



HAL
open science

Conception et réalisation de circuits de génération de fréquence en technologie FDSOI 28nm

Alexandre Fonseca

► **To cite this version:**

Alexandre Fonseca. Conception et réalisation de circuits de génération de fréquence en technologie FDSOI 28nm. Autre. Université Nice Sophia Antipolis, 2015. Français. NNT : 2015NICE4100 . tel-01284746

HAL Id: tel-01284746

<https://theses.hal.science/tel-01284746>

Submitted on 8 Mar 2016

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.

**UNIVERSITE NICE SOPHIA ANTIPOLIS
POLYTECH'NICE-SOPHIA**

**École Doctorale des Sciences et Technologies de
l'Information et de la Communication**

Électronique pour Objets Connectés

THESE

Pour obtenir le titre de
Docteur en Sciences spécialité Électronique
de l'Université Nice Sophia Antipolis

présentée et soutenue par

Alexandre FONSECA

**Conception et réalisation de circuits de génération de fréquences
en technologie FDSOI 28nm**

Thèse dirigée par Philippe LORENZINI et Gilles JACQUEMOD
Soutenue le 2 Décembre 2015

Jury :

H. BARTHELEMY	Rapporteur	Professeur, Université du Sud Toulon Var
Y. DEVAL	Rapporteur	Professeur, ENSEIRB Bordeaux
P. LORENZINI	Directeur	Professeur, UNS Sophia Antipolis
G. JACQUEMOD	Co-directeur	Professeur, UNS Sophia Antipolis
E. de FOUCAULD	Encadrant	Ingénieur, CEA-LETI Grenoble
P. NOUET	Examineur	Professeur, Polytech'Montpellier
R.M. SAUVAGE	Invitée	Ingénieur, DGA Bagneux

Avant-propos

Le travail présenté dans ce mémoire a été effectué au sein du Laboratoire d'Architectures et Infrastructures Radiofréquences (LAIR) du Département Architecture Conception et Logiciels Embarqués (DACLE) du Laboratoire d'Electronique et de Technologie de l'Information (LETI) du Département de Recherche technologique (DRT) du Commissariat à l'Energie Atomique (CEA), situé sur le site de Grenoble.

Je remercie en premier lieu Monsieur Marc Belleville et Monsieur Pierre Vincent pour m'avoir accueilli dans le département/laboratoire.

Je remercie également Monsieur Philippe Lorenzini et Monsieur Gilles Jacquemod, mes co-directeurs de thèse pour m'avoir proposé ce sujet passionnant, m'avoir conseillé techniquement tout au long de ma thèse et m'avoir aidé à rédiger ce mémoire.

Je remercie vivement Monsieur Emeric de Foucauld, mon tuteur désigné, et Monsieur Gérard Billiot, un expert de la microélectronique radiofréquence, pour m'avoir guidé et conseillé avec une disponibilité exemplaire tout au long de mes trois années de thèse.

J'adresse également mes remerciements à Messieurs H. Barthélémy, Y. Deval, pour m'avoir fait l'honneur de présider, de rapporter et d'examiner ce travail.

Je tiens à exprimer toute ma sympathie aux toute l'équipe du LAIR qui m'a accompagnée au cours de ces trois années de thèse et plus particulièrement à J. Prouvée, O. Rozeau, D. Morche, M. Pelissier, G. Waltener, C.Jany ainsi qu'à mes collègues thésards M. Verdy, A. Ratiu, O.Martin, P. Moron Guerra et G. Tant.

Enfin, il me tient à cœur de remercier ma femme, Mme Tânia Emanuelle Mende De Almeida, mon meilleur ami Pierre Pétolet, et ma mère pour m'avoir soutenu et aidé dans les moments difficiles.

Sommaire

Avant-propos.....	1
Sommaire	ii
Glossaire.....	1
Introduction générale	3
1. Des objets communicants vers les objets connectés	3
2. Travaux de Thèse	5
Chapitre I - Standard, technologie et architecture pour l'autonomie	7
1. Introduction	7
2. Le standard Bluetooth Low Energy	7
2.1. Bluetooth Low Energy et Internet des Objets	7
2.1.1. L'émergence de l'IOT.....	7
2.1.2. Avantages du BLE pour l'IOT	8
2.1.3. Transaction BLE type	10
2.1.4. Spécifications du BLE.....	11
2.2. Communications sans fil.....	12
2.2.1. Bases de fonctionnement d'un lien radio	12
2.2.2. Composants d'un module radio	17
2.2.3. Influence des blocs sur les performances	18
2.3. Synthèse de fréquences pour le BLE	19
2.3.1. Caractéristiques d'une synthèse	19
2.3.2. Transposition en fréquence non idéale.....	22
2.3.3. Résumé des spécifications pour la synthèse	22
2.4. Conclusion : L'apport du BLE	22
3. La technologie CMOS FDSOI	23
3.1. Les enjeux de l'évolution microélectronique	23
3.2. Evolution de la technologie microélectronique	25
3.2.1. La fin de la loi de Moore ?	25
3.2.2. Limitations des transistors MOS Bulk dus aux nœuds agressifs	27
3.2.3. La technologie FDSOI.....	30
3.2.4. La variabilité de la tension de seuil dans les nœuds agressifs	32
3.2.5. Méthodes de simulation de la variabilité.....	33
3.2.6. Méthodes de correction de la variabilité	35
3.3. Conclusion	36
4. Architectures de synthèse de fréquences	36

4.1.	Introduction.....	36
4.2.	Architectures de synthèse	37
4.2.1.	Types d’asservissement.....	37
4.2.2.	Types de PLL	39
4.2.3.	Types d’oscillateurs	41
4.2.4.	PLL Entière Vs Fractionnaire	43
4.2.5.	Diviseurs fractionnaire : modulo variable Vs commutation de phases.....	45
4.3.	La PLL à diviseur fractionnaire sur phase (FPD).....	48
4.3.1.	Différents chemins menant à la même architecture	48
4.3.2.	Historique et état de l’art de la PLL FPD.....	48
4.3.3.	Création Vs réutilisation des phases du VCRO	51
4.3.4.	DPLL à TDC implicite Vs TDC explicite	52
4.3.5.	Diviseur à commutation Vs resynchronisation.....	52
4.3.6.	Exemple d’implémentation FPD.....	53
4.3.7.	Amélioration de la base de temps.....	54
4.4.	Conclusion	55
5.	Conclusion.....	55
Chapitre II – Modélisation système		58
1.	Introduction.....	58
1.1.	Méthodologie de modélisation	58
2.	Etude du bloc VCRO - Diviseur.....	59
2.1.1.	Diviseur par commutation et diviseur par resynchronisation.....	59
2.2.	Modèles	61
2.2.1.	VCRO à phases multiples.....	61
2.2.2.	Diviseur entier	62
2.2.3.	Multiplexeur à commutation	63
2.2.4.	Multiplexeur à resynchronisation	64
2.2.5.	Contrôleur de division fractionnaire	66
2.3.	Résultats	67
2.3.1.	Simulation temporelle des phases du VCO	67
2.3.2.	Simulation du FPD à commutation de phases.....	68
2.3.3.	Simulation du FPD à resynchronisation par phase.....	69
2.4.	Conclusion	71
3.	Etude d’une PLL à base de diviseur FPD	71
3.1.	Introduction.....	71
3.2.	Modèles	72
3.2.1.	Détecteur de Fréquence et Phase (PFD)	72
3.2.2.	Pompe de Charge (CP).....	73

3.2.3.	Filtre Passe-Bas (LPF).....	74
3.3.	Simulation temporelle de la PLL fractionnaire	74
3.4.	Conclusion	75
4.	Etude du bruit de phase.....	76
4.1.	Introduction.....	76
4.2.	Extraction du bruit de chaque contributeur.....	77
4.3.	Résultats	79
4.3.1.	Simulation dans le domaine temporel	79
4.3.2.	Simulation dans le domaine fréquentiel	80
4.4.	Conclusion	81
5.	Etude de génération des spurious	82
5.1.	Introduction.....	82
5.2.	Modèles	83
5.2.1.	Méthode temporelle	83
5.2.2.	Méthode fréquentielle	85
5.3.	Résultats	85
5.3.1.	Simulation temporelle.....	85
5.3.2.	Simulation fréquentielle.....	86
5.4.	Conclusion	87
6.	Etude de calibration des phases	88
6.1.	Introduction.....	88
6.2.	Architectures de calibration	88
6.2.1.	Calibration par comparateur Bang-Bang, Filtre numérique et DAC.....	89
6.2.2.	Calibration par Vtune distribué par multiplexeur analogique	89
6.2.3.	Conclusion sur l'architecture de calibration.....	90
6.3.	Modélisation.....	90
6.4.	Résultats de calibration progressive	91
6.5.	Conclusion	93
7.	Conclusion.....	93
Chapitre III – Conception, réalisation et test de VCRO		96
1.	Introduction	96
2.	Spécifications	97
3.	Implémentation transistor, Layout & Simulation	98
3.1.	VCRO single ended à 6 phases (3 inverseurs).....	98
3.1.1.	Introduction.....	98
3.1.2.	Implémentation.....	98
3.1.3.	Layout	101
3.1.4.	Simulation.....	104

3.2.	VCRO single ended à 30 phases (15 inverseurs).....	106
3.2.1.	Implémentation.....	106
3.2.2.	Layout.....	107
3.2.3.	Simulation.....	108
3.3.	VCRO différentiel à 16 phases (8 cellules).....	110
3.3.1.	Implémentation.....	110
3.3.2.	Layout.....	111
3.3.3.	Simulation.....	112
3.4.	VCRO différentiel à 16 phases à 4x Fout.....	114
3.4.1.	Implémentation.....	114
3.4.2.	Layout.....	117
3.4.3.	Simulation.....	119
4.	Réalisation.....	121
4.1.	Dessin des masques.....	121
4.2.	Simulations post layout.....	121
4.3.	Fabrication.....	123
4.4.	Récapitulatif et Conclusion.....	123
5.	Mesures et tests des VCRO.....	124
5.1.	Introduction.....	124
5.2.	Mesure à fréquence nominale.....	124
5.3.	Mesures de la fréquence d'oscillation des VCRO.....	125
5.4.	Mesure du bruit de phase des VCRO.....	127
5.5.	Mesure du bruit de phase en boucle fermée.....	129
5.6.	Conclusion.....	130
6.	Conclusion.....	132
Chapitre IV – Etude de la PLL à resynchronisation de phases.....		134
1.	Introduction.....	134
2.	Diviseur fractionnaire.....	134
2.1.	Introduction.....	134
2.2.	Architecture.....	135
2.3.	Spécifications.....	135
2.4.	Implémentation au niveau transistor.....	135
2.4.1.	Calculateur.....	135
2.4.2.	Diviseur Entier.....	137
2.4.3.	Multiplexeur à resynchronisation.....	139
2.4.4.	Conclusion.....	141
2.5.	Simulation.....	143
2.6.	Conclusion.....	145

3.	Détecteur de phase et filtre.....	145
3.1.	Introduction.....	145
3.2.	Spécifications.....	146
3.3.	Implémentation.....	146
3.3.1.	Comparateur de phases	146
3.3.2.	Pompe de charge.....	147
3.3.3.	Filtre de boucle.....	149
3.4.	Simulation.....	149
3.4.1.	Comparateur de phases	150
3.4.2.	Pompe de charge.....	151
3.4.3.	Filtre de boucle.....	152
3.5.	Conclusion	152
4.	Buffer 50Ω	152
4.1.	Introduction.....	152
4.2.	Spécifications.....	153
4.3.	Implémentation.....	153
4.4.	Layout	153
4.5.	Simulation.....	154
5.	PLL.....	155
5.1.	Introduction.....	155
5.2.	Simulation.....	155
5.3.	Conclusion	156
6.	Perspectives.....	156
6.1.	Introduction.....	156
6.2.	Logique complémentaire.....	157
6.3.	Oscillateur en anneau en logique complémentaire	159
7.	Conclusion.....	159
	Conclusion Générale	161
	Publications.....	163
	Références.....	164
	Annexes.....	167
1.	Modèle de VCRO à phases.....	167
2.	Modèle de diviseur entier.....	168
3.	Modèle du multiplexeur à commutation.....	169
4.	Modèle du multiplexeur à resynchronisation	170
5.	Modèle du contrôleur.....	171
6.	Modèle de détecteur de phase.....	172
7.	Modèle de pompe de charge.....	173

8.	Fonction d'estimation de la DSP du bruit de phase à partir des mesures des périodes	174
9.	Modélisation du bruit de phase et des spurious à partir du mismatch.....	175
10.	Fonction de génération de mismatch des phases en fonction du jitter RMS.....	179

Glossaire

ACK	Acknowledgement
BER	Bit Error Rate
BG	Back Gate (Grille arrière des transistors FDSOI)
BLE	Bluetooth Low Energy
Box	Buried Oxide - oxyde enterré
BW	Bandwidth
CML	Current Mode Logic
CMOS	Complementary Metal Oxide Semiconductor
CP	Charge Pump
CSR	Cambridge Silicon Radio
DAC	Digital to Analog Converter
DCVSL	Differential Cascode Voltage-Switch Logic
DCVSLR	Differential Cascode Voltage-Switch Logic with series Resistor
DF	Digital Filter - Filtre numérique
DLL	Delay Locked Loop
DSP	Densité spectrale de puissance
DTC	Digital to Time Converter - Générateur de délai ajustable numériquement
FDSOI	Fully Depleted Silicon on Insulator - complètement appauvri de silicium sur isolant
FOM	Figure of Merit - figure de mérite
FPD	Fractional Phase Divider - Diviseur fractionnaire à phases
GFSK	Gaussian Frequency Shift Keying
ILO	Injection Locked Oscillator
IOT	Internet des Objets
IP	Intellectual Property
ISM	Industrial, Scientific and Medical
LPF	Low Pass Filter
MCS	MonteCarlo Simulation
PFD	Phase Frequency Detector
PLL	Phase Locked Loop
PLS	Post Layout Simulation
PN	Phase Noise
SNR	Signal to noise Ratio
TDC	Time to Digital Converter
TF	Transformée de Fourier
UTBB	Ultra Thin Body and BOx - canal et oxyde ultra fin
VCO	Voltage Controlled Oscillator - oscillateur contrôlable en tension
VCRO	Voltage Controlled Ring Oscillator - oscillateur en anneau contrôlable en tension

Introduction générale

1.Des objets communicants vers les objets connectés

Depuis plusieurs décennies, les objets communicants ont envahi notre quotidien et le nombre de données transmises, associées à de nombreux standards de communication sans fil, a également explosé. De nos jours, les objets sont connectés entre eux, sans parfois d'intervention humaine, et connectés à Internet ; on parle alors d'Internet des objets (IOT : Internet of Things). Toutefois, parmi toutes les données disponibles sur internet, qui sont au nombre inimaginable de 50 pétaoctets (50 000 téraoctets), quasiment toutes ont été créées ou insérées par un humain. Le problème est que l'homme est limité en temps, en attention et en précision, et devrait se consacrer à d'autres activités, plus basiques ou plus créatives, que seul un humain peut effectuer.

C'est de cette constatation que l'IOT fut imaginé, pour que les objets du quotidien puissent fournir des données à Internet pour les usagers (cf. Figure 1). Ces objets connectés, permettent l'utilisation intelligente des données collectées automatiquement, qu'elles soient sécuritaires, médicales, alimentaires, vestimentaires, météorologiques, de trafic, sociales, professionnelles ou commerciales. Ainsi la capture, le traitement des données et la prise de certaines décisions sont automatisés.

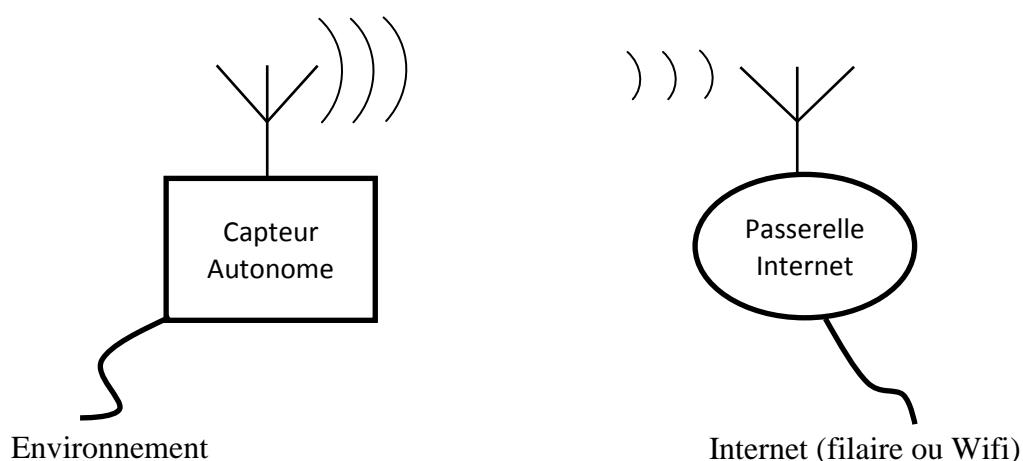


Figure 1 - Principe de l'internet des objets

Une grande partie des applications de l'IOT demande des objets communicants sans fil fonctionnant sur batterie (mobilité et facilité d'installation) à faible coût et longue autonomie (petite

batterie ou récupération d'énergie et faible consommation). La transmission d'une faible quantité de données à intervalle régulier d'un grand nombre d'objets vers Internet impose l'utilisation d'une passerelle, fixe (domicile, lieux publics, etc.) ou mobile (Smartphone). Les standards de radiocommunications adaptés à ces applications sont par exemple Bluetooth Low Energy (BLE) [1], ANT+ [2], ou ZigBee [3].

Récapitulatif des points que l'objet communicant de demain doit respecter :

- Communication radio sans fil sur batterie (mobilité, facilité d'installation) ou autonome en énergie (Récupération d'énergie)
- Bon marché (faible coût, batterie peu chère, pas d'entretien)
- Autonome et sans maintenance (faible consommation et/ou récupération)

La communication sans fil est le but premier, le deuxième point viendra avec son utilisation massive. Toutefois, l'autonomie longue et durable, qui nous fera oublier que ces objets intelligents sont parmi nous, n'est pas acquise et doit être recherchée.

Même (ou surtout) si ces objets utilisent la récupération d'énergie, ils doivent consommer très peu pour pouvoir s'autoalimenter et être bas coût. Pour avoir une faible consommation, il faut que la radio et le capteur consomment peu. Si le capteur peut facilement être optimisé pour consommer peu, la diminution de la consommation de la radio passera par :

- Utilisation parcimonieuse,
- Economie de trames,
- Economie de bit par trame,
- Distance d'émission faible,
- Rendement radio élevé.

En conclusion, le déploiement à grande échelle de l'internet des objets en général et des réseaux de capteurs en particulier, nécessite le développement de circuits et systèmes électroniques de plus en plus économes en énergie. Les circuits de génération de fréquences sont connus pour être un des blocs particulièrement énergivores. L'objectif de ce travail de thèse est double, d'une part développer une synthèse de fréquences très faible consommation et d'autre part démontrer les performances de la technologie FDSOI pour des applications analogiques et radiofréquences.

Pour illustrer nos travaux de thèse, nous avons choisi comme exemple d'application le standard Bluetooth Low Energy (BLE). L'objectif qui a été fixé est l'optimisation de l'architecture d'une PLL pour parvenir à une consommation inférieure au milliwatt.

2. Travaux de Thèse

Dans le premier chapitre seront exposés les trois ingrédients nécessaires pour obtenir un objet communicant autonome à longue durée de vie, tel que l'IOT l'envisage. Le premier ingrédient est le standard, BLE dans notre cas, spécialement développé pour les objets autonomes, leur permettant de transmettre peu de données à intervalles réguliers avec un bon rendement. Le deuxième ingrédient est la technologie CMOS avancée : le FDSOI, qui permet de créer des blocs analogiques et radiofréquences avec une consommation plus faible par rapport aux technologies précédentes et des dispersions technologiques plus faibles que les technologies actuelles. Cette diminution permet de réaliser des circuits plus performants sur une même surface. Le troisième ingrédient est l'architecture de la synthèse de fréquences du module radiofréquence qui joue énormément sur la consommation et donc l'autonomie. Ce bloc est très important car il génère la fréquence permettant de communiquer sur le canal radio.

Le deuxième chapitre analyse les résultats de trois types de modélisation système de l'architecture particulière de cette synthèse de fréquence, et propose, en conclusion, les modifications à apporter pour satisfaire le cahier des charges. La première étude concerne le fonctionnement de l'architecture et doit permettre de définir les points clés à respecter pour son implémentation. La deuxième étude concerne le comportement des bruits afin de définir les meilleurs paramètres architecturaux permettant de respecter les spécifications. Enfin, la dernière étudie l'impact de l'architecture sur la génération de raies spectrales parasites pour estimer leur amplitude en fonction de la variabilité du signal de contre-réaction, afin d'en limiter les effets.

Le troisième chapitre est ainsi consacré à la conception, la réalisation et le test de VCRO en technologie FDSOI 28nm. Plusieurs topologies sont présentées et intégrées sur un circuit de test. Les premiers résultats de mesure sont encourageants, mais nécessitent sans doute d'être complétés. En effet, la sensibilité des circuits à la tension d'alimentation (pushing de l'ordre de 5 GHz/V) a rendu les mesures du bruit de phase très délicates. La consommation mesurée reste inférieure à 0,8 mA et la surface de chaque VCRO est de l'ordre de 600 μm^2 .

Dans le quatrième et dernier chapitre, nous proposons une topologie de la PLL à resynchronisation de phases, chaque bloc ayant été conçu et optimisé au niveau circuit. Nous terminons ce chapitre par quelques pistes d'amélioration, notamment la technique d'auto-calibration par auto-polarisation de la grille arrière en utilisant une logique complémentaire qui a fait l'objet d'un dépôt de brevet. Une conclusion générale terminera ce mémoire.

Chapitre I - Standard, technologie et architecture pour l'autonomie

1. Introduction

Dans ce chapitre sont exposés les trois points les plus importants pour obtenir un système cohérent en termes d'économies d'énergie, permettant de concevoir des objets communicants par ondes radiofréquences autonomes et à longue durée de vie.

Le premier point est le standard de communication radiofréquence BLE qui a été optimisé pour consommer moins que sa version initiale (Bluetooth standard). Cette partie explique pourquoi le standard a été créé et rappelle son fonctionnement et ses caractéristiques. On détaillera ce standard après avoir expliqué les bases du fonctionnement des communications radiofréquences (RF) et les spécifications qu'une synthèse de fréquences doit respecter, ainsi que les caractéristiques du standard BLE qui permettent d'offrir une faible consommation.

Le deuxième point est la technologie CMOS FDSOI qui permet de fabriquer des circuits très économe en énergie. Cette partie discute les qualités du FDSOI qui nous aiderons à atteindre les spécifications, ainsi qu'à effectuer une analyse comparative des technologies CMOS précédentes.

Enfin, le troisième point concerne l'architecture du système radiofréquence, qui doit être conçue en suivant une approche bas coût et faible consommation. Dans cette partie sont présentés les différents types de synthèse de fréquences, de PLL, d'oscillateurs et de diviseurs et en particulier l'architecture à diviseur à commutation de phases. Pour conclure il sera expliqué quels sont les avantages espérés d'utiliser ces trois points ensemble.

2. Le standard Bluetooth Low Energy

2.1. Bluetooth Low Energy et Internet des Objets

2.1.1. L'émergence de l'IOT

Les télécommunications radiofréquences sont utilisées depuis longtemps pour la radiodiffusion sur de grandes distances vers un grand nombre de récepteurs, de par la facilité et leur faible coût de déploiement (peu de travaux de câblage, peu d'émetteurs nécessaires). Depuis que la

technologie nécessaire pour établir des télécommunications radiofréquences (émission et réception) a été rendu portable, celles-ci se sont développées exponentiellement et aujourd'hui le monde compte plus de terminaux mobiles que d'habitants [4]. La prochaine étape du développement des télécommunications est l'IOT, dont l'idée de base est de rendre les objets du quotidien communicants et autonomes pour qu'ils nous libèrent de tâches ingrates et répétitives de récupérations de données. Les premiers objets à être connectés sont ceux qui sont connectés au réseau électrique, car ils n'ont pas d'exigence sévère d'économie d'énergie. Mais les nombreuses applications mobiles ont un déploiement limité, de par leur manque d'autonomie vis-à-vis de l'utilisateur, et surtout électrique, car le résultat est mitigé si nous devons les recharger régulièrement. Leur autonomie doit augmenter pour que le marché croisse. En 2016 plus de la moitié des mobiles seront des Smartphones avec accès à Internet. Ces Smartphones peuvent servir de passerelle vers l'Internet pour les objets connectés à proximité de l'utilisateur, via un standard adapté à la faible consommation, tel que le BLE.

2.1.2. Avantages du BLE pour l'IOT

Le BLE, renommé Bluetooth Smart pour les consommateurs, est un standard de communication de type réseau personnel sans fil (WPAN). Il a été défini par le Bluetooth Special Interest Group (SIG)



en ayant en tête la réutilisation du hardware Bluetooth existant avec un protocole totalement différent, avec comme destinée l'IOT. Ces objets communicants du quotidien, existants ou à inventer, ont généralement besoin d'envoyer l'état d'un capteur ou de recevoir une commande (quelques octets) à intervalle régulier, comme par exemple dans le domaine médical, le sport, la sécurité ou la domotique.

L'objectif du BLE est simple : apporter à l'objet distant (capteur autonome) de quoi pouvoir transmettre et recevoir quelques octets à bonne distance (cent mètres en champ libre) avec une autonomie élevée (par rapport aux standards existants) de plusieurs mois à quelques années (en fonction de la périodicité de réveil) à partir d'une pile bouton, pour obtenir un prix très faible (quelques euros).

La solution retenue par le BLE pour réussir cela est de favoriser l'autonomie de l'objet recueillant les données par rapport au routeur qui le connecte à Internet ou à l'utilisateur. La solution technique est de désactiver complètement la radio quand on ne communique pas de données (pas d'écoute côté objet). Le routeur doit écouter le canal radio jusqu'à ce que l'objet se

réveille et lui envoie les données, après quoi le routeur peut en envoyer à son tour avant que la cible se rendorme.

Toute cette communication est optimisée :

- Pas de dialogue d'établissement de la communication, (dé-) connexion, etc.
- Pas d'envoi de données si le routeur ne répond pas
- Pas d'accusé de réception à un accusé de réception (lazy acknowledgement)
- Transaction complète limitée à 3ms (dont 1ms de calcul de trame)

Une autonomie de plusieurs années est possible de la manière suivante :

- Faible consommation hors communication (capteur, calcul trame, veille)
- Faible rapport cyclique d'utilisation de la radio (optimisation protocole)
- Faible consommation de la radio

La consommation des modules radio BLE existants est de 18mA [5] à 10mA [6] sous 3V en communication et de 0,5µA à 2µA en veille profonde (quelle que soit la tension). Concernant le rapport cyclique d'utilisation de la radio, la périodicité des transactions varie de quelques minutes pour un capteur de température jusqu'à 250ms pour un capteur de présence, pour un temps de transaction minimum de 3ms.

En optimisant la consommation de la radio, on peut gagner en autonomie, surtout sur les objets les plus réactifs. Le Tableau 1 démontre cela pour 3 périodicités différentes, avec une radio à l'état de l'art existant (15mW) et une autre 5 fois plus économe, une pile bouton d'énergie ($E_{BATTERIE}$) égale à 200mWh et un temps de transaction (D_{ACTIF}) constant de 3ms. Les formules utilisées permettent de calculer le rapport cyclique d'utilisation de la radio (1) et l'autonomie (2). On constate qu'en diminuant la consommation de la radio par 5, on augmente l'autonomie de 1,5 à 4,8 fois selon le temps de veille programmé. En concevant une PLL très faible consommation (<1mW) grâce au FDSOI, cela permettrait de concevoir une radio pour le Bluetooth LE consommant moins de 3mW.

$$\alpha = \frac{D_{ACTIF}}{Periode} \quad (1)$$

$$Autonomie = \frac{E_{BATTERIE}}{P_{VEILLE} + \alpha \cdot P_{RADIO}} \quad (2)$$

Tableau 1 - Comparaison du gain de l'autonomie en fonction de l'intervalle de communication

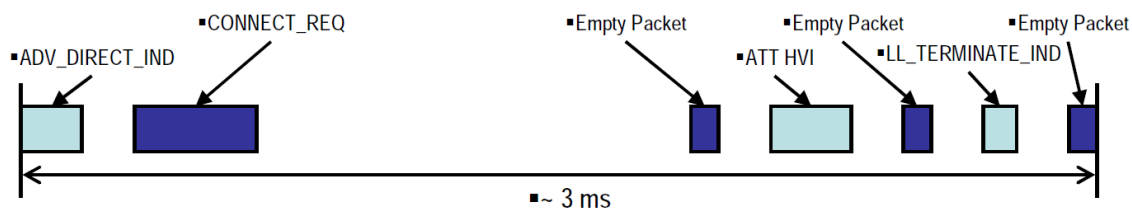
Périodicité	P_{MAX} Radio	P_{MOY} Radio	P_{MOY} Veille	Autonomie	Gain
60s	15mW	750nW	1µW	13,0 ans	-
60s	3mW	150nW	1µW	19,9 ans	x1,53
10s	15mW	4,5µW	1µW	4,15 ans	-
10s	3mW	0,9µW	1µW	12 ans	X2,9

1s	15mW	45µW	1µW	6 mois	-
1s	3mW	9µW	1µW	27,7 mois	x4,6
250ms	15mW	180µW	1µW	46 jours	X0,01
250ms	3mW	36µW	1µW	225 jours	x4,8

2.1.3. Transaction BLE type

La Figure 2 illustre une transaction BLE type, donné par le manuel de fonctionnement de BLE et réalisé par la société CSR [7]. Cet exemple montre la simplicité de l'établissement de la communication et la rapidité de la transaction.

Time (us)	Master Tx	Radio Active (us)	Slave Tx
0		176	ADV_DIRECT_IND
326	CONNECT_REQ	352	
1928	Empty Packet	80	
2158		144	Attribute Protocol Handle Value Indication
2452	Empty Packet (Acknowledgement)	80	
2682		96	LL_TERMINATE_IND
2928	Empty Packet (Acknowledgement)	80	



© CSR plc 2010. All rights reserved.

Figure 2 - Détails temporels d'une transaction BLE

2.1.3.1. Signalement

Le protocole BLE est simple: la passerelle (maître) écoute le réseau sur 3 fréquences de signalement (advertising) où l'objet distant (esclave) doit émettre pour signaler qu'il s'est réveillé. Ce sont les canaux 37 à 39 (2402, 2426 et 2480MHz) qui occupent ces fonctions, ils ont été choisis pour être en dehors des canaux WiFi les plus utilisés et ainsi diminuer les possibilités de collisions (cf. Figure 3). L'esclave peut transmettre 8 à 39 octets dans la trame de signalement, le maître peut ensuite répondre qu'il a reçu les données ou engager une connexion qui permettra d'envoyer des données à l'esclave.

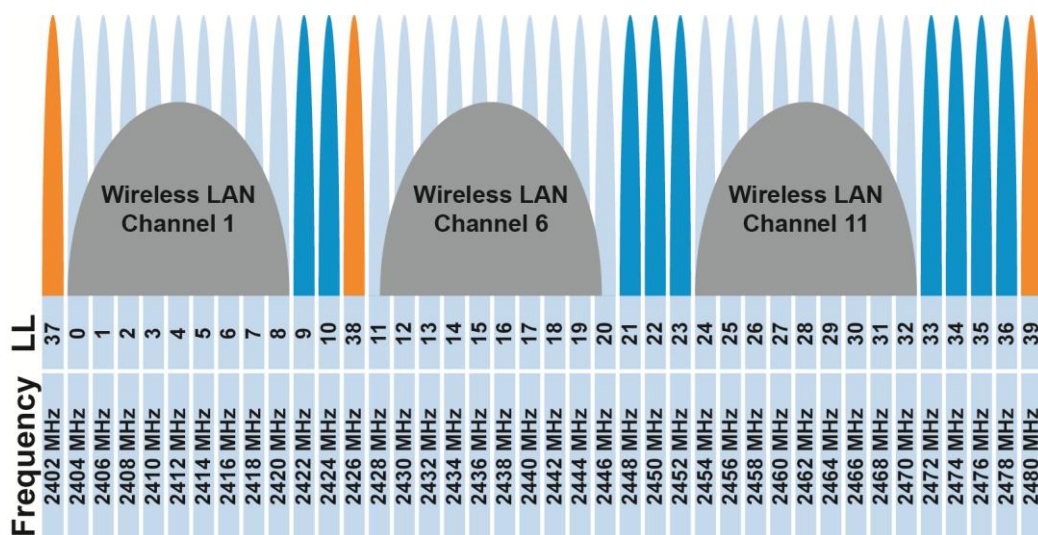


Figure 3 - Répartition des canaux BLE de signalisation (orange) et les canaux les plus utilisés du WiFi (1, 6, 11)

2.1.3.2. Connexion

Le maître capte le signalisation de l'esclave et y répond avec une requête de connexion dans la foulée. Puis au bout d'un temps borné entre 1,25ms à 11,25ms, le maître envoie un paquet vide pour signaler qu'il est prêt à recevoir.

2.1.3.3. Transmission des données

L'esclave ayant reçu la demande de connexion prépare les données à envoyer (mesures des capteurs, mise en forme et calcul de la trame) et envoie la trame dès qu'il reçoit le paquet vide signalant la disponibilité du maître.

2.1.3.4. Fin de transmission

A la fin de chaque paquet, le maître acquitte l'avoir reçu avec un paquet vide. Si l'esclave ne le reçoit pas, il retransmettra le paquet en question, sinon il enverra le paquet suivant ou alors un paquet de fin de transmission.

2.1.3.5. Récapitulatif

La transmission complète s'opère en trois millisecondes s'il n'y a pas de problème de transmission (BER élevé) impliquant une retransmission d'une trame. Cette rapidité de transmission permet d'économiser de l'énergie au niveau de l'esclave mais aussi de laisser du temps libre au maître pour communiquer avec d'autres esclaves.

2.1.4. Spécifications du BLE

Les spécifications des standards Bluetooth Low Energy [7] et Bluetooth « standard » [8] ont été regroupées dans le Tableau 2 pour comparaison. Elles seront expliquées dans la section suivante.

Tableau 2 - Comparaison des spécifications du Bluetooth 'normal' et du Bluetooth Low Energy

Modules Radio	Bluetooth Low Energy	Bluetooth « standard »
Portée (en champ libre)	150m	100m
PIRE (Puissance d'émission)	10mW (10dBm)	100mW (20 dBm)
Consommation en standby	1µA	1mA
Consommation RF active	<15mA	<30mA
Consommation	0,5 à 0,01 (a.u.)	1 (référence)
Autonomie sur pile	20-70M connexions (>1an)	7j (standby) ou 4h (voix)
Mode de communication	Advertising (écoute du maître)	Connexion (par l'esclave)
Topologie	Etoile	Etoile
N° d'esclaves pour un maître	Limité par le TDMA ~1000@4s	7 + 255 en standby (parked)
Latence (non connecté)	6ms	100ms
Durée de trame	3ms	100ms
Connexion	Périodique de 7,5ms à 32s	Continue
Données utile trame	1 à 31 octets	0 à 343 octets
Voix possible	Non	Oui
Fréquence	2402 à 2480 MHz	2400 à 2483,5 MHz
Canaux radio	40 x 2MHz	79 x 1MHz
Modulation	1Mbit/s GFSK	1-3Mbit/s GFSK
Indice de modulation	0,5	0,35
Débit utile de données	0,27Mbit/s	0,7-2,1Mbit/s
Robustesse	AFH ¹ , 24-bit CRC, 32-bit MIC ² , lazy and low power ACK	Fast AFH, 24-bit CRC, FEC ⁵ , fast ACK
Sécurité	128-bit AES ⁴ CCM ⁵	56/128 AES CCM
Profils de communication	Unique : GATT ⁶	35+ profils spécialisés

¹ AFH : Adaptive Frequency Hopping = Saut de fréquence adaptatif

² MIC : Message Integrity Check = Vérification d'intégrité du message

³ AES : Advanced Encryption Standard

⁴ CCM : Counter with Cipher Block Chaining-Message Authentication Code

⁵ FEC : Forward Error Correction

⁶ GATT : Generic ATtribute Profile

2.2. Communications sans fil

2.2.1. Bases de fonctionnement d'un lien radio

2.2.1.1. Bilan de liaison

Les communications sans fil numériques permettent de transmettre des données entre un émetteur et un récepteur par le biais des ondes radioélectriques (ondes électromagnétiques à fréquence inférieure à 300GHz) par le biais d'antennes (monopole/dipôle, boucle, patch ou parabole). Pour permettre la communication, le bilan de liaison [9] (cf. Figure 4) doit être positif (de 3

à 10dB), c'est-à-dire que la somme de la puissance d'émission et des gains d'amplification (dB, dBm ou dBi) doit être supérieure à la somme des pertes et de la sensibilité du récepteur :

- + Puissance d'émission (dBm)
- + Gain d'amplificateur de puissance (dB)
- + Gain d'antenne d'émission (dBi)
- + Gain d'antenne de réception (dBi)
- Pertes du canal de transmission (dB)
- Pertes d'adaptation d'impédance des interconnexions (dB)
- Sensibilité du récepteur (dBm)
- = Bilan de liaison (dB)

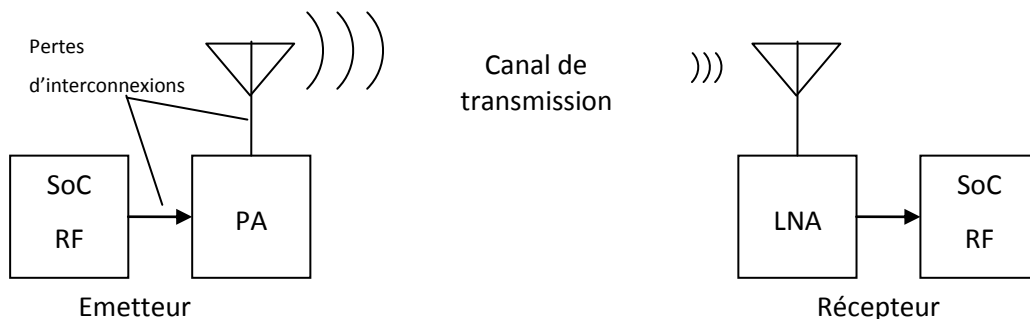


Figure 4 - Lien radio et bilan de liaison

La puissance d'émission s'entend à la sortie du circuit (System on Chip - SoC) radiofréquence (RF). Le gain d'amplification est celui de l'amplificateur de puissance (Power Amplifier - PA). Le gain d'antenne dépend de sa directivité, une antenne parabolique est directionnelle et possède un gain d'antenne élevé par rapport à une antenne isotrope (uniforme dans toutes les directions) qui est le référentiel 0dBi. Une antenne directionnelle permet d'économiser de l'énergie en n'émettant que dans la direction du récepteur mais n'est pas envisageable pour des objets mobiles à bas coût (pas d'antenne active), l'antenne doit donc être isotrope (ou s'en rapprocher) pour émettre et recevoir dans toutes les directions. Les pertes en espace libre (sans obstacle) dépendent de la fréquence du signal émis. Pour la bande utilisée par le BLE, ISM 2,4GHz, les pertes sont d'environ 80dB à 100 mètres et proportionnelles au carré de la distance. Chaque obstacle supplémentaire (arbre, vitre ou mur) ajoute des pertes et diminue la portée (respectivement 10, 2 et 20dB). Comme l'atténuation dans l'air des ondes électromagnétiques est proportionnelle au carré de la distance, la distance de communication maximum théorique est proportionnelle à la racine carrée du rapport entre la puissance de l'émetteur et la sensibilité du récepteur.

Les pertes d'adaptation apparaissent à chaque interconnexion de composants (puce, guide d'onde, antenne, etc.) côté émission ou réception et doivent être minimisées pour gagner en consommation pour la même portée. La sensibilité du récepteur dépend de nombreux critères comme la bande passante du signal reçu ou la puissance allouée à l'amplification. Communément, moins le signal contient d'informations, plus il est facile de les extraire, et souvent un module radio dépense plus d'énergie à la réception qu'à l'émission à cause de l'amplification du signal.

2.2.1.2. Réglementation

La puissance émise par l'antenne est règlementée en termes de Puissance Isotrope Rayonnée Equivalente (PIRE – en Watts), c'est la puissance maximale d'émission définie dans la direction où la puissance émise est maximale. Chaque pays réglemente l'attribution des licences d'émission (voir Tableau 3) et la PIRE par bande de fréquence. Certaines bandes sont libres d'exploitation (pas de licence), comme par exemple la bande ISM à 2,45GHz (de 2400 à 2500 MHz) utilisée par le BLE. Sur cette bande la PIRE en France est limitée à 100 mW (20 dBm) pour la bande 2 400 à 2 483,5 MHz sauf en France métropolitaine et en extérieur de 2 454 à 2483,5 MHz où elle est limitée à 10 mW (10 dBm); cette dernière limitation n'est pas présente dans les DOM-TOM et le reste de l'Europe.

Tableau 3 - Allocation des fréquences ISM en France

Bandes ISM	Bande	PIRE	Extérieur	Utilisateurs licenciés
6,765 - 6,795 MHz	30,0 kHz			Services fixes et mobiles
13,553 - 13,567 MHz	14,0 kHz			Services fixes et mobiles
26,957 - 27,283 MHz	326,0 kHz	10mW		Bande radio civile
40,660 - 40,700 MHz	40,0 kHz	16,4mW		Services fixes et mobiles
433,05 - 434,79 MHz	1,74 MHz	10mW ($\alpha \leq 10\%$)		Radio Amateur & radar
2,4 - 2,5 GHz	100 MHz	100mW	10mW (F > 2454 MHz)	Radio Amateur, lien microondes & radar
5,725 - 5,875 GHz	150 MHz	25mW		Radio Amateur, stations terrestres, lien microondes & radar
24 - 24,25 GHz	250 MHz	100mW		Radio Amateur & radar
61 - 61,5 GHz	500 MHz	100mW		lien microondes et radar
122 - 123 GHz	1 GHz	100mW		Radio Amateur et lien microondes
244 - 246 GHz	2 GHz	100mW		Radio Amateur, Radar & Radio Astronomique

2.2.1.3. Fréquence porteuse et types de modulation

Pour envoyer un signal radio, une porteuse $P(t)$ (fréquence centrale de la bande de fréquence allouée) est modulée pour y insérer les données à envoyer. Pour récupérer les données cette porteuse doit être démodulée par la méthode appropriée au type de modulation.

$$P(t) = A * \cos(B\omega t + \theta + C) \quad (3)$$

A, B et C représentent les trois types de modulation possible, à savoir, la modulation d'amplitude, de fréquence et de phase. Les données sont codées selon le standard : soit sur une

faible occupation spectrale (Figure 5), par modulation d'amplitude (AM), de fréquence (FM), de phase (PM) ou une combinaison amplitude/phase (QAM), soit ou sur une large occupation spectrale (Figure 6) en de courtes impulsions (IR-UWB), par modulation de phase ou de position.

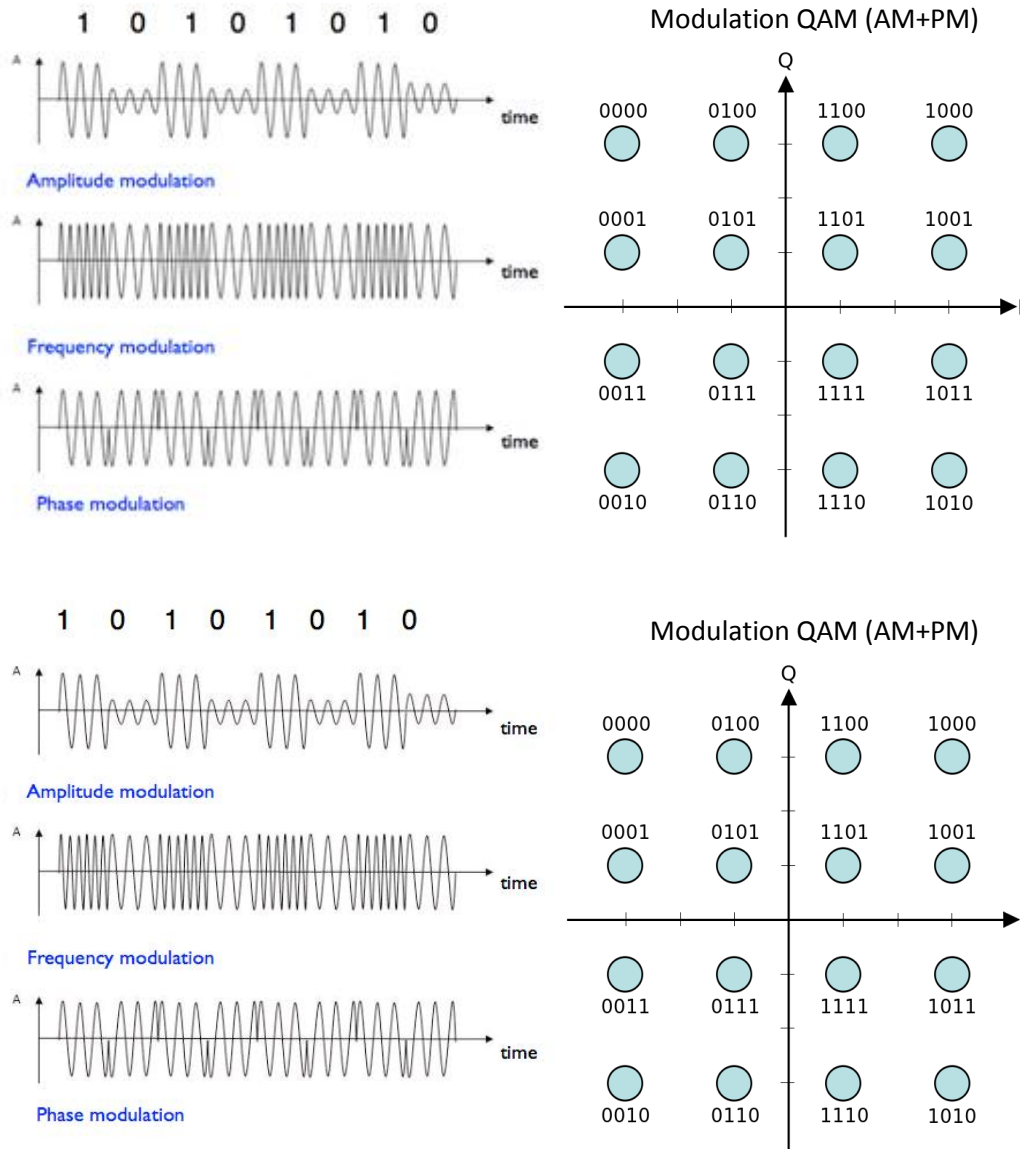


Figure 5 - Types de modulation à bande étroite (AM, FM, PM et QAM)

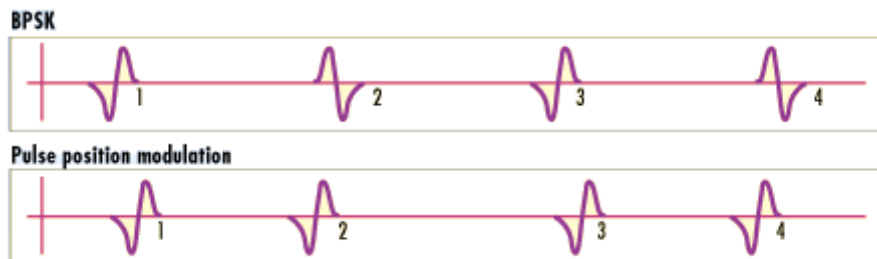


Figure 6 - Modulation bande large par impulsion (IR-UWB)

Le débit de données maximum théorique est fixé par le nombre de bits codés par symbole (nombre de valeurs de fréquence, de phase et/ou d'amplitude discrétisées) et la fréquence de modulation de ces symboles. La bande passante occupée est proportionnelle à la fréquence de modulation mais aussi au facteur de modulation.

2.2.1.4. *Étalement spectral et sensibilité*

En augmentant le facteur de modulation, on occupe un spectre plus important, illustré par la Figure 7, et on améliore donc la portée de détection par le récepteur. C'est le choix qui a été fait pour le BLE pour améliorer la portée sans changer ni la fréquence de modulation ni la puissance d'émission par rapport au Bluetooth. De plus pour simplifier les objets communicants et le protocole de communication, un seul débit est possible en BLE (1Mbps).

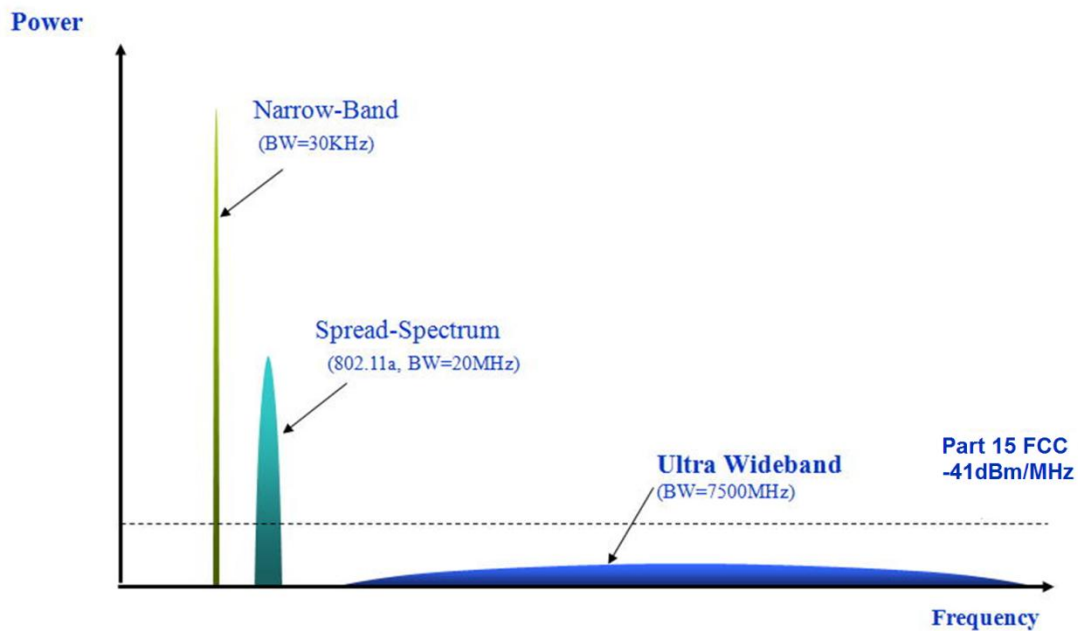


Figure 7 - Occupation spectrale de différents standards

2.2.1.5. *Bruit, SNR et BER*

Chaque composant depuis la modulation des données dans l'émetteur jusqu'à la démodulation des données dans le récepteur ajoute son bruit propre (thermique, flicker, intermodulations, distorsions) au signal et fait baisser le rapport signal à bruit (SNR). D'autres bruits peuvent s'ajouter au signal dans le canal de propagation, ceux-ci proviennent de réflexions du signal ou d'interférences avec des signaux perturbateurs (blockers) à proximité fréquentielle. Ainsi, même si le signal reçu a une puissance suffisante, si le SNR est trop faible le décodage des données est impossible. Les diverses corrections d'erreurs (CRC, redondance de données, multi-antennes)

permettent d'en corriger une partie. Au final c'est le taux d'erreur binaire (BER) qui exprime la qualité d'une transmission numérique.

La sensibilité du récepteur est proportionnelle au débit binaire, associé à un certain type et vitesse de modulation. Les spécifications de sensibilité sont donc données pour un certain BER (0.1% en Bluetooth) à un certain débit binaire. Quand la réception est mauvaise (SNR faible), les erreurs de décodage entraînent une chute du débit binaire, car les paquets doivent être réémis jusqu'à bonne réception (acknowledge). Avec certains standards comme le WiFi, il est possible de diminuer le débit pour garder un bon BER. Le BLE n'étant pas prévu pour une application haut débit mais vise une économie d'énergie, il n'y a pas de changement de modulation possible, ni de réémission en cas de non réception (en mode non connecté).

2.2.2. Composants d'un module radio

Les composants constitutifs d'un module de communication radiofréquence dépendent de l'architecture et de la modulation utilisée. Il sera présenté ici les composants typiques d'un module Bluetooth avec un émetteur à modulation directe et à démodulation hétérodyne.

2.2.2.1. *Emetteur à modulation directe*

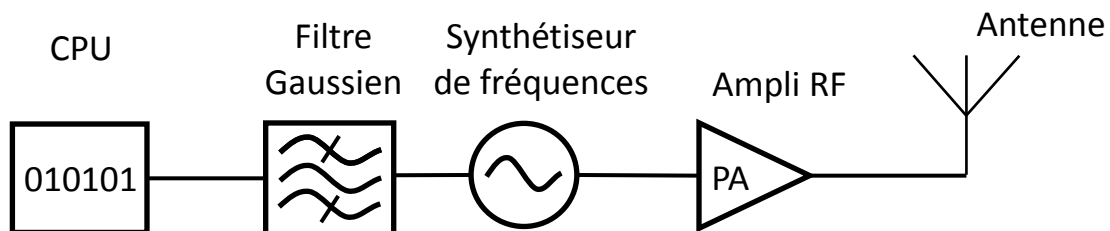


Figure 8 - Schéma synthétique d'un émetteur à modulation directe

La modulation directe (cf. Figure 8) signifie que le signal en bande de base module directement la fréquence d'un oscillateur commandé dont la fréquence est centrée sur celle du canal. Selon les spécifications du Bluetooth, le signal en bande de base (données numériques à émettre) doit passer par un filtre gaussien avant de moduler les sauts de fréquence pour la vitesse de transition et donc l'occupation spectrale du signal RF généré. La synthèse de fréquence transpose le signal de modulation autour de la fréquence du canal, puis l'amplificateur de puissance amplifie le signal RF modulé pour qu'il soit émis par l'antenne.

2.2.2.2. Récepteur à démodulation hétérodyne

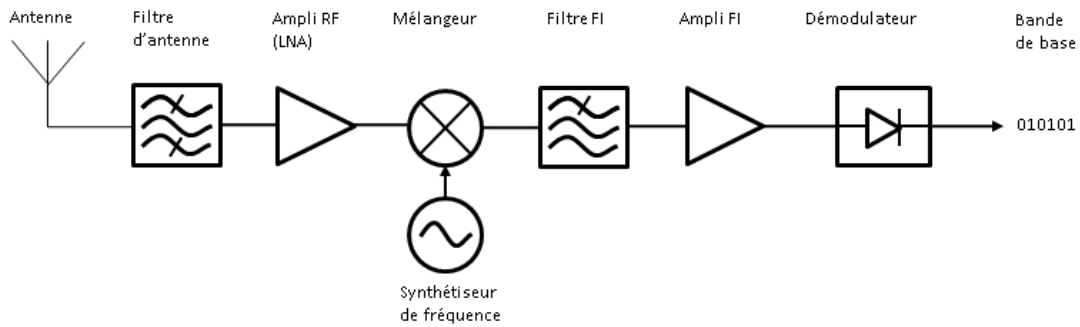


Figure 9 - Schéma synthétique d'un récepteur hétérodyne

Un récepteur hétérodyne (cf. Figure 9) utilise un mélangeur pour transposer le signal RF vers une fréquence plus basse, celle-ci pouvant être nulle (Zéro-IF). Il convient alors d'éliminer la fréquence image d'un récepteur hétérodyne. En multipliant le signal de l'antenne avec un signal à fréquence fixe, des intermodulations sont produites. Le filtre FI ne garde que le signal à la fréquence intermédiaire ($F_{\text{ANTENNE}} - F_{\text{SYNTHESE}}$) et après amplification le démodulateur extrait les données numériques. Il existe de nombreux types de démodulateur (démodulation I/Q, discriminateur à délai, détecteur de quadrature, etc.), mais ce n'est pas l'objet de ces travaux. Finalement, les données sont traitées par un processeur numérique.

2.2.2.3. Réutilisation de blocs

La synthèse de fréquences et l'antenne sont communes à l'émission et à la réception, et ne peuvent donc être utilisées simultanément. L'antenne est seulement commutée d'une voie vers l'autre. La synthèse doit pouvoir être modulée directement (méthode économe en énergie).

2.2.3. Influence des blocs sur les performances

Le LNA (low noise ampliflier) effectue la première étape d'amplification du signal reçu par l'antenne. Son bruit doit être extrêmement bas car il est ensuite amplifié par toute la chaîne de réception.

Le mélangeur transpose le signal reçu à la fréquence radio vers une fréquence plus basse en le mélangeant (multiplication) au signal fourni par le synthétiseur de fréquences, il doit être le plus linéaire possible sous peine de diminuer le rapport signal à bruit en générant des intermodulations non désirées.

Le synthétiseur de fréquences fabrique la fréquence nécessaire à la transposition, à partir d'un oscillateur de référence (généralement à quartz). Il doit avoir un faible bruit de phase sinon une

partie des signaux adjacents au signal RF recherché se retrouvera dans la bande de fréquence du signal utile après transposition. L'importance de la synthèse de fréquence est détaillée dans la section suivante car c'est le composant que nous avons choisi de concevoir et optimiser en technologie FDSOI.

Le filtre de fréquence intermédiaire (FI) et l'amplificateur associé élève la valeur du signal pour que le démodulateur puisse différencier les bits de données. L'amplificateur ou le démodulateur s'adapte à la puissance de réception (dynamique du signal) pour éviter la saturation et réussir à décoder le signal.

Les filtres RF et FI ajoutent un peu de bruit mais ont surtout une précision assez faible s'ils sont intégrés, et peuvent donc avoir une bande passante autre que celle conçue.

2.3. Synthèse de fréquences pour le BLE

Comme vu précédemment, la synthèse de fréquences est un bloc constitutif du module radio nécessaire pour transposer le signal vers (et depuis) le canal de communication radiofréquence. A partir d'un signal de référence à fréquence fixe, la synthèse doit générer une fréquence réglable pour adresser tous les canaux de communications du standard visé, en respectant les spécifications de ce dernier. De plus, en émission, la synthèse peut avoir à générer un signal modulé si l'architecture de modulation choisie est la modulation directe.

2.3.1. Caractéristiques d'une synthèse

Les performances que doit respecter une synthèse de fréquences utilisée dans un module de communication radiofréquence sont décrites ci-dessous, et sous chaque section sont expliqués les objectifs qui ont été fixés.

2.3.1.1. Consommation

La consommation du générateur de fréquences est un des paramètres les plus importants pour l'autonomie du système. Pour augmenter cette autonomie, on peut, quand le lien radiofréquence le permet, diminuer la puissance d'émission. Mais on peut aussi, au prix d'une complexité non négligeable, diminuer la consommation des autres composants. La consommation de la synthèse de fréquences de l'émetteur/récepteur n'est pas une donnée spécifiée par le standard, mais la consommation totale en émission/réception est conseillée à moins de 15mW. Néanmoins, pour parvenir à une bonne autonomie de l'objet communicant, il faudrait que la consommation soit autour de 3 milliwatts. Le synthétiseur de fréquences consomme habituellement jusqu'à 30% du

total du module radio [10, p., 5], et donc l'objectif de consommation fixé pour ce travail de thèse est de l'ordre du milliwatt.

2.3.1.2. Surface occupée

La surface occupée par la synthèse, mais surtout celle de la radio est décisive, car elle impacte directement le coût de la solution. C'est pour cela qu'il a été choisi de se concentrer sur une architecture très faible surface, tel qu'un oscillateur en anneau. Ce type d'oscillateur est connu pour avoir un très mauvais comportement en termes de bruit de phase (jitter) mais être très agressif vis-à-vis de la surface de silicium et de la consommation. L'objectif est alors d'intégrer un VCO, basé sur un oscillateur en anneau, dans une PLL afin de diminuer ce bruit de phase par un asservissement (calibration) du système en utilisant les grilles arrières des transistors MOS FDSOI de chaque inverseur.

2.3.1.3. Gamme de fréquences

La synthèse doit respecter la gamme de fréquences fixée par le standard, en tenant compte des variations technologiques de plus en plus importantes avec l'évolution de la microélectronique (loi de Moore). Pour respecter, ce point les oscillateurs en anneau utilisent couramment une calibration au démarrage. Pour le BLE, la gamme de fréquence est de 2400 à 2480 MHz.

2.3.1.4. Dynamique de Fréquence

La dynamique de fréquence est l'écart maximum entre deux fréquences à synthétiser, pour le BLE c'est 78MHz. Après une calibration initiale de la fréquence centrale de l'oscillateur, l'oscillateur doit avoir une dynamique de fréquence suffisante, ce qui est rarement un problème pour les oscillateurs en anneau.

2.3.1.5. Pas de synthèse

Le pas de synthèse est l'écart minimal adressable par la synthèse entre deux canaux de fréquences du standard choisi, 2MHz dans le cas du BLE. Il est fixe et déterminé par l'architecture, à cause du type de diviseur (entier ou fractionnaire), du pas du diviseur et de la fréquence de référence.

2.3.1.6. Temps de démarrage et de commutation de canal

Le temps de démarrage est celui que la synthèse met pour atteindre la fréquence cible à partir de l'alimentation du circuit. Ce paramètre va impacter la consommation moyenne, car plus le

démarrage est lent, plus la consommation moyenne sera élevée. La vitesse de démarrage dépend du facteur de qualité de l'oscillateur, qui influence le temps de stabilisation de la fréquence et le temps d'accrochage de la PLL. Ce dernier dépend de la bande passante et de la marge de phase du système bouclé. Dans le cas d'un oscillateur en anneau, le démarrage est très rapide (moins de 100 périodes) car le facteur de qualité est faible.

Le temps de commutation de canal est celui que met la synthèse à passer d'une fréquence cible à une autre. Ce paramètre dépend principalement de la bande passante et de la marge de phase du système de synthèse. Une bande passante trop faible limite la vitesse de synchronisation, et une marge de phase non optimale ($z=1/\sqrt{2}$; $MP=67^\circ$) ralentit la stabilisation. Le standard Bluetooth définit un délai maximum de 150 microsecondes pour changer de canal. L'oscillateur en anneau et la PLL à bande passante élevée permettent d'effectuer la commutation de canal très rapidement.

2.3.1.7. Bruit de phase et Spurious

Le bruit de phase est la performance en termes de modulation de la phase d'un dispositif électronique dont la sortie est périodique. Il exprime la densité spectrale de puissance qui n'est pas à la fréquence de la porteuse. Le bruit de phase est défini par convention par la puissance par bande de un Hertz, par rapport à la puissance de la porteuse, à un certain éloignement de celle-ci. Son unité est le dBc/Hz (ou dBrad²/Hz) à (x) Hertz de la porteuse. Ex : -90dBc/Hz@1MHz.

2.3.1.8. Pushing

Le pushing est une spécification de l'oscillateur faisant partie de la synthèse en boucle ouverte. Il représente la variation de la fréquence de sortie de l'oscillateur en fonction de la variation de la tension d'alimentation. Il donne une idée de la réjection ou sensibilité de l'oscillateur à une alimentation qui n'est pas stable ou propre. Cela permet d'estimer le bruit créé en sortie en fonction du bruit sur la tension d'alimentation et donc de définir les spécifications de bruit de l'alimentation.

2.3.1.9. Pulling

Le pulling est aussi une spécification de l'oscillateur en boucle ouverte et représente la variation de la fréquence de sortie en fonction de la charge vue à sa sortie. Si la charge varie dans le temps, cela produira du bruit de phase supplémentaire que l'asservissement essaiera de supprimer. Ce phénomène peut se produire si la charge en sortie est un multiplexeur (état ouvert / passant) à une fréquence différente ou un modulateur donc la tension analogique moyenne fait évoluer les capacités parasites des transistors.

2.3.2. Transposition en fréquence non idéale

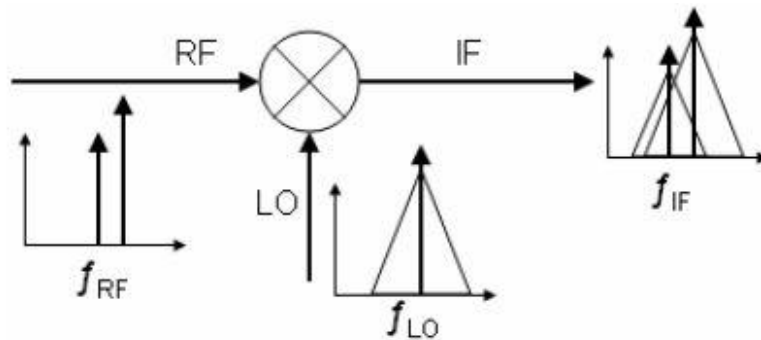


Figure 10 - Transposition en Fréquence avec du Bruit

La fréquence générée par la synthèse va permettre de transposer un des canaux de communication en bande de base pour décodage. Si la fréquence générée est bruitée, les signaux adjacents au canal, potentiellement plus puissants, seront aussi transposés en basse fréquence (cf. Figure 10), diminuant le SNR du signal reçu.

2.3.3. Résumé des spécifications pour la synthèse

Le Tableau 4 rappelle les spécifications que doit attendre la PLL que nous désirons concevoir dans ce travail de thèse.

Tableau 4 - Résumé des spécifications de la synthèse de fréquence à concevoir

Spécification	Valeur	Unité
Consommation	~1	mW
Surface	<0,01	mm ²
Gamme de fréquence	2402 à 2480	MHz
Dynamique	78	MHz
Pas de synthèse	2	MHz
Démarrage	<1	ms
Délai de saut	<150	µs
Bruit de phase	<-90	dBc@1MHz
Spurious	<-70	dBc@1MHz

2.4. Conclusion : L'apport du BLE

Le Bluetooth Low Energy est un standard de communication réellement adapté aux communications à faible consommation. Il permet la mise en œuvre de plusieurs solutions simplificatrices pour la conception et le fonctionnement du module radio, permettant d'augmenter grandement la longévité des objets connectés sur batterie. La synthèse de fréquences étant un bloc important et consommateur d'énergie au sein d'un module radio, nous nous fixons comme objectif

l'horizon [12]. Les physiciens tel que M. Kaku ont aussi prédit la fin de la loi de Moore [13]. Qu'est-ce qu'il y aura après le silicium et la technologie CMOS ? Il y a beaucoup de propositions : nanotubes de carbone, graphène, optique, dispositifs moléculaires, etc. (voir Figure 12 [4] et Figure 13 [14]).

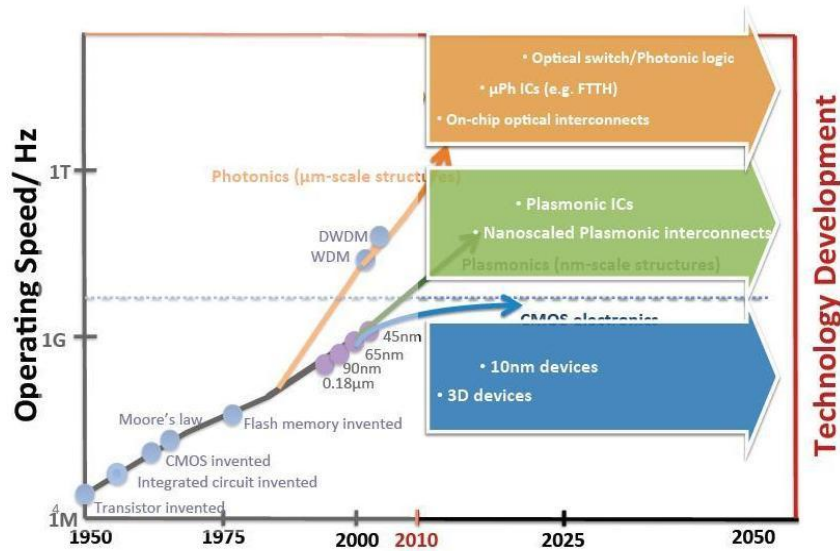


Figure 12 - Tendances des technologies électroniques [4]

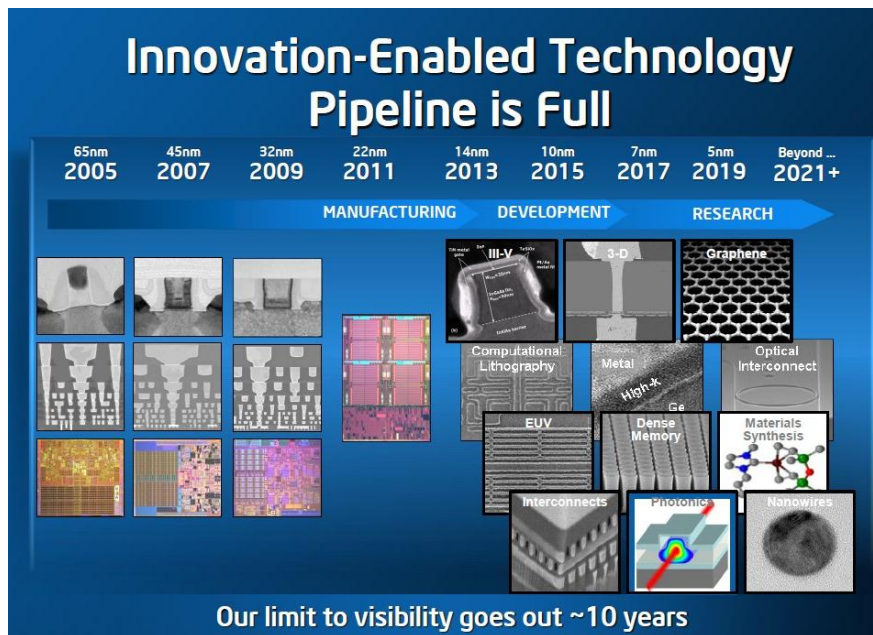


Figure 13 - Point de vue d'Intel sur les innovations technologique en électronique [14]

Un des points important est la manufacturabilité, qui peut être résumé par la capacité à fabriquer des circuits intégrés rentables. Jusqu'à maintenant, les solutions industrielles mettent en avant la technologie CMOS à base de silicium. Pour passer outre le problème de technologies agressives comme le nœud 28 nm et en deçà, deux nouvelles solutions sont apparues : les transistors FinFet et les transistors UTBB-FDSOI.

De plus, alors que les blocs numériques subissent directement la loi de Moore par une diminution de leur surface, les blocs analogiques, en raison notamment des composants passifs, ne suivent pas cette loi (voir Figure 14 [15]). Les blocs numériques, n'ayant comme spécification que le temps de propagation et la consommation maximum, peuvent être conçus de taille minimale contrairement aux blocs analogiques qui eux doivent respecter une panoplie de spécifications. Ces spécifications dépendent de l'erreur relative entre composants dont les dimensions, à cause des variations des procédés de fabrication, doivent rester grandes par rapport au minimum technologique. Cela se traduit, au fil des nœuds technologiques, par une diminution proportionnelle des blocs numériques et une diminution beaucoup moins rapide des blocs analogiques et RF.

Le seul moyen permettant de diminuer la taille totale d'un circuit mixte (analogique, RF et mixte) est soit de passer des fonctions analogiques dans le domaine numérique soit d'améliorer la technologie (variabilité, capacités parasites, courant de fuite, variations en fonction de la température, etc.) pour que les blocs analogiques réduits gardent leurs performances.

