (transformation de la tension appliquée en force exercée sur le résonateur pilote), que pour la détection (transformation des déformations du résonateur en charges électriques).

Avant de décrire la campagne de mesures effectuée sur ce prototype, il convient d'insister sur un défaut récurrent des gyromètres vibrants : le couplage en quadrature, ou couplage mécanique. Ce phénomène, décrit dans la littérature [DGKLT12], est dû à une inclinaison des modes pilote et détecteur du diapason par rapport au plan de la cellule. L'effet de cet angle est représenté en rouge sur la figure 4.5, et induit, par une projection du pilote sur le détecteur, un signal en sortie, proportionnel au pilote, même en l'absence de rotation. Bien que démodulé en quadrature par rapport au signal CORIOLIS utile, des erreurs de phase même minimes dans le système engendrent alors un biais sur la sortie. Ce couplage peut être très élevé et a été mesuré à 9000 °/s sur les cellules utilisées.

Un autre point à préciser est l'absence volontaire de convertisseur analogique-numérique (ADC) dans la chaîne de mesure, à des fins de simplification pour une éventuelle qualification (spatiale par exemple) de l'électronique. Seuls les amplificateurs de charges sont conservés par rapport à une architecture analogique et les ADC que l'on trouve dans un étage classique de numérisation sont remplacés par de simples comparateurs, permettant au FPGA d'effectuer des mesures de phases par datation des passages par 0.

En effet, en présence d'un important signal en quadrature V_Q, la mesure d'un petit signal en phase v_Ω est accessible par le déphasage qu'il induit sur V_Q. Pour le signal détecteur V_S, cela s'exprime par :

$$V_S = \underbrace{V_Q \cos \omega t}_{\text{quadrature}} + \underbrace{v_\Omega \sin \omega t}_{\text{phase}} = V \cos(\omega t + \phi) \tag{4.2}$$

avec :

$$\begin{cases} V^2 &= V_Q^2 + v_\Omega^2 \\ \phi &= \arctan \frac{v_\Omega}{V_Q} \end{cases}$$

Lorsque $v_\Omega \ll V_Q$, l'approximation $\arctan x \approx x$ est valable, soit $v_\Omega \approx \phi V_Q$ et $V \approx V_Q$.

V_Q sert alors de facteur d'échelle. Cette mesure de phase ϕ a aussi l'avantage de rejeter les fluctuations du pilote car le signal Coriolis est directement proportionnel (au sens complexe, grandeurs soulignées dans les équations ci-dessous) au signal pilote :

$$\begin{cases} \underline{V_D} &= \underline{H_D}\underline{X} \\ \underline{V_S} &= \underline{H_S}\left(c\underline{X} + 2m\Omega\underline{G_D}\underline{\dot{X}}\right) \end{cases} \tag{4.3}$$

où Ω est la vitesse de rotation à mesurer, $\underline{H_D}$ et $\underline{H_S}$ sont les fonctions de transfert des amplificateurs de charges (elles ont la même phase et ne diffèrent que par le gain). $\underline{G_D}$ est la fonction de transfert du résonateur hors plan à la fréquence pilote (on montre que $G_D \approx \frac{1}{\Delta F}$, avec ΔF l'écart de fréquence entre les deux modes), m est la masse équivalente du résonateur et c est le taux de couplage de la cellule ($\sin(\theta)$ sur la figure 4.5).

$$\frac{\underline{V_S}}{\underline{V_D}} = \frac{H_S}{H_D}\left(c + j\frac{2m\omega}{\Delta F}\Omega\right) \tag{4.4}$$

Au final, $\tan\phi = \frac{2m\omega}{c\Delta F}\Omega$, et lorsque ϕ est petit, soit lorsque Ω est petit devant le couplage :

$$\phi = \frac{2m\omega}{c\Delta F}\Omega \tag{4.5}$$

ω reste très proche de $2\pi F_{DRIVE}$ et les fluctuations de la PLL sont négligeables. $c\Delta F$ est

Fréquence du compteur de fréquence dans le FPGA	F_C	300 MHz
Fréquence pilote du résonateur	F_{DRIVE}	5889 Hz
Fréquence d'échantillonnage	F_S	20 Hz
Couplage apparent servant de pleine échelle		900 °/s

TABLE 4.3 – Rappel des paramètres de performances de la cellule VIG utilisée pour les expériences.

une constante qui varie un peu avec la température (de l'ordre du pourcent), affectant légèrement le facteur d'échelle, mais de façon très répétable.

4.2.3 Réalisation de mesures

Les phases des modes pilote et détecteur sont mesurées par lecture de l'accumulateur de phase du synthétiseur au moment du passage par zéro des sinusoïdes pilote et détecteur. Les instants de passage par zéro sont identifiés par des comparateurs qui déclenchent la mémorisation de la phase du synthétiseur dans des registres mémoires. La précision sur la date de l'évènement est donc d'une période d'horloge du FPGA, soit 3,3 ns à 300 MHz. Le microcontrôleur embarqué lit ensuite ces phases et les transmet à l'ordinateur hôte, après décimation car le débit de transmission est insuffisant pour transmettre les valeurs à chaque période du signal. La phase du pilote est asservie à $\frac{\pi}{2}$ par le micro-contrôleur afin d'entretenir la résonance, et la phase du détecteur, proche de la phase pilote, permet de calculer Ω à partir de la mesure de ϕ, selon l'équation (4.2).

Au préalable, le couplage a été réduit d'un facteur 10 par ré-injection du pilote sur le détecteur. Le procédé consiste simplement à ajuster une impédance de ré-injection entre la sortie pilote (sur la figure 4.5) et l'entrée de l'amplificateur de charge de détection. Le couplage résiduel est ainsi réduit à 900 °/s et la sensibilité est augmentée d'un facteur 10, tout en restant dans les conditions d'approximation de la mesure par déphasage.

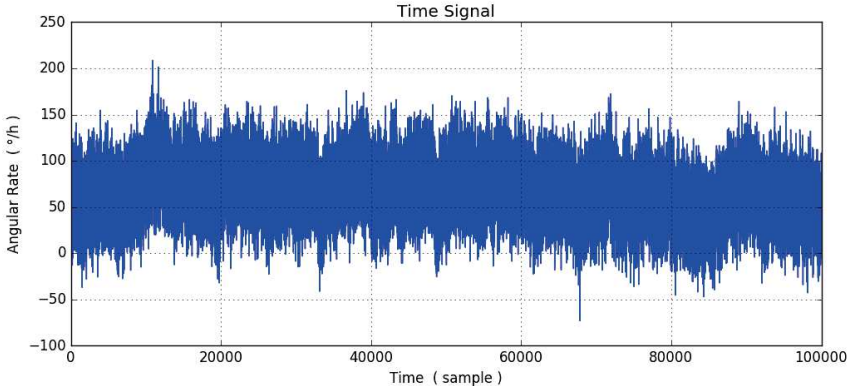


FIGURE 4.6 – Signal temporel de sortie du gyromètre au repos

Le déphasage pilote-détecteur est tracé figure 4.6, ramené en °/h en utilisant le facteur d'échelle de l'équation (4.3). Ce dernier est déterminé pour une rotation stabilisée sur table tournante de 90 °/s ; le déphasage entre pilote et détecteur est alors de 0,1 rad (10% de la pleine échelle), soit 5,7 °, ce qui se traduit par 800 périodes d'horloge à 300 MHz. Le facteur d'échelle est donc de 9 périodes d'horloge par degré par seconde, ou $0,002$ période d'horloge par degré par heure.

La variance d'Allan associée est représentée figure 4.7.

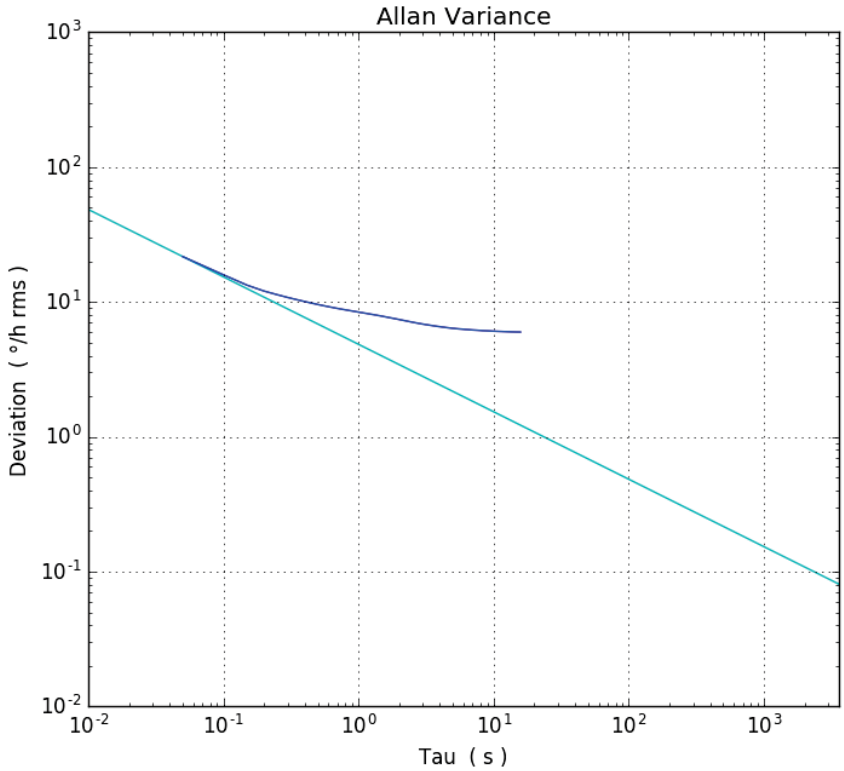


FIGURE 4.7 – Variance d'Allan de la sortie du gyromètre. Variance calculée à partir des mesures (bleu). Asymptote pour les temps d'intégration courts (clair).

4.2.4 Analyse des résultats

La mesure de phase est faite sur de nombreux échantillons (en fait à chaque période de la sinusoïde à 5889 Hz) et peut donc bénéficier d'une décimation afin de donner une grandeur moyennée pour une fréquence d'échantillonnage donnée. Par exemple, à 20 Hz pour les mesures précédentes, la mesure du déphasage est une moyenne de 294 mesures (5889 Hz ÷ 20 Hz). Son écart-type est donc réduit d'un facteur $\sqrt{294}$.

L'ordinateur fournit les mesures de phases à 20 Hz, avec un écart type observé de $3,8 \times 10^{-4}$ °, soit $0,054\, T_C$ (période d'horloge FPGA). En remontant à la fréquence por-

Rotation Ω	Phase			
90 °/s	0,1 rad	5,7 °	$810\,T_C$	Facteur d'échelle
400 °/h	$1{,}2 \times 10^{-4}$ rad	$7{,}1 \times 10^{-3}$ °	$0,9\,T_C$	RMS à F_0
22 °/h	$6{,}7 \times 10^{-6}$ rad	$3{,}8 \times 10^{-4}$ °	$0,054\,T_C$	RMS à F_S
0,08 °/h	$2{,}5 \times 10^{-8}$ rad	$1{,}4 \times 10^{-6}$ °	$0,0002\,T_C$	RMS à 1 h (ARW)

TABLE 4.4 – Resumé des performances obtenues avec le prototype gyrométrique

teuse, on en déduit que l'écart type sur les mesures brutes est de $0,9\,T_C$ ($0.054 \times \sqrt{294}$). Une variable aléatoire uniforme dans l'intervalle $\left[-\frac{1}{2}; \frac{1}{2}\right]$ ayant pour écart type $\sqrt{\frac{1}{12}} = 0,3$, la valeur de $0,9$ observée montre que les mesures initiales sont comprises dans un intervalle de 3 périodes d'horloge, soit 10 ns, au lieu d'une en théorie. Ceci est malgré tout satisfaisant, compte tenu des réglages de gains non optimaux qui ont finalement laissé en entrée du comparateur des signaux faibles, de l'ordre de quelques dizaines de milliVolt, alors que la plage d'entrée du comparateur est de l'ordre du Volt.

L'Angular Random Walk (ARW) est par définition la valeur à 1 h de l'asymptote de la variance d'ALLAN pour les temps d'intégration courts. Sur la figure 4.7, cet ARW est estimé à 0,08 °/$\sqrt{\text{h}}$ (ligne claire), une valeur du même ordre que celle obtenue sur des maquettes précédentes de l'équipe, pour une architecture analogique. La démonstration qu'une plateforme essentiellement numérique peut gérer la fonction gyrométrique dans une électronique intégrée est donc acquise.

