

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Ecole Nationale Polytechnique**



Département d'Electronique

Centre de Développement des Technologies Avancées

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Electronique

**Contribution à la Conception des PLL dédiées  
aux Applications RF**

Travail réalisé par :

**Oussama HADDAD**

Présenté et soutenu publiquement, le 21 Juin 2017, devant le jury composé de :

Président :	M. Rabia AKSAS	Professeur	ENP Alger
Rapporteurs :	M. Mohamed TRABELSI	Professeur	ENP Alger
	M. Abdelhalim A. SAADI	Maitre de Recherche/B	CDTA
Examineur :	M. Hicham BOUSBIA-SALAH	Professeur	ENP Alger



**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Ecole Nationale Polytechnique**



Département d'Electronique

Centre de Développement des Technologies Avancées

Mémoire de projet de fin d'études

Pour l'obtention du diplôme d'ingénieur d'état en Electronique

**Contribution à la Conception des PLL dédiées  
aux Applications RF**

Travail réalisé par :

**Oussama HADDAD**

Présenté et soutenu publiquement, le 21 Juin 2017, devant le jury composé de :

Président :	M. Rabia AKSAS	Professeur	ENP Alger
Rapporteurs :	M. Mohamed TRABELSI	Professeur	ENP Alger
	M. Abdelhalim A. SAADI	Maitre de Recherche/B	CDTA
Examineur :	M. Hicham BOUSBIA-SALAH	Professeur	ENP Alger

## **Remerciements**

*Je tiens tout d'abord à remercier **Dieu** le miséricordieux, qui m'a donné la force et la patience d'accomplir ce Travail.*

*Je tiens à exprimer ma profonde gratitude à Monsieur **Abdelhalim Saadi**, de m'avoir offert une attention particulière et pour son dynamisme, ses compétences scientifiques et sa disponibilité tout au long de ce parcours qui m'ont permis de mener à bien ce projet. Qu'il soit assuré de ma reconnaissance.*

*Un remerciement spécial à Monsieur **Mohammed Trabelsi**, de m'avoir accordé sa confiance et de m'avoir accompagné tout au long de cette expérience avec beaucoup de patience et de pédagogie.*

*Toutes mes reconnaissances les plus vives s'adressent à Monsieur **Rabia Aksas** pour l'honneur qu'il me fait en présidant le jury de ma soutenance, et à Monsieur **Hicham Bousbia-Salah** en acceptant d'examiner mon travail.*

*Enfin, tous mes remerciements et reconnaissances aux membres de l'équipe « **Analog RFIC** » du Centre de Développement des Technologies Avancées. Et à tous ceux qui ont, de près ou de loin, contribué à la réalisation du présent travail.*

## *Dédicace*

*A ma mère et mon père*

*A mes sœurs ainsi qu'à leurs enfants*

*A mon frère et à toute ma famille*

*A tous mes amis et mes collègues*

*A tous ceux qui travaillent jour et nuit pour un monde meilleur*

*Et enfin, à tous les opprimés sur terre*

## ملخص

يركز العمل المعروض في هذه المذكرة على تصميم حلقات اقفال الطور لتطبيقات الاتصالات اللاسلكية . المذبذب المحلي هو أحد العناصر الأساسية لوحدة الاتصال اللاسلكي لأنه يسمح بتوليد ترددات الموجة الحاملة المستعملة لتعديل إشارة الرسالة . يمكن استخدام حلقة اقفال الطور على أساس مضخة الشحن للعب هذا الدور نظرا لاستقرار التردد الذي تولده بالمقارنة مع المذبذبات الأخرى، لكن هذا التصميم يقدم نوعا من التعقيد في التصميم نظرا لأنه يتكون من دوائر التناظرية والرقمية .بناءا على نهج التصميم من أعلى إلى أسفل، العمل الذي قمنا به يقوم على ثلاثة عناصر :تصميم على مستوى النظام لحلقة مقفلة الطور من الدرجة 3 على أساس مضخة الشحن، تحسين الاستقرار ووقت اكتساب النظام، والتحقق من صحة التصميم من قبل المحاكاة السلوكية للتصميم المقترح بواسطة Verilog-AMS. تصميم لمذبذب ذو تحكم بفرق الكمون ذو رنان LC في التكنولوجيا سيموس 0.18 ميكرون .وأخيرا، تصميم مقسم التردد المبرمج ومكوناته، وهي، ما قبل المقسم  $N / N + 1$  بواسطة الخلايا المعدلة، والعدادات P و S بواسطة الخلايا القياسية.

**الكلمات المفتاحية:** حلقة اقفال الطور، مولف الترددات، المذبذب المحلي، مذبذب ذو تحكم بفرق الكمون، التكنولوجيا سيموس 0.18 ميكرون، مقسم التردد المبرمج، ما قبل المقسم، Verilog-AMS

## Abstract

The work presented in this paper focuses on the design of phase locked loops (PLLs) for RF applications. The local oscillator is one of the essential elements of the RF modules because it allows the generation of carrier frequencies used to modulate or demodulate the signals. Charge pump based PLL can be used to play this role to provide frequency stability, this architecture presents a certain complexity in the design since it consists of analog and digital circuits. Based on a top-down design approach, the work we have done is contains three elements: a system-level design of a 3<sup>rd</sup> order charge pump based PLL with optimized stability and system acquisition time, and validating the design by a behavioral simulation of the proposed architecture in Verilog-AMS. Second, design of a LC VCO in 0.18- $\mu\text{m}$  CMOS technology. Finally, design of the programmable frequency divider.

**Keywords:** PLL, Frequency Synthesizers, Local Oscillator, VCO, 0.18- $\mu\text{m}$  CMOS, programmable frequency divider, Prescaler, Verilog-AMS.

## Résumé

Le travail présenté dans ce mémoire porte sur la conception des boucles à verrouillage de phase (PLL) destinés aux applications RF. L'oscillateur local est l'un des éléments indispensables d'un module RF car il permet la génération des fréquences porteuses, la PLL à pompe de charge, peut être implémentée pour jouer ce rôle compte tenu la stabilité des fréquences qu'elle génère, cette architecture présente une certaine complexité lors de la conception vu qu'elle se compose de circuits analogiques et numériques. Dans une approche de conception descendante, le travail que nous avons réalisé s'articule sur trois volets : une conception niveau système de la PLL à pompe de charge de 3e ordre, en optimisant la stabilité et le temps d'acquisition du système, et en validant la conception par une simulation comportementale de l'architecture proposée en Verilog-AMS. Une conception d'un VCO LC en technologie CMOS 0.18- $\mu\text{m}$ . Et finalement une conception du diviseur de fréquence programmable.