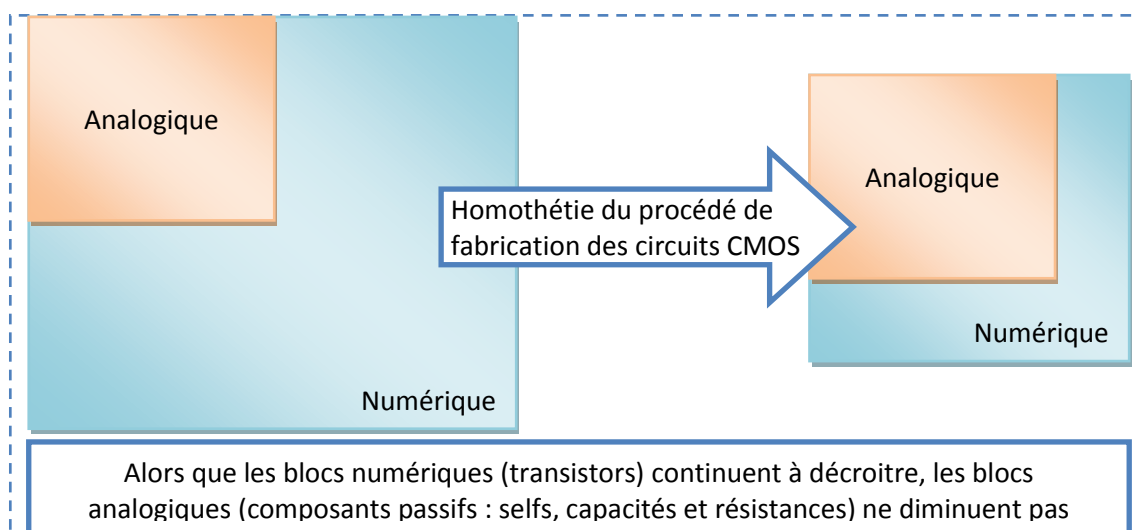


Figure 14 - Le problème de diminution de taille de l'analogique

3.2. Evolution de la technologie microélectronique

3.2.1. La fin de la loi de Moore ?

L'histoire de la microélectronique peut être résumée par la Figure 15. En effet, la microélectronique est présente dans tous les aspects de notre vie de tous les jours et les circuits intégrés permettent de réaliser de plus en plus de fonctions complexes. Depuis quelques années, deux domaines existent : « More Moore » et « More than Moore », comme expliqués en Figure 16.

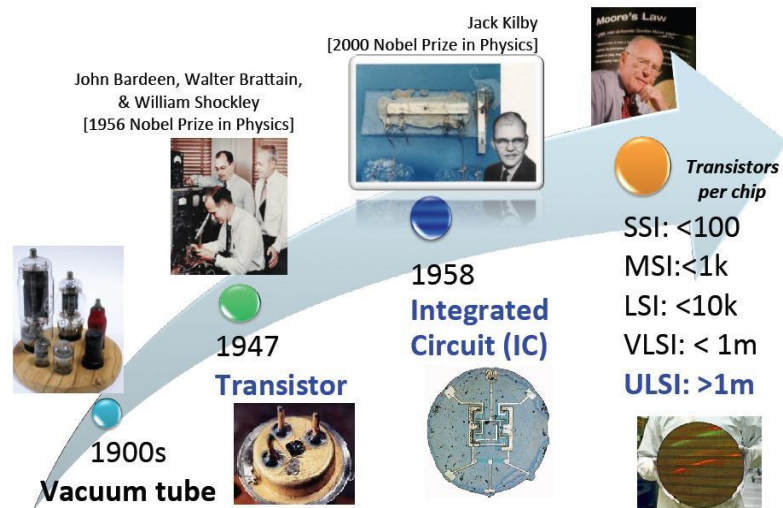


Figure 15 - L'histoire de la microélectronique

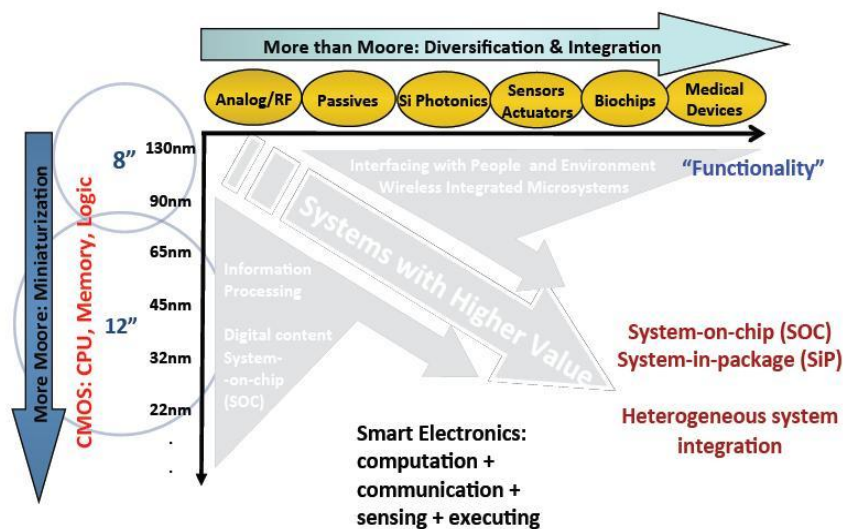


Figure 16 - More Moore Vs More than Moore

Le domaine du « More Moore » est défini internationalement comme la tentative de développer des technologies CMOS avancées afin de réduire les coûts associés par fonction, tout en augmentant les performances des circuits. Presque 70% du total du marché total des composants semiconducteurs est impacté par la miniaturisation du CMOS avancé dans le domaine « More Moore ». Ces 70% comprennent trois groupes de composants de taille similaire: les microprocesseurs, les mémoires de masse et la logique numérique.

“More than Moore” est basé sur, ou dérivé de, la technologie silicium, mais pas nécessairement dans le sens de la loi de Moore. Cela se réfère à un ensemble de technologies qui permettent des fonctions micro-nano-électroniques non-numériques, comme :

- Radiofréquences
- Haute tension et puissance

- Eclairage et imagerie à base de composants à l'état solide
- Ultrason médical, circuits bioconnectés, laboratoire sur puce et microfluidique
- Récupération d'énergie
- Capteur et actionneurs sur plateforme CMOS

Les composants "More than Moore" fournissent la conversion des informations non-numériques ou non-électroniques, comme mécanique, thermique, acoustique, chimique, optique et biomédical, en données numériques et vice-versa.

Dans le futur, les transistors seront si petits que la physique quantique prendra le pas sur la physique classique et les électrons commenceront à fuir en dehors des fils par exemple. La fin de la loi de Moore est à l'horizon, due aux inconvénients des transistors MOS Bulk sur les nœuds agressifs (14nm et en dessous).

3.2.2. Limitations des transistors MOS Bulk dus aux nœuds agressifs

Qu'est-ce qu'un bon transistor pour les applications numériques ? C'est un interrupteur idéal, à savoir :

- Pas de fuite quand il est ouvert
- Pas de pertes ohmiques quand il est fermé
- Capable de commuter à faible V_{TH} et donc faible V_{dd} (faible consommation)

Pour pouvoir suivre la loi de Moore, les tailles des transistors silicium sur substrat doivent diminuer, ce qui induit certains inconvénients. Le premier concerne l'oxyde de grille. Pour pouvoir garder constant le champ électrique entre drain et source, la tension d'alimentation doit diminuer proportionnellement à la taille du transistor. Une des conséquences de cette diminution est le besoin de diminuer l'épaisseur d'oxyde pour diminuer la tension de seuil du transistor, de 300nm sur les générations MOS précédentes, jusqu'à 1,2nm aujourd'hui. Et comme ce n'est pas un isolant parfait, il fuit et peut éventuellement claquer si le champ électrique est trop grand. Avec l'arrivée du nœud 90nm, les technologues ont introduit l'isolant « High-K » (Oxyde d'Hafnium, HfO_2) où k est la constante diélectrique, qui correspond à la capacité à isoler.

Les technologies diminuant, le courant I_D est limité. Le courant I_{DS} dépend de la taille du canal d'inversion. Plus le canal est court, moins les porteurs ont le temps d'accélérer sous l'effet du champ électrique (saturation de la vitesse). Le silicium contraint, imagé sur la Figure 17 [16], est une des solutions que les ingénieurs ont inventée pour augmenter la mobilité des porteurs.

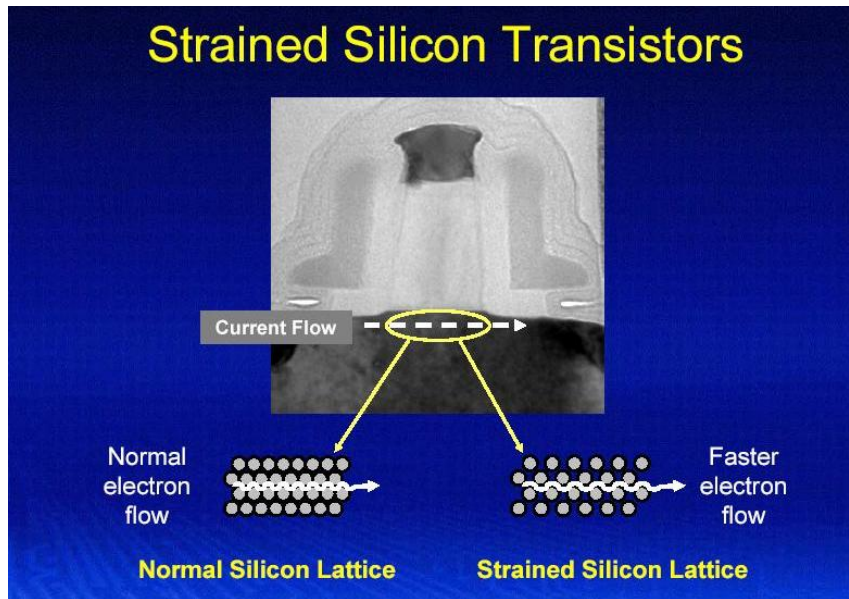


Figure 17 - Silicium contraint introduit par Intel [16]

La mise en conduction apparait sous l'effet qu'un champ électrique suffisamment élevé, créé par la tension de grille, et qui va générer une couche d'inversion : le canal de conduction. La diminution et le contrôle du seuil est important pour les circuits à basse consommation (faible tension d'alimentation). L'introduction d'une grille métallique associée à un diélectrique High-K, appelée « High-K MG », favorise cette mise en conduction.

La réduction de la taille des transistors induit des effets de canal court (SCE – Short Channel Effect) et de diminution du seuil induite par le drain (DIBL - Drain Induced Barrier Lowering), le transistor fuira d'autant plus (voir Figure 18).

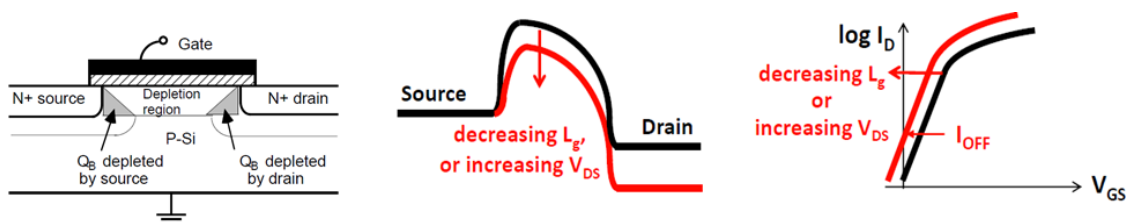
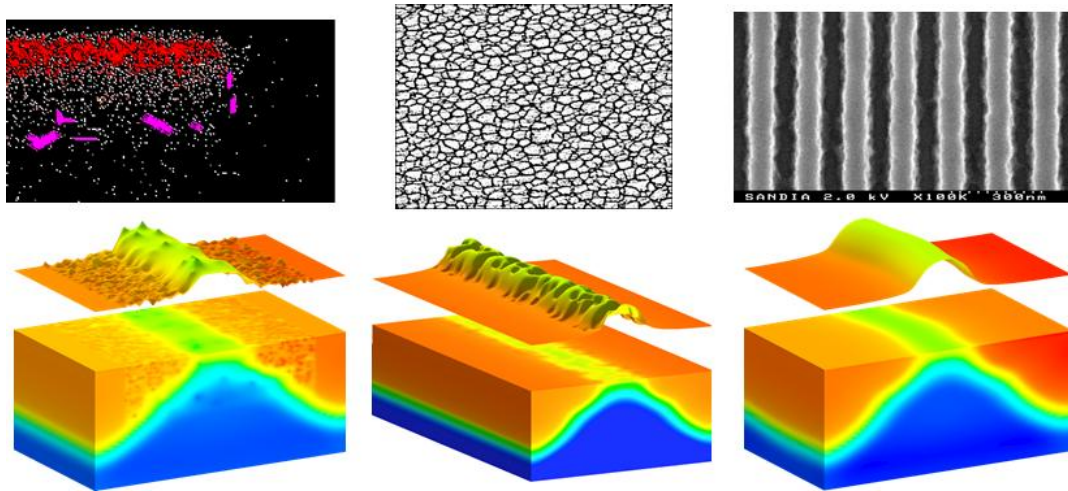


Figure 18 - SCE et DIBL

Le dernier problème, et non le moindre, concerne la variabilité. Au fur et à mesure que les transistors sont réduits vers la dizaine de nanomètres, la variation de la tension de seuil devient un problème préoccupant. Comme montré sur la Figure 19 [17], cela est dû aux fluctuations aléatoires des dopants (RDF - Random Dopant Fluctuations) dans la région du canal ainsi qu'à la rugosité des interfaces (LER - Line Edge Roughness), et dans une moindre mesure, par la variation de l'épaisseur d'oxyde.



(a) Aléas de dopage (b) Granularité grille métal ou poly (c) Rugosité des interfaces

Figure 19 - Différentes sources de variabilité

Cependant, le dopage du canal par une distribution aléatoire d'une faible concentration de dopants est le principal problème rencontré par les technologues pour la diminution des transistors sur substrat (Bulk). En effet, à partir du nœud 22 nm, seulement 50 atomes de dopants devraient se trouver dans la zone du canal. On s'attend donc à avoir une grande variabilité entre 2 transistors adjacents du même circuit ! La variation des dopants cause directement une variation de la tension de seuil entre deux transistors identiques et adjacents sur la même puce (Variabilité de V_{Th}). La solution sur les nœuds inférieurs à 22nm est donc d'utiliser des transistors non-dopés. L'UTBB FDSOI et le FinFet ont adopté des transistors non-dopés, dont la tension de seuil (V_{Th}) est ajustée entre autres par le travail de sortie de la grille métallique.

En vue des difficultés à réduire les dimensions des transistors CMOS planaires en gardant un contrôle grille-canal acceptable, les transistors multi-grille (MuGFET) tel le FinFet, ont été proposés comme option technologique pour remplacer la technologie existante [18].

Le terme FinFet a été inventé par les professeurs de Berkeley pour décrire un transistor non-planaire, double grille, fabriqué sur substrat SOI [19]. Ces transistors, représentés en Figure 20, utilisent une grille horizontale, empilée au-dessus de deux grilles verticales, pour atteindre environ trois fois plus de surface de contrôle sur le canal. Cette configuration unique de grille entourant le canal sur trois cotés permet un bien meilleur contrôle électrostatique et une meilleure résistance aux fluctuations des dopants. Grâce à cette structure 3D innovante et un canal plus étroit, cette technologie apporte de bien meilleures performances et une plus faible consommation que les transistors planaires sur substrat [20].

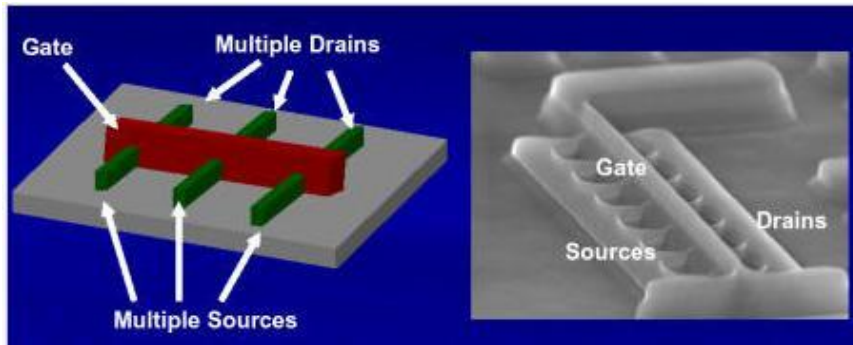
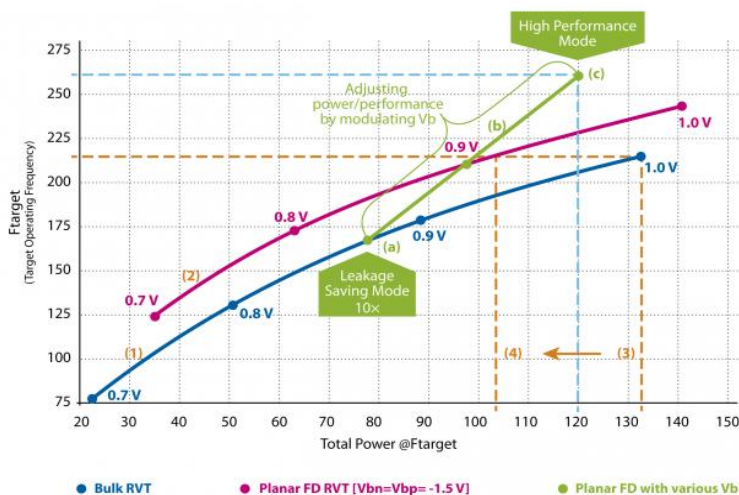


Figure 20 - Vue schématique et physique (MEB) des transistors Intel tri-gate [19]

Cette technologie, ne faisant pas partie de ce travail de thèse, ne sera pas plus détaillée dans ce mémoire. Par contre, les transistors FDSOI sont décrits dans la section suivante.

3.2.3. La technologie FDSOI

La technologie FDSOI est basée sur une mince couche (5 à 20nm) de silicium sur une mince couche (5 à 50nm) d'oxyde enterré (Buried Oxide - BOx) [21]. Les transistors sont construits sur la fine couche de silicium non-dopée (appauvrie en charges) qui a plusieurs avantages par rapport au Bulk. Comme le canal est complètement appauvri, la variation aléatoire des dopants qui touchait le CMOS Bulk est réduit ce qui améliore les performances à VDD plus faible. Le FDSOI revendique une amélioration du ratio Consommation/Performance de l'ordre de 30 à 40% face au CMOS Bulk 20nm (Figure 21) [22], se positionnant comme une réelle alternative déjà en production.



- (a) La polarisation inverse des BGs permet de diminuer la fuite par un facteur 10
- (b) Cette ligne est le FD-SOI 20 nm avec polarisation arrière
- (c) Avec la polarisation arrière du FD-SOI, Il est possible d'atteindre 269 MHz avec 120 mW sous 1V
- (1) Cette ligne est le 20 nm Bulk
- (2) Cette ligne est le 20 nm FD-SOI
- (3) La technologie Bulk utilise plus de 130 mW pour atteindre 223 MHz sous 1 V
- (4) Pour atteindre la même fréquence il ne faut que 100mW sous 0,9V

Figure 21 - Comparaison de consommation des technologies Bulk Vs FDSOI [22]

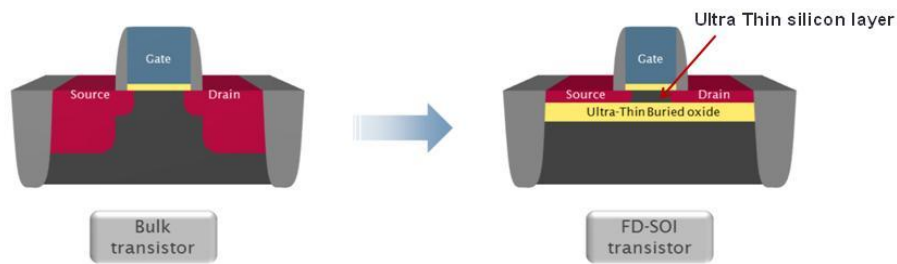


Figure 22 - Du transistor MOS au transistor FDSOI

Appelé transistor UTBB (Ultra Thin Body and Buried Oxide), les transistors FDSOI correspondent à une simple évolution du MOS traditionnel, comme exposé en Figure 22. Les motivations derrière le transistor UTBB sont les suivantes :

- Pas de changement fondamental de la géométrie des transistors. C'est une simple évolution de MOS traditionnel. Il peut être développé à un coût et un effort raisonnable (moins de 10%).
- Immunité aux effets de canal court (Short Channel Effects - SCE) grâce au canal ultra fin (Tsi $\sim 1/3$ LGate). Tous les chemins possibles entre source et drain sont proches de la grille, ainsi, la pente sous le seuil et le DIBL (Drain Induced Barrier Lowering) permettent d'excellentes valeurs.
- Immunité à la variabilité. Cette technologie n'a pas besoin de dopage ou d'implantations pour contrôler la pente sous le seuil V_{TH} . Le plus gros problème, la variabilité de la tension, est en grande partie réglé. L'absence de dopant dans le canal induit une grande mobilité et une bonne performance.
- Multi-Vt : dans le procédé Bulk, la gamme de tension de seuil disponible est obtenue par un dopage plus ou moins fort du canal. Ce n'est pas le cas en FDSOI car il n'y a pas de dopage. Mais plusieurs méthodes permettent d'ajuster le travail de sortie de la grille du transistor (Work Function) déterminant la tension de seuil.
- Possibilité de commande ou calibration par la grille arrière (BG)

Cette dernière caractéristique est très intéressante pour la conception analogique et a été utilisée dans notre application. La Figure 23 représente le transistor UTBB avec son contact de grille arrière et la Figure 24 [21] montre l'influence de la grille arrière sur la tension de seuil. Cette caractéristique justifie l'utilisation de la grille arrière pour modifier efficacement la consommation ou la fréquence de fonctionnement, ou calibrer une PLL, ce qui est quasiment impossible avec une technologie Bulk classique.

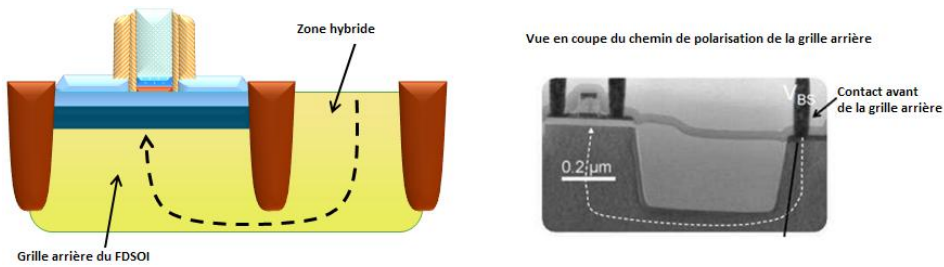


Figure 23 - Possibilité de polarisation de la grille arrière

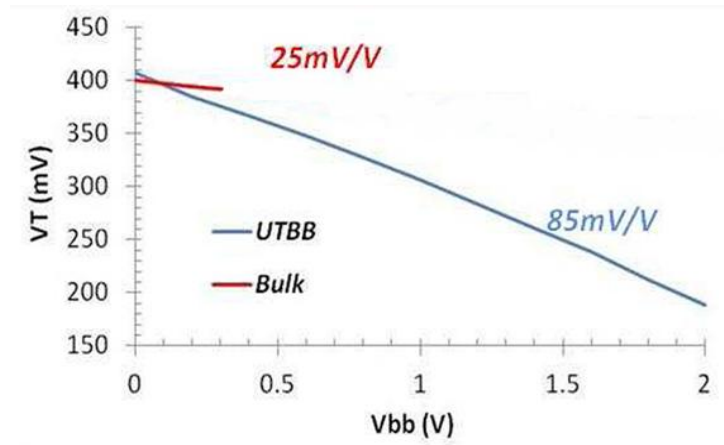


Figure 24 - Variation de V_{TH} Vs polarisation de la grille arrière

Cette technologie permet de réaliser toutes les configurations habituelles (différentes tensions de seuil) avec seulement 4 types de transistors comme le montre la Figure 25.

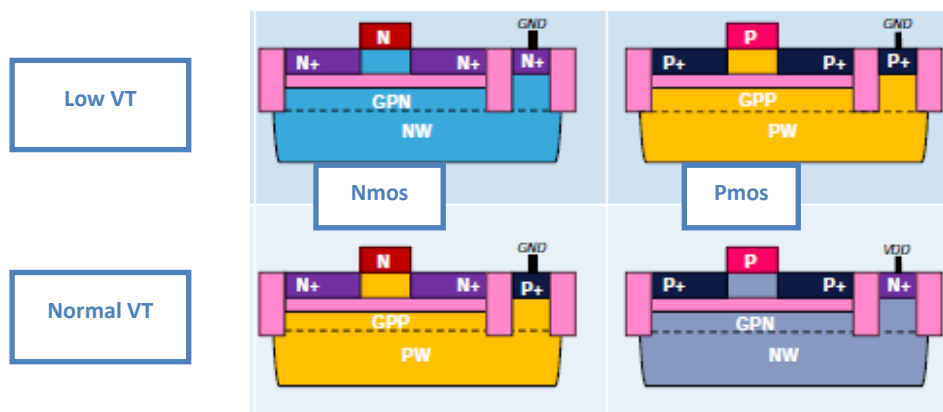


Figure 25 - Les 4 types de transistors FDSOI

3.2.4. La variabilité de la tension de seuil dans les nœuds agressifs

Alors que la technologie est réduite dans la gamme du deca-nanomètre, les variations de la tension de seuil (V_{TH}) deviennent un problème de plus en plus grand. Ces phénomènes sont

regroupés sous le coefficient Pelgrom (A_{VT}) [23], [24], utilisé pour définir la variation de V_{Th} pour chaque transistor ($\sigma_{VT}=A_{VT}/\sqrt{WL}$). Comme exposé sur la Figure 26 [25], l' A_{VT} pour les circuits CMOS Bulk 32 nm est au-dessus de 2,5 à 3,5 mV.µm [26]. Pour l'UTBB-FDSOI l' A_{VT} est à 1,1 et 1,25 mV.µm pour les nœuds 32 et 22 nm, en grande partie grâce au canal non dopé et à la diminution significative de l'influence du RDF (Random Dopant Fluctuations).

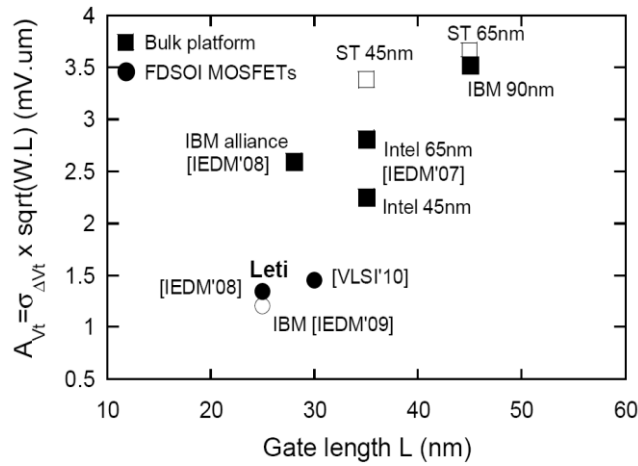


Figure 26 - Coefficient de Pelgrom Vs longueur de grille

L'écart type de la tension de seuil dû aux variations du procédé de fabrication entre la technologie CMOS bulk standard et la technologie FDSOI est donnée par les équations suivantes (équations 4 à 6 [27]). Cet écart type est ainsi réduit par un facteur 3.

$$\begin{aligned} \sigma_{VT} &= A_{VT} \frac{1}{\sqrt{WL}} \\ &= \frac{\sqrt[4]{2\varepsilon_s \varepsilon_0 q^3 N_a \Phi_d}}{\varepsilon_{ox} \varepsilon_0} T_{OX} \frac{1}{\sqrt{WL}} \end{aligned} \quad (4)$$

$$\sigma_{VT} \propto N_a^{0.25} \text{ with } \begin{cases} N_{a_{Bulk}} \approx 1E18 cm^{-3} \\ N_{a_{SOI}} \approx 1E16 cm^{-3} \end{cases} \quad (5)$$

$$\sigma_{VT} \propto N_a^{0.25} \text{ with } \begin{cases} N_{a_{Bulk}} \approx 1E18 cm^{-3} \\ N_{a_{SOI}} \approx 1E16 cm^{-3} \end{cases} \quad (6)$$

3.2.5. Méthodes de simulation de la variabilité

En prenant l'exemple d'un oscillateur en anneau, les délais sont inversement proportionnels à la résistance « ON » du transistor et à la capacité de charge (capacité de grille et capacité parasite).

Les variations locales du procédé de fabrication affectent la conduction du transistor (en faisant varier la tension de seuil) et la charge (en faisant varier la largeur et l'espacement des fils). Cela se traduit par une variation globale de la fréquence et une variation locale des temps de propagation de chaque transistor. La Figure 27 présente ainsi ces variations en fonction de la tension de grille [17].

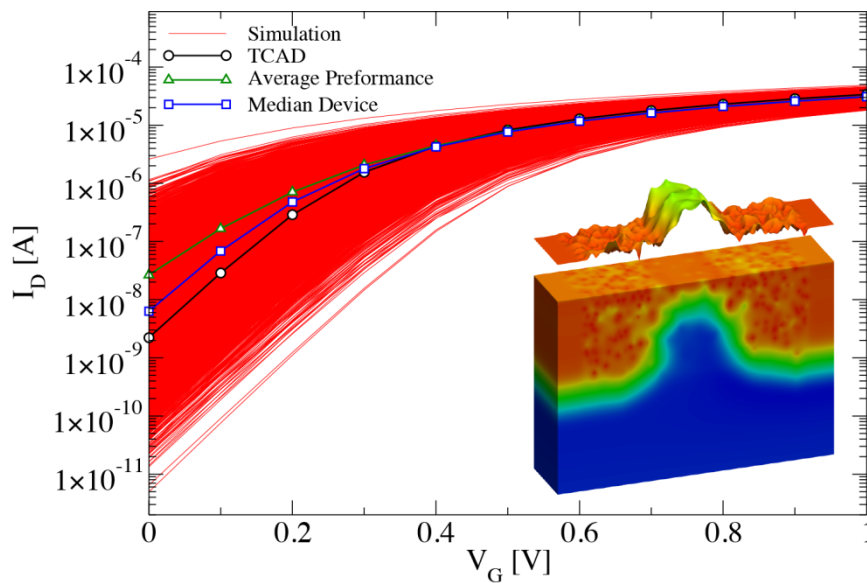


Figure 27 - Influence de la variabilité sur la tension de seuil et le courant du transistor [17]

Par exemple, dans une pompe de charge, la variabilité impacte la tension de seuil et donc l'équilibre des courants, ce qui donne lieu à une augmentation du bruit de phase de la PLL par augmentation du temps d'activation.

Chaque phénomène peut être prédit par différents types de simulations : Simulation MonteCarlo (MCS) et Post-Layout (PLS). La MCS prédit assez bien (avec moyenne et variance) la variation des délais due à la variation de la tension de seuil, mais ne prend pas en compte les capacités parasites. La PLS, qui est une simulation sur l'extraction des parasites du layout, donne le cas nominal et les pires cas (RCmin/RCmax) des déviations des capacités parasites.

Plusieurs théories cohérentes avec les variations du procédé de fabrication ont été déduites à partir de simulations : pour diminuer la variabilité du délai de propagation due aux transistors, leur longueur ne doit pas être la valeur minimale de la technologie (28 nm). De plus la largeur doit être maximale (pour moins de bruit) à courant constant. Pour diminuer la variation due à la charge, les capacités parasites doivent être minimisées par rapport aux capacités de grille (moins de déviation et valeurs plus grandes), et les signaux des phases de l'oscillateur en anneau doivent être aussi espacés et aussi courts que la technologie le permet dans les trois dimensions. Les capacités parasites des

transistors (grille vers source/drain) peuvent être minimisées en augmentant leur distance et le pas (pitch) des grilles des transistors multi-digités.

3.2.6. Méthodes de correction de la variabilité

Le transistor FDSOI possède une grille arrière (BackGate) qui peut être commandée entre les rails d'alimentation et agit sur la tension de seuil des transistors. Dans de nombreux circuits, numériques, analogiques ou radiofréquences, cela peut servir à ajuster certaines performances, tels que le gain, l'offset, le temps de propagation ou la fréquence de fonctionnement. Quelques méthodes utilisables pour les synthèses de fréquences sont brièvement présentées ci-dessous.

3.2.6.1. *Frequency Locked Loop*

Une boucle à verrouillage de fréquence (FLL) est une méthode d'asservissement de la fréquence de sortie des oscillateurs utilisés dans les synthèses de fréquences, soumis à la variabilité de la technologie [29]. Il est possible que l'oscillateur ne couvre pas correctement la gamme de fréquences recherchée. Cette méthode permet de calibrer la fréquence centrale de l'oscillateur au démarrage de la synthèse. On doit agir sur un point de commande qui soit indépendant de l'asservissement de la synthèse, comme par exemple la tension d'alimentation ou la charge capacitive ou la grille arrière des transistors de l'oscillateur.

Son fonctionnement est très simple, après avoir configuré la synthèse avec les paramètres nominaux, on lance la calibration par FLL jusqu'à obtenir la fréquence nominale. Un changement de température du circuit, peut obliger à relancer la calibration par FLL.

3.2.6.2. *Duty Cycle Correction*

La correction du rapport cyclique pour respecter un rapport cyclique de 50% est très importante dans les circuits non différentiels qui utilisent les 2 fronts pour fonctionner, comme par exemple un signal d'horloge qui commande 2 bascules (flip-flops) sur des fronts opposés. La calibration peut s'effectuer en temps réel par une pompe de charge simplifiée qui agit sur les grilles arrière (backgate) du FDSOI pour corriger l'erreur.

3.2.6.3. *Propagation delay correction*

La correction du délai de propagation peut être nécessaire dans certains cas où l'on désire avoir des temps de propagation identiques (parallèle ou série). En agissant sur la grille arrière du FDSOI, il n'est pas nécessaire de complexifier le circuit pour y parvenir. La rétroaction peut être faite grâce à un comparateur de phases simplifié s'il s'agit de signaux devant arriver au même moment.

3.3. Conclusion

La technologie FDSOI permet, par rapport à la technologie CMOS Bulk 28nm, soit de travailler à une fréquence de fonctionnement 30% supérieure à consommation identique, soit de réduire de 30% la consommation à fréquence identique. En outre, cette technologie possède une consommation statique plus faible que les nœuds technologiques précédents et une variabilité plus faible qu'une technologie Bulk de même dimension, diminuant ainsi la taille des blocs analogiques en maintenant leurs performances.

Dans le cas de l'architecture de la PLL présentée au sous chapitre suivant, la variabilité du temps de propagation des phases est importante. Pour diminuer cette variabilité, la technologie doit être rapide et avoir peu de variabilité. La technologie FDSOI est un excellent candidat (mismatch temporel = variabilité x temps de propagation). De plus si une calibration d'un quelconque paramètre électrique est nécessaire, la grille arrière du FDSOI permet de modifier la tension de seuil des transistors sans ajouter aucun transistor et donc sans modifier l'architecture du bloc concerné.

4. Architectures de synthèse de fréquences

4.1. Introduction

Une synthèse de fréquences est un circuit qui crée un signal à une certaine fréquence (normalement paramétrable) à partir d'une fréquence de référence (habituellement fixe). La référence est souvent un oscillateur à quartz, qui est une référence de fréquence de bonne qualité, stable à moyen ou long terme (sans ou avec régulation en température du quartz). Le signal de sortie est créé par un oscillateur commandé qui possède des performances moins bonnes mais qui seront améliorées grâce à la référence.

Dans ce paragraphe figurent les principaux types de synthèse de fréquences, les architectures et oscillateurs existants, et leurs performances. Sur la partie concernant les PLL, les architectures les plus utilisées pour les applications basse consommation seront détaillées, et leurs avantages et inconvénients seront rappelés. Et finalement seront expliqués les chemins qui ont conduit au choix de l'architecture à commutation de phases pour ce travail de thèse, et l'état de l'art spécifique à cette architecture.

4.2. Architectures de synthèse

4.2.1. Types d'asservissement

Il existe plusieurs types de synthèse de fréquences selon le type de verrouillage. Dans la plupart des cas une erreur est mesurée et convertie en valeur correctrice pour l'oscillateur (ou la chaîne de délai) commandé(e). La référence est habituellement un oscillateur à quartz au bruit extrêmement faible. Quant au verrouillage par injection (Injection Locking), ce n'est pas un système asservi mais en boucle ouverte, il est néanmoins possible d'y ajouter une PLL ou FLL pour assurer le fonctionnement désiré.

4.2.1.1. PLL (Phase Locked Loop) :

Appelée aussi boucle à verrouillage de phase, la PLL se verrouille par rapport à l'erreur de déphasage entre un signal de référence et l'oscillateur interne (cf. Figure 28). Le bruit du signal généré par la PLL est très propre en bruit de phase car, dans la bande passante, il est égal au bruit de la référence additionné de $10\log(N)$, sauf si d'autres blocs de la boucle sont plus bruyants. N est le facteur de multiplication en fréquence entrée-sortie, ce qui correspond au facteur de division de la rétroaction dans la boucle. A l'accrochage, on obtient en sortie $F_{VCO} = N \cdot F_{REF}$. En dehors de la boucle, le bruit du signal généré est égal au bruit de l'oscillateur, voir Figure 29. Son fonctionnement et ses autres caractéristiques (temps de démarrage et pas de fréquence) sont connus et ont déjà été largement étudiés [28].

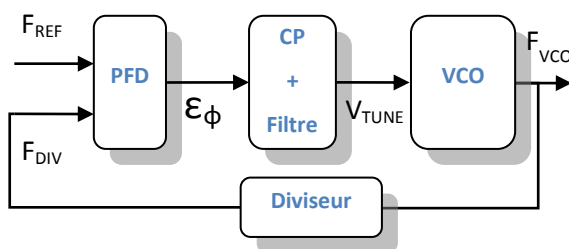


Figure 28 - Composants d'une PLL Analogique

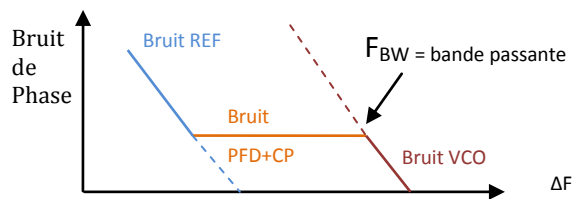


Figure 29 - PLL avec boucle de calibration

4.2.1.2. FLL (Frequency Locked Loop)

La FLL, aussi dénommée boucle à verrouillage de fréquence, asservit son oscillateur par rapport à une erreur de fréquence [29], [30]. La mesure de fréquence peut être analogique (Convertisseur fréquence – tension) ou numérique (comptage de périodes). Les performances en bruit de phase sont mauvaises car la phase n'est pas du tout asservie quand il y a verrouillage de fréquence (erreur de phase de +/- une période). Cette solution est donc à exclure d'emblée.

4.2.1.3. DLL (Delay Locked Loop)

La boucle à verrouillage de délai, ou DLL, est asservie par rapport à une erreur de délai (déphasage). C'est un système du premier ordre, implicitement stable [31]. L'architecture la plus usuelle veut que la chaîne de délais soit pilotée par la fréquence de référence et la sortie asservie pour être en phase avec l'entrée. Le nombre de cellules dans la chaîne est égal à la valeur de multiplication de fréquence, qui est obligatoirement une valeur entière. Toutes les sorties de la chaîne de délais sont combinées pour former une fréquence plus élevée (sortie de synthèse). Le bruit de phase est comparable à une PLL (à part la génération de raies parasites à la fréquence de la référence) et la consommation est légèrement plus élevée à bruit équivalent due à la recombinaison des fronts.

4.2.1.4. ILO (Injection locked oscillator)

Le verrouillage par injection (de courant) est une méthode qui permet de synchroniser un oscillateur à haute fréquence sur une des raies spectrales d'un signal à plus basse fréquence [32]. C'est une méthode de synthèse de fréquences sans rétroaction, il n'y a donc pas d'assurance que le verrouillage de l'oscillateur est fait sur la raie voulue sauf à ajouter une FLL ou une PLL qui agit sur la fréquence d'oscillation libre de l'oscillateur. Le bruit de phase d'un oscillateur verrouillé par injection est très bon car il reprend celui de la référence à $10\log(N)$ près (N étant la valeur de multiplication entière). L'ILO est principalement utilisé à des fréquences millimétriques où la réalisation d'un diviseur est très difficile ou consommerait trop pour que la solution globale soit viable.

4.2.1.5. Conclusion

La synthèse de fréquences à base de DLL permet une intégration plus élevée et une stabilité accrue mais produit des raies spectrales et ne permet pas de produire de multiplication fractionnaire de fréquence.

L'ILO est une méthode prometteuse qui nécessite d'injecter la référence dans l'oscillateur et d'ajouter une PLL pour vérifier la synchronisation sur la raie spectrale correcte, et ne permet pas de produire de multiplication de fréquence fractionnaire.

La PLL possède des avantages indéniables en termes de bruit de phase et de consommation, de pas de fréquence et de raies spectrales parasites.

Pour les raisons citées, les travaux de recherches ont été orientés vers une synthèse de fréquences à base de PLL, si possible avec une architecture faible consommation et fractionnaire.

4.2.2. Types de PLL

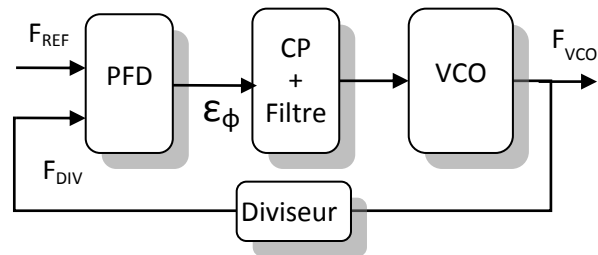


Figure 30 - Composants d'une PLL Analogique

Une PLL est un système bouclé (cf. Figure 30) qui asservit un oscillateur à fréquence variable pour qu'il produise une fréquence multiple de la fréquence de référence fournie à l'entrée. La valeur de multiplication peut être entière ou fractionnaire. L'oscillateur peut être contrôlé en fréquence par une tension (VCO) ou un code digital (DCO). Si l'entrée est une tension, elle est fabriquée la plupart du temps par une pompe de charges (CP) dont la sortie est filtrée par un filtre passe bas (LPF). Si l'entrée est numérique c'est un filtre numérique qui calcule les bits de commande. Le comparateur de phases (PFD) calcule l'erreur de phase entre la fréquence de référence (F_{ref}) et la fréquence divisée (F_{div}). En changeant la valeur N du diviseur, la PLL va asservir la fréquence de sortie pour devenir N fois la fréquence de référence. Selon le degré de numérisation des blocs constitutifs de la PLL, il y a 3 types de PLL possible au niveau hardware [33]:

4.2.2.1. LPLL

LPLL (Linear PLL) ou PLL analogique : le comparateur de phases est un multiplieur analogique, dont la valeur moyenne commande l'oscillateur, le filtre peut être passif ou actif. Tous les composants traitent des signaux analogiques : fréquence, phase ou tension analogique. Le comparateur analogique a le défaut d'être sensible à l'amplitude et au rapport cyclique des signaux d'entrée, ce qui en fait une solution plus bruyante que le comparateur « numérique » de la DPLL, qui n'est sensible qu'aux fronts.

4.2.2.2. DPLL

(D)PLL (Digital PLL) ou PLL « mixte » : Le comparateur de phases et le diviseur sont à base de circuits numériques, mais à aucun moment il n'y a d'échantillonnage. Les signaux de sortie du comparateur sont binaires mais transportent des informations analogiques (durée) qui modulent le courant de la pompe de charge. La conversion de différence de phase vers la commande en tension (ou courant) de l'oscillateur se fait donc par le biais de portes logiques mais l'information analogique

est préservée. Le bruit est meilleur car le détecteur de phase est uniquement sensible à la phase des fronts (montants ou descendants) des signaux à comparer (référence et sortie diviseur)

4.2.2.3. ADPLL

ADPLL (All Digital PLL) ou PLL numérique : tous les composants sont à base de circuits numériques pour gagner en taille et/ou en consommation, souvent au prix des performances. Les ADPLL utilisent un TDC pour effectuer la mesure de déphasage du VCO par rapport à la référence, et le filtrage ainsi que la commande de l'oscillateur sont numériques. Le TDC dépense beaucoup d'énergie (plusieurs milliwatts) pour que ses performances (résolution, dynamique, fréquence) soient acceptables. La plupart des TDC sont conçus comme une ligne à retard ou un oscillateur en anneau, en plus de l'oscillateur commandé de la PLL.

Le filtre analogique est remplacé par un filtre numérique beaucoup moins encombrant. Il permet beaucoup plus de flexibilité qu'un filtre analogique, tel qu'un filtre polyphase [10] ou adaptatif [34].

Le VCO est renommé DCO (Digitally Controlled Oscillator) car sa commande est numérique. La résolution (pas en fréquence) du DCO est déterminée par le nombre de bits de sa commande, et peut être limitée par la finesse de la technologie. Pour atteindre les performances de bruit de phase exigées par les standards de communications radiofréquences, un modulateur Sigma Delta est utilisé [35], [36], [37], [38]. Ce modulateur Sigma Delta doit fonctionner à une fréquence bien plus élevée que la fréquence de comparaison et sa consommation n'est pas négligeable.

4.2.2.4. Conclusion

L'utilisation d'une bande passante élevée permet de réduire les inconvénients de la DPLL (autodécharge de la capacité, grande surface de filtre de boucle analogique passif, bruit de courant de la pompe de charge). Réciproquement, d'un point de vue de la surface d'intégration, l'ADPLL à faible bande passante possède certains avantages par rapport à une DPLL à filtre passif, qui s'estompent à haute bande passante (>1MHz). Un filtre de boucle passif peut occuper une faible surface (<0,01mm²) si le courant de CP est faible (<50μA) et que la densité des capacités et résistances de la technologie utilisée le permet (>5fF/μm² et >1000Ω/carré).

Concernant la consommation, l'optimisation des blocs de chaque (DPLL et ADPLL) peut jouer en faveur de l'une ou de l'autre : le PFD consomme moins d'un TDC mais la pompe de charge de la DPLL consomme plus qu'un filtre numérique. Quand la fréquence est élevée par contre, à cause

d'une grande bande passante par exemple, le modulateur sigma delta, le TDC et le filtre numérique de l'ADPLL consomment d'avantage et le filtrage du bruit est plus difficile.

L'architecture qui est la plus prometteuse en termes d'économie d'énergie et qui devrait atteindre plus facilement les spécifications de bruit de phase à grande bande passante apparait donc comme étant la DPLL.

4.2.3. Types d'oscillateurs

Un oscillateur commandé de PLL est un circuit oscillant à une fréquence dépendant de la commande (en tension ou numérique). Les types d'oscillateurs sont :

4.2.3.1. Oscillateur à base de circuit résonnant LC

Un oscillateur LC (Figure 31) est un circuit résonnant à la fréquence $= \frac{1}{2\pi\sqrt{LC}}$. Pour que l'oscillation soit entretenue un amplificateur monté en résistance négative compense les pertes des composants passifs (R).

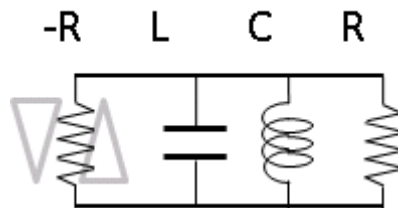


Figure 31 – Schéma d'oscillateur LC avec résistance parasite et résistance négative

L'ordre de grandeur de la surface d'une inductance de 2,5 nH est de 150µmx150µm en technologie FDSOI et la capacité nécessaire pour former le résonateur LC est d'autant plus grande que la fréquence d'oscillation est basse. Si il n'y a pas d'inductance dans le DK (Design Kit) de la technologie utilisée ou qu'elles ne correspondent pas aux besoins, il est nécessaire d'effectuer des simulations électromagnétiques (de type HFSS) pour la caractériser (pertes résistives dans les fils et magnétiques dans le substrat). Les oscillateurs LC sont plus simples à utiliser dans les circuits de communications radiofréquences car ils possèdent le meilleur rapport bruit de phase/consommation, ou FOM (Figure de mérite, cf. relation 7) (Voir Tableau 5). Le coefficient de qualité (Q) exprime les pertes dans l'oscillateur ; plus il est élevé, moins l'amplificateur (résistance négative) consommera et moins l'oscillateur sera sensible au pushing/pulling.

Tableau 5 - Etat de l'art (meilleure FOM) de chaque topologie d'oscillateur en sub-mW

Caractéristiques Unités	Alim. Volt	Puiss. mW	Freq MHz	Techno nm	PN ¹ dBc/Hz	BW ² PN dBc/Hz	Taille mm ²	Jitter psRMS	FOM dB
LC-tank [39]	1	0,35	2300	90	-121@3M	-72	0,25	1,65	-183
Ring [40]	0,8	0,7	700	32	-113@1M	-100	0,0025	3,3	-171

Active L [41]	1,8	1,16	4950	200	-81@500k	NC	0,0005	NC	-160
---------------	-----	------	------	-----	----------	----	--------	----	------

PN = Phase noise : Bruit de phase
 BW = Bandwidth : bande passante de la PLL
 BW PN: bruit de phase dans la bande

$$FOM = 10 \log \left(\frac{Power}{1mW} \right) + 20 \log \left(\frac{\Delta F}{F_{osc}} \right) + Lf \quad (7)$$

4.2.3.2. Oscillateur à base d'inductance active

L'idée de l'oscillateur à inductance active est de remplacer l'inductance (ayant une surface non-négligeable) par un montage électronique qui émule une inductance [41]. Un girateur inverse la partie complexe d'une impédance, et simule ainsi une inductance à partir d'une capacité. La FOM de ce type d'oscillateur est communément inférieure de plus de 20dB (cf. Tableau 5).

4.2.3.3. Oscillateur à base d'inverseurs en anneau

Un oscillateur en anneau (RO : Ring Oscillator, cf. Figure 32) occupe beaucoup moins de surface et possède une FOM moins bonne qu'une inductance mais meilleure que l'oscillateur à base d'inductance active [40]. Si l'oscillateur est single-ended (sortie non différentielle) et qu'il n'y a aucune rétroaction interne aux cellules-décalai, le pushing (sensibilité aux variations d'alimentation) est très élevé. Cela peut être corrigé en ajoutant un régulateur de tension sur l'alimentation ou une rétroaction sur un autre paramètre influençant ce délai.

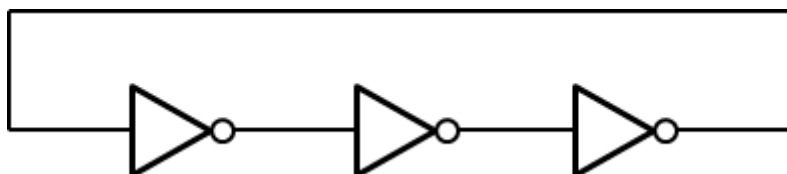


Figure 32 - Oscillateur en anneau single-ended (trois inverseurs)

L'oscillateur en anneau possède d'autres qualités comme la rapidité de démarrage (facteur de qualité faible), la simplicité de mise en œuvre et de contrôle (pas de capacité variable obligatoire si contrôle en courant ou tension), la possibilité de générer des phases (réparties régulièrement dans le temps) qui peuvent permettre une division fractionnaire (voir section Diviseur à commutation de phases) et un gain / dynamique en fréquence élevé (utile pour couvrir un grand nombre de bandes de fréquences).

4.2.3.4. Conclusion

Pour respecter les objectifs fixés de faible coût de production, la faible surface occupée par un oscillateur en anneau semble plus adéquate. La FOM est moins bonne (-150 à -170dB) qu'un oscillateur LC (-180 à -200dB) mais permet de respecter les contraintes fixées, avec une FOM de -

160dB : consommation de 1mW et bruit de phase de -90dBc/Hz à 1MHz d'offset de la porteuse à 2,45GHz. De plus, au début des travaux de thèse, aucune inductance ni aucun varactor (capacité variable) n'était intégré au DK, ni n'avait encore été modélisé en FDSOI. La mise en œuvre d'un oscillateur en anneau était donc beaucoup plus rapide.

4.2.4. PLL Entière Vs Fractionnaire

4.2.4.1. PLL entière

Dans une PLL entière, la fréquence de sortie est un multiple entier de la fréquence de comparaison, et le pas de fréquence de sortie est égal ou supérieur à la fréquence de comparaison. Pour les standards de communications radiofréquences à bande étroite, la largeur des canaux, i.e. le pas de résolution de la PLL, est faible devant la fréquence de la porteuse (ex : Bluetooth LE : 2MHz versus 2450MHz). Si on utilise une PLL entière, la fréquence de comparaison est très faible (égale au pas du canal) et la bande passante de la PLL est donc très faible (fréquence maximale à laquelle le VCO est corrigé), car la bande passante doit être inférieure d'environ une décade à la fréquence de comparaison pour assurer la stabilité du système (8).

$$\begin{aligned}
 F_{OSC} &= F_{REF} * Diviseur \\
 F_{BW} &\leq 0.1 * F_{REF} : F_{REF} = N * F_{CANAL}
 \end{aligned}
 \tag{8}$$

4.2.4.2. Bande passante élevée et bruit de phase du VCO

Une faible bande passante de PLL induit un filtre de boucle intégré difficilement intégrable sur silicium et filtre peu le bruit du VCO (interne ou lié à l'alimentation) aux fréquences supérieures à la bande passante. La limite haute de la bande passante dans une PLL entière est la fréquence de référence (plus de diviseur en entrée) qui est elle-même limitée par le pas de canal. A faible fréquence de comparaison d'autres phénomènes peuvent apparaître (autodécharge du condensateur de filtrage). Une des solutions pour diminuer ces phénomènes est d'augmenter la bande passante de la PLL, ce qui nécessite une augmentation de la fréquence de comparaison dans la même proportion. Augmenter la bande passante permet aussi de relâcher les contraintes de bruit de phase du VCO, car son bruit est mieux filtré, voir Figure 33 [42].

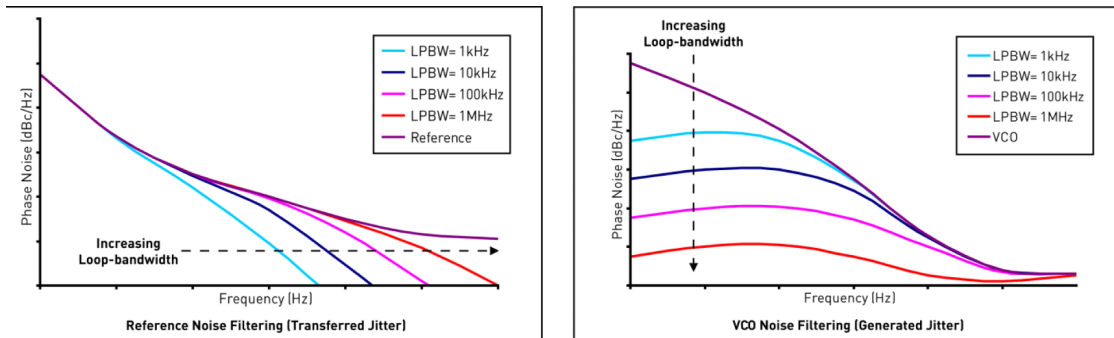


Figure 33 - Evolution du bruit de phase en fonction de la bande passante

4.2.4.3. PLL Fractionnaire

Pour diminuer l'espace occupé par le filtre LPF et diminuer l'influence du VCO sur le bruit total, il faut encore augmenter la bande passante de la PLL. Pour cela il faut s'affranchir du fait que la fréquence de canal est un multiple entier de la fréquence de comparaison, et donc il faut parvenir à effectuer une division fractionnaire de la fréquence de sortie (9). Deux implémentations sont possibles pour réaliser cette division fractionnaire :

$$F_{OSC} = F_{REF} * (Entier + Fraction) \quad (9)$$

La première possibilité est d'utiliser un diviseur à modulo ou $N/N+1$; cela correspond à passer rapidement entre deux modes de division entiers [43]. Le filtre de boucle filtre l'erreur de division fractionnaire sur plusieurs cycles. Toutefois, si la PLL a une grande bande passante, cette erreur ne sera pas filtrée et il y aura génération de « spurious » autour de la fréquence de sortie (voir section suivante).

La deuxième possibilité est de retarder l'arrivée du signal de sortie du diviseur, d'une fraction de période du VCO, avant l'entrée du comparateur de phases [44]. Cette solution peut également générer des spurious mais seulement si les fractions de période du VCO ont une erreur de phase assez élevée pour dépasser le bruit de phase en sortie (voir section suivante).

4.2.4.4. Conclusion

Nous avons constaté que plus la bande passante de la PLL est élevée, et moins le bruit de phase du VCO était prépondérant dans le bruit de phase total. Toutefois, pour que l'augmentation de la bande passante se transforme en amélioration du bruit de phase, il faut que le bruit dans la bande soit plus faible, c'est à dire le bruit du PFD et de la CP. Il faudra donc les optimiser convenablement.

Après avoir vérifié qu'une PLL fractionnaire était la plus à même de respecter les spécifications fixées pour ce travail de thèse, il nous reste à examiner quel type d'architecture est la plus économe en surface et énergie. Elles sont présentées dans le paragraphe suivant.

4.2.5. Diviseurs fractionnaire : modulo variable Vs commutation de phases

4.2.5.1. Diviseur fractionnaire à modulo variable

Les PLL fractionnaires les plus connues et utilisées sont à base de diviseur N/N+1 ou modulo variable. Un diviseur à modulo variable est un diviseur à plus de deux valeurs de division possibles (Ex : 3/4 ou 7/8), il existe des diviseurs à 3 ou 4 valeurs différentes. Un contrôleur commande combien de cycles sont effectués dans chaque mode de division. Un diviseur entier programmable vient augmenter la valeur de division totale pour former le diviseur fractionnaire de la PLL (voir Figure 34):

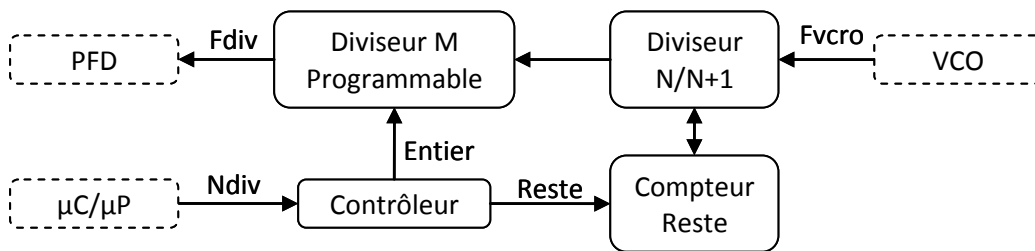


Figure 34 - Schéma bloc du diviseur fractionnaire à base de modulo et blocs en relation directe

Pour obtenir une division fractionnelle de la fréquence VCO, on moyenne sur P périodes (de Fref) plusieurs valeurs de division entière différentes. Par exemple, pour obtenir une division fractionnelle $N + \alpha$ (N entier et α fraction entre 0 et 1), on effectue une division par N/N+1 avec un rapport $\alpha (=F/P)$, soit N+1 pendant F périodes sur P (et N le reste du temps : F-P périodes) (10)(10). La division fractionnaire est donc vraie en moyenne mais pas en instantanée.

$$F_{OSC} = F_{REF} * \frac{((N + 1) * F + N * (F - P))}{P} \quad (10)$$

4.2.5.2. Erreur de division fractionnaire

Cette variation dans le temps du facteur de division est en fait une « erreur de division fractionnaire » générée à la comparaison à F_{REF} (cf. Figure 35). Celle-ci se retrouve à l'entrée du VCO, et par transposition de fréquence, fait apparaître des raies spectrales (spurious) à Fref et ses sous multiples autour de la porteuse [45]. Les spurious sont filtrés par la PLL si la bande passante est assez

faible, jusqu'à 1/100 de la fréquence de comparaison [38]. Mais cela diminue grandement le temps de réponse de la PLL (délai d'accrochage) ainsi que la réjection du bruit de phase par la boucle.

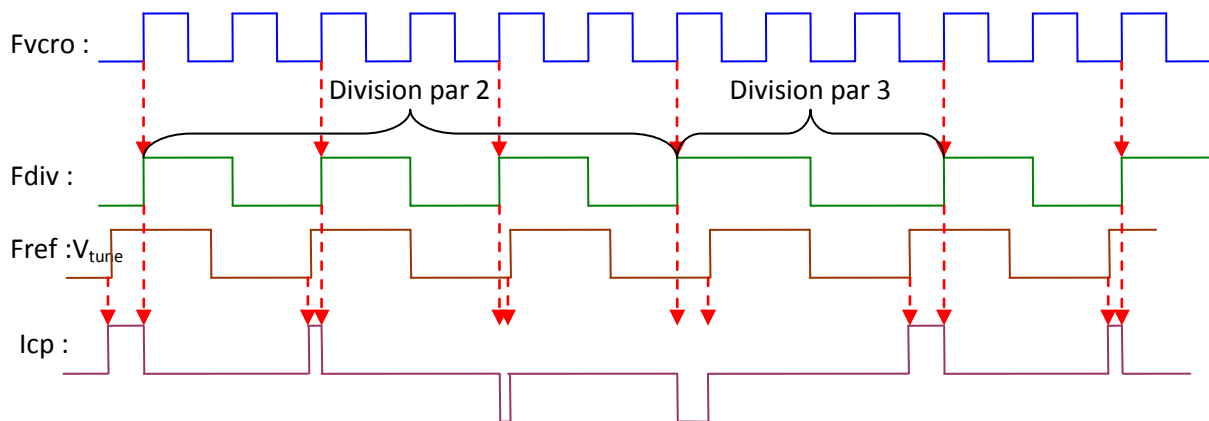


Figure 35 - Fonctionnement temporel du diviseur N, N+1 : division par 2,25

4.2.5.3. Solution sigma-delta

En ajoutant un modulateur Sigma-Delta, on rend aléatoire l'apparition des erreurs de fraction de la division fractionnaire. On met ainsi en forme le bruit pour qu'il s'éloigne dans les hautes fréquences, là où il sera plus simple à filtrer par le filtre de boucle. Ainsi la fréquence de coupure peut être plus élevée et le temps de réponse plus faible, mais toujours moindre qu'avec une PLL à division entière [45].

4.2.5.4. Solution compensation de l'erreur fractionnaire

Il est possible d'ajouter une compensation dans la boucle (sur le courant de la pompe de charge par exemple) pour compenser l'erreur de comparaison fractionnelle à chaque période Fref. Cela permet de relâcher les contraintes sur le filtrage et d'améliorer les performances [46], [47], [48].

4.2.5.5. Solution diviseur fractionnaire à phases (FPD)

L'idée à l'origine du FPD est de rendre la division parfaitement fractionnaire à chaque comparaison, on supprime ainsi les spurious fractionnaires. Il est possible d'effectuer cette division parfaitement fractionnaire en ajoutant un délai multiple d'une fraction de période Fvcro à chaque période de Fdiv. Cette fraction de délai peut provenir des phases d'un VCRO (phases pures ou recombinaisons [35]), d'une ligne à retard calibrée [49], d'un VCRO verrouillé par injection [36], ou encore en intégrant le TDC dans le VCRO [37]. Le diviseur fractionnaire à phases est peu utilisé en radiofréquence car il nécessite l'utilisation d'un VCRO, au bruit médiocre, ou d'un TDC dont la consommation s'ajoute à celle du VCO.

Quatre phases du VCO :

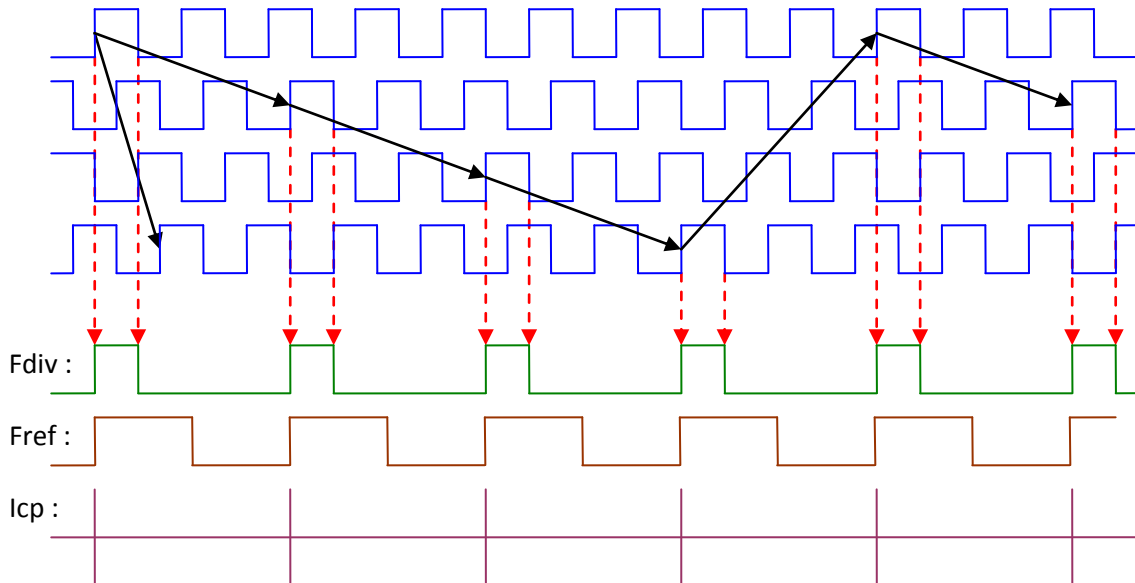


Figure 36 - Fonctionnement temporel du diviseur à commutation de phase : division par 2,25

On peut voir sur la Figure 36 que le signal F_{div} est parfaitement fractionnaire, grâce à l'utilisation des phases du RO pour la resynchronisation du signal de division entière. Comme la division est toujours fractionnaire, cela n'engendre pas d'injections cycliques de bruit par la pompe de charge à l'entrée du VCO, supprimant ainsi les spurious. Mais cette architecture possède d'autres défauts connus : la création de spurious si les phases ne sont pas parfaitement réparties temporellement. Il est néanmoins possible de calibrer les phases avec une certaine précision, qui est proportionnelle aux spécifications de spurious que l'on souhaite atteindre. Les recherches sur les techniques de calibration sont peu nombreuses comparées au diviseur modulo mais existent [50], [51], [52], [53]. Tous les détails de cette architecture sont présentés dans le paragraphe suivant.

4.2.5.6. Conclusion

Le diviseur entier $N/N+1$ parvient à faire une division fractionnaire en moyenne seulement, ce qui produit des injections de courant d'erreur de division fractionnaire par la pompe de charge sur le filtre de boucle alors que la PLL est stabilisée. Ce n'est pas le cas de l'architecture à commutation de phases dans le cas idéal, c'est-à-dire, si les phases de l'oscillateur en anneau sont parfaitement réparties dans le temps. Mais les variations technologiques font qu'elles ne le sont jamais parfaitement. Des techniques de calibration ont été publiées pour limiter les spurious de cette architecture mais elles complexifient l'architecture et augmentent la consommation en ajoutant un modulateur sigma delta ou un filtre numérique sur un interpolateur de phase (mélangeur temporel).

4.3. La PLL à diviseur fractionnaire sur phase (FPD)

Différentes équipes dans le monde ont créé ou amélioré différentes implémentations possibles de la PLL à commutation de phases pour pallier aux problèmes de la division fractionnaire. D'autres équipes les ont revisités en essayant d'améliorer certaines architectures de (D)PLL avec ou sans TDC. Ces architectures utilisent toutes des TDC implicites ou explicites (internes ou externes à l'oscillateur). Ainsi le vocabulaire utilisé dans les publications ou brevets pour définir ces architectures est très varié : TDC-Less [35], Embedded-TDC [37], Phase-selection [54], Phase-synthesis [55], Phase-switching [52], Clock-shortening [56], Edge-combining [57], Phase-sampling [58], Phase-quantizer [45] ou Phase-interpolator [51]. On trouvera, dans la littérature, des combinaisons croisées de ces termes. Les différentes publications existantes, qui peuvent se réunir sous le terme générique de PLL à commutations de phases, sont présentées dans la section suivante.

4.3.1. Différents chemins menant à la même architecture

Dans une PLL mixte (CP + LPF analogique) à division fractionnaire à base de modulateur sigma-delta, on peut diminuer la consommation en remplaçant ce dernier par un DTC (Digital to Time converter). Ce bloc, pouvant générer des délais sous-multiples de la période VCO, permet d'effectuer une division fractionnaire instantanée.

En partant d'une PLL numérique à base de TDC et DCO, on peut améliorer la surface occupée et la consommation en intégrant le TDC dans le DCO, et on obtient ainsi une PLL à TDC intégré. Le bruit dans la bande est élevé du fait que la faible résolution temporelle du TDC est insuffisante pour une commande tout numérique.

Dans les deux cas, la résolution temporelle de la division fractionnaire est proportionnelle à la fréquence et au nombre de cellules du VCRO. En intégrant ce VCRO à TDC dans une PLL mixte (CP + LPF analogique), on disposera d'une PLL à division fractionnaire par phases (FPD).

4.3.2. Historique et état de l'art de la PLL FPD

L'état de l'art des PLL à commutation de phases est résumé dans les Tableaux Tableau 6 et 7. Les brevets sont intercalés au milieu des publications pour rendre compte de l'ordre chronologique de leur apparition.

Un brevet [56] déposé en 1991 par Level One communications met en œuvre pour la première fois un multiplexeur pour sélectionner une par une les phases d'un oscillateur en anneau ainsi qu'un diviseur entier à sa sortie pour effectuer une division fractionnaire dans une PLL.

Une publication de PLL de 1999 [55] par Texas Instruments, possédant un synthétiseur de phases et de fréquences utilisant les 32 phases du VCO, permet de générer des fréquences variant dans le temps (spread spectrum) ou encore toutes fréquences multiples de celle de la PLL. Dans certains cas, la fréquence synthétisée est seulement une moyenne temporelle de la fréquence. Le synthétiseur sera réutilisé ultérieurement pour créer une PLL à commutation de phases.

En 2000, une PLL à commutation de phases est publiée [50]. La sortie du VCO, après division par un nombre entier, commande des « flip-flop », qui resynchronisent le signal divisé par les fronts montants des phases internes du RO. Cette architecture permet d'effectuer une division fractionnaire par commutation de phases avec des portes logiques relativement lentes et un RO rapide, la technologie utilisée étant du CMOS 0,35 μ m. De plus, cette PLL dispose d'une calibration du mismatch de toutes les phases pour supprimer les spurious. Cette architecture permet également de fournir des signaux I/Q en parfaite quadrature.

Un brevet de diviseur fractionnaire à commutation de phases [44] a été déposé en 2000 par Lucent Technologies. Le contrôleur du diviseur est implémenté avec une machine d'état afin de contrôler le multiplexeur qui sélectionne les phases d'un oscillateur en anneau. Puis un diviseur entier en sortie du multiplexeur vient achever ce diviseur fractionnel. La machine d'état est contrôlée par le facteur de division (fixe en fonctionnement établi), les phases du RO et la sortie du diviseur entier. Cette architecture a l'avantage d'être simple, fiable et fonctionnelle.

La même année, un autre brevet [59] sur la même architecture à sélection de phases pour obtenir un diviseur non-entier a été déposé par Siemens. Un brevet [54] avec la même architecture que [44] est déposé en 2003 par Agere Systems. La différence est le niveau de détail de l'implémentation et le fait de l'intégrer dans une PLL.

En 2004, un brevet est déposé par IBM [60] sur une architecture de diviseur à commutation de phases séparant les phases paires des phases impaires pour effectuer la division fractionnaire. En 2005, une PLL à commutation de phases [52] a été publiée, en utilisant le même principe de fonctionnement que précédemment. Toutefois, pour limiter le mismatch et éviter une calibration des phases, celles-ci sont générées par 2 niveaux de diviseurs par 2, qui à chaque division créent des phases en quadrature. Les 4 sorties du VCO fonctionnant à 4 fois la fréquence de sortie (10GHz) sont successivement divisées pour créer 8 puis 16 phases réparties dans le temps à une fréquence de

2,5GHz. La deuxième différence d'implémentation est que le diviseur entier asynchrone est disposé à la sortie d'un multiplexeur haute fréquence qui raccourcit la période d'une fraction de phase ($22,5^\circ$) sans « glitch » à chaque fois qu'il reçoit l'ordre de commutation. Un compteur envoie P fois le signal de commutation toutes les N phases pour produire P périodes raccourcies d'une fraction de phase et N-P périodes normales.