Des marges de progrès sont néanmoins possibles sur plusieurs points :

— une meilleure compensation du couplage mécanique par ré-injection afin de descendre à un couplage résiduel de l'ordre de la pleine échelle souhaitée, soit environ 100 °/s, ou même moins dans des applications spatiales où les vitesses de rotation sont assez faibles, y compris dans les modes de défaillance ;

— un meilleur réglage des gains d'amplificateurs de charges et de plage de tension en entrée de comparateurs, de façon à mesurer avec une erreur inférieure à une période d'horloge les passages par zéro ;

— l'ajout d'une deuxième mesure par période, sur les fronts descendants en complément des fronts montants, permettant de gagner un facteur $\sqrt{2}$ sur la résolution sans électronique externe supplémentaire.

Au total, un facteur 30 est encore à gagner, pour espérer obtenir un ARW meilleur que 0,003 °/$\sqrt{\text{h}}$, aux portes de la classe navigation.

Conclusion

Résumé de la démarche

Ce mémoire a présenté le travail effectué en vue de concevoir une architecture électronique pour les micro-capteurs inertiels vibrants en quartz développés à l'Onera.

L'étude a débuté par un rappel des lois physiques fondamentales de la mesure inertielle, avant de présenter brièvement les capteurs qui effectuent ces mesures. Cette présentation se voulait non exhaustive et a surtout conduit à introduire les micro-capteurs inertiels de l'Onera, leurs principes de fonctionnement et leurs performances, qui ont servi par la suite à dimensionner l'électronique.

Ces capteurs sont des résonateurs et requièrent donc un système d'excitation afin de former un oscillateur. Cette entrée en matière a alors présenté le principe des oscillateurs analogiques avant de s'intéresser à des systèmes numériques d'excitation de résonateurs analogiques. Couplé à un système de mesure de phase, qui sert aussi à mesurer la fréquence, une architecture générique de l'électronique numérique de conditionnement des capteurs de l'Onera a été définie. Cette architecture repose sur deux blocs essentiels :

— un système de comptage de fréquence ;
— un système d'excitation des capteurs vibrants.

L'architecture définie, ce mémoire s'est ensuite intéressé à la théorie associée à ces deux blocs.

L'attention s'est portée en premier lieu sur les techniques de comptage et leurs évolutions pour mener au comptage réciproque. Afin d'exploiter plus de données, cette technique a conduit au comptage par marquage temporel et au calcul d'une régression linéaire pour mesurer la fréquence, améliorant ainsi la résolution de la mesure par rapport au comptage réciproque. Ces deux techniques, comptage réciproque et marquage temporel, nécessitent le même matériel et ne diffèrent que par le traitement logiciel associé. Ainsi, une même électronique peut fonctionner en comptage réciproque, plus simple et plus rapide à mettre en place (utile pour des besoins de développement et de test), ou par calcul de régression linéaire, plus performante (mais plus compliquée à mettre en œuvre).

Pour la seconde fonction numérique, l'architecture retenue utilise une synthèse numérique directe, ou Direct Digital Synthesis (DDS) en Anglais. L'étude a alors défini les

paramètres qui caractérisent un tel système et examiné l'impact de chacun sur le spectre de sortie. Elle a ainsi établi des contraintes entre certains de ces paramètres et montré leur influence ou leur absence d'influence sur les spectres obtenus. En particulier, le nombre de bits d'adresse de la LUT impacte directement la pureté spectrale et ce chapitre a donné une équation afin de déterminer cette valeur en fonction de la résolution voulue sur la mesure de fréquence (donc sur la mesure de l'accélération).

En outre, en vue de s'affranchir au maximum des composants analogiques, la solution retenue a envisagé l'utilisation d'un convertisseur Σ/Δ du premier ordre. Ce type de convertisseur complexifie le composant DDS et modifie le spectre de sortie en créant un bruit de quantification à haute fréquence, facilement éliminé par un simple filtre passif (type R-C) en sortie de DDS, seul élément analogique de la fonction.

L'étude théorique achevée, le travail de conception a pu débuter par une architecture de compteur de fréquence, limitée à deux entrées de captures (ce qui correspond à un accéléromètre), mais tout à fait extensible à un nombre d'entrées supérieur.

Cette architecture utilise au mieux les ressources disponibles sur le FPGA retenu et les solutions proposées par les outils de développement. Cependant, elle n'en est pas dépendante pour autant et il est possible de la porter sur d'autres micro-contrôleurs, moyennant éventuellement l'adaptation de l'interface avec le bus, s'il diffère du bus AXI utilisé ici.

Bien que cette architecture soit conçue pour atteindre une résolution fréquentielle de 3×10^{-10} en mesure simple, les mesures effectuées ont révélé des performances inférieure d'un facteur 1000 environ. L'observation de bruit semblables sur les signaux simples motive à calculer une mesure différentielle qui atteint cette fois-ci les performances escomptées : le bruit commun disparaît. Des expériences complémentaires ont alors montré que la limitation résidait en dehors du périphérique, dans le système de génération automatique de l'horloge de référence. Seule la conception d'un périphérique dédié à la formation des signaux d'horloge pourrait fournir une réponse précise quant à l'origine exacte de la perturbation observée.

Concernant le périphérique DDS, une exploitation des propriétés de symétrie d'une sinusoïde permet de réduire la taille de la LUT, ce qui autorise soit un convertisseur numérique-analogique de plus grande résolution, soit une mémoire plus petite et donc un système plus compact. Enfin, les contraintes de performances des capteurs ont permis de dimensionner une DDS adaptée à l'aide des équations établies lors de l'étude théorique.

Une fois les composants matériels développés, il ne restait plus qu'à concevoir un algorithme pour entretenir la résonance des capteurs et mesurer la fréquence des accéléromètres. Ce mémoire a proposé des solutions logicielles et estimé la puissance nécessaire à leur exécution.

Perspectives

Le travail présenté dans ce manuscrit pose les bases d'une architecture numérique pour les capteurs inertiels vibrants de l'Onera. Il reste à améliorer les performances en comptage simple du compteur de fréquence, mais aussi à poursuivre le développement du composant DDS.

En ce qui concerne le compteur de fréquence, ou plus généralement le système de datation d'évènements, qui sert également à la mesure de phase, l'augmentation des performances passe par la précision de datation des transitions de signaux. Les horloges internes dans les FPGA sont aujourd'hui communément de 300 MHz, offrant donc une résolution de datation de 3 ns. Elles ne dépassent pas 600 MHz cependant. Pour améliorer cette résolution, des techniques sont présentes dans la littérature et seront à explorer : horloge multiphase, chaînes de délais, verniers temporels. Toutes ces techniques sont entièrement numériques et ont montré des résolutions sub-nanoseconde.

Au sujet de la DDS, les tests et expériences sur le composant développé doivent se poursuivre, notamment afin d'étudier en détail les spectres réels obtenus en sortie. De plus, l'architecture complète devra aussi effectuer une lecture des mesures gyrométriques. Dans ce cadre, l'accès à la phase courante (c'est-à-dire la sortie de l'accumulateur de phase) sera particulièrement profitable et seul un système intégré comme l'architecture numérique proposée dans cette thèse rend possible l'exploitation de cette information.

Se pose aussi la question du composant hôte de l'architecture. Ce travail a commencé avec un Spartan 6 de Xilinx, mais trois générations successives de FPGA se sont succédées depuis chez Xilinx et les outils de développement ont également évolué. Le marché des FPGA évolue très vite, et nul doute qu'une architecture apparemment gourmande en ressources aujourd'hui semblera dérisoire demain ; on observe un facteur 10 entre le Spartan-6 et son équivalent d'entrée de gamme 5 ans plus tard, et un facteur presque 1000 avec les modèles les plus avancés aujourd'hui.

Ce composant déterminé, l'architecture matérielle figée et le code embarqué abouti, il restera à déterminer la consommation électrique du système conçu en fonctionnement normal. Des choix d'optimisations ou de simplifications (réduction de la LUT de la DDS, réduction du nombre de bits de l'accumulateur de phase ou du compteur de fréquence...) pourront alors réduire la consommation si le besoin s'en fait sentir. De même, des modes basses consommations pourraient être développés toujours afin de réduire cette consommation (interruptions toutes les n captures sur le compteur de fréquence par exemple, pour ne pas réveiller le processeur qui serait en veille...). Mais cette mesure de la consommation n'aura de sens qu'une fois les fonctionnalités validées et les performances souhaitées atteintes.

Pour conclure, il faut rappeler que l'approvisionnement de FPGA dans un grade spatial est très délicat, du fait du monopole américain et des restrictions fédérales dans ce

domaine. L'avantage d'une description d'architecture qui fait abstraction du matériel (langage VHDL) est de pouvoir la synthétiser sur d'autres supports. Il existe bien un FPGA spatial européen, l'ATF280F d'Atmel, mais il ne fonctionne qu'à 50 MHz maximum (soit 6 fois trop lentement pour un compteur de fréquence) et ne dispose pas d'assez de mémoire pour héberger le processeur et une DDS (il en faudrait 9 pour une plate-forme inertielle complète). Une autre possibilité de réalisation est un ASIC, option déjà envisagée par l'Onera dans le cadre d'une autre thèse et d'autres travaux.

Annexes

Annexe A

Simulateur de spectres de sortie de DDS

Cette annexe présente le code développé pour simuler les spectres de sortie d'un module de synthèse numérique directe en fonction de ses paramètres de conception. L'annexe présente en réalité une troncature du code, nettoyé de toutes les fonctions annexes et utilitaires comme, par exemple, les fonctions pour tracer les courbes. Seules restent les fonctions mathématiques, celles qui mènent effectivement aux calculs des spectres.

A.1 Pourquoi Python ?

Matlab fait office de référence lorsqu'il s'agit de simulations en ingénierie. Toutefois, il existe d'autres langages tout aussi propice à effectuer ce genre de calculs.

Parmi toutes les possibilités, Python se dégage du lot. À l'origine, Python n'est pas du tout un langage destiné aux traitements scientifiques. Il s'agit d'un langage de programmation orienté objet, aussi bien apte à exécuter de petits scripts qu'à écrire des programmes complexes. Langage interprété, il ne nécessite pas de compilation et dispose d'un mode interactif. En outre, Python fonctionne sur plusieurs plate-formes : Windows, Linux, Mac, Android...

Plusieurs bibliothèques, en particulier Numpy et Matplotlib, fournissent aujourd'hui les outils nécessaires à une utilisation scientifique du langage. iPython par exemple, fournit une console très similaire à Matlab. Dernier point, l'interpréteur officiel permet de charger des bibliothèques partagées (fichiers `.dll` sous Windows ou `.so` sous Linux). Cette particularité lui permet de réutiliser les codes écrits en Fortran par exemple, langage très répandu dans la communauté scientifique, ou en C/C++ pour gagner en performance. Son caractère libre, sa syntaxe relativement simple à apprendre et les avantages cités en ont fait, aujourd'hui, un langage apprécié des scientifiques.

Le simulateur calcule des spectres, calculs connus pour nécessiter de la mémoire et

utiliser fortement le processeur. Par conséquent, il fallait un langage capable de fonctionner à la fois sur des machines du quotidien, le plus souvent sous Windows, et des serveurs de calculs, sous Linux. Et un langage capable en plus d'utiliser les bibliothèques de calcul de transformées de Fourier, écrites en C. Voilà pourquoi le choix s'est porté sur Python.

A.2 Précisions sur le code

Les commentaires présents dans le code doivent normalement apporter les précisions nécessaires à sa compréhension. Ils contiennent aussi parfois quelques remarques générales. Toutefois, voici certaines précisions afin de clarifier l'utilité de certaines lignes, leur fonctionnement, ou insister sur certains points.

A.2.1 Spécificités de Python

Pour le lecteur non initié à Python, une précision s'impose au sujet du mot-clé `@property`. Sans entrer dans les détails, cette instruction modifie la méthode définie juste après pour la transformer en attribut. Autrement dit, au lieu de devoir appeler la méthode `instanceDDS_Simu.method()`, l'utilisateur peut se passer des parenthèses. Cela s'avère particulièrement appréciable lorsque la méthode retourne des éléments semblables à des tableaux : `instanceDDS_Simu.method[:10]` par exemple.

Employé comme le fait le code présenté rend aussi l'attribut ainsi créé immuable, c'est-à-dire que l'utilisateur ne peut pas en modifier son contenu (`instanceDDS_Simu.method[:10] = X` ne fonctionne pas).

A.2.2 Méthode `__calc_pts`

Cette méthode cherche à optimiser le nombre de points de la simulation afin de réduire les artefacts de calculs causés par la transformée de Fourier discrète. Dans l'idéal, si le premier point de la simulation se trouve à 0, alors le point suivant le dernier point se trouve aussi à 0. Il n'y a alors aucun artefacts. Dans le cas contraire, c'est-à-dire que le point suivant le dernier point de la simulation ne tombe pas sur 0, cela signifie qu'il y a un nombre non entier de période.