**Mots-clés :** PLL, Synthétiseurs de Fréquence, Oscillateur Local, VCO, 0.18- $\mu\text{m}$  CMOS, Etude comportementale, Prédiviseur, Verilog-AMS.

# Table des Matières

<i>Liste des Figures</i> .....	
<i>Liste des Tableaux</i> .....	
<i>Liste des abréviations</i> .....	
<i>Liste des symboles</i> .....	
<b><i>Introduction générale</i></b> .....	<b>13</b>
<b><i>Chapitre 1</i></b> .....	
<b><i>1.1. Introduction</i></b> .....	<b>16</b>
<b><i>1.2. Position des Synthétiseurs dans les émetteurs/récepteurs RF</i></b> .....	<b>16</b>
<b><i>1.3. Fonction de génération de fréquence, boucle à verrouillage de phase et synthétiseur</i></b> .	<b>17</b>
<b><i>1.4. Synthétiseur de fréquence à division entière</i></b> .....	<b>18</b>
<b><i>1.5. Spécifications des Synthétiseurs</i></b> .....	<b>19</b>
1.5.1. Plage d'accord.....	20
1.5.2. Pas de synthèse.....	20
1.5.3. Temps d'établissement.....	20
1.5.4. Le bruit de phase .....	20
1.5.4.1. Définition.....	20
1.5.4.2. Impact du bruit de phase sur la transmission.....	22
1.5.5. Raies parasites.....	24
<b><i>1.6. Éléments Constituants les PLLs/Synthétiseurs</i></b> .....	<b>25</b>
1.6.1. Le Comparateur de phase et de fréquence .....	26
1.6.2. La pompe de charge .....	26
1.6.3. Le Filtre de boucle .....	26
1.6.4. L'oscillateur contrôlé par tension.....	27
1.6.5. Le diviseur de fréquence .....	27
<b><i>1.7. Modélisation des éléments de la PLL</i></b> .....	<b>27</b>
1.7.1. Modèle du comparateur de phase et de fréquence et de la pompe de charge.....	27
1.7.2. Modèle du filtre de boucle .....	30
1.7.3. Modèle de l'oscillateur.....	32
1.7.4. Modèle du diviseur de fréquence .....	32
<b><i>1.8. Fonctionnement d'une PLL, Modèle et types</i></b> .....	<b>33</b>
1.8.1 Modèle continu de la PLL.....	33
1.8.2. Stabilité, précision d'un système asservi .....	34

1.8.3. PLL d'ordre 2 type 2 .....	36
1.8.4. PLL d'ordre 3 type 2 .....	36
1.8.5. Etude du bruit de phase de la PLL .....	38
<b>1.8. Conclusion .....</b>	<b>39</b>
<b>Chapitre 2 .....</b>	<b>40</b>
<b>2.1. Introduction .....</b>	<b>40</b>
<b>2.2 Flot de conception d'un système mixte .....</b>	<b>40</b>
<b>2.2. Présentation des outils de CAO utilisé pour l'étude .....</b>	<b>41</b>
2.2.1. Matlab et Simulink .....	41
2.2.2. Langages de Description Hardware .....	41
2.2.3. Présentation du langage Verilog-AMS .....	42
2.2.4. Environnement de simulation Cadence .....	44
<b>2.3. Conception système du synthétiseur moyennant MATLAB .....</b>	<b>45</b>
2.3.1. Spécification du standard Zigbee .....	45
2.3.1.1. Bruit de phase .....	46
2.3.1.2. Raies parasites .....	47
2.3.2. Dérivation du modèle de référence .....	47
2.3.3 Calcul des paramètres du filtre .....	48
2.3.4 Évaluation du bruit de phase .....	50
<b>2.4. Conception comportementale des éléments du synthétiseur .....</b>	<b>51</b>
2.4.1. Conception de la Source de référence .....	51
2.4.2. Conception du Comparateur de phase et de fréquence .....	52
2.4.3. Conception de la pompe de charge .....	55
2.4.4. Conception du Filtre de boucle .....	57
2.4.5. Conception de l'oscillateur .....	57
2.4.5. Conception du diviseur de fréquence .....	58
<b>2.5. Etude comportementale du système complet .....</b>	<b>61</b>
2.4.1. Paramétrage de l'outil de simulation .....	61
2.4.2. Résultats préliminaires de la simulation .....	61
2.4.3. Niveau des parasites sur la tension de contrôle .....	66
<b>2.6. Conclusion .....</b>	<b>67</b>
<b>Chapitre 3 .....</b>	<b>68</b>
<b>3.1. Introduction .....</b>	<b>68</b>



<b>3.2. La technologie CMOS.....</b>	<b>68</b>
3.2.1. Les dispositifs en technologies CMOS .....	68
3.2.1.1. Le MOSFET .....	68
3.2.1.2. Résistance .....	70
3.2.1.3. Capacité .....	71
3.2.1.4. Inductance .....	72
3.2.1.5 Varactors .....	73
<b>3.3. Notions sur les oscillateurs .....</b>	<b>74</b>
3.3.1. Théorie des oscillateurs.....	74
3.3.2. Topologies des oscillateurs .....	75
3.3.3. Etude de l'oscillateur LC à base d'une paire différentielle .....	76
<b>3.4. Conception d'un VCO LC en technologie CMOS 0.18<math>\mu</math>m .....</b>	<b>78</b>
3.4.1. Conception du résonateur LC.....	78
3.4.2. Résonateur LC accordé en fréquence.....	79
3.4.3. Détermination de la résistance équivalente du résonateur .....	81
3.4.3. Conception du circuit actif .....	82
3.4.3. Dimensionnement de la source de courant.....	83
<b>3.5. Simulation du VCO LC.....</b>	<b>84</b>
3.5.1. Résultats de simulation .....	87
<b>3.6. Insertion du CMOS VCO LC dans le Synthétiseur .....</b>	<b>91</b>
<b>3.7. Conclusion.....</b>	<b>93</b>
<b>Chapitre 4.....</b>	<b>.....</b>
<b>4.1. Introduction.....</b>	<b>95</b>
<b>4.2. Conception du diviseur .....</b>	<b>95</b>
4.2.1 Conception du prédiviseur .....	96
4.2.2. Simulation du diviseur par 2 .....	100
4.2.3. Simulation du prédiviseur 2/3 .....	100
4.2.4. Simulation du prédiviseur 32/33 (t=410ps).....	101
4.2.4 Conception des compteurs P et S .....	103
4.2.4.1. Résultat de simulation du compteur (P/S).....	103
4.2.5. Simulation du diviseur complet .....	105
<b>4.3. Conception de la PFD.....</b>	<b>107</b>
<b>4.4. Conclusion.....</b>	<b>108</b>