En 2006, Texas Instruments publie un article sur une PLL à commutation de phases [53] à partir d'une modification de l'architecture proposée par [55]. Néanmoins aucune réalisation n'a été proposée, seule une étude de faisabilité permet de valider ce concept sans simulation de spurious, ni de bruit de phase.

En 2010, une DPLL à TDC intégré [37], l'équivalent numérique de la PLL à commutation de phases, innove cette fois avec une interpolation des phases du VCO « single-ended » à 5 cellules pour multiplier par 2 le nombre de phases disponibles et créer un TDC à 20 niveau de quantification. Pour le DCO, des résistances de contre-réaction filtrent le mismatch des phases et un modulateur sigma-delta affine la précision de la commande en courant. La même année, une autre DPLL à TDC intégré [35], mais cette fois avec des cellules où un LDO peut alimenter un nombre variable d'inverseurs (commande numérique de switchs) pour moduler numériquement la fréquence du RO.

Toujours en 2010, une publication de PLL [49] est proposée avec une architecture de diviseur à commutation inédite : après le VCO c'est une DLL qui fabrique les phases pour le diviseur à commutation de phases (multiplexeur + diviseur entier). Ce diviseur fractionnaire permet l'utilisation d'un oscillateur LC, aux meilleures performances de bruit de phase, au sein d'une PLL fractionnaire à large bande passante, optimisant ainsi les spurious, le bruit de phase et la consommation.

En 2011, une PLL à commutation de phases [61] met en œuvre une calibration externe des phases de l'oscillateur (cellules différentielles à gain variable) à l'aide de filtres numériques FIR et d'un Phase Blender (mélangeur temporel différentiel des phases).

Un brevet est déposé en 2012 par Texas Instruments [62], sur la même architecture que [15], en utilisant un Phase Blender au lieu d'un multiplexeur. La différence se situe au niveau implémentation, pas au niveau fonctionnel.

Enfin en 2013, une variante de [35] effectue une modulation FSK indirecte en utilisant un Phase Interpolator (PI) [51]. Ce PI est commandé numériquement par un modulateur sigma-delta qui reçoit par le biais d'un filtre FIR gaussien le train de bits de données. Pour commander le DCO, le filtre numérique reçoit l'erreur d'un comparateur de phases Bang-Bang à 5 bits.

Tableau 6 - Récapitulatif des performances des PLL fractionnaires à phases publiées

	[50]	[52]	[53](simulation)	[49]
Année	2001	2005	2006	2010
Architecture	Div+ Ph resync + CP+LPF	phase create & select + Div + CP + LPF	Phase synthesis + CP + LPF	DLL Phase generate +Mux+Div+CP+LPF
Spurious suppression	Oui	Non	Non	Non
Fréquence	1,7-1,9GHz	0,8-2,5GHz	250MHz	480MHz
In-Band Noise	-105dBc@100kHz	-110dBc@100kHz	-	-106dBc@100kHz
Out of band Noise	-110dBc@3MHz	-96dBc@1MHz	-	-100dBc@1MHz
Fract. Spurious	-55dBc@3,25MHz	-	-	-66dBc (Ref)
Jitter RMS	-	3,8ps	40ps pk-pk	5,8psRMS
FREF	25MHz	40MHz	4MHz	20MHz
BW _{PLL}	1MHz	750kHz	-	1,7MHz
Surface	0,48mm ²	0,06mm ²	-	0,1mm
Consommation	20mA 3,0V	10mA VCO 3,3V	-	2,5mA 1,2V
Technologie	0,35µm	120nm	130nm	90nm
FOM	-137dB	-148dB	-	-148dB

Tableau 7 - Récapitulatif des performances des PLL fractionnaires à phases publiées (suite)

	[37]	[35]	[61]	[51]
Année	2010	2010	2011	2013
Architecture	ΣΔDCO + TDC intégré	ΣΔDCO + TDC intégré	Dig. FIR PI+ Div + CP + LPF	FSK on PI + Div + CP + LPF
Méthode de suppression du bruit	Non	Non	Oui, FIR dans le PI	Non
Fréquence	600-800MHz	3GHz	1GHz	1GHz
Noise In-Band	-93dBc@1kHz	-	-106dBc@100kHz	-96dBc@100kHz
Noise Out of band	-83dBc@100kHz -98dBc@1MHz	-	-101dBc@1MHz	-110dBc@1MHz
Fract. Spurious	-45dBc@1,6MHz	-	-66dBc (Ref)	-
Jitter RMS	20-30ps RMS	1,4psRMS	-	2,4psRMS
F _{REF}	2-40MHz	-	32MHz	25MHz
BW _{PLL}	~qq 10kHz	-	3,2MHz	~200kHz
Surface	0,027mm ²	0,038mm ²	0,31mm ²	0,25mm ²
Consommation	2,66mA 1,1-1,3V	9,3mA	16,8mW	7,4mW
Techno	65nm	65nm	130nm	130nm
FOM	-156 à -151dB	-	-148dB	-161dB

4.3.3. Création Vs réutilisation des phases du VCRO

Certaines architectures n'utilisent pas les phases du RO pour effectuer la division fractionnaire mais les créent : à partir de diviseur en quadrature [52], d'une DLL [49], d'un RO verrouillé par injection du VCO, ou d'un interpolateur de phases [61], [51]. Les phases créées sont réparties dans le temps et l'une d'entre elles est sélectionnée par un multiplexeur, pour effectuer la division fractionnaire.

Il y a donc un gain en flexibilité sur le choix du type d'oscillateur si on crée les phases nécessaires à la division fractionnaire, mais une perte en économie d'énergie et en surface. Pour

notre application basse consommation, l'architecture à réutilisation de phases du VCRO est plus adaptée.

4.3.4. DPLL à TDC implicite Vs TDC explicite

Dans une DPLL, le TDC peut être explicite ou implicite. Il est explicite s'il possède sa propre base de temps comme une ligne à retard ou un oscillateur en anneau. Il est implicite si sa base de temps est commune à l'oscillateur (en anneau) de la PLL. Dans le cas du TDC implicite la résolution du TDC évolue proportionnellement à la période et au nombre de phases du VCRO, le TDC explicite peut aussi être verrouillé par injection ou par DLL. Le fait que les fractions évoluent en même temps que la fréquence permet de limiter les spurious. Le TDC implicite est avantageux en termes d'économie d'énergie mais oblige à utiliser un oscillateur en anneau pour la PLL, qui a de moins bonnes performances en bruit de phase, ce qui n'est pas très gênant pour notre application.

4.3.5. Diviseur à commutation Vs resynchronisation

Le diviseur entier peut être positionné avant ou après le multiplexeur. S'il est positionné après, c'est un diviseur à commutation de phases, s'il est avant c'est un diviseur à resynchronisation de phases.

4.3.5.1. Diviseur entier après multiplexeur : commutation de phases

Le diviseur fractionnaire à commutation de phases, dont une implémentation possible est donnée Figure 37, consiste à positionner un multiplexeur toujours actif sur les phases d'un RO, et le diviseur à valeur fixe en sortie du multiplexeur. Le multiplexeur doit passer d'une phase sélectionnée à l'autre et ne doit générer ni glitch ni saut de phase, sous peine d'erreur de comptage du diviseur. Après simulation et implémentation, un problème majeur a été soulevé, le multiplexeur étant activé en permanence, et sa capacité parasite étant élevée, il fait augmenter sensiblement la consommation totale de la PLL.

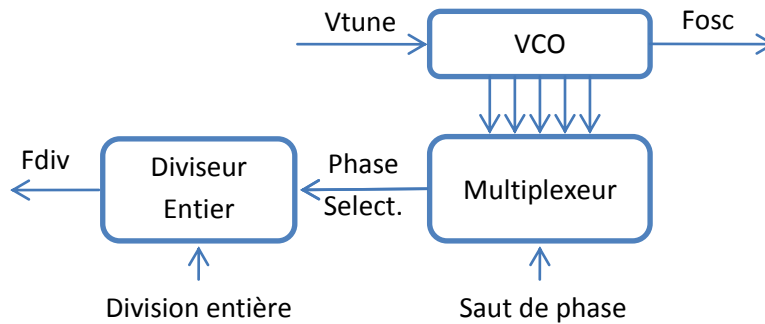


Figure 37 - Blocs fonctionnels du FPD à commutation de phases

4.3.5.2. Diviseur entier avant multiplexeur : resynchronisation par phase

En disposant le diviseur entier à la sortie du VCO, le multiplexeur n'est activé que pendant que 1 ou 2 périodes F_{vco} par période F_{ref} . Le multiplexeur est plus simple à concevoir car la synchronisation peut être étalée sur plusieurs périodes, et le risque de créer un glitch ou de rater une période est beaucoup plus faible.

Le fonctionnement du diviseur entier est identique à celui d'un diviseur à double modulo, qui varie avec un rapport proportionnel à la partie fractionnaire de la division. Le multiplexeur comporte un additionneur commandé à F_{div} qui calcule la phase du VCO à sélectionner. La valeur du saut entre les phases sélectionnées consécutivement au moment de la division fractionnaire, dépend de la fraction de division. Si le saut fait reboucler le compteur, un signal 'Carry' est envoyé au diviseur entier pour prolonger la division suivante d'une période VCO. Ce comportement est schématisé par la Figure 38.

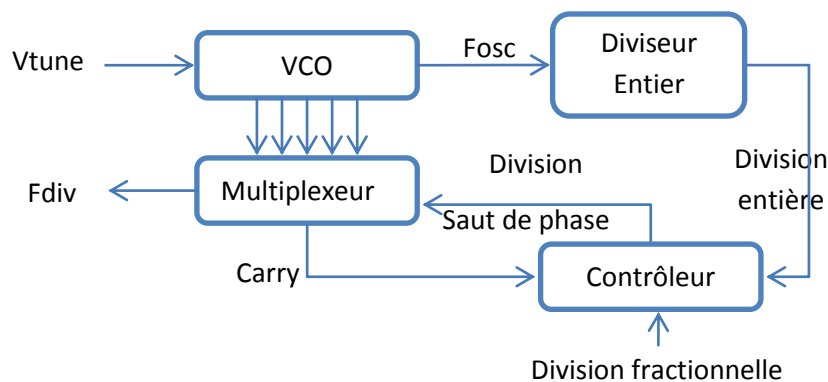


Figure 38 - Blocs fonctionnels du FPD à synchronisation par phase

4.3.6. Exemple d'implémentation FPD

Dans un système radio, le microcontrôleur en charge de commander la PLL envoie le facteur de division (N_{div}) au contrôleur du diviseur (voir Figure 39). Celui-ci calcule pour chaque cycle la

position du multiplexeur (fraction) et la valeur du compteur (partie entière). Le diviseur à commutation de phases utilise un compteur N reconfigurable dont l'entrée est fixe : la sortie du VCO (F_{vco}). La technologie FDSOI permet de réaliser ce compteur asynchrone avec des portes logiques offrant une consommation maîtrisée malgré une entrée à 2,5GHz et le fait qu'il compte par pas de 1. Sa sortie (Activation) est resynchronisée par une des phases sélectionnées par le multiplexeur. Ce multiplexeur doit minimiser le mismatch et le jitter des phases provenant du RO, car contrairement au bruit du diviseur entier, son bruit est ajouté au signal F_{div} .

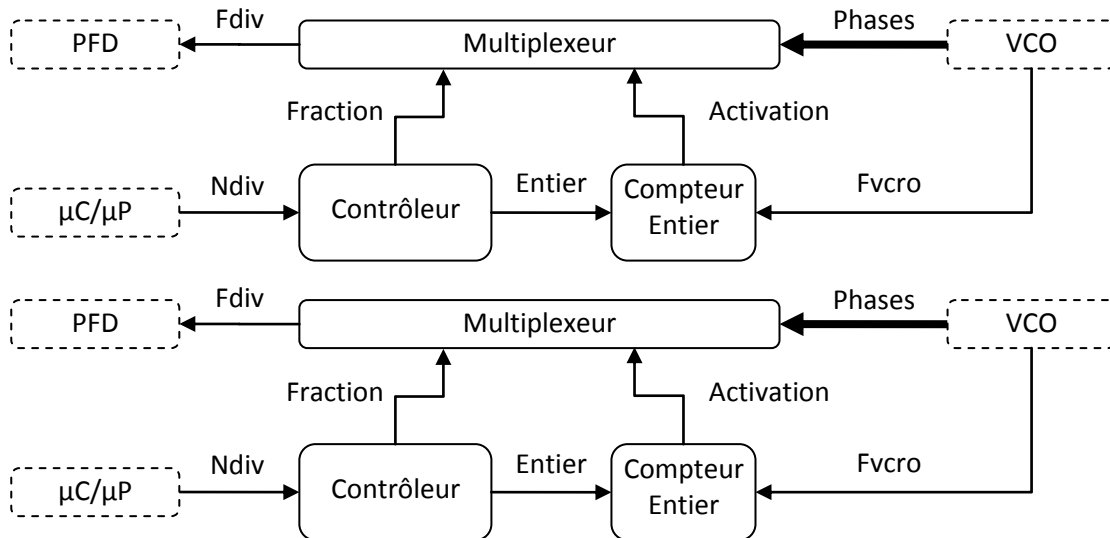


Figure 39 - Schéma bloc du diviseur à commutation de phases et blocs en relation directe

4.3.7. Amélioration de la base de temps

La division fractionnaire sans spurious demande de pouvoir générer un délai qui soit :

- un sous-multiple de la période du VCO ($1/N_{phases}$),
- ne soit pas plus bruité que le VCO,
- ajustable par multiples de la fraction minimum sans mismatch,
- varie proportionnellement à la période du VCO quand celui-ci change de fréquence.

Si la base de temps respecte ces contraintes, on parvient à réaliser une division fractionnaire parfaite qui ne produit pas de spurious. On peut alors augmenter la bande passante de la PLL jusqu'à $1/10$ (ou même jusqu'à $1/7$ dans certains cas) de la fréquence de référence/comparaison.

Si la base de temps ne respecte pas ces contraintes, il y aura génération de spurious visibles au-dessus du bruit de phase de la PLL. Ces spurious apparaissent car la division fractionnaire n'est plus assez bonne à chaque comparaison. C'est le cas que l'on rencontre le plus souvent dans la

réalité, à cause de la variabilité des technologies CMOS, même si les lignes et les transistors ont été parfaitement dimensionnés sur le layout.

Plusieurs astuces existent pour supprimer ces défauts, dont certaines ont déjà été publiées, comme la calibration des délais dans l'oscillateur en anneau [50], ou l'interpolation des phases issues de l'oscillateur [48]. Une des autres possibilités serait d'utiliser un DTC (Digital to Time Converter) à la place d'une DLL ou des phases d'un RO, avec une dynamique et une résolution suffisante pour effectuer la division fractionnaire sans générer de spurious. Dans notre application il faudrait une dynamique de 420ps et une résolution de 250fs. Cela permettrait l'utilisation d'un VCO à base d'inductance et réduirait la consommation du VCO pour le même bruit de phase.

4.4. Conclusion

Le choix de l'architecture de PLL à diviseur fractionnaire sur phase a été un processus long qui a commencé par le choix du type de synthèse de fréquences, en passant par le type de PLL et finissant par le type d'architecture de PLL mixte.

Il a fallu rechercher dans des publications qu'il était possible d'utiliser un oscillateur en anneau pour des applications à la fois basse consommation et faible bruit. Cela nous a conduit à choisir une architecture assez peu étudiée et pourtant très prometteuse, surtout grâce aux avancées de la technologie FDSOI.

5. Conclusion

Le seul choix imposé au début de ces travaux était la recherche des applications analogiques ou RF qui pourraient être améliorées grâce à la technologie FDSOI. Le choix de se concentrer sur une synthèse de fréquences a été imposé par l'équipe du laboratoire, afin de connaître les potentialités de cette nouvelle technologie vis-à-vis du bruit et de la consommation. Le choix du protocole Bluetooth Low Energy qui allait servir de base de réflexion pour définir les spécifications et concevoir la synthèse a été une évidence au vue de la technologie utilisée, extrêmement faible consommatrice d'énergie et extrêmement rapide, et de ses applications dans le domaine de l'internet des objets.

L'architecture retenue pour réaliser cette synthèse est le fruit de nombreuses recherches sur l'état de l'art. Cette architecture comporte certains défauts (sensibilité à la variabilité, bruit de phase et génération de spurious) dont les effets devraient être mitigés par les qualités de la technologie FDSOI (rapidité, faible variabilité). Enfin, cette technologie permet de procéder à une calibration du Vth de chaque transistor (par la polarisation de la grille arrière), et donc de la fonctionnalité sans

ajouter de transistors « non-fonctionnels » pouvant nuire à la consommation ou la fréquence maximale atteignable.

Dans le prochain chapitre, deux façons d'implémenter l'architecture de PLL fractionnaire à phases (par commutation ou par resynchronisation) seront étudiées.

Chapitre II – Modélisation système

1. Introduction

Dans le chapitre précédent, nous avons présenté plusieurs architectures de PLL et avons retenu une topologie à commutation de phases pour évaluer les performances induites par la technologie FDSOI. Dans ce chapitre, nous étudierons par des simulations système le fonctionnement de l'architecture à FPD, le comportement en bruit de phase et la génération des raies spectrales dus à la variabilité de la technologie CMOS utilisée. Puis nous présenterons un tableau rappelant toutes les spécifications extraites de ces simulations que devront respecter les blocs de la PLL que nous nous proposons de concevoir. Pour conclure, nous proposerons deux méthodes de calibration des phases du VCO. Les résultats de simulation nous permettront de retenir la meilleure des deux. Tous les modèles développés dans ce chapitre sont reportés en annexe.

Nous rappelons en Figure 40 le schéma-bloc de la PLL dont chaque bloc sera étudié séparément dans ce chapitre, avant de simuler l'ensemble.

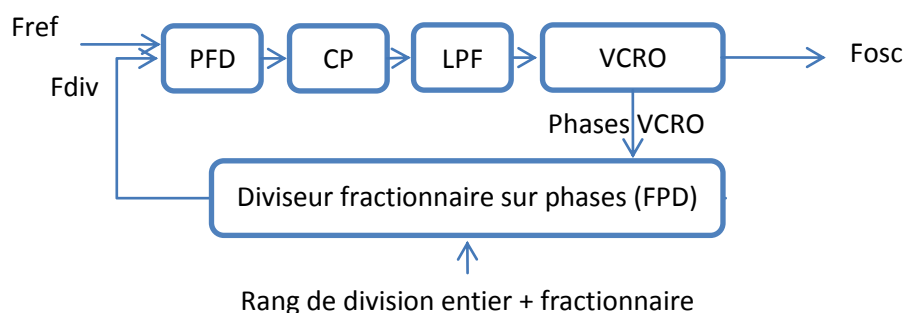


Figure 40 - Blocs fonctionnels de la PLL fractionnaire

1.1. Méthodologie de modélisation

La méthodologie de modélisation que nous avons utilisée est dite « meet-in-the middle ». En effet, nous avons développé des modèles de haut niveau (niveau système) pour simuler le comportement de PLL fractionnaires sur phase, afin de retenir la meilleure architecture pour notre application. Ces modèles ont été écrits en langage comportementaux Matlab et VerilogAMS. Toutefois afin d'estimer au mieux les performances de chaque topologie, de chaque bloc ainsi décrit, nous avons raffiné ces modèles en affectant à certains de leurs paramètres génériques des valeurs issues de simulation Spice (en prenant en compte les caractéristiques des transistors UTBB FDSOI 28nm). Dans certains cas, nous avons effectué des simulations mixtes VerilogAMS (au niveau

ystème) et Spice (au niveau transistor). Cette étape de modélisation nous a permis de définir les blocs et sous blocs de la PLL que nous allons réaliser, leurs fonctions et leurs spécifications, qui serviront de cahier des charges lors de la conception des circuits dans les chapitres 3 et 4. Les signaux échangés entre les blocs VerilogAMS sont compatibles avec les blocs au niveau transistor (signaux entre 0 et 1 Volt et fronts non idéaux) pour permettre une implémentation progressive des blocs et des simulations mixtes (SPICE + VerilogAMS).

2. Etude du bloc VCRO - Diviseur

2.1.1. Diviseur par commutation et diviseur par resynchronisation

Pour étudier le fonctionnement d'une PLL à division par commutation de phases, une simulation temporelle des blocs fonctionnels composant le diviseur a été mise en place. Les deux implémentations décrites au chapitre 1 (Figure 41 & Figure 42), seront modélisées afin de choisir laquelle est plus avantageuse en termes de bruit de phase, de consommation, de flexibilité et de simplicité d'implémentation.

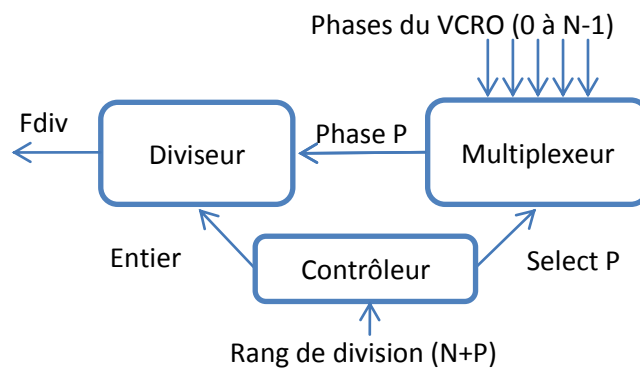


Figure 41 - Blocs fonctionnels du FPD à commutation de phases

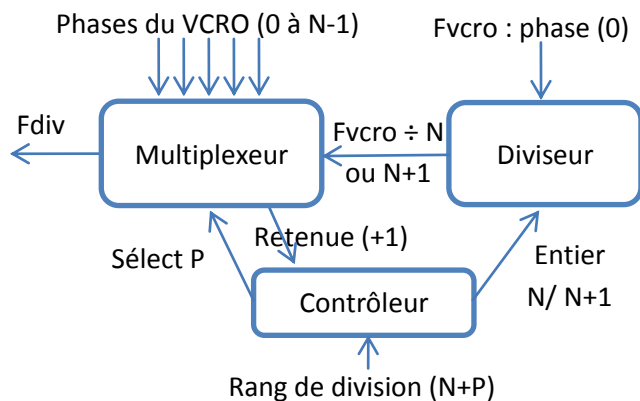


Figure 42 - Blocs fonctionnels du FPD à resynchronisation

La Figure 43 illustre par un chronogramme le fonctionnement du diviseur par commutation de phases. Dans cet exemple, nous avons choisi un VCO à 6 phases et une division par 3,5. Ainsi la première sortie du VCO (F_{osc} = phase 0) et la 4^{ème} sortie (phase 3) servent à calculer la partie fractionnaire de $\frac{1}{2}$. La partie entière de la division est réalisée par le diviseur. On peut constater que dans cette version, le multiplexeur est un bloc énergivore car d'une part il travaille à fréquence maximale et d'autre part il doit rester actif en permanence. De plus, le multiplexeur doit effectuer les changements de phase sans générer de glitch sous peine d'erreur de division.

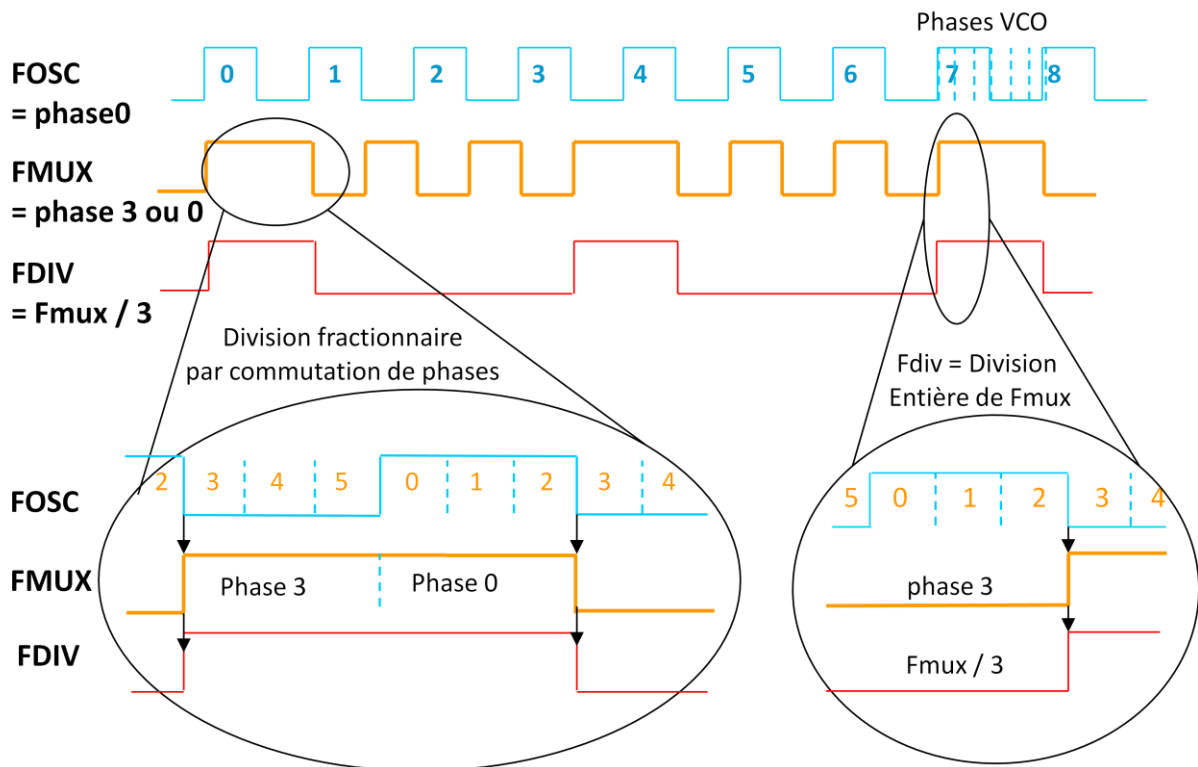


Figure 43 - Chronogramme du diviseur à commutation de phases

Dans la seconde version du diviseur, dite à resynchronisation, le diviseur ne dispose d'aucun de ces inconvénients. Son multiplexeur n'est activé que 2 phases F_{vcro} par cycle à la fréquence F_{div} (plus faible), ce qui consomme peu et ne crée pas de glitch. La contrepartie de cette économie d'énergie se fera au détriment d'une augmentation de la complexité d'implémentation. En effet le diviseur entier doit obligatoirement supporter le mode $N/N+1$: le multiplexeur signale à chaque débordement du compteur de saut de phases que la période F_{div} suivante doit être plus longue d'une période F_{vcro} . Ce comportement est illustré par la Figure 44 dans les mêmes conditions, à savoir 6 phases et une division par 3,5.

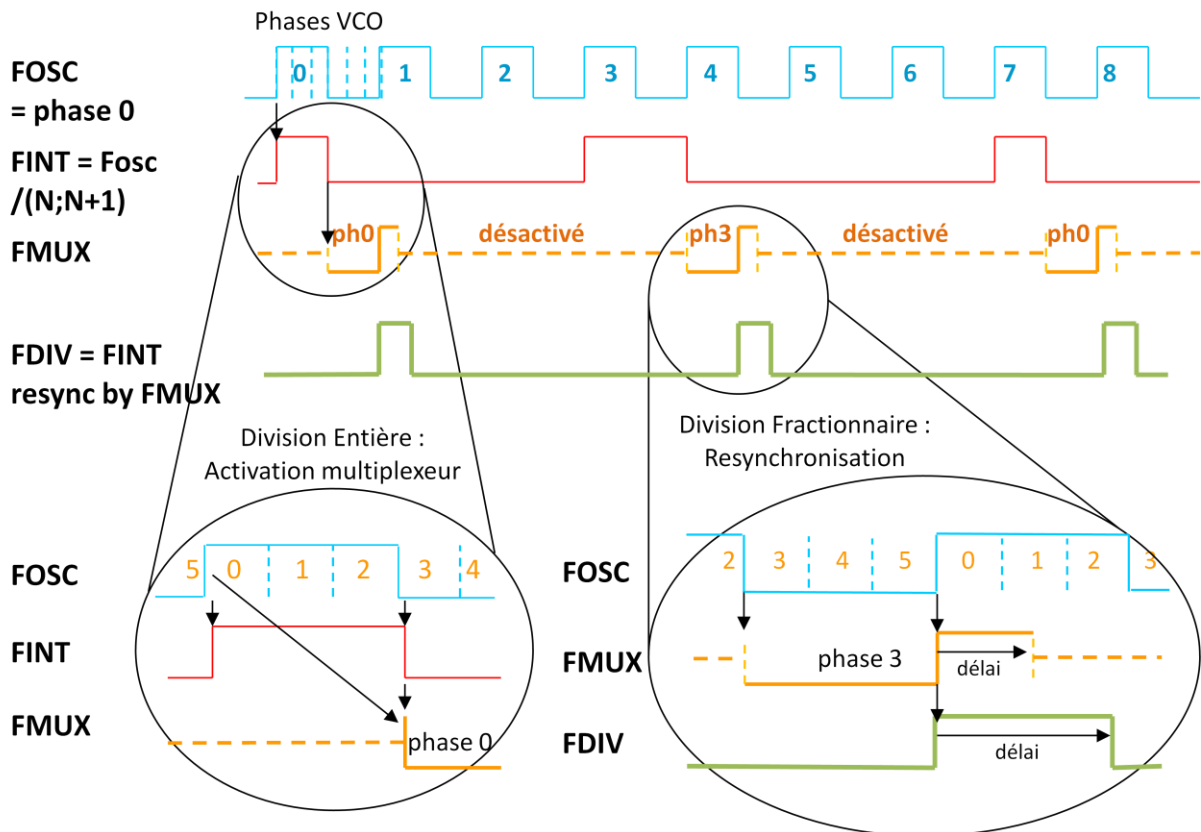


Figure 44 - Chronogramme du diviseur à synchronisation

Sur les deux chronogrammes, on vérifie que la fréquence de sortie Fdiv correspond à la fréquence de sortie du VCO Fosc divisée par 3,5. Le rapport cyclique entre les deux topologies est différent, mais ne correspond pas à une contrainte du cahier des charges de la PLL. Nous allons maintenant aborder la modélisation et la simulation au niveau système de ces blocs.

2.2. Modèles

2.2.1. VCRO à phases multiples

Le modèle du VCRO permet de modifier la fréquence de sortie via la tension d'entrée 2 fois par période. Il génère N phases, réparties sur la période du VCRO ($360^\circ/N$), pour simuler un oscillateur en anneau. Les paramètres suivants peuvent être modifiés :

- Le nombre de phases en sortie du VCO (N)
- La fréquence minimale, maximale et le gain (fmin, fmax, Kv)
- La tension correspondant à la fréquence nominale (V_f0)
- Les rails d'alimentation de la sortie (Vlo, Vhi)
- Le slewrate, fonction de la demi-période et d'une variable réglable (RiseForce)

La Figure 45 illustre ce modèle (voir annexe 1) avec les signaux d'entrée et de sortie, les paramètres génériques ainsi que les équations liant ces grandeurs, où "ain" est la tension de commande (d'entrée) du VCRO.

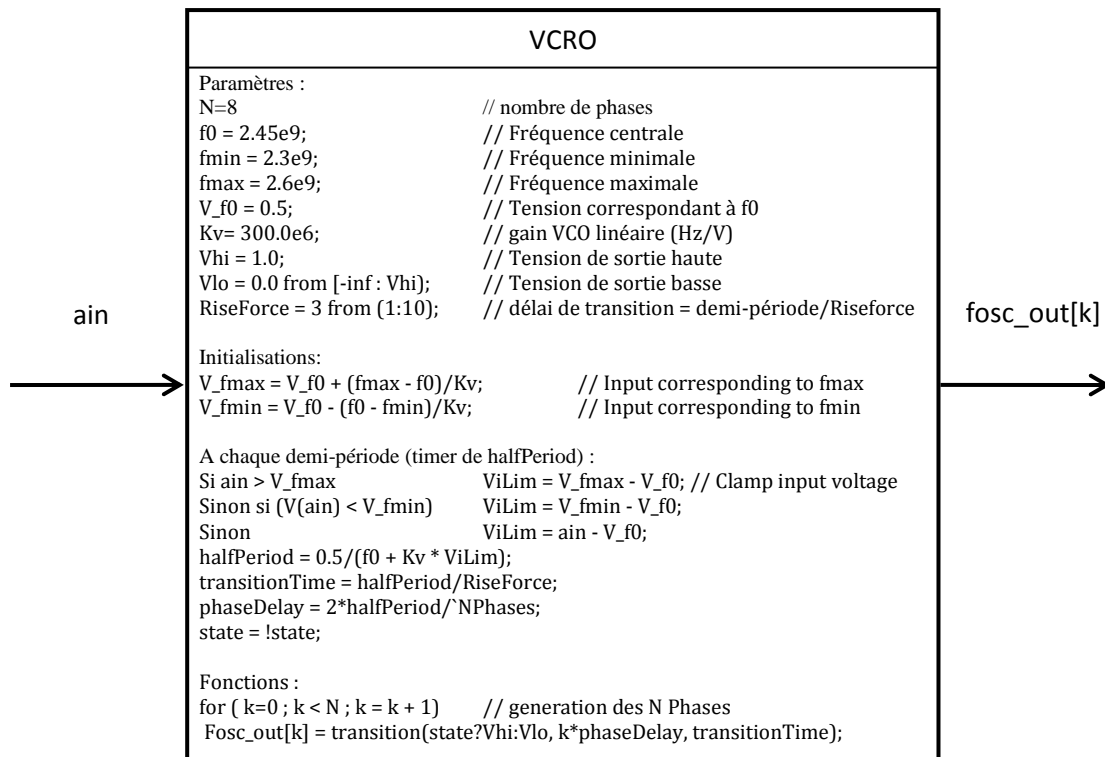


Figure 45 - Modèle de VCRO à Phases

2.2.2. Diviseur entier

Le modèle du diviseur (cf. annexe 2) possède une entrée analogique permettant de modifier la valeur de division entière.

Les paramètres suivants peuvent être modifiés au démarrage de la simulation :

- Le temps de propagation (Tp) et le temps de transition (Tt)
- La tension de seuil en entrée (Vth)
- Les rails d'alimentation de la sortie (Vlo, Vhi)

La Figure 46 illustre ce modèle avec les signaux d'entrée et de sortie, les paramètres génériques ainsi que les équations liant ces grandeurs.

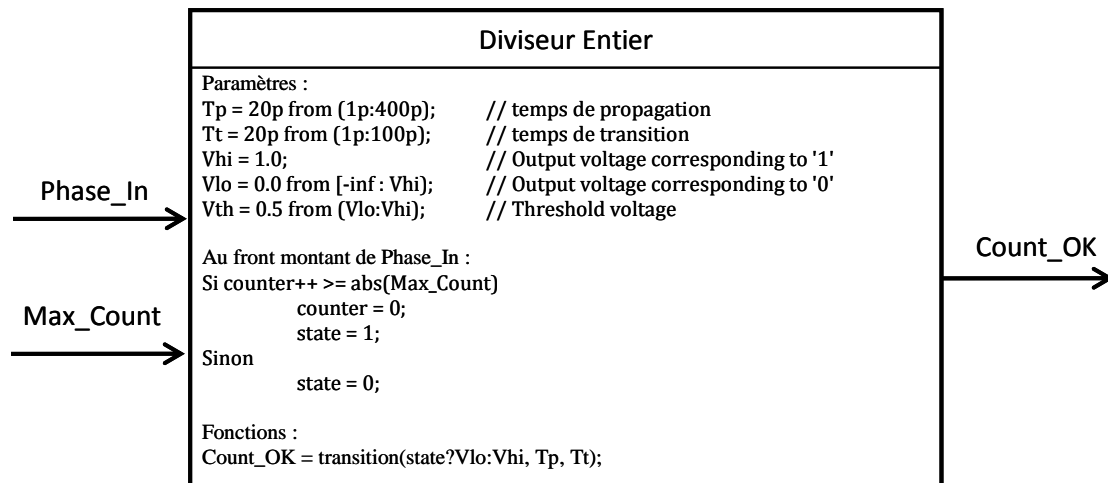


Figure 46 - Modèle de diviseur entier

2.2.3. Multiplexeur à commutation

Le modèle du multiplexeur à commutation est illustré par la Figure 47 (cf. annexe 3). Il est composé d'un additionneur à modulo et du multiplexeur. L'additionneur détermine la phase à transmettre au diviseur pour chaque division entière et réaliser ainsi la partie fractionnelle de la division (saut de phase). Le multiplexeur effectue le changement de phase sans créer de glitch ou manquer une période de division.

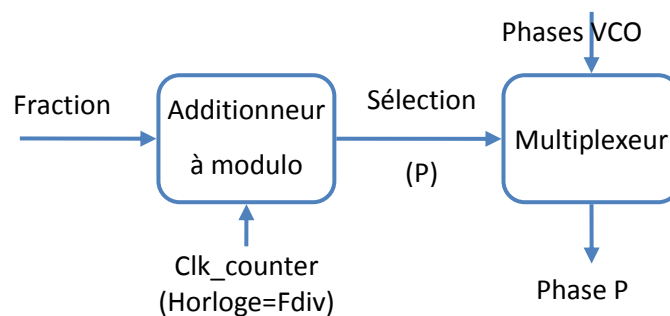


Figure 47 - Schéma fonctionnel du multiplexeur à commutation

Il agit de la façon suivante (voir Figure 48) : quand un front montant du signal de division entière (Clk_Counter) arrive, le calcul de la phase à commuter est activé (change_flag=1) et la sortie maintient sa valeur. Toutefois la phase sélectionnée n'est changée qu'à un des fronts de la nouvelle phase. Si la commutation s'effectue entre deux phases proches (saut < $N/2$; N =nombre de phases du VCO), alors le changement se produit au front descendant, sinon au front montant. Cela évite les erreurs de commutation jusqu'au prochain front (montant ou descendant) de la future phase.

Saut phase MUX > N/2 (ex : 0->4)

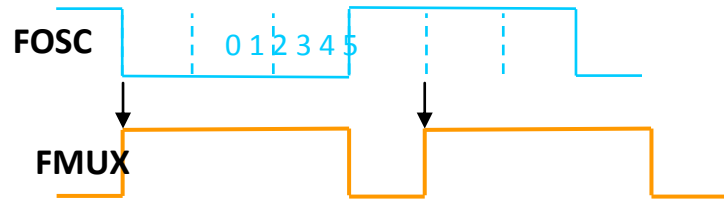


Figure 48 - Schéma temporel du glitch (impulsion indésirable) si le cas saut > N/2 n'est pas pris en compte

Les paramètres suivants peuvent être modifiés au démarrage de la simulation :

- Le temps de propagation (T_p) et le temps de transition (T_t)
- La tension de seuil en entrée (V_{th})
- Les rails d'alimentation de la sortie (V_{lo} , V_{hi})

La Figure 49 illustre ce modèle avec les signaux d'entrée et de sortie, les paramètres génériques ainsi que les équations liant ces grandeurs.

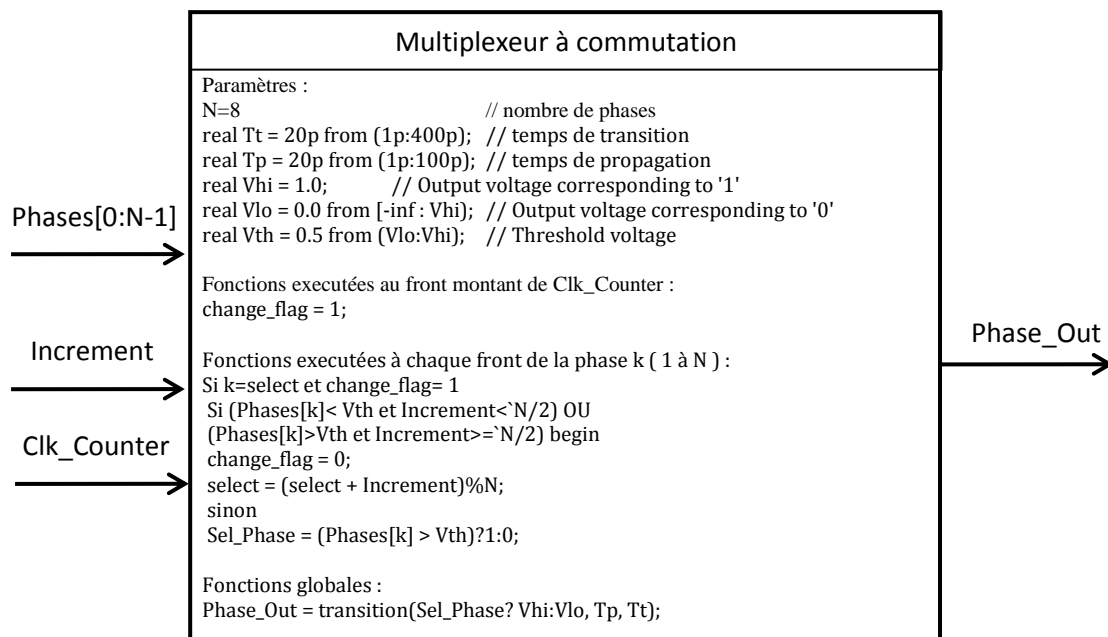


Figure 49 - Modèle de multiplexeur à commutation

2.2.4. Multiplexeur à resynchronisation

Le modèle de multiplexeur à resynchronisation est composé d'un calculateur et du multiplexeur (cf. Figure 50 et annexe 4). Le calculateur est composé d'un additionneur à modulo comme le multiplexeur à commutation mais avec signal de dépassement (overflow), afin de signaler

au diviseur $N/N+1$ pour qu'il passe en mode $N+1$. Le multiplexeur détermine la phase qui va resynchroniser la division entière avant d'être ajoutée à la partie fractionnaire.

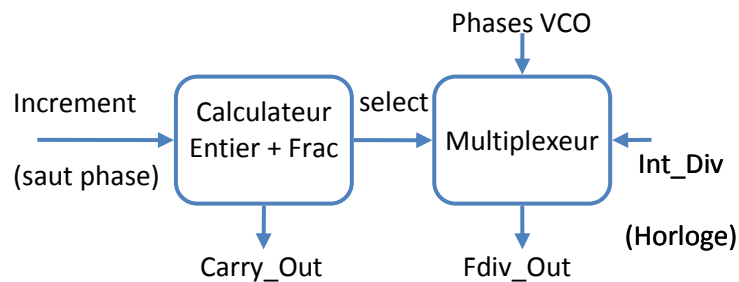


Figure 50 - Schéma fonctionnel du multiplexeur à commutation

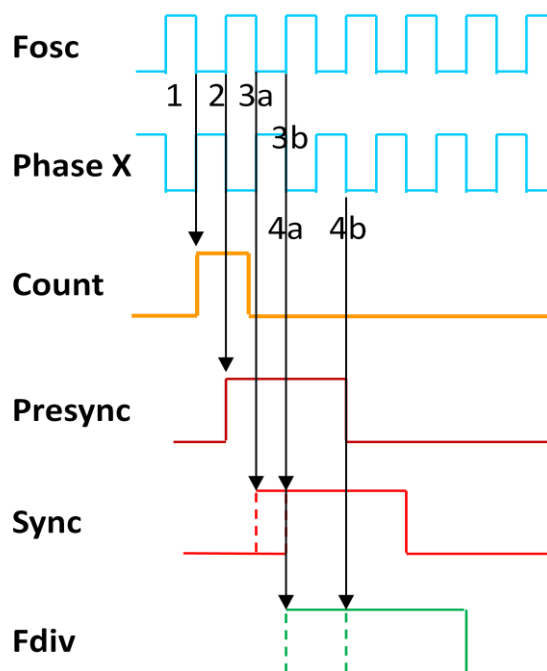


Figure 51 - Resynchronisation en 4 temps

Une machine d'état, représentée par les variables d'état Presync et Sync ($x,x = P,S$), gère la synchronisation en 4 temps ($x,x = 0,0; 0,1; 1,1; 1,0$), correspondant aux phases 1 à 4 de la Figure 51 :

Phase 1 - Après N fronts descendants de F_{osc} , le diviseur entier génère le signal 'Count'.

Phase 2 - Au front montant suivant de F_{osc} , le signal 'Count' est propagé dans la variable d'état 'Presync'. Cette première étape permet de s'affranchir du temps de propagation du diviseur.

Phase 3 - La valeur de Presync est propagé dans Sync au front montant de F_{osc} si le saut de phase est inférieur à la moitié des phases existantes ($\text{saut} < N/2$) et sinon au front descendant de F_{osc} ($\text{saut} \geq N/2$). Ceci permet de ne pas faire d'erreur de synchronisation quand le saut de phase est grand (front montant de la phase finale arrivant après le front descendant de la phase initiale).

Phase 4 - Quand Sync et Presync sont à l'état haut, le prochain front descendant de Fosc génère le signal de division Fdiv. La génération du signal Fdiv remet à zéro les variables d'état du diviseur pour le préparer à la prochaine division (P,S = 0,0).

Les paramètres suivants peuvent être modifiés au démarrage de la simulation :

- Le délai entrée-sortie (Td) et le temps de transition (Tt)
- La tension de seuil en entrée (Vth)
- La tension de sortie correspondant à un '1' (Vhi) et à un '0' (Vlo)

La Figure 52 illustre ce modèle avec les signaux d'entrée et de sortie, les paramètres génériques ainsi que les équations liant ces grandeurs.

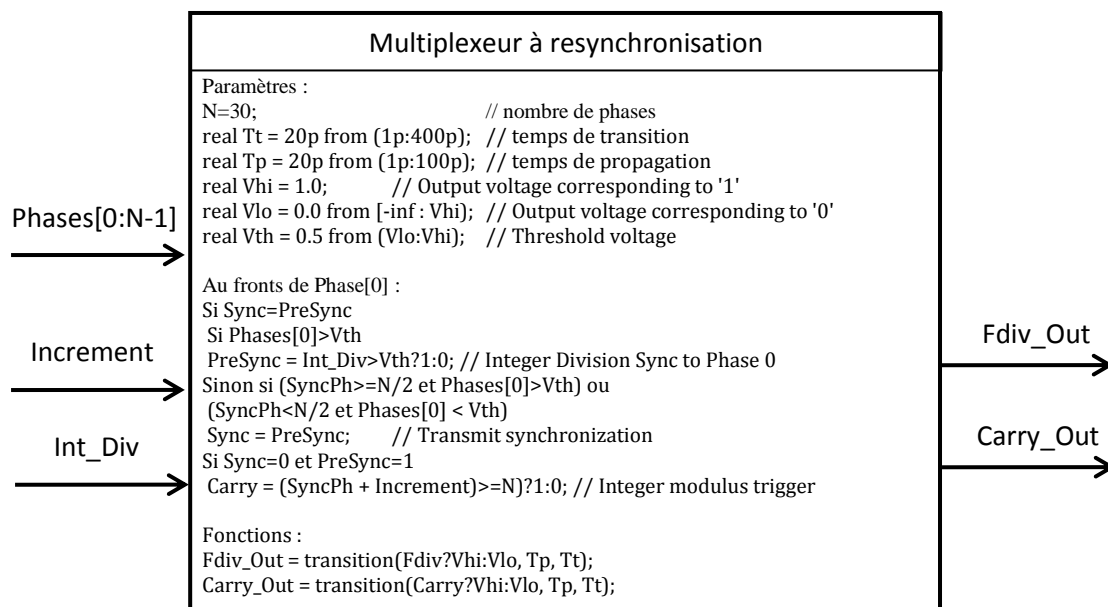


Figure 52 - Modèle de multiplexeur à resynchronisation

2.2.5. Contrôleur de division fractionnaire

Ce contrôleur n'est utilisé que pour le fonctionnement dual-modulus du diviseur entier dans la version à resynchronisation. Il permet d'ajouter 1 à la valeur de division à la demande du multiplexeur. Il a été modélisé (voir annexe 5) en VerilogAMS comme les autres blocs de cette architecture.

Les paramètres suivants peuvent être modifiés au démarrage de la simulation :

- Le délai entrée-sortie (Td) et le temps de transition (Tt)
- La tension de seuil en entrée (5Vth)
- La tension de sortie correspondant à un '1' (Vth) et à un '0' (Vlo)

- Le nombre de phases du VCO (Phases)

La Figure 53 illustre ce modèle avec les signaux d'entrée et de sortie, les paramètres génériques ainsi que les équations liant ces grandeurs.

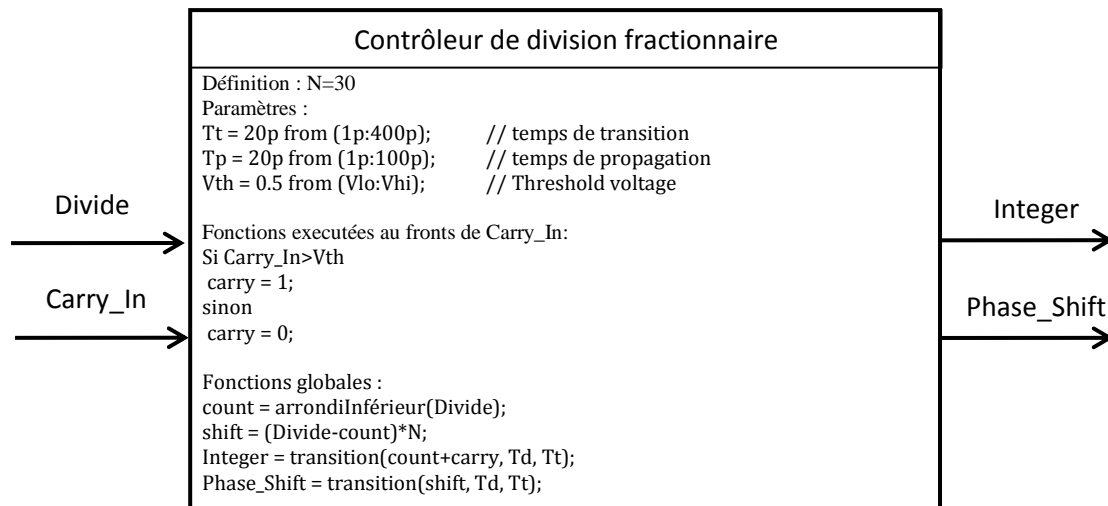


Figure 53 - Modèle de multiplexeur à resynchronisation

2.3. Résultats

2.3.1. Simulation temporelle des phases du VCO

Dans un premier temps nous avons simulé un VCRO à 8 phases. Ce circuit servira de générateur de phases pour valider les diviseur FPD énoncés précédemment : diviseur à commutation et diviseur à resynchronisation. Dans cet exemple, illustré par la Figure 54, la fréquence d'oscillation est de 2,45GHz et les phases sont équiréparties tous les 45°.

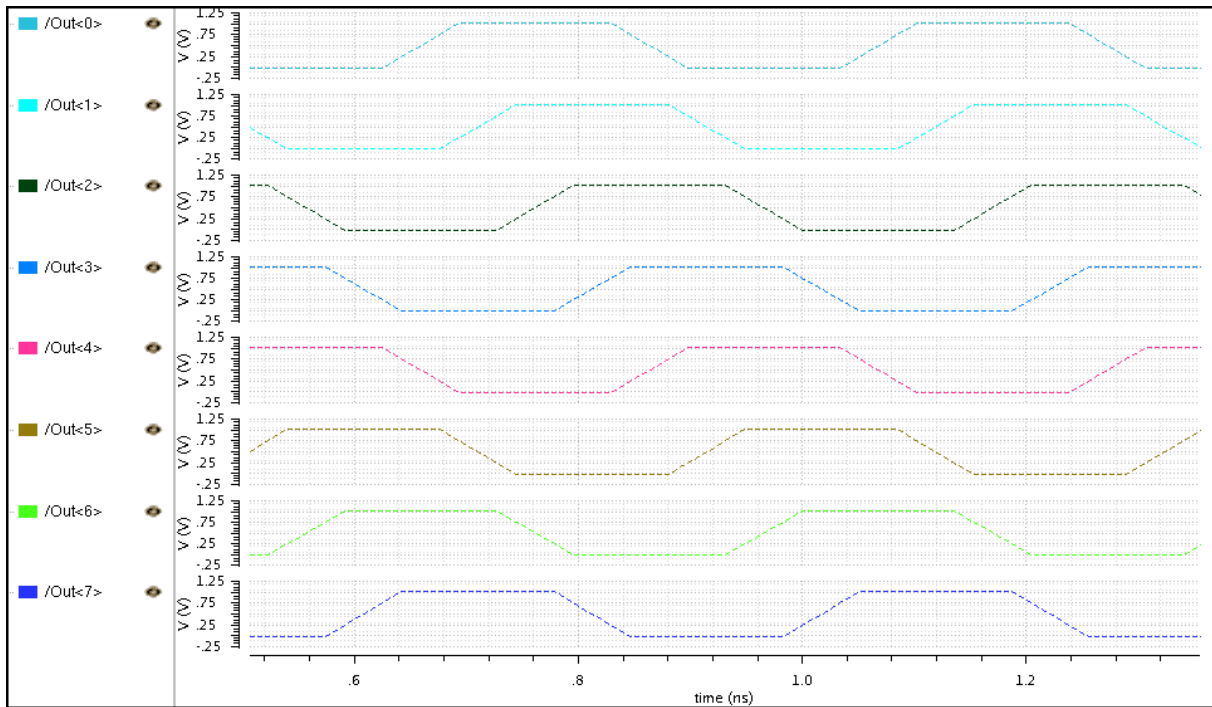


Figure 54 - Simulation temporelle du modèle de VCO à 8 phases

Pour simuler le diviseur à resynchronisation, le VCRO devra générer 30 phases, alors que dans le cas du diviseur à commutation nous avons conservé un VCRO à 8 phases.

2.3.2. Simulation du FPD à commutation de phases

Dans le cas présenté en Figure 55, l'oscillateur génère 8 phases (phase 0 en brun - Out<0>) et le multiplexeur fournit une de ces phases (en rouge) au diviseur entier. La sortie de ce diviseur est retransmise au multiplexeur pour lui indiquer un changement de phase. La partie fractionnaire de la division est réalisée par le saut de sept phases (valeur en orange, phase sélectionnée en bleu), la division entière est de 4 (valeur constante en vert), soit une division fractionnaire de 4 et 7/8, soit 4,875.

On peut voir que la période du signal de sortie du multiplexeur (en rouge) est plus longue de 7/8 de période toutes les 4 périodes du VCRO. Mais par contre en sortie du diviseur entier on peut constater que la période est parfaitement constante, malgré une division fractionnaire.

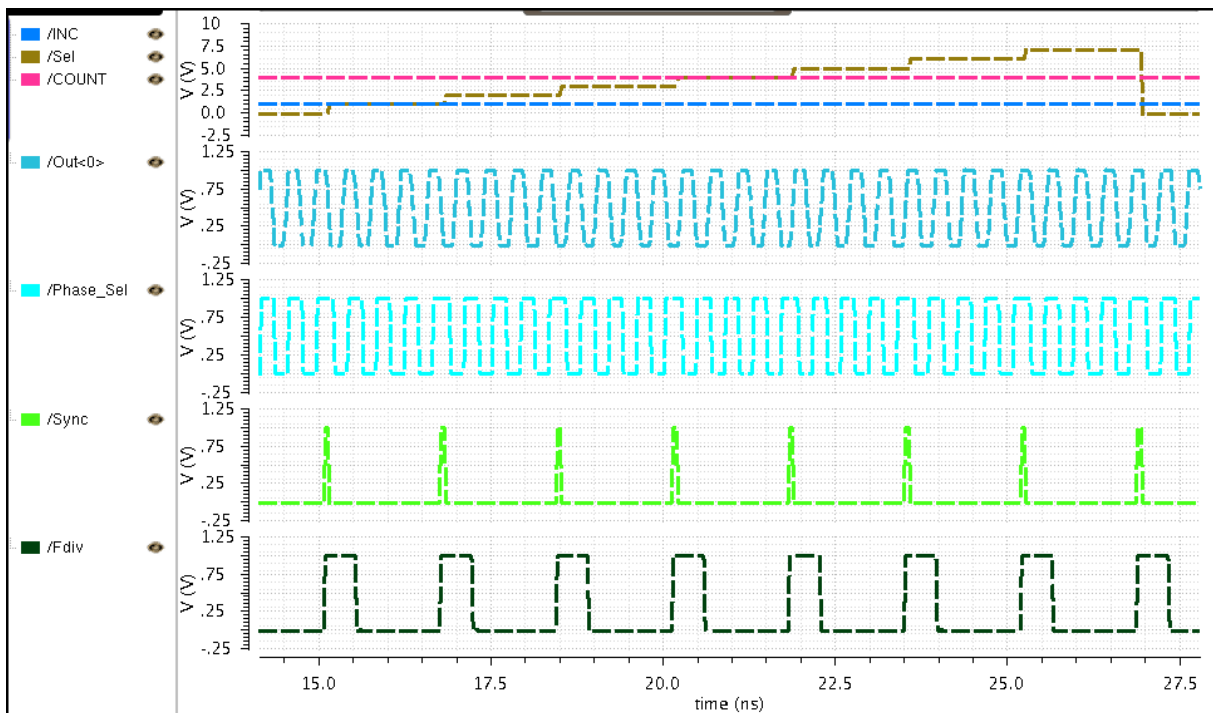


Figure 55 - Signaux du FPD à commutation (Div = 4 + 7/8)

Si on change la partie fractionnaire de la division à 1/8, le fonctionnement reste identique, les sauts de phase sont plus courts (Figure 56).

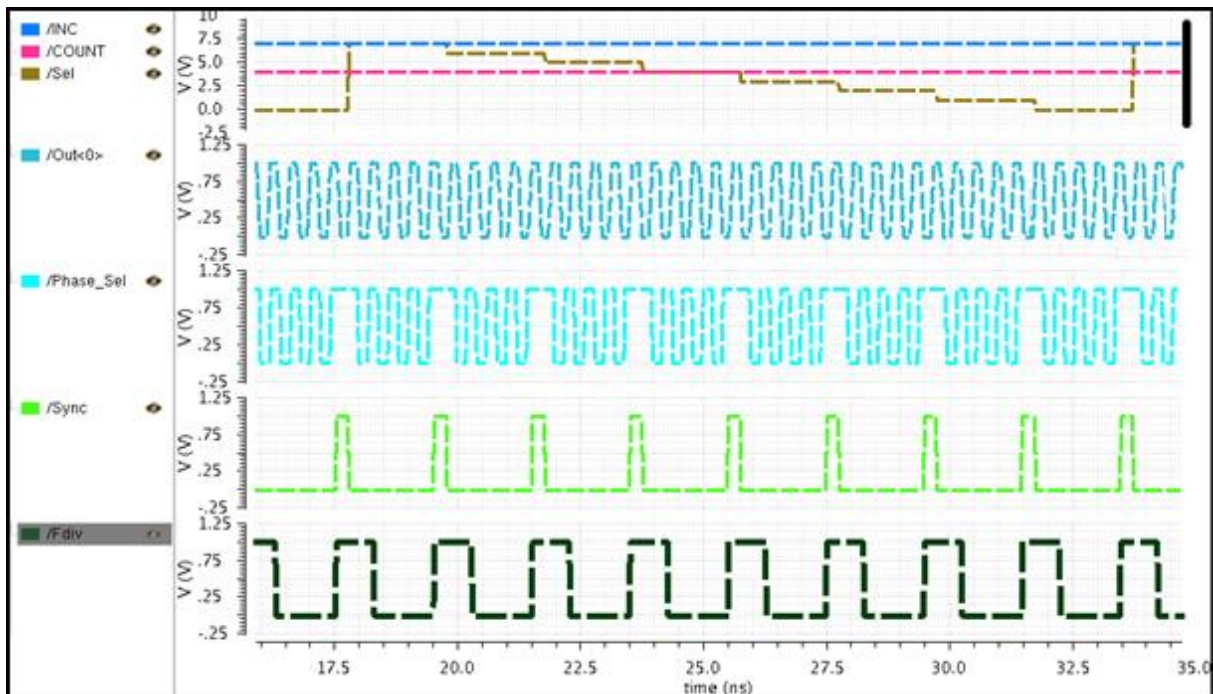


Figure 56 - Signaux du FPD à commutation (Div = 4 + 1/8)

2.3.3. Simulation du FPD à resynchronisation par phase

Dans le cas présenté en Figure 57, l'oscillateur génère 30 phases et le diviseur entier est câblé sur l'une d'entre elles (la première en rose, nommée Fosc). Le multiplexeur est déclenché par la division entière (en bleu) et la resynchronise une première fois (en blanc) au front montant de la première phase et une deuxième fois (en marron) au dernier front avant le front montant de la phase de division fractionnaire (en violet). L'écart entre 2 phases sélectionnées (valeur en vert et de la sélection en jaune) est de 1 à N-1 phases (pour N phases au total). Le diviseur entier (en orange clair) varie au cours du temps avec un rapport cyclique permettant d'obtenir la valeur de division fractionnaire (orange foncé). La division fractionnaire est réalisée par le saut de 5 phases, la division entière est de 4, soit une division fractionnaire de $4 + \frac{5}{30}$, soit 4,167. La fréquence instantanée du diviseur entier varie d'un cycle à l'autre en mode fractionnaire ($N/N+1$). Le multiplexeur resynchronise la sortie du diviseur entier à une des phases du RO pour que la sortie Fdiv soit une vraie division fractionnaire.

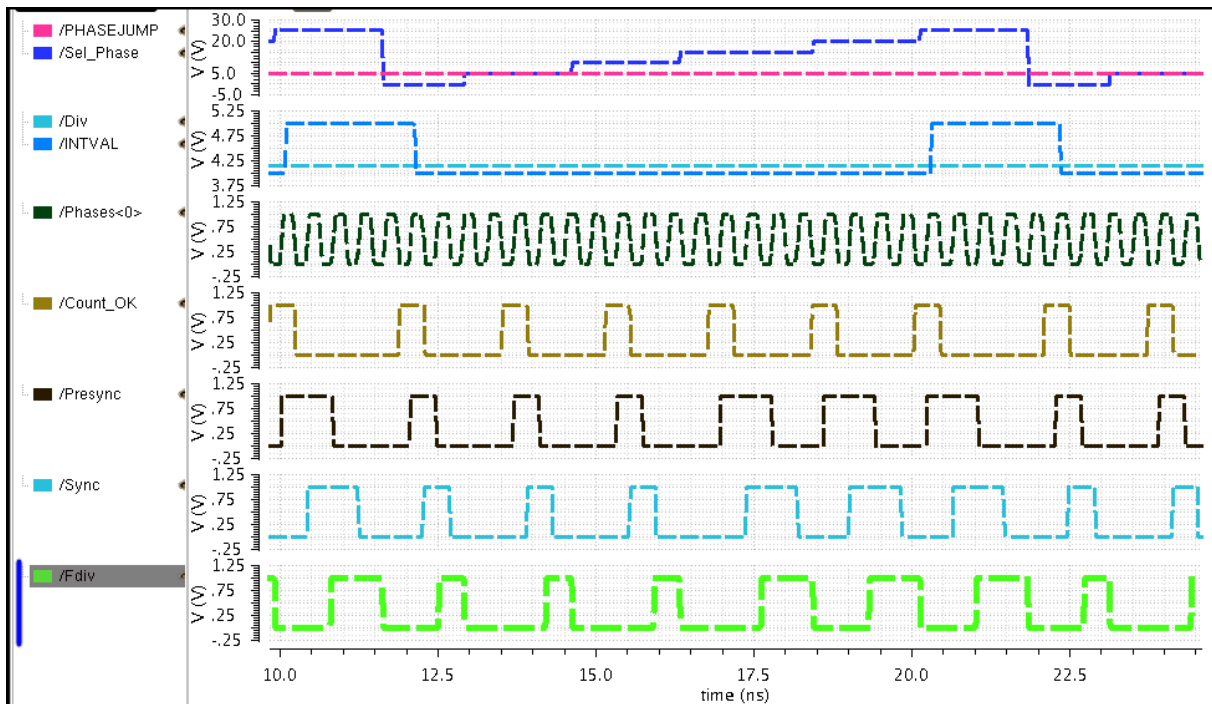


Figure 57 - FDP à resynchronisation ($K_{div} = 4 + \frac{5}{30}$)

Sur la Figure 58, on peut constater que les phases sont sélectionnées en ordre inverse (en jaune) car la partie fractionnaire est de $\frac{25}{30}$. La largeur des impulsions est différente de période en période mais seul le front montant contient l'information de phase, le comparateur de phases n'étant sensible qu'au front montant.

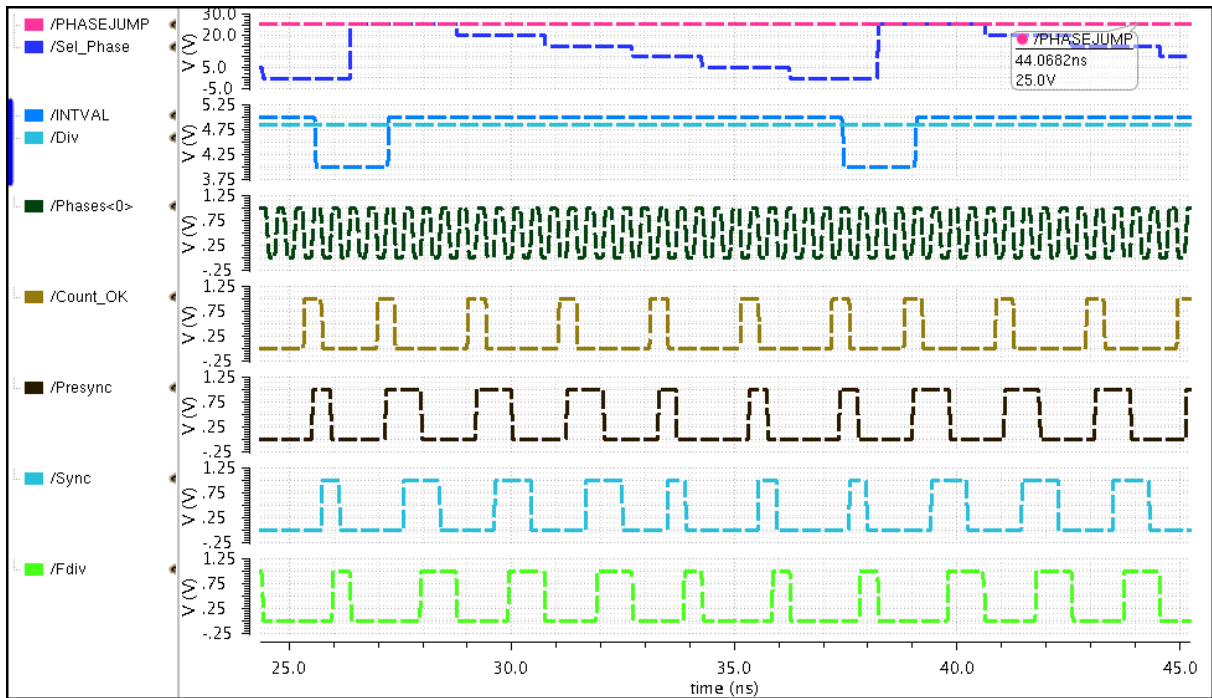


Figure 58 - FPD à resynchronisation (Kdiv = 4+ 25/30)

2.4. Conclusion

Les simulations ont validé les architectures proposées pour réaliser la division fractionnaire qui sera intégrée dans la PLL finale. De plus, le soin apporté à l'écriture des modèles nous permettra de les transposer facilement en schéma électrique afin d'implémenter chaque bloc au niveau transistor. Lors de ces étapes de conception, nous pourrons pour chaque bloc remplacer le code VerilogAMS par une description SPICE et réaliser des simulations mixtes afin de valider chaque bloc séparément dans son environnement final (les autres blocs restant décrits en VerilogAMS).

En particulier, nous avons ainsi validé notre nouvelle topologie de division par resynchronisation qui devrait être plus économe en énergie. Dans la suite de ce mémoire notre effort portera principalement sur cette structure.

Dans la partie suivante seront modélisés les autres blocs nécessaires pour réaliser une PLL et la simuler en boucle fermée. Le chapitre suivant est consacré uniquement à la conception et réalisation du VCRO, bloc le plus important de notre architecture.

3. Etude d'une PLL à base de diviseur FPD

3.1. Introduction

Pour étudier le temps de réponse et le fonctionnement transitoire d'une PLL complète (Figure 59), visant une application de type BLE, des modèles en verilogAMS d'un comparateur de phases (PFD) et d'une pompe de charge (CP) ont été développés, ainsi qu'un filtre de boucle en SPICE. Les blocs VCO, et diviseur FPD à resynchronisation présentés à la section précédente ont été réutilisés.

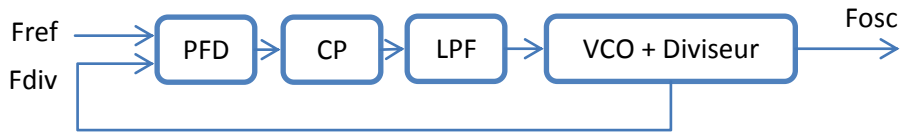


Figure 59 - Blocs fonctionnels de la PLL fractionnaire

3.2. Modèles

3.2.1. Détecteur de Fréquence et Phase (PFD)

Les modèles de PFD que nous avons développés sont de type trois états (Up, Down et OFF). Ils disposent de deux entrées sensibles aux fronts (montants ou descendants), d'une entrée de reset et de deux sorties logiques. Chacune des sorties répond positivement à un front de l'entrée correspondante (Fref ou Fdiv), et quand la seconde entrée reçoit le même type de front les deux sorties passent à zéro. Quand les deux entrées sont en phase, les deux sorties sont activées pendant une très brève durée, évitant ainsi la zone de gain nul (dead zone). Le modèle est réglable sur les paramètres les plus basiques tels que :

- Le délai de propagation (t_p), de transition (t_{TR}) et de reset (td_rst)
- Le déclenchement sur front montant ou descendant (EDGE)
- La valeur initiale de la sortie (state_init)
- La tension de seuil en entrée (V_{th})
- La tension de sortie du VCO correspondant à un '1' (V_{hi}) et à un '0' (V_{lo})

La Figure 60 illustre ce modèle avec les signaux d'entrée et de sortie, les paramètres génériques ainsi que les équations liant ces grandeurs. On retrouvera en annexe 6 son code VerilogAMS.