La méthode cherche donc un nombre de périodes avec la plus petite partie décimale possible (idéalement 0, soit un nombre entier de périodes), et avec un écart du nombre de points total de 5% au maximum (variable globale `percentile`) par rapport aux nombre de points demandés par l'utilisateur.

Pour se faire, la méthodes commence par définir les bornes inférieures et supérieures du nombre de périodes de simulation, ici entre 95% et 105% de la valeur choisie par l'utilisateur. Elle cherche ensuite la période pour laquelle le nombre de points total de la simulation présente la plus petite partie décimale (ligne 130).

En détail, les lignes 124 à 126 incluses forment un tableau de nombre entiers de périodes de simulation qui fourniraient un nombre de points entre 95% et 105% de la valeur souhaitée par l'utilisateur. Ce tableau vaut par exemple `[127, 128, 129]`.

La ligne 128 calcule le nombre de points fictifs que ces périodes fourniraient, soit un nombre décimal de points dans la simulation (`self.__ppp` est un flottant). Soit, avec le tableau exemple donné juste avant et pour `self.__ppp = 507.34`, le tableau `[64432.18, 64939.52, 65446.86]`.

La ligne 130 retient alors l'indice du tableau avec la plus petite valeur décimale, `0` dans l'exemple. Puis la méthode fournit le nombre de points, entier et donc réaliste cette fois-ci, dans la simulation (ligne 132), ainsi que le nombre, toujours entier, de périodes correspondantes.

A.2.3 Méthode `__dophase`

Le lecteur notera qu'en présence d'un accumulateur de phase, ce qui se traduit dans le simulateur par l'existence de l'attribut `__M`, le calcul effectué à la ligne 155 correspond *exactement* au fonctionnement du compteur, soit mathématiquement, avec k le temps horloge :

$$phase(k) \equiv k \times M \left[2^{counter_nbits}\right].$$

A.2.4 Méthode `dofft`

Cette annexe présente la dernière version et version la plus communément utilisée. Ce simulateur a aussi testé le calcul de cette FFT par OpenCL, outils qui permettent d'utiliser plusieurs processeurs de calculs en parallèle, et notamment le GPU, processeur de la carte graphique.

Le nombre de points dans une transformée définit directement la résolution fréquentielle de la densité spectrale obtenue. Pour obtenir une meilleure résolution, il faut davantage de points. Mais un calcul de FFT se montre particulièrement gourmand.

Avec la version présentée, un calcul sur environ un million de points demande environ une minute sur une machine avec les caractéristiques suivantes :

— CPU Intel Core i5 750 @2,75 GHz ;
— 8 GiB ;
— système d'exploitation 64 bits (Windows 7 ou Gentoo Linux).

L'augmentation ne serait-ce que d'un facteur 5 du nombre de points passe le temps de simulation de l'ordre de la minute à plusieurs dizaines de minutes. Ce qui ne devient plus réaliste lorsqu'il s'agit de lancer plusieurs simulations.

Voilà pourquoi un calcul en OpenCL, qui tire aussi parti de la carte graphique, existe aussi. Cependant, cette approche présente des problèmes de portabilité, notamment d'ali-

gnement mémoire (en particulier avec les GPU). Mais une fois en place et après vérification des résultats pour s'assurer qu'il n'y a pas d'erreur d'alignement, le temps de calcul s'en trouve grandement réduit : d'une trentaine de minutes pour 5 millions de points à quelques minutes. Le processeur graphique et ses capacités de calcul matriciel très fortement parallélisé contribue pour beaucoup à ce gain.

A.3 Code source

```
#!/usr/bin/python2
# -*-coding: utf-8-*-

"""Ce module définit une classe DDS_Simu qui, à partir des paramètres donnés à
son contructeur, se charge de tracer les spectres de sorties d'une DDS avant et
après le DAC.
"""

import sys
import numpy as np
import pyfftw
# Test de FFT par OpenCL
#import pyopencl as pycl
#import pyopencl.array
#import pyfft.cl
#import math


############################### VARIABLES GLOBALES ##############################
# Nombre de bits du compteurs
counter_nbits = 48
# Nombre de bits d'adresse de la LUT.
lut_nbits = 18
# Nombre de bits du DAC réel.
dac_nbits = 16
# Nombre de bits du DAC fictif sigma/delta. 0 signifie pas de sigma/delta.
# Pour obtenir un sigma/delta sans aucun DAC réel, dac_nbits peut valoir 0 ou 1.
sd_nbits = 0
# Fréquence de sortie souhaitée.
fout = 60532
# Horloge (compteur, DAC et sigma/delta)
fclk = 50e6
# Nombre de points générés.
N_default = 100e3
# Tolérance pour rechercher le meilleur passage par 0.
percentile = 0.05



class DDS_Simu(object):
    """Classe coeur du module. Le constructeur définit un ensemble de paramètre
qui peuvent soit être fournis soit utiliser les variables globales.
Puis diverses méthodes permettent de calculer tous les paramètres intermédiaires
en vue de tracer les spectres.
"""

    # Initialisation des variables qui ne seront pas remplies par le
```

```
    # constructeur.
    # Phase, amplitude en sortie de LUT (même si il n'y en a pas) et sortie
    # de la DDS.
    _phase = None
    _output = None
    _ampl = None

    def __init__(self, f=None, cnt=None, lut=None, dac=None, sd=None, fs=None, N=None):
        # Exécution du constructeur par défaut (pas très utile dans les faits
        # mais si un jour cette classe doit hériter d'une autre classe, tout est
        # prêt).
        super(DDS_Simu, self).__init__()
        # Nombre de bits de l'accumulateur de phase.
        if cnt == None :
            self.__counter_nbits = globals()['counter_nbits']
        else :
            self.__counter_nbits = cnt
        # Nombre de bits d'adresse de la LUT.
        if lut == None :
            self.__lut_nbits = globals()['lut_nbits']
        else :
            self.__lut_nbits = lut
        # Nombre de bits du DAC réel.
        if dac == None :
            self.__dac_nbits = globals()['dac_nbits']
        else :
            self.__dac_nbits = dac
        # Nombre de bits du DAC sigma/delta.
        if dac == None :
            self.__sd_nbits = globals()['sd_nbits']
        else :
            self.__sd_nbits = sd
        # Horloge du système, fréquence d'échantillonnage.
        if fs == None :
            self.__fs = float(globals()['fclk'])
        else :
            self.__fs = float(fs)
        # Fréquence cible.
        if f == None :
            self.__f = float(globals()['fout'])
        else :
            self.__f = float(f)
        # Nombre de points de la simulation.
        if N == None :
            self.__nperiods, self.__npts = self.__calc_pts(globals()['N_default'])
        else :
            self.__nperiods, self.__npts = self.__calc_pts(N)


    @property
    def __ppp(self):
        """Retourne le nombre de points par période.
"""
        # On s'assure que le retour soit bien un flottant.
        return float(self.__fs) / self.__f


    @property
    def __M(self):
```

```
        """Retourne la valeur de l'incrément de phase le plus proche pour
que la fréquence de sortie de la DDS soit aussi proche que possible de
la fréquence souhaitée.
"""
        if self.__counter_nbits > 0 :
            return int(np.round(self.__f / self.__fs * (1<<self.__counter_nbits)))
        else :
            return None


    def __calc_pts(self, N):
        """Calcule et retourne le nombre de points de la simulation afin
d'obtenir un nombre entier de périodes et une valeur la plus proche de 0
possible dans le centile donné.
"""
        # Définition de la tranche du nombre de périodes dans laquelle
        # la recherche de l'optimum doit s'effectuer.
        lower = int(np.floor((1 - percentile) * N / self.__ppp))
        upper = int(np.floor((1 + percentile) * N / self.__ppp))
        periods = np.arange(lower, upper)
        # Soit un nombre de points total de...
        points = periods * self.__ppp
        # Quel point est le plus proche de son arrondi ?
        k = np.argmin(np.abs(points - np.round(points)))
        # Nombre de points optimal.
        n_pts = int(round(points[k]))
        # Soit un nombre de périodes...
        n_periods = periods[k]
        return n_periods, n_pts


    @property
    def phase(self):
        """Retourne le vecteur de phase (i.e. la phase pour chaque front
d'horloge).
"""
        try :
            if self._phase.size :
                return self._phase
        except AttributeError :
            self._phase = self.__dophase()
            return self._phase


    def __dophase(self):
        """Calcule le vecteur de phase.
"""
        if self.__M :
            return np.array(np.arange(self.__npts) * self.__M % (1<<self.__counter_nbits),\
                            dtype = np.uint64)
        else :
            w = 2*np.pi * self.__f / self.__fs
            return np.mod(np.arange(self.__npts) * w, 2*np.pi)


    @property
    def ampl(self):
        """Retourne le vecteur d'amplitude (i.e. l'amplitude pour chaque front
d'horloge).
```

```
"""
        try :
            if self._ampl.size :
                return self._ampl
        except AttributeError :
            # Préparation du tableau pour des FFT avec pyfftw.
            self._ampl = pyfftw.n_byte_align_empty(self.__npts, 32, 'float64')
            self._ampl[:] = self.__doampl()
            return self._ampl


    def __doampl(self):
        """Calcule le vecteur d'amplitude.
"""
        # TODO: optimisation possible du code. Les cas où il y a une LUT et
        # un compteur ou pas de LUT et un compteur sont très similaires.
        # La seule différence provient de la troncature de phase qui a lieu ou
        # non.
        # Une LUT sur 1 bit correspond à la sortie du MSB de l'accumulateur
        # de phase seule.
        if self.__lut_nbits > 0 :
            # La sortie de la LUT demeure analogique. C'est le DAC qui va ensuite
            # tronquer les valeurs. La LUT elle, ne tronque que la phase, pas
            # l'amplitude.
            # Préparation des valeurs. Les valeurs sont exprimées entre 0 et 2.
            # Décalage de la phase à l'origine de pi / 2 pour ne pas commencer
            # le sinus par un 0. Cela revient à un cosinus.
            lut = np.cos(np.array([i * 2 * np.pi / (1<<self.__lut_nbits) for i in\
                  range(1<<self.__lut_nbits)], dtype = np.float64)) + 1
            # En présence d'un accumulateur de phase, il faut tronquer la phase.
            if self.__counter_nbits > 0 :
                # Nombre bits à décaler.
                bitshift = self.__counter_nbits - self.__lut_nbits
                # Décalage de la phase.
                phase_trunc = self.phase>>bitshift
            # S'il n'y a pas de compteur, alors il faut arrondir la phase
            # analogique.
            else :
                phase_trunc = np.uint32(np.around(self.phase / (2*np.pi) * (1<<self.__lut_nbits)))
            # Retour du résultat entre 0 et 1.
            retval = lut[phase_trunc]
            return retval / 2
        # Autre cas, il y a un accumulateur de phase mais pas de LUT.
        elif self.__counter_nbits > 1 :
            # Retour du sinus directement.
            # Encore une fois, décalage de la phase à l'origine et amplitude
            # entre 0 et 2.
            retval = np.array(np.cos(self.phase.astype(np.float64) /\
                             (1<<self.__counter_nbits) * 2 * np.pi),\
                             dtype = np.float64) + 1
        # Pas de compteur, pas de LUT.
        else :
            # Retour direct du sinus également. Mêmes remarques sur l'amplitude
            # et la phase à l'origine.
            retval = np.cos(self.phase) + 1
        # Retour du résultat pour les deux derniers cas, toujours entre 0 et 1.
        return retval / 2
```

```
    @property
    def output(self):
        """Retourne le vecteur de sortie (la grandeur après le DAC) si un DAC
est défini (i.e. self.__dac_nbits > 0 ou self.__sd_nbits > 0).
"""
        if self.__dac_nbits > 0  or self.__sd_nbits > 0 :
            try :
                if self._output.size :
                    return self._output
            except AttributeError :
                # Préparation du tableau pour des FFT avec pyfftw.
                self._output = pyfftw.n_byte_align_empty(self.__npts, 32, 'float64')
                self._output[:] = self.__dooutput()
                return self._output
        else :
            return self.ampl


    def __dooutput(self):
        """Calcule le vecteur de sortie.
"""
        # La présence du sigma/delta complique la tâche.
        # La technique sigma/delta implique deux tailles de DAC :
        # - celle du DAC réel, qui va définir les différents niveaux de tension
        # de sortie possible;
        # - celle fictive qui va définir la taille de la sortie de la LUT.
        # Cas où il n'y a pas de sigma/delta.
        if self.__sd_nbits == 0 :
            if self.__dac_nbits > 0 :
                # Dans ce cas, il faut arrondir la sortie de la LUT.
                return np.around(self.ampl * ((1<<self.__dac_nbits) - 1),\
                                 out = np.zeros(self.__npts, dtype = np.uint32))
            else :
                # Cas d'un DAC infini. On retourne l'amplitude directement.
                # À noter qu'il s'agit du même cas que celui testé dans
                # self.output.
                # Normalement, cette ligne n'est jamais exécutée.
                return self.ampl
        # Le nombre de bits du DAC réel doit être plus petit que celui virtuel
        # du DAC sigma/delta.
        elif self.__sd_nbits > self.__dac_nbits :
            # Cas sigma/delta.
            # Initialisation des variables.
            output = np.zeros(self.__npts, dtype = np.int64)
            sigma, delta = 0, 0
            # Valeur max virtuelle du DAC sigma/delta.
            sd_max = (1<<self.__sd_nbits) - 1
            # Y a-t-il un DAC ?
            if self.__dac_nbits > 1 :
                # Cas le plus complexe, le sigma delta dispose de plusieurs
                # valeurs de sortie et pas juste 0 ou 1.
                # Ensemble des valeurs possibles.
                avail_range = np.linspace(0, (1<<self.__sd_nbits)-1, 1<<self.__dac_nbits)
                # Indice max dans avail_range.
                max_avail = len(avail_range) - 1
                # Pour gagner du temps on stocke la dernière position de sortie.
                last = 0
                # Inititialisation de la boucle.
                sigma = self.ampl[0] * sd_max
```

```
                while last < max_avail and sigma > avail_range[last] :
                    # La valeur maximale possible n'a pas encore été atteinte.
                    last += 1
                # Première valeur de sortie.
                output[0] = avail_range[last]
                # Sans oublier le résidu.
                delta = sigma - output[0]
                # Reste à définir les autres...
                for i in range(1, self.__npts):
                    # Ajout du résidu précédent et de la valeur courante.
                    sigma = delta + self.ampl[i] * sd_max
                    # La valeur de la sortie varie peu. Pour gagner du temps,
                    # au lieu de tester toutes les valeurs, on part de la valeur
                    # courante.
                    # Faut-il augmenter la sortie ?
                    while last < max_avail and sigma >= avail_range[last + 1] :
                        # Incrément suivant.
                        last += 1
                    # Ou la diminuer ?
                    while last > 0 and sigma < avail_range[last] :
                        # Incrément précédent.
                        last -= 1
                    # Sortie.
                    output[i] = avail_range[last]
                    # Résidu.
                    delta = sigma - output[i]
                    # Petite vérification.
                    if delta < 0 :
                        sys.stderr.write("ERREUR dans l'algorithme au rang {}.\n\
                                          \tsigma -- delta -- output : {} -- {} -- {}\
                                          \n".format(\i, sigma, delta, output[i]))
                        break
            else :
                # Cas plus simple, il n'y a que deux valeurs possibles, 0 ou 1.
                for i in range(self.__npts):
                    # Ajout du résidu précédent et de la valeur courante.
                    sigma = delta + self.ampl[i] * sd_max
                    # Comparaison à la valeur max.
                    if sigma >= sd_max :
                        # sigma est plus grand, la sortie vaut 1.
                        output[i] = 1
                        # Et il faut retrancher la valeur max.
                        delta = sigma - sd_max
                    else :
                        # sigma est plus petit, la sortie vaut 0.
                        output[i] = 0
                        # Sauvegarde du résidu pour l'itération suivante.
                        delta = sigma
                    # Petite vérification.
                    if delta < 0 :
                        sys.stderr.write("ERREUR dans l'algorithme au rang {}.\n\
                                          \tsigma -- delta -- output : {} -- {} -- {}\
                                          \n".format(i, sigma, delta, output[i]))
                        break
        else :
            sys.stderr.write("ERREUR : le nombre de bits du DAC réel ne peut pas\
                              être plus grand que celui du DAC sigma/delta.\n")
            # Définition de output pour ne pas générer d'erreur ailleurs.
            output = np.zeros(self.__npts)
```

```
        # output est prêt !
        return output


    def dofft(self):
        """Calcule les FFT de l'amplitude avant le DAC et de la sortie après
le DAC.
"""
        # Fréquences pour lesquelles un point de la FFT va exister.
        frq = np.linspace(0, self.__fs / 2, self.__npts / 2 + 1)
        # Séléction et normalisation de l'information utile.
        ampl_fft = pyfftw.interfaces.numpy_fft.fft(self.ampl, threads = 5)[:self.__npts / 2 + 1]\
                   * 2 / self.__npts + sys.float_info.epsilon
        # Le biais (fréquence nulle) est maintenant deux fois trop grand.
        ampl_fft[0] /= 2
        # Calcul à faire pour la sortie aussi ?
        if (self.__sd_nbits > 0) or (self.__dac_nbits > 0):
            output_fft = pyfftw.interfaces.numpy_fft.fft(self.output, threads = 5)[:self.__npts / 2 + 1]\
                         * 2 / self.__npts + sys.float_info.epsilon
            output_fft[0] /= 2
        else :
            output_fft = ampl_fft
        return frq, ampl_fft, output_fft
```