<i>Conclusion générale</i> .....	<b>109</b>
<i>Références bibliographiques</i> .....	<b>111</b>
<i>Annexe</i> .....	<b>114</b>

## *Liste des Figures*

Figure 1. 1 : Le synthétiseur de fréquence dans les front end RF.....	16
Figure 1. 2: Techniques utilisées pour la synthèse de fréquence [2].....	17
Figure 1. 3: Schéma synoptique d'un synthétiseur à division entière.....	19
Figure 1. 4 : Bruit de phase d'un oscillateur. (a) cas idéal. (b) cas réel.....	22
Figure 1. 5 : Impact du bruit de phase sur la réception [33].....	23
Figure 1. 6 : Impact du bruit de phase sur l'émission [33].....	24
Figure 1. 7 : Spectre de sortie d'un synthétiseur avec des raies parasites [11].....	25
Figure 1. 8 : Architecture d'une PLL à pompe de charge [10].....	26
Figure 1. 9 : Diagramme d'état du PFD (↑ signifiant front montant) [9].....	28
Figure 1. 10: Implémentation d'un PFD à l'aide de circuits logiques.....	29
Figure 1. 11 : Modèle de la pompe de charge.....	29
Figure 1. 12: Diagramme temporel du PFD et de la pompe de charge [10].....	29
Figure 1. 13: Caractéristique approchée du PFD [9].....	30
Figure 1. 14: Filtre passif linéaire de premier et de second ordre [9].....	31
Figure 1. 15: Modèle linéaire continu de la PLL.....	33
Figure 1. 16 : Diagramme de Bode du gain en boucle ouverte d'un système.....	34
Figure 1. 17 : Etude de la précision d'un système boucle fermée.....	35
Figure 1. 18 : Modélisation du bruit dans les PLLs.....	38
Figure 1. 19: Fonction de transfert des bruits liés au PLLs [2].....	39
Figure 2. 1 Flot de conception d'un circuit mixte analogique/digital [20].....	41
Figure 2. 2 : Flot de simulation des circuits mixtes avec Cadence [20].....	45
Figure 2. 3 : Réponse de la fonction de transfert en boucle ouverte.....	50
Figure 2. 4 : Réponse indicielle de la PLL en boucle fermée.....	50
Figure 2. 5: Le bruit de phase de la PLL.....	51
Figure 2. 6: Setup de simulation du PFD.....	52
Figure 2. 7: Résultat de la simulation du modèle comportementale du PFD.....	55
Figure 2. 8 : Setup de simulation de la pompe de charge.....	56
Figure 2. 9: Signaux de simulation de la pompe de charges.....	56
Figure 2. 10 : Setup de simulation du VCO.....	58
Figure 2. 11 : Simulation comportementale du VCO.....	58
Figure 2. 12: Tension de sortie en fonction de la tension de contrôle.....	58
Figure 2. 13: Simulation comportementale du diviseur de fréquence.....	60
Figure 2. 14: Simulation du modèle comportementale du diviseur. (a) : entrée du diviseur. (b) : sortie du diviseur. .....	60
Figure 2. 15 : Setup de simulation du synthétiseur de fréquence.....	63
Figure 2. 16: Simulation du Synthétiseur. Les signaux de haut en bas sont : Référence, reset, sortie du diviseur, up et down de la pompe de charge, l'erreur (tension de contrôle).....	64
Figure 2. 17 : Phase de démarrage du synthétiseur.....	65
Figure 2. 18: Phase d'acquisition de la fréquence.....	65
Figure 2. 19: Phase de la commande des canaux (variation du rapport N).....	66
Figure 2. 20: Niveau des parasites sur la tension de contrôle.....	66
Figure 3. 1 : Physique du transistor MOSFET.....	70
Figure 3. 2 : Résistance à film mince. (a) structure sur silicium. (b) schéma équivalent en technologie CMOS..	71

Figure 3. 3: Résistance implantée. (a) structure sur silicium. (b) schéma équivalent. ....	71
Figure 3. 4: Capacité MOS. (a) structure en technologie CMOS. (b) Schéma équivalent. ....	72
Figure 3. 5 : Capacité MIM. (a) Structure sur Silicium. (b) Symbole. (c) Schéma équivalent. ....	72
Figure 3. 6 : Layout et schéma équivalent de l'inductance RF en technologie CMOS .....	73
Figure 3. 7: Structure et symbole d'une capacité MOS .....	73
Figure 3. 8 : Modélisation d'un oscillateur .....	74
Figure 3. 9: Mécanisme d'amplification et maintien du régime d'oscillation .....	75
Figure 3. 10: Topologies d'oscillateur. (a) Colpitts. (b) Anneau. (c) paire différentielle .....	76
Figure 3. 11: Oscillation à travers le modèle de la résistance négative. (a) idéal. (b) Réel. ....	77
Figure 3. 12: Entretien des oscillateurs à travers un circuit de résistance négative.....	78
Figure 3. 13 : Modèle petits signaux de la paire différentielle. ....	78
Figure 3. 14 : Résonateur accordé en fréquence. (a) Circuit sous Virtuoso Cadence. (b) Réponse $ S_{11} $ pour différentes tension de contrôle .....	80
Figure 3. 15: Setup de simulation pour la déduction de la résistance équivalente du résonateur accordé en fréquence .....	81
Figure 3. 16 : Déduction de l'impédance d'entrée du résonateur accordé en fréquence .....	82
Figure 3. 17: Montage miroir de courant qui va jouer le rôle de la source de courant du VCO.....	83
Figure 3. 18: Montage du circuit actif à résistance négative : paire différentielle et miroir de courant .....	84
Figure 3. 19: Circuit du VCO LC en technologie CMOS 0.18- $\mu\text{m}$ .....	86
Figure 3. 20 : Setup de simulation du circuit VCO-LC CMOS 0.18- $\mu\text{m}$ .....	86
Figure 3. 21 : Régimes transitoire et établi du VCO .....	88
Figure 3. 22 : la distribution spectrale du signal de sortie (Puissance de sortie) pour différentes valeurs de la tension de contrôle .....	88
Figure 3. 23 : Forme d'onde du signal de sortie du VCO pour différentes valeur de la tension de contrôle .....	89
Figure 3. 24 : Variation de la fréquence du signal de sortie en fonction de la tension d'accord ( $V_{\text{tune}}$ ).....	90
Figure 3. 25 : le gain du VCO $K_{\text{vco}}$ en fonction de la tension d'accord $V_{\text{tune}}$ .....	90
Figure 3. 26 : Bruit de phase du VCO.....	91
Figure 3. 27: La réponse du synthétiseur dans la phase de démarrage. Les signaux sont de haut en bas : référence, reset, sortie diviseur, down, up, erreur. ....	92
Figure 3. 28 : La réponse du synthétiseur à un saut de fréquence de 2.4 à 2.475 GHz. Les signaux sont de haut en bas : référence, reset, sortie diviseur, down, up, erreur. ....	92
Figure 3. 29 : Réponse du synthétiseur à la phase de commande des canaux. (a) vue d'ensemble. (b) zoom de la vue.....	93
Figure 4. 1 : Architecture du diviseur de fréquence [31] .....	96
Figure 4. 2 : Architecture du prédiviseur .....	97
Figure 4. 3 : Architecture du diviseur par deux.....	97
Figure 4. 4 : Architecture du diviseur par 3 .....	98
Figure 4. 5 : Architecture du prédiviseur 2/3 .....	98
Figure 4. 6 : Bascule logique Clocked-CMOS.....	98
Figure 4. 7 : Obtention d'une bascule TSPC à partir d'une bascule Clocked-CMOS .....	99
Figure 4. 8 : Schématique de la bascule TSPC utilisé dans ce travail.....	99
Figure 4. 9 : Réponse du diviseur par 2 moyennant une simulation sur Cadence, signal d'entrée bleu. Signal de sortie rouge.....	100
Figure 4. 10 : Réponse du prédiviseur 2/3 pour $MC=0$ et $MC = 1$ . Signal injecté en rouge. Signal de sortie en jaune. MC en vert. ....	101
Figure 4. 11 Setup de la simulation du prédiviseur 32/33 .....	102
Figure 4. 12 : Réponse du prédiviseur 32/33 pour $MC = 0$ et $MC=1$ . Signal injecté en rouge. Signal de sortie en jaune. MC en vert. ....	102