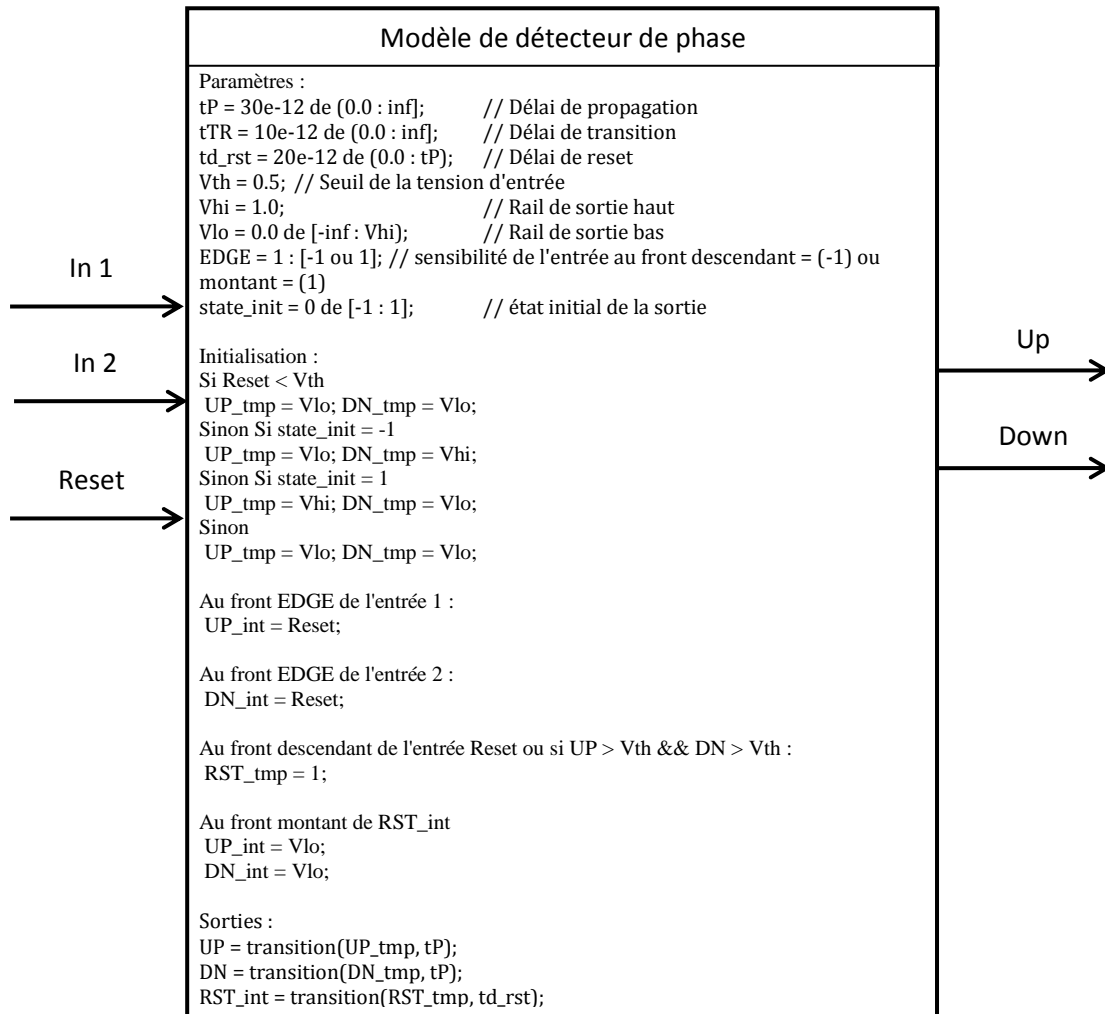


Figure 60 - Modèle de détecteur de phase

3.2.2. Pompe de Charge (CP)

Notre modèle de CP (cf. annexe 7) possède deux entrées en tension et une sortie en courant. Suivant que l'une ou l'autre des entrées est à l'état haut, la source de courant du haut et/ou du bas est activée. Le modèle est réglable sur les paramètres suivants :

- La tension de seuil en entrée (Vth)
- Les paramètres des sources de courant haute et basse (courant ON (Ion), résistance saturée (r_{DS}), résistance linéaire (r_{ON}), et résistance OFF (r_{OFF}))

La Figure 61 illustre ce modèle avec les signaux d'entrée et de sortie, les paramètres génériques ainsi que les équations liant ces grandeurs.

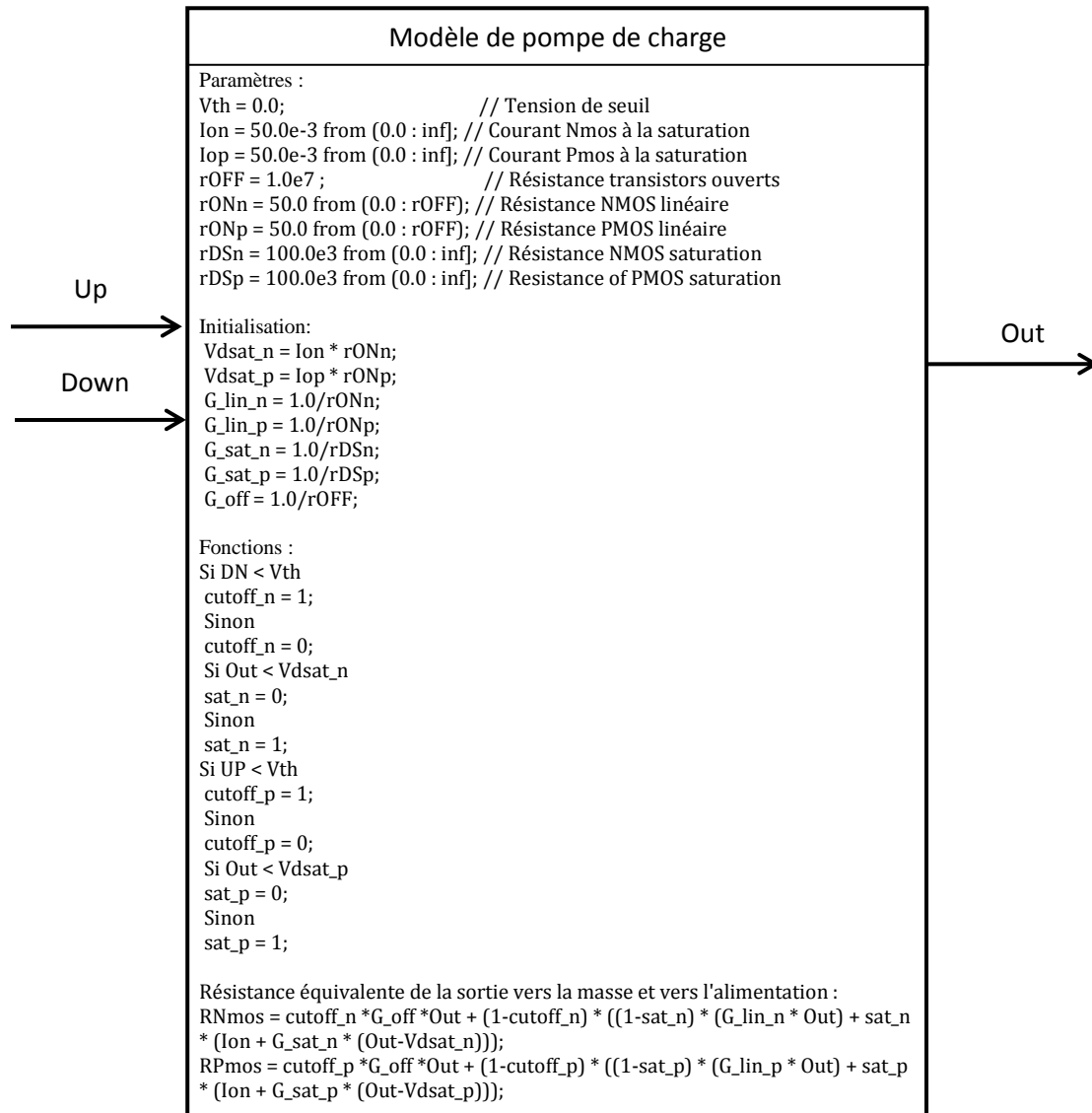


Figure 61 - Modèle de détecteur de phase

3.2.3. Filtre Passe-Bas (LPF)

Le modèle du filtre passe bas est écrit en SPICE (cf. Figure 62), la bande passante de la PLL est conçue pour être à 3MHz et la marge de phase à 45°. Les valeurs sont calculées à l'aide d'un fichier Excel d'après les formules du Dr. Allen [63] : R1=128kΩ ; C1=1,56pF ; C2=120fF.

```

.SUBCKT DSG_FILTER VCP VTUNE
R1 NET01 0 128K
C1 VTUNE NET01 1.56p
C2 VTUNE 0 120f
.ENDS

```

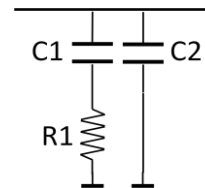


Figure 62 - Modèle Spice et schéma du filtre passe bas

3.3. Simulation temporelle de la PLL fractionnaire

Pour simuler le changement de canal, la PLL modélisée est commandée par une référence de 30MHz et un rang de division qui varie pour obtenir une fréquence de sortie de 2450MHz puis 2452MHz. Le changement de canal est effectif en moins d'une demi-microseconde (475ns) à 1% près (20kHz), comme illustré sur la Figure 63. Entre 2 canaux il n'y a pas de glissement de période et le système est stable (dépassement de 5%/100kHz). Le rapport de fréquence entre bande passante de la PLL et fréquence de comparaison (référence si non divisée) est traditionnellement fixé à 10 (par sécurité pour la stabilité) et peut descendre, pour certaine PLL à l'état de l'art, à 7. En choisissant un rapport de 10, les corrections d'erreur (injection de courant par la pompe de charge) sont peu filtrées et provoquent des dépassements instantanés jusqu'à 100% du saut de fréquence.

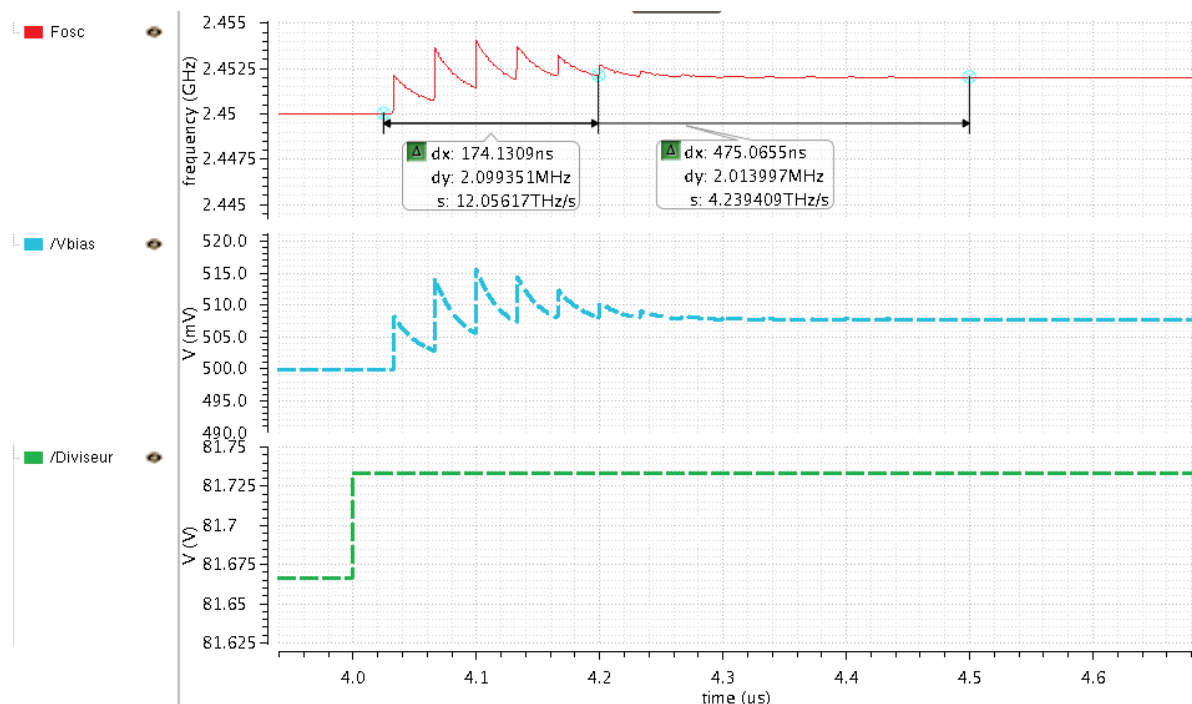


Figure 63 – Simulation de la PLL à FPD lors d'un changement de canal +2MHz

Cette simulation finale valide l'architecture globale de la PLL ainsi que l'utilisation du FPD à resynchronisation.

3.4. Conclusion

Le fonctionnement de la PLL est proche de celui d'une PLL fractionnaire où la fréquence en sortie du diviseur varie entre (F_{osc}/N) et $(F_{osc}/N+1)$. Toutefois, aucun artifice n'est utilisé pour créer une division fractionnaire moyenne ni pour mettre en forme ou filtrer le bruit (par exemple par un modulateur sigma-delta). Grâce à l'architecture FPD, la fréquence de comparaison et la bande passante peuvent être très élevées, améliorant la rapidité d'accrochage et de démarrage. Le pas de fréquence est égal au rapport de la fréquence de comparaison sur le nombre de phases du VCRO.

Ainsi, pour un VCRO à 30 phases et un pas de 2MHz (BLE), la fréquence de comparaison atteint 60MHz et la bande passante peut atteindre 6MHz. Cette architecture de PLL consomme peu et peut être améliorée, grâce à sa rapidité de démarrage, en étant désactivée pendant les silences radio du standard BLE.

4. Etude du bruit de phase

4.1. Introduction

L'étude de bruit de phase est une technique commune à toutes les PLL pour connaître quelles sont la ou les parties de la PLL qui sont les contributeurs majoritaires du bruit total en sortie. Les simulateurs tels que Eldo sont capables de simuler le bruit de phase d'une PLL complète en boucle fermée si c'est un diviseur entier ou un diviseur N/N+1 standard. Toutefois, en raison de la topologie de notre diviseur sur phase, il est impossible, à notre connaissance, de simuler le bruit de phase de la PLL en boucle fermée. Nous avons donc opté, dans un premier temps, pour la simulation de cette partie dans le domaine temporel.

Les bruits intrinsèques des blocs suivants (Figure 64) influent sur le bruit total de la PLL :

- Fref : Fréquence de référence : Bruit de phase typique d'un quartz
- PFD/CP : Détecteur de phase et pompe de charge : Bruit ramené à un courant
- LPF : Filtre passe bas : Bruit Thermique
- VCO : Oscillateur commandé en tension : Bruit de phase
- DIV : Diviseur fractionnaire : Jitter (+10 log(N))

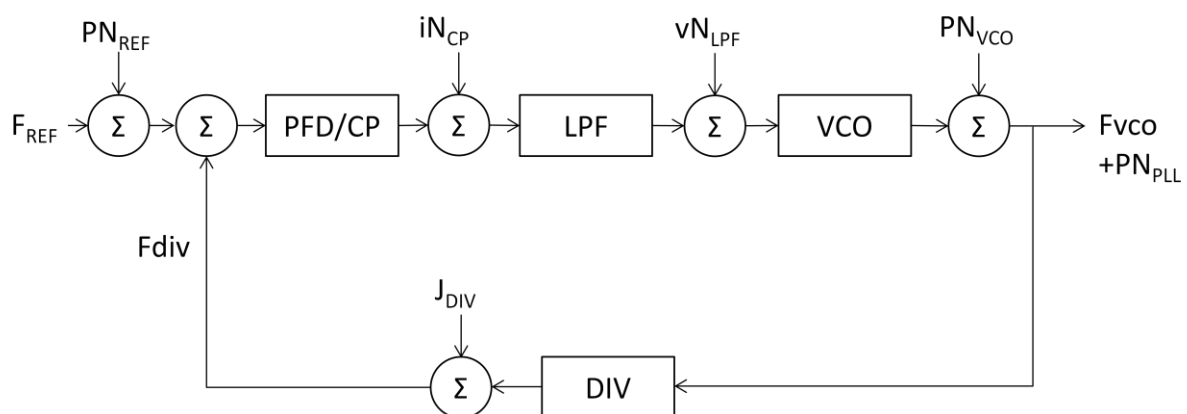


Figure 64 - Modèle linéaire du bruit de phase d'une PLL avec chaque contributeur

Pour réaliser des simulations prenant en compte des ordres de grandeurs réalistes, chaque bloc a été implémenté et simulé au niveau transistor pour en extraire les valeurs de bruit. Le diviseur étant difficile à simuler et à mesurer en termes de bruit de phase, sa contribution en bruit a été ignoré dans cette simulation. Cette approximation semble justifiée par le fait que ce circuit, réalisé par quelques portes logiques, ne génère que peu de bruit. Le bruit de phase total de la PLL a été calculé selon deux techniques différentes pour comparaison : sommation de la Densité Spectrale de Puissance (DSP) du bruit de chaque contributeur ou TF du jitter par la méthode de K. Kundert [64].

4.2. Extraction du bruit de chaque contributeur

Afin de réaliser une simulation du bruit de phase la plus réaliste possible, nous avons extrait, à partir de simulations SPICE, les caractéristiques de bruit de chaque contributeur.

La tension de bruit du filtre de boucle (vN_{LPF}) du 2ème ordre, ayant une bande passante de 3MHz et une marge de phase de 60°) est de 45nV/√Hz, et décroît de -40dB par décade après la fréquence de coupure. Les valeurs des composants sont identiques à la section précédente : $R1=120k\Omega$; $C1=1,67pF$ et $C2=128fF$.

Le courant de bruit du détecteur de phase (iN_{CP}), qui englobe le bruit du comparateur de phases dont le temps d'activation minimum est de 50ps et le bruit de la pompe de charge dont le courant est de 50μA, est de 6fA/√Hz pour l'intervalle de tension de sortie compris entre 0,1 et 0,9V.

Le modèle électrique de quartz (LCR//C) de la Figure 65 a été utilisé pour simuler le bruit de phase dû à la référence. Il est basé sur un quartz typique de 30MHz avec un facteur de qualité $Q=50000$ et une résistance série de 20Ω. L'oscillateur est un inverseur FDSOI avec une résistance de contre réaction de 1MΩ et un deuxième inverseur pour remettre en forme le signal.

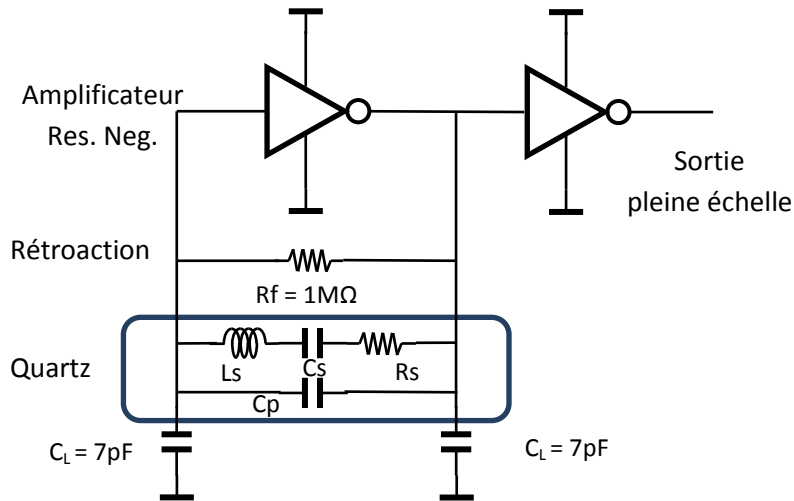


Figure 65 - Modèle d'oscillateur à Quartz utilisé

Dans la simulation finale de la PLL, nous avons injecté le bruit de phase de l'oscillateur à quartz sous forme de droites par morceaux (piecewise) en utilisant les valeurs extraites de la simulation à chaque décade, comme illustré sur la Figure 66.

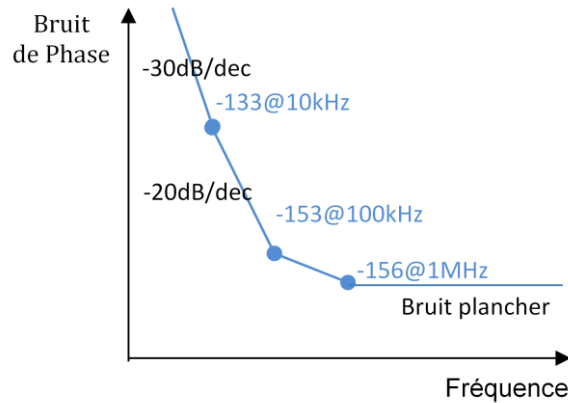


Figure 66 - Bruit de phase de l'oscillateur à quartz par segments

Le VCRO (single ended, 15 phases) présenté au chapitre III a été simulé en bruit de phase (voir Figure 67). Il présente un bruit de phase de -51, -60, -89, -115 et -137dBc/Hz à 10kHz, 100kHz, 1MHz, 10MHz et 100MHz d'offset de la porteuse respectivement.

Comme dans le cas de la référence (Quartz), le bruit du VCO est injecté par morceaux pour le calcul du bruit de phase de la PLL. La Figure 67 illustre ce modèle où les valeurs sont extraites d'une simulation Spice.

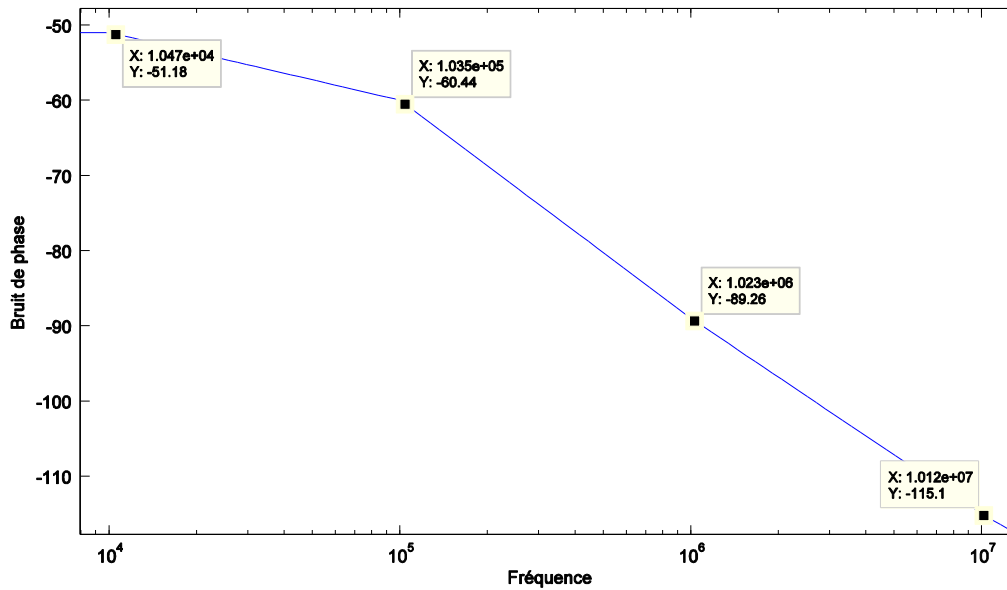


Figure 67 - Bruit de phase du VCO modélisé

4.3. Résultats

4.3.1. Simulation dans le domaine temporel

La première technique utilise la méthode développée par Ken Kundert [64]. Les modèles VerilogAMS reproduisant le bruit des blocs de la PLL sont simulés dans le domaine temporel. Le jitter est enregistré en sauvegardant la déviation du front montant de la division par rapport à l'idéal. Une fonction Matlab (cf. annexe 8) permet, par transformée de Fourier, de ramener ce Jitter en bruit de phase. Dans la pratique nous avons enregistré plus de 400 000 périodes pour effectuer ce traitement. La Figure 68 illustre ce résultat représentant le bruit de phase de la PLL.

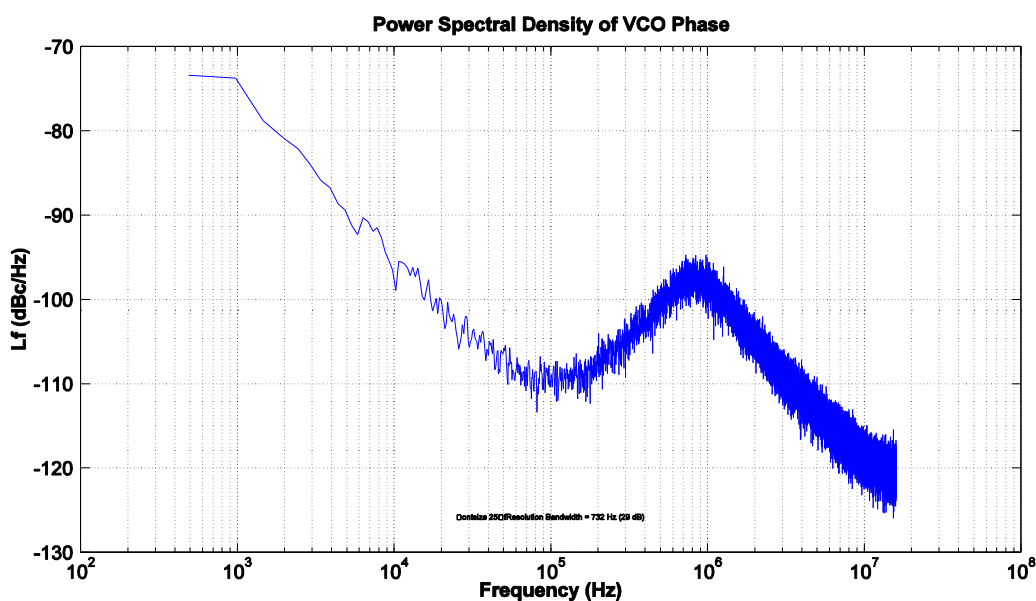


Figure 68 - Bruit de phase par la méthode de K. Kundert

Pour effectuer cette simulation, nous avons, suivant la méthode de Kundert, remplacé les caractéristiques de bruit de chaque contributeur par un jitter équivalent. Ce jitter est soit de type cumulatif dans les oscillateurs (VCO, Quartz), soit de type synchrone dans les blocs pilotés (FPD, PFD/CP).

La formule (11) permet de calculer l'écart type du jitter cumulatif, en reproduisant un bruit thermique qui diminue de -20dB par décade, en fonction du bruit de phase (Lf) à un certain offset (ΔF) de la fréquence d'oscillation (F_{osc}). Le jitter cumulatif de l'oscillateur de référence à quartz est de 19,2fs (133dBc@10kHz à $F_{osc}=30$ MHz) et de 360fs pour l'oscillateur en anneau commandé en tension (90dBc@1MHz à $F_{osc}=2450$ MHz).

$$J^2 = 2 \times L(\Delta F) \times \frac{\Delta F^2}{F_{osc}^3} \quad (11)$$

Le jitter synchrone est soit le résultat d'une simulation de bruit (diviseur, PFD) soit le calcul du jitter équivalent (12) d'un bruit de phase. Par exemple pour la pompe de charge, le jitter équivalent est ramené à l'entrée du comparateur de phases.

$$J_{EE} = \frac{T}{K_{PFD}} \sqrt{\frac{\text{var}(n)}{2}} ; \text{var}(n) = \int_0^{\infty} S_n(f) \cdot df \quad (12)$$

Le calcul de l'intégrale du Jitter équivalent du comparateur de phases a été effectué avec les outils Eldo/Mentor Graphics. Le résultat est un Jitter RMS équivalent de 10fs à l'entrée du PFD. Le bruit du diviseur fractionnaire a été ignoré car sa sortie est resynchronisée sur une des phases du VCRO et n'ajoute donc aucun bruit. Le bruit généré par la sélection cyclique de phases au sein du diviseur FPD avec mismatch génère un jitter déterministe qui sera étudié à la section suivante.

4.3.2. Simulation dans le domaine fréquentiel

La deuxième technique, plus rapide, consiste à utiliser un fichier Matlab (cf. annexe 9) pour calculer la contribution de chaque bloc sur la sortie. Dans un premier temps, la contribution en bruit de chaque bloc est calculée à la sortie de la PLL (mise en forme par le gain de boucle linéarisé). La somme de toutes les contributions donne le bruit total de la PLL, illustrée par la Figure 69.

L'avantage par rapport à la méthode précédente est la rapidité du calcul (pas de simulation transitoire), la précision des résultats (pas de valeur imprécise) et la possibilité de définir le bruit par morceaux (plus réaliste).

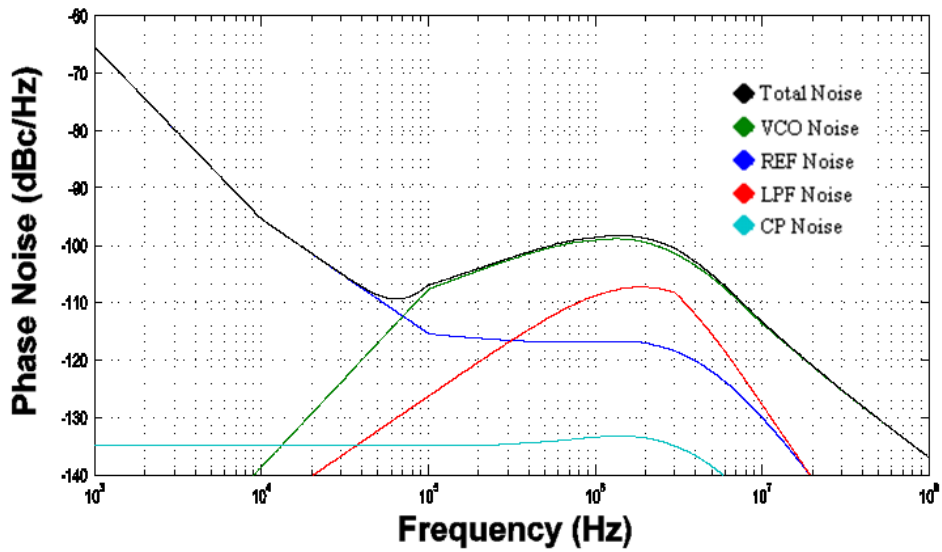


Figure 69 - Contribution des blocs et total du bruit de phase de la PLL par la méthode fréquentielle

4.4. Conclusion

En superposant les deux types de simulations, on peut constater qu'elles donnent des résultats très proches malgré la grande différence d'implémentation (voir Figure 70). La méthode temporelle présente un bruit plus élevé dans la bande 800 – 1000kHz par rapport à la modélisation fréquentielle. Cette différence provient vraisemblablement d'un phénomène de «peaking», qui est une augmentation du gain de boucle avant la fréquence de coupure. Cette augmentation est due à une marge de phase rendue trop faible par les retards engendrés dans les différents blocs (réponse du VCO, du diviseur, du comparateur de phases) qui ne sont pas pris en compte dans le modèle fréquentiel.

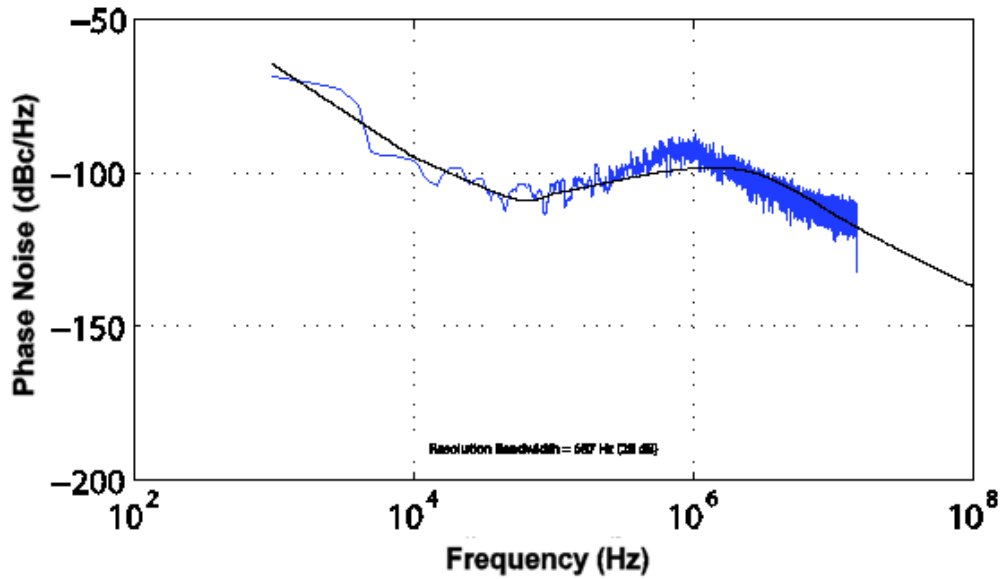


Figure 70 - Comparaison des deux méthodes

Cette modélisation nous a permis de vérifier que le VCO est le principal contributeur en termes de bruit de phase de la PLL, au-delà de la fréquence de coupure. Toutefois, on peut constater qu'en simulation système, ce bruit respecte le cahier des charges, à savoir -90dBc/Hz@1MHz. La pompe de charge possède un bruit relativement bas et le filtre de boucle est assez bruyant, à cause d'un faible courant de pompe de charge, mais non prédominant.

En dessous de la fréquence de coupure du filtre de boucle passe bas, on vérifie que le bruit de la référence est le principal contributeur.

5. Etude de génération des spurious

5.1. Introduction

Nous avons montré, au chapitre I, lors de l'étude de l'architecture d'une PLL à FPD, que cette topologie pouvait générer des raies parasites (ou spurious). Nous nous proposons dans ce paragraphe d'étudier ce phénomène dû au mismatch (ou désappariement) entre les temps de propagation des phases du VCO. Les fréquences où apparaissent les spurious sont des sous-multiples de la fréquence de comparaison et dépendent du nombre de phases utilisées (Tableau 8).

Tableau 8 - Fréquence d'apparition des spurious Vs phases utilisées (N parmi 30)

Nb phases utilisées	Fréquence des Spurious Vs. Fréquence de Référence						
	1/30	1/15	1/10	1/6	1/5	1/3	1/2
2							X
3						X	

Nb phases utilisées	Fréquence des Spurious Vs. Fréquence de Référence						
	1/30	1/15	1/10	1/6	1/5	1/3	1/2
5					X		
6				X		X	X
10			X		X		X
15		X			X	X	
30	X	X	X	X	X	X	X

Pour prédire la puissance de ces spurious en fonction de l'amplitude de la variabilité des erreurs de délai de propagation (mismatch), nous utilisons la technique de Ken Kundert. On exécute une simulation temporelle de la PLL en boucle fermée sur plus de 40000 périodes Fvcro. La FFT de ce signal donne la DSP du bruit de phase en sortie de la PLL.

En modifiant le modèle VerilogAMS précédent du VCRO afin d'ajouter un mismatch (aléatoire et réglable en amplitude) entre les phases de sortie, on peut simuler la PLL par la même méthode. Toutefois, lors de la nouvelle simulation, les spurious vont apparaître avec une puissance fonction de l'amplitude du mismatch. Pour plus de fidélité, on peut régler cette amplitude de mismatch avec la variabilité extraite du VCO au niveau transistor par une analyse MonteCarlo.

Une deuxième solution pour connaître la puissance des spurious est de simuler seulement la sortie du diviseur fractionnaire. Les phases avec mismatch sont réorganisées en fonction du rang de division puis répétées pour former une suite de périodes bruitées. On effectue ensuite une FFT pour extraire la puissance des spurious à la sortie du diviseur. Il convient de noter que cette transformée de Fourier doit être effectuée sur un nombre de points suffisamment important afin d'obtenir la résolution fréquentielle nécessaire correspondant à la largeur spectrale des spurious. Ces raies doivent être mises en forme par le gain en boucle fermée de la PLL, afin de fournir leur puissance en sortie. Les deux méthodes ont été utilisées comme moyen de double vérification des calculs.

5.2. Modèles

5.2.1. Méthode temporelle

Pour générer les spurious et le bruit de phase en même temps nous avons ajouté la génération d'un jitter déterministe à la sortie de la fréquence divisée du modèle de VCO à diviseur intégré. La fréquence du VCO n'est donc jamais générée, divisant le temps de simulation par le rapport de division. Le modèle permet de faire varier un grand nombre de paramètres liés au VCO et au diviseur :

- La fréquence générée (Vmin/max, Fmin/max, ratio de division)

- Le signal de sortie (tt - temps de transition, td - temps de propagation, Vhi/Vlo - tensions haute et basse)
- Les bruits de phase thermique du VCO et du diviseur (vcoJitter, dividerJitter)
- Et le jitter déterministe produisant les spurios (phaseMismatch)

Le modèle contient aussi une modélisation de la calibration avec une erreur absolue (cal_ps) et enregistre les périodes dans un fichier à partir d'un instant donné (outStart). Le fichier est ensuite traité par le même script Matlab que pour l'analyse temporelle précédente (Annexe 8) pour afficher la DSP du bruit avec les spurios.

Modèle VerilogAMS du Diviseur avec génération de spurios :

```
`define N 30 // from [3:100] number of phases

module vco (in, out);
input in; output out; electrical in, out;

parameter real Vmin=0;
parameter real Vmax=Vmin+1 from (Vmin:inf);
parameter real Fmin=1G from (0:inf);
parameter real Fmax=3G from (Fmin:inf);
parameter real outStart=1m from (0:inf);
parameter real ratio = 80 from (1:inf);
parameter real vcojitter = 0 from [0:0.25/Fmax]; // VCO period jitter
parameter real dividerJitter = 0 from [0:0.25*ratio/Fmax]; // VCO period jitter
parameter real phaseMismatch = 5p from [0:100p]; // Phase mismatch std value
parameter real cal_ps = 25p from [0:100p]; // Calibration top value
parameter integer seed_init = 1; // initialization pseudo random system
parameter real Vlo=-1, Vhi=1;
parameter real tt=0.1/Fmax from (0:inf);
parameter real td=1/Fmax from (10*(dividerJitter+phaseMismatch):inf);
parameter real ttol=1u/Fmax from (0:ratio/Fmax);
real MmVector[0:N-1]; //mismatch in picoseconds
real freq, phase, dT, dt, accJSD, syncJSD, detJSD, prev, Vout, divider;
integer n, accSeed, syncSeed, fp, i, Jump, phaseSD, divP, divN;

analog begin
@(initial_step) begin
    divN=ratio-0.5;
    divP=(ratio-divN)*N;
    divider = divN + divP/N; // real divider ratio
    accSeed = 561;
    syncSeed = 243;
    // phaseSeed = 864;
    accJSD = vcojitter*sqrt(2*divider); // why? correct output.
    syncJSD = dividerJitter;
    detJSD = phaseMismatch;
    Vout = Vlo;
    fp = $fopen("periods.m");
    $fstrobe( fp, "%0.10e", ratio);
    if(divP!=0) begin
        // Max common divider
        for ( i=1 ; i<=divP ; i=i+1 ) if( divP%i==0 && `N%i==0 ) Jump = i;
        $strobe("Jump %d for divP %d, divN %d", Jump, divP, divN);
        //generate variability
        for ( i=0 ; i<N ; i=i+Jump ) begin
            MmVector[i] = $rdist_normal( seed_init, 0, detJSD );
            if(MmVector[i]>cal_ps)
                MmVector[i]=cal_ps;
            if(MmVector[i]<-cal_ps)
                MmVector[i]=-cal_ps;
            $strobe("MmVector[%d]=%0.10e", i, MmVector[i]);
        end
    end
end
```

```

i=0;
end
// compute the freq from the input voltage
freq = (V(in) - Vmin)*(Fmax - Fmin) / (Vmax - Vmin) + Fmin;
// bound the frequency (this is optional)
if (freq > Fmax) freq = Fmax;
if (freq < Fmin) freq = Fmin;
// compute frequency with frequency divider
freq = (freq/divider)/(1 + dT*freq/divider);
// phase is the integral of the freq modulo 1
phase = idtmod(freq, 0.0, 1.0, -0.5);
// update jitter twice per period, trigger at +/- 1/4 = +/-pi/2
@(cross(phase - 0.25, +1, ttl)) begin
Vout = Vhi;
dT = $rdist_normal(accSeed, 0, accJSD);
dt = $rdist_normal(syncSeed, 0, syncJSD);
i = (i+divP)%N;
end
@(cross(phase + 0.25, +1, ttl)) begin
Vout = Vlo;
dT = $rdist_normal(accSeed, 0, accJSD);
if ($abstime >= outStart)
    $fstrobe( fp, "%0.10e", $abstime - prev);
prev = $abstime;
end
V(out) <+ transition(Vout, td+dt+MmVector[i], tt);
end
endmodule

```

5.2.2. Méthode fréquentielle

Pour générer les spurious au niveau fréquentiel, il faut calculer la TF du jitter déterministe produit par le FPD. Les mismatches des phases sont générés aléatoirement (variabilité) et pris en compte lors de la sélection de la phase correspondante par le multiplexeur (division fractionnaire). On répète cette opération jusqu'à obtenir une résolution fréquentielle suffisante pour effectuer la FFT (voir annexes 9 et 10).

5.3. Résultats

5.3.1. Simulation temporelle

La simulation temporelle est effectuée sous Matlab. Les résultats provenant du calcul de la DSP de bruit par la méthode de Kundert sont présentés sur la Figure 71. Le bruit de phase est identique aux simulations antérieures auxquelles s'ajoutent les spurious dus au jitter déterministe du diviseur fractionnaire.

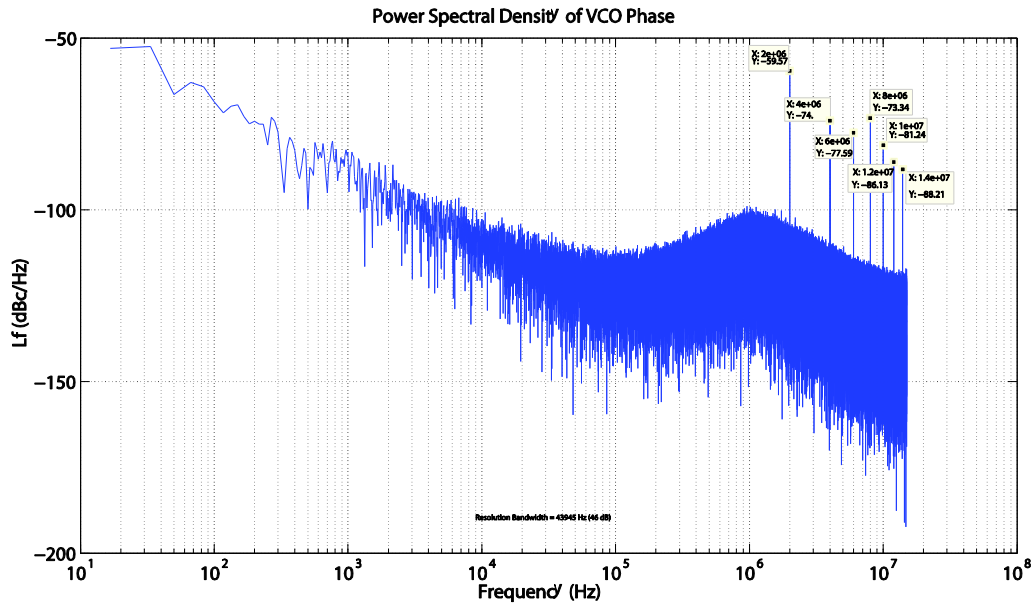


Figure 71 – Apparition des spurious de -60, -74, -78, -73, -81, -86 et -88dBc à 2, 4, 6, 8, 10, 12 et 14MHz

5.3.2. Simulation fréquentielle

La méthode de simulation fréquentielle du bruit de phase dans la PLL a été réutilisée et un modèle de génération des spurious a été ajouté. Le mismatch nominal entre les phases a été déterminé par des simulations SPICE (au niveau transistor) et estimée à 5ps_{RMS} . Après simulation, le spurious à 2MHz est estimé à environ -62 dBc (cf. Figure 72).

En diminuant ce mismatch à $0,26\text{ps}_{\text{RMS}}$, la puissance des spurious simulée est beaucoup plus faible : le spurious principal descend à -90 dBc (voir Figure 73).

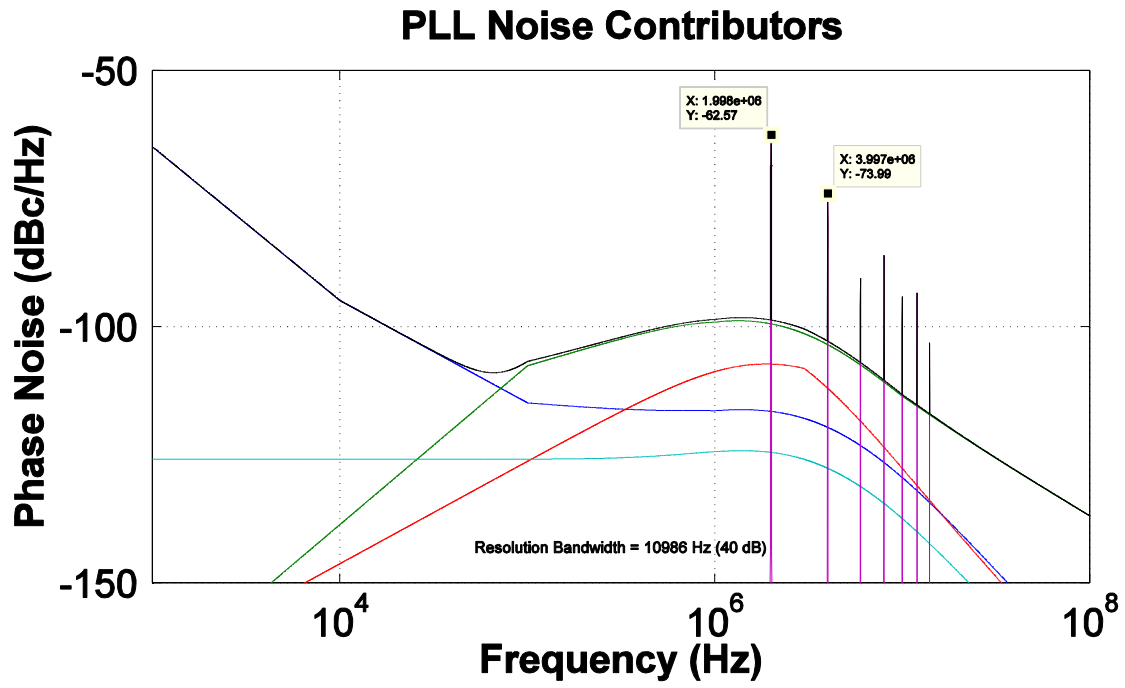


Figure 72 - Exemple de génération de spurious avec 5psRMS de mismatch

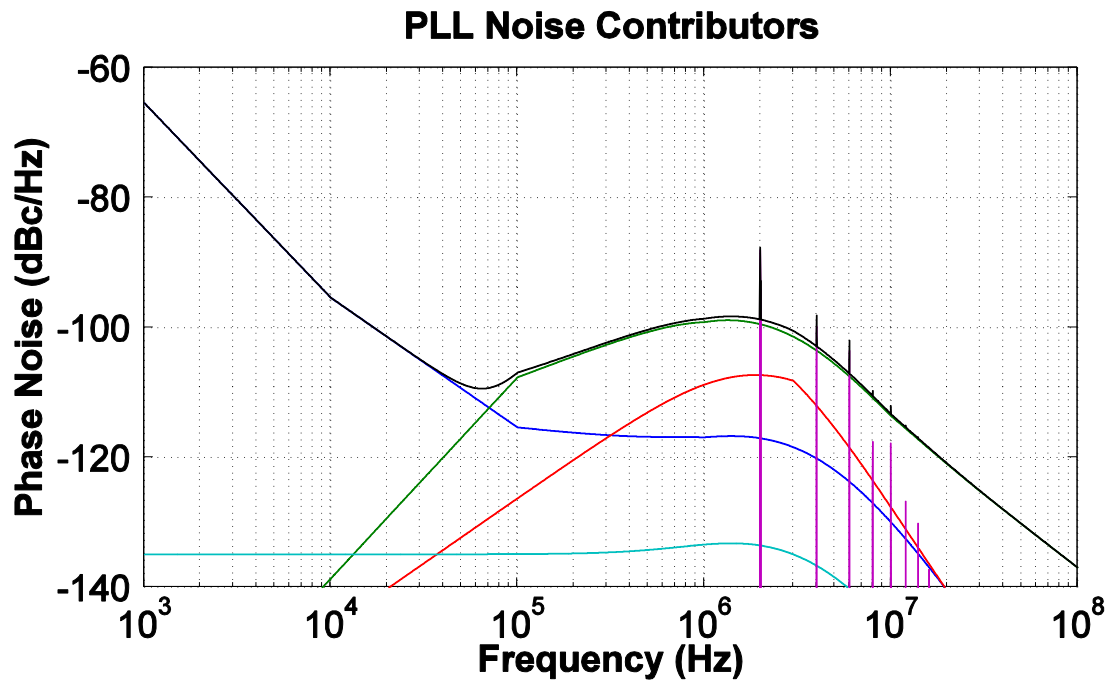


Figure 73 - Exemple de génération de spurious avec 0,26psRMS de mismatch

5.4. Conclusion

Le modèle de génération de spurious a permis de comparer l'impact des spurious par rapport au bruit de phase et ainsi aider à la conception du système. L'optimisation des paramètres de la PLL (bande passante, marge de phase, ordre du filtre, gain du VCO, bruit de phase admissible de chaque bloc à chaque fréquence) peut être effectué de façon itérative simplement et rapidement, en prenant en compte non seulement le bruit de phase mais aussi les spurious.

Les deux méthodes ne donnent pas exactement les mêmes résultats. Il est difficile dans l'état actuel de nos connaissances et sans avoir effectué de mesures, de déterminer laquelle est la plus précise. En revanche, la méthode fréquentielle est plus rapide et permet plus facilement de modifier les paramètres permettant d'optimiser la conception de la PLL.

La puissance des spurious extraite des simulations est satisfaisante (-60dBc à quelques MHz de la porteuse) par rapport aux architectures à l'état de l'art pour des consommations identiques ou proches (<6mW).

En tout état de cause, quelque soit la méthode de simulation (fréquentielle ou temporelle), nous observons qu'il est nécessaire de calibrer les sorties (phases) du VCO, afin de diminuer le mismatch qui doit être inférieur à 250fs.

6. Etude de calibration des phases

6.1. Introduction

Les PLL fractionnaires existantes font appel à différentes techniques pour diminuer l'amplitude des spurious (voir Chapitre I) mais celles-ci ne sont pas toutes applicables à l'architecture de la PLL étudiée (à resynchronisation de phases).

Par exemple, la compensation du bruit produit par le diviseur fractionnaire à base de modulateur sigma-delta est relativement simple à réaliser. En effet, l'erreur entre division instantanée et division moyenne est grande, jusqu'à une demi-période de la référence, et connue, malgré le caractère semi-aléatoire du modulateur sigma-delta.

Dans notre cas, l'erreur de phase produite par le diviseur fractionnaire à base d'oscillateur en anneau est faible (<10ps) et inconnue, car elle est due au processus de fabrication. Toutefois, cette valeur est fixe, car le mismatch d'une phase change peu au fil du temps. Il est donc nécessaire de mesurer et corriger cette erreur de phase conduisant à la production des spurious.

6.2. Architectures de calibration

En partant du constat que les spurious produits par la PLL sont le résultat d'un jitter déterministe (erreur systématique) dû à une ou plusieurs phases, il est alors possible de calibrer la phase en moyennant l'erreur. Deux techniques de calibration sont proposées et présentées ci-après.

6.2.1. Calibration par comparateur Bang-Bang, Filtre numérique et DAC

Cette méthode permet de se servir du jitter déterministe propagé vers le PFD pour le calibrer. La mesure de l'erreur se fait par le biais d'un comparateur bang-bang, dont la zone morte définit la précision de calibration, mis en parallèle au comparateur de la PLL. L'erreur moyenne est filtrée au moyen d'un filtre numérique, noté DF sur la Figure 74. Pour calibrer la ou les cellules provoquant l'écart de temps de propagation, cette architecture propose un DAC à chaîne de résistances (cf. Figure 74). La cellule calibrée du VCRO doit avoir une dynamique de 30ps (valeur maximale de gigue) et le DAC qui pilote sa grille arrière (avec une dynamique de 1V) doit avoir une résolution de 8 bits pour permettre une résolution temporelle de 130fs ($30/2^8=117fs$). Ses sorties, au nombre de 1 par phase (30 en tout), sont connectées par le biais d'interrupteurs CMOS (pass-gate) à l'une des tensions intermédiaire de la chaîne de résistances, entre VSS et VDD.

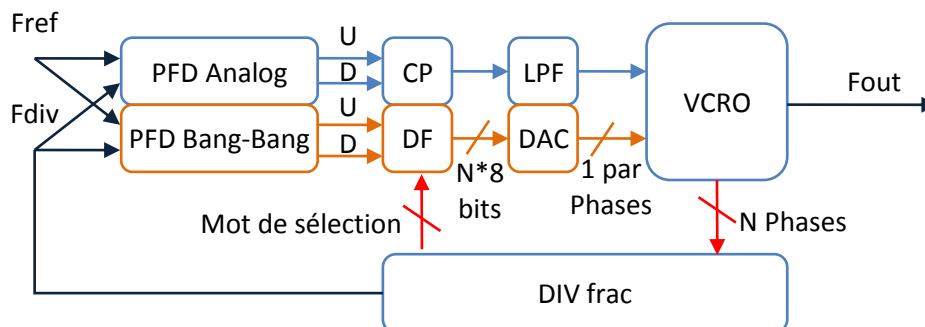


Figure 74 - PLL Hybride : boucle analogique et calibration numérique

6.2.2. Calibration par Vtune distribué par multiplexeur analogique

Cette méthode de calibration, comme la précédente, utilise l'erreur moyenne disponible au PFD pour effectuer la calibration, mais cette fois, la boucle de calibration est intégrée à celle de la PLL. Ceci est possible en séparant le Vtune en autant de commandes qu'il y a de phases et en calibrant uniquement la phase qui est sélectionnée. Pour effectuer cette sélection de la commande Vtune, un multiplexeur analogique est nécessaire, rendant cette topologie très complexe à implémenter, comme illustrée par la Figure 75.

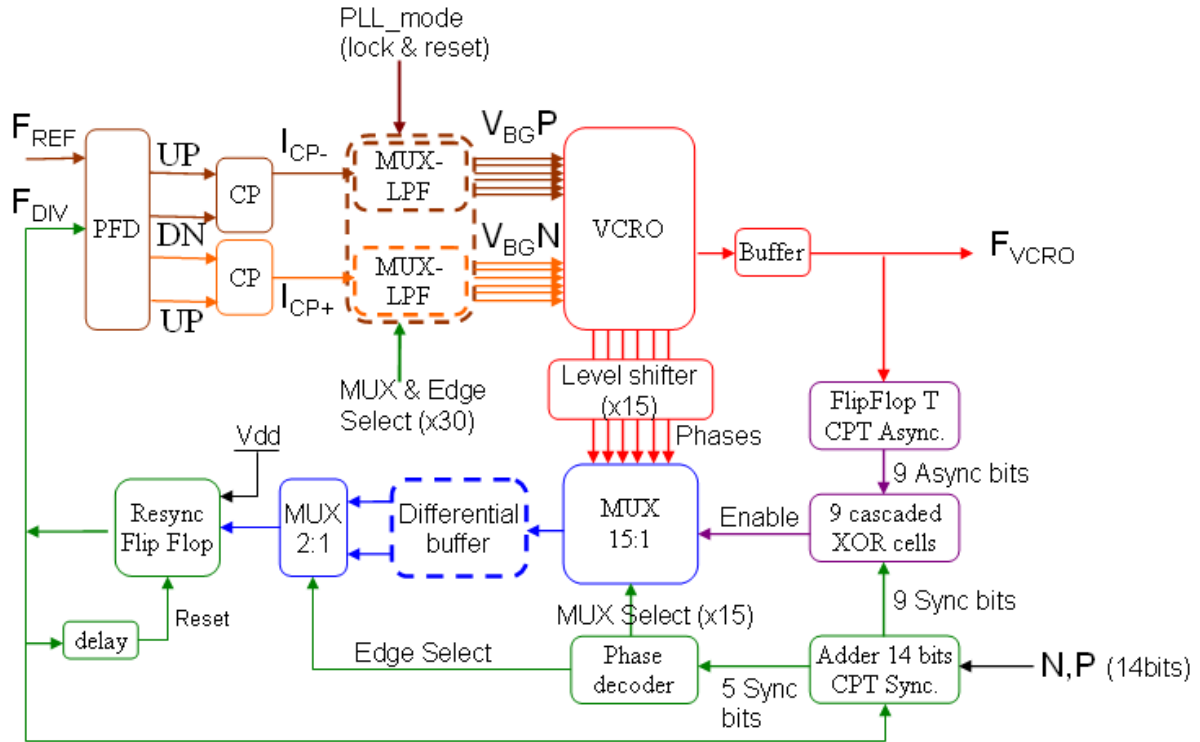


Figure 75 - PLL à boucle et calibration analogique

6.2.3. Conclusion sur l'architecture de calibration

La première technique de calibration est numérique, donc économe en surface de silicium. En outre, elle est plus linéaire que la version analogique à cause de la fuite des grilles arrières et des mises à jour à la fréquence du pas de synthèse (1 ou 2MHz). Les deux architectures de calibration présentées ne sont que des ébauches demandant d'avantage de modélisation et de simulation. La partie suivante présente un schéma de génération de spurious qui permet de simuler la calibration progressive d'une PLL.

6.3. Modélisation

La modélisation suivante permet de calculer rapidement la puissance des spurious correspondant à une valeur particulière des mismatches pour vérifier la diminution de la puissance des spurious.

Pour diminuer le temps de la simulation, la PLL est étudiée en boucle ouverte. Un premier VCRO génère un signal divisé sans mismatch et un autre avec pour simuler les signaux F_{div} et F_{ref} . Ces signaux sont envoyés à un comparateur de phases (PFD + CP) dont le signal filtré va moduler l'entrée d'un 2ème VCRO (idéal). Cette méthode permet de mesurer la modulation autour de la

Modèle VerilogAMS du VCRO à sortie divisée avec et sans jitter déterministe :

```

`define N 30 // from [3:100] number of phases

module VCO_1phases(ain, fosc_out);

parameter real f0 = 2.45e9 from (0.0:inf]; // Output center freq
parameter real fmin = 2.3e9; // Minimum allowable output freq
parameter real fmax = 2.6e9; // Maximum allowable output freq
parameter real V_f0 = 0.5; // Input voltage corresponding to f0
parameter real Kv = 500.0e6 from (0.0:inf]; // VCO linear gain (Hz/V)
parameter real Vhi = 1.0; // Output voltage corresponding to '1'
parameter real Vlo = 0.0 from [-inf : Vhi]; // Output voltage corresponding to '0'
parameter real Vth = 0.5 from (Vlo:Vhi); // Threshold voltage
parameter real tTR = 10p from (0.0 : 10p); // Output Fout Transition Time

input ain;
electrical ain;
output fosc_out;
electrical fosc_out;

real ViLim, halfPeriod;
real V_fmax, V_fmin;
integer state;

analog begin

    // Initialization
    if (analysis("static")) begin
        state = 1; // init node at 1
        V_fmax = V_f0 + (fmax - f0)/Kv; // Input corresponding to fmax
        V_fmin = V_f0 - (f0 - fmin)/Kv; // Input corresponding to fmin
        halfPeriod = 0.5/(f0 + Kv * ViLim);
    end

    // Clamp input voltage to limit frequency excursion
    if (V(ain) > V_fmax) ViLim = V_fmax - V_f0;
    else if (V(ain) < V_fmin) ViLim = V_fmin - V_f0;
    else ViLim = V(ain) - V_f0;

    // VCO (update frequency) and dependant variables
    @(timer(0,halfPeriod)) begin
        state = !state;
        halfPeriod = 0.5/(f0 + Kv * ViLim);
    end

    V(fosc_out) <+ transition(state?Vhi:Vlo , 0, tTR);

end
endmodule

```

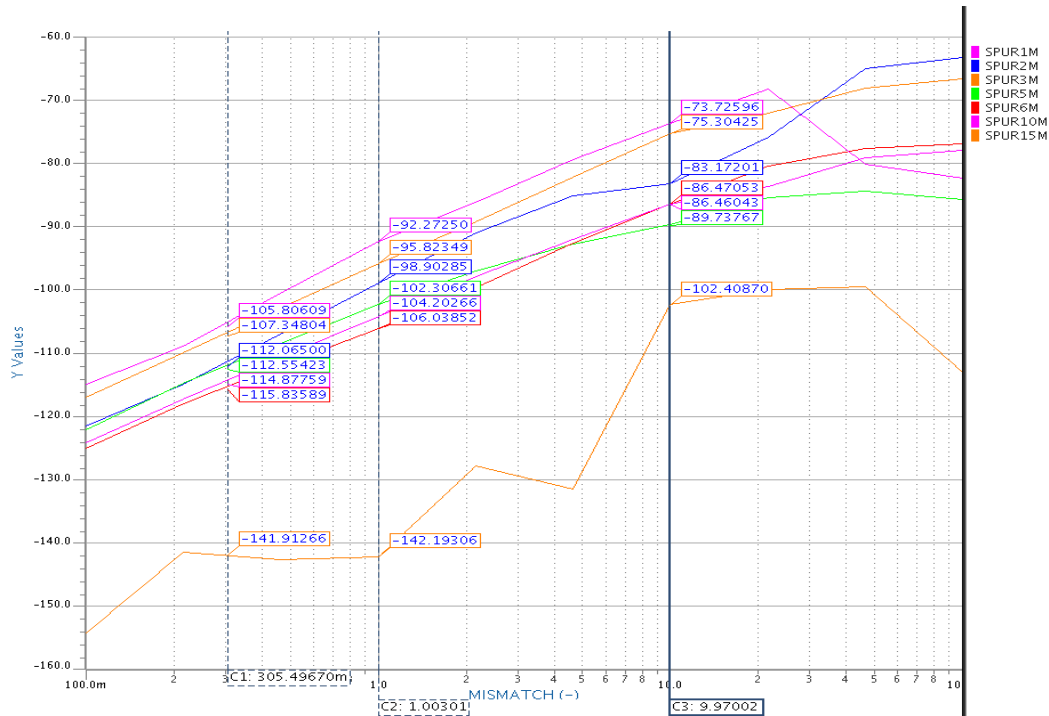


Figure 77 - Résultats de simulation de l'effet de la résolution de mesure de l'erreur sur la puissance des spurios

6.5. Conclusion

Les architectures de calibration étudiées n'atteignent pas des suppressions de spurios inférieure à -90dBc/Hz à la fréquence du pas de synthèse (1MHz dans l'étude) mais ont l'avantage de présenter des architectures techniquement réalisables dans la technologie étudiée (FDSOI 28nm).

Ainsi le filtre numérique décrit dans la première architecture de calibration présentée est facilement synthétisable (filtre passe bas numérique) et occupe peu de surface silicium et le multiplexeur de l'architecture analogique est aussi assez simple à réaliser. Néanmoins les aspects de fuite/autodécharge et isolation/injection de charges sont des points sensibles pour la réalisation de ce multiplexeur analogique. Quelle que soit l'architecture de calibration retenue, elle devra limiter le mismatch à une valeur inférieure à 150fsRMS.

7. Conclusion

Dans ce chapitre, nous avons modélisé puis simulé les modes de fonctionnement de la PLL à FPD et nous avons validé sa faisabilité. Puis nous avons étudié le bruit de phase dans cette PLL, qui est identique à une PLL entière pour définir les spécifications globales et de chaque bloc de la PLL.

Connaissant le principal défaut de cette architecture à base de diviseur fractionnaire sur phases (génération de spurious), nous les avons pris en compte dans les simulations afin de proposer une architecture permettant d'en limiter les effets. Ces simulations nous ont permis de définir le maximum de mismatch que devaient présenter les différentes phases du VCO.

Deux méthodes de calibration ont été proposées sans conclure, à ce niveau d'étude, sur la plus efficace des deux. Toutefois, elles nous ont d'ores et déjà permis de valider les spécifications à atteindre, à savoir $150f_{\text{RMS}}$ de désappariement entre phases.

En conclusion, ce chapitre nous a permis de retenir comme architecture, la PLL à diviseur fractionnaire à resynchronisation, et de valider le cahier des charges proposé au chapitre I en ajoutant une spécification supplémentaire : erreur (mismatch) entre phases inférieure à $150f_{\text{s}}$.

Dans le chapitre suivant nous concevrons et réaliserons plusieurs topologies de VCRO, puis nous étudierons l'implémentation finale de la PLL.

Chapitre III – Conception, réalisation et test de VCRO

1.Introduction

Ce chapitre présente les 4 versions différentes de VCRO qui ont été réalisées et mesurées. Un buffer « 50 Ω » a été conçu et optimisé pour extraire le signal des VCRO à 2,45GHz et sera présenté au chapitre IV. Un circuit test (ou test chip) a été réalisé pour mesurer les 4 VCRO (3 oscillant à la fréquence de 2,45GHz et 1 à 10 GHz avant division par 4). Les autres parties de la PLL seront décrites au chapitre IV.

L'architecture à base d'oscillateur en anneau à nombreuses phases a été choisie comme oscillateur commandé (VCRO) pour utiliser ces phases à des fins de division fractionnaire. Le pas de synthèse étant égal à la fréquence du quartz divisée par le nombre de phases de l'oscillateur, un nombre croissant de phases permet une fréquence de quartz plus élevée, à pas de fréquence identique, et donc une bande passante plus élevée. Pour choisir l'architecture qui a les meilleures performances (i.e. meilleure FOM, cf. relation 7 chapitre I), plusieurs architectures d'oscillateurs ont été testées (cellules délai, longueur et fréquence différentes) et 4 d'entre elles ont été retenues pour implémentation, fabrication et mesure, à des fins de comparaison.

Les cellules délai utilisées sont le fruit d'une sélection et optimisation pour leur consommation et leurs performances en bruit de phase. Toutes les autres cellules délai étudiées avaient une FOM (bruit de phase sur consommation) moins bonnes mais présentaient d'autres qualités (pushing plus faible, rapport cyclique plus proche de 50%, etc.). On trouvera une description détaillée de cette étude sur les cellules délai au chapitre I. Pour simplifier, moins il y a d'échanges de charges et plus l'oscillateur est économe en énergie, inversement plus les transistors sont grands et moins il y a de bruit de phase. Ces deux objectifs sont antagonistes mais il est possible de trouver un compromis. En effet, diminuer la consommation tout en diminuant le nombre de transistors qui échangent des charges à chaque commutation permet d'augmenter la taille de ces transistors pour diminuer le bruit. Cette constatation nous a permis de réaliser des cellules à seulement 2 ou 4 transistors. Les simulations et optimisations ont été effectuées sous Cadence avec le simulateur Eldo et ont permis par raffinement des dimensions des transistors d'atteindre les spécifications fixées.

Les lignes de propagation des signaux hautes fréquences (2,5 ou 10GHz) sont très sensibles à la qualité portée au niveau du dessin des masques : la capacité parasite compte pour plus de 20% dans

le délai de propagation et donc de la même manière sur la consommation et la fréquence du VCO. Les lignes de signaux sont donc routées en métal 5 à 7 en déplacement horizontal, soit à bonne distance des transistors, des alimentations/polarisations (métal 1), des autres lignes de signaux (au même niveau) et des lignes/vias d'alimentation/polarisation (dessus/cotés, métal 8 à métal 1).

La résistance des lignes n'a par contre que peu d'impact sur le temps de propagation et le bruit de phase. Leur valeur est d'environ 10Ω entre deux transistors, ce qui est négligeable par rapport à la résistance R_{ON} des transistors, qui est de l'ordre de 100Ω à $10 \text{ k}\Omega$ ($L=30\text{nm}$ et $W=8\mu\text{m}$ à 80nm). Leur largeur a ainsi été choisie de taille minimale, c'est à dire 50 nm sur les métaux 1 à 6 et 150 nm sur les métaux 7 et 8.

2.Spécifications

Pour rappel, les spécifications qui ont été fixées pour donner une dimension réaliste au design de cette PLL sont les suivantes :

- Puissance consommée par la PLL : 1mW dont $0,8\text{mW}$ alloué au VCO,
- PLL à bruit modeste : $-90\text{dBc}/\text{Hz}@1\text{MHz}$,
- Fréquence de sortie de $2,45\text{GHz}$ et pas de synthèse de 1MHz .
- Dispersion des phases de moins de 5ps d'écart type (avant calibration).

Les valeurs simulées de consommation et de fréquence nominale ont toujours été effectuées avec les buffers de sortie en charge et avec les parasites extraits du layout ajoutés. Les simulations en bruit de phase ont été effectuées avec une simulation Steady State Transient alliée à une simulation du bruit de phase (SST + Noise) à la fréquence nominale des VCRO.

Les dispersions technologiques influent sur les performances des transistors (un buffer petit peut avoir une grande variation du temps de propagation). Le temps de propagation entre phases a été simulé par analyse de MonteCarlo pour vérifier que l'écart type du mismatch respectait les spécifications, elles-mêmes liées à la puissance des spurious générés par la PLL fractionnaire.

Par ailleurs, nous n'avons pas attaché beaucoup d'importance au rapport cyclique des différents VCRO. En effet, celui-ci peut facilement être ramené à 50% par les buffers de sortie (réglage de $\pm 4\text{ps}$ des fronts de montée et descente). Un réglage de ce rapport cyclique peut encore être réalisé par le circuit qui double le nombre de phases (dans le cas des VCRO single-ended).

Les buffers ont donc été conçus pour avoir une consommation faible (par rapport à l'oscillateur) tout en présentant une immunité importante aux déviations technologiques pour

limiter le mismatch entre les phases. Dans la pratique, lorsque nous parlons de mismatch entre phases du VCRO, nous avons inclus celui du buffer.

Trois sources participent à ce mismatch (Figure 78), le premier buffer (σ_{T1}), la cellule délai (σ_{T2}) et le deuxième buffer (σ_{T3}). T1 et T3 peuvent atteindre 10ps. La variabilité du procédé de fabrication influencera d'autant plus cette valeur que leurs dimensions seront faibles.

T2 dépend du nombre de cellules et de la fréquence du VCRO ($T = 1/(F_{vcro} * N)$) et peu des variations technologiques car les dimensions (W et L) des transistors (inverseurs) sont grandes pour minimiser le bruit de phase.

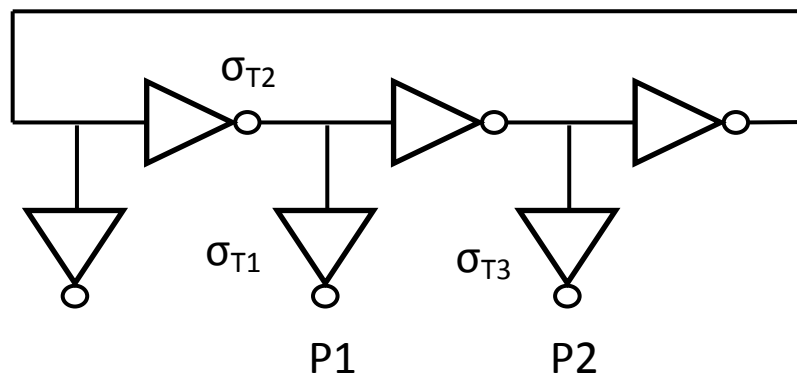


Figure 78 - Mesure de mismatch entre phases

3. Implémentation transistor, Layout & Simulation

3.1. VCRO single ended à 6 phases (3 inverseurs)

3.1.1. Introduction

Le premier oscillateur conçu est un VCRO (Voltage Controlled Ring Oscillator) à 3 inverseurs, soit le nombre minimum. Cette architecture a été choisie car, d'après les simulations effectuées, c'est l'oscillateur qui a la meilleure FOM, mais il est peu adapté à la FPD car il a seulement un rapport 6 entre la fréquence de référence et le pas de synthèse. Il a aussi servi de banc d'essai/apprentissage à l'optimisation des mismatches et des capacités parasites qui ont été utilisés pour les autres oscillateurs.

3.1.2. Implémentation

Le VCRO (cf. Figure 80) est composé de 3 cellules délai (RO<0:2>, Figure 79 gauche) et de 3 buffers de sortie de phase (Buf<0:2>, Figure 79 droite). Chaque cellule délai du VCO est composée de seulement 2 transistors, et chaque transistor est disposé dans un caisson d'isolement (Triple Well) pour que la grille arrière soit isolée électriquement (du substrat et des autres grilles arrières). Cela n'est possible qu'en utilisant un Pmos Low-Vth et un Nmos Normal-Vth (voir Figure 25 du premier chapitre sur la technologie FDSOI, types de transistors).