Annexe B

Détails sur les périphériques développés

La description matérielle des périphériques conçus se fait en VHDL, mais elle aurait tout aussi bien pu se faire en Verilog.

Bien qu'il s'agisse d'un langage ouvert, ce qui facilite la portabilité du code, la conception d'un SoPC [1] autour d'un MicroBlaze et dans un Spartan 6 présente toutefois quelque spécificités ; spécificités auxquelles il faudra prêter attention lors du portage des périphériques.

Cette annexe vise à présenter la conception générale d'un périphérique afin de mettre en lumière ces spécificités.

B.1 Périphérique sur bus AXI

L'ensemble du système développé repose sur un processeur MicroBlaze, processeur développé par Xilinx et optimisé pour leurs FPGA. Aujourd'hui, le MicroBlaze dispose d'un bus AXI comme interface avec ses périphériques. Ce bus, développé à l'origine par la société ARM, tend à se démocratiser et à devenir la norme *de facto* des bus des microcontrôleurs embarqués.

En vue de faciliter la conception de périphérique, les outils de développements proposés par Xilinx proposent un cadre paramétrable. La figure B.1 représente ce cadre, où le périphérique développé rassemble et inter-connecte deux blocs : l'IPIF (IP Interface) et le `user_logic` que l'utilisateur modifie pour fournir les fonctions souhaitées.

Les pointillés sur la figure démarquent des connexions optionnelles, non proposées par les outils, mais que le développeur peut rajouter.

L'IPIF se charge des communications avec le bus et fournit divers signaux accessibles dans le fichier `user_logic.vhd`, et qui dépendent notamment des paramètres entrés par

1. System on Programmable Chip : système sur puce reprogrammable.

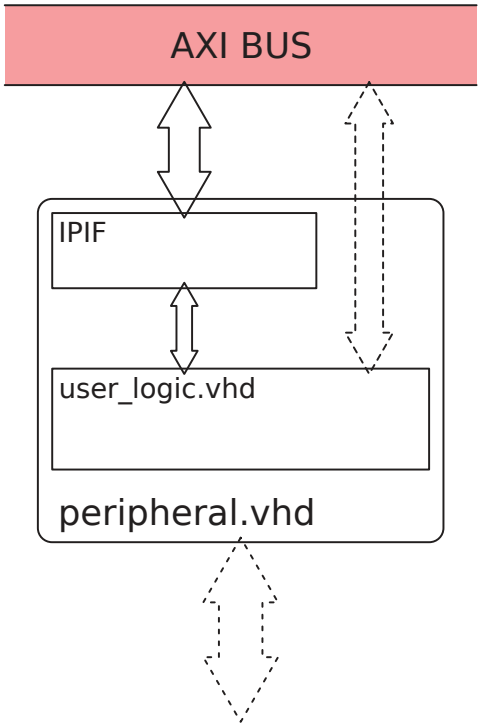


FIGURE B.1 – Architecture d'un périphérique AXI développés avec les outils Xilinx

le développeur. L'IPIF réduit ainsi considérablement le travail à fournir pour gérer le bus et laisse le développeur se concentrer sur le périphérique en lui-même.

Cette interface ne repose sur aucun composant spécifique dans le FPGA, ou tout du moins, pourrait être synthétisée dans un autre FPGA. Toutefois, il s'agit d'un bloc propriétaire Xilinx sous licence qui ne peut être utilisé que sur des composants Xilinx.

Par conséquent, lors du portage des périphériques sur d'autres composants ou en vue d'une intégration sur un ASIC [2], il faudra remplacer ce bloc. Le site OpenCores [3] propose des interfaces esclaves AXI en Verilog. Sa traduction en VHDL et son adaptation au périphérique développé pour remplacer l'IPIF ne devrait pas poser de problème particulier.

B.2 Macros et composants spécifiques au Spartan 6

Le chapitre 3 a très rapidement introduit l'architecture d'un Spartan 6. Les outils de synthèse permettent d'adresser chaque composants qui compose un CLB [4], ou plus particulièrement les composants spécifiques, comme des DSP48, les blocs mémoires... ou les blocs spécifiques comme les blocs de gestion des horloges. Un document de référence [Xilc] liste l'ensemble des ces fonctionnalités accessibles.

2. Application-Specific Integrated Circuit. Cela désigne, par abus et opposition au FPGA, un circuit intégré non reprogrammable.
3. www.opencores.org
4. Configurable Logic Block

De plus, les outils proposent certains composants fictifs : ces composants n'existent pas réellement dans le FPGA mais les outils les interprètent et tire pleinement partie des ressources disponibles pour concevoir la fonction qu'ils proposent. Il s'agit de macro. Par exemple, il existe une macro compteur qui utilise et configure automatiquement un bloc DSP48 pour fournir la fonction demandée, un compteur. Le compteur de fréquence emploie justement cette macro.

Encore une fois, lors du portage du composant, il faudra s'assurer de concevoir effectivement un compteur. La figure B.2 présente l'architecture générique d'un compteur synchrone avec possibilité de pré-chargement, de reset et de choix du sens de comptage. Il s'agit, encore aujourd'hui, de l'architecture la plus répandue de compteur binaire simple. Le lecteur remarquera la répétition d'un même bloc à partir de la deuxième bascule (sortie Q1), bloc repris à la figure B.3. L'ajout d'un bloc de ce type, c'est-à-dire l'ajout d'un bit à la valeur de comptage, multiplie par deux la valeur maximale atteignable.

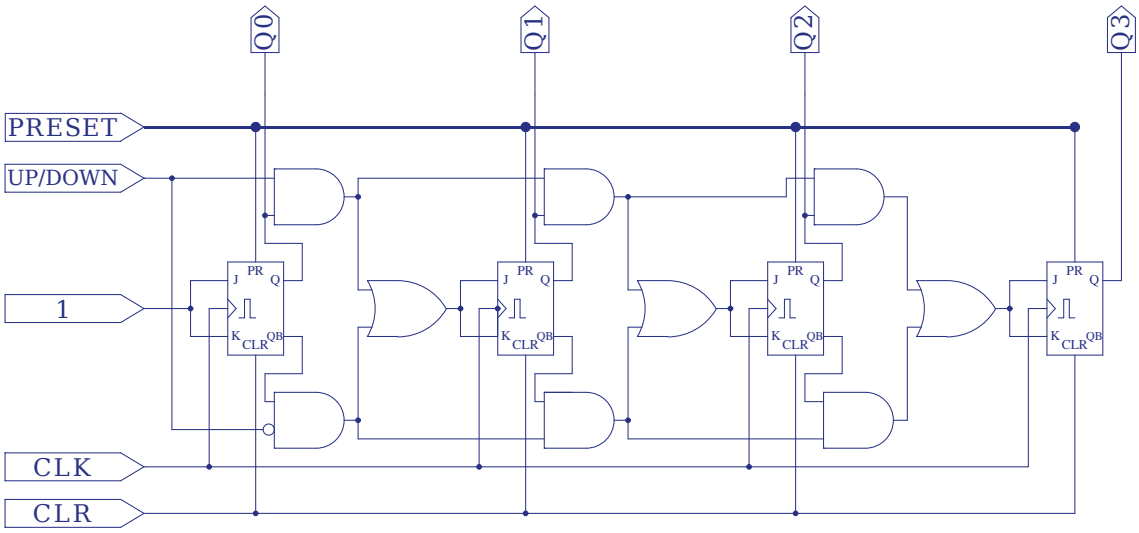


FIGURE B.2 – Architecture d'un compteur synchrone à bascules JK

Le périphérique DDS utilise aussi cette macro, et stocke également des données dans des blocs mémoire du FPGA. Pour ce faire, le code repose encore une fois sur des composants spécifiques au Spartan 6, des blocs RAM (présentés à la section 3.6.2). Il faudra donc modifier cette mémoire lors de l'utilisation de la DDS sur un autre composant.