Figure 4. 13 : Périodes de la sortie du mode MC=0 et MC=1.....	102
Figure 4. 14 : zoom sur la réponse en mode MC=0 .....	103
Figure 4. 15 : Simulation du compteur avec N = 2 .....	104
Figure 4. 16 : Simulation du compteur avec N = 3 .....	104
Figure 4. 17 : Simulation du compteur avec N = 5 .....	105
Figure 4. 18 : Setup de la simulation du diviseur de fréquence.....	105
Figure 4. 19 : Résultats de la simulation du diviseur de fréquence .....	106
Figure 4. 20 : Périodes du signal pour S=0, S= 4 et S=5.....	106
Figure 4. 21 : Schématique de la PFD synthétisé.....	107
Figure 4. 22 Code verilog niveau porte logique synthétisé .....	108

## *Liste des Tableaux*

Tableau 2. 1 : Spécifications du protocole Zigbee .....	46
Tableau 2. 2 : valeurs utilisées dans les sources d'impulsions pour la simulation du PFD.....	53
Tableau 3. 1 : Topologies des oscillateurs [24].....	75
Tableau 3. 2: paramètres de l'inductance du résonateur LC.....	79
Tableau 3. 3 : Dimension du circuit d'accord de fréquence .....	79

## Liste des abréviations

**RF** : Radiofréquence

**CMOS**: Complementary Metal Oxide Semiconductor

**IOT**: Internet Of Things

**PLL**: Phase-locked Loop

**VCO**: Voltage Controlled Oscillator

**PFD**: Phase and Frequency Detector

**CP**: Charge Pump

**FD**: Frequency Divider

**CAO** : Conception Assisté par Ordinateur

**LO** : Local Oscillator

**MOSFET**: Metal Oxide Semiconductor Field Effect Transistor

**ITRS**: International Technology Roadmap for Semiconductors

**MIM**: Metal- Insulator- Metal

**TSPC**: True Single Phase Clock

## Liste des symboles

$f_{\text{out}}$  : fréquence de sortie

$f_{\text{ref}}$ : fréquence du signal de référence

$f_{\text{div}}$ : fréquence du signal de retour

$K_{\text{vco}}$ : gain du VCO Hz / V

$I_0$  : Courant de la pompe de charge

$M_v$  : Marge de phase

$\Phi, \varphi$ : Phases

$\omega$ : vitesse angulaire

$L \{ \Delta f \}$  : Bruit de phase

$R_p$  : Fonction d'autocorrélation

$R(\tau)$  : Résistance parasite

$S(f)$  : Densité spectrale de puissance



# **Introduction générale**

## **Introduction générale**

L'année 2020 constitue un échéancier important pour le déploiement de plusieurs services relatifs aux technologies de l'information et de la communication. Parmi ces services, l'internet des objets (*Internet of Things : IoT*) a bénéficié d'un intérêt particulier de la part des industriels ou même académiques. En effet, l'IoT se présente comme un marché très prometteur d'environ 157.05 Billion de Dollars à l'horizon 2021. En outre, la 5<sup>ème</sup> génération de la radio mobile connaîtra aussi un déploiement très prochain et va servir comme réseau de base pour l'IoT. Aussi, une thématique très intéressante qui se présente est le concept de l'INDUSTRIE 4.0 qui va bouleverser les différents secteurs industriels en leur octroyant une certaine intelligence et autonomie.

Un constat très important mérite d'être signalé par rapport à toutes ces technologies, est le fait qu'elles utilisent la communication sans-fil comme moyen d'interconnexion ou networking. A cet effet, la filière de la conception des modules de communication radiofréquence (RF) sera fortement sollicitée pour répondre à ce besoin croissant. Cependant, ces modules RF seront soumis à des spécifications sévères en termes de consommation de puissance, haute performance et un taux d'intégration important doit être achevé. Afin de répondre à ces exigences les concepteurs de circuits RF doivent faire usage de technologies monolithiques, en particulier la technologie CMOS qui offre une capacité d'intégration remarquable et une implémentation de circuits intégrés à faible consommation à condition qu'une conception adéquate soit réalisée.

En effet, l'établissement d'une communication RF est conditionnée par l'allocation d'un spectre de fréquence bien précis qui doit être respecté par les infrastructures déployées et par conséquent par la circuiterie des modules RF permettant d'établir une telle communication. Ainsi, ces modules RF, appelés aussi Front-end, doivent être dotés de blocs assurant la génération des fréquences porteuses, et ce, en adéquation avec les ressources radio allouées. Ces circuits de génération de fréquence sont connus sous le nom d'oscillateur local. Ils peuvent prendre plusieurs formes et configurations en fonction de l'architecture retenue par leur concepteur (oscillateur contrôlé en tension, oscillateur en anneau, oscillateur piézoélectrique...).