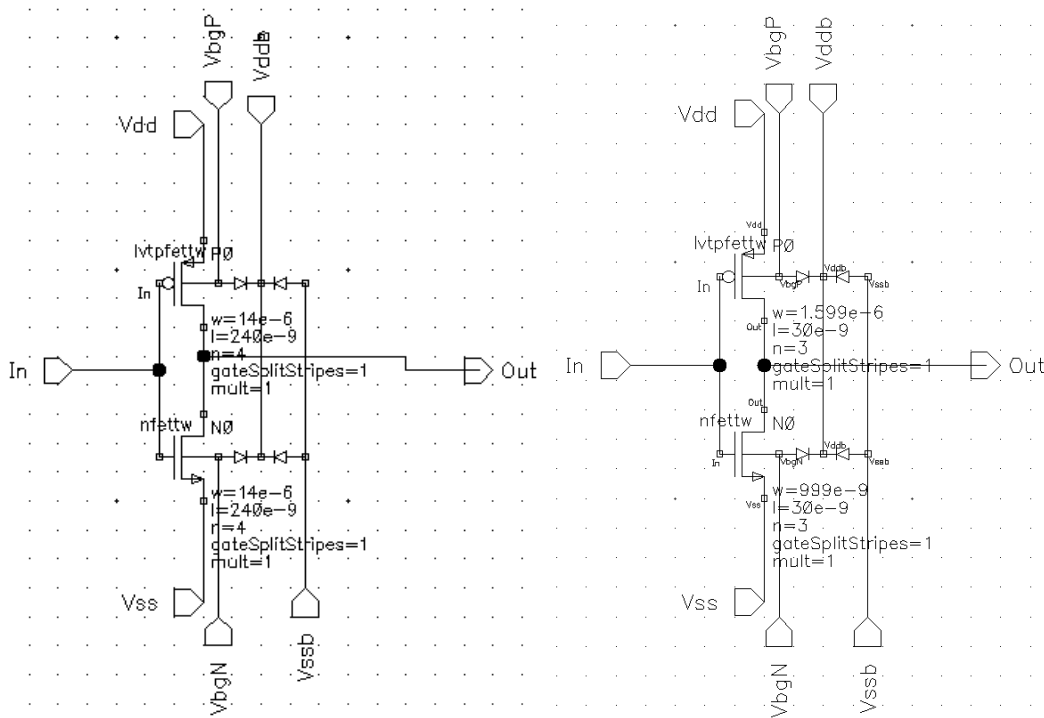


Figure 79 - Inverseur du VCO à 3 phases (gauche) et buffer de sortie (droite)

Les dimensions $W*L$ des transistors Pmos et Nmos du VCO sont $14\mu\text{m} \times 240\text{nm}$. Ce rapport Pmos/Nmos atypique ($=1$) permet d'avoir une meilleure FOM (bruit de phase et consommation faible à fréquence fixe) qu'un rapport favorisant un rapport cyclique de 50% ($=1,4$ pour P-LVt et N-NVt).

Les buffers de sortie de VCO (Figure 79) sont de même type mais le caisson d'isolement est commun aux 2 transistors de chaque phase. Leurs tailles ($W*L$) est de $1,6\mu\text{m} \times 30\text{nm}$ pour le Pmos et de $1\mu\text{m} \times 30\text{nm}$ pour le Nmos, ce rapport P/N permet de rattraper le décalage en rapport cyclique du VCO et d'améliorer le slewrate du circuit, ce qui diminue la génération de bruit de phase.

Les différents condensateurs disposés sur le schéma de la Figure 80 représentent les capacités parasites extraites après optimisation du layout (Figure 85). Les valeurs extraites sont : $C_{out_{VCO}}=5,8\text{fF}$, $C_{inout_{VCO}}=2\text{fF}$ (miller VCO), $C_{out_{BUF}}=0,8\text{fF}$, $C_{inout_{BUF}}=0,2\text{fF}$ (miller Buffer).

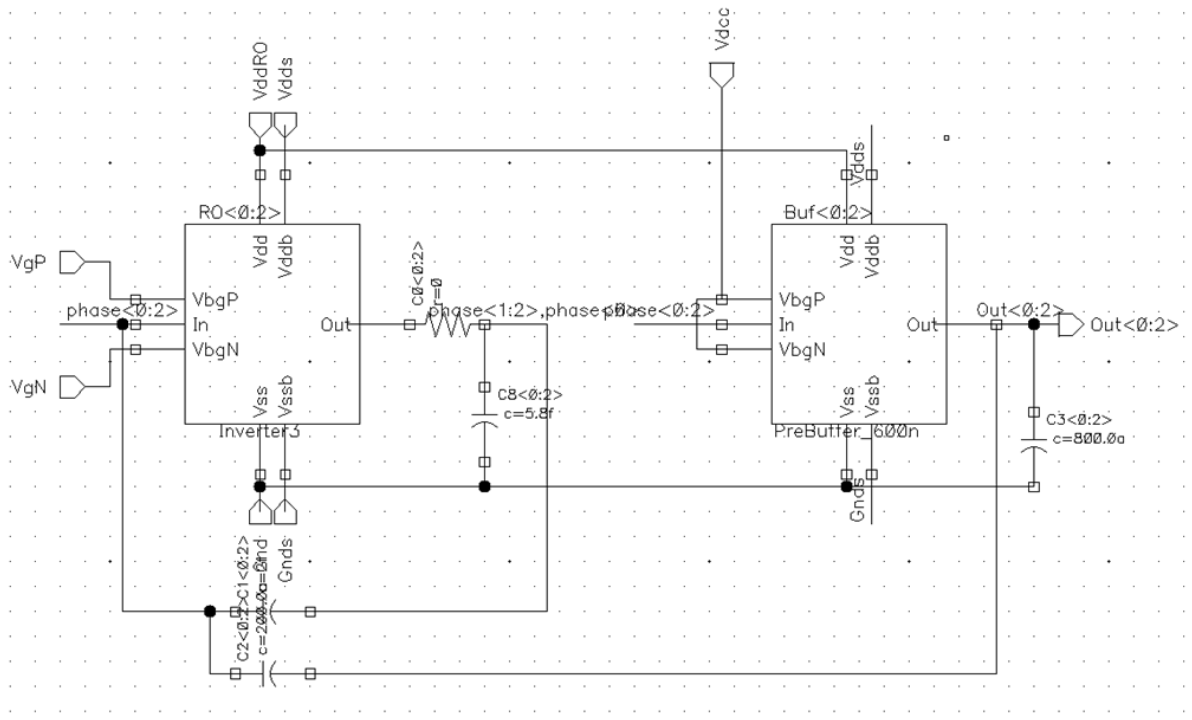


Figure 80 - Schéma du VCO (Inverseur x3) et des buffers de sortie (x3)

Sur la Figure 81, on peut voir comment ont été disposées les commandes en fréquence du VCO (V_{BG_N} en vert foncé et V_{BG_P} en orange) sur les grilles arrières des Nmos et Pmos des cellules délai du VCO (en bleu) et la commande de correction du rapport cyclique ($V_{BG_{BUF}}$ en violet) sur les grilles arrières des transistors des buffers (en vert clair). Dans toutes les simulations, $V_{BG_{BUF}}$ sera fixée à une valeur permettant d'obtenir un rapport cyclique de 50%. V_{BG_N} correspondant à la tension de commande ($V_{tune}=V_{GN}$) du VCRO et V_{BG_P} en complémentaire ($V_{GP} = V_{dd} - V_{GN}$).

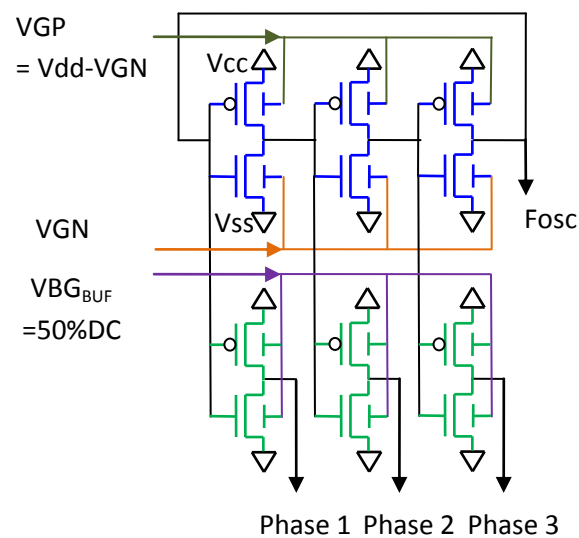


Figure 81 - Schéma des commandes en fréquence et rapport cyclique

3.1.3. Layout

L'implémentation au niveau layout de ce VCRO (Figure 85), permet de voir les 7 plots d'entrée et les 3 plots de sortie. Les entrées permettent d'alimenter le VCRO et les buffers de sortie ($V_{dd_{VCO}}$ et $G_{nd_{VCO}}$), de polariser les caissons d'isolement ($V_{dd_{TW}}$) et le substrat ($G_{nd_{SUB}}$), de modifier la fréquence du VCO (V_{bg_P} et V_{bg_N}) et le rapport cyclique des buffers de sortie ($V_{bg_{BUF}}$). Les 3 sorties ($Out_{<2 :0>}$) permettent de générer ensuite les 6 phases du VCO à la fréquence du VCO (2,45GHz) avec un déphasage de 60° ($360/6$). Les transistors Nmos et Pmos du VCRO sont alimentés (V_{dd} , G_{nd}) par le dessus pour diminuer la résistance série : chaque transistor est alimenté par ses propres vias à partir du métal 2 (bleu ciel), pour que la résistance sur l'alimentation soit très faible et que la chute de tension due à l'appel de courant d'une cellule délai ne soit pas transmise à une autre par l'alimentation.

Les lignes entre les cellules délai (Figure 82) ont été dessinées de manière à avoir une longueur strictement identique et une capacité parasite extraite très faible (5,8fF). En effet ces lignes ont été réalisées en métal 5 avec plus d'un micromètre d'isolant dans toutes les directions et de valeur identique, car de longueur et espacement identiques, pour ne pas ajouter du mismatch au délai de propagation entre phases.

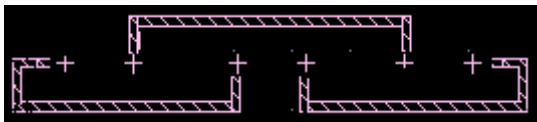


Figure 82 - Détail du layout des lignes entre cellules délai



Figure 83 - Détail du layout des lignes entre cellule délai et buffers

Les lignes entre cellules délai et buffer (Figure 83) ont été optimisées de façon à ce que leurs longueurs respectives soient identiques pour obtenir un mismatch de délai de propagation faible et donc un délai entre phases identique (avant fabrication). De plus il y a une différence importante entre le schéma idéal d'un oscillateur en anneau et la réalité de son implémentation au niveau layout (Figure 84). Cette alternance de l'orientation des cellules permet de diminuer la longueur moyenne des lignes de propagation, diminuant la fréquence maximale du VCO mais rendant les temps de propagation entre phases quasi-identiques.

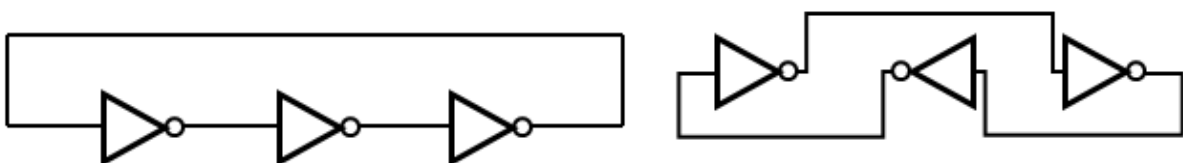


Figure 84 - Différence entre schéma idéalisé (à gauche) et schéma de layout réel (à droite)

Nous reprenons sur la Figure 86 la représentation layout du VCRO 1 en ayant supprimé les lignes d'alimentation. Cette figure nous permet de mieux visualiser les cellules délai et les buffers.

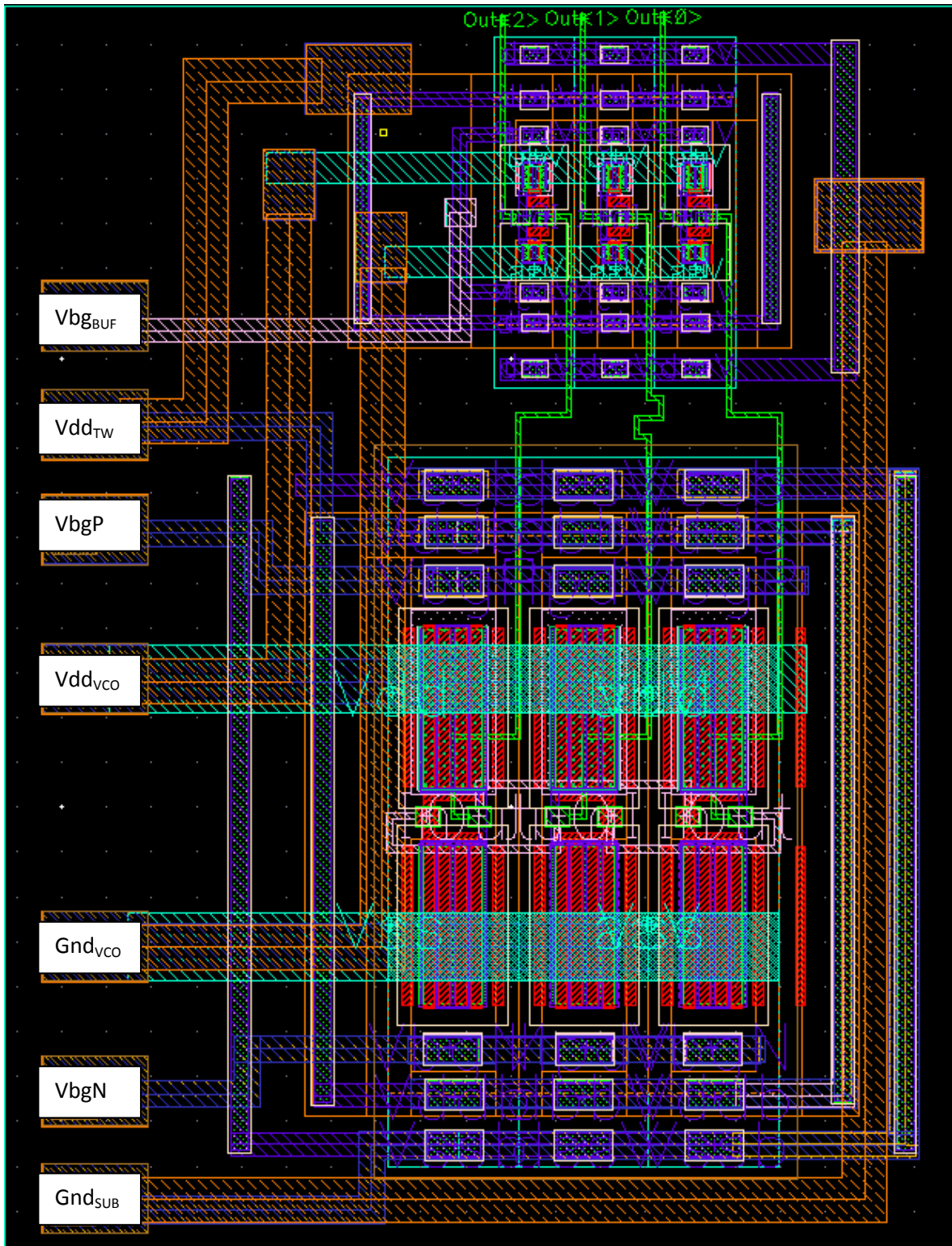


Figure 85 - Layout du VCO n°1 - Single Ended 3 cellules

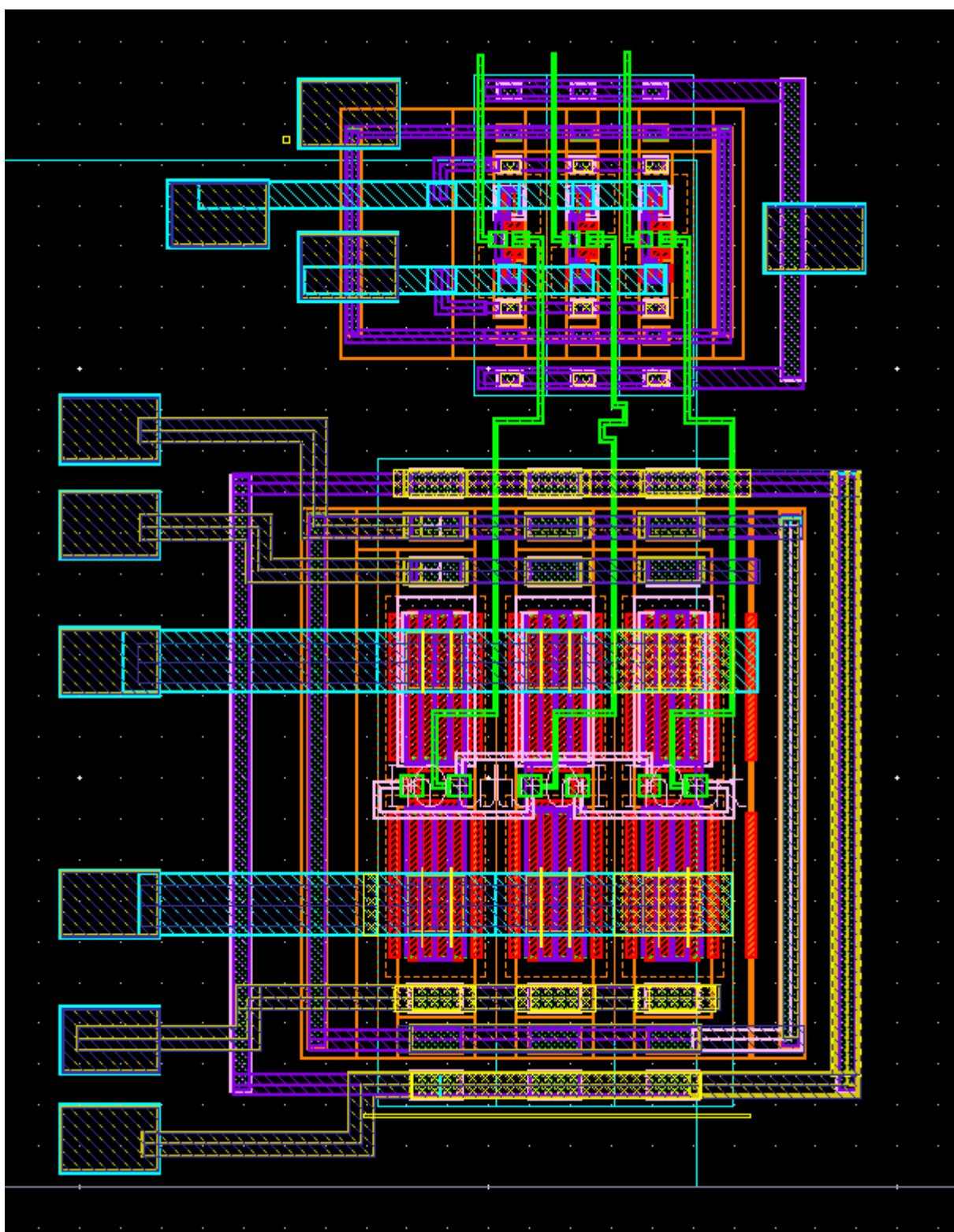


Figure 86 - Layout VCO 1 sans lignes d'alimentation

3.1.4. Simulation

Sur la Figure 87 on peut observer la variation de la fréquence du VCRO et du courant consommé en fonction de la tension de la grille arrière des transistors Nmos (VGN) pour une tension d'alimentation de 1V. La grille arrière Pmos est commandée en opposition ($VGP = V_{dd} - VGN$). On constate que la fréquence nominale est de 2530MHz avec un gain de 556MHz/V à $VGN = 0,5V$. A cette tension nominale, la consommation est de 860 μA avec une variation de 242 $\mu A/V$. La Figure 88 montre le même phénomène mais cette fois en fonction de la tension d'alimentation qui varie de plus ou moins 10%. La fréquence nominale est de 2517MHz avec un gain de 5,35GHz/V et la consommation nominale est de 878 μA avec une variation de 3mA/V avec la tension nominale $VGN=0,5V$.

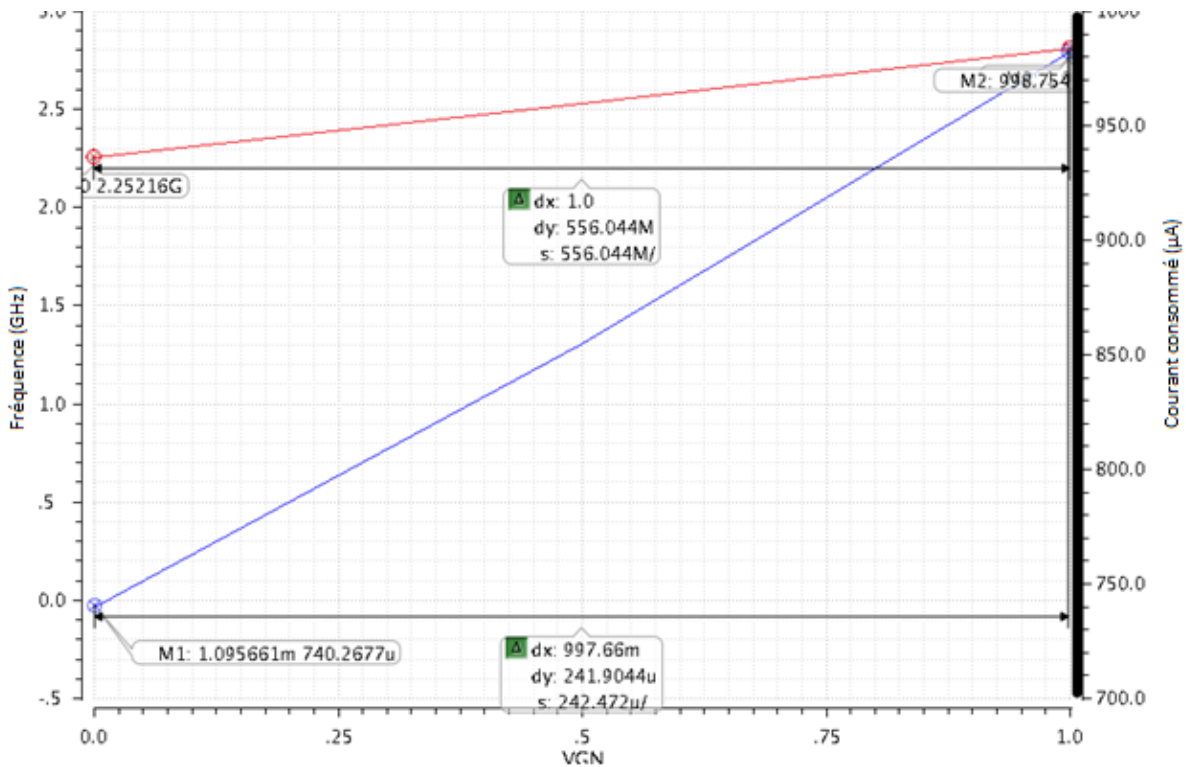


Figure 87 - Fréquence (rouge) et consommation (bleu) du VCO à 6 phases en fonction des tensions de grille arrière

Les simulations MonteCarlo (cf. Figure 89) montrent la fréquence de sortie et le courant consommé par le VCO, nous n'avons pas représenté l'histogramme classique d'une simulation MC afin de mettre en évidence la concordance entre la consommation de courant et la fréquence d'oscillation. On vérifie ainsi que le courant est proportionnel à la fréquence en tenant compte des variations technologiques. L'écart type de la fréquence de sortie est de 45,4MHz autour de 2,53GHz (50 tirages). L'écart type des temps de propagation entre phases est de 640fs à la montée et 745fs à la descente.

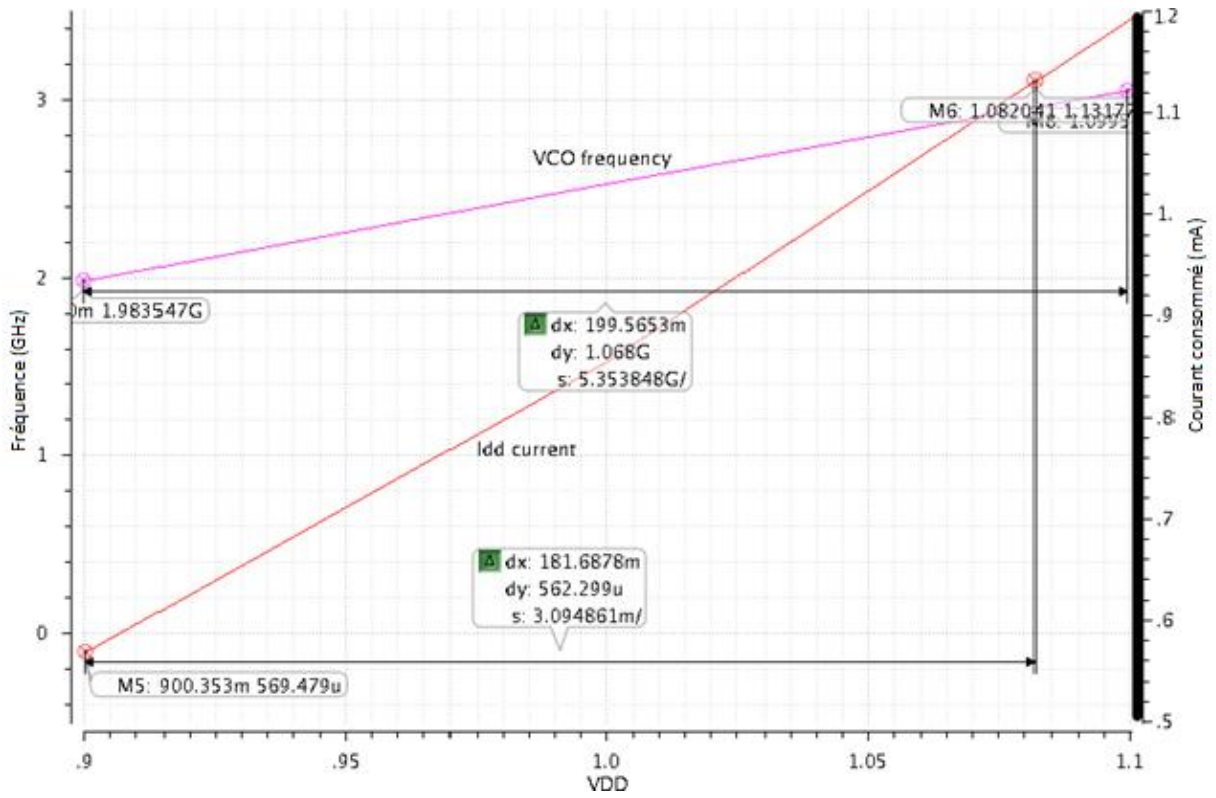


Figure 88 - Consommation et fréquence du VCO à 6 phases en fonction de la tension d'alimentation

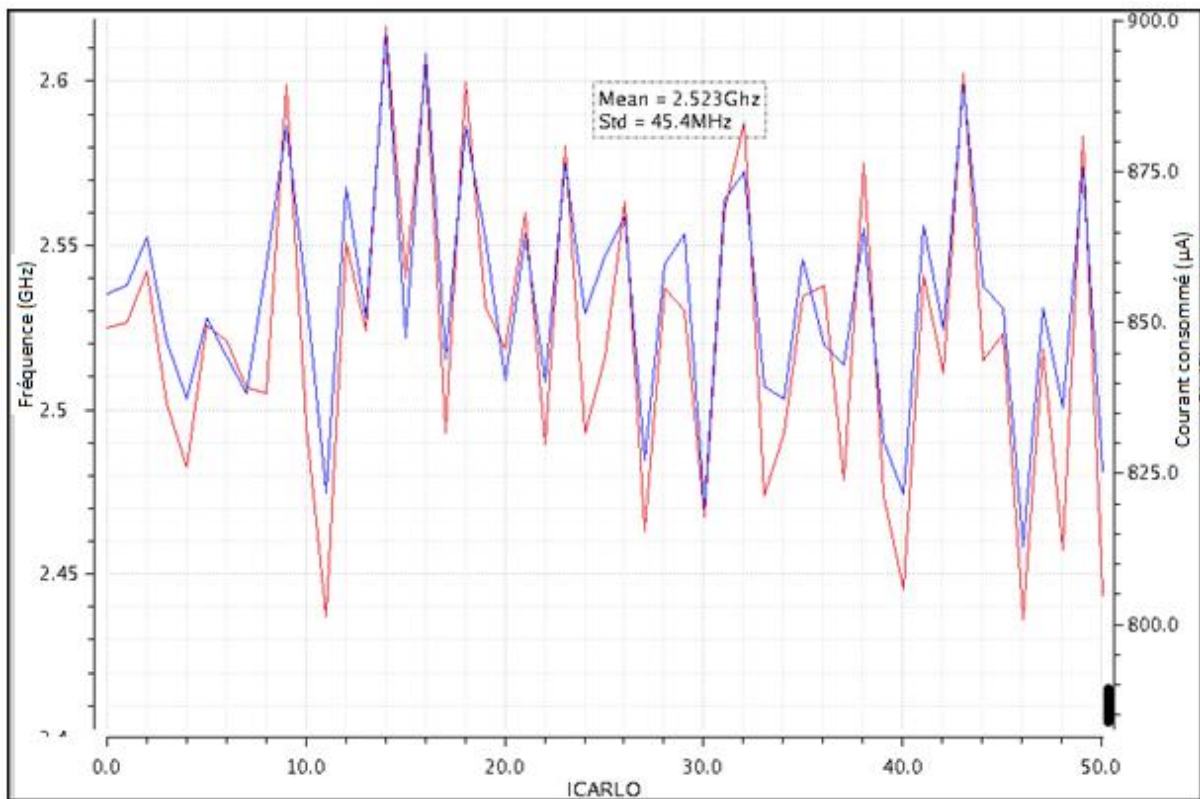


Figure 89 – Simulations MonteCarlo (fréquence et courant consommé) du VCRO n°1

3.2. VCRO single ended à 30 phases (15 inverseurs)

Le deuxième oscillateur conçu est un VCRO à 15 inverseurs, qui correspond à la valeur maximale utilisable pour la bande 2,4GHz. Ce VCRO a été choisi car le plus adapté à la division fractionnaire par phases. Grâce à ses 15 inverseurs, il permet un rapport 30 entre la fréquence de référence et le pas de synthèse, soit une référence de 30MHz (quartz à fréquence fondamentale) pour des canaux espacés de 1MHz (Bluetooth) ou 2MHz (BLE).

3.2.1. Implémentation

Le VCRO présenté sur la Figure 91 est composé de 15 cellules délai (RO<0:14>) détaillées en Figure 90, et de 15 buffers de sortie (Buf<0:14>) identiques au VCO précédent. Les cellules et les buffers sont conçus de la même manière que le VCO n°1. Les dimensions WxL des transistors sont de $W=4\mu\text{m}$ et $L=72\text{nm}$ pour le Pmos et le Nmos. Le rapport Pmos/Nmos atypique ($=1$) permet une fois encore d'obtenir une meilleure figure de mérite (FOM).

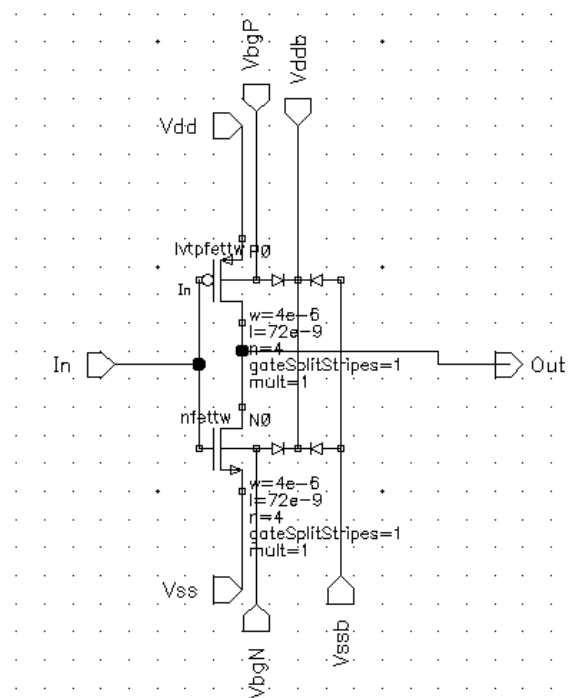


Figure 90 - Cellule Délai du VCRO à 15 cellules

Les différents condensateurs disposés sur le schéma du VCO (Figure 91) représentent les capacités parasites extraites après optimisation du layout (Figure 92). Les valeurs extraites sont : $C_{out_{VCO}}=1\text{fF}$, $C_{inout_{VCO}}=0,67\text{fF}$ (miller VCO) et $C_{out_{BUF}}=0,5\text{fF}$.

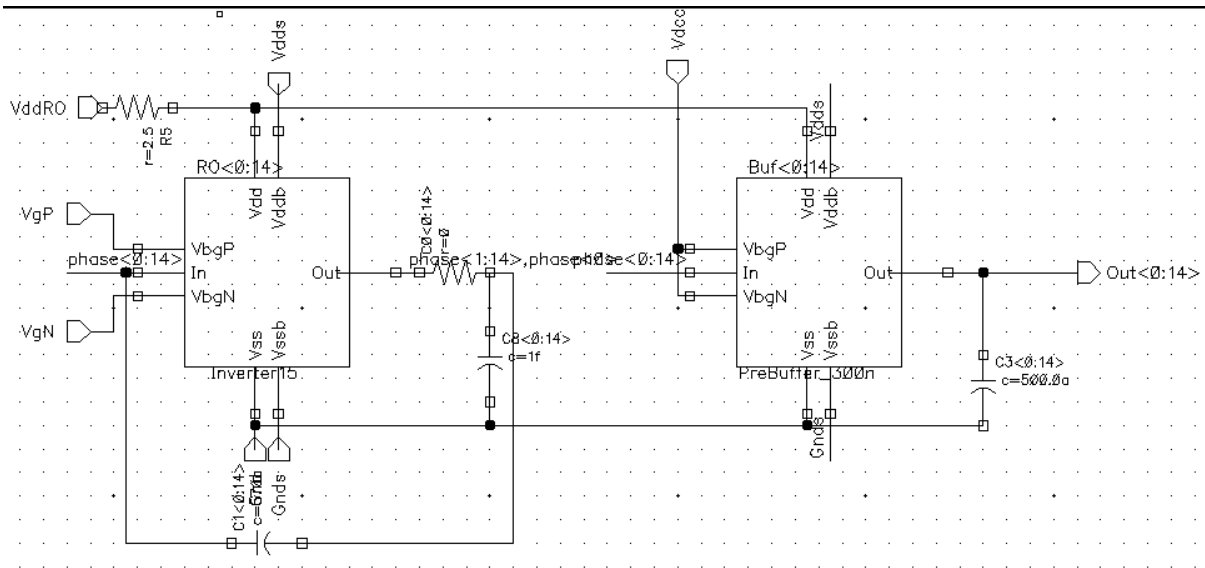


Figure 91 - Schéma du VCRO (15 inverseurs) et buffers (15 inverseurs)

3.2.2. Layout

Sur le dessin des masques de ce VCRO, présenté Figure 92, les 5 plots d'entrée (plots rectangulaires oranges) sont répartis respectivement, 3 à droite et 2 à gauche. Les 2 derniers plots d'entrée ne sont pas représentés sur cette figure. A droite on retrouve, de haut en bas les plots correspondant aux tensions de commande (VGP et VGN) de la fréquence du VCRO, ainsi que le plot permettant le contrôle du rapport cyclique des buffers de sortie (Vbg_{BUF}). Les plots de gauche correspondent aux polarisations des caissons d'isolement (Vdd_{TW}) et du substrat (Gnd_{SUB}). Les 2 alimentations restantes non représentées, arrivent à la verticale des transistors, permettant d'alimenter le VCO et les buffers de sortie (Vdd_{VCO} et Gnd_{VCO}). Les 15 sorties (Out<14 :0>) sont dessinées en haut du layout après les buffers inverseurs et permettent de générer les 30 phases de synchronisation du diviseur fractionnaire. Les transistors Nmos et Pmos du VCO sont alimentés (Vdd, Gnd) par le dessus pour diminuer la résistance série : chaque transistor est alimenté par ses propres vias. Ainsi la faible résistance de l'alimentation permet que le bruit d'une cellule délai ne soit pas transmis à une autre par l'alimentation.

Comme dans le cas du VCRO n°1, les lignes entre les cellules délais (Figure 82) ont été implémentées de manière à avoir une longueur strictement identique et une capacité parasite extrêmement faible (1fF). Cette valeur est plus faible que celle du premier VCO car ces lignes ont été gravées en métal 7 (entre cellules délai et buffers). Les capacités Miller des cellules délai ont été optimisées en éloignant l'aller et le retour des lignes $C_{inout_{VCO}}=0,67fF$. La capacité Miller du buffer de sortie $C_{out_{BUF}}$ vaut 0,5fF.

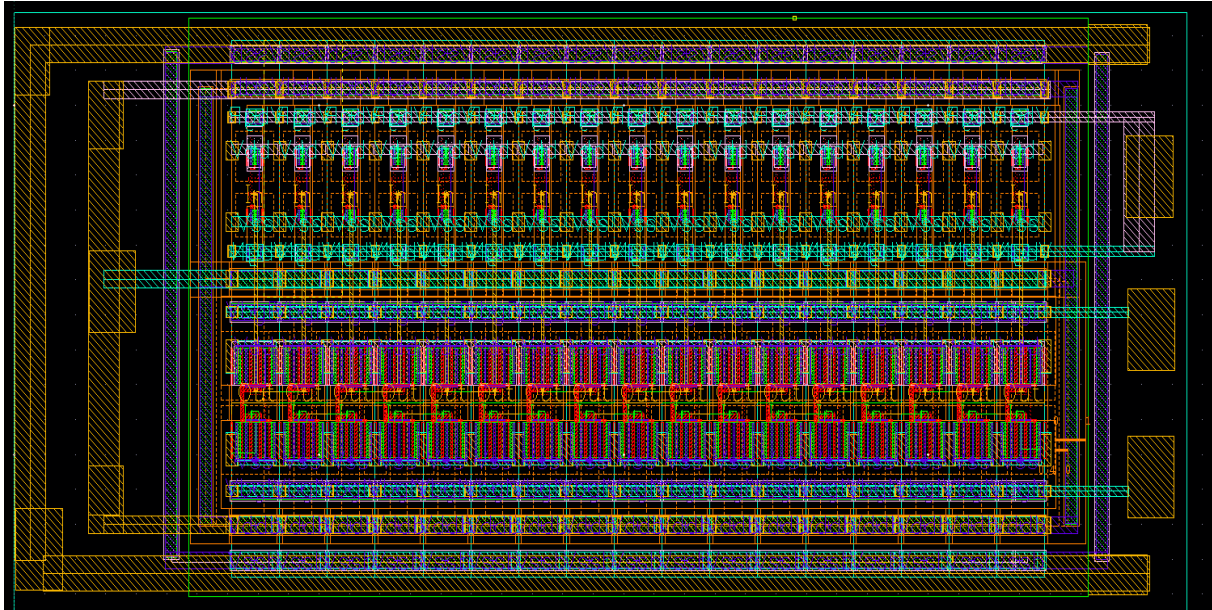


Figure 92 - Layout du VCO à 15 cellules délai (moitié du bas) et ses buffers (moitié du haut)

3.2.3. Simulation

Les simulations ont été menées en suivant le même protocole que celui utilisé pour le premier VCRO. A la tension de grille arrière nominale, $V_{GN}=0,5V$, nous observons une fréquence nominale de 2451MHz avec un gain de 550MHz/V et une consommation nominale de 851 μA avec une variation de 205 $\mu A/V$. On retrouve ces valeurs sur la Figure 93 qui représente les variations de la fréquence et de consommation de courant du VCRO en fonction de la tension de commande sous une tension d'alimentation de 1V. De manière identique, la Figure 94 présente ces mêmes variations en fonction de la tension d'alimentation, 10% autour de sa valeur nominale de 1V à la tension nominale de commande $V_{GN}=0,5V$. Les résultats présentent une fréquence nominale de 2438MHz avec un gain de 5,5GHz/V et une consommation de 869 μA avec une variation de 3,02mA/V.

Les simulations Monte Carlo, non représentées dans ce mémoire, donnent un écart type de la fréquence de sortie de 42MHz, et une variation du rapport cyclique de 0,05%. L'écart type de délai entre phases est de 245fs à la descente et 340fs à la montée. Ces valeurs sont assez encourageantes (car faibles) pour que les phases soient utilisées à des fins de division fractionnaire, mais elles nécessiteront toutefois une calibration pour atteindre les valeurs nécessaires pour faire « disparaître » les spurious (<150fs).

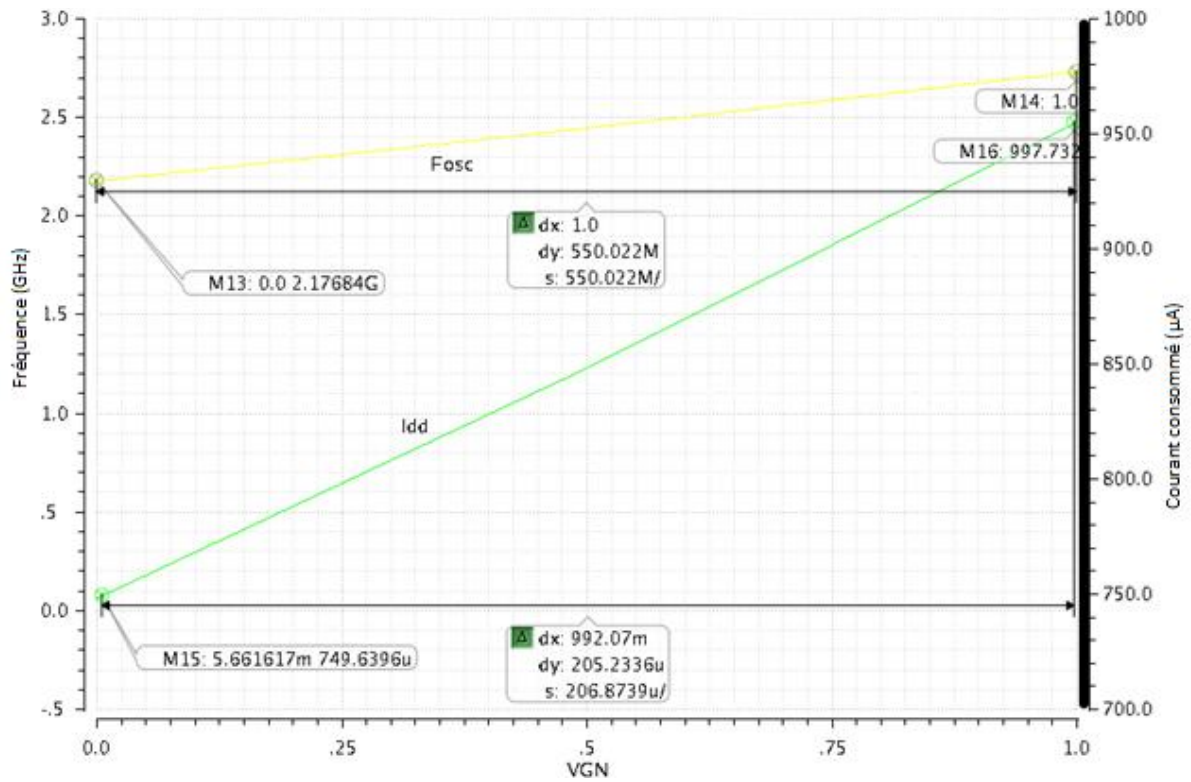


Figure 93 - Consommation et fréquence du VCO 30 phases en fonction de la tension des grilles arrières

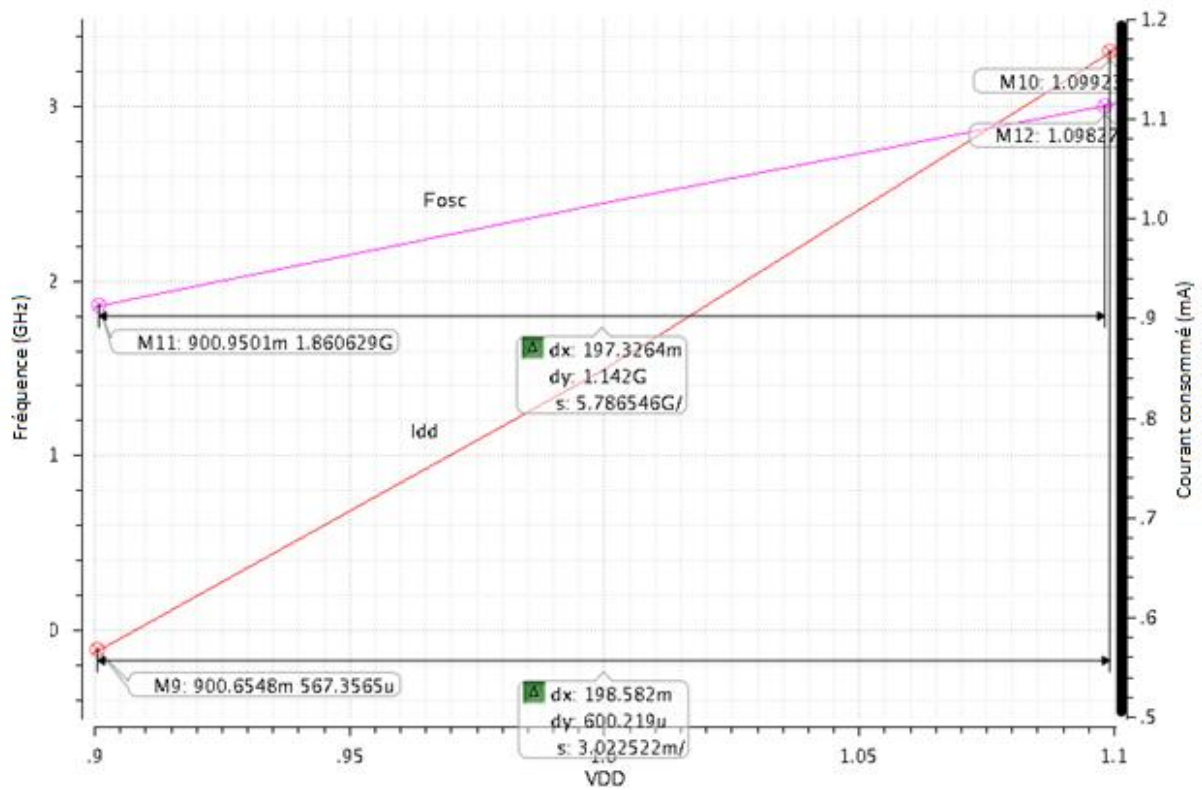
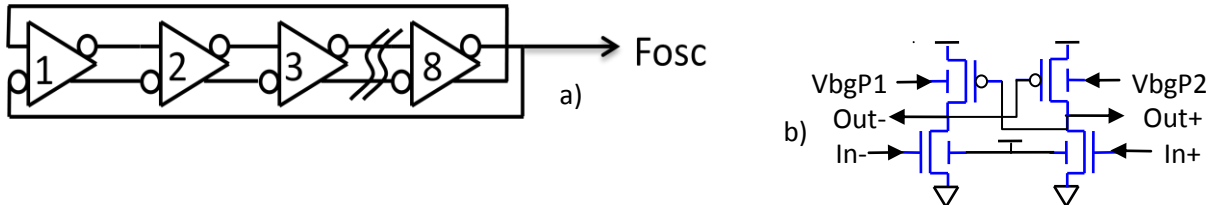


Figure 94 - Consommation et fréquence du VCO 30 phases en fonction de la tension d'alimentation

3.3. VCRO différentiel à 16 phases (8 cellules)

3.3.1. Implémentation

La illustre cet oscillateur différentiel où le schéma d'une cellule élémentaire est représenté sur la Figure 95.b.



Le schéma de ce VCRO sous cadence est représenté Figure 97. Il est composé de 8 cellules délai différentielles de type DCVSL détaillées en Figure 96 (RO<0:7>), et de 16 buffers de sortie de phase single-ended identiques aux VCRO précédents (Buf<0:15>). Chaque cellule est composée de seulement 4 transistors MOS LVT (2N et 2P) et ne possède pas de source de courant commune, qui génère du bruit de phase et augmente la consommation statique. Le rapport cyclique n'est pas optimal (40%) mais le bruit de phase est meilleur qu'avec les cellules de type DCVSLR (résistance en série entre Nmos et Pmos) ou ULPD-DCVSL (diode en série). Les largeurs des transistors Low-Vth sont $W=1,8\mu\text{m}$ et leurs longueurs sont de $L=50\text{nm}$ pour les Nmos et $L=45\text{nm}$ pour les Pmos.

Les différents condensateurs disposés sur le schéma du VCO (voir Figure 97) représentent les capacités parasites extraites après optimisation du layout (cf. Figure 98). Ces valeurs extraites sont respectivement $C_{\text{out}_{\text{VCO}}}=2,6\text{fF}$ et $C_{\text{out}_{\text{BUF}}}=0,6\text{fF}$.

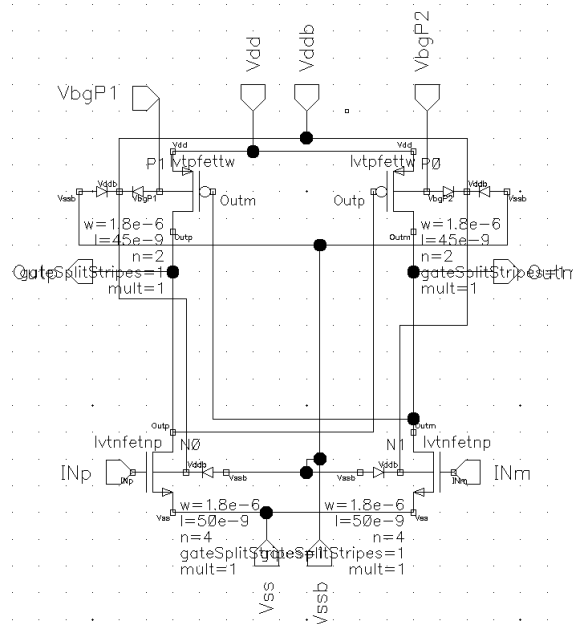


Figure 96 - Cellule délai du VCO différentiel à 16 phases

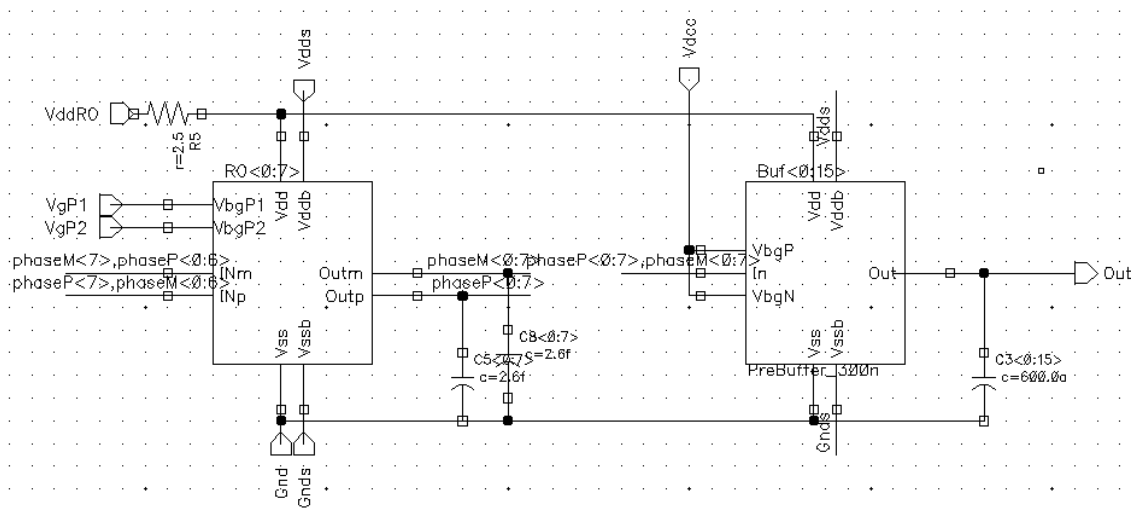


Figure 97 - Schéma VCO à 8 cellules différentiel avec buffers

3.3.2. Layout

Le layout de ce VCRO, illustré par la Figure 98, montre que les 16 buffers de sortie occupent environ 60% de la surface totale du circuit. La partie du VCRO proprement dite (soit 40% de la surface) est représentée par la couleur vert-pomme. La taille importante allouée à ces buffers, visibles en haut et en bas du layout, provient principalement des lignes d'interconnexion (routing).

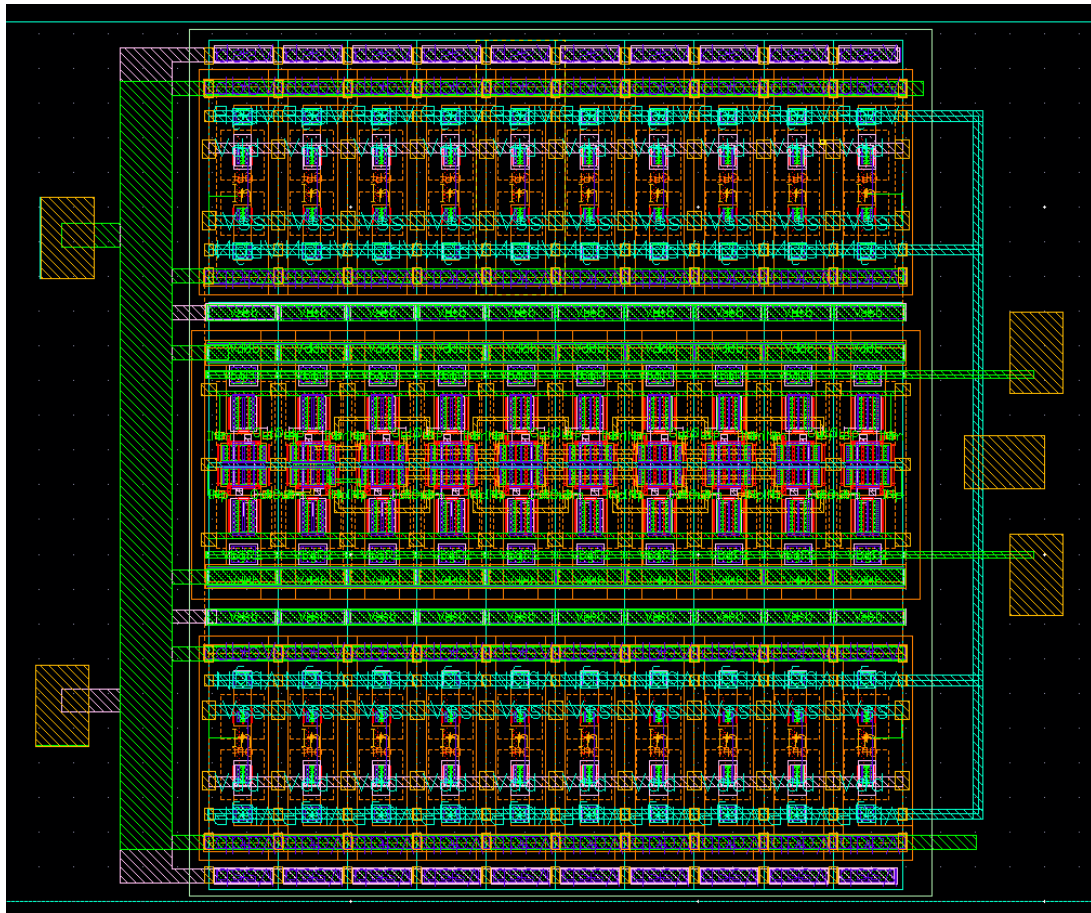


Figure 98 - Layout du VCRO différentiel à 8 cellules (milieu) et leurs 16 buffers (haut et bas)

Les plots à gauche représentent les polarisations du substrat ($V_{dd_{TW}}$ et Gnd_{SUB}) et ceux de droite sont les commandes des grilles arrières (V_{bgP1} du transistor Pmos1 en haut, V_{bgP2} du transistor Pmos2 en bas et $V_{bg_{BUF}}$ des transistors Nmos/Pmos du Buffer de sortie au milieu). Les alimentations des cellules délai et des buffers, étant réalisées à la verticale de chaque transistor, ont été rendues transparentes pour permettre de voir les zones actives.

3.3.3. Simulation

Sur la Figure 99, on peut observer la variation de la fréquence du VCRO et du courant consommé en fonction de la tension des 2 grilles arrières Pmos (V_{bgP1} et V_{bgP2}) pour une tension d'alimentation de 1V. On constate que, à 0,5V de tension de la grille arrière, la fréquence nominale est de 2423MHz avec un gain de -252MHz/V et une consommation nominale de 967 μ A avec une variation de -138 μ A/V. La Figure 100 illustre ces évolutions de fréquence et de courant en fonction de la tension d'alimentation (+/-10%). La fréquence nominale est de 2435MHz avec un gain de 4,9GHz/V et la consommation nominale est de 960 μ A avec une variation de 2,85mA/V.

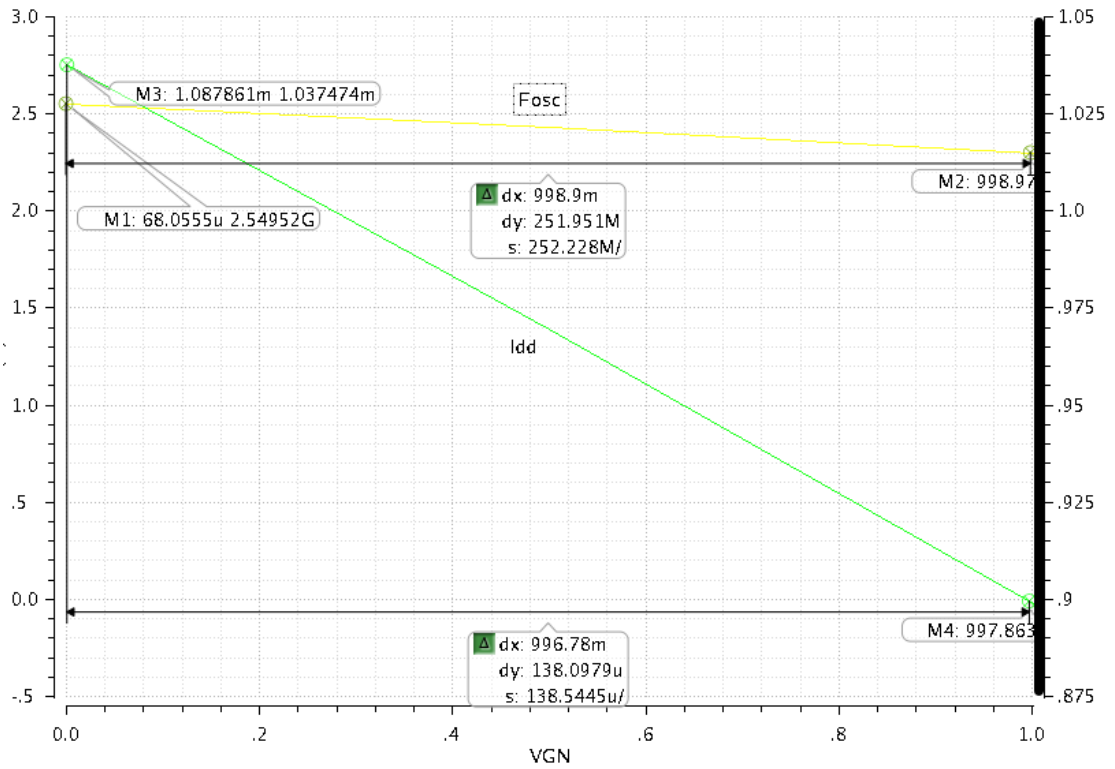


Figure 99 - Consommation et fréquence du VCO différentiel 16 phases en fonction des grilles arrières

Les simulations Monte Carlo donnent un écart type de la fréquence de sortie de 30,8MHz et l'écart type des temps de propagation entre phases est de 127fs à la montée et 250fs à la descente. Ces valeurs restent faibles vis à vis de la production de spurios.

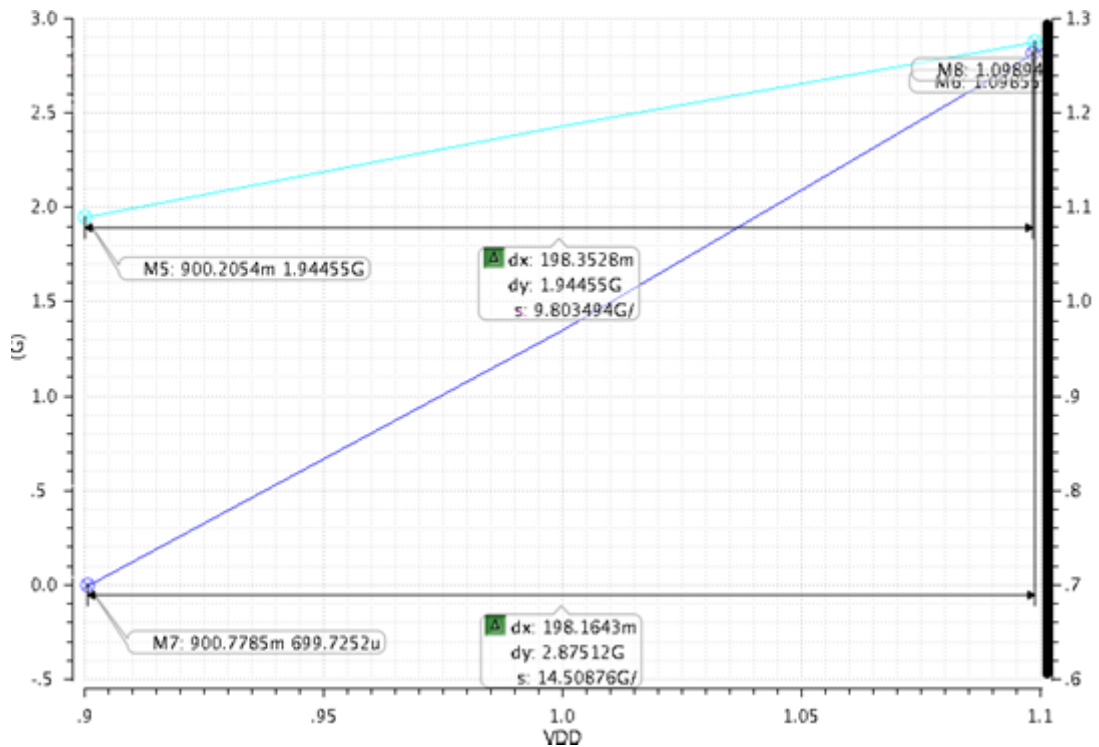


Figure 100 - Fréquence (cyan) et consommation (bleu) du VCRO n°4 en fonction de la tension d'alimentation

3.4. VCRO différentiel à 16 phases à 4x Fout

3.4.1. Implémentation

La Figure 101 illustre cet oscillateur à 2 cellules différentielles ainsi que le détail du diviseur.

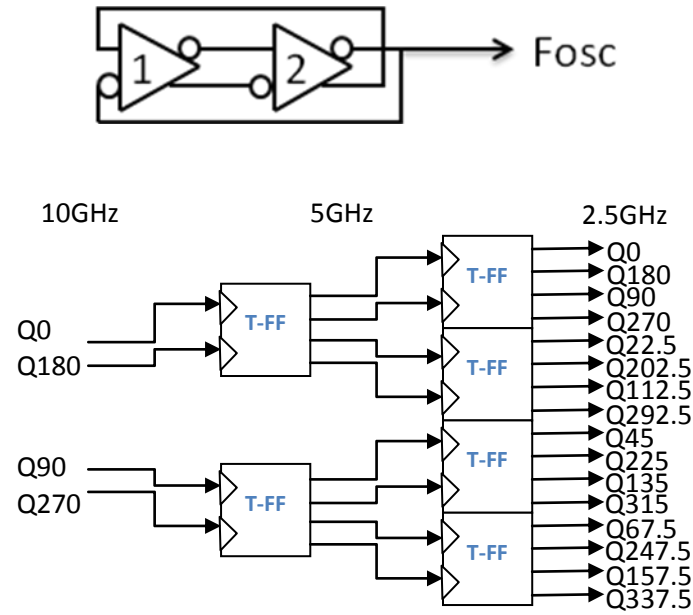


Figure 101 - Schéma bloc du VCO et schéma bloc du diviseur

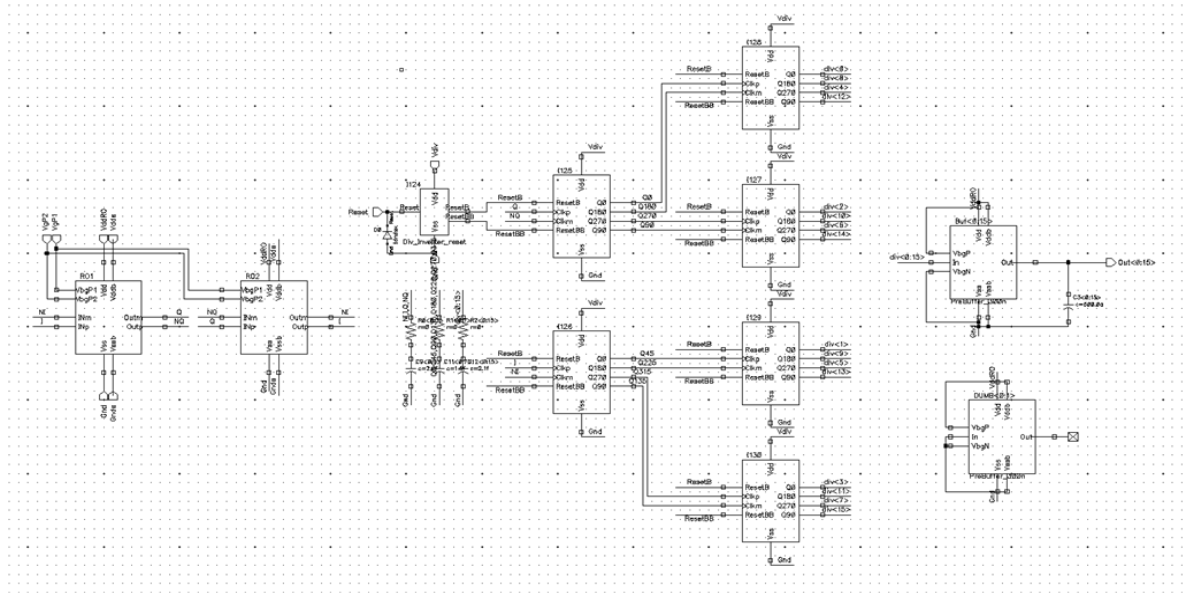


Figure 102 - Schéma du VCO différentiel à 4*Fout

Le VCRO (cf. Figure 102) est composé de 2 cellules délai différentielles (RO<0:1>) illustrées en Figure 103 de type DCVSL (Differential Cascode Voltage-Switch Logic), d'un diviseur à bascule, FlipFlop T, différentielle développée spécifiquement pour cette application et de 16 buffers de sortie

de phase single-ended (Buf<0:15>) identiques aux VCROs précédents. Les cellules délai utilisées dans ce VCRO (cf. Figure 103) sont identiques au VCRO précédent avec des longueurs de transistors Nmos et Pmos légèrement plus grandes (L=55nm pour les Nmos et L=50nm pour les Pmos) car le diviseur à base de flip-flops représente une charge plus faible que les buffers de sortie. De plus seulement 2 cellules délai au lieu de 8 sont utilisées pour obtenir une fréquence 4 fois plus élevées, soit 10GHz.

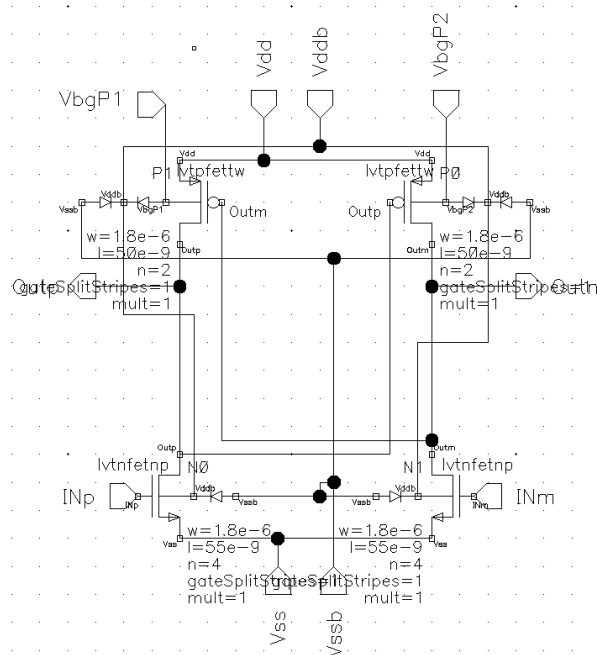


Figure 103 - Schéma d'une cellule délai du VCO différentiel à 4*Fout

Les 4 phases de l'oscillateur sont fournies à 2 Flip-Flop T différentielles (Figure 104) qui divisent la fréquence par 2 et multiplient le nombre de phases par 2, donnant au total 8 phases à une fréquence de 5GHz. Ces 8 phases sont ensuite fournies à 4 Flip Flop T différentielles pour générer 16 phases au total à 2,5GHz. Ces 16 phases passent par un buffer de même type que pour les VCRO précédents pour pouvoir être utilisées ensuite par le diviseur fractionnaire.

Les flip-flop T différentielles sont en fait une modification des flip-flop conventionnelles (Figure 105) afin d'augmenter leur fréquence de fonctionnement et diminuer leur consommation statique et dynamique. La consommation statique est diminuée par rapport à un diviseur CML dont la consommation statique est non nulle et la consommation dynamique est diminuée par rapport à un diviseur à bases de cellules standard car le fait qu'elles soient différentielles supprime la nécessité d'inverser les signaux de commande des horloges et d'inverser le signal de sortie pour le reboucler. La flip-flop T est une mise en cascade de 2 bascules D et d'un inverseur pour reboucler la sortie sur l'entrée.