Il existe sur Internet des codes VHDL pour des ROM [5]. Toutefois, aucun de ces codes ne peut utiliser au mieux les ressources disponibles de la cible visée (FPGA ou ASIC). Mais ils fonctionneront dans les deux cas.

Par conséquent, deux options se présentent :

5. Read Only Memory, mémoire en lecture seule

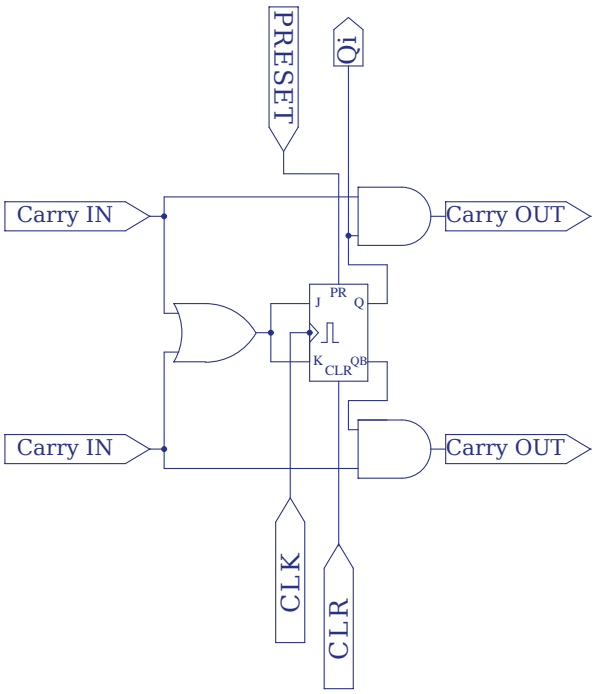


FIGURE B.3 – Étage d'un compteur universel à bascule JK

— soit réécrire la LUT avec un de ces codes génériques et ne pas chercher à optimiser le périphérique sur la cible choisie ;
— soit adapter cette LUT à chaque cible afin de tirer au mieux partie des ressources disponibles.

La première solution présente l'avantage de pouvoir fonctionner indépendamment sur FPGA ou sur ASIC. La seconde exploite mieux le composant choisi et peut résulter en une surface réduite et/ou une consommation réduite.

Bien que les blocs RAM du Spartan 6 permettent un accès en lecture et en écriture, la DDS développée ne permet pas de modifier le contenue de la LUT une fois le système programmé. Pour un FPGA, cela ne pose pas de problème puisqu'une nouvelle synthèse permet de modifier le contenue de la LUT si besoin.

En revanche, dans le cas d'un ASIC comme cible, une fois la LUT programmée, elle ne plus plus changer. Lors du portage de la DDS pour cet ASIC, il peut ainsi s'avérer préférable d'utiliser une mémoire également inscriptible, et accessible par le processeur, afin de pouvoir changer le contenu de la LUT lorsque la DDS ne fonctionne pas. Cela complique alors le composant car une mémoire inscriptible demande plus de ligne de code qu'une simple ROM. Si l'ASIC dispose de mémoire, il peut s'avérer plus intéressant dans ce cas d'utiliser cette mémoire disponible que de re-développer une mémoire pour la DDS.

En résumé, l'adaptation du périphérique DDS devra porter une attention particulière à la LUT.

Annexe C

Variance d'Allan

L'oscillateur parfait n'existe pas : un oscillateur réel présente toujours des défauts en phase et/ou en fréquence. Les outils d'analyse statistique habituels ne suffisent pas à caractériser les performances et défauts des oscillateurs. La moyenne et la variance notamment nécessitent une infinité d'échantillons et la moyenne n'apporte aucune information puisqu'il s'agit de la fréquence de l'oscillateur.

C.1 Estimation de la variance vraie

Pour un signal à mesurer qui comporte N échantillons y_k échantillonnées avec une période τ, avec N grand mais fini, la littérature [SM90] [VER12] montre qu'un estimateur de la variance s'exprime par :

$$\sigma^2(N) = \frac{1}{N-1} \sum_{i=1}^{N} \left(y_i - \frac{1}{N} \sum_{j=1}^{N} y_j \right) \tag{C.1}$$

Cet estimateur constitue lui aussi une variable aléatoire sur laquelle il est possible d'effectuer des calculs statistiques, pourvu que le nombre N reste très petit devant le nombre total d'échantillons. En outre, c'est un estimateur sans biais (c'est-à-dire qu'il tend vers la variance vraie pour N grand) pour le bruit blanc de fréquence [Rut78], mais pas pour les autres bruits. L'IEEE l'a ainsi retenu pour caractériser les oscillateurs, et a fixé N à 2. La variance à deux échantillons, ou variance d'Allan se définit alors par :

$$\begin{aligned} \sigma_y^2(\tau) &= \left\langle \sigma^2(2) \right\rangle \\ &= \left\langle \sum_{i=1}^{2} \left[y_i - \frac{1}{2}(y_1 + y_2) \right]^2 \right\rangle \\ \sigma_y^2(\tau) &= \frac{1}{2} \left\langle (y_2 - y_1)^2 \right\rangle \end{aligned} \tag{C.2}$$

Le tableau C.1 [VER12] donne la correspondance entre les densités spectrale de phase

des bruits du modèles en loi de puissance [VER17] et leurs variances d'Allan associées. Il montre notamment que les bruit blanc de phase et bruit de scintillation de phase ne se distinguent pas nettement à l'aide de la variance d'Allan : le premier est en τ^{-2} et le second en $\ln(\tau)\tau^{-2}$. Sur un tracé logarithmique de cette variance[1] en fonction de τ, une seule droite en τ^{-2} apparaîtra.

Type de bruit	**Origine du bruit**	**Densité spectrale de phase ($S_y(f)$)**
Bruit blanc de phase	Bruit blanc externe	$h_2 f^2$
Bruit de scintillation de phase	Bruit électronique	$h_1 f$
Bruit blanc de fréquence	Bruit thermique	$h0$
Bruit de scintillation de fréquence	Résonateur	$h_{-1} f^{-1}$
Marche aléatoire de fréquence	Environnement	$h_{-2} f_{-2}$

Type de bruit	**Variance d'Allan ($\sigma_y^2(\tau)$)**
Bruit blanc de phase	$\frac{3h_2 f_h}{4\pi^2\tau^2}$
Bruit de scintillation de phase	$\frac{(1{,}038+3\ln(2\pi f_h \tau))h_1}{4\pi^2}\tau^2$
Bruit blanc de fréquence	$\frac{h_0}{2\tau}$
Bruit de scintillation de fréquence	$2\ln(2)h_{-1}$
Marche aléatoire de fréquence	$\frac{2\pi^2 h_{-2}\tau}{3}$

Table C.1 – Variance d'Allan des bruits du modèle en lois de puissance (f_h est une fréquence de coupure haute)

C.2 Calcul pratique de la variance d'Allan

La section 3.2 a présenté la variance d'Allan du compteur de fréquence développé. Le compteur fournit les valeurs brutes C_k. La période instantanée $T_{k,n}$ du signal calculée sur n points se déduit des C_k par la formule :

$$T_{k,n} = \frac{C_k - C_{k-n}}{n} T_{clk}, n > 0 \tag{C.3}$$

Où T_{clk} est la période de l'horloge du compteur.

Les $T_{k,n}$ servent alors à calculer la variance d'Allan selon la formule (C.2). Par exemple, pour calculer $\sigma_T^2(T)$ (où T est la période du signal mesuré), il vient :

$$\sigma_T^2(T) = \frac{1}{2}\left\langle (T_{k+1,1} - T_{k,1})^2 \right\rangle$$

1. En réalité, on trace souvent l'écart-type et non la variance.

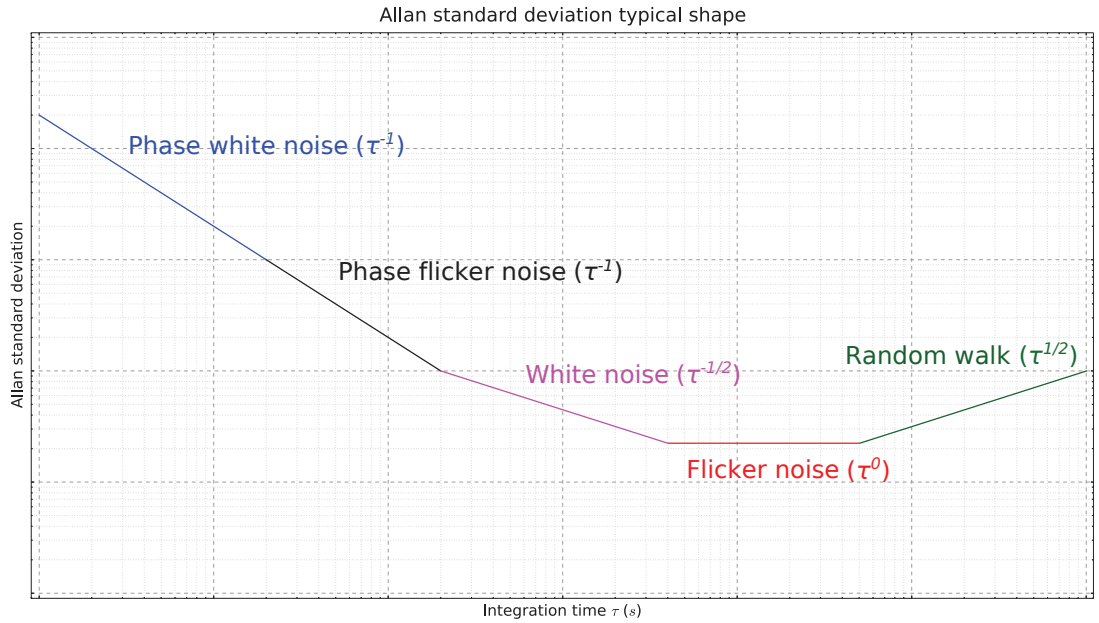


FIGURE C.1 – Forme caractéristique de l'écart type ($\sigma_y(\tau)$) d'ALLAN

De même, pour calculer $\sigma_T^2(2 \times T)$:

$$\sigma_T^2(T) = \frac{1}{2}\left\langle (T_{k+1,2} - T_{k,2})^2 \right\rangle$$

La figure C.2 présente ce calcul et la réalisation de la courbe sous forme graphique.