Néanmoins, ces oscillateurs ont tendances à perdre leurs précisions et présentent avec le temps une certaine déviation par rapport à la fréquence ciblée. En conséquence, ces oscillateurs sont insérés dans des systèmes plus complexes afin de contrôler ce shift en

fréquence ou même extraire d'autres fréquences qui seront utiles pour le Front-end en question. Parmi les systèmes qui asservissent les oscillateurs : les boucles à verrouillage de phase, PLL en anglais (*Phase Locked Loop*), sont les plus utilisées dans les applications RF. Ainsi vu l'intérêt et l'importance d'un tel circuit omniprésent dans les front-ends RF nous avons décidé de nous investir dans son étude et sa conception. De plus, afin de nous positionner dans un cadre spécial tout en étant adéquat avec les tendances actuelles citées ci-dessus, nous allons essayer de faire une conception d'une PLL en répondant au standard Zigbee qui sera utilisé dans le déploiement de l'*IoT*.

En réalité, dans les applications RF nous parlons souvent de synthétiseur de fréquence plutôt que PLL. En fait, un synthétiseur de fréquence est une PLL dont la boucle de retour n'est pas passive mais contient un autre circuit qui est généralement un diviseur de fréquence. Néanmoins, dans la littérature nous confondons PLL et synthétiseur vu qu'ils présentent la même architecture et assurent la même fonction. A cet effet, dans le présent document nous allons utiliser ces deux termes pour en définir le même circuit sauf dans le cas où nous devons faire la différence.

La conception des PLL présente une certaine complexité vu qu'il s'agit d'un circuit mixte qui traite des signaux analogiques et des signaux numériques. En conséquence, un concepteur de PLL doit être en possession d'un background dans la conception, et des circuits analogiques, et des circuits numériques. De plus, des connaissances relatives aux technologies d'implémentation sont nécessaires pour l'intégration des circuits mixtes mentionnés, dans notre cas, nous allons utiliser la technologie CMOS 0.18- $\mu\text{m}$ . Par ailleurs, un volet très important doit être considéré dans la conception des PLLs, en l'occurrence, la maîtrise des outils de CAO (Conception Assistée par Ordinateur), sachant que trois types de modules de CAO sont utilisés dans ce cadre, à savoir, des modules de simulation des circuits analogiques, des modules de simulation de circuits digitaux et ceux utilisés pour la simulation des circuits mixtes.

Sachant que les PLLs peuvent être vues comme des systèmes composés, deux approches sont adoptées pour leur concrétisation. La première approche, nommée bottom-up qui consiste à concevoir les éléments de la PLL au niveau d'abstraction transistor puis les connecter pour aboutir au système final. La seconde approche, nommée top-down qui consiste à entreprendre la conception du point de vue système moyennant des langages de programmation de haut niveau puis commencer à remplacer les blocs élémentaires par leurs circuits réels possédants

un niveau d'abstraction bas. Dans le présent travail nous avons opté pour la deuxième approche.

Ainsi afin de concevoir une PLL/Synthétiseur répondant à la norme Zigbee, nous procédons à travers les chapitres suivants,

Dans le premier chapitre, nous allons revoir toute la théorie nécessaire relative aux circuits de génération de fréquence plus spécialement les PLLs et synthétiseur de fréquence. Cette étude nous permettra de constituer une base solide qui nous servira dans la conception niveau Système du circuit envisagé.

Dans le second chapitre, nous allons entamer la conception niveau Système de la PLL. Il s'agit d'une conception comportementale qui sera effectuée en utilisant Matlab et le langage de description Hardware Verilog-AMS. Nous commençons d'abord par l'étude de stabilité et le temps de réponse du système, ensuite nous abordons la conception des éléments constituant la PLL puis nous procédons à leur interconnexion. L'environnement et les outils de CAO seront exposés au passage afin de valider la conception comportementale.

Une fois la conception niveau système optimisée, il va falloir remplacer les briques élémentaires de la PLL par leurs circuits niveau transistor. Comme l'oscillateur contrôlé en tension (VCO) est considéré comme le cœur de la PLL, nous allons présenter dans le chapitre 3 la conception d'un VCO en technologie CMOS 0.18- $\mu\text{m}$  où toutes les étapes de la conception seront illustrées. Une fois optimisé, le VCO sera inséré dans la PLL à la place du VCO conçu à base du Verilog-AMS, ainsi la PLL devient un circuit mixte, nous pourrons voir son comportement en utilisant la CAO présentée dans le chapitre 2.

Dans le dernier chapitre, nous allons traiter la partie relative aux circuits digitaux de la PLL. Nous allons présenter la conception d'un diviseur à base d'une architecture retenue de la littérature. Le diviseur à son tour est constitué de sous-circuits, la conception et la simulation de ces derniers seront aussi abordées.

# **Chapitre 1**

## **Fonction de Génération de Fréquence RF**

## 1.1. Introduction

Dans ce chapitre nous allons présenter la théorie nécessaire qui nous permettra de comprendre ce que c'est la fonction de génération de fréquence dans les circuits front-end radiofréquence. En premier lieu, l'architecture des front-ends RF sera exposée afin de voir la position des synthétiseurs de fréquence. Dans un second temps, les spécifications relatives aux synthétiseurs seront présentées, l'objectif est de définir les critères de base qui permettront d'évaluer une conception donnée d'un tel circuit. Enfin, les modèles propres aux blocs élémentaires du synthétiseur seront aussi abordés en vue d'octroyer les concepteurs avec une logique qui les aidera à réaliser une étude comportementale du point de vue système.

## 1.2. Position des Synthétiseurs dans les émetteurs/récepteurs RF

Un synthétiseur de fréquence génère un ensemble de signaux de fréquences données avec la plus grande stabilité et précision possible. C'est un module très important dans les systèmes de télécommunication modernes. Il représente le circuit qui assure la fonction de génération des fréquences dans les front-ends RF (Figure 1.1).

Dans un front-end RF, le synthétiseur de fréquence constitue l'Oscillateur Local (OL), son signal de sortie est utilisé pour la sélection des canaux et la transposition de fréquence. Du côté émetteur, le signal LO de fréquence élevée est utilisé pour transposer le signal bande de base en haute fréquence pour garantir la transmission à travers le faisceau hertzien. Du côté récepteur, ce même signal est utilisé pour transposer le signal RF en une fréquence plus basse (Fréquence Intermédiaire ou Bande de base) afin de lui extraire le signal utile.