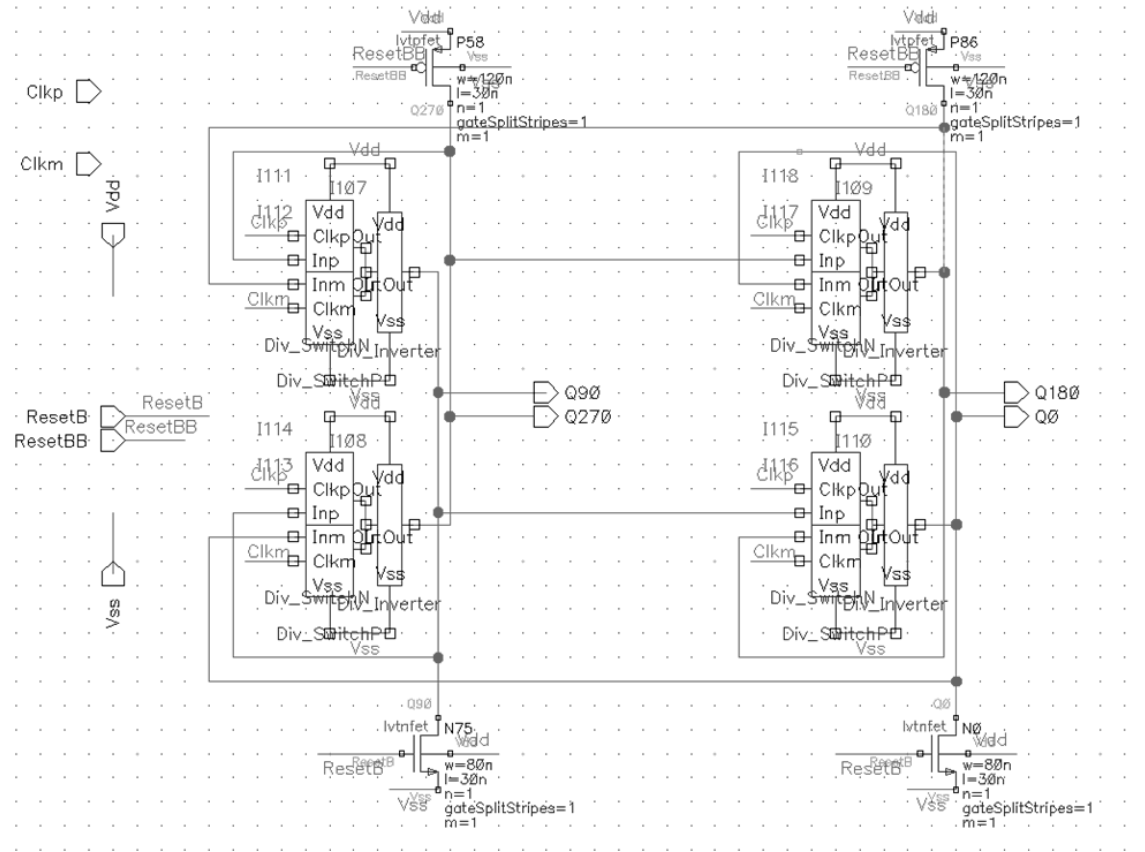


Figure 104 - Schéma d'une flip flop T différentielle à base de 2 latches imbriqués

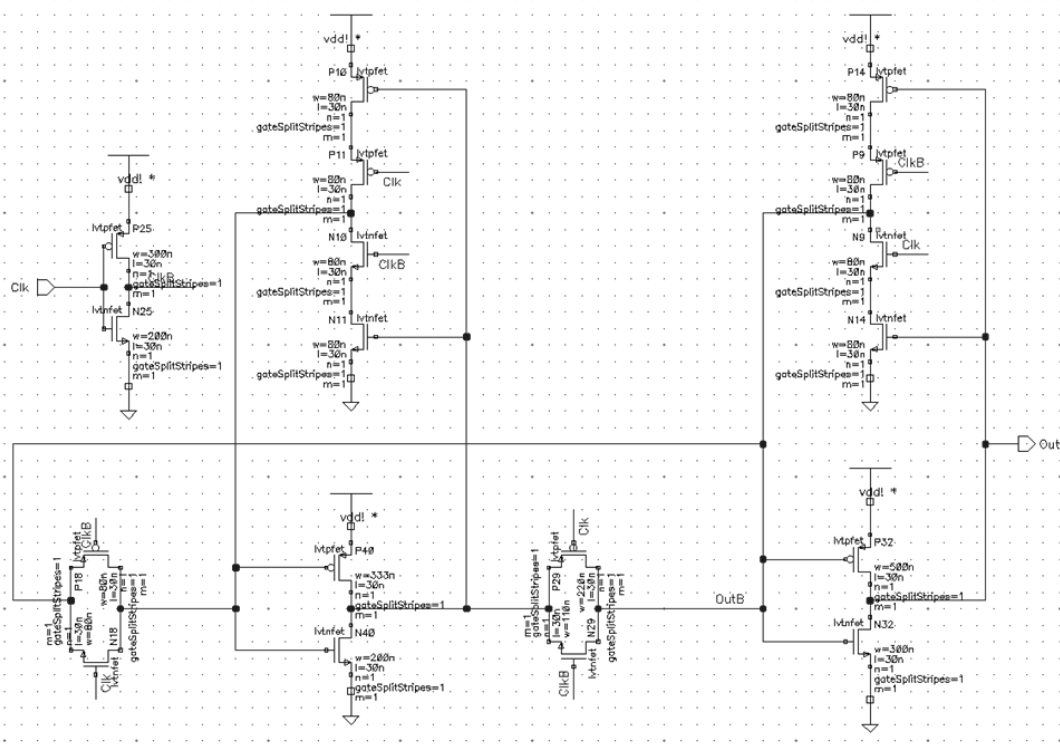


Figure 105 - Schéma équivalent d'une flip flop T non-différentielle

3.4.2. Layout

Le layout de ce VCRO (cf. Figure 106) permet de constater que la surface occupée par les cellules délai (en haut) est très faible par rapport au total. On peut distinguer facilement les 2 étages de division par 2 en dessous des cellules délai avec 2 puis 4 bascules T différentielles côte à côte. Plus bas, on aperçoit les lignes de transmission des phases, de longueurs strictement identiques pour ne pas déphaser les signaux, entre le deuxième étage du diviseur et les 16 buffers de sortie. Finalement, au bas de la figure, on distingue les 16 buffers de sortie des phases.

En regardant le diviseur en détail (cf. Figure 107), on distingue une Flip Flop T et le circuit de reset (initialisation) à sa gauche. On peut apercevoir les Pmos en haut et les Nmos en bas de la Flip Flop T sur cette figure 107. Les transistors N et P des passgates et des inverseurs sont éclatés pour augmenter la densité. Les deux zones tout en bas de la figure sont des diodes en inverse qui protègent les transistors des effets d'antenne lors de la fabrication.

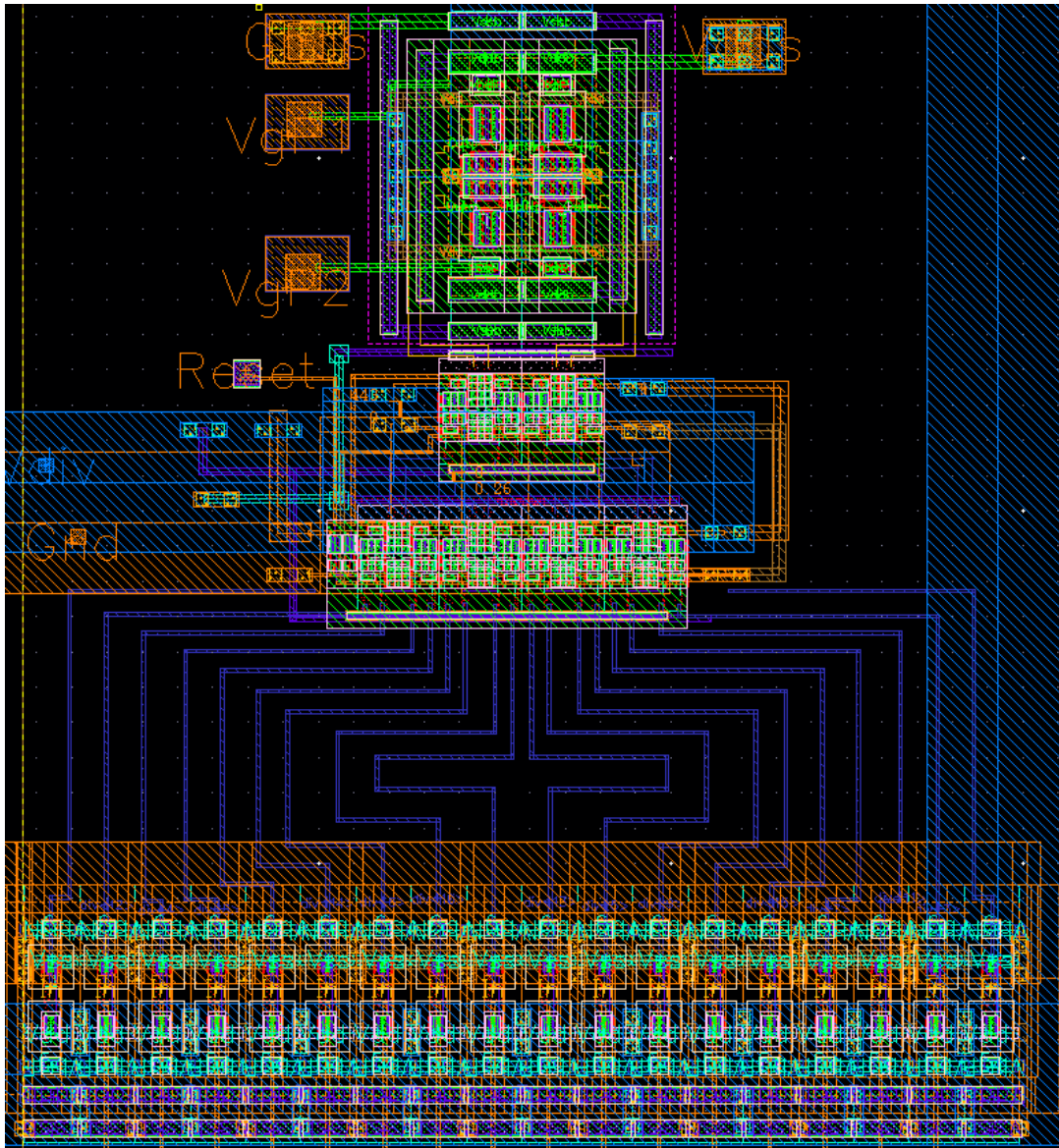


Figure 106 - Layout du VCO à $4 \cdot F_{out}$ avec son diviseur de fréquence par 4, les lignes de propagation et les buffers

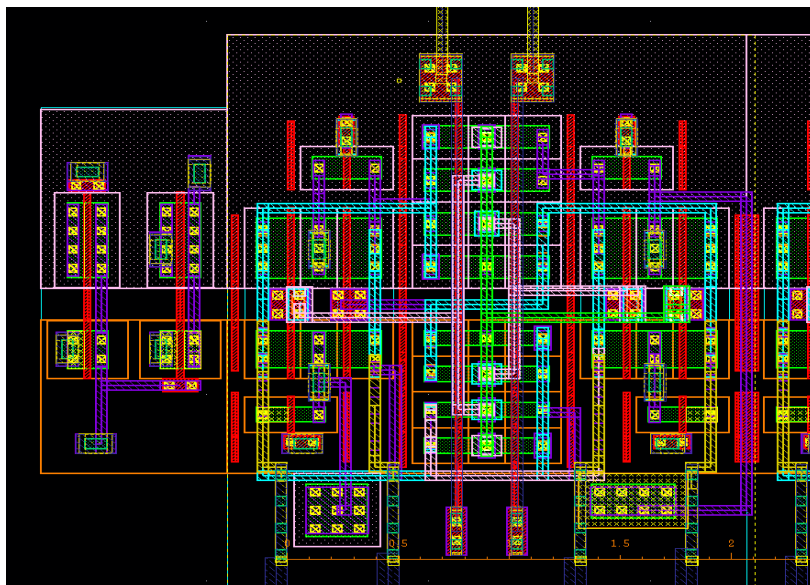


Figure 107 - Détail d'une FlipFlop

3.4.3. Simulation

Le protocole de simulation est identique aux précédents. Les Figure 108 et 109 résument l'ensemble des résultats de simulation. Ils montrent que pour une tension de grilles arrières (VbgP1 et VgpP2) de 0,5V, la fréquence nominale, sous une tension d'alimentation de 1V, est de 9913MHz avec un gain de -979MHz/V. La sortie du diviseur par 4 donne alors la fréquence désirée, à savoir 2478MHz avec un gain de -245MHz/V. La consommation nominale du VCO et des buffers de sortie est de 821 μ A avec une variation de -126 μ A/V et la consommation nominale du diviseur est de 306 μ A avec un gain de -28,5 μ A/V. En faisant varier la tension d'alimentation, on constate que la fréquence nominale du VCO est de 9,853GHz pour un gain de 18,83GHz/V. La sortie divisée par 4 est alors de 2463MHz avec un gain de 4,7GHz/V. La consommation nominale du VCO et des buffers de sortie est de 781 μ A avec une variation de 2,55mA/V et la consommation nominale du diviseur est de 321 μ A avec un gain de 922 μ A/V.

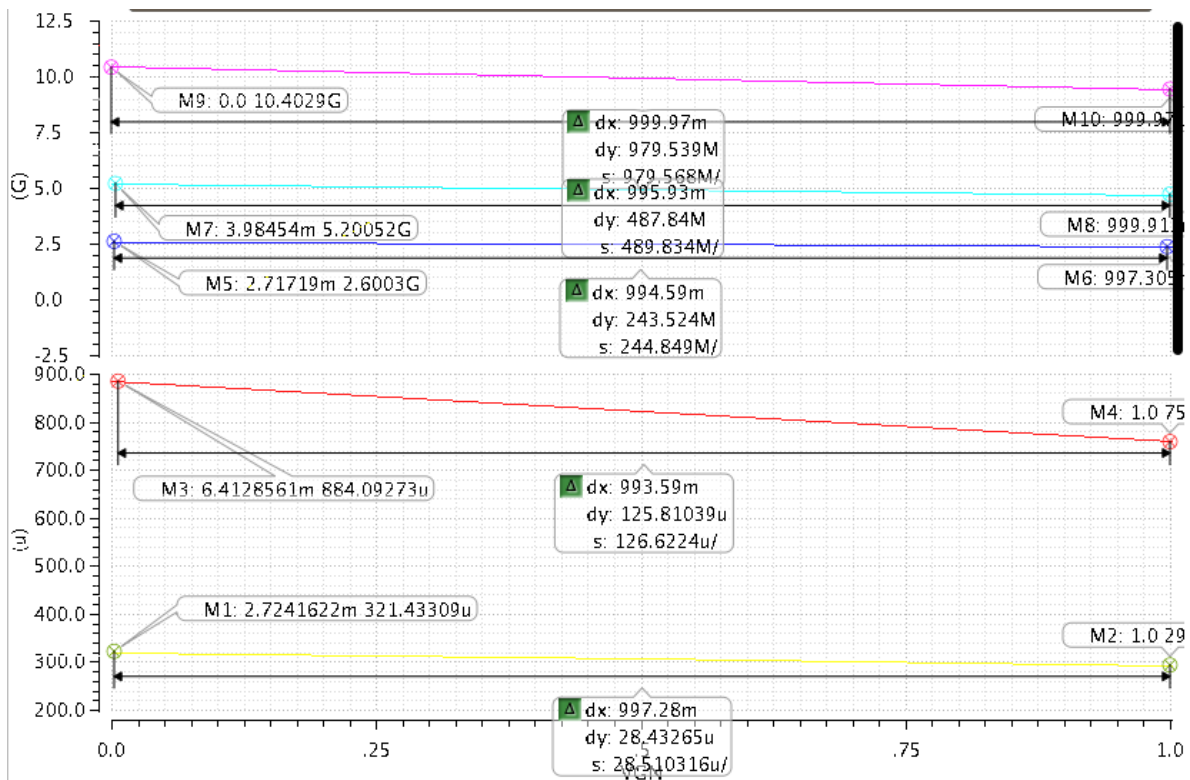


Figure 108 - Consommation et fréquence du VCO différentiel à 4*Fout en fonction des grilles arrières

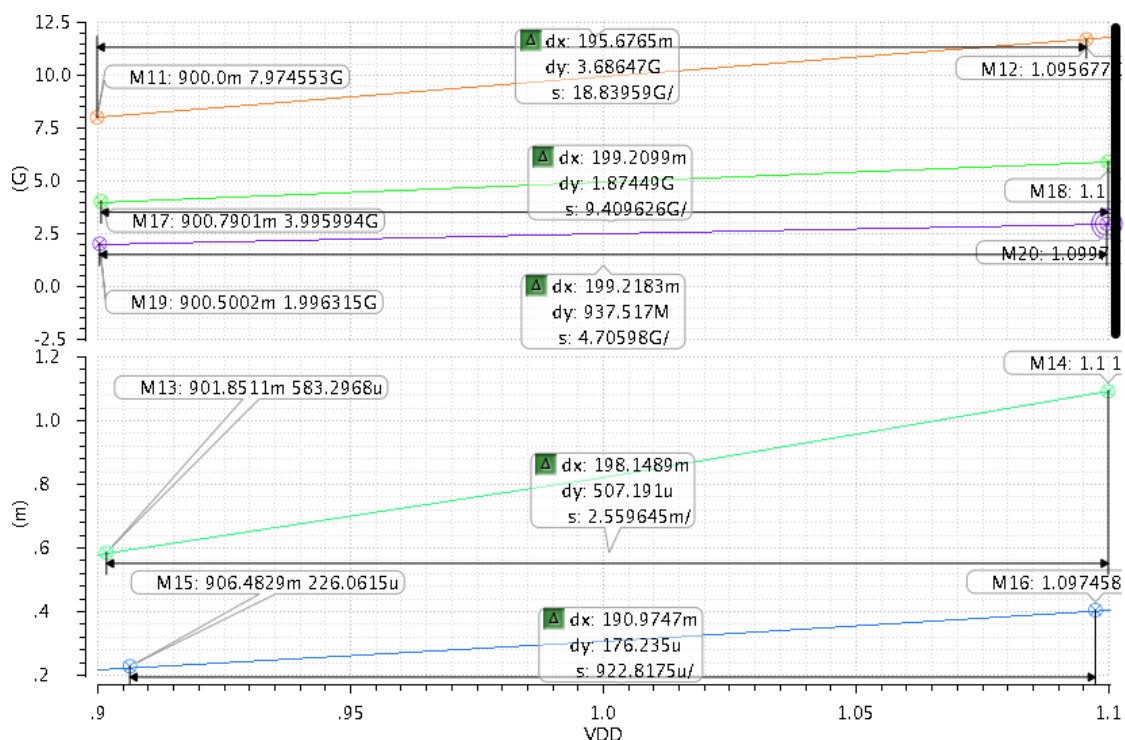


Figure 109 - Consommation et fréquence du VCO différentiel à 4*Fout en fonction de la tension d'alimentation

Les analyses Monte Carlo montrent un écart type de la fréquence de sortie de 40,4MHz à la sortie du diviseur (à 2,5GHz), et l'écart type des temps de propagation entre phases est de 581fs à la montée et 570fs à la descente.

On constate que ces valeurs sont plus élevées que le cas d'un oscillateur à la fréquence nominale (oscillateur à 8 cellules différentielles). La moitié de cette valeur de mismatch est due au couple VCRO-diviseur, l'autre moitié est due aux buffers de sortie. Ces derniers ont été conçus avec une taille faible pour pouvoir être chargé par le diviseur à flip-flop (impédance de sortie non négligeable). En plus d'une variabilité accrue par rapport au VCRO précédent, on doit ajouter la consommation du diviseur qui est non-négligeable (plus de 300µA).

Le diviseur utilisé à bases de flip flop permet d'aller jusqu'à plus de 20GHz en simulation et est très efficace énergétiquement par rapport à des diviseurs CML, mais pas autant qu'un oscillateur sans diviseur à plus basse fréquence. Néanmoins, si ce diviseur est utilisé avec un oscillateur à très haute fréquence à base de résonateur LC (à 10GHz ou 20GHz), il est possible de recréer des phases qui permettront de faire fonctionner une PLL fractionnaire, solution plus flexible et moins couteuse énergétiquement qu'avec un diviseur variable (N/N+1) avec une topologie CML à la fréquence de 20GHz.

4. Réalisation

4.1. Dessin des masques

Nous avons intégré sur la même puce, dont le layout est représenté par la Figure 110, les 4 VCRO décrits précédemment et leurs buffers 50Ω (qui seront détaillés au chapitre IV) ainsi que des capacités, en jaune, de filtrage des tensions de commande et de découplage des alimentations.

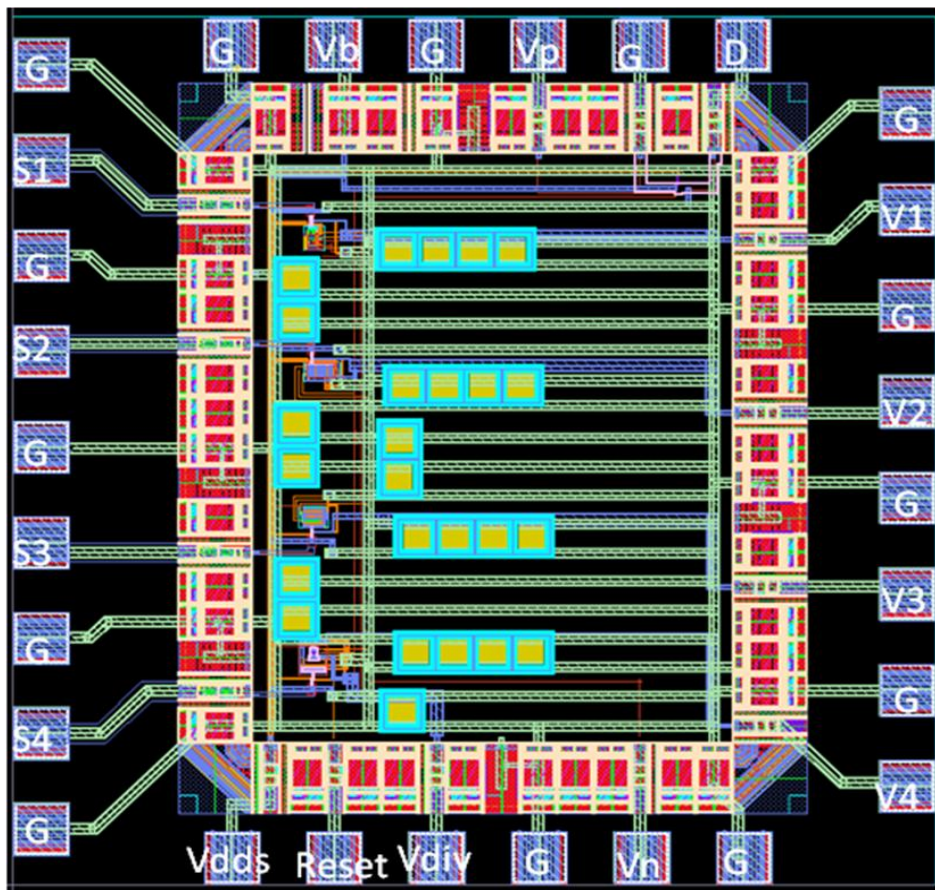


Figure 110 - Layout de la puce complète

Les plots (en bleu) de droite (V1-4) correspondent aux alimentations de chaque VCRO. Les plots de gauche (S1-4) sont ceux des sorties à haute fréquence de chaque VCRO. Les plots en haut et en bas de la Figure 110 servent à polariser les grilles arrières (Vb, Vn), le substrat (Vdds) et alimenter le diviseur (Vdiv) et la commande de Reset, ainsi que les buffer 50Ω (Vb).

4.2. Simulations post layout

La simulation transitoire post-layout de la puce de test, illustrée sur la Figure 111, présente le signal de sortie des 4 VCRO. On peut vérifier qu'ils se comportent de manière identique et

fournissent la même fréquence, à savoir 2,45GHz. La Figure 112 présente le bruit de phase post-layout du premier VCRO (single-ended à 3 inverseurs). Les simulations post-layout en fréquence et en bruit étant satisfaisantes, le circuit dessiné a été validé pour fabrication.

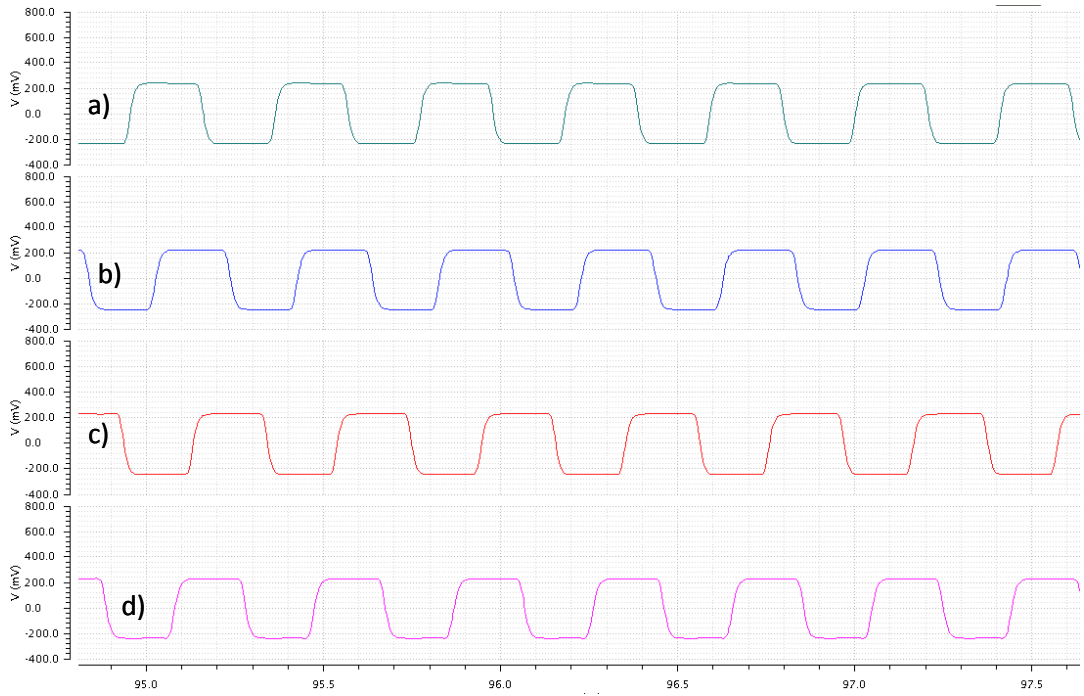


Figure 111 - Simulations Post-Layout des 4 VCOs après le buffer 50Ω : a) Inverseur single 3 cellules, b) Inverseur single 15 cellules, c) Inverseur différentiel 8 cellules, d) Inverseur différentiel 2 cellules divisé par 4

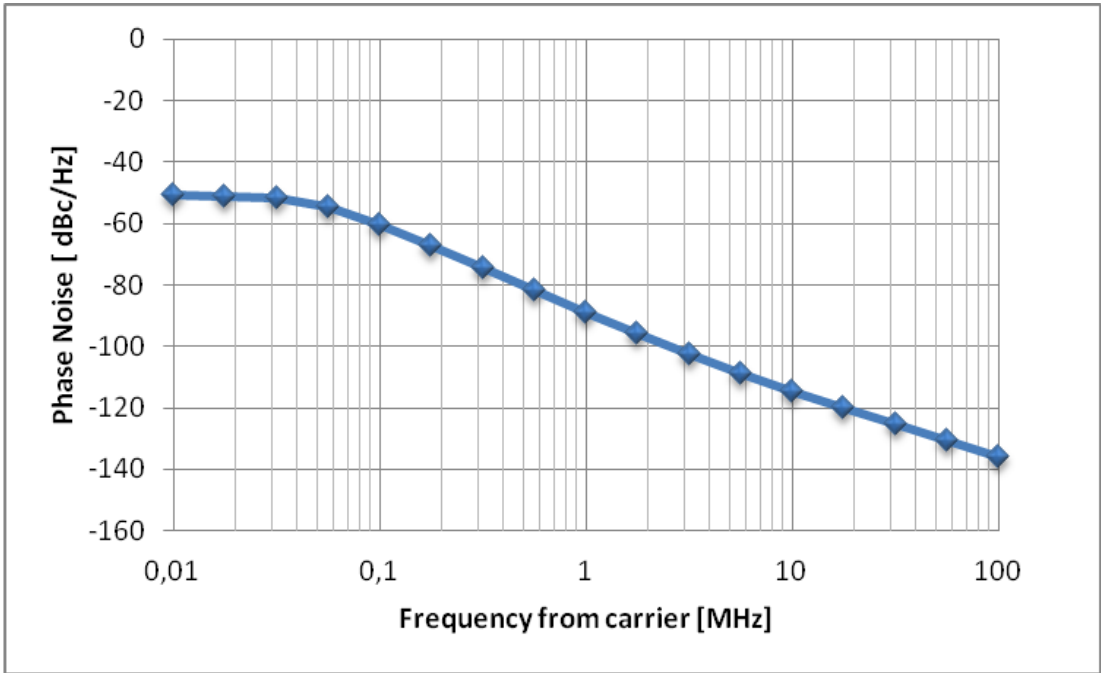


Figure 112 - Bruit de phase après layout du VCRO n°1 (à trois inverseurs)

4.3. Fabrication

Le circuit a été réalisé par la société STM, site de Crolles, via le service du CMP (Circuits Multi-Projets). Les photographies (Figure 113 et Figure 114) des puces reçues montrent les motifs de remplissages ajoutés par le CMP pour respecter les règles de layout concernant la densité (carrés d'aluminium ou de cuivre).

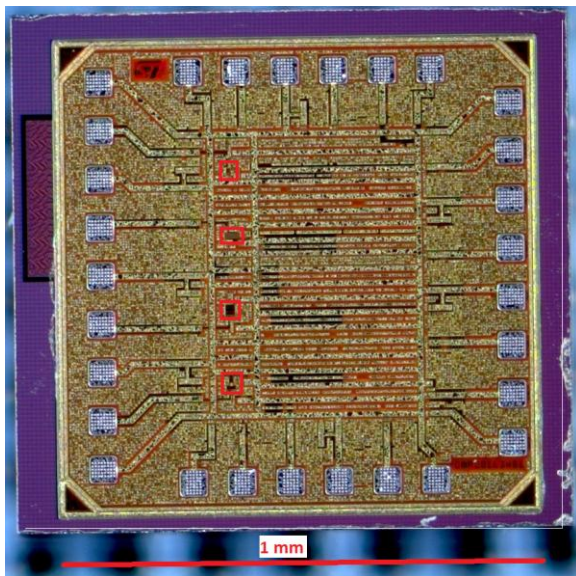


Figure 113 - Photographie du circuit réalisé avec l'emplacement des 4 VCRO

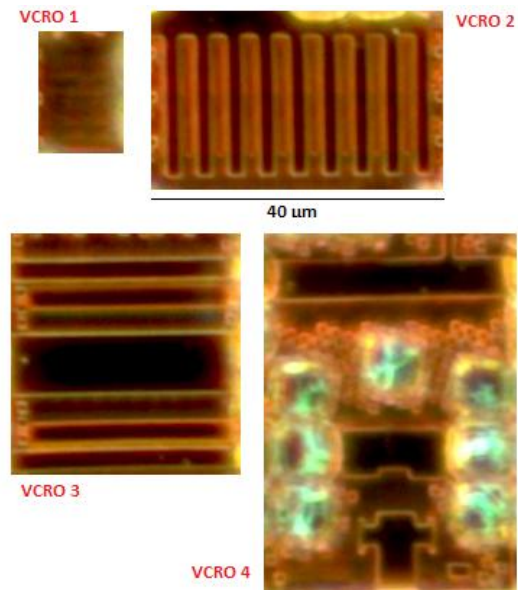


Figure 114 - Photographies zoomées des 4 VCRO

4.4. Récapitulatif et Conclusion

Dans ce paragraphe nous résumons sous la forme de 3 tableaux les caractéristiques simulées des 4 VCRO présentés précédemment.

Dans le premier tableau (Tableau 9) est résumé pour chaque VCO le nombre de cellules, les tailles des transistors Pmos et Nmos, la surface active (somme des $W \cdot L$), la surface silicium, la consommation et la fréquence nominale. Dans le cas particulier de ce tableau, nous avons également rappelé les caractéristiques principales du diviseur par 4.

Le second tableau (Tableau 10) rappelle le bruit de phase à 1MHz et 3MHz de la porteuse, et les FOM correspondantes, ainsi que le gain par rapport à la tension d'alimentation (Pushing) et le gain par rapport à la tension de grille arrière (Tuning).

Le troisième tableau (Tableau 11) résume d'autres paramètres relevés lors des simulations comme le « ripple » du courant d'alimentation (donnant la rapidité de la commutation dans l'oscillateur en anneau), le « ripple » du courant d'alimentation du buffer 50Ω (donnant la rapidité de

la commutation en sortie), les écarts types (Sd) de rapport cyclique (DC) et des temps de transition des phases dans le VCRO ainsi que la variabilité de la fréquence du VCRO avant et après layout.

Tableau 9 - Caractéristiques des VCRO et diviseur par 4

VCO N°	Cellules	Wp*Lp μm + Wn*Ln [μm]	Surface Active [μm ²]	Surface [μm ²]	Idd [μA] (Vdd = 1V)	Fréquence [MHz]
1	3 inv.	14*0,24 / 14*0,24	20,16	375	860μ	2460
2	15 inv.	4,5*0,08 / 4,5*0,08	10,8	450	851μ	2440
3	8 diff.	1,8*0,45 / 1,8*0,50	27,36	625	967μ	2500
4	2 diff.	1,8*0,5 / 1,8*0,55	7,56	1140	821μ	10120
Diviseur par 4	6 flip-flop diff. + Reset	320n/120n*30n 200n/80n*30n	0,259	Inclus dans VCRO 4	306μ	2530

Tableau 10 - Pushing, Tuning, bruit de phase, gain et figure de mérite des VCRO

VCO N°	PN @1MHz [dBc/Hz]	PN @3MHz [dBc/Hz]	FOM @1MHz [dB]	FOM @3MHz [dB]	Pushing [GHz/V]	Tuning [MHz/V]
1	-88,9	-101,7	-157,7	-160,9	5,35	556
2	-87,3	-100,7	-156,4	-160,2	5,5	550
3	-82,4	-94,7	-150,8	-153,6	4,9	-252
4 (/4)	-79,3	-91	-148,3	-150,5	4,7	-245

Tableau 11 - Divers autres caractéristiques des VCRO

VCO N°	Idd Ripple VCRO [mA]	Idd Ripple buffer [mA]	Sd OutDC [%]	Sd Phase descente [fs]	Sd Phase montée [fs]	Variabilité [MHz]	Var. PLS [MHz]
1	0,6	4,1s	0,02%	640fs	745fs	45,4	53
2	0,2	4,2	0,05%	245fs	340fs	42,2	57
3	0,2	3,48	0,38%	127fs	250fs	30,8	48
4 (/4)	0,28	3,45	0,23%	570fs	581fs	40,4	52

On peut constater que le VCRO n°2 est celui qui a la meilleure FOM, si on exclut le VCO n°1 qui n'a pas assez de phases pour être utilisé dans un diviseur fractionnaire (rapport 6 entre fréquence de la référence et pas de fréquence) et on peut constater que c'est aussi celui qui a le plus de phases.

5. Mesures et tests des VCRO

5.1. Introduction

Toutes les mesures ont été effectuées avec un prober manuel et des instruments de laboratoires, dont 2 analyseurs de source (mesure de bruit de phase) : Rohde & Schwarz FSUP Signal Source Analyser et Agilent E5052A Signal Source Analyser. L'espacement des pointes est de 100μm. Les pointes d'alimentation (côté droit de la puce) sont pourvues de découplage et les pointes RF (côté gauche de la puce) ont des plots de masse de chaque côté (GSG).

5.2. Mesure à fréquence nominale

La consommation et la tension de fonctionnement à 2,45GHz des 4 VCRO, du diviseur par 4 et des buffers 50Ω ont été mesurées (cf. Tableau 12) grâce à une alimentation Keithley et un oscilloscope à haute fréquence d'échantillonnage (80GSa/s) (voir Figure 115). L'impédance d'entrée des commandes en fréquence (Vp et Vn) est d'environ 1,25MΩ (0,8μA sous 1V) pour chacune des entrées. Cette fuite est due aux diodes en inverse des grilles arrières des 4 VCRO.

Tableau 12- Mesures de consommation à 2,45GHz

VCRO N°	Tension	Consommation
1	1,05 V	0,75 mA
2	1 V	0,81 mA
3	1,09 V	0,85 mA
4	1,05 V	0,75 mA
Diviseur	1,05 V	0,52 mA
Buffer 50Ω	1 V	4,2 mA

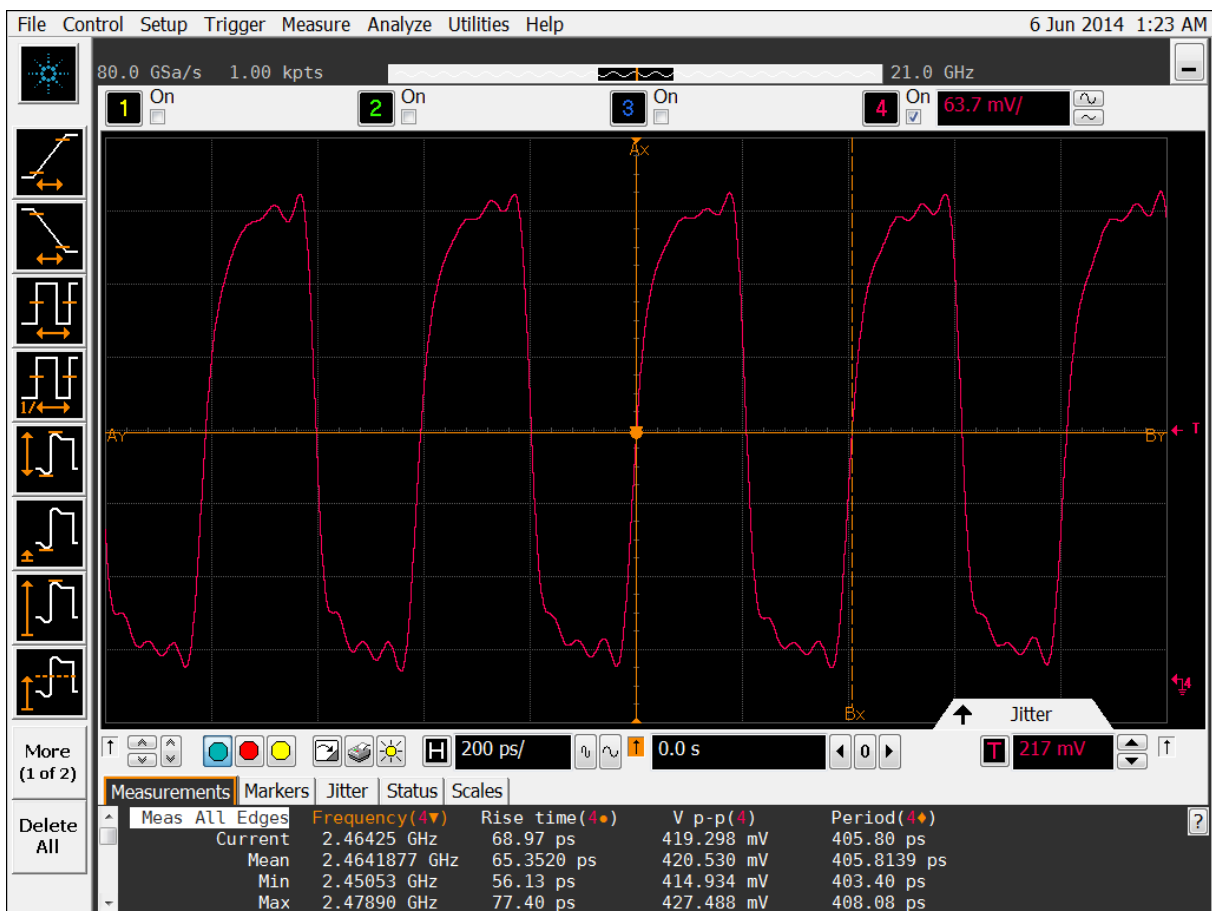


Figure 115 - Mesure à l'oscilloscope du signal de sortie

5.3. Mesures de la fréquence d'oscillation des VCRO

Les mesures de fréquence des VCRO ont été réalisées sans diviseur externe. Les mesures de la fréquence d'oscillation en fonction de la tension de commande sont illustrées par la Figure 116

tandis que celles en fonction de la tension d'alimentation sont illustrées par la Figure 117. On constate que le tuning et le pushing des VCRO single-ended sont plus élevés que ceux des VCRO différentiels.

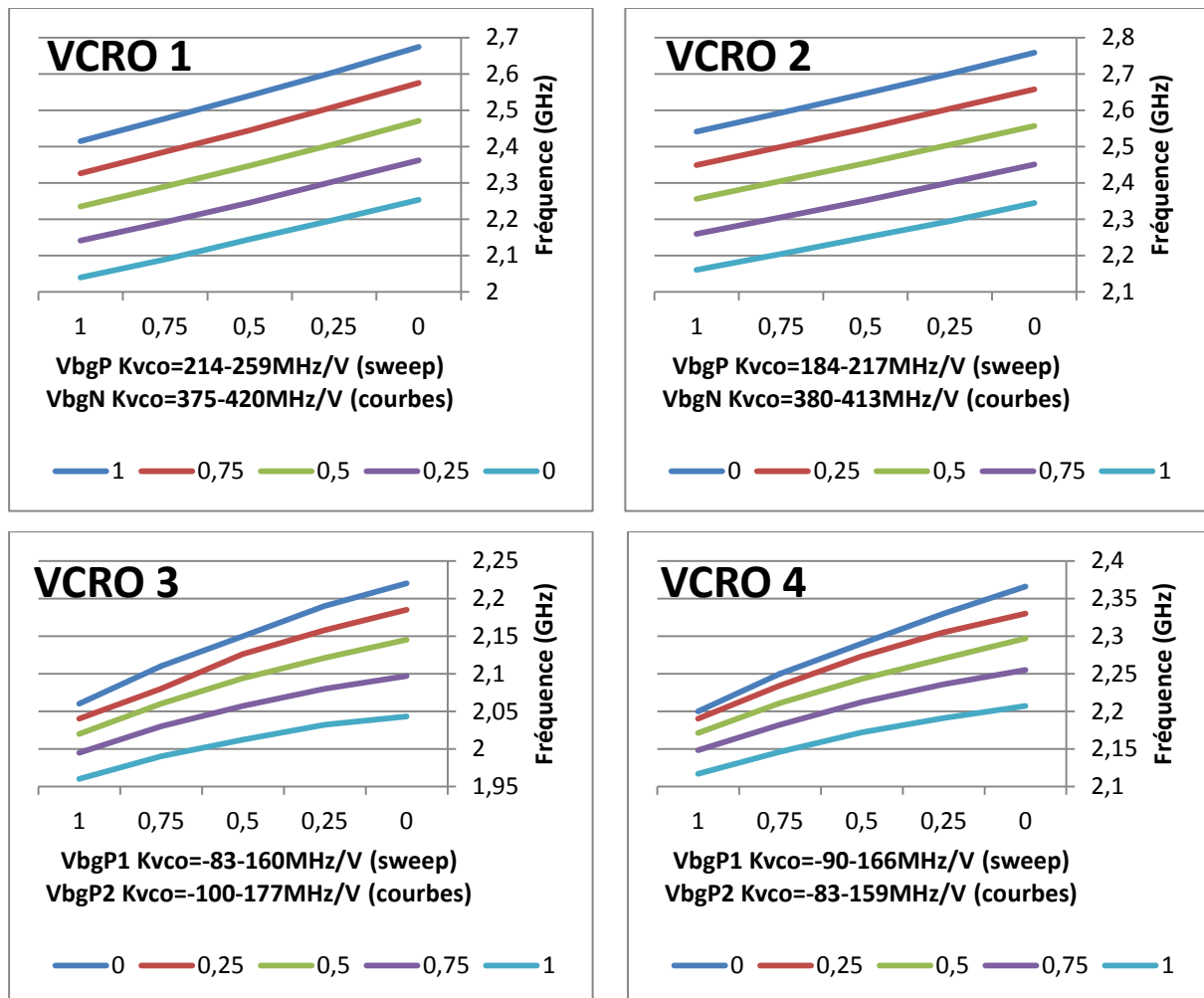


Figure 116 - Mesures du tuning des 4 VCRO

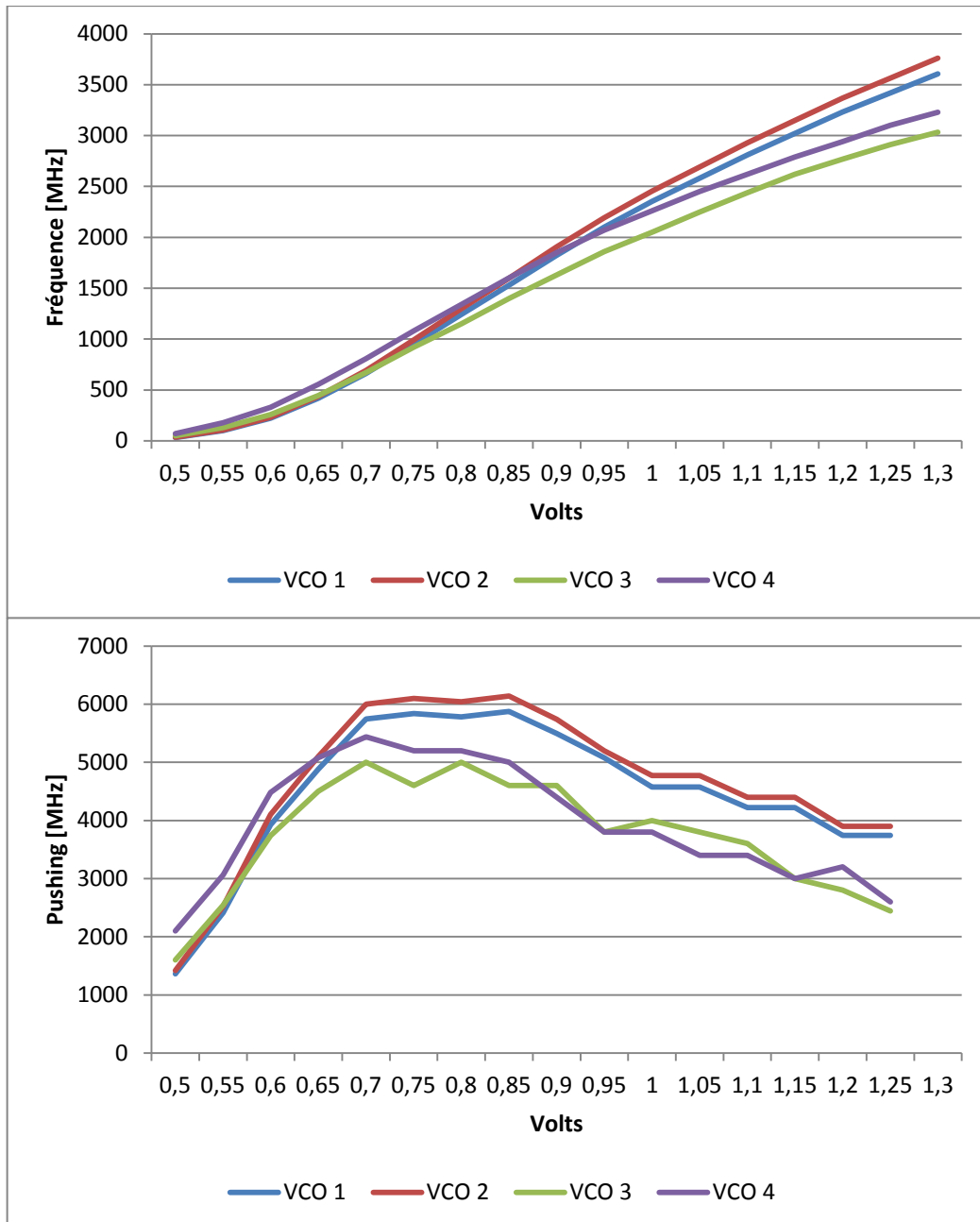


Figure 117 - Mesures de la fréquence et calcul du pushing des 4 VCROs

5.4. Mesure du bruit de phase des VCRO

Les mesures du bruit de phase en boucle ouverte ont été réalisées par le biais des analyseurs de source/synthèse de fréquence. Pour améliorer la mesure du bruit de phase des oscillateurs peu stables en fréquence (bruyants), ces appareils disposent d'une sortie en tension qui permet d'asservir la fréquence de l'oscillateur (comme dans une PLL). La bande passante de cet asservissement est au maximum 30kHz, ce qui n'est pas suffisant dans le cas de nos circuits fabriqués. En raison de cette bande passante, les mesures de bruit de phase, illustrées par la Figure 118, sont difficiles à valider. Néanmoins le bruit au-delà de 1MHz semble correct mais est plus faible que dans les simulations

(-95dBc@1MHz au lieu de -89dBc@1MHz), ce qui est très peu probable. La Figure 119 permet de comparer les résultats de simulation et de mesure, et montre l'écart observé (remontée en mesure, difficile à comprendre) entre 10 et 100kHz.

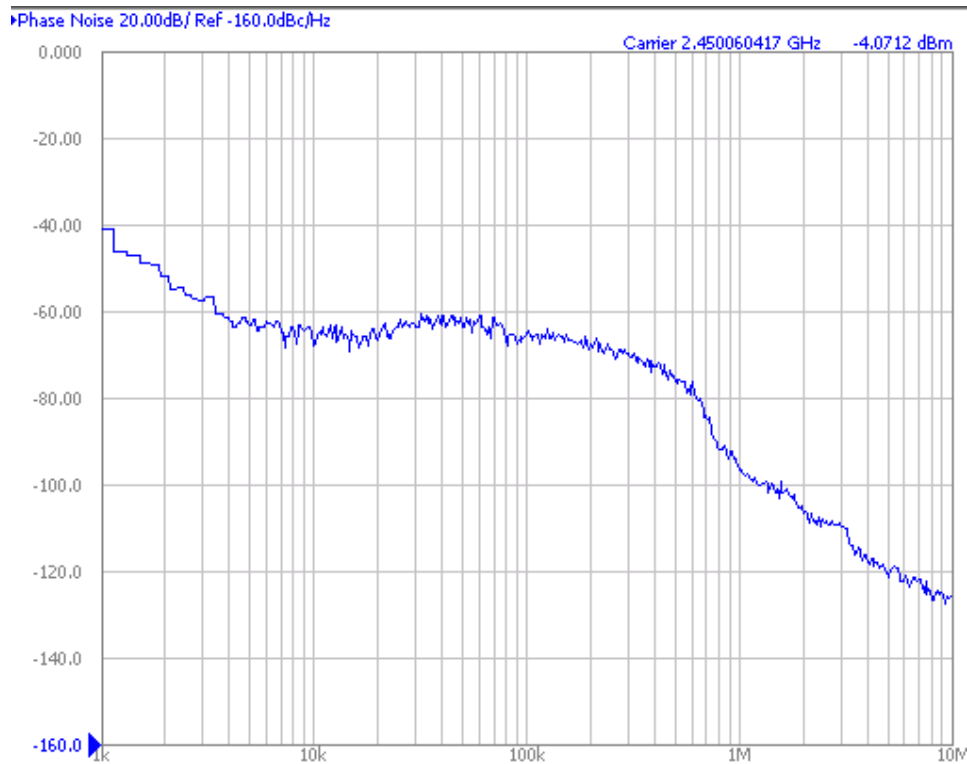


Figure 118 - Bruit de phase du VCRO 1 à 2,45GHz et Vdd = 1,04V et Vtune = 840,3mV

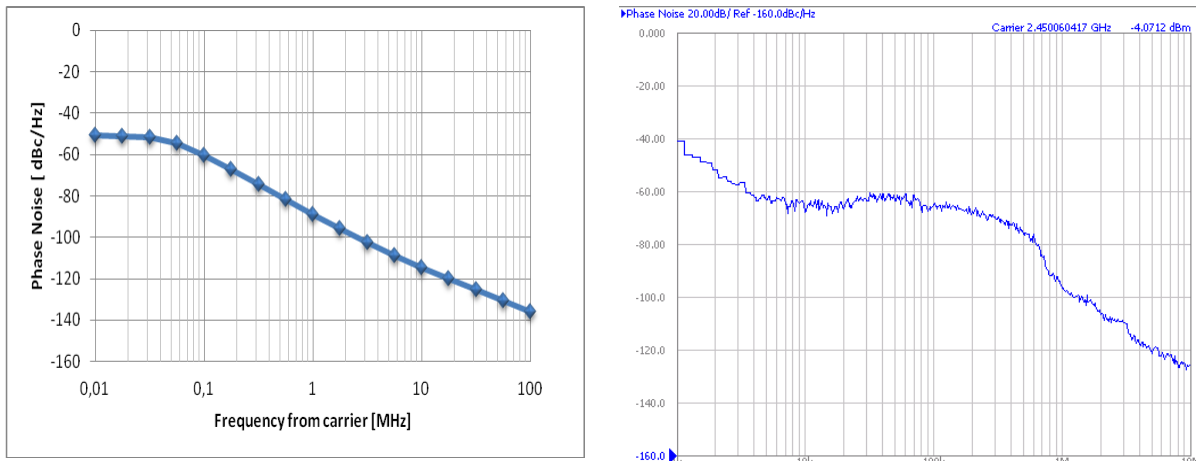


Figure 119 - Comparaison des bruits de phase simulé et mesuré

Nous avons ensuite tenté de retrouver le bruit de phase à l'aide de la mesure de Jitter effectuée par un oscilloscope haute fréquence (80GSa/s). La mesure donne une « Period Jitter » de 540fs_{RMS} (voir Figure 120), correspondant à un bruit de phase de -86,6dBc@1MHz, beaucoup plus proche des valeurs de simulation. Les autres VCRO (2, 3 & 4) ont un jitter de 710fs, 790fs et 840fs ce qui correspond à un bruit de phase respectivement de -84,3 ; -83,3 et -82,8dBc/Hz toujours à 1 MHz

de la porteuse. Ces grandeurs semblent là encore réalistes, mais nécessiteraient d'être validées par d'autres campagnes de mesures.

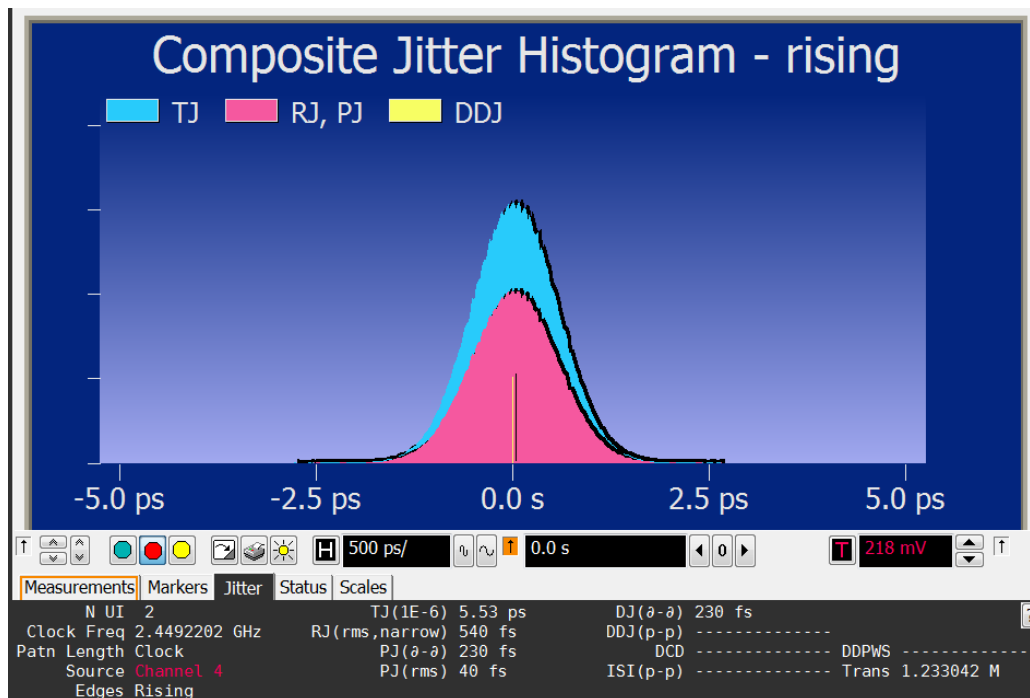


Figure 120 - Mesure de jitter du VCRO 1

5.5. Mesure du bruit de phase en boucle fermée

Pour pouvoir effectuer des mesures précises de bruit de phase, le test chip a été mesuré en boucle fermée grâce à la mise en place d'une PLL externe. Le circuit qui a permis de réaliser cette mesure est une carte d'évaluation du circuit intégré EV-ADF4153A, intégrant un circuit de VCO et le circuit de PLL fractionnaire ADF4153A. En débranchant le VCO de la carte, nous utilisons la PLL externe avec nos VCRO. La carte d'évaluation a été modifiée pour rendre accessible la tension de commande du VCRO et adapter la bande passante et la marge de phase de la PLL. La sortie RF de nos VCRO est câblée sur l'entrée du diviseur. Les composants passifs déterminant la bande passante de la PLL sur la carte ont été modifiés pour obtenir la bande passante la plus élevée possible. Malheureusement cette bande passante est limitée par la capacité parasite de nos VCRO (plot commun aux 4), par la capacité de filtrage interne à notre circuit (10pF) et par le VCO désactivé de la carte (20pF) donnant au total une capacité de 50pF sur le nœud de commande en fréquence. Cette capacité parasite 50pF limite la bande passante atteignable par la PLL à 0,5MHz, en gardant une marge de phase raisonnable de 50°. Le VCO n'a pas été dessoudé pour gagner en bande passante car la carte est utilisée par d'autres équipes. La fréquence de verrouillage de la PLL est de 2450MHz (25MHz x 98). Les mesures de bruit de phase des 4 VCRO ont été effectuées avec la méthode décrite ci-dessus. Les résultats sont illustrés par la Figure 121 et regroupés dans le paragraphe « Conclusion ».

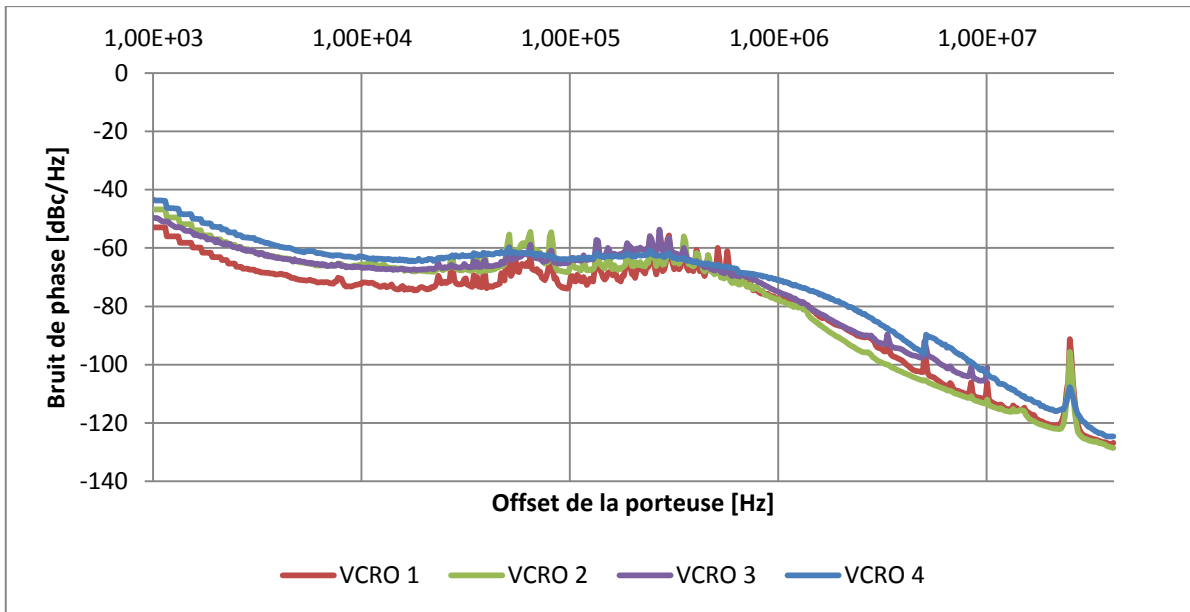


Figure 121 - Bruit de phase des 4 VCRO asservi par une PLL externe 0,5MHz 50°

Le VCRO n°4 étant très bruyant, il est mal verrouillé lors de la mesure (glitch à 5MHz pour toutes les mesures), il a donc été mesuré à plus basse fréquence, grâce à un diviseur externe par 8 (voir Figure 122). Le bruit de phase à 1MHz est de -109dBc/Hz, ce qui équivaut à -78dBc/Hz à la fréquence de 2450MHz ($+10 \cdot \log(32)$).

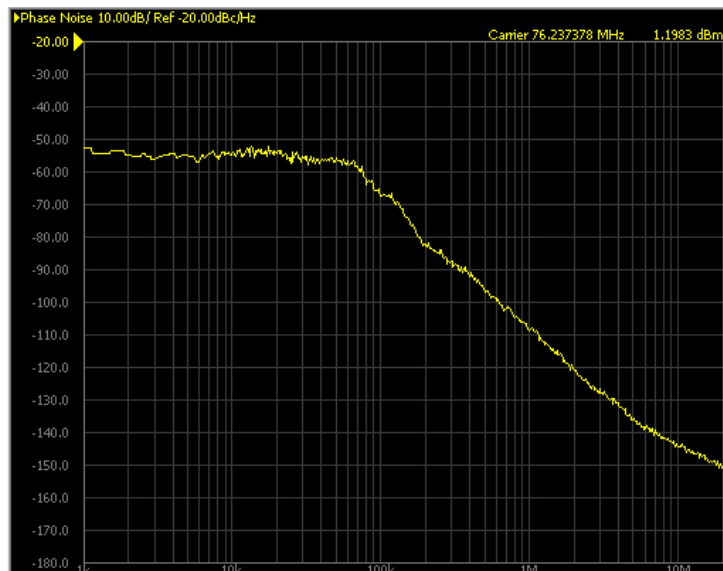


Figure 122 - VCRO 4 divisé par 32

5.6. Conclusion

Les performances des VCRO en termes de consommation et de gain extraites des simulations sont comparées dans le Tableau 13 aux mesures effectuées à 2,45GHz. On peut constater que la consommation est respectée mais que les caractéristiques qui dépendent de la tension

d'alimentation (Tuning) et surtout de grille arrière (Pushing) sont assez différentes des simulations. En effet, lors des mesures de tuning, il convient d'avoir une alimentation la plus stable possible en raison des fortes valeurs de pushing (#5GHz/V).

Tableau 13 - Performances nominales des 4 VCRO

VCRO (phases)	Consommation Simulée [μW]	Consommation Mesurée [μW]	Pushing simulé [GHZ/V]	Pushing Mesuré [GHZ/V]	Tuning simulé [MHz/V]	Tuning Mesuré [MHz/V]
1 (6)	800	788	5,35	4,57	556	P:-214-260 N:375-420
2 (30)	740	810	5,5	4,77	550	P:-184-217 N: 380-413
3 (16)	900	926	4,9	4	-252	P1:-83 -160 P2:-100 -177
4 (16) / 4	800+306	788 + 550	4,7	4,2	-245	P1:-90-166 P2:-83-159

Les mesures de bruit de phase ont été difficiles à cause de l'instabilité en fréquence des oscillateurs provoquée par la sensibilité aux variations d'alimentation (forte valeur de pushing). Utiliser une PLL, ayant la bande passante pour laquelle les VCRO ont été conçus (3MHz) ainsi qu'un filtrage de leur alimentation par un LDO interne très faible bruit et large bande (architecture feed-forward), devrait permettre de mesurer avec une meilleure précision le bruit de phase des VCRO fabriqués.

Toutefois, les 4 VCRO fabriqués semblent très bruyants : entre -70 et -80dBc/Hz en boucle fermée, soit 10dB au-dessus des valeurs de simulation (cf. Tableau 14). Les valeurs mesurées en boucle ouverte n'ont pas été validées car en-deçà des valeurs de simulation, sans réelle explication à ce jour. Les mesures en boucle fermée par contre sont assez pessimistes. Toutefois, il convient de nuancer ces résultats par le fait que la PLL externe mise en place n'est pas parfaitement adaptée aux VCRO conçus. Le bruit de phase du comparateur de phases, de la pompe de charge ou la fuite en courant des composants externes du filtre passe bas ont sans doute joué un rôle non négligeable sur ces valeurs. La formule (13) permet de calculer le bruit de phase équivalent du Jitter RMS mesuré d'un VCRO en donnant la fréquence du VCRO (F_{vcro}) et l'offset à laquelle on désire connaître le bruit de phase équivalent (ΔF). Cette formule a été adaptée du travail de Kundert sur la modélisation du bruit de phase des oscillateurs. Les valeurs calculées se révèlent être assez proches des simulations des VCRO single-ended mais trop optimiste pour les VCRO différentiels. Cette méthode ignore le bruit de Flicker, qui était prépondérant dès 1MHz dans les simulations.

$$Bruit_de_phase = 10 \log \left(\frac{Jitter^2 * F_{vcro}^3}{2 * \Delta F^2} \right) \quad (13)$$

Tableau 14 - Performances des 4 VCRO en bruit de phase [dBc/Hz@1MHz] et FOM [dBc/Hz]

VCRO N°	Simulation		Mesure en boucle ouverte			PLL		Mesure Jitter - transitoire		
	Bruit	FOM	Bruit	Equiv. @1MHz	FOM	Bruit	FOM dBc/Hz	Jitter	Equiv. @1MHz	FOM
1	-89	-158	-105dBc/Hz @2MHz	-99	-167	-77	-145	542fs	-86	-154
2	-87	-156	-104dBc/Hz @2MHz	-98	-167	-77	-144	668fs	-85	-154
3	-82	-151	-100dBc/Hz @2MHz	-94	-162	-75	-143	448fs	-88	-156
4	-79	-148	-102dBc/Hz @2MHz	-96	-162	-71	-137	587fs	-86	-152

6. Conclusion

Dans ce chapitre nous avons implémenté au niveau transistor (technologie FDSOI 28nm) 4 VCRO différents, les avons fabriqués au sein d'un circuit de test et nous avons mesuré leurs performances selon différentes méthodes.

Les résultats montrent que leur fréquence de fonctionnement et leur consommation sont très proches des simulations. Seul le diviseur par 4 à 10GHz connaît une grande variance, due au fait que les lignes de signaux sont très proches et nombreuses. A l'opposé, le gain en fréquence par rapport aux différentes tensions d'alimentation/polarisation est assez différent des simulations, dû au fait que les modèles ne sont adaptés qu'aux circuits numériques et que les effets de la grille arrière sur un circuit haute fréquence à grande excursion de tension soient assez mal connus.

Le bruit de phase des 4VCRO a été très difficile à mesurer et possède une grande incertitude (jusqu'à 10dB) selon la méthode utilisée. La présence d'une PLL et d'un LDO interne à grande bande passante aurait dû être implémentée pour garantir la précision des mesures.

Les VCRO ayant été implémentés sans PLL, les spurious n'ont pas pu être mesurés. Toutefois les modèles de simulation des variations technologiques étant assez précis, nous ne pensons pas qu'il y ait de différence significative de génération de jitter déterministe et de spurious.

En conclusion, ce chapitre nous a permis de tester les différentes architectures de VCRO et nous permet de conclure qu'un diviseur single-ended a des performances équivalentes à un VCRO différentiel et consomme légèrement moins.

Dans le chapitre suivant, nous concevrons les blocs manquants pour réaliser une PLL en technologie FDSOI. Chaque bloc sera conçu et optimisé séparément. Nous présenterons également la conception et la réalisation du Buffer 50Ω utilisé dans ce chapitre.

Chapitre IV – Etude de la PLL à resynchronisation de phases

1. Introduction

Ce chapitre présente les blocs de la PLL qui ont été conçus et simulés. Il convient de noter que nous n'avons pas eu le temps de dessiner les masques, les faire fabriquer et mesurer la PLL complète. D'autres circuits ont été conçus pour réaliser des simulations comparatives pour effectuer des choix d'implémentation dans ce travail de thèse, certains seront brièvement évoqués pour expliquer les simulations ou les choix effectués.

Les blocs implémentés de la synthèse de fréquences sont l'oscillateur en anneau (présentés au chapitre précédent et déclinés en 4 versions différentes), le diviseur à sélection de phases (2 versions différentes) et le détecteur de phase. Ce dernier est composé d'un comparateur de phases (numérique à 3 états), d'une pompe de charge (avec miroirs de courant) et d'un filtre de boucle du 2^{ème} ordre intégré sur silicium. Un buffer 50Ω a été conçu et optimisé pour extraire le signal carré de l'oscillateur à 2,45GHz. Ce circuit a été réalisé et testé au chapitre III. Toutes les simulations au niveau transistor ont été effectuées à la tension nominale du FDSOI, soit 1V. A part les VCRO présentés au chapitre précédent, et le buffer 50 Ω présenté dans ce chapitre, les autres blocs de la PLL n'ont pas été réalisés (layout et fabrication) ni mesurés. La consommation totale n'a donc pas été vérifiée.

Nous terminerons ce chapitre par quelques perspectives d'amélioration, notamment la technique d'auto-calibration par auto-polarisation de la grille arrière qui a fait l'objet d'un dépôt de brevet.

2. Diviseur fractionnaire

2.1. Introduction

Le rôle du diviseur fractionnaire est de transformer les phases fournies par le VCRO en un signal de plus basse fréquence qui est un sous multiple fractionnaire de la fréquence de ces phases. Ceci est possible grâce à la technique de la division resynchronisée sur phases ou division par commutation de phases expliquées au chapitre I. La faisabilité du diviseur FPD a été démontrée dans

le chapitre II. Nous allons maintenant démontrer la faisabilité de son implémentation au niveau transistor, qui est facilitée par les performances induites par la technologie FDSOI 28nm et en deçà.

2.2. Architecture

L'architecture du diviseur à commutation consomme plus d'énergie à cause du multiplexeur actif en permanence et travaillant à haute fréquence, c'est donc l'architecture à resynchronisation qui a été retenue pour l'implémentation au niveau transistor. Le diviseur FPD est composé d'un diviseur entier $N/N+1$, d'un additionneur à modulo pour le calcul des phases et d'un multiplexeur (cf. Figure 123 & Figure 124).

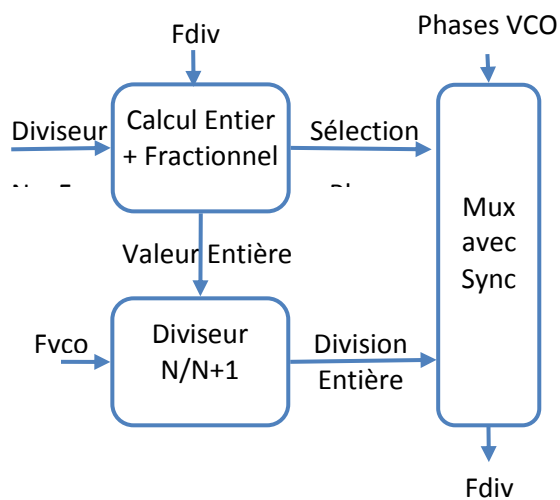


Figure 123 - Schéma bloc du FPD à resynchronisation

2.3. Spécifications

La consommation allouée au diviseur est de $100\mu\text{W}$, soit la moitié du budget restant des 1mW totaux après l'attribution de $800\mu\text{W}$ au VCRO. La fréquence de fonctionnement est de $2,5\text{GHz}$ pour les étages recevant les signaux du VCRO. Toutefois, les simulations sont effectuées à deux fois la fréquence nominale (5GHz) pour diminuer le temps d'optimisation/extraction des parasites du layout.

2.4. Implémentation au niveau transistor

2.4.1. Calculateur

Le calculateur permet de préparer la prochaine valeur de comparaison du diviseur entier et la prochaine phase de resynchronisation par le multiplexeur. Il est composé de deux additionneurs

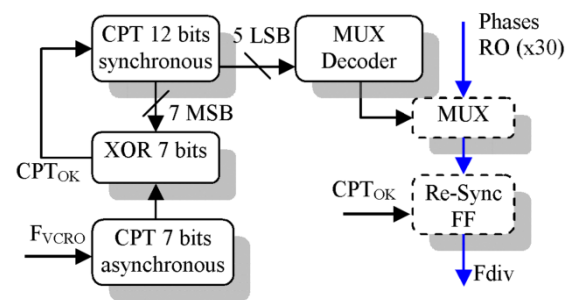


Figure 124 - Schéma bloc détaillé plus détaillé

commandés par la sortie du diviseur entier F_{divInt} , comme le montre la Figure 125. L'additionneur de la partie entière reboucle sur lui-même modulo 256 (8 bits comme le diviseur entier) pour que le compteur entier asynchrone n'ait pas à être remis à zéro, ce qui est difficile à réaliser à 2,45GHz. L'additionneur de la partie fractionnaire est au modulo du nombre de phases du VCRO, dès qu'il y a dépassement, la valeur entière est incrémentée de 1. Pour un nombre de phases qui est une puissance de deux (2^N), un seul additionneur suffit.