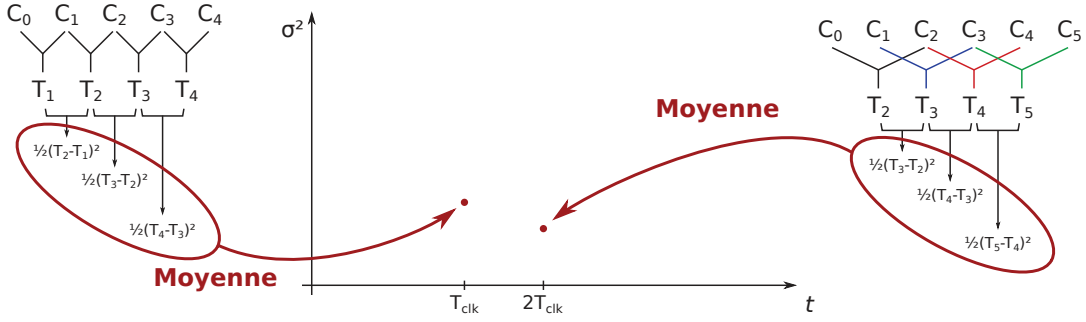


FIGURE C.2 – Représentation graphique du calcul de la variance d'ALLAN

Annexe D

Codes logiciels

D.1 Estimation du déphasage

Cette annexe présente un code possible pour le calcul de l'erreur de phase à partir des données du compteur, ainsi que détaillé dans la section 3.4.

```
// Power of 2 of the number of elements to store.
#define n_nbits        (5)
#define n              (1<<n_nbits)
#define n_mask         (n - 1)
#define next_idx(X)    ((X + 1) & n_mask)

int32_t phase_shift_err (uint32_t stimulus_msb, uint32_t stimulus_lsb,\
                         uint32_t response_msb, uint32_t response_lsb)
{
// Array storing the successive periods
static uint16_t periods[n] = { 0 };
// Index in the array
static uint8_t period_idx = 0;
// Period estimator
static uint16_t period_hat = 0;
// Variables given on the previous function call
static uint32_t stimulus_msb_last = 0;
static uint32_t stimulus_lsb_last = 0;
// Variable for various deltas
uint16_t period = 0;
// Expected response date.
uint32_t response_expected;
// Computed error signal
int32_t phase_shift_err = 0;
// Loop variable
uint8_t i;

// Substract the oldest period.
```

```
period_hat -= periods[next_idx(period_idx)];

/* The difference between dates should not be greater than 2^16 (case of
 * a 5 kHz VIG measured by a 300 MHz counter).
 * It means the period calculated here should not be greater than 1.
 * If it is, generate the maximun error signal. */
// Calculate MSB period. Ensure modulus is well handled.
period = (uint16_t) (stimulus_msb - stimulus_msb_last);
if (period > 1) {
    // Store maximum period and phase error.
    periods[period_idx] = 0xFFFF;
    phase_shift_err = 0x7FFF * n;
} else {
    // Calculate the real period and ensure modulus is well handled.
    period = (uint16_t) (stimulus_lsb - stimulus_lsb_last);
    // Store the computed period.
    periods[period_idx] = period;
}

// Calculate the new estimate.
period_hat += periods[period_idx];

// Manage the index.
period_idx = next_idx(period_idx);

// Remember the current values.
stimulus_msb_last = stimulus_msb;
stimulus_lsb_last = stimulus_lsb;

// Compute the phase error.
if (phase_shift_err != 0){
    // Frequency is too slow. Phase error already set.
} else {
/* Estimate the date the reponse should happen at
 * (i.e. stimulus + period_hat/4) and handle LSB overflow.
 * Don't forget period_hat is in n times the estimator.*/
response_expected = (uint32_t) (n * stimulus_lsb + period_hat>>2);
}

/* Compute phase error and ensure response is not way too late.
 * When there is an overflow in expected response LSB, greater effective
 * response LSB could mean two things:
 * - either response is in advance, in which case MSB shall be the same as
 *   stimulus MSB;
 * - either response is way too late, in which case MSB will have increased
 *   by at least 1. */
if ((response_expected < (uint32_t) (n * response_lsb))
```

```
    && (stimulus_msb != response_msb)) {
    // Response is way too late. Generate maxiumum error signal.
    phase_shift_err = 0x7FFF * n;
} else {
    // Compute phase shift error.
    phase_shift_err = (response_lsb - response_expected) * n;
}

return phase_shift_err;
}
```

Annexe E

Considériciennes

E.1 Transmission des données de comptage par liaison série

La section 3.1.5 a présenté l'architecture du compteur développé et la section 1.4, l'architecture général du micro-contrôleur implémenté sur FPGA. Le système communique par l'intermédiaire d'une liaison UART série. Un composant sur la carte SP605 se charge ensuite de convertir les données UART série pour les transmettre et les recevoir sur un bus USB[1]. Ce composant pourrait parfaitement s'intégrer au FPGA, mais le composant externe en question est directement câblé sur un port mini USB, inaccessible autrement.

Comme l'explique la section 3.2, un prescaler logiciel s'applique ensuite sur les données capturées : seule une valeur toutes les P périodes est transmise. La mise en trame suit ensuite un schéma fixe, quelque soit le nombre de signaux mesurés. Le tableau E.1 présente les trames envoyées par le FPGA lors des acquisitions réalisées afin d'obtenir les courbes présentées dans ce document. Ces trames se décomposent en trois parties et se présentent sous forme de valeurs hexadécimales. La première partie fournit des informations sur la ligne transmise :

- un numéro de ligne sur 2 octets (TX Idx) – quatre caractères hexadécimaux ;
- trois blocs de 2 octets (48 bits donc), qui donne la valeur du compteur au moment de la transmission, ou date courante – douze caractères hexadécimaux.

Les deux parties suivantes fournissent les données intéressantes, le résultat des mesures. Elles comportent chacune les informations suivantes :

- l'index du compteur de captures, c'est-à-dire le nombre de captures qui ont eu lieu, sur 16 bits (les valeurs sont modulo 65536) – à nouveau quatre caractères hexadécimaux ;
- la dernière valeur de compteur capturée, ou dernière date capturée – douze caractères hexadécimaux.

1. Universal Serial Bus

TX Idx	Date Compteur			Cap Idx	Capture 1			Cap Idx	Capture 2		
0094	0005	D45E	F7E3	68F9	0005	D45E	EBC1	9247	0005	D45E	DA53
0095	0005	D4AB	3121	6CE0	0005	D4AB	2341	962F	0005	D4AB	25A3
0096	0005	D4F7	7ABB	70C8	0005	D4F7	6E48	9A16	0005	D4F7	5D6B
0097	0005	D543	B1FB	74AF	0005	D543	A5CA	9DFE	0005	D543	A8BC
0098	0005	D58F	FCA9	7897	0005	D58F	F0DB	A1E5	0005	D58F	E08E
0099	0005	D5DC	3617	7C7E	0005	D5DC	2865	A5CD	0005	D5DC	2BE7
009A	0005	D628	80C5	8066	0005	D628	7376	A9B4	0005	D628	63B8
009B	0005	D674	B7E1	844D	0005	D674	AAFF	AD9C	0005	D674	AF11
009C	0005	D6C1	0283	8835	0005	D6C0	F611	B183	0005	D6C0	E6E3
009D	0005	D70D	399F	8C1C	0005	D70D	2D9A	B56B	0005	D70D	323C
009E	0005	D759	866F	9004	0005	D759	78A1	B952	0005	D759	6A03
009F	0005	D7A5	BDD3	93EB	0005	D7A5	B021	BD3A	0005	D7A5	B553
00A0	0005	D7F2	085D	97D3	0005	D7F1	FB28	C121	0005	D7F1	ED1A
00A1	0005	D83E	436F	9BBA	0005	D83E	32A8	C509	0005	D83E	386A
00A2	0005	D88A	8FA9	9FA2	0005	D88A	7DAF	C8F0	0005	D88A	7031
00A3	0005	D8D6	C7C7	A389	0005	D8D6	B52F	CCD8	0005	D8D6	BB81

TABLE E.1 – Trames brutes transmises par le FPGA

Les deux premiers champs fournissent des informations de débogage. D'une part, si les numéros de ligne ne se suivent pas, cela signifie qu'une ligne n'a pas été transmise, elle a été perdue quelque part dans la chaîne de transmission. D'autre part, la valeur du compteur est lue au moment de la transmission, ce qui signifie, compte-tenu du code logiciel utilisé, que cette date doit être postérieure aux dates des captures transmises sur cette ligne.

Pour les captures, les index assurent une robustesse dans les transmissions. En fait, les index indiquent le nombre de périodes comptées par le compteur depuis la dernière transmission. Dans le cas où seule une fréquence est mesurée (une seule capture branchée), la différence de ces index vaut P, la valeur du prescaler. Lorsque deux fréquences sont mesurées en même temps, la différence d'index ligne à ligne de la fréquence la plus élevée vaut P. Si une ligne vient à manquer (décelable par un index de ligne manquant), la différence entre l'index de capture d'une ligne et celui de la ligne précédente fournit le nombre de captures qui a eu lieu entre les deux lignes. Dans tous les cas, la connaissance de la fréquence de comptage, de la différence entre les deux valeurs capturées, et du nombre de captures fournit la fréquence du signal mesurée. Difficile de parler de fréquence instantanée puisqu'il s'agit en réalité d'une moyenne de la fréquence sur P périodes (avec P le prescaler). Concrètement, avec les valeurs du tableau E.1, pour calculer la fréquence entre la première et la seconde ligne pour la capture 1, il vient (une fois les valeurs hexadécimales converties en entiers) :

$$\begin{aligned} f_{CLK} &= 300\,\text{MHz} \\ f_k &= \frac{300\,10^6}{25042821953 - 25037827009} \times (27872 - 26873) \\ f_k &= 60000,67\,\text{Hz} \end{aligned}$$

Cette expression met ainsi en évidence l'importance de deux paramètres : la différence des dates ligne à ligne, et la différence des index de capture ligne à ligne. Et si une ligne

vient à manquer, cette même expression continue de fournir une mesure de fréquence correcte malgré la ligne manquante. Seul change le temps d'intégration, qui passe de P périodes (normalement d'une ligne à l'autre), à $x \times P$ si x lignes manquent entre deux lignes transmises.

E.2 Introduction au fonctionnement d'un processeur, au pipeline et estimation de la puissance de calcul

L'exécution d'une instruction processeur, par exemple la multiplication de deux entiers, se décompose en étapes séquentielles. Le processeur aura terminé l'exécution de l'instruction lorsqu'il aura complété la dernière étape de la séquence.

Un processeur peut par exemple découper ses instructions selon les trois étapes suivantes :

— chargement et décodage de l'instruction a exécuter – IF, instruction fetch ;
— exécution de l'instruction – EX, execute ;
— sauvegarde du résultat – WB, write back.

Chaque étape dure en règle générale 1 cycle d'horloge, sauf pour des opérations compliquées (division de flottants par exemple). Ainsi, pour exécuter 3 instructions, il faudra 9 cycles d'horloge. L'exécution est séquencée, comme le montre l'image E.1.



FIGURE E.1 – Exécution d'instructions dans un processeur sans pipeline

Or, le processeur n'utilise pas tous ses blocs simultanément. Lorsque le processeur se trouve à une étape, l'étape EX par exemple, alors les blocs logiques dédiés aux deux autres étapes, IF et WB, demeurent inactifs, inutilisés.

Par conséquent, rien n'empêche d'exécuter d'autres instructions dans les blocs inactifs. Autrement dit, avec la décomposition en trois étapes présentée plus haut, cela revient à exécuter trois instructions en parallèle, comme le montre la figure E.2. Ainsi, il ne faut plus que 5 cycles d'horloges pour exécuter 3 instructions.

En particulier, le processeur exécute une instruction par cycle d'horloge à partir de la troisième étape de la première instruction, c'est-à-dire la complétion de la première

instruction.

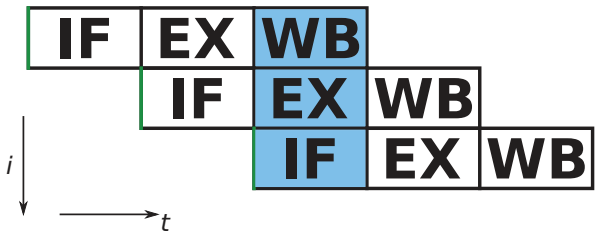


FIGURE E.2 – Exécution d'instructions dans un processeur avec un pipeline à 3 niveaux

Le nombre d'étape qui forment une instruction s'appelle profondeur du pipeline. Les blocs logiques dédiés à l'exécution de ces étapes dans le processeur, les étages de pipeline.

Toutefois, une profondeur de pipeline ne fixe pas nécessairement le nombre d'instructions exécutées en parallèle. Par exemple, un processeur peut proposer le pipeline suivant :

— IF – instruction fetch ;
— ID – instruction decode ;
— EX – execute ;
— EX – execute ;
— MEM – memory access ;
— WB – write back.

L'étape execute, celle qui exécute effectivement l'instruction demandée, se répète deux fois. Ce genre de pipeline permet ainsi de faire fonctionner les étapes annexes plus rapidement que l'exécution, qui peut avoir besoin de plus d'un cycle d'horloge pour se terminer, comme une instruction de division par exemple.

Dans ce cas, le processeur pourra exécuter 5 instructions en parallèle bien que la profondeur de son pipeline soit de 6.

Enfin, il arrive que certaines instructions bloquent ou rendent impossible ce parallélisme. Par exemple, si une instruction attend le résultat de l'instruction précédente : tant que le processeur n'a pas stocker ce résultat, l'instruction qui en dépend ne pourra pas y accéder. Autre exemple : certaines instructions peuvent nécessiter plus de cycle d'horloge que ceux prévus par le pipeline (très souvent le cas pour la division).

La profondeur du pipeline d'un processeur demeure arbitraire ; chaque processeur séquence ses instructions en fonction de ses étages de pipeline qui lui sont spécifiques.

Par conséquent, même si une même opération occupe autant de cycles d'horloge en théorie sur deux processeurs différents, le cumul et l'enchaînement des instructions rendront cette ressemblance totalement hors de propos. Et ce, à cause des pipelines différents

entre les deux processeurs. Ainsi, comparer le nombre d'instruction par seconde de processeurs avec des architectures différentes se révèle absurde puisque cette mesure demeure propre à chaque architecture de processeur.

Voilà pourquoi il existe d'autres échelles de comparaison, notamment le FLOPS (FLoating Point Operations Per Second) qui désigne le nombre d'opérations par seconde sur des nombre à virgule flottantes. Lorsque le processeur ne présente pas d'unité de calcul en virgule flottante, le Dhrystone MIPS (Mega Instructions Per Second) sert alors de référence, en particulier pour la puissance de calcul sur des entiers. Malgré ce que son nom mentionne, il ne s'agit pas d'une mesure absolue du nombre d'instructions par seconde qu'un processeur peut exécuter, mais d'une équivalence en MIPS que la référence aurait eu (le VAC 11/780). Souvent notée DMIPS, cette mesure rend possible la comparaison entre elles de différentes architectures de processeurs.