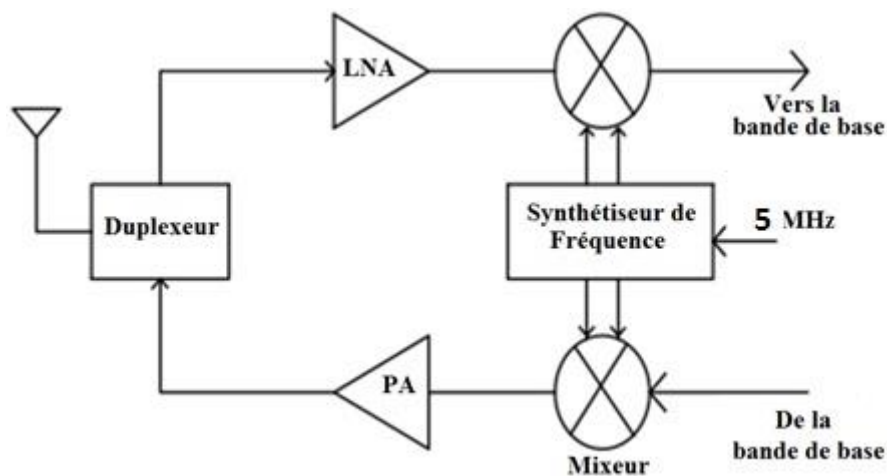


Figure 1. 1 : Le synthétiseur de fréquence dans les front end RF

### 1.3. Fonction de génération de fréquence, boucle à verrouillage de phase et synthétiseur

Lorsqu'on parle dans la littérature de fonction de génération de fréquence, on a tendance à confondre boucle à verrouillage de phase (PLL) et synthétiseur de fréquence. En réalité, la PLL représente la forme la plus simple d'un synthétiseur de fréquence, ou encore elle est utilisée comme le cœur d'un synthétiseur. La différence principale entre ces deux circuits, réside dans la présence ou non d'un étage de division de fréquence. Dans une PLL la boucle de retour est du type unitaire, alors que la synthèse de fréquence nécessite une division de la fréquence traitée.

Dans les applications RF on parle souvent de synthétiseur au lieu de PLL, notamment, lorsque les fréquences ciblées dépassent les 100 MHz [1]. Ainsi, nous allons nous focaliser sur les synthétiseurs. La figure 1.2 présente les techniques les plus importantes de synthèse de fréquences, on peut noter qu'il existe une variété de méthodes en fonction de plusieurs critères de classification. Dans notre travail, nous allons nous concentrer sur la synthèse de fréquence à division entière vu qu'elle est la technique la plus répandue et la mieux maîtrisée actuellement.

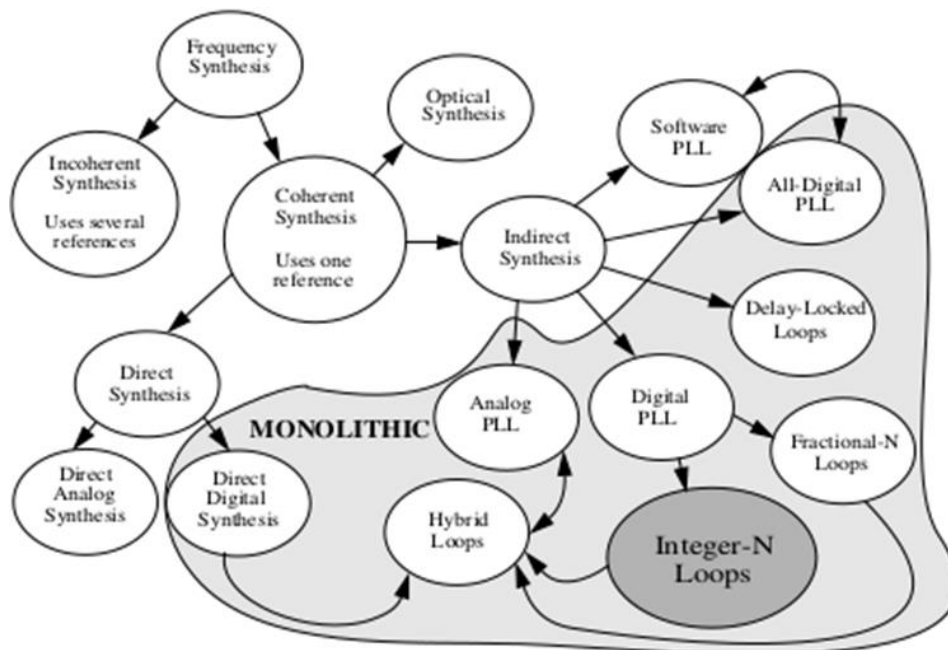


Figure 1. 2: Techniques utilisées pour la synthèse de fréquence [2]

## 1.4. Synthétiseur de fréquence à division entière

La synthèse de fréquence à division entière est une technique de synthèse indirecte qui consiste à asservir la phase et la fréquence du signal de sortie avec un signal d'entrée dont la phase et la fréquence sont très stables et très précises à un facteur  $N$  près [3].

Le nom indirect vient du fait que l'on multiplie une fréquence de référence par  $N$  en mettant un diviseur par  $N$  dans la boucle de retour. La synthèse indirecte est implémentée à l'aide d'une boucle à verrouillage de phase que l'on désigne par PLL (Phase-Locked Loop). La figure 1.3 donne la topologie générale d'un tel système.

La boucle à verrouillage de phase est un système à boucle de retour qui minimise la différence de phase entre un signal d'entrée dit de référence  $f_{ref}$  et un signal de retour  $f_{div}$ . Un détecteur de phase (PD : *Phase Detector* / PFD : *Phase Frequency Detector*) génère une erreur de phase dont la composante DC est proportionnelle à la différence de phase entre  $f_{ref}$  et  $f_{div}$ . Un filtre de boucle (du type passe-bas) extrait cette composante DC et l'applique à un oscillateur contrôlé par tension (VCO : *Voltage Controlled Oscillator*), ce qui changera la fréquence de sortie  $f_{out}$ .

Comme les synthétiseurs de fréquence doivent avoir une fréquence de sortie variable, un diviseur de fréquence (FD : *Frequency Divider*) de valeur programmable  $N$  est rajouté dans la boucle de retour. Si la valeur de  $N$  est entière, la PLL est dite à division entière.

Dans le régime établi, la différence de phase est constante ce qui signifie que la relation  $f_{out} = N \times f_{ref}$  est vérifiée. On dit alors que la PLL est verrouillée. La fréquence de sortie est  $N$  fois le signal de référence qui est le pas de synthèse.