La sortie de l'additionneur de la partie entière (N_{val}) commande le déclenchement du diviseur entier. La sortie de l'additionneur de la partie fractionnaire commande la sélection de la phase correcte dans le multiplexeur. Il n'y a aucune difficulté d'implémentation tant au niveau consommation que fréquence de fonctionnement car l'horloge F_{divInt} est à la fréquence de la référence qui est inférieure à 50MHz.

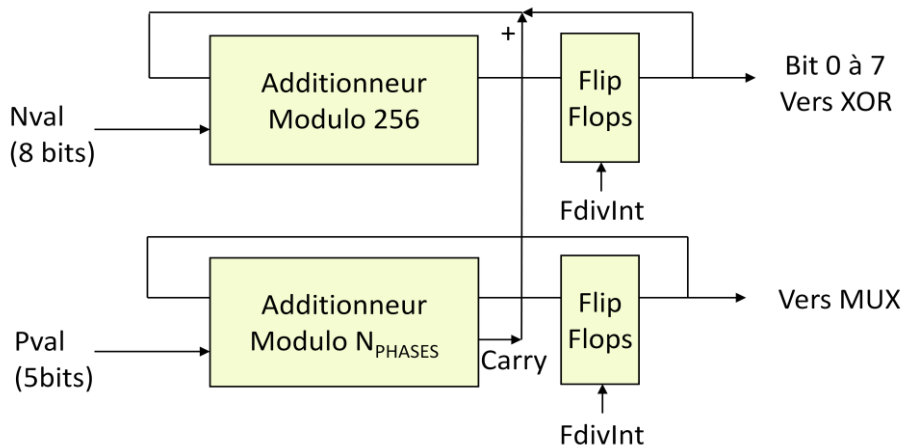


Figure 125 - Schéma bloc détaillant le calculateur

L'additionneur est une IP full-adder de ST Microelectronics en FDSOI. La flip-flop, présentée sur la **Erreur ! Source du renvoi introuvable.**, a été développée pour ajouter des fonctions d'initialisation, comme le chargement d'une valeur, à une bascule D basique pour pouvoir initialiser les additionneurs au démarrage.

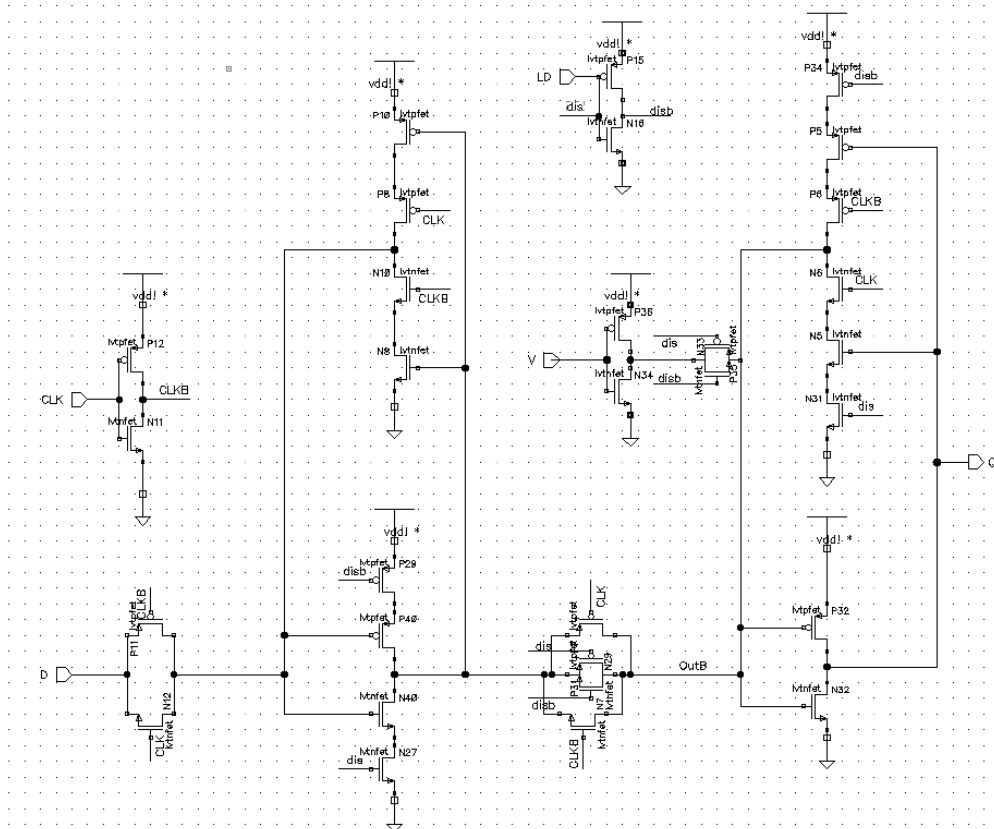


Figure 126 - Flip Flop avec pré-charge de valeur

2.4.2. Diviseur Entier

Le diviseur entier (cf. Figure 127) est composé d'un compteur asynchrone et d'un comparateur, présentés sur la Figure 129 et la Figure 130. La fréquence du VCRO entre dans la première des flip-flops T et se propage par division successive par 2 aux 7 autres. Les 8 portes XOR comparent les sorties des flip-flops au mot 8 bit calculé par l'additionneur modulo 256. Le temps de propagation entre la première flip-flop et la dernière porte XOR est inférieur à une période du VCRO, et a été testé à 5GHz sans extraction des parasites au niveau layout.

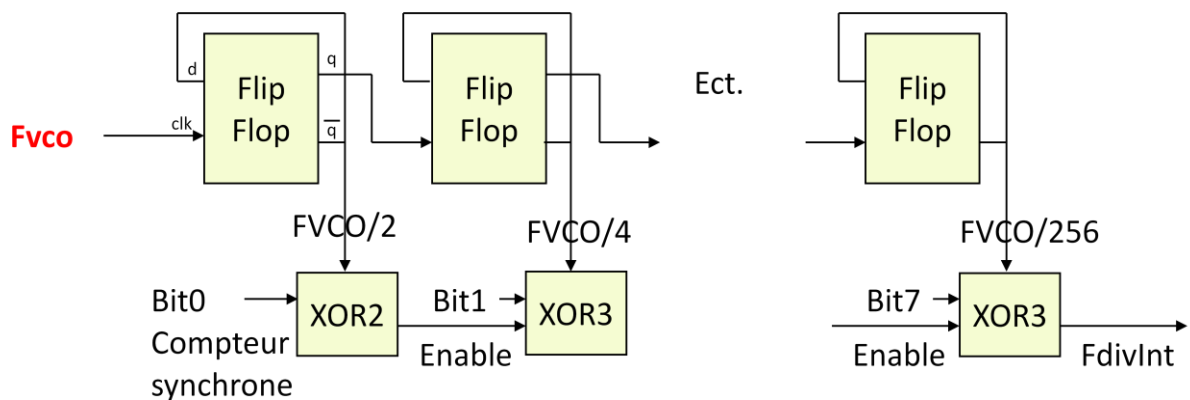


Figure 127 - Schéma bloc du diviseur entier à base de FlipFlop T et XOR 3 entrées

L'architecture asynchrone du diviseur permet d'économiser de l'énergie car l'étage N ne reçoit un signal d'horloge qu'une fois toutes les 2^N périodes du VCRO. La surface de silicium aussi est optimisée car il n'y a pas de portes logiques autres que nos flip-flops simplifiés, car ne possédant ni Reset, ni chargement parallèle. Elles ont été optimisées pour fonctionner à une fréquence assez élevée (5GHz en simulation) et avoir une consommation maîtrisée (50µA pour le diviseur complet).

Les flip-flop T spécialement conçues (cf. Figure 129) sont basées sur une architecture intégrant seulement 4 inverseurs et 4 interrupteurs (voir Figure 128), plus un inverseur pour l'entrée horloge. Chaque largeur de transistor a été optimisée pour permettre de travailler à des fréquences élevées en maîtrisant la consommation. Par exemple, le premier interrupteur et les inverseurs de feedback sont de taille minimale (80/30nm) car leur charge est faible. Le deuxième interrupteur est plus large car il amorce le basculement de la sortie. Les inverseurs directs sont de taille proportionnelle à leur charge (étage suivant, FF + XOR3) en un temps minimal (12,5ps par latch).

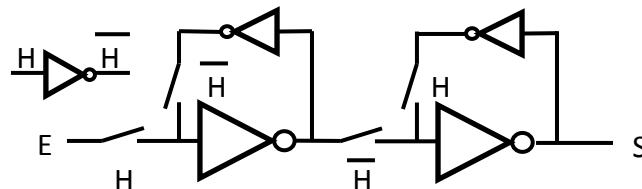


Figure 128 - Schéma idéal d'une flip flop D dans l'architecture choisie

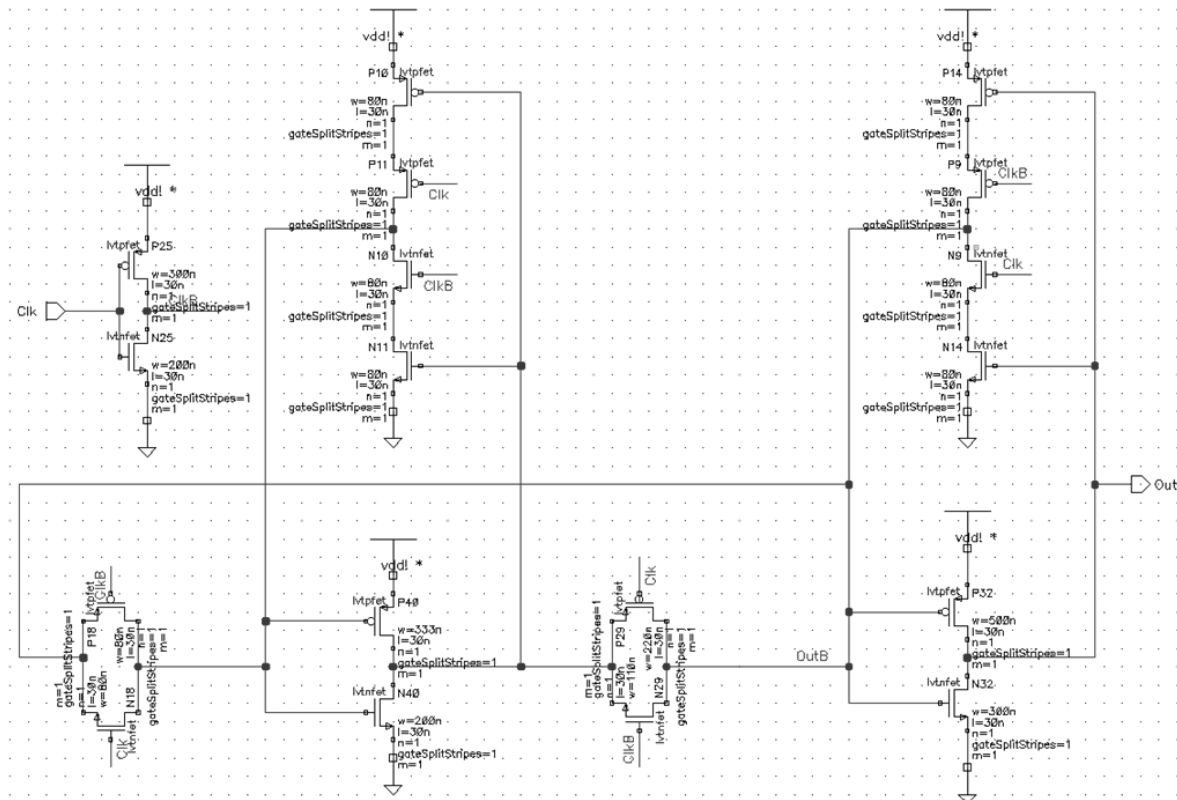


Figure 129 - FlipFlop T du compteur asynchrone

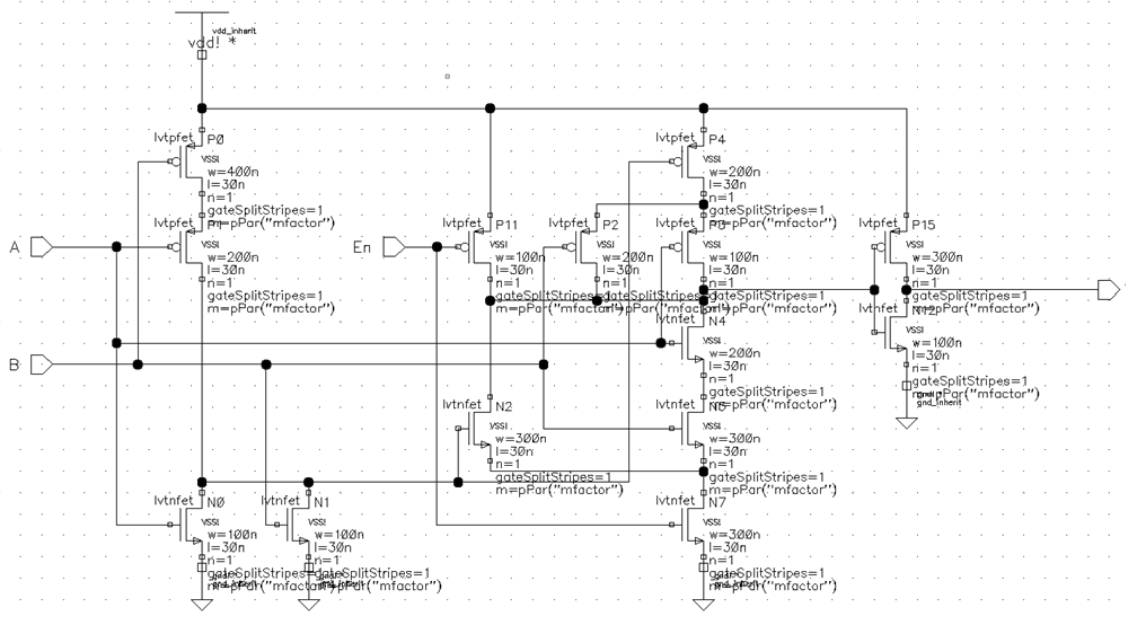


Figure 130 - XOR 2 + Enable ultra rapide à 14 transistors

Le comparateur est constitué d'une chaîne de 8 portes XOR2 avec entrée de validation. Ces portes, composées de seulement 14 transistors, effectuent la comparaison (A xor B) seulement si le résultat de l'étage précédent est vrai. Cela permet de diminuer la consommation, la surface occupée et le temps de propagation, mais surtout d'éviter de propager des glitches vers le diviseur entier.

2.4.3. Multiplexeur à resynchronisation

Le multiplexeur, présenté sur la Figure 131 est composé d'un décodeur, d'un multiplexeur et d'une flip-flop. Le décodeur effectue le décodage du mot de sélection grâce à 30 portes AND à 4 entrées (standard cells ST). Le multiplexeur sélectionne la phase grâce à 30 cellules à 3 états (cf. Figure 134). Une flip-flop, modifiée avec l'ajout de 2 entrées set et reset (cf. Figure 135), resynchronise la sortie du diviseur entier au front montant de la phase sélectionnée. Une fois Fdiv généré, une temporisation (~60ps) active le reset des flip-flop de sélection ce qui désactive le multiplexeur.

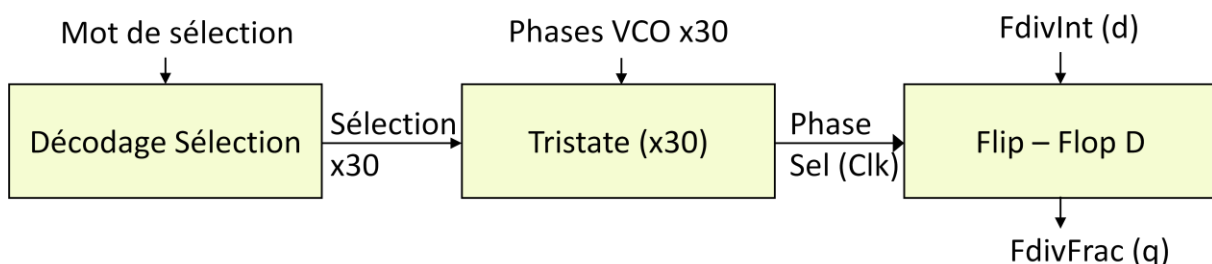


Figure 131 - Schéma bloc détaillant l'implémentation du multiplexeur à resynchronisation

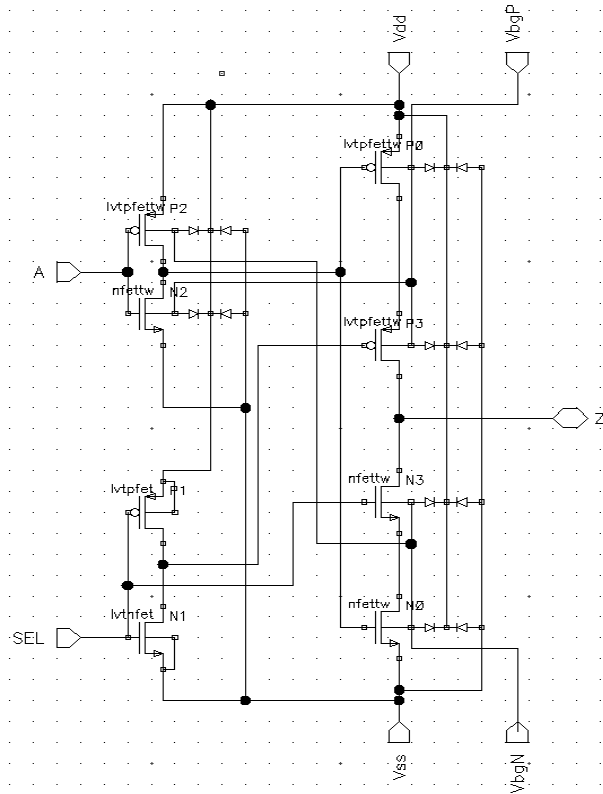


Figure 132 - Cellule de sélection 3 états

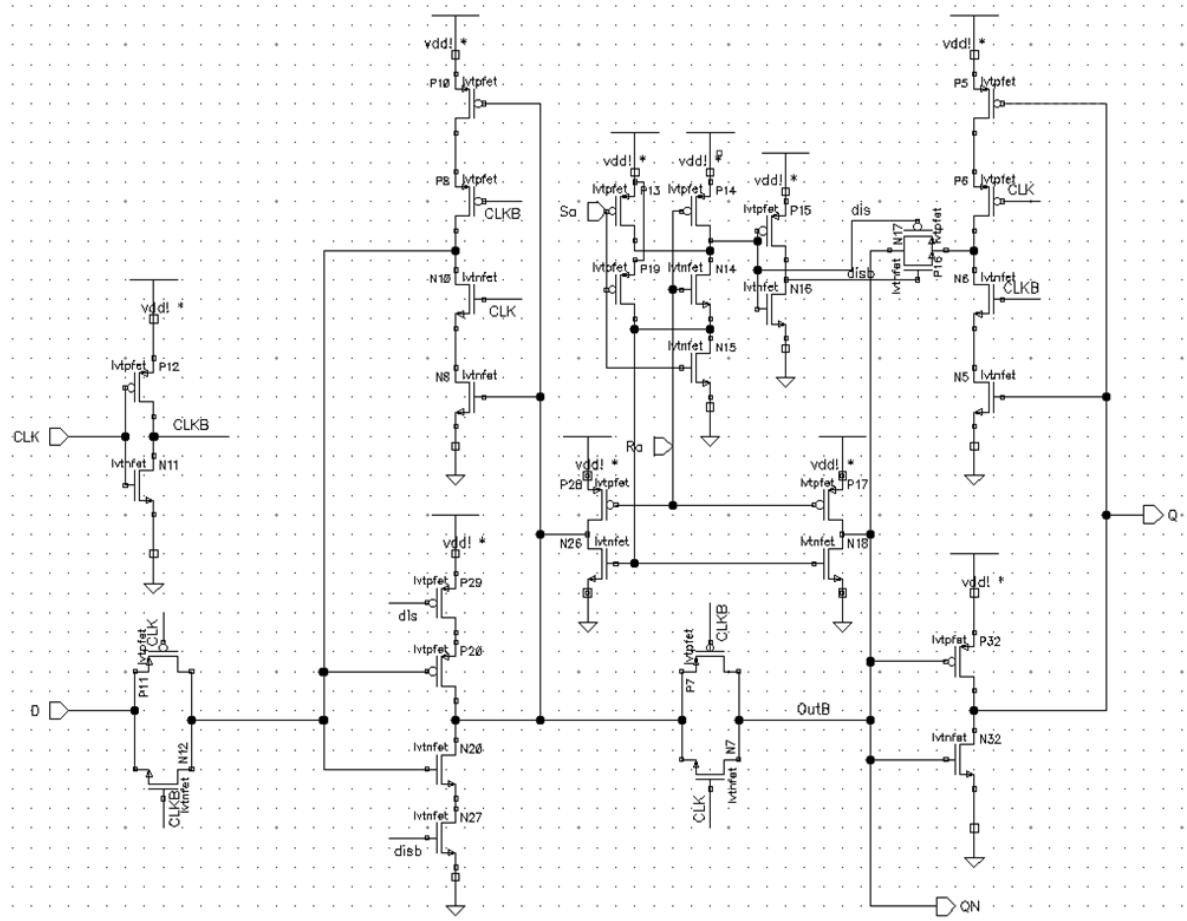


Figure 133 - Flip Flop avec entrée Set et Reset

2.4.4. Conclusion

Le schéma final du diviseur fractionnaire implémenté regroupe les 3 blocs décrits ci-dessus (calculateur, diviseur entier et multiplexeur) ainsi que d'autres blocs servant à initialiser, synchroniser ou temporiser les signaux internes du diviseur fractionnaire, faisant au total plusieurs centaines de transistors. On retrouvera, sur la Figure 136, un schématique de ce circuit sous Cadence.

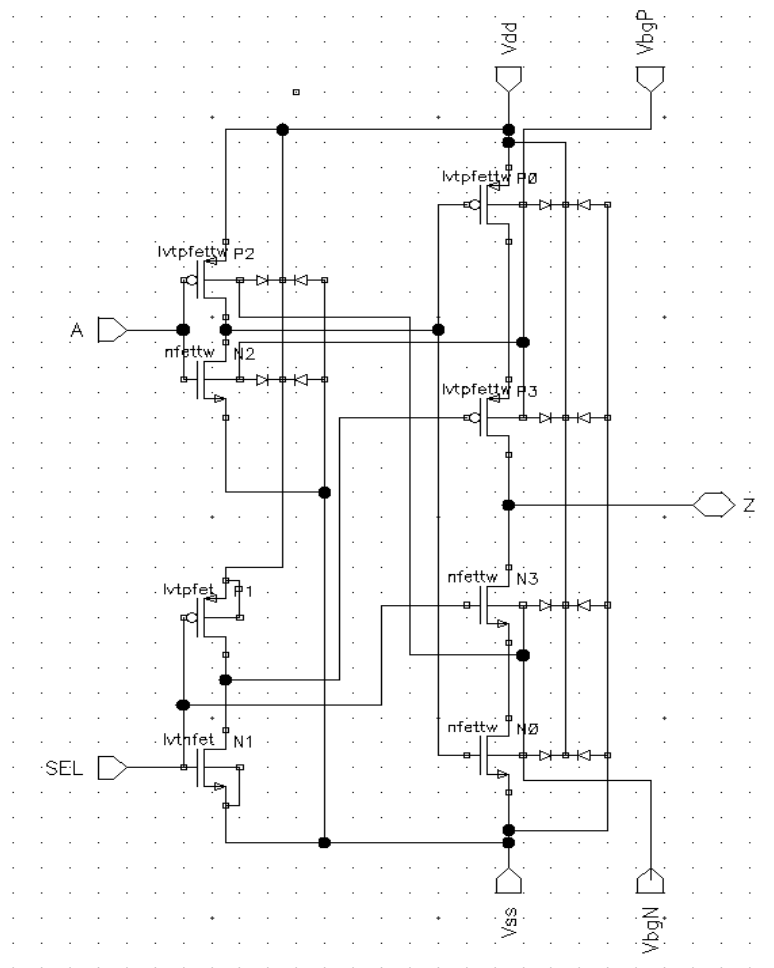


Figure 134 - Cellule de sélection 3 états

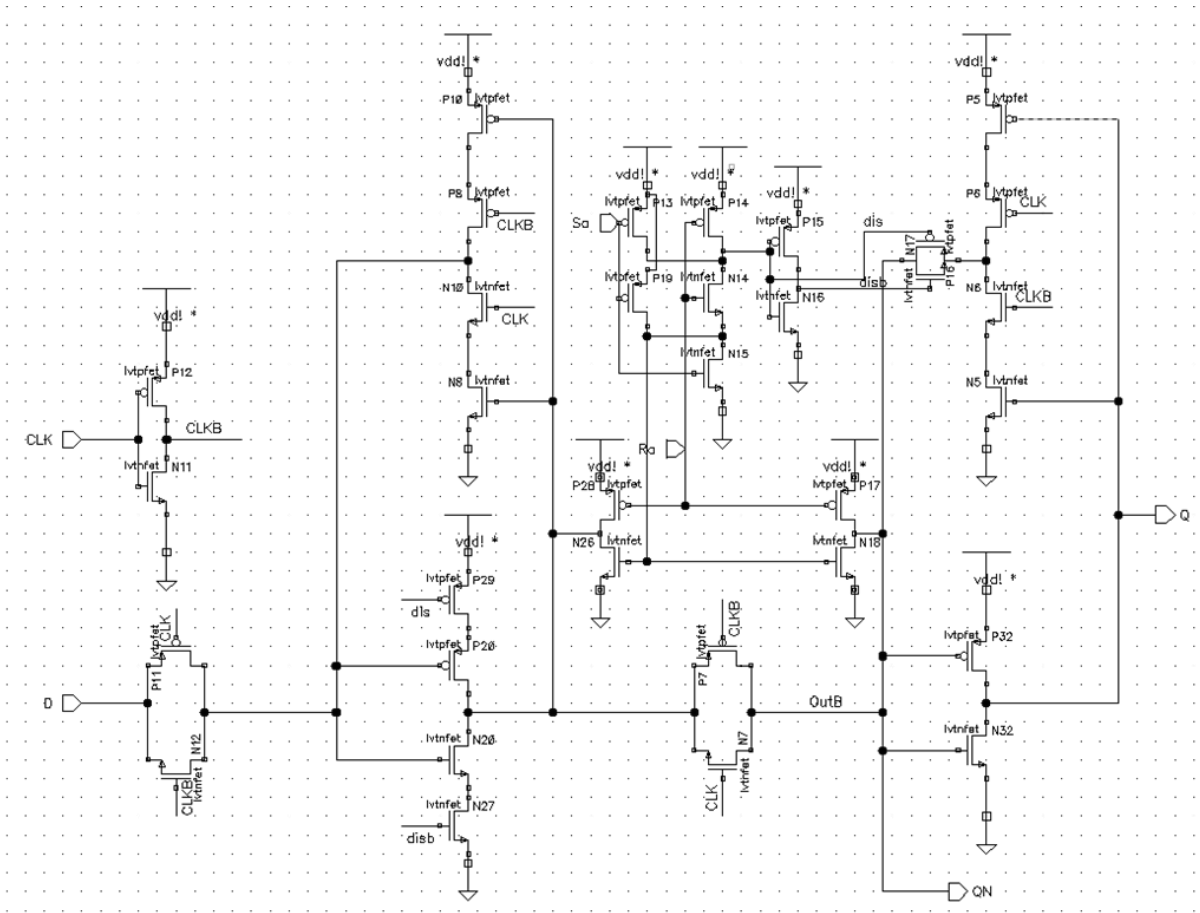


Figure 135 - Flip Flop avec entrée Set et Reset

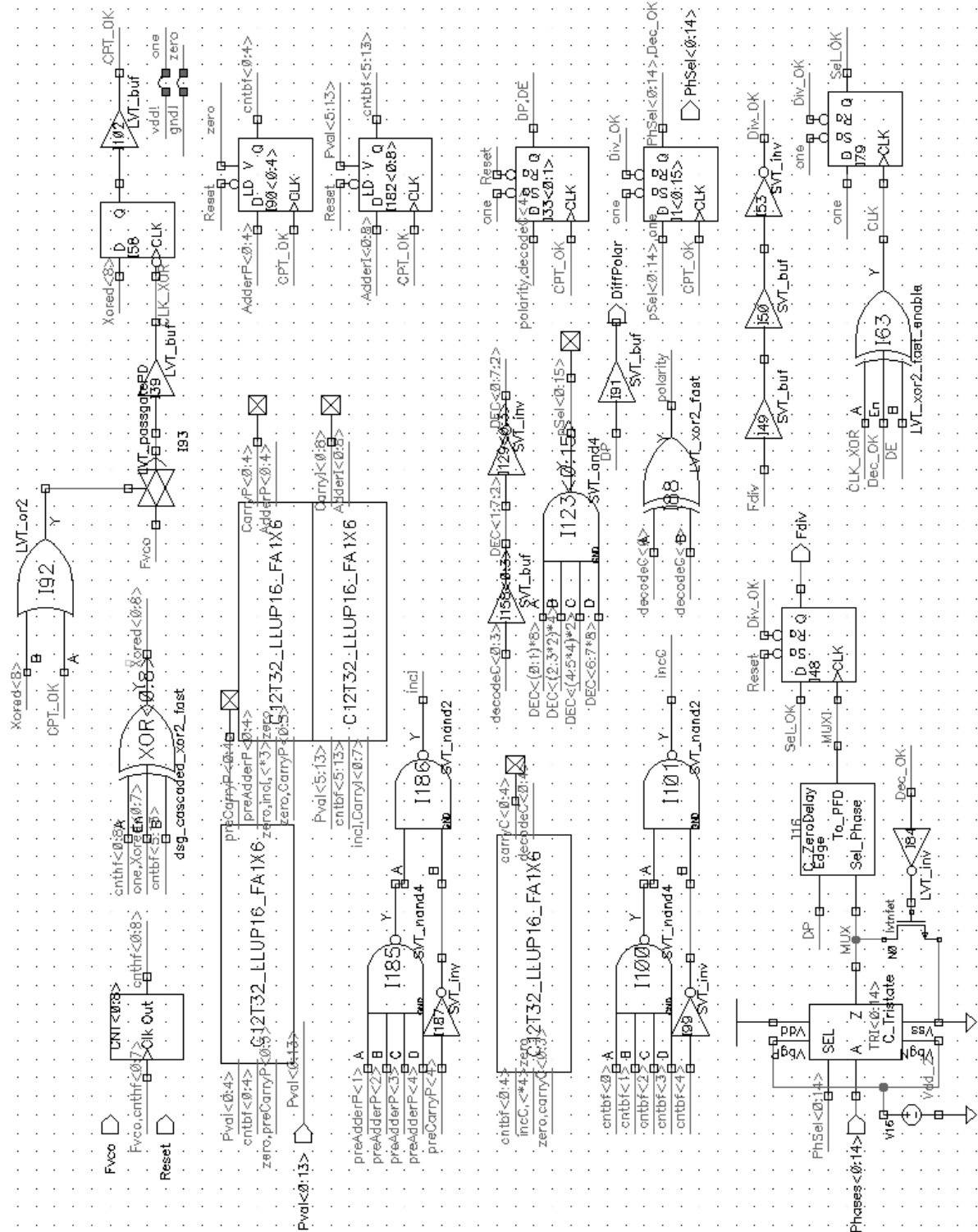


Figure 136 - Schéma du diviseur fractionnaire complet

2.5. Simulation

La simulation du diviseur à resynchronisation a, dans un premier temps, été effectuée avec un VCRO idéal comme générateur de phases. Diverses simulations ont permis d'éliminer certaines architectures et d'affiner l'architecture choisie (présentée à la section précédente). En effet, des

problèmes de synchronisation et de vitesse de commutation ayant été soupçonnés, quelques ajouts de portes logiques et de buffers ont rendu l'architecture fonctionnelle.

La simulation de la partie fractionnaire est illustrée sur la Figure 135. Dans cet exemple, nous avons utilisé un VCRO à 16 phases, soit une valeur fractionnaire pouvant varier de 0/16 (chronogramme rouge en haut) à 15/16 (chronogramme jaune pâle en bas). A chaque coup d'horloge $FDIV = Fosc / Rang_de_division$ (16MHz), représenté par l'axe des abscisses, le multiplexeur sélectionne la phase de resynchronisation. Pour une valeur de fraction donnée (chronogramme 0 à 15), le chronogramme correspondant affiche la valeur de la phase sélectionnée (entre 0 et 15, pour chaque chronogramme). Ainsi pour une fraction de 0/16 (premier chronogramme), cette valeur reste à 0; et pour une valeur de 15/16 (dernier chronogramme), cette valeur décroît d'une unité à chaque coup d'horloge, passant par toutes les valeurs entières de 15 à 0.

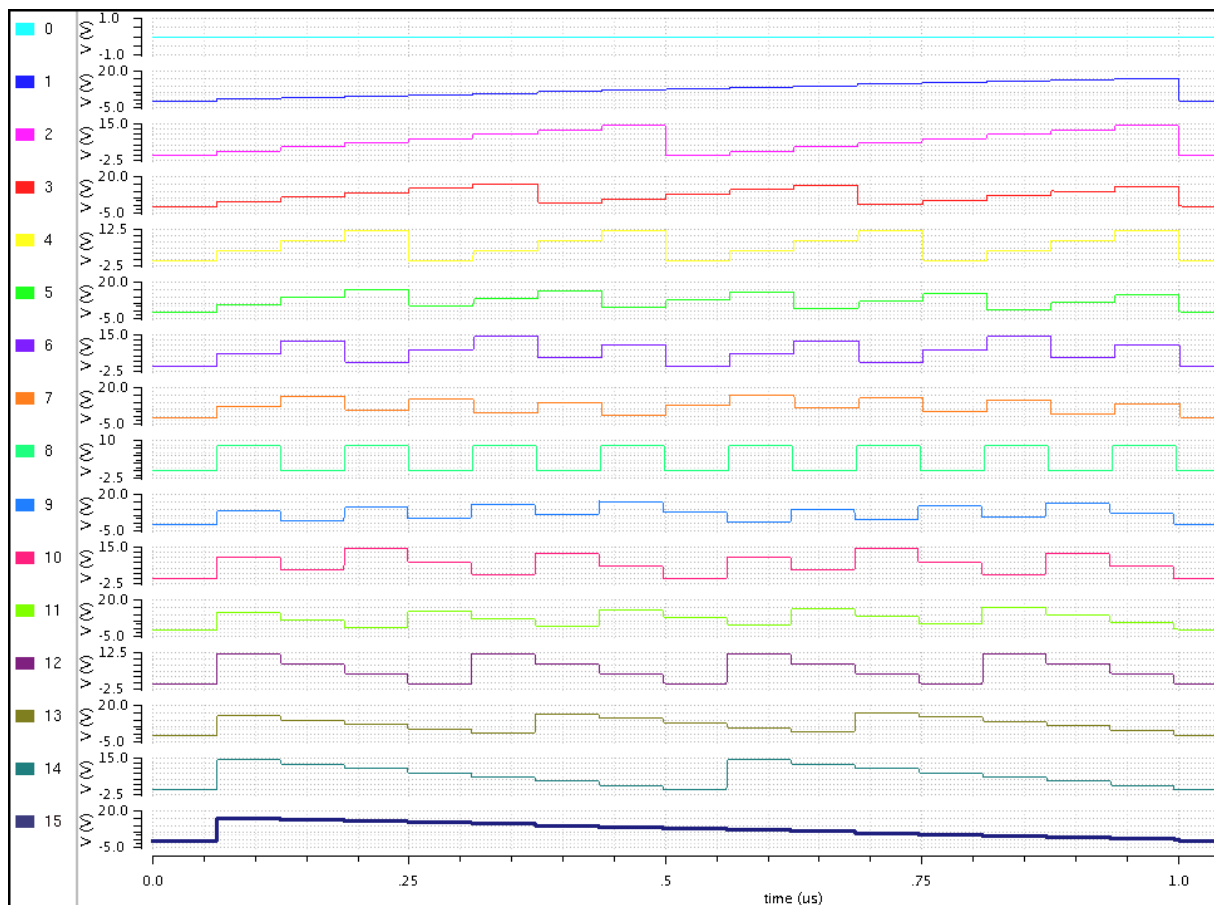


Figure 137 - Valeurs prises par le mot de sélection du multiplexeur pour toutes les valeurs fractionnaires de la division en fonction des cycles de division (Fdiv)

La simulation de la partie entière et la visualisation de la sortie finale sont illustrées par la Figure 138. Le premier chronogramme représente la phase 0 du VCRO, le dernier est le signal de sortie du diviseur entier. Au centre se trouve le signal généré par la resynchronisation. Pour pouvoir

visualiser le décalage dans le temps du signal de sortie fractionnaire, le rang de division a été choisi à 4 et 1/8, soit 4,125.

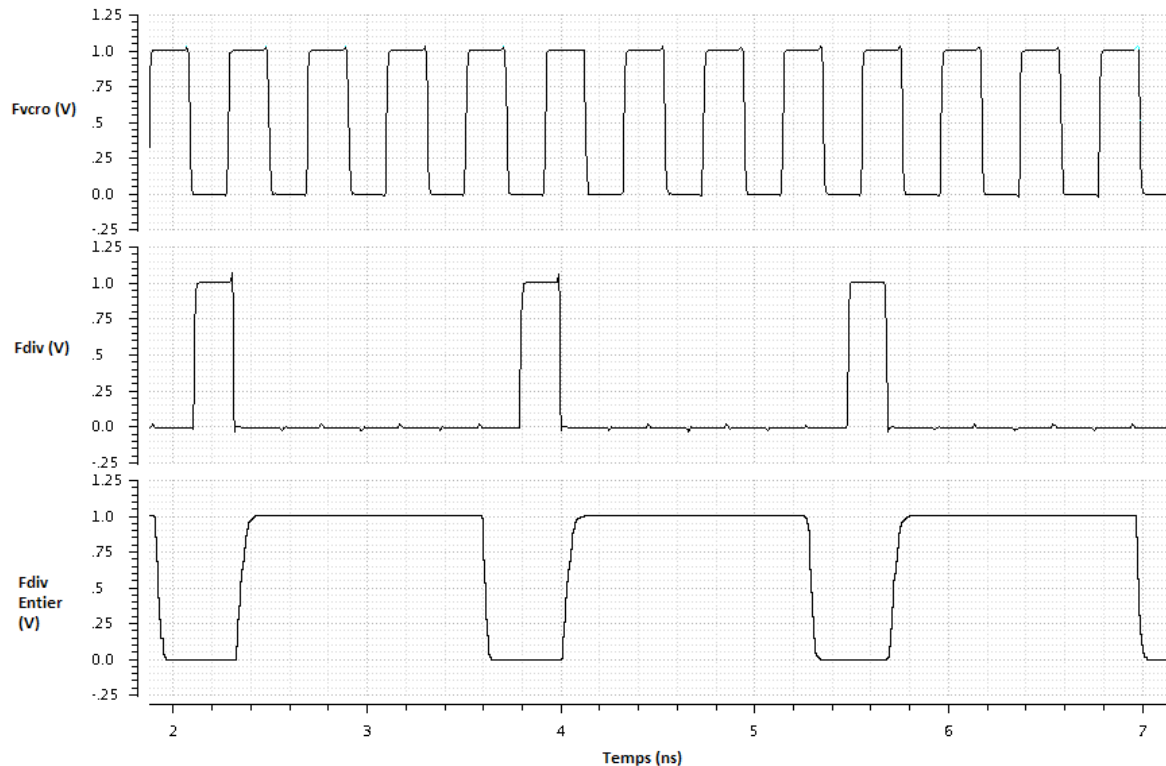


Figure 138 - Division par 4,125 (Fvcro, Fdiv et FdivInt)

2.6. Conclusion

Le diviseur entier divise le signal du VCRO à 2,5GHz et peut changer de rang de division ($N/N+1$) à chaque division (400ps max, testé aussi avec 200ps). Le multiplexeur s'active et se désactive en moins d'une demi-période VCRO (200ps max, test avec 100ps) avec une consommation maîtrisée (inférieure à 50 μ A pour l'ensemble des 3 blocs). Le calculateur se charge de préparer les valeurs entière et fractionnaire pour le prochain cycle de division.

3. Détecteur de phase et filtre

3.1. Introduction

Une PLL est un système qui asservi la phase du VCRO par une rétroaction, maintenant une erreur nulle entre la phase du diviseur, et donc du VCRO, à celle de la référence (oscillateur à quartz). La mesure de l'erreur est réalisée par le détecteur de phase, et la correction de la phase sur le VCRO

par la pompe de charge après filtrage. La bande passante et la marge de phase de la PLL sont ajustées par les paramètres du filtre de boucle.

Pour notre application, la bande passante de la PLL est élevée (3MHz), et les spécifications de bruit de phase de la PLL pour le canal adjacent ($-90\text{dBc}/\text{Hz}@1\text{MHz}$) tombent dans la bande. Le bruit dans la bande étant défini principalement par le bruit du comparateur de phases, celui-ci a dû être optimisé. Il convient de noter que le bruit de la référence (Quartz) peut également être important avant la fréquence de coupure, mais nous ne pouvons que le subir.

3.2. Spécifications

D'après les simulations système effectuées avec les modèles du chapitre II, la bande passante optimale retenue est de 3MHz et la marge de phase de 50° . Le bruit de phase vu à l'entrée du détecteur doit être inférieur à $-100\text{dBc}/\text{Hz}$ à 3MHz. Cette valeur est compatible avec les spécifications de bruit du VCRO ($-90\text{dBc}/\text{Hz}@1\text{MHz}$) moins 10dB dû au filtrage de la boucle de -20dB par décade et de l'écart de 3MHz ($20\text{dB}/\text{DEC} * \log_{10}(3[\text{MHz}])$) (voir Figure 139).

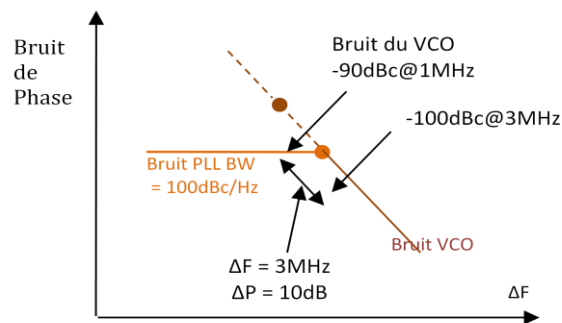


Figure 139 - Bruit de phase dans la bande

Pour atteindre un bruit de phase de $-100\text{dBc}/\text{Hz}$ et une consommation inférieure à $50\mu\text{A}$, le courant de pompe de charge a été optimisé à $50\mu\text{A}$ et la durée minimale d'activation (dead-time) à 50ps par des simulations au niveau système grâce aux modèles développés au chapitre II.

3.3. Implémentation

3.3.1. Comparateur de phases

La topologie du comparateur de phases a été reprise de la littérature (cf. Figure 140) : c'est un comparateur de phases numérique à trois états (three-state phase-comparator) avec entrée Reset.

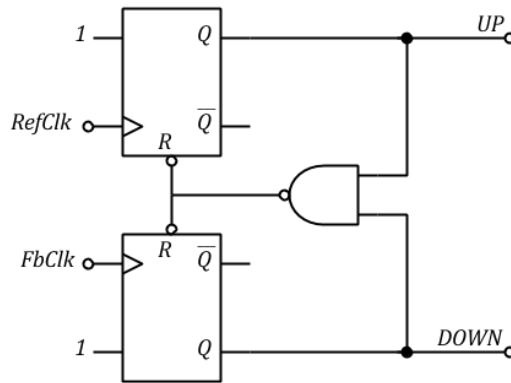


Figure 140 - Schéma logique d'un comparateur de phases élémentaire

Cette architecture a été optimisée, pour respecter la spécification de temps minimum d'activation (50ps), en modifiant la taille des transistors de chaque porte logique. Les largeurs de transistors sont données sur la Figure 141. Les valeurs WpDivWn représentent le rapport de tailles entre Pmos sur Nmos, et les valeurs mfactor représentent la largeur du transistor Nmos en multiple de la taille minimale (80nm). La longueur des transistors a été prise à la valeur minimale de la technologie FDSOI, à savoir 30nm.

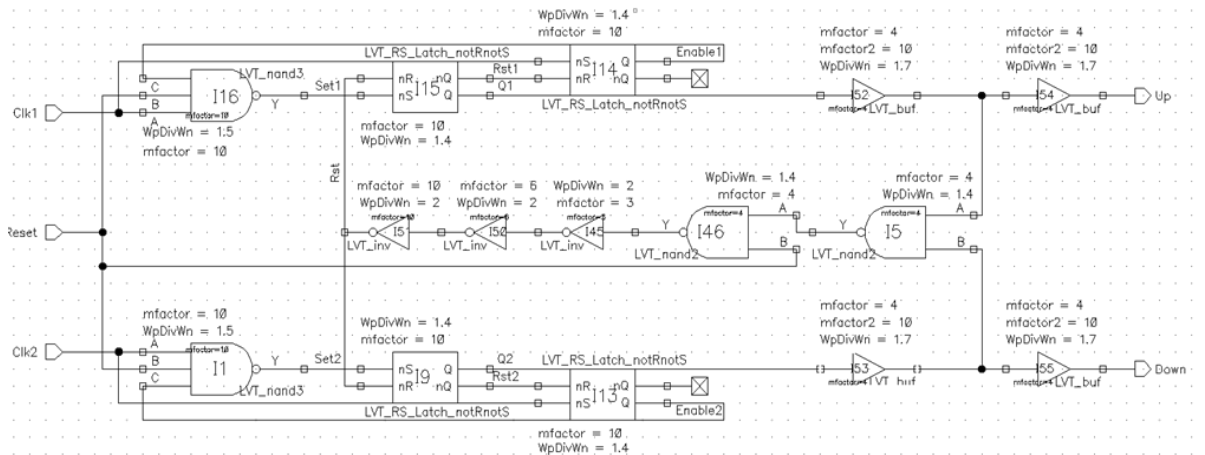


Figure 141 - Schéma bloc du comparateur de phases avec les dimensions des portes

3.3.2. Pompe de charge

Le circuit de pompe de charge, illustré par la (Figure 144) est basé classiquement sur des miroirs de courant. Le courant de 50µA n'est pas une spécification contraignante. Toutefois les simulations de mismatch ont montré qu'un déphasage entre les signaux de commande (Up et Down respectivement leurs signaux complémentaires) provoquait une injection de charge dans le VCRO que le filtre de boucle ne peut filtrer entièrement. La solution retenue, la plus efficace, est d'équilibrer les commandes à l'aide d'un buffer différentiel (cf. Figure 142). Ce buffer est utilisé au

sein d'une porte logique de génération des signaux en opposition, présentée sur la Figure 143. Cette porte ajoute une charge fictive sur le signal non inversé pour pré-équilibrer les signaux de commande. Le bruit blanc de la pompe de charge a été optimisé en augmentant les dimensions des sources de courant et le bruit transitoire a été optimisé en ajoutant des capacités de 100aF aux nœuds haute impédance de ces sources de courant.

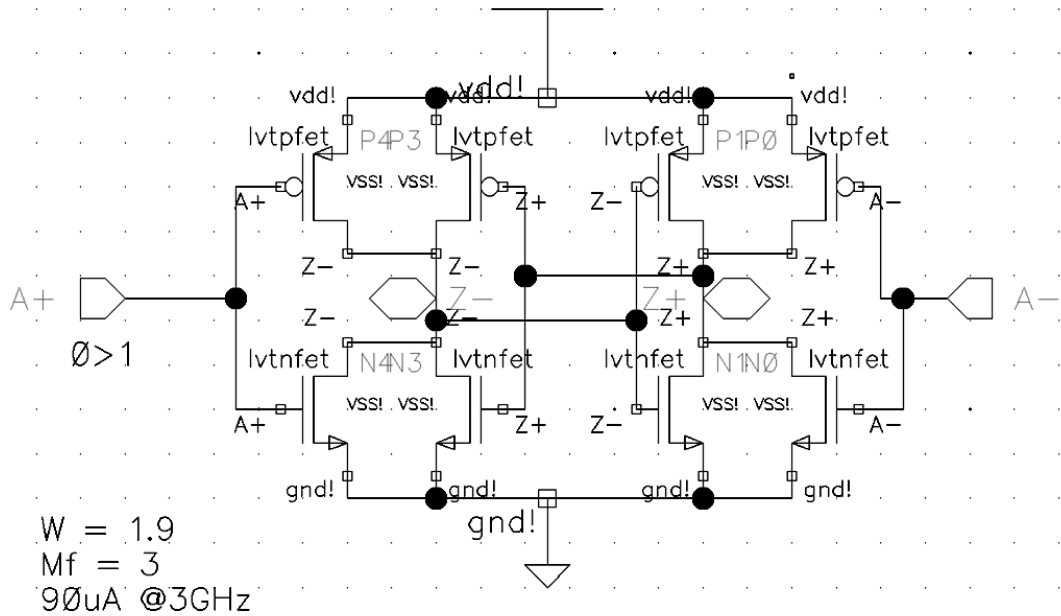


Figure 142 - Buffer différentiel d'équilibrage des fronts

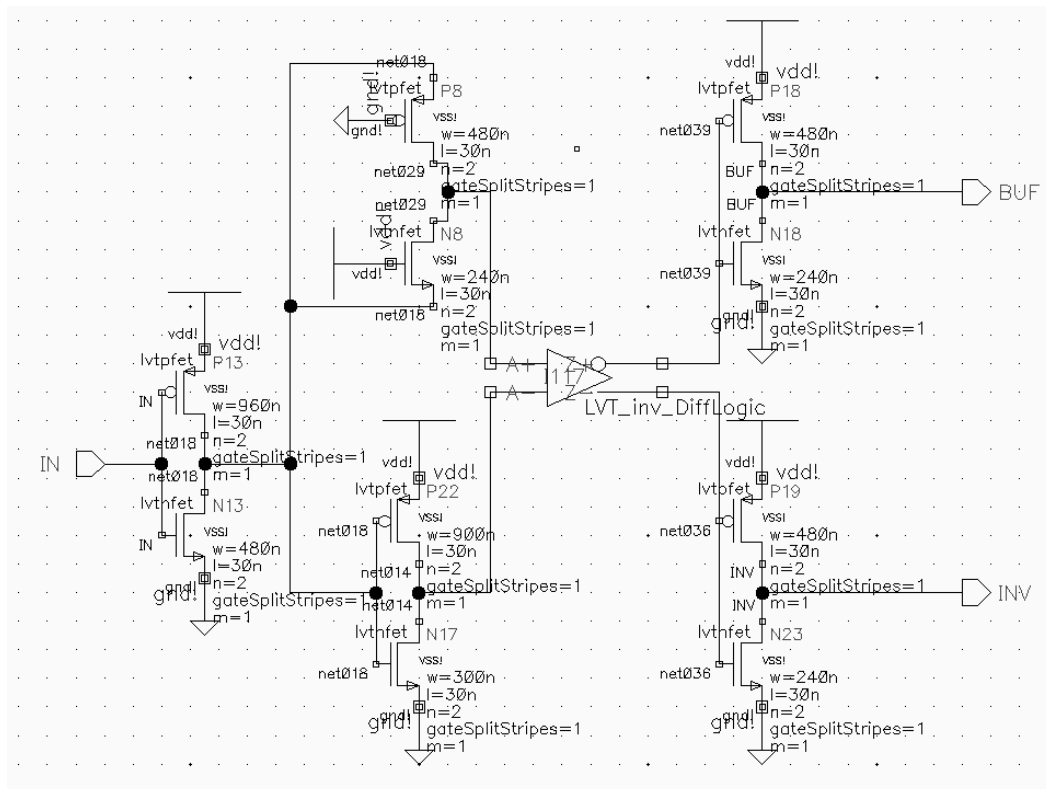


Figure 143 - Porte logique de création de 2 signaux en opposition

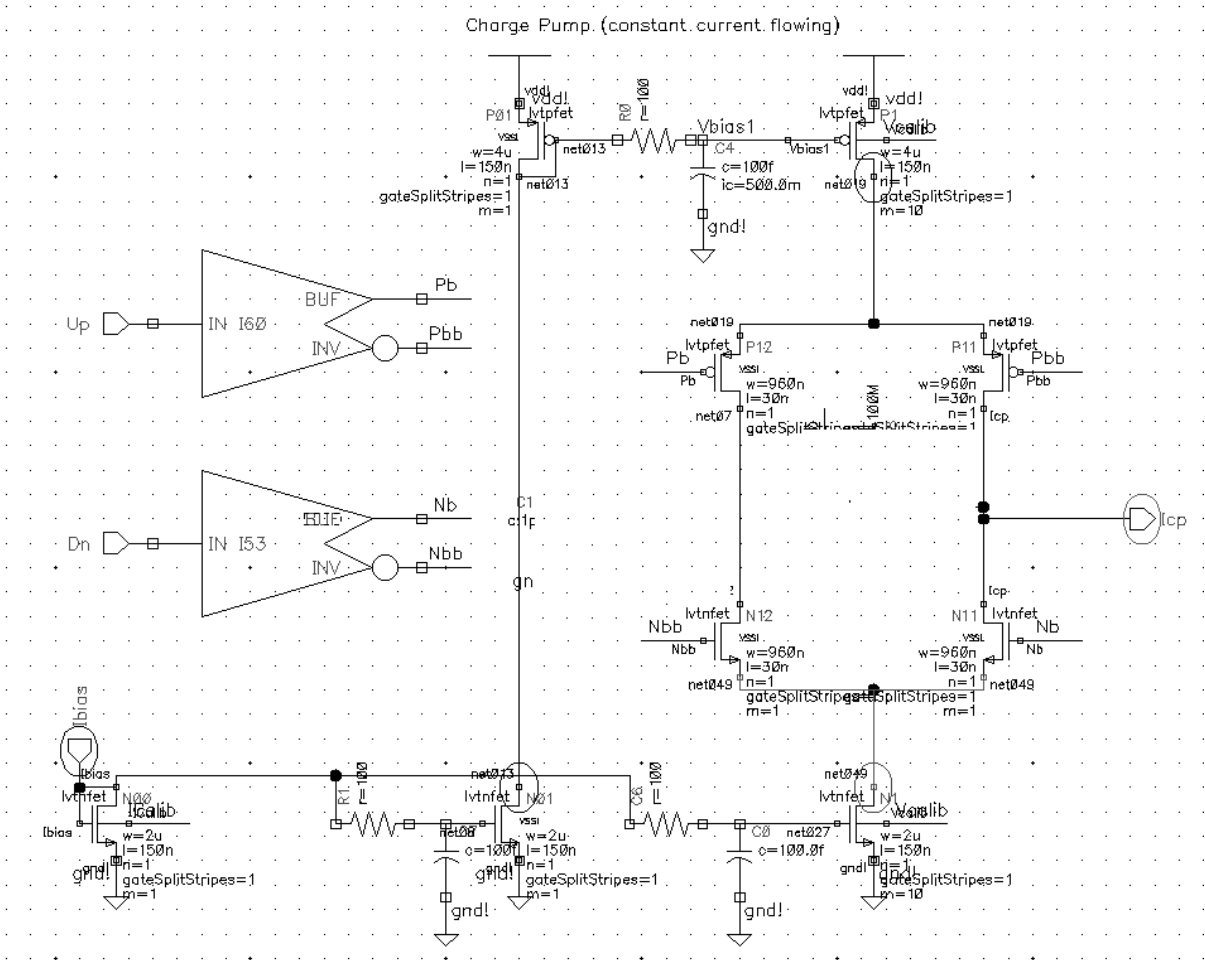


Figure 144 - Schéma au niveau transistor de la pompe de charge

3.3.3. Filtre de boucle

Les valeurs du filtre de boucle pour parvenir à une PLL avec une marge de phase de 50° et une bande passante de 3MHz sont les mêmes que lors de la simulation système du chapitre II : $R1 = 0\Omega$, $R2 = 128K\Omega$, $C1 = 1,56pF$ et $C2 = 120fF$ (cf. Figure 145).

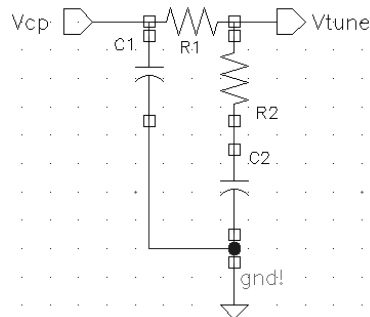


Figure 145 - Schéma du filtre de boucle

3.4. Simulation

3.4.1. Comparateur de phases

La première simulation temporelle, présentée Figure 146, montre que le temps de propagation du détecteur, quand l'une ou l'autre des entrées du comparateur de phases est en avance de 100ps (haut et centre), est de 54ps. Le temps d'activation minimal des sorties Up/Down, quand les deux entrées sont en phase (bas), est de 81ps. Malgré les efforts d'optimisation, le dead-time de 50ps n'a pas été atteint.

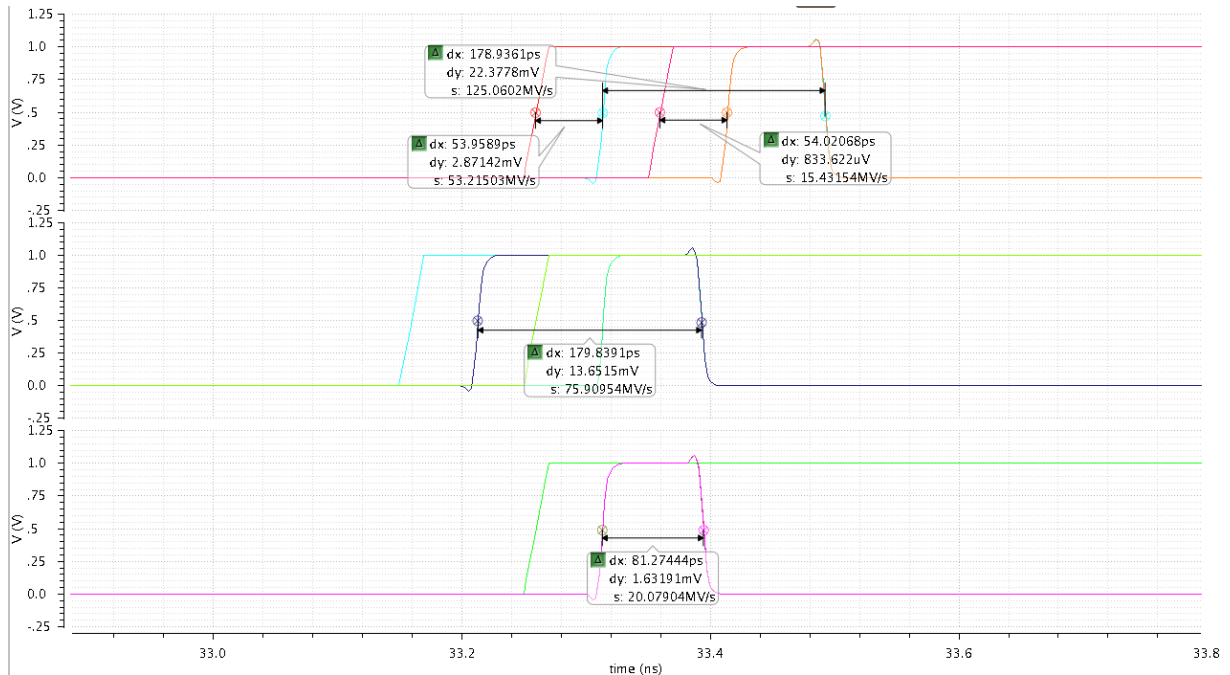


Figure 146 - Temps de propagation et temps d'activation minimal

La consommation du PFD, illustrée par la Figure 147 est de $30\mu\text{A}$ à l'arrêt (due aux polarisations de la pompe de charge) et de $2,87\text{mA}$ crête à la commutation, pour une moyenne de $50\mu\text{A}$ ($40\mu\text{A}$ pour la pompe de charge).

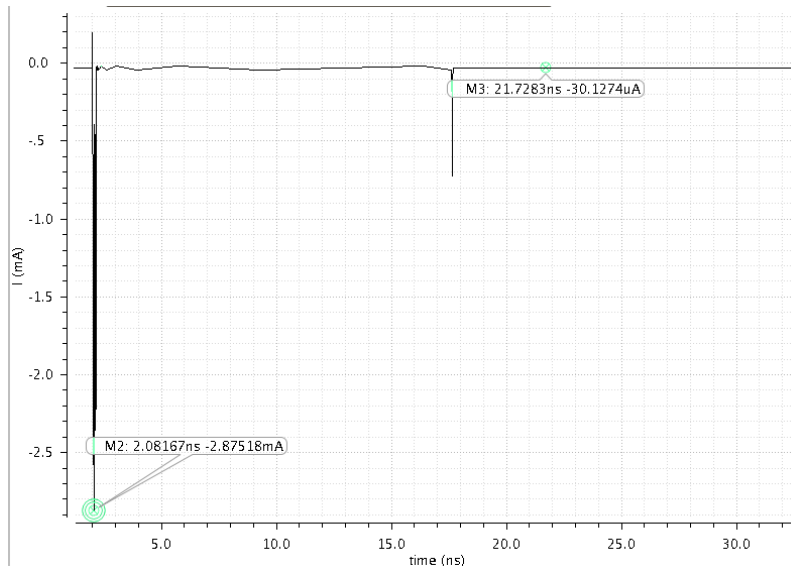


Figure 147 - Consommation en pic et en standby

3.4.2. Pompe de charge

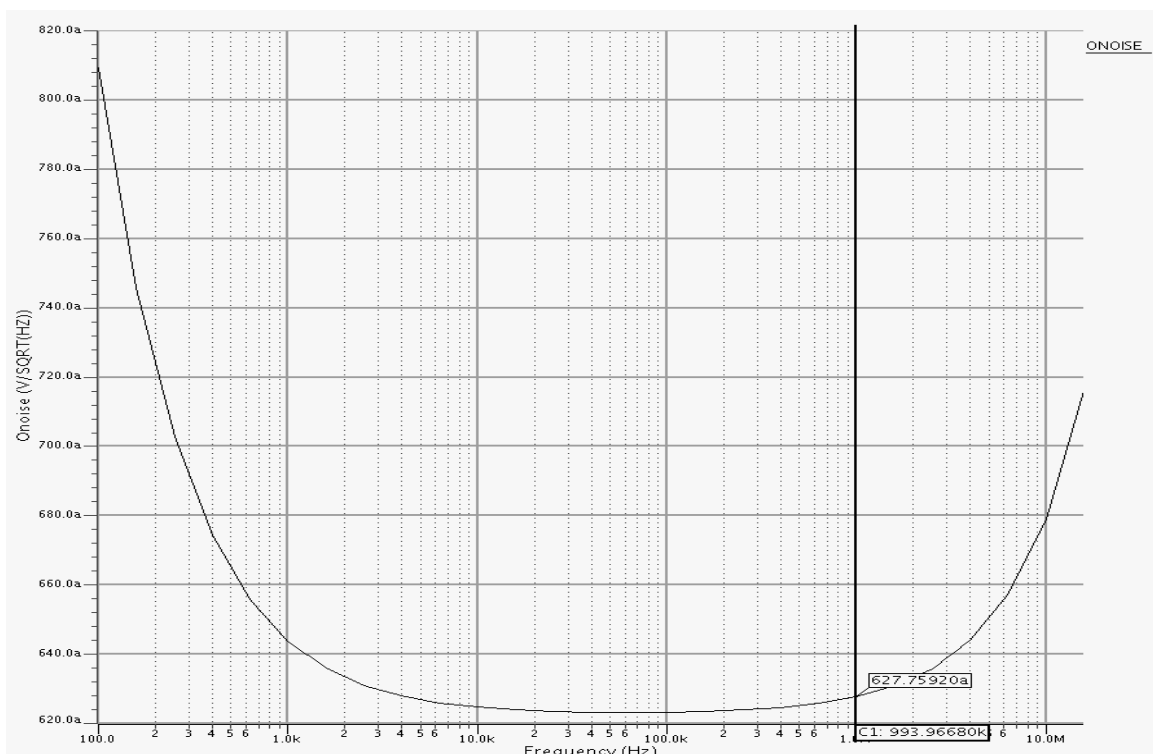


Figure 148 - Bruit de phase simulé de la pompe de charge (à 0,5V)

L'injection de charges dans le filtre de boucle a été limitée à 15 femto-coulomb par comparaison grâce à l'équilibrage des signaux de commande. Le courant de bruit simulé est inférieur à 1fA/√Hz sur toute la bande passante de la PLL à 0,5V (cf. Figure 148). Quand on va jusqu'aux rails d'alimentation, le bruit augmente pour des valeurs de capacités allant de 17fF à 1V et 13fF à 0V. En prenant une valeur de 12fF, le bruit de phase ramené à l'entrée du détecteur de phase est inférieur à

-135dBc/Hz. La DSP du bruit est égale au carré du courant divisé par le gain du détecteur (14). Le bruit de phase équivalent en sortie est égal à cette DSP de bruit multiplié par la fonction de transfert vers la sortie au carré. La forme du bruit simulé (Figure 148) correspond bien à la théorie (cf. Figure 149).

$$DSP_{CP_NOISE} = (I_{CP_NOISE} * \pi / I_{CP})^2 \quad (14)$$

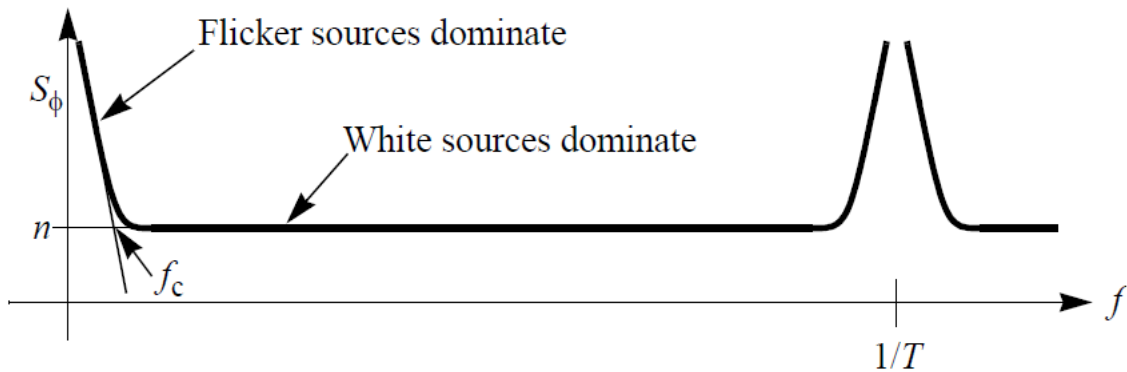


Figure 149 - Bruit de phase modélisé par K.Kundert

3.4.3. Filtre de boucle

Le filtre de boucle passif est supposé idéal et n'a donc pas été simulé. Son bruit thermique a déjà été calculé par la formule correspondante.

3.5. Conclusion

Le délai d'activation minimum de 50ps n'a pas pu être respecté bien qu'il semblait atteignable au moment de la définition des spécifications de la PLL. Pour implémenter la PLL entière en respectant pleinement la spécification de bruit de phase dans la bande, il faut soit que le courant de pompe de charge soit augmenté proportionnellement, soit que l'architecture de comparateur de phases à 3 états basé sur des portes logiques soit modifiée ou changée par une autre architecture.

4. Buffer 50Ω

4.1. Introduction

Les buffers 50Ω permettent d'adapter les signaux hautes fréquences en vue du test (Sortie sur un connecteur BNC ou SMA ou test sous pointes vers des appareils de type VNA, Analyseur de

spectre ou mesureur de bruit de phase). Ils ont été, en particulier, intégrés dans la sortie des VCRO qui ont été testés au chapitre précédent.

4.2. Spécifications

Leur impédance de sortie doit être de 50Ω et leur impédance d'entrée assez élevée pour être pilotée par un des buffers de phase des VCRO (transistors Nmos de 1μ et Pmos de 1.6μ m de large).

4.3. Implémentation

Le buffer 50Ω , présenté sur la Figure 150, comporte plusieurs étages (1μ m, $4,25\mu$ m) pour convertir son impédance d'entrée assez élevée vers l'étage de sortie qui a un r_{dsON} (résistance passante) de 29Ω (Nmos et Pmos), en offrant un bon compromis entre résistance de sortie et charge capacitive. Une résistance de 21Ω ajoutée en série à la sortie permet d'avoir une impédance de 50Ω au total. Cette architecture très simple permet de s'affranchir de sources de courant et offre une faible consommation. Une capacité de découplage doit être ajoutée en externe pour les mesures.

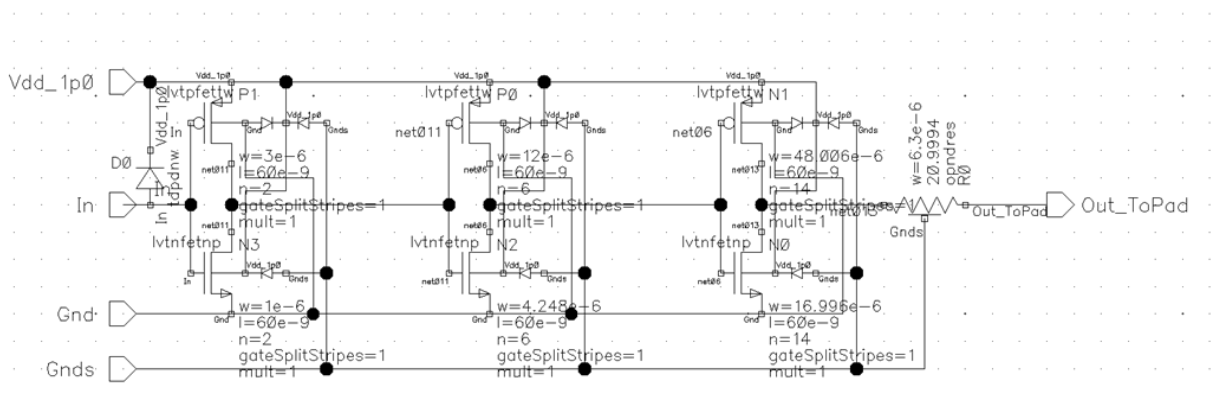


Figure 150 - Schéma du buffer 50 ohm

4.4. Layout

Le dessin des masques de ce buffer 50Ω est présenté sur la Figure 151, où on peut observer les 3 étages (1er à gauche) et la résistance de sortie à droite. Les transistors sont tous dans un caisson d'isolement (triple well) pour éviter des injections de bruit par le substrat dans les VCRO. Toutes les alimentations se font par le dessus.