Bibliographie

[All87] D.W. Allan. Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 34(6) :647–654, 1987.

[AN01] F. Ayazi and K. Najafi. A harpss polysilicon vibrating ring gyroscope. *Microelectromechanical Systems, Journal of*, 10(2) :169–179, 2001.

[Bli49] Warren H. Bliss. Electronic digital counters. *Electrical Engineering*, 68(4) :309–314, 1949.

[Cola] Colibrys. Application note – accelerometer for ahrs application.

[Colb] Colibrys. Datasheet – ms9000.d / single axis analog accelerometer.

[Colc] Colibrys. Datasheet – rs9010.b / single axis tactical grade accelerometer.

[Cor31] G. G. Coriolis. Mémoire sur le principe des forces vives dans les mouvements relatifs des machines. *Journal de l'École Polytechnique*, XIII, February 1831.

[Cor35] G. G. Coriolis. Mémoire sur les équations du mouvement relatif des systèmes de corps. *Journal de l'École Polytechnique*, XV, 1835.

[Dev] Analog Devices. Dual-axis, high-g, imems® accelerometers.

[DGKLT12] M. Descharles, J. Guérard, H. Kokabi, and O. Le Traon. Closed-loop compensation of the cross-coupling error in a quartz Coriolis Vibrating Gyro. *Sensors & Acuators A :Physical*, 181 :25–32, 2012.

[EAM88] A.A. El-Amawy and A. Mirbod. An efficient software-controlled pll for low-frequency applications. *Industrial Electronics, IEEE Transactions on*, 35(2) :341–344, 1988.

[EML11] N. Erfani Majd and M. Lotfizad. A novel low power digitally controlled oscillator with improved linear operating range. *World Academy of Science, Engineering and Technology*, (51) :1267 – 1272, 2011.

[Erd66] Harold F. Erdley. Vibrating bar transducer, march 1966.

[GBKN91] P. Greiff, B. Boxenhorn, T. King, and L. Niles. Silicon monolithic micromechanical gyroscope. In *Solid-State Sensors and Actuators, 1991. Digest of Technical Papers, TRANSDUCERS '91., 1991 International Conference on*, pages 966–968, 1991.

[GIO13] Vincent GIORDANO. Génération de fréquence. *Techniques de l'ingénieur Métrologie temps-fréquence*, base documentaire : TIB415DUO.(ref. article : r682), 2013. fre.

[GK03] J. Geen and D. Krakauer. New imems® angular-rate-sensing gyroscope. Technical report, Analog Devices, 2003.

[HMS+06] R. Hopkins, J. Miola, R. Setterlund, B. Dow, and W. Sawyer. The silicon oscillating accelerometer : A high-performance mems accelerometer for precision navigation and strategic guidance applications. *TECHNOLOGY DIGEST*, page 4, 2006.

[Hon05] Honeywell. Accelerex® rba500 accelerometer, August 2005.

[IK77] Y. Inabe and K. Katoaka. An integrated jk flip-flop circuit. *Solid-State Circuits, IEEE Journal of*, 12(4) :403–406, 1977.

[Jac72] Earl D. Jacobs. Accelerometer, nov 1972.

[JLTM97] D. Janiaud, O. Le Traon, and S. Muller. Transducteur monolithique d'accélération, mar 1997.

[Joh05] S. Johansson. New frequency counting principle improves resolution. In *Frequency Control Symposium and Exposition, 2005. Proceedings of the 2005 IEEE International*, pages 8 pp.–, 2005.

[Kal04] J. Kalisz. Review of methods for time interval measurements with picosecond resolution. *Metrologia*, (41) :17 – 32, 2004.

[Lab] Jet Propulsion Laboratory. http://microdevices.jpl.nasa.gov/capabilities/nano-and-micro-systems/mems-gyroscope.php.

[LBHK] Daniel Lapadatu, Bjørn Blixhavn, Reidar Holm, and Terje Kvisterøy. Sar500 - a high-precision high-stability butterfly gyroscope with north seeking capability.

[LC81] W.C. Lindsey and Chak Ming Chie. A survey of digital phase-locked loops. *Proceedings of the IEEE*, 69(4) :410–431, 1981.

[LF98] R. Leonardson and S. Foote. Simmatm accelerometer for inertial guidance and navigation. In *Position Location and Navigation Symposium, IEEE 1998*, pages 152–160, Apr 1998.

[LF08] B. Le Foulgoc. *Évaluation du potentiel de performances micro-accéléromètres inertiels vibrants en silicium*. PhD thesis, Université Paris-Est, 2008.

[LG10] R. Levy and V. Gaudineau. Phase noise analysis and performance of the vibrating beam accelerometer. In *Frequency Control Symposium (FCS), 2010 IEEE International*, pages 511 –514, june 2010.

[Lil30] J.E. Lilienfeld. Method and apparatus for controlling electric currents, january 1930.

[Lil33] J.E. Lilienfeld. Device for controlling electric current, march 1933.

[LSWD08] Xiaodong Liu, Yanyan Shi, Meng Wang, and Jiaojiao Deng. Direct digital frequency synthesizer based on curve approximation. In *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*, pages 1 –4, april 2008.

[LTJMB98] O. Le Traon, D. Janiaud, S. Muller, and P. Bouniol. The via vibrating beam accelerometer : concept and performance. In *Position Location and Navigation Symposium, IEEE 1998*, pages 25 –29, apr 1998.

[MC01] A.M. Madni and L.E. Costlow. A third generation, highly monitored, micromachined quartz rate sensor for safety-critical vehicle stability control. In *Aerospace Conference, 2001, IEEE Proceedings.*, volume 5, pages 2523–2534 vol.5, 2001.

[MGM+12] B. Maréchal, J. Guérard, F. Mailly, P. Nouet, G. Papin, R. Levy, and O. Le Traon. Accurate frequency counting architecture for vibrating accelerometers on fpga. In *Design of Circuits and Integrated Systems 2012*, july 2012.

[MGM+14] B. Maréchal, J. Guérard, F. Mailly, P. Nouet, and O. Le Traon. Direct digital synthesiser (dds) design parameters optimisation for vibrating mems sensors. In *Design, Test, Integration & Packaging of MEMS/MOEMS 2014*, april 2014.

[MI68] P.M.C. Morse and K.U. Ingard. *Theoretical Acoustics*. International series in pure and applied physics. McGraw-Hill, 1968.

[Nis80] Errol P. Eer Nisse. Miniature quartz resonator force transducer, august 1980.

[NLJ60] I. Newton, T. Leseur, and F. Jacquier. *Philosophiae Naturalis Principia Mathematica*. Number vol. 2 in Philosophiae Naturalis Principia Mathematica. Philibert, 1760.

[Noo69] J. Noordanus. Frequency synthesizers-a survey of techniques. *Communication Technology, IEEE Transactions on*, 17(2) :257 –271, april 1969.

[NS87] H.T. Nicholas and H. Samueli. An analysis of the output spectrum of direct digital frequency synthesizers in the presence of phase-accumulator truncation. In *41st Annual Symposium on Frequency Control. 1987*, pages 495–502, 1987.

[NSK88] H.T. Nicholas, H. Samueli, and B. Kim. The optimization of direct digital frequency synthesizer performance in the presence of finite word length effects. In *Frequency Control Symposium, 1988., Proceedings of the 42nd Annual*, pages 357–363, Jun 1988.

[NST97] S. Norsworthy, R. Schreier, and G. Temes. *Delta-Sigma Data Converters :Theory, Design, and Simulation*. Wiley-IEEE Press, 1997.

[RAD12] Jean-Claude RADIX. Accéléromètres inertiels. *Techniques de l'ingénieur Grandeurs mécaniques*, base documentaire : TIB407DUO.(ref. article : r1930), 2012. fre.

[RJDdC95] T.V. Roszhart, H. Jerman, J. Drake, and C. de Cotiis. An inertial-grade, micromachined vibrating beam accelerometer. In *Solid-State Sensors and Actuators, 1995 and Eurosensors IX.. Transducers' 95. The 8th International Conference on*, volume 2, pages 656–658. IEEE, 1995.

[Rol12] I. Roland. *Gyro 3 axes AsGa*. PhD thesis, Université Paris-Sud, 2012.

[RS13a] Enrico RUBIOLA and Fabrice STHAL. Instrumentation temps-fréquence améliorations des techniques de comptage. *Techniques de l'ingénieur Métrologie temps-fréquence*, base documentaire : TIB415DUO.(ref. article : r687), 2013. fre.

[RS13b] Enrico RUBIOLA and Fabrice STHAL. Instrumentation temps-fréquence partie 1. *Techniques de l'ingénieur Métrologie temps-fréquence*, base documentaire : TIB415DUO.(ref. article : r686), 2013. fre.

[Rut78] J. Rutman. Characterization of phase and frequency instabilities in precision frequency sources : Fifteen years of progress. *Proceedings of the IEEE*, 66(9) :1048–1075, Sept 1978.

[SDD10] J.-M. Stauffer, O. Dietrich, and B. Dutoit. Rs9000, a novel mems accelerometer family for mil/aerospace and safety critical applications. In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pages 1–5, May 2010.

[Ser68] Norman R. Serra. Technical report on the quartz resonator digital accelerometer. In *AGARD (Nato Advisory Group for Aerospace Research & Development) Conference Proceedings #43, Inertial Navigation Systems & Components*, pages 487–516, 1968.

[Sho51] W. Shockley. Circuit element utilizing semiconductive material, september 1951.

[Sin06] A.K. Singh. *Digital Principles : Foundation of Circuit Design and Application*. New Age International (P) Limited, 2006.

[SM90] Petre Stoica and Randolph L Moses. On biased estimators and the unbiased cramér-rao lower bound. *Signal Processing*, 21(4) :349 – 350, 1990.

[Sys] Stanford Research Systems. Sr620 datasheet.

[TRG71] J. Tierney, C. Rader, and B. Gold. A digital frequency synthesizer. *Audio and Electroacoustics, IEEE Transactions on*, 19(1) :48 –57, March 1971.

[Van97] J. Vankka. Methods of mapping from phase to sine amplitude in direct digital synthesis. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 44(2) :526 –534, March 1997.

[VE] D. Van Ess. Signals from noise : Calculating delta-sigma snrs.

[VER12] François VERNOTTE. Stabilité temporelle et fréquentielle des oscillateurs : outils d'analyse. *Techniques de l'ingénieur Métrologie temps-fréquence*, base documentaire : TIB415DUO.(ref. article : r681), 2012. fre.

[VER17] François VERNOTTE. Stabilité temporelle et fréquentielle des oscillateurs : modèles. *Techniques de l'ingénieur Métrologie temps-fréquence*, base documentaire : TIB415DUO.(ref. article : r680), 2017. fre.

[Xila] Xilinx. Spartan-6 fpga block ram resources.

[Xilb] Xilinx. Spartan-6 fpga clocking resources.

[Xilc] Xilinx. Spartan-6 libraries guide for hdl designs.

[XWS10] Dunzhu Xia, Shourong Wang, and Xia Sheng. Silicon microgyroscope based on spll and lmsd. In *Electrical and Control Engineering (ICECE), 2010 International Conference on*, pages 1896–1899, 2010.