Considérons deux signaux, respectivement, celui de référence et celui provenant du diviseur de fréquence,

$$S_{ref} = A \cos (\omega_{ref} t + \varphi_{ref}(t)) \quad (1.1)$$

$$S_{fd} = A \cos (\omega_{fd} t + \varphi_{fd}(t)) \quad (1.2)$$



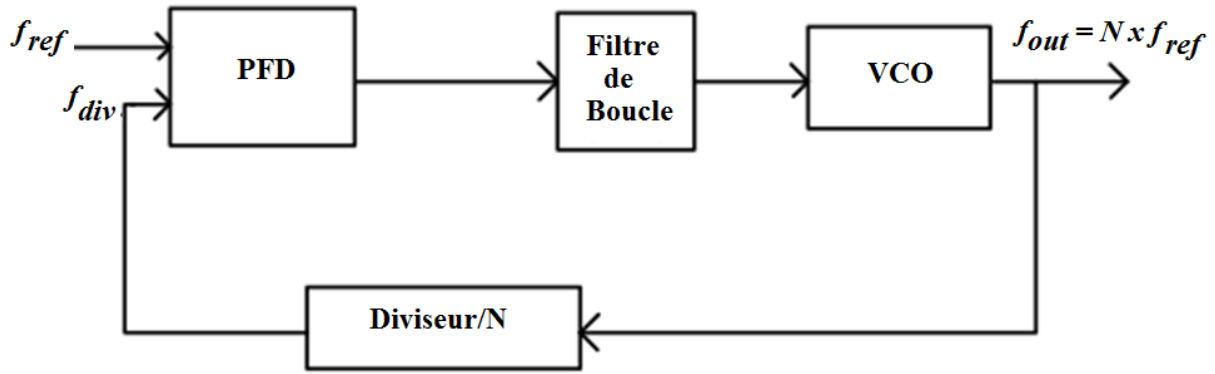


Figure 1. 3: Schéma synoptique d'un synthétiseur à division entière

Leurs phases sont données par,

$$\beta_{\text{ref}}(t) = \omega_{\text{ref}}t + \phi_{\text{ref}}(t) \quad (1.3)$$

$$\beta_{\text{fd}}(t) = \omega_{\text{fd}}t + \phi_{\text{fd}}(t) \quad (1.4)$$

Et leurs fréquences instantanées s'expriment par,

$$2\pi f_{\text{ref}}(t) = \frac{d\beta_{\text{ref}}}{dt} = \omega_{\text{ref}} + \frac{d\phi_{\text{ref}}(t)}{dt} \quad (1.5)$$

$$2\pi f_{\text{fd}}(t) = \frac{d\beta_{\text{fd}}}{dt} = \omega_{\text{fd}} + \frac{d\phi_{\text{fd}}(t)}{dt} \quad (1.6)$$

Si la différence de phase est constante i.e. :  $\beta_{\text{ref}}(t) - \beta_{\text{fd}}(t) = \text{cst}$

On aura  $f_{\text{ref}} - f_{\text{fd}} = 0$ , ce qui veut dire que les fréquences sont égales et par conséquent l'égalité suivante sera vérifiée.

$$f_{\text{Rf}}(t) = f_{\text{fd}}(t) = \frac{f_{\text{out}}}{N} \Rightarrow f_{\text{out}} = N f_{\text{ref}} \quad (1.7)$$

## 1.5. Spécifications des Synthétiseurs

Les synthétiseurs de fréquences doivent satisfaire un grand nombre de contraintes selon le standard de télécommunication ciblé. Ces contraintes sont généralement dénommées spécifications, nous allons en présenter les plus pertinentes dans ce qui suit.

### 1.5.1. Plage d'accord

La plage d'accord est la plage de fréquences que le synthétiseur doit couvrir, elle est définie par le standard ciblé, par exemple le standard IEEE 802.15.4 (Zigbee) exige une plage d'accord de 2.4 à 2.48 GHz [4].

### 1.5.2. Pas de synthèse

Le pas de synthèse représente l'écart entre deux fréquences adjacentes synthétisables par le synthétiseur. Il est décrit par l'espacement entre les canaux du standard ciblé, par exemple le standard IEEE 802.15.4 possède 16 canaux espacés de 5 Mhz [4].

### 1.5.3. Temps d'établissement

Le temps d'établissement d'un synthétiseur est le temps nécessaire pour passer d'un état stable de fréquence à un autre, par exemple le temps d'établissement du standard IEEE 802.15.4 est de 192 us.

### 1.5.4. Le bruit de phase

#### 1.5.4.1. Définition

Le bruit de phase est la représentation des variations aléatoires de la phase du signal de sortie d'un oscillateur. Dans le cas idéal, le signal de sortie d'un synthétiseur est une sinusoïde pure décrit par,

$$V(t) = V_0 \cos(2\pi f_0 t) \quad (1.8)$$

Où  $V_0$  et  $f_0$  représentent respectivement l'amplitude et la fréquence du signal. Sa représentation dans le domaine fréquentiel est composée de deux fonctions Dirac de fréquences  $\pm f_0$ .

Dans le cas réel, l'amplitude et la fréquence présentent des variations, l'équation (1.8) s'exprime alors comme suit,

$$V(t) = (V_0 + v(t)) \cos(2\pi f_0 t + \phi(t)) \quad (1.9)$$

Où  $v(t)$  et  $\phi(t)$  représentent ces variations.

Bien que les variations d'amplitudes soient présentes, elles peuvent être annulées ou réduites [5], donc en supposant des variations d'amplitudes négligeables, et en utilisant les relations trigonométriques, l'équation (1.9) peut être exprimée par,

$$V(t) = V_0 \cos(2\pi f_0 t + \phi(t)) = V_0 \cos(2\pi f_0 t) - \phi(t) V_0 \sin(2\pi f_0 t) \quad (1.10)$$

Sa densité spectrale est donnée par,

$$S_v(f) = \frac{V_0^2}{2} [\delta(f - f_0) + S_\phi(f - f_0)] \quad (1.11)$$

Où  $S_\phi$  représente la densité spectrale du bruit de phase, elle s'écrit comme suit,

$$S_\phi(f) = \int_{-\infty}^{+\infty} R_\phi(\tau) e^{-j2f\tau} d\tau \quad (1.12)$$

De plus,  $R_\phi$  est la fonction d'autocorrélation du bruit de phase, elle est donnée par,

$$R_\phi(\tau) = E[\phi(t)\phi(t - \tau)] = \int_{-\infty}^{+\infty} \phi(t)\phi(t - \tau) dt \quad (1.13)$$

On peut noter à partir de (1.11) que la densité spectrale du signal est constituée de la puissance à la fréquence porteuse  $f_0$  et de la puissance du bruit de phase à un offset  $\Delta f = f - f_0$ .