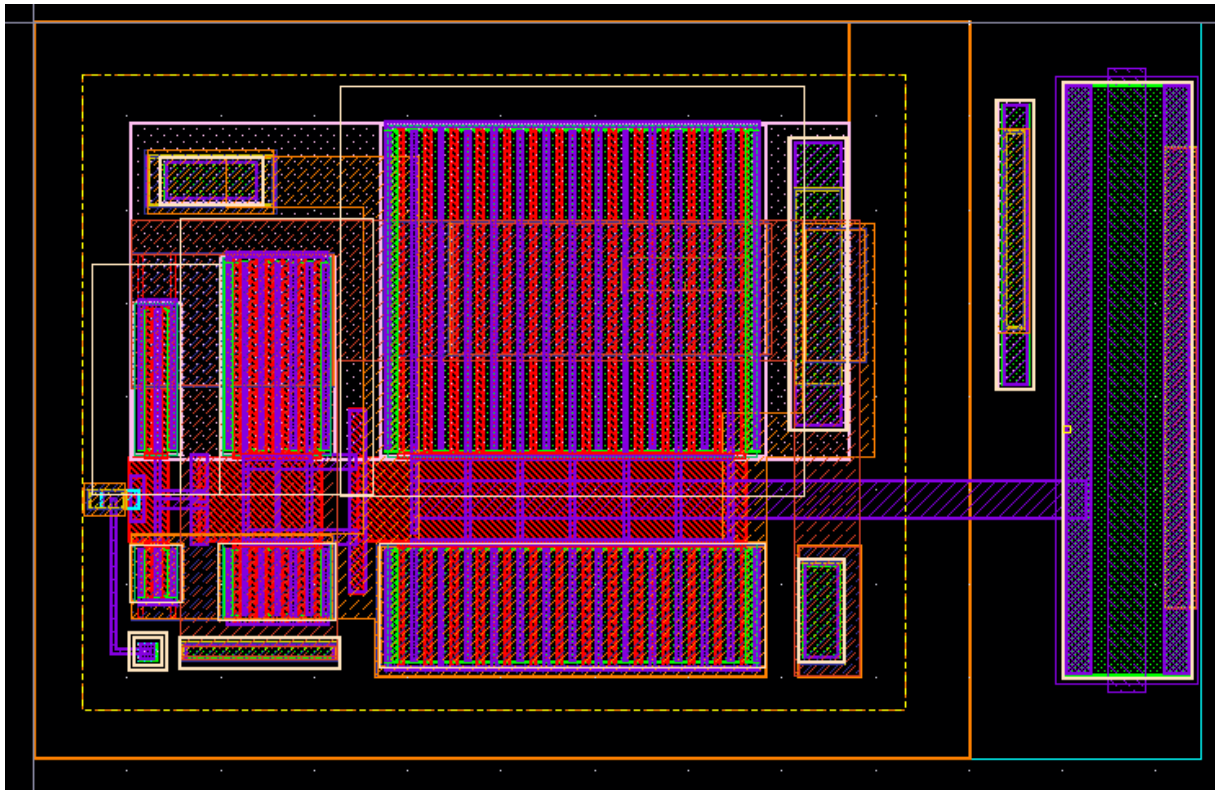


Figure 151 - Layout du buffer 50ohm

4.5. Simulation

Des simulations en charge ont permis de vérifier que ce buffer fonctionne avec la capacité parasite du pad RF (280fF) et une capacité parasite externe de 1pF (défaut câble 50Ω). La consommation totale du buffer chargé par une ligne 50Ω à 2,5GHz est de 4,2mA pour une excursion de plus de 500mV (+/-250mV autour de 0) (voir Figure 152). La courbe 'Out' est mesurée aux bornes d'une résistance de 50Ω après une capacité de découplage (série) de 1nF.

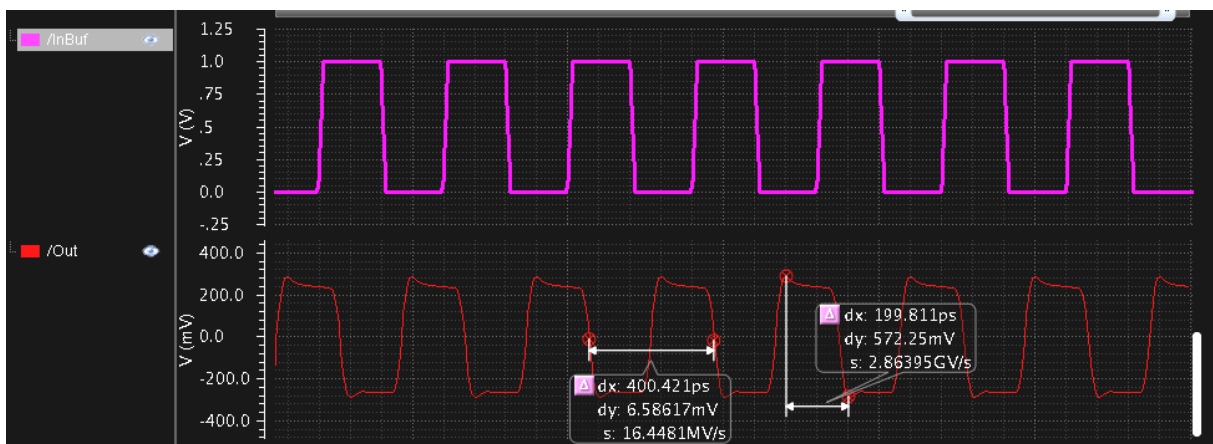


Figure 152 - Simulation du buffer 50ohm à 2,5GHz

Le buffer 50Ω était la pièce manquante pour réaliser le circuit de test intégrant les 4 VCRO. Malgré une architecture peu commune, ses performances sont excellentes : faible variabilité grâce à des transistors très grands, idem pour le rapport cyclique et sa consommation est très faible (4,2mA).

5.PLL

5.1. Introduction

Les différents blocs ayant été conçus et simulés séparément, nous pouvons étudier le comportement de la PLL finale. Les résultats de simulation au niveau transistor, sont présentés au paragraphe suivant.

5.2. Simulation

En utilisant tous les blocs au niveau transistor décrits dans ce chapitre, une simulation de PLL entière a été effectuée (voir Figure 153). La tension de départ du Vtune (en vert) est hors bande au démarrage de la PLL (fuite de la CP), il s'agit donc d'un accrochage de démarrage. On peut y observer plusieurs glissements de périodes jusqu'à la rentrée dans la bande (<2,5GHz). Le VCRO utilisé pour cette simulation a été l'un des premiers développé mais non présenté; il dispose de 5 cellules délai (inverseur simple).

Le temps d'accrochage au démarrage est de 75ns et celui entre bande est inférieur à 50ns. Cette rapidité est favorable à la réduction de la consommation.

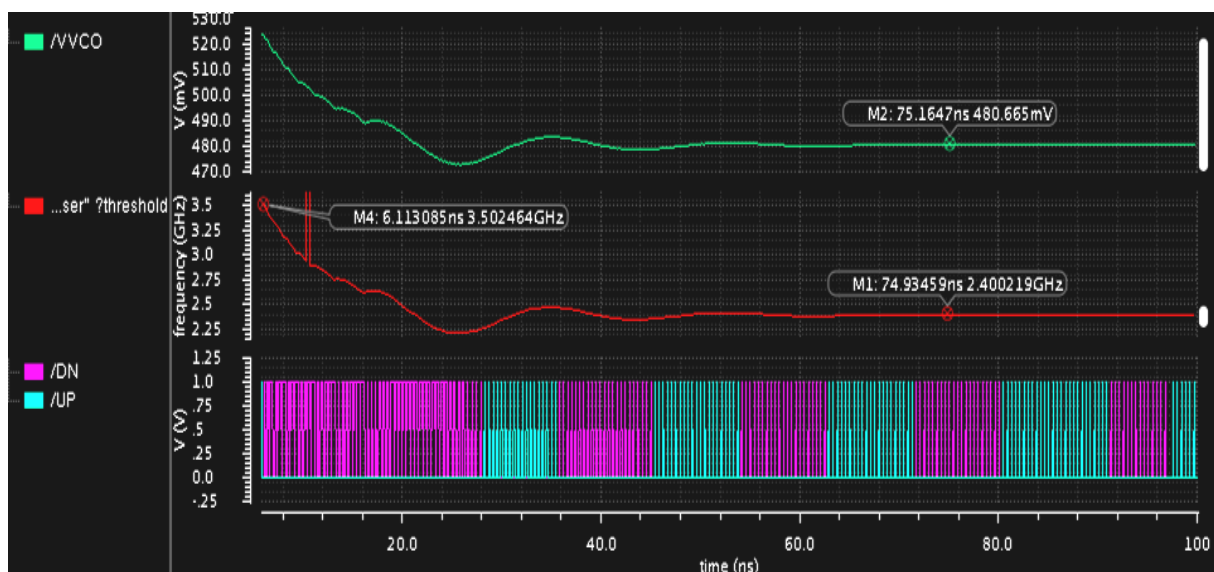


Figure 153 - Simulation accrochage PLL

5.3. Conclusion

Malheureusement, nous n'avons pas réalisé le dessin des masques de l'ensemble de la PLL par manque de temps, et n'avons donc pas effectué ni simulation post-layout, ni test de la PLL complète.

La consommation totale de la PLL a été estimée, à titre d'exemple, en sommant la consommation de chaque bloc :

- VCRO : 800 μ A
- Diviseur : 50 μ A
- PFD+CP : 50 μ A

Soit au total une consommation de 900 μ A sans les buffers de sortie.

La surface totale de la PLL a été estimée, de la même manière, en sommant la surface approximative des blocs dessinés et non-dessinés (par leur nombre et taille de composants) :

- VCRO : 40 μ m x 20 μ m
- Diviseur + PFD + CP : 10 μ m x 40 μ m
- Filtre passe bas 14 μ m x 14 μ m (C1) + 4 μ m x 4 μ m (C2) + 40 μ m x 5 μ m

Soit au total une surface approximative de 1600 μ m².

6. Perspectives

6.1. Introduction

Nous avons montré que moins les phases du VCRO ne possédaient de mismatch, moins la PLL finale ne produisait de spurious. Au lieu de réaliser un circuit de calibration du mismatch, nous avons conçu un VCRO qui en possédait le moins possible. Nous avons ainsi repensé la topologie de base des inverseurs en nous inspirant des cellules différentielles utilisées au chapitre III sur la réalisation des VCRO. En utilisant avantageusement les possibilités offertes par la technologie FDSOI, notamment la polarisation de la grille arrière, nous avons revisité la topologie des cellules numériques complémentaires en réalisant des inverseurs totalement symétriques.

Pour illustrer ces nouvelles cellules numériques complémentaires entièrement symétriques et leur comportement, nous rappelons sur la Figure 154, le schéma électrique d'un inverseur classique en technologie CMOS Bulk standard. La Figure 155 présente 4 possibilités de cette implémentation avec 4 polarisations différentes des grilles arrières, en technologie FDSOI.

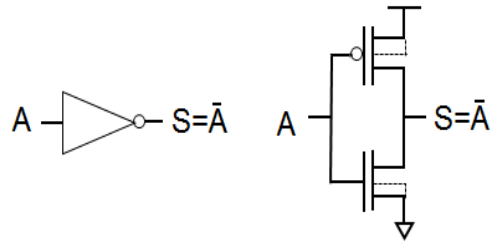


Figure 154 – Implémentation d'un inverseur en technologie CMOS bulk

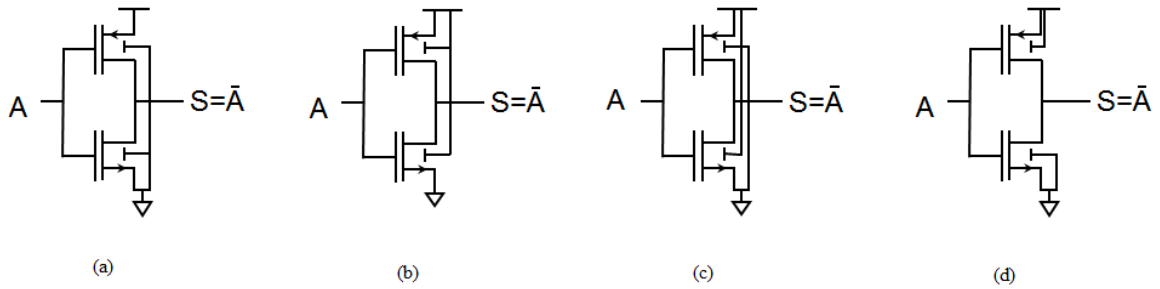


Figure 155 – Differentes implémentations d'inverseurs classiques en technologie FDSOI

Nous avons comparé les temps de montée et de descente (t_{pLH} et t_{pHL}) pour ces 4 configurations. Les valeurs moyennes de ces temps ont été déterminées par des simulations MonteCarlo et données sur le Tableau 15.

Tableau 15 - Simulation MonteCarlo des temps de retard

	(a)	(b)	(c)	(d)
t_{pHL} (ps)	6.25	6.89	6.87	6.23
t_{pLH} (ps)	6.48	6.09	6.52	6.10
Ratio (%)	3.5	13.1	5.31	2.11

Ces résultats montrent que la configuration (d) est la meilleure avec une différence de 2% entre t_{pLH} et t_{pHL} (valeur la plus faible). Nous pouvons noter que cette configuration correspond à celle d'un inverseur CMOS Bulk classique (Bulk du transistor Nmos polarisé a Vss et celui du Pmos à Vdd). Dans le paragraphe suivant, nous comparerons les performances de notre nouvelle technologie à cette configuration.

6.2. Logique complémentaire

La Figure 156 décrit l'implémentation d'un inverseur complémentaire où les deux entrées doivent être complémentaires, ainsi que les 2 sorties, en réalisant une auto-polarisation des transistors UTBB-FDSOI.

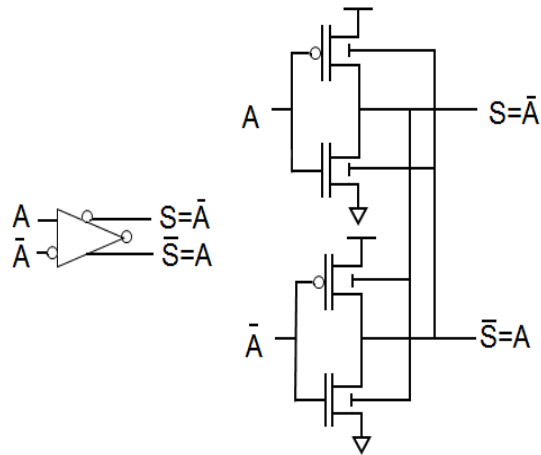


Figure 156 – Implémentation d'un inverseur complémentaire en technologie FDSOI avec auto-polarisation de la grille arrière

L'idée principale est très simple. Lors d'une transaction, les deux inverseurs commutent symétriquement l'un de l'autre. Ainsi le transistor Nmos, le plus rapide, va accélérer le transistor Pmos de l'autre inverseur puisque la sortie du premier commande la grille arrière du second. Le même raisonnement s'applique si c'est le transistor Pmos qui conduit mais de manière symétrique. Il y a donc une symétrisation des 2 inverseurs. Les sorties complémentaires vont donc se croiser à $V_{dd}/2$ et par symétrie on aura $t_{p_{HL}} \approx t_{p_{LH}}$. La figure 155 illustre cette simulation transitoire. Ce concept a été validé par des simulations MonteCarlo, non présentées dans ce mémoire, qui ont donné une valeur moyenne de la gaussienne égale à 500mV ($=V_{dd}/2$) et un écart type de $\sigma=2,7\text{mV}$ [65]. Des simulations MonteCarlo similaires ont été réalisées à partir de la configuration (d) (cf. Figure 157) et ont montré un écart type de $\sigma=5\text{mV}$ [65]. Nous avons ainsi quasiment divisé par 2 cet écart type.

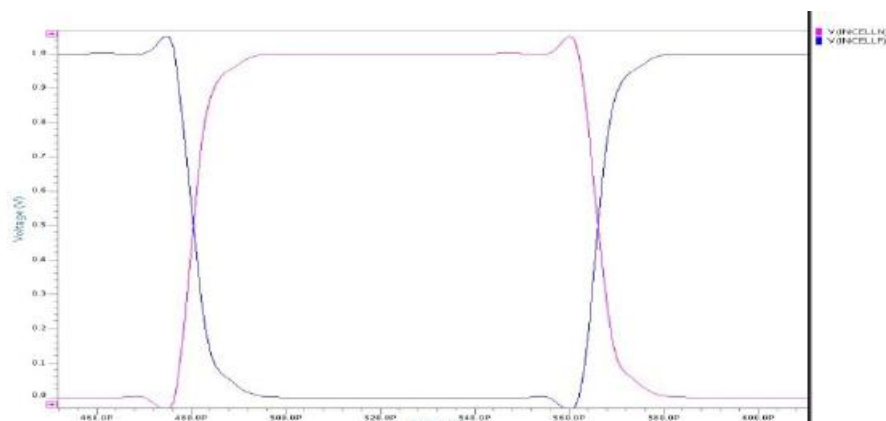


Figure 157 - Simulation transitoire d'un inverseur en logique complémentaire (nouvelle topologie)

6.3. Oscillateur en anneau en logique complémentaire

Cette nouvelle topologie (inverseur complémentaire) nous offre 2 avantages très importants pour la réalisation d'un oscillateur en anneau. Le premier concerne le rapport cyclique qui sera très proche de 50% du fait que $t_{pHL} = t_{pLH}$. Deuxièmement, cette topologie, comme dans le cas des cellules différentielles permet de réaliser un oscillateur avec un nombre pair d'inverseurs, et donc en équilibrant les phases ce qui est important pour une PLL à commutation ou resynchronisation de phases. Pour vérifier cette hypothèse, nous avons réalisé des simulations MonteCarlo sur un oscillateur en anneau comportant 8 cellules complémentaires. Ces analyses nous ont permis de mesurer l'écart de phase sur les 8 sorties du VCRO décalées normalement de 45° . Les résultats sont présentés sur la Figure 158.

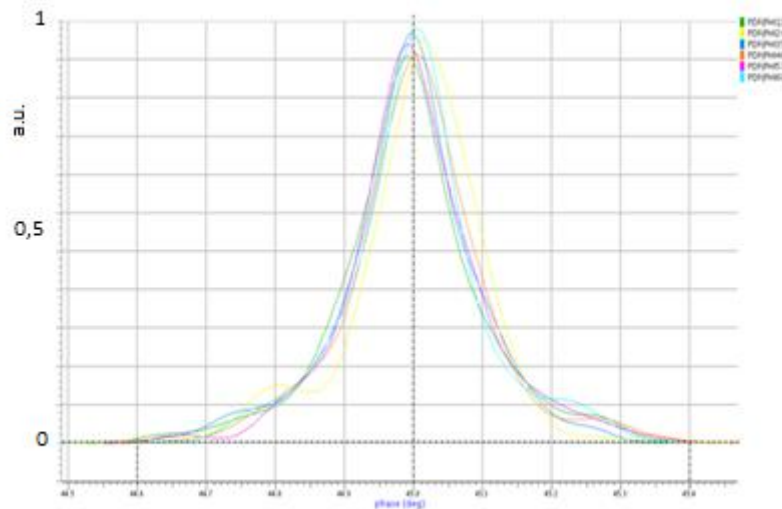


Figure 158 - Simulation MonteCarlo des phases

On vérifie que les inverseurs ont bien le même comportement, les courbes étant très proches les unes de autres. La valeur moyenne de chaque phase est de 45° par rapport à la précédente et l'écart type est de $0,13^\circ$ soit 1,8%.

Cette topologie permet de réaliser un VCO en quadrature (QVCO) avec une très faible dispersion des phases, ce qui est fondamental pour la conception d'un récepteur à réjection d'image.

7. Conclusion

Dans ce chapitre nous avons présenté l'implémentation au niveau transistor (en technologie FDSOI 28nm), les blocs permettant de concevoir une PLL basée sur les VCRO développés

précédemment. Chaque bloc a été validé par des simulations Spice afin d'évaluer ses performances fonctionnelles, sa consommation et son bruit.

Par manque de temps, nous n'avons pas réalisé le dessin des masques des blocs de la PLL décrits dans ce chapitre. Toutefois, nous les avons optimisés pour qu'ils puissent travailler au double de la fréquence nominale et ainsi s'assurer qu'ils fonctionneront une fois fabriqués sans nécessiter de correction. Finalement, nous avons validé le fonctionnement de la PLL au niveau transistor en simulant un changement de fréquence.

Nous proposons que tous les blocs de la PLL conçue soient fabriqués sur la même puce avec un régulateur de tension faible bruit (type LDO) pour réussir à mesurer les performances simulées. Le bruit de phase des VCRO aurait dû être à $-90\text{dBc/Hz}@1\text{MHz}$ mais les mesures ont été rendues difficiles sans la présence d'une PLL interne. Les niveaux de spurious sont estimés à -60dBc sans calibration et ont été impossibles à mesurer sans l'effet de la PLL.

Nous avons terminé ce chapitre par des propositions d'amélioration concernant l'auto-calibration des inverseurs du VCRO par une auto-polarisation de la grille arrière, qui a fait l'objet d'un dépôt de brevet. Les premiers résultats de simulation sont très prometteurs et cette technique peut être étendue à toutes les cellules numériques et à certains blocs analogiques comme des paires différentielles ou des miroirs de courant.

Conclusion Générale

Le nombre des données échangées par les individus sur la planète est exorbitant, et les données générées, enregistrées, échangées, traitées et utilisées automatiquement pour prendre des décisions doit encore évoluer pour nous décharger de ce poids. C'est ce que l'internet des objets propose mais son objectif est encore loin d'être atteint. La faute à une mauvaise communication entre les services de collection de données et les services d'information aux personnes ? Sans doute, mais pas uniquement. Le secret du développement en masse de l'IOT réside en grande partie dans le prix de revient et dans l'autonomie/la consommation que les objets communicants ont à proposer. Les premiers produits Bluetooth smart arrivent sur le marché mais restent cantonnés à des usages très limités, du tracker sportif/santé au porte clé de localisation de trousseau. Quand les produits s'aventurent à être connectés en permanence (24/24,7/7) ils sont handicapés par une grosse batterie ou une prise murale. Le standard Bluetooth Smart a participé au déploiement de ces objets, en diminuant la taille de leur batterie et donc leur coût mais le chemin reste long pour une intégrabilité totale (dans tous les objets de notre quotidien).

Diminuer le coût de revient et augmenter l'autonomie de ces objets passent par l'optimisation des circuits actuels et l'utilisation d'architectures particulières adaptées à des technologies agressives. C'est dans cet esprit que nous avons mené ce travail de thèse, en revisitant la topologie des PLL à base d'inverseurs en anneau et de diviseurs à commutation de phases et en utilisant la technologie FDSOI. Nous avons ainsi démontré la faisabilité de la conception d'une synthèse de fréquences ultra basse consommation. Nous avons fabriqué plusieurs VCRO dans une technologie de pointe encore réservée aux circuits numériques et mesurer, certes avec difficultés, des performances honorables ($-85\text{dBc}@1\text{MHz}$ @ $2,45\text{GHz}$) tout en occupant que très peu de surface, avec une consommation très faible.

Pour conclure, nous pensons que cette architecture mérite encore d'être étudiée et des mesures d'un circuit de PLL fractionnaire complète ne peuvent que lui rendre justice. Néanmoins nous conseillons de prendre un soin très particulier, au niveau du dessin des masques, à éviter d'insérer un mismatch physique tel qu'une différence de longueur de ligne de propagation des phases. Une calibration supplémentaire améliorera bien sûr le niveau des spurious, estimé à environ -60dBc à la fréquence du canal adjacent sans calibration.

En outre, il est d'usage dans une conclusion générale de proposer des perspectives à ce travail. Au-delà de l'amélioration des mesures du bruit de phases et de l'implémentation complète de la PLL,

nous avons présenté, dans le chapitre IV, une nouvelle topologie de cellules numériques complémentaires. Nous avons appliqué cette implémentation pour réaliser pour réaliser un oscillateur en anneau en utilisant cette technique d'auto-calibration des grilles arrières. Ce procédé a fait l'objet d'un dépôt de brevet et les premiers résultats de simulation sont extrêmement encourageants. Une nouvelle thèse a démarré sur ce sujet, au sein du laboratoire.

Publications

Revue internationale

A. Fonseca, E. de Foucauld, P. Lorenzini & G. Jacquemod, «Low power 28nm FDSOI 2.45 GHz PLL», *Journal of Low Power Electronics*, vol. 10, n° 1, 2014, p. 149-162

G. Jacquemod, A. Fonseca, E. de Foucauld, Y. Leduc & P. Lorenzini, «2.45 GHz 0.8 mW voltage-controlled ring oscillator (VCRO) in 28 nm Fully Depleted Silicon On Insulator (FDSOI) Technology», *Frontiers of Materials Science*, vol. 9, Issue 2, p. 156-162, 2015

Conférences internationales

A. Fonseca, E. de Foucauld, P. Lorenzini & G. Jacquemod, «CMOS technology beyond 22 nm», ICSS, Las Vegas, 2013, p. 152-153, papier invité

A. Fonseca, E. de Foucauld, P. Lorenzini & G. Jacquemod, «Process variation compensation for PLL on FDSOI 28nm», VARI/PATMOS, Karlsruhe, 2013

G. Jacquemod, A. Fonseca, Y. Leduc, E. de Foucauld & P. Lorenzini, «Analog Design in FDSOI 28 nm technology and beyond», CISIS, 3rd Annual World Congress of Emerging InfoTech, Dalian, 2014, p. 112, papier invité

A. Fonseca, E. de Foucauld, P. Lorenzini & G. Jacquemod, «Fractional phase divider PLL phase noise and spurious modeling», VARI/PATMOS, Palma de Mallorca, 2014

A. Fonseca, G. Jacquemod, Y. Leduc, E. de Foucauld & P. Lorenzini, «VCO Design in SOI technologies», NEWCAS, Special Session «*Frequency synthesis – New designs, new technologies*», Trois Rivières, 2014, p. 420-423

G. Jacquemod, A. Fonseca, E. de Foucauld, Y. Leduc & P. Lorenzini, «2.45 GHz 0.8 mW VCRO in 28nm FDSOI Technology», ICSS, Hong Kong, 2014, p. 74-75, papier invité

G. Jacquemod, Y. Leduc P. Lorenzini, E. de Foucauld, & A. Fonseca, «Self-calibration of analog and mixed cells using back-gate auto-biasing transistor in 28nm FDSOI Technology and beyond», *Nanotechnology and Materials Science*, Dubai, 2015, p. 97, papier invité

Conférences nationales

A. Fonseca, E. de Foucauld, P. Lorenzini & G. Jacquemod, «PLL fractionnelle large bande sans sigma-delta en FDSOI 28nm : Suppression des intermodulations par calibration sur la grille arrière», JNRDM, Grenoble, 2013

Brevet déposé

G. Jacquemod, E. de Foucauld, Y. Leduc, A. Fonseca & P. Lorenzini, «Procédé et dispositif d'auto-calibration de circuits multi-grilles», N° 1553096, Réf. 69726FR FLO/MBA, 10 avril 2015

Références

- [1] « Bluetooth Low Energy Technology | Bluetooth Technology Website » [En ligne], Disponible sur: <http://www.bluetooth.com/Pages/low-energy-tech-info.aspx>, [Consulté le: 09-sept-2014]
- [2] « THIS IS ANT - the Wireless Sensor Network Solution » [En ligne], Disponible sur: <http://www.thisisant.com/>, [Consulté le: 09-sept-2014]
- [3] « ZigBee Alliance Home Page » [En ligne], Disponible sur: <http://www.zigbee.org/Home.aspx>, [Consulté le: 09-sept-2014]
- [4] « Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018 », Cisco [En ligne], Disponible sur: http://cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html, [Consulté le: 09-sept-2014]
- [5] « BLE112 Bluetooth Smart Module – Bluegiga » [En ligne], Disponible sur: <https://www.bluegiga.com/en-US/products/bluetooth-4.0-modules/ble112-bluetooth-smart-module/technical-specs/>, [Consulté le: 09-sept-2014]
- [6] « nRF51822 - Bluetooth low energy and 2.4GHz proprietary SoC », Nordic Semiconductor, 2014 [En ligne], Disponible sur: <https://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822>, [Consulté le: 09-sept-2014]
- [7] J Decuir, « Bluetooth 4.0: Low Energy », CSR, 2010
- [8] « Specification | Adopted Documents | Bluetooth Technology Special Interest Group », 09-sept-2014 [En ligne], Disponible sur: <https://www.bluetooth.org/en-us/specification/adopted-specifications>, [Consulté le: 09-sept-2014]
- [9] S Buettrich, « Calcul du bilan de liaison radio », sept-2005
- [10] A Balankutty, S-A Yu, Y Feng, et P Kinget, « A 0.6V 32.5mW Highly Integrated Receiver for 2.4GHz ISM-Band Applications », in, *Solid-State Circuits Conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International*, 2008, p. 366-620
- [11] G. Moore, « Cramming more components onto integrated circuits », *Electronics Magazine*, vol. 38, n° 8, 1965
- [12] I Paul, « The end of Moore’s Law is on the horizon, says AMD », *PCWorld Processors Blog*, 03-avr-2013
- [13] M Kaku, « Tweaking Moores’s law: Computers of the post-silicon era », 2013
- [14] K Mistry, « Transistor Scaling: The Age of Innovation », 2014
- [15] Intel Labs, « Connecting the future : It’s a wireless world », présenté à, Intel Developer Forum (IDF), 2012
- [16] I Bell, « 65nm Process Technology – Nearest Future of Intel Semiconductor Technologies », présenté à, IDF, Russie, 2004
- [17] A Asenov, « Device-Circuit Interplay in the Simulation of Statistical CMOS Variability », présenté à, VARI, Nice, 2012
- [18] M. Jurczak, N. Collaert, A. Veloso, T. Hoffman & S. Biesemans, « Review of FINFET technology », présenté à, SOI Conference, 2009
- [19] X Huang, « Sub 50-nm FinFET: PMOS », in, *International Electron Devices Meeting Technical Digest*, 1999, p. 67
- [20] A Narayanan, « Battle of Fins and Boxes », *Mentor IC Design Blog*, 07-déc-2012
- [21] P. Flatresse, R. Wilson, « SOC Variability Reduction: The UTBB FD-SOI Way », présenté à, VARI, Darmstadt, 2013
- [22] « FD-SOI – A Look at Consortium Benchmarking Results | Advanced Substrate News », *Advance Substrate News*, 02-mai-2012
- [23] A. Makosiej, O. Thomas, A. Amara, A. Vladimirescu, « CMOS SRAM scaling limits under optimum stability constraints », in, *IEEE ISCAS*, Beijing, 2013, p. 1460-1463

- [24] M.J.M. Pelgrom et al., « Matching properties of MOS transistors », *IEEE J. Solid-State-Circuits*, vol. 24, n° 5, p. 1433-1439, oct. 1989
- [25] F. Andrieu, O. Weber, S. Baudot, C. Fenouillet-Beranger, O. Rozeau, J. Mazurier, P. Perreau, J. Eymery, O. Faynot, « Fully Depleted Silicon-On-Insulator with back gate bias and strain for low power and high performance applications », présenté à, IEEE ICICDT, Grenoble, 2010
- [26] O. Weber, O. Faynot, F. Andrieu, C. Buj-Dufouquet, C. Allain, P. Scheiblin, J. Foucher, N. Daval, D. Lafond, L. Tosti, L. Brevard, O. Rozeau, C. Fenouillet-Beranger, M. Marin, F. Bœuf, D. Delprat, K. Bourdelle, B.Y. Nguyen et S. Deleonibus, « High Immunity to Threshold Voltage Variability in Undoped Ultra-Thin FDSOI MOSFETs and its Physical Understanding », présenté à, IEEE IEDM, San Francisco, 2008
- [27] T. Skotnicki, « Low Power and High Speed at Low Voltage for Mobile Applications », présenté à, IEEE IEDM, San Francisco, 2010
- [28] C Barrett, « Fractional / Integer-N PLL Basics » Texas Instruments, 1999
- [29] Semtech, « Digital Frequency Locked Loop », Technical Note 8000.09
- [30] A Djemouai, M. Sawan, et M Slamani, « New frequency-locked loop based on CMOS frequency-to-voltage converter: design and implementation », *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.*, vol. 48, n° 5, p. 441-449, mai 2001
- [31] « Lecture 15: Delay-Locked Loops (DLLs) », présenté à, ECEN620: Network Theory Broadband Circuit Design Fall 2014, Sam Palermo
- [32] C-F Liang et K-J Hsiao, « An injection-locked ring PLL with self-aligned injection window », in, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, 2011, p. 90-92
- [33] « Phase-locked loop », *Wikipedia, the free encyclopedia* 14-sept-2014
- [34] L Xiu, W Li, J Meiners, et R Padakanti, « A novel all-digital PLL with software adaptive filter », *IEEE J. Solid-State Circuits*, vol. 39, n° 3, p. 476-483, mars 2004
- [35] W Grollitsch, R Nonis, et N Da Dalt, « A 1.4psrms-period-jitter TDC-less fractional-N digital PLL with digitally controlled ring oscillator in 65nm CMOS », in, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, 2010, p. 478 - 479
- [36] F Opteynde, « A 40nm CMOS all-digital fractional-N synthesizer without requiring calibration », in, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, 2012, p. 346-347
- [37] M S-W Chen, D Su, et S Mehta, « A calibration-free 800MHz fractional-N digital PLL with embedded TDC », in, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, 2010, p. 472-473
- [38] R B Staszewski, D Leipold, K Muhammad, et P T Balsara, « Digitally controlled oscillator (DCO)-based architecture for RF frequency synthesis in a deep-submicrometer CMOS Process », *IEEE Trans. Circuits Syst. II Analog Digit. Signal Process.*, vol. 50, n° 11, p. 815-828, 2003
- [39] J Masuch et M Delgado-Restituto, « A 350 μ W 2.3 GHz integer-N frequency synthesizer for body area network applications », in, *2011 IEEE 11th Topical Meeting on Silicon Monolithic Integrated Circuits in RF Systems (SiRF)*, 2011, p. 105-108
- [40] J Liu, S Jeon, T-K Jang, D Kim, J Kim, J Park, et H Park, « A 0.8V, sub-mW, varactor-tuning ring-oscillator-based clock generator in 32nm CMOS », in, *Solid State Circuits Conference (A-SSCC), 2011 IEEE Asian*, 2011, p. 337 -340
- [41] H Xiao et R Schaumann, « A low-voltage low-power CMOS 5-GHz oscillator based on active inductors », in, *9th International Conference on Electronics, Circuits and Systems, 2002*, 2002, vol. 1, p. 231-234 vol.1
- [42] J Wilson, « Planet Analog - Articles - Jitter attenuation 101: How do you pick the right PLL bandwidth? », 04-oct-2010
- [43] M Stork et P Kaspar, « Fractional phase-locked loop frequency synthesizer », in, *International Symposium on Signals, Circuits and Systems, 2003. SCS 2003*, 2003, vol. 1, p. 129- 132 vol.1
- [44] P Larsson, « Fractional frequency divider », 6157694, 05-déc-2000

- [45] K J Wang, A Swaminathan, et I Galton, « Spurious Tone Suppression Techniques Applied to a Wide-Bandwidth 2.4 GHz Fractional-N PLL », *IEEE J. Solid-State Circuits*, vol. 43, n° 12, p. 2787-2797, Dec.
- [46] A Swaminathan, K J Wang, et I Galton, « A Wide-Bandwidth 2.4 GHz ISM Band Fractional-N PLL With Adaptive Phase Noise Cancellation », *IEEE J. Solid-State Circuits*, vol. 42, n° 12, p. 2639-2650, 2007
- [47] S Pamarti, L Jansson, et I Galton, « A wideband 2.4-GHz delta-sigma fractional-NPLL with 1-Mb/s in-loop modulation », *IEEE J. Solid-State Circuits*, vol. 39, n° 1, p. 49-62, Jan.
- [48] S Levantino, D Tasca, G Marzin, M Zanuso, C Samori, et A L Lacaïta, « A wideband fractional-N PLL with suppressed charge-pump noise and automatic loop filter calibration », in, *2012 IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, 2012, p. 177-180
- [49] A S Kamath et B Chattopadhyay, « A 13MHz input, 480MHz output Fractional Phase Lock Loop with 1MHz bandwidth », in, *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, p. 501-504
- [50] C-H Park, O Kim, et B Kim, « A 1.8-GHz self-calibrated phase-locked loop with precise I/Q matching », *IEEE J. Solid-State Circuits*, vol. 36, n° 5, p. 777-783, 2001
- [51] R Nonis, W Grollitsch, T Santa, D Cherniak, et N Da Dalt, « A 2.4psrms-jitter digital PLL with Multi-Output Bang-Bang Phase Detector and phase-interpolator-based fractional-N divider », in, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*, 2013, p. 356-357
- [52] E Tatschl-Unterberger, S Cyrusian, et M Ruegg, « A 2.5GHz phase-switching PLL using a supply controlled 2-delay-stage 10GHz ring oscillator for improved jitter/mismatch », in, *IEEE International Symposium on Circuits and Systems, 2005. ISCAS 2005*, 2005, p. 5453-5456 Vol. 6
- [53] E Bilhan, F Ying, J M Meiners, et L Xiu, « Spur-Free Fractional-N PLL Utilizing Precision Frequency and Phase Selection », in, *2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, 2006, p. 139-142
- [54] D G Martin, « Fractional PLL employing a phase-selection feedback counter », 6526374, 25-févr-2003
- [55] H Mair et L Xiu, « An architecture of high-performance frequency and phase synthesis », *IEEE J. Solid-State Circuits*, vol. 35, n° 6, p. 835-846, 2000
- [56] W S Jenningscheck, « Clock adapter using a phase locked loop configured as a frequency multiplier with a non-integer feedback divider », 5059924, 22-oct-1991
- [57] J Masuch et M Delgado-Restituto, « A 1.1-mW-RX -dBm Sensitivity CMOS Transceiver for Bluetooth Low Energy », *IEEE Trans. Microw. Theory Tech.*, vol. 61, n° 4, p. 1660-1673, avr. 2013
- [58] T-W Liao, J-R Su, et C-C Hung, « Ring-VCO based low noise and low spur frequency synthesizer », in, *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2013, p. 1861-1864
- [59] C Sasan, M Soenke, et S Andreas, « Frequency divider circuit for frequency synthesizer », DE19844953, 05-janv-2000
- [60] F Bredin, B Gabillard, et F A Roger Meunier, « Programmable non-integer fractional divider », US6748408 B1, juin-2004
- [61] D-W Jee, Y Suh, H-J Park, et J-Y Sim, « A 0.1-fref BW 1GHz fractional-N PLL with FIR-embedded phase-interpolator-based noise filtering », in, *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, 2011, p. 94-96
- [62] M U Erdogan, « 1 to 2n-1 fractional divider circuit with fine fractional resolution », US20120092051 A1, avr-2012
- [63] Dr. Phillip E. Allen, « LECTURE 090 – PLL DESIGN EQUATIONS AND PLL MEASUREMENTS »
- [64] K Kundert, *Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers* www.designers-guide.org/Analysis/PLLnoise.pdf, 2003

Annexes

1. Modèle de VCRO à phases

```
`define NPhases 8 // from [3:100] number of phases

module VCO_Nphases(ain, fosc_out);
parameter real f0 = 2.45e9 from (0.0:inf);           // Output center freq
parameter real fmin = 2.3e9;                       // Minimum allowable output freq
parameter real fmax = 2.6e9;                       // Maximum allowable output freq
parameter real V_f0 = 0.5;                         // Input voltage corresponding to f0
parameter real Kv = 300.0e6 from (0.0:inf);        // VCO linear gain (Hz/V)
parameter real Vhi = 1.0;                          // Output voltage corresponding to '1'
parameter real Vlo = 0.0 from [-inf : Vhi];        // Output voltage corresponding to '0'
parameter real RiseForce = 3 from (1:10);          // Transition force

input ain;
electrical ain;
output [0:NPhases-1] fosc_out;
electrical [0:NPhases-1] fosc_out;
real ViLim, halfPeriod, phaseDelay, transitionTime;
real V_fmax, V_fmin;
integer state, state_inv;
genvar k;

analog begin
  if (V(ain) > V_fmax)          ViLim = V_fmax - V_f0; // Clamp input voltage to limit frequency excursion
  else if (V(ain) < V_fmin)    ViLim = V_fmin - V_f0;
  else                        ViLim = V(ain) - V_f0;

  if (analysis("ic", "dc")) begin // Initialization
    state = 1; // init node at 1
    V_fmax = V_f0 + (fmax - f0)/Kv; // Input corresponding to fmax
    V_fmin = V_f0 - (f0 - fmin)/Kv; // Input corresponding to fmin
    halfPeriod = 0.5/(f0 + Kv * ViLim);
    transitionTime = halfPeriod/RiseForce;
    phaseDelay = 2*halfPeriod/NPhases;
  end

  @(timer(0,halfPeriod)) begin // VCO (update frequency) and dependant variables
    state = !state;
    halfPeriod = 0.5/(f0 + Kv * ViLim);
    transitionTime = halfPeriod/RiseForce;
    phaseDelay = 2*halfPeriod/NPhases;
  end

  for ( k=0 ; k < NPhases ; k = k + 1) // N Phases triggering
    V(fosc_out[k]) <+ transition( state?Vhi:Vlo, k*phaseDelay, transitionTime );
  end
endmodule
`undef NPhases
```


2.Modèle de diviseur entier

```
module Cycle_Counter(Phase_In,Count_OK,Max_Count);

parameter real Td = 20p from (1p:400p);          // Transition delay
parameter real Tt = 20p from (1p:100p);         // Propagation delay
parameter real Vhi = 1.0;                       // Output voltage corresponding to '1'
parameter real Vlo = 0.0 from [-inf : Vhi];     // Output voltage corresponding to '0'
parameter real Vth = 0.5 from (Vlo:Vhi);       // Threshold voltage

input Phase_In, Max_Count;
output Count_OK;
voltage Phase_In, Count_OK, Max_Count;

integer counter, state;

analog begin
    @(initial_step)                               // initialization
        begin
            counter = 0;
            state = 0;
        end
    @(cross(V(Phase_In) - Vth,1))                 // counter
        begin
            counter = counter + 1;
            if(counter >= abs(V(Max_Count))) begin
                counter = 0;
                state = 1;
            end else
                state = 0;
        end
    end
    V(Count_OK) <+ transition(state?Vlo:Vhi, Td, Tt); //output
end
endmodule
```

3.Modèle du multiplexeur à commutation

```
`define N 8 //  
  
module Phaser(Phases,Clk_Counter,Increment,Phase_Out, Phase_N, Sync_Out);  
  
input [0:(N-1)] Phases;  
input Increment, Clk_Counter;  
output Phase_Out;  
voltage [0:(N-1)] Phases;  
voltage Clk_Counter,Increment,Phase_Out;  
  
parameter real Tp = 20p from (1p:400p); // Propagation delay  
parameter real Tt = 20p from (1p:100p); // Transition delay  
parameter real Vhi = 1.0; // Logic '1'  
parameter real Vlo = 0.0 from [-inf : Vhi]; // Logic '0'  
parameter real Vth = 0.5 from (Vlo:Vhi); // Threshold voltage  
  
integer change_flag, select, Sel_Phase;  
genvar k;  
  
analog begin  
    @(cross(V(Clk_Counter) - Vth,1)) // phase change  
        change_flag = 1;  
  
    for (k = 0; k < `N; k = k+1)  
        @(cross(V(Phases[k]) - Vth))  
        if(k==select) begin // enters if selected phase  
            if(change_flag) begin  
                if ((V(Phases[k]) < Vth && V(Increment)<N/2) || (V(Phases[k]) > Vth && V(Increment)>=N/2))  
                    begin  
                        change_flag = 0;  
                        select = (select + V(Increment))%`N;  
                    end  
                end else  
                    Sel_Phase = (V(Phases[k]) > Vth)?1:0;  
            end  
        end  
  
        //output  
        V(Phase_Out) <+ transition(Sel_Phase? Vhi:Vlo, Tp, Tt);  
    end  
endmodule  
`undef N
```

4. Modèle du multiplexeur à resynchronisation

```

`define N 30
module PhaseSync(Phases,Int_Div,Increment,Fdiv_Out,Carry_Out);

input [0:(N-1)] Phases;
input Increment, Int_Div;
output Fdiv_Out, Carry_Out;
voltage [0:(N-1)] Phases;
voltage Int_Div,Increment,Fdiv_Out,Carry_Out;

parameter real Tp = 20p from (1p:400p);          // Propagation delay
parameter real Tt = 20p from (1p:100p);         // Transition delay
parameter real Vhi = 1.0;                        // Logic '1'
parameter real Vlo = 0.0 from [-inf : Vhi];     // Logic '0'
parameter real Vth = 0.5 from (Vlo:Vhi);       // Threshold voltage

integer Carry, PreSync, Sync, SyncPh, Fdiv;
genvar k;

analog begin
    @(initial_step) begin                          // initialization
        Sync = 1;
        if(V(Increment)>0)
            Carry=1;
        else
            Carry=0;
    end
    @(cross(V(Phases[0]) - Vth)) begin
        if(Sync==PreSync) begin
            if(V(Phases[0])>Vth)
                PreSync = V(Int_Div)>Vth?1:0; // Integer Division Synchronise to Phase 0
            end else if((SyncPh>= `N/2 && V(Phases[0]) > Vth) || (SyncPh<`N/2 && V(Phases[0]) < Vth))
                Sync = PreSync; // Autorize synchronisation
            if(Sync==0 && PreSync==1)
                Carry = (SyncPh + V(Increment) >= `N)?1:0; // Integer modulus trigger
        end
        @(cross(V(Fdiv_Out) - Vth,-1)) // next selected phase calculation (phase + jump = next)
        SyncPh = (SyncPh + V(Increment))%`N;
        for (k = 0; k < `N; k = k+1)
            @(cross(V(Phases[k]) - Vth,1))
                if(k==SyncPh)
                    Fdiv = Sync; // Fractional Division Synchronization
    end
    //outputs
    V(Fdiv_Out) <+ transition(Fdiv? Vhi:Vlo, Tp, Tt);
    V(Carry_Out) <+ transition(Carry? Vhi:Vlo, Tp, Tt);

end
endmodule

```

5. Modèle du contrôleur

```
module Carry_Adder(Divide, Carry_In, Integer, Phase_Shift);

parameter real Td = 20p from (1p:400p);           // Transition delay
parameter real Tt = 20p from (1p:100p);          // Transition inertia (rise/fallTime)
parameter real Vth = 0.5;                         // Threshold voltage
parameter real Phases = 30 from (1:100);         // Phase Number

input Divide, Carry_In;
output Integer, Phase_Shift;
voltage Divide, Carry_In, Integer, Phase_Shift;

integer carry, count, shift;
analog begin
    count = floor(V(Divide));
    shift = (V(Divide)-count)*Phases;

    // carry detect
    @(cross(V(Carry_In) - Vth))
    if(V(Carry_In)>Vth)
        carry = 1;
    else
        carry = 0;

    //output
    V(Integer) <+ transition(count+carry, Td, Tt);
    V(Phase_Shift) <+ transition(shift, Td, Tt);
end //analog
endmodule
```

6. Modèle de détecteur de phase

```

module PFD (U1, U2, Reset, UP, DN);

input U1, U2;
electrical U1, U2;
input Reset;
electrical Reset;
output UP, DN;
electrical UP, DN;

parameter real tP = 30e-12 from (0.0 : inf); // Propagation delay
parameter real tTR = 10e-12 from (0.0 : inf); // Transition delay
parameter real td_rst = 20e-12 from (0.0 : tP); // Reset signal delay
parameter real Vth = 0.5; // Threshold input voltage
parameter real Vhi = 1.0; // Output voltage corresponding to '1'
parameter real Vlo = 0.0 from [-inf : Vhi]; // Output voltage corresponding to '0'
parameter integer EDGE = 1 from [-1 : 1] exclude 0; // PFD triggering edge (-1)=falling (1)=rising
parameter integer state_init = 0 from [-1 : 1]; // Initial state of the PFD

integer rst, rst_tmp;
real UP_tmp, DN_tmp, UP_int_tmp, DN_int_tmp, UP_int, DN_int;

analog begin
    if (analysis("static")) begin // initial state setup
        if (V(Reset) < Vth) begin
            UP_tmp = Vlo;
            DN_tmp = Vlo;
        end else if (state_init == -1) begin
            UP_tmp = Vlo;
            DN_tmp = Vhi;
        end else if (state_init == 1) begin
            UP_tmp = Vhi;
            DN_tmp = Vlo;
        end else begin
            UP_tmp = Vlo;
            DN_tmp = Vlo;
        end
        UP_int_tmp = UP_tmp
        DN_int_tmp = DN_tmp
    end

    @(cross(V(U1) - Vth, EDGE)) // First input trigger
        UP_int_tmp = V(Reset);

    @(cross(V(U2) - Vth, EDGE)) // Second input trigger
        DN_int_tmp = V(Reset);

    UP_int = transition(UP_int_tmp, tP); // Output delay
    DN_int = transition(DN_int_tmp, tP);

    V(UP) <+ UP_int; // Output update (without delay)
    V(DN) <+ DN_int;

    @(cross(UP_int - Vth) or cross(DN_int - Vth)) // internal reset compute
        rst_tmp = (UP_int > Vth && DN_int > Vth)? 1 : 0;

    rst = transition(rst_tmp, td_rst); // internal reset delay

    @(cross(rst - Vth, +1) or cross(V(Reset) - Vth, -1)) begin // internal or external reset action
        UP_int_tmp = Vlo;
        DN_int_tmp = Vlo;
    end
end
endmodule

```

7. Modèle de pompe de charge

```

module charge_pump (aVdd, aVss, aout, UP, DN );

inout aVdd, aVss, aout;                                // positive power supply, negative power supply, analog output
electrical aVdd, aVss, aout;
input UP, DN;                                          // first digital input, second digital input
electrical UP, DN;

parameter real Vth = 0.0;                             // Threshold input voltage
parameter real Ion = 50.0e-3 from (0.0 : inf];       // Magnitude of current produced by NMOS transistor at Vds=Vdsat
parameter real Iop = 50.0e-3 from (0.0 : inf];       // Magnitude of current produced by PMOS transistor at Vsd=Vdsat
parameter real rOFF = 1.0e7 ;                         // Resistance of transistor in cutoff region
parameter real rONn = 50.0 from (0.0 : rOFF);        // Resistance of NMOS transistor in linear region
parameter real rONp = 50.0 from (0.0 : rOFF);        // Resistance of PMOS transistor in linear region
parameter real rDSn = 100.0e3 from (0.0 : inf];     // Resistance of NMOS transistor in saturation region, parallel to current source
parameter real rDSp = 100.0e3 from (0.0 : inf];     // Resistance of PMOS transistor in saturation region, parallel to current source

real Vdsat_n, Vdsat_p, G_lin_n, G_lin_p, G_sat_n, G_sat_p, G_off;
integer cutoff_n, sat_n, cutoff_p, sat_p;

analog begin
@(initial_step) // conductances and threshold voltages calculation
begin
Vdsat_n = Ion * rONn;
Vdsat_p = Iop * rONp;
G_lin_n = 1.0/rONn;
G_lin_p = 1.0/rONp;
G_sat_n = 1.0/rDSn;
G_sat_p = 1.0/rDSp;
G_off = 1.0/rOFF;
end
@(initial_step or cross(V(aout,aVss) - Vdsat_n) or cross(V(DN) - Vth)) // Nmos region
begin
if (V(DN) < Vth)
cutoff_n = 1;
else begin
cutoff_n = 0;
if (V(aout,aVss) < Vdsat_n)
sat_n = 0;
else
sat_n = 1;
end
end
// NMOS equation: cutoff conductance + linear region conductance + Ion + saturation conductance
I(aout,aVss) : I(aout,aVss) == cutoff_n * (G_off * V(aout,aVss)) + (1-cutoff_n) * ((1-sat_n) * (G_lin_n * V(aout,aVss)) + sat_n * (Ion +
G_sat_n * (V(aout,aVss)-Vdsat_n)));

@(initial_step or cross(V(aVdd,aout) - Vdsat_p) or cross(V(UP) - Vth)) // Pmos region
begin
if (V(UP) < Vth)
cutoff_p = 1;
else begin
cutoff_p = 0;
if (V(aVdd,aout) < Vdsat_p)
sat_p = 0;
else
sat_p = 1;
end
end
// PMOS equation
I(aVdd,aout) : I(aVdd,aout) == cutoff_p * (G_off * V(aVdd,aout)) + (1-cutoff_p) * ((1-sat_p) * (G_lin_p * V(aVdd,aout)) + sat_p * (Iop +
G_sat_p * (V(aVdd,aout)-Vdsat_p)));

end
endmodule

```

8. Fonction d'estimation de la DSP du bruit de phase à partir des mesures des périodes

```
scrsz = get(0,'ScreenSize'); % screensize (positionnement identique qq soit l'écran)
close all;

% Process period data to compute Sphi(df)
echo off;
nfft=2^10; % should be power of two
winLength=nfft;
overlap=nfft/2;
winNBW=1.5; % Noise bandwidth given in bins
% Load the data from the file generated by the VCO
load periods.m;
% output estimates of period and jitter
ratio = periods(1);
periods = periods(20:end);
T=mean(periods);
J=std(periods);
maxdT = max(abs(periods-T))/T;
% print data
fprintf('T = %.3gs, F = %.3gHz\n',T, 1/T);
fprintf('Fref = %.3gHz, Fpll = %.3gHz, Ndiv = %d\n', 1/T, ratio/T, ratio);
fprintf('Jabs = %.3gs, Jrel = %.2g%%\n', J, 100*J/T);
fprintf('max dT = %.2g%%\n', 100*maxdT);
fprintf('periods = %d, nfft = %d\n', length(periods), nfft);
% compute the cumulative phase of each transition
phases = 2*pi*cumsum(periods-T)/T;
% compute power spectral density of phase
%[Sphi,f] = psd(phases,nfft,1/T,winLength,overlap,'linear');
[Sphi,f] = periodogram(phases,hamming(length(phases)),length(phases),1/T,'power'); % does not need resolution bandwidth
normalization
%[Sphi,f] = pwelch(phases,winLength,overlap,nfft,1/T,'power');

%[Sphi,f] = pwelch(phases,winLength,overlap,nfft,1/T,'onesided','psd');
% correct for scaling in PSD due to FFT and window
%Sphi = winNBW*Sphi/nfft;
%Norm to resolution bandwidth
%rbw=winNBW/(T*nfft);
%Sphi=Sphi/rbw;
%Return to VCO output from divider output
Sphi=Sphi*ratio;%*T;
% smoothing function
%Sphi = smooth(Sphi,8);

% plot the results
figure(2);
semilogx(f,10*log10(Sphi)); %f,10*log10(SphiSm)
set(gcf(),'Position',[0 2*scrsz(4)/3 scrsz(3)/4 scrsz(4)/3]);
set(gca, 'FontSize', 25);
title('\fontsize{25}\bfPower Spectral Density of VCO Phase');
xlabel('\fontsize{25}\bfFrequency (Hz)');
ylabel('\fontsize{25}\bfLf (dBc/Hz)');
RBW=sprintf('Resolution Bandwidth = %.0f Hz (%.0f dB)',rbw, 10*log10(rbw));
imtext(0.5,0.07, RBW);

hold on;
grid on;
axis([1e4 1e8 -160 -60]);
```

9. Modélisation du bruit de phase et des spurious à partir du mismatch

```
clear all
close all
clc

%% Parameters
% I/O with fractional divider
Fosc=2.402e9;
N_Ph=30;
Fref=30e6;% Fcanal = Fref/N_Ph
Fbw=3e6; % bande passante de la PLL (doit être inférieur à Fref/7
MP=60; % Marge de phase en degrés (n'agit sur rien : changer "b" dans LPF)

% calculs
Fcanal=Fref/N_Ph;

% de la chaine directe %
% Filtre de boucle : (R1 + C1)//C2 %
Ip=50e-6/(2*pi); % A
Kv=260e6*(2*pi); % MHz/V

% de l'affichage
start = 3; % starting frequency (10^start)Hz
stop = 8; % stopping frequency (10^stop)Hz
PointsPerDecade = 1000; % Number of points per decade (def = 1000)
PointsFFT = 2^16; % Number of points for Divider Spurious FFT (def = 2^16 = 65536)

%% Noise definitions
% Quartz noise per segments
QValue = [-103,-133,-153,-156]; % 100, 1k, 10k, 100k, 1M
QFreq = [1e3,1e4,1e5,1e6];

% VCO Noise per segments
VValue = [-51,-60,-89,-115,-137]; % 100, 1k, 10k, 100k, 1M
VFreq = [1e4,1e5,1e6,1e7,1e8];

% LPF Noise measurements : set 0 for calculus from Resistor R1
Vnoise = 45e-9; % [nV/sqrtHz] simu (eldo ac noise) = 45 ; calculus = 52 = sqrt(4*kB*T*R)

% Frequency Detector (PFD+CP) output noise (when driven by the same frequency at both inputs)
Inoise = 12e-15; % simulation result between 0.1 and 0.9V; Maximums are 17fF at 1V and 13fF at 0V

% Divider Noise
Jitter = 0.26e-12; % déviation standard du temps de propagation = variance^2

%% alignement des graphiques sur les fenetres
scrsz = get(0,'ScreenSize'); % screensize (positionnement identique qq soit l'écran)
if(scrsz(3) > 2000)
    scrsz(3) = scrsz(3)/2; % si double écran utiliser seulement celui de gauche
end;

%% nombre de points

Fsize = PointsPerDecade*(stop-start); % number of frequency points
fprintf('Resolution fréquentielle de %i points par décade\r\n',PointsPerDecade);

f = logspace(start,stop,Fsize);
w = f*2*pi;

% Variable de Laplace
p = 1i*2*pi.*f;
%s = tf('s');

%% Foncton de transfert du filtre de boucle (ALLEN) %%%
```



```

% calculs
Ndiv=Fosc/Fref;
Channel=ceil((Ndiv-floor(Ndiv))*N_Ph);

% Charge pump cascodée sur Filtre de boucle : (R1 + C1)/C2
b=13; % à définir selon la marge de phase désirée (13 ~ 60°)
rbp1=sqrt(b+1);
t0=rbp1/(2*pi*Fbw);
C1=Ip*Kv/Ndiv*b/(b+1)*t0^2/rbp1;
R1=t0/C1;
C2=C1/b;

% Ip=100e-6; % A
% Kv=300e6; % MHz/V
% C1=2.5e-12;
% R1=58e3;
% C2=0.365e-12;

R3=0;
%R3=R3+53e3;
%C3=1e-12;

%Fp=tf([C1*R1 1],[C1*C2*R1 C1+C2 0]);
Fp=(1+R1*C1.*p)/((C1+C2).*p+C1*C2*R1.*p.*p);
%Fp=tf(Fp2,[C3*R3 1]);
%Fp=Fp./(1+R3*C3.*p);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Fonction de transfert en boucle ouverte %%

FTD = Fp.*Ip*Kv./p;
FTR = 1/Ndiv;
FTBO = FTD.*FTR;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Fonction de transfert en boucle fermee %%

FTBF = FTD./(1+FTBO);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FIGURE 1 : FTBO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(1);
set(1,'Position',[0 2*scrsz(4)/2 scrsz(3)/2 scrsz(4)/2]);
subplot(2,2,1), semilogx(f,20*log10(abs(FTBO)));
xlabel('Frequency (Hz)')
ylabel('Phase Noise (dBc/Hz)')
title('FTBO')
grid on;
subplot(2,2,3), semilogx(f, atan2d(imag(FTBO),real(FTBO)), 'r');
xlabel('Frequency (Hz)')
ylabel('Phase Noise (dBc/Hz)')
title('FTB')
grid on;
subplot(1,2,2), semilogx(f,20*log10(abs(FTBF)));
grid on;
%subplot(2,1,2), semilogx(f, atan2d(imag(FTBF),real(FTBF)), 'r');
%grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Définition de la DSP de chaque bruit de phase %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(2);

semilogx(0,0);
set(2,'Position',[scrsz(3)/2 2*scrsz(4)/2 scrsz(3)/2 scrsz(4)/2]);
xlabel('\fontsize{25}\bfFrequency (Hz)')
ylabel('\fontsize{25}\bfPhase Noise (dBc/Hz)')
title('\fontsize{25}\bfPLL Noise Contributors')
set(gca, 'FontSize', 25);
grid on;
hold on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%% Bruit de la reference %%

PNQuartz = zeros(1,Fsize);
QNbSeg = size(QFreq,2);

for k=1:Fsize
    if f(k)<=QFreq(1)
        PNQuartz(k)=QValue(1)-(QValue(1)-QValue(2))*log10(f(k)./QFreq(1));
    elseif f(k)>QFreq(1) && f(k)<=QFreq(QNbSeg)
        for i=2:QNbSeg
            if f(k)>QFreq(i-1) && f(k)<=QFreq(i)
                PNQuartz(k)=QValue(i)-(QValue(i-1)-QValue(i))*log10(f(k)./QFreq(i));
            end;
        end;
    elseif f(k)>QFreq(QNbSeg)
        PNQuartz(k)=QValue(QNbSeg);
    end;
end;

DSPQuartz = 10.^(PNQuartz./10);
FQuartz = FTBF;
NoiseQuartz = DSPQuartz.*abs(FQuartz).*abs(FQuartz);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Bruit du VCO %%

PNVCO = zeros(1,Fsize);
VNBseg = size(VFreq,2);

for k=1:Fsize
    if f(k)<=VFreq(1)
        PNVCO(k)=VValue(1)-(VValue(1)-VValue(2))*log10(f(k)./VFreq(1));
    elseif f(k)>VFreq(1) && f(k)<=VFreq(VNBseg)
        for i=2:VNBseg
            if f(k)>VFreq(i-1) && f(k)<=VFreq(i)
                PNVCO(k)=VValue(i)-(VValue(i-1)-VValue(i))*log10(f(k)./VFreq(i));
            end;
        end;
    elseif f(k)>VFreq(VNBseg)
        PNVCO(k)=VValue(VNBseg);
    end;
end;

DSPVCO = 10.^(PNVCO./10);
FVCO = 1./(1+FTBO);
NoiseVCO = DSPVCO.*abs(FVCO).*abs(FVCO);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Bruit du LPF %%

if Vnoise == 0
    Vnoise = sqrt(4 * 1.38e-23 * 300 * R1);
end
PNLPF = ones(1,Fsize)*10*log10(Vnoise^2); % 20*log car Vnoise en V/sqrt(Hz); si V^2/Hz alors 10*log
for k=1:Fsize
    if f(k)>Fbw
        PNLPF(k)=PNLPF(k)-20*log10(f(k)./Fbw); % 20 log seulement parce que c'est un 2ème ordre !!! sinon c'est plus
    end;
end;

DSPLPF = 10.^(PNLPF./10);
FLPF = Kv./p./(1+FTBO);
NoiseLPF = DSPLPF.*abs(FLPF).*abs(FLPF);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Bruit du PFD + CP %%

DSPPFD = (Inoise*pi/Ip)^2; % Normalised Phase Noise at the PFD input
FPPFD = FTD./(1+FTBO);
NoisePPFD = DSPPFD.*abs(FPPFD).*abs(FPPFD);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Bruit du Diviseur à phase_switching %%
[dspdiv_fft,f_fft] = SwitchingPhaseSpurious(Fref, Jitter, N_Ph, Channel, PointsFFT);

```

```

deltaF = f_fft(2)-f_fft(1);
fprintf('Resolution fréquentielle de %dHz\r\n',deltaF);

DSPDIV = zeros(1,Fsize);
% mapping frequencies
for k=2:Fsize-1
if f(k)>f_fft(1) && f(k)<=f_fft(end)
indexes = find( f_fft>=f(k) & f_fft<f(k+1));
if ~isempty(indexes)
DSPDIV(k) = max(dspdiv_fft( indexes )); % recherche la valeur maximum dans l'intervalle entre f(k) et f(k+1)
else
DSPDIV(k) = dspdiv_fft(find(f_fft<f(k),1,'last')+1); % astuce pour prendre la prochaine valeur de la gamme de f_fft
end;
end;
end;

FDIV = FTD.*FTD./(1+FTBO);%;FTBF;
NoiseDIV = DSPDIV.*abs(FDIV).*abs(FDIV);

NoiseDIV(find(f<2e5)) = 0;

%%%%%%%%%%%%%%
%%%%%%%% Total des bruit de phase dans la PLL %%%%%%%%%

NoisePLL = NoiseQuartz + NoiseVCO + NoiseLPF + NoisePFD + NoiseDIV;

%All noises :
semilogx(f, 10*log10(NoiseQuartz), f, 10*log10(NoiseVCO), f, 10*log10(NoiseLPF), f, 10*log10(NoisePFD), f, 10*log10(NoiseDIV), f,
10*log10(NoisePLL), 'k');

hold on;
grid on;
axis([f(1) f(Fsize) -150 -50]);

rbw=1.5*Fref/(PointsFFT);
RBW=sprintf('Resolution Bandwidth = %.0f Hz (%.0f dB)',rbw, 10*log10(rbw));
imtext(0.5,0.07, RBW);

```

10. Fonction de génération de mismatch des phases en fonction du jitter RMS

```

function [Sphi,f] = SwitchingPhaseSpurious( Fdiv, Jitter, Phases, DivFrac, Points )
% Fdiv = 30e6
% Jitter = 260e-13
% Phases = 30
% DivFrac = 1
% Points = 2^16
% [Sphi,f] = SwitchingPhaseSpurious( Fdiv, Jitter, Phases, DivFrac, Points );
% simulates spurious noise of switching phase fractional divider architecture

T=1/Fdiv;

%% 1 channel; 1 phases random inter-phase periods mismatch
vec = randn(Phases,1);
% Phase propagation delays between consecutive phases, with correction of std and mean
PhaseMm = Jitter*(vec-mean(vec))/std(vec);

% Generate fractional divider phase jumping
CycleMm = zeros(1,Phases);
for k = 0:(Phases-1)
    CycleMm(k+1) = PhaseMm(1+mod(DivFrac*k,Phases));
end

% size(CycleMm)
% CycleMm = [ 2.7539881316e-12 0 -1.1838970850e-11 0 7.9145413396e-13 0 4.0540685069e-12 0 -1.9818033848e-12 0 -
2.9801314323e-12 0 4.8593143751e-12 0 7.9717731061e-12 0 3.2449742971e-12 0 2.2977279383e-13 0 -1.7368945331e-12 0
2.6775783718e-12 0 -2.4672393486e-12 0 -3.3032263182e-12 0 5.9051403400e-13 0 ]
% size(CycleMm)

% generate a signal of n=Points repeating the ordered sequence of TPh n=Repeat times
Repeat = 2^round(log2(Points/Phases));
% recalculate nfft
nfft=Phases*Repeat; % should be power of two
% Generate periodic signal
temp = zeros(1,Points);
temp = CycleMm; % first sequence
for k = 1:Repeat-1
    temp=[temp,CycleMm]; % concatenate temp with another period
end
CyclesMm = temp;% + 0.1*jitter*randn(1,nfft); % adding divider synchronous jitter

% %% random noise
% CyclesMm=jitter*randn(Points,1);
% nfft=Points; % should be power of two
% CycleMm = zeros(1,Phases);
% for k = 0:(Phases-1)
% CycleMm(k+1) = PhaseMm(1+mod(DivFrac*k,Phases));
% end
% max(CycleMm)

%% Process windowing data to compute Sphi(df)
winLength=nfft;
overlap=nfft/2;
% winNBW=1.5; % Noise bandwidth given in bins
% rbw=winNBW/(T*nfft);

%% output estimates of period and jitter
fprintf('T = %.3gs, F = %.3gHz\n',T, 1/T);
fprintf('Jabs = %.3gs, Jrel = %.2g%%\n', Jitter, 100*jitter/T);
fprintf('periods = %d, nfft = %d\n', length(CyclesMm), nfft);

%% compute
% compute the cumulative phase of each transition
phases=2*pi*cumsum(CyclesMm)/T;
% compute power spectral density of phase
% [Sphi,f]=pwelch(phases,winLength,overlap,nfft,Fdiv);
[Sphi,f]=psd(phases,nfft,Fdiv,winLength,overlap,'linear');
```

```
%Norm to resolution bandwidth and window  
Sphi=Sphi*T/nfft;
```

Conception et réalisation de circuits de génération de fréquences en technologie FDSOI 28nm

Le déploiement à grande échelle de l'internet des objets en général et des réseaux de capteurs en particulier, nécessite le développement de circuits et systèmes électroniques de plus en plus économes en énergie. Les circuits de génération de fréquences sont connus pour être un des blocs particulièrement énergivores. L'objectif de ce travail de thèse est double, d'une part développer une synthèse de fréquences très faible consommation et d'autre part démontrer les performances de la technologie FDSOI pour des applications analogiques et radiofréquences. Pour illustrer nos travaux, nous avons choisi comme exemple d'application le standard Bluetooth Low Energy, BLE.

Dans le premier chapitre sont présentées les spécifications du standard BLE, notamment les caractéristiques qui auront un impact sur la conception du circuit de synthèse de fréquences. Les spécificités de la technologie FDSOI sont ensuite rappelées, notamment la variabilité du procédé de fabrication et les possibilités offertes par la polarisation de la grille arrière. Enfin, nous terminons ce chapitre par une étude de l'état de l'art des architectures de transmetteurs radiofréquences.

Le deuxième chapitre présente les résultats de trois types de modélisation système de l'architecture de cette synthèse de fréquence que nous avons retenue, à savoir la synchronisation par phases. La première étude concerne le fonctionnement de l'architecture et doit permettre de définir les points clés à respecter pour son implémentation. La deuxième étude concerne le comportement des bruits afin de définir les meilleurs paramètres architecturaux permettant de respecter les spécifications. Enfin, la dernière concerne l'impact de l'architecture sur la génération de raies spectrales parasites. Cette étude nous a permis de fixer le cahier des charges du VCRO (Voltage Controlled Ring Oscillator) qui sera développé au chapitre suivant.

Le troisième chapitre est ainsi consacré à la conception, la réalisation et le test de VCRO en technologie FDSOI 28nm. Plusieurs topologies sont présentées et intégrées sur un circuit de test. Les premiers résultats de mesure sont encourageants, mais nécessitent sans doute d'être complétés. En effet, la sensibilité des circuits à la tension d'alimentation (pushing de l'ordre de 5 GHz/V) a rendu les mesures du bruit de phase très délicates. La consommation mesurée reste inférieure à 0,8 mA et la surface des circuits est de l'ordre de 600 μm^2 . Dans le quatrième et dernier chapitre, nous proposons une topologie de la PLL à resynchronisation de phases, chaque bloc a été conçu et optimisé au niveau circuit. Nous terminons ce chapitre par quelques pistes d'amélioration, notamment la technique d'auto-calibration par auto-polarisation de la grille arrière en utilisant une logique complémentaire qui a fait l'objet d'un dépôt de brevet.

Design and realization frequency synthesis circuits in 28 nm FDSOI technology

The large-scale deployment of internet of things, and sensors' network in particular, requires the development of energy-efficient electronic systems. The frequency synthesis circuits are known to be one of the energy-consuming blocks. The aim of this Ph.D. work is double, on one hand to develop a very low power consumption frequency synthesis and on the other hand to demonstrated the performances of the FDSOI technology for analog and radiofrequency applications. To illustrate our works, we chose as example of application the Bluetooth Low Energy, BLE, standard.

In the first chapter are presented BLE specifications, in particular the characteristics which will impact the design of the frequency synthesis circuit. The specificities of the FDSOI are reminded, in particular the process variability and the possibilities were offered by the back-gate biasing. Finally, we conclude this chapter with a state of the art study of the transmitter architectures.

The second chapter presents the results of three types of system modeling of the chosen architecture for the frequency synthesis, called phase synchronization. The first study concerns the functioning of the architecture and has to allow to define the key points to be achieved for its implementation. The second study concerns the behavior of the noises to define the best architectural parameters allowing to respect the specifications. Finally, the last one concerns the impact of this topology on the spurious generation. This study allowed us to fix the specifications of the VCRO (Voltage Controlled Ring Oscillator) developed in chapter 3.

The third chapter is so dedicated to the design, realization and test of VCRO in 28nm FDSOI technology. Several topologies are presented and integrated on a test-chip. The first measurement results are encouraging, but require to be completed. Indeed, the phase noise measurements are very complicates due to the pushing (about 5 GHz/V). The moderate power consumption remains lower than 0.8 mA and the die surface is around 600 μm^2 . In the fourth and last chapter, we propose a topology of the PLL using resynchronization of phases, each block was designed and optimized at the level circuit. We end this chapter with some improvement ideas, in particular the self-calibration using back-gate auto-biasing of complementary logic cells, which was patented.