Table des figures

1.1 Schéma de principe des accéléromètre à technologie Q-Flex® (d'après [Jac72]) 9
1.2 Technologie iMEMS pour les accéléromètres [Dev] 10
1.3 Schéma de principe de l'accéléromètre SOA du Draper Laboratory (d'apres [HMS+06]) 13
1.4 Une cellule VIA 15
1.5 Variance d'Allan de l'accéléromètre VIA (normalisée par rapport à la fréquence de résonance au repos) 16
1.6 Illustration de l'effet Sagnac : deux signaux lumineux qui partent dans des directions opposées présentent une différence de marche lorsqu'ils se rejoignent si le mobile en leur centre entre en rotation 17
1.7 La technologie iMEMS® 19
1.8 Structure et principe du gyromètre de Sensonor 20
1.9 Structure du QRS de Systron Doner 21
1.10 Une cellule VIG 22
1.11 Mode de vibration pilote et détecteur du diapason du VIG 22
1.12 Schéma bloc d'un oscillateur auto-entretenu 23
1.13 Schéma blocs d'un oscillateur numérique avec un résonateur analogique . . 25
1.14 Principe de fonctionnement d'une DDS 26
1.15 Architecture numérique retenue pour les capteurs vibrants de l'Onera (les amplificateurs de charges ne sont pas représentés, ils se situent juste avant et juste après le capteur) . 30

2.1 Comptage simple d'un signal sinusoïdal 33
2.2 Comptage réciproque d'un signal sinusoïdal 34
2.3 Exemple d'une chaîne à retard à bascules D. Plutôt que d'avoir une sortie toutes les portes, il est possible d'en avoir toutes les n portes. 37
2.4 Glissement du gate de comptage réciproque en marquage temporel 38
2.5 Différents intervalles de mesure à l'aide des mêmes données issues du marquage temporel 39
2.6 Forme caractéristique de la variance d'Allan et identification des contributeurs majoritaires 41

2.7 Comptage de fréquence par marquage temporel. On relève les dates, exprimées en nombre de périodes d'horloge, des fronts montants du signal d'entrée. La fréquence du signal est donnée par la pente de la droite de régression. 42
2.8 Schéma de principe d'une DDS 45
2.9 Cas limite du théorème de NYQUIST-SHANNON où $f_{clk} = 2f_{out}$. Selon la phase de l'échantillonnage, la sortie peut demeurer nulle (figure 2.9a) ou former un créneau (figure 2.9b). 51
2.10 Spectre de sortie d'un signal à 60 kHz avec une fréquence d'échantillonnage de 120 kHz 52
2.11 Sortie d'une DDS dans un cas non idéal avec un taux de sur-échantillonnage faible (2 ici) 53
2.12 Sortie d'une DDS dans un cas non idéal avec un taux de sur-échantillonnage élevé 53
2.13 Mise en évidence des symétries (zoom de la figure 2.12 autour de $f = 60$ MHz) 54
2.14 Spectres de sorties de deux DDS qui ne diffèrent que par leur nombre de bits sur l'accumulateur de phase 54
2.15 Zoom sur les spectres de la figure 2.14 55
2.16 Représentation de l'erreur de phase sur le cercle de phase 57
2.17 Allure du signal d'erreur de phase causé par la troncature 58
2.18 Motif de mot de commande W qui ne donne pas lieu à troncature de phase 59
2.19 Spectre de sortie en l'absence de troncature de phase : fondamental parfait (en haut, à gauche, amplitude $\frac{1}{2}$), bruit spectral confondu avec le bruit de calcul, et raies parasites à la fréquence de NYQUIST-SHANNON. 59
2.20 Motif de mot de commande qui génère les pires spurs 60
2.21 Spectres de sortie dans un pire cas (LUT de 14 et 18 bits en rouge et bleu respectivement) 61
2.22 Spectres de sortie dans des cas quelconques (LUT de 14 et 18 bits en rouge et bleu respectivement) 63
2.23 Représentations spectrales de l'équation (2.31) 64
2.24 Probabilité d'erreur de quantification d'un DAC 67
2.25 Représentation des erreurs de non-linéarité d'un DAC 69
2.26 Erreurs de linéarité d'un DAC 70
2.27 Comparaison des sorties d'un DAC conventionnel et d'un convertisseur Σ/Δ par rapport à une sinusoïde de référence 71
2.28 Algorithme d'un DAC Σ/Δ 71
2.29 Schéma bloc d'un modulateur Σ/Δ 72
2.30 Densité spectrale de bruit d'un DAC Σ/Δ du premier ordre (normalisée par rapport à la fréquence d'échantillonnage et au pas de quantification. . . 73

2.31 Sortie d'une DDS avec un DAC de 4 bits et un convertisseur Σ/Δ 75
2.32 Spectre du signal de la figure 2.31 76
2.33 Spectre du signal de la figure 2.31 en échelle logarithmique. Le fondamental de la sinusoïde est présent à l'amplitude 3×10^4, bien au dessus du bruit et des spurs. 77

3.1 Organisation d'un *Configurable Logic Block* 81
3.2 Architecture simplifiée d'un bloc DSP48A1 83
3.3 Câblage d'un additionneur en compteur 83
3.4 Fonctionnement d'un monostable non réenclenchable 85
3.5 Architecture du périphérique de comptage de fréquence développé 86
3.6 Variance d'Allan obtenue lors de la mesure d'un signal à 1 kHz (normalisée par rapport à F_0) 88
3.7 Variance d'Allan d'un Agilent 33522A obtenue avec le FPGA pour une fréquence de comptage de 300 MHz 90
3.8 Variance d'Allan d'une mesure différentielle en rouge, à partir des mesures simples en bleu et vert. 90
3.9 Erreur des différences de date à la moyenne ($P = 1000$) 91
3.10 Erreur des différences de date à la moyenne brute et extraction d'une fluctuation lente 93
3.11 Erreur des différences de date à la moyenne sur 300 points environ ($P = 1000$) 94
3.12 Schéma de principe de l'expérience de discrimination de la source des imperfections 94
3.13 Illustration du biais d'horloge, «clock skew» en Anglais : les temps d'arrivée sur les composants A et B diffèrent 96
3.14 Représentation du déphasage entre le signal d'excitation et la réponse du capteur 98
3.15 Câblage d'un additionneur en accumulateur 106
3.16 Architecture simplifiée d'une rangée du FPGA. Image tirée de [Xila] 107
3.17 Symétries dans une sinusoïde 108
3.18 DDS avec table demie-onde et utilisation du MSB de l'accumulateur de phase pour former le MSB du DAC 109
3.19 Chevauchement des 16 bits intermédiaires pour la programmation du mot de commande 112
3.20 Capture d'écrans de mesures avec l'oscilloscope 115
3.21 Spectre de sortie mesuré de la DDS, avant filtrage, pour trois fréquences d'horloge : 25 MHz (rouge), 50 MHz (vert) et 100 MHz (bleu) 116
3.22 Amplitudes relatives des harmoniques en fonction de la fréquence d'horloge de la DDS 117

4.1 Prototype d'un accéléromètre VIA 121
4.2 Signaux accélérométriques différentiels 123
4.3 Variance d'Allan pour la cellule accélérométrique. (rouge, vert) variance pour les cellules séparées. (bleu) variance du signal différentiel. 124
4.4 Prototype d'un gyromètre VIG 125
4.5 Architecture électronique du gyromètre vibrant. Les propriétés piézoélectriques du quartz sont utilisées aussi bien pour l'excitation (transformation de la tension appliquée en force exercée sur le résonateur pilote), que pour la détection (transformation des déformations du résonateur en charges électriques). 126
4.6 Signal temporel de sortie du gyromètre au repos 128
4.7 Variance d'Allan de la sortie du gyromètre. Variance calculée à partir des mesures (bleu). Asymptote pour les temps d'intégration courts (clair). . . . 129

B.1 Architecture d'un périphérique AXI développés avec les outils Xilinx 148
B.2 Architecture d'un compteur synchrone à bascules JK 149
B.3 Étage d'un compteur universel à bascule JK 150

C.1 Forme caractéristique de l'écart type ($\sigma_y(\tau)$) d'Allan 153
C.2 Représentation graphique du calcul de la variance d'Allan 153

E.1 Exécution d'instructions dans un processeur sans pipeline 161
E.2 Exécution d'instructions dans un processeur avec un pipeline à 3 niveaux . 162

Liste des tableaux

1.1 Notations employées dans les équations mécaniques 6
1.2 Performances des QA3000-030 d'Honeywell et ADXL278 d'Analog Devices 10
1.3 Performances du RS9000 de Colybris [SDD10] [Colc] [Colb] [Cola] 11
1.4 Exemples de caractéristiques d'accéléromètres VBA 12
1.5 Performances des accéléromètres SOA et MVBA 13
1.6 Performances de l'accéléromètre en quartz d'Honeywell 14
1.7 Caractéristique d'une cellule VIA 16
1.8 Performances de quelques gyromètres MEMS du marché 19
1.9 Performances du VIG de l'Onera 21

2.1 Dénomination des bruits pour un gyromètre 40
2.2 Signification des variables employées dans ce chapitre 42
2.3 Notation des paramètres d'une DDS 45
2.4 Formation d'un signal à la même fréquence à partir de deux DDS différentes 55
2.5 Paramètres de la DDS de la figure 2.16 56
2.6 Erreur de phase au cours du fonctionnement d'une DDS 58
2.7 Paramètre de la DDS de la figure 2.21 61
2.8 Paramètres de la DDS des figures 2.31, 2.32 et 2.33 74

3.1 Caractéristiques des slices en fonction de leur type. 82
3.2 Nombre de périodes d'horloge par période du signal à mesurer 84
3.3 Performances d'un accéléromètre à cellules VIA (sources [LG10]) 86
3.4 Signification des variables employées dans ce chapitre 99
3.5 Notation de l'équation la correction PID 104
3.6 Nombre d'adresses et tailles des données stockées dans un BRAM de 16 KiB 107
3.7 Caractéristiques d'une cellule VIA 111
3.8 Paramètre de la DDS à concevoir pour satisfaire les performances des capteurs 113
3.9 Paramètre de la DDS à concevoir pour satisfaire les performances des capteurs 114

4.1 Paramètres des accéléromètres de la cellule DIVA utilisée 122
4.2 Paramètres de la cellule gyrométrique VIG utilisée 126

4.3 Rappel des paramètres de performances de la cellule VIG utilisée pour les expériences. 128
4.4 Resumé des performances obtenues avec le prototype gyrométrique 130

C.1 Variance d'Allan des bruits du modèle en lois de puissance (f_h est une fréquence de coupure haute) 152

E.1 Trames brutes transmises par le FPGA 160

Résumé de la thèse

L'Onera développe des capteurs inertiels MEMS vibrants avec des performances qui peuvent intéresser des applications spatiales. Les électroniques analogiques traditionnellement associées ne sont a priori pas limitantes par rapport aux performances physiques des capteurs. En revanche, elles se montrent encombrantes, non reconfigurables, et ne délivrent pas les grandeurs mesurées sous forme numérique à l'ordinateur de bord. En outre, dans le cadre d'une utilisation spatiale, elles sont sujettes à dépendance et à obsolescence : le remplacement d'un composant implique une nouvelle qualification.

Cette thèse propose une nouvelle architecture numérique générique, limitant au maximum les composants analogiques nécessaires. Les travaux portent principalement sur deux capteurs développés par l'Onera : l'accéléromètre à lame vibrante VIA et le gyromètre vibrant à effet Coriolis VIG, mais sont justement transposables à d'autres familles. Une première fonction clé identifiée est la datation d'évènements pour la mesure de fréquence et de phase, une seconde concerne la synthèse numérique directe de fréquence pour le pilotage de résonateurs, et une troisième traite la génération de signaux sinusoïdaux purs à partir des trains binaires délivrés par le système numérique. Ces fonctions sont réalisées sous forme de périphériques numériques autour d'un processeur embarqué, le tout synthétisé sur composant programmable FPGA.

Le manuscrit débute par un rappel des lois physiques et des technologies de mesure inertielle, suivi d'une revue des oscillateurs, analogiques et numériques, afin de déterminer l'architecture numérique souhaitée. Le chapitre suivant aborde de façon théorique les fonctions numériques envisagées, et en détermine les éléments de performance. La mise en oeuvre de ces fonctions et les premiers résultats expérimentaux sont ensuite présentés d'abord au niveau de la fonction seule, et enfin dans une architecture complète incluant le capteur et le logiciel embarqué, pour fournir de vraies mesures inertielles. Ces résultats encouragent le déploiement d'électronique numérique dans les prochaines générations de capteurs.

Thesis summary

Onera has been developing vibrating inertial MEMS sensors with performances good enough for space uses. Associated conventional analog electronics are not limiting the physical performances of the sensors. They are, however, bulky, not reconfigurable, and do not deliver digital measurements to the on-board computer. Furthermore, when used for space applications, they have to cope with dependency and obsolescence requiring a new qualification when any part is changed.

This thesis offers a new digital generic architecture with as few analog parts as possible. Work has been focused on two sensors developed by Onera : the VIA, a vibrating beam accelerometer, and the VIG, a Coriolis vibrating gyro, but can address other sensors. A first digital function identified is event timestamping for frequency and phase measurements ; a second key function is the direct digital synthesis of the oscillating sensors driving signal ; the third one generates pure sine signals from binary sequences output from the digital platform. These function are implemented as peripherals of an embedded processor on a FPGA.

This dissertation firstly reminds physical laws and technologies of inertial measurements, followed by a quick review of oscillators, analog and digital, in order to introduce the chosen digital architecture. A following chapter studies the theory of the digital functions considered and identifies their performances. Afterwards, realisations and first experimental results are exposed, at a function level first, at a global level then, with the sensor and the embedded software to provide real inertial readings. The results gathered boost the idea of deploying digital electronics in future sensor releases.