On définit le bruit de phase à bande unilatérale par le rapport de la densité spectrale de bruit dans une bande unitaire à une fréquence d'offset  $\Delta f$  de la porteuse, sur la densité spectrale de puissance de la fréquence porteuse, l'expression (1.14) formalise cette définition [6].

$$L\{\Delta f\} = 10 \log \frac{P_{noise}(f_0 + \Delta f, 1Hz)}{P_{carrier}} = 10 \log \frac{S_\phi(\Delta f)}{2} \quad (1.14)$$

Une autre manière pour quantifier le bruit de phase est son spectre qui est représenté sur la Figure 1.4. Il peut être ainsi représenté par une somme de sinusoides étalées autour de la fréquence porteuse, ceci, va en effet affecter les performances de transmission du système.

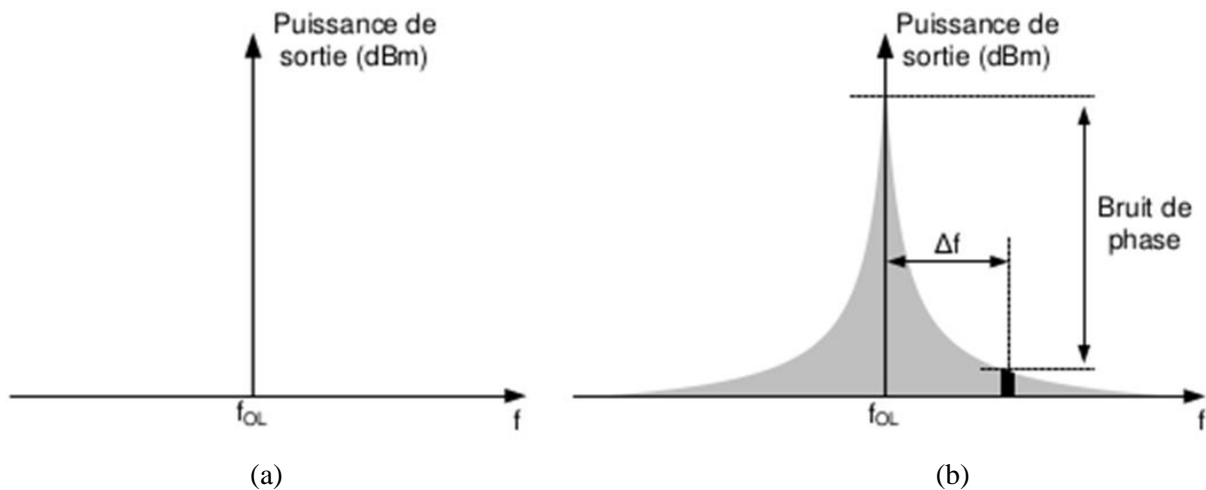


Figure 1. 4 : Bruit de phase d'un oscillateur. (a) cas idéal. (b) cas réel

#### 1.5.4.2. Impact du bruit de phase sur la transmission

Dans un émetteur, les composantes de bruit et les raies parasites peuvent être mélangées avec le signal bande de base et produisent en conséquence des émissions à des fréquences indésirables. Ces émissions indésirables peuvent interférer avec d'autres canaux ou systèmes voisins.

Dans un récepteur, les composantes de bruit et les raies parasites peuvent être mélangées avec d'autres signaux RF et interférer avec le signal désiré

De plus, lors d'une transmission où une modulation de phase est utilisée, le bruit de phase va perturber l'information, ce qui pourra entraîner une incertitude sur la décision lors de la démodulation entraînant une augmentation du taux d'erreur.

La figure 1.5 illustre cet impact,  $S_{RX}(f)$  et  $S_{TX}(f)$  représentent la densité spectrale du signal à l'entrée du récepteur et de l'émetteur respectivement,  $S_{RY}(f)$  et  $S_{TY}(f)$  représentent la densité spectrale de l'oscillateur local et finalement  $S_{RZ}(f)$  et  $S_{TZ}(f)$  sont les sorties des mélangeurs de réception et de l'émission

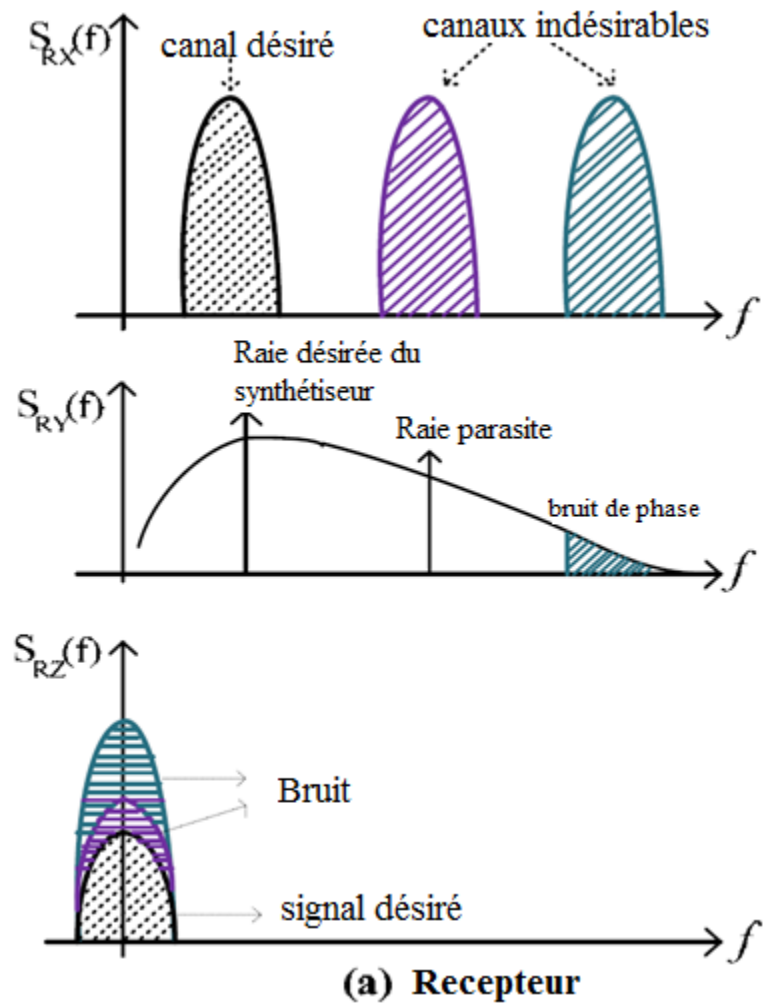


Figure 1. 5 : Impact du bruit de phase sur la réception [33]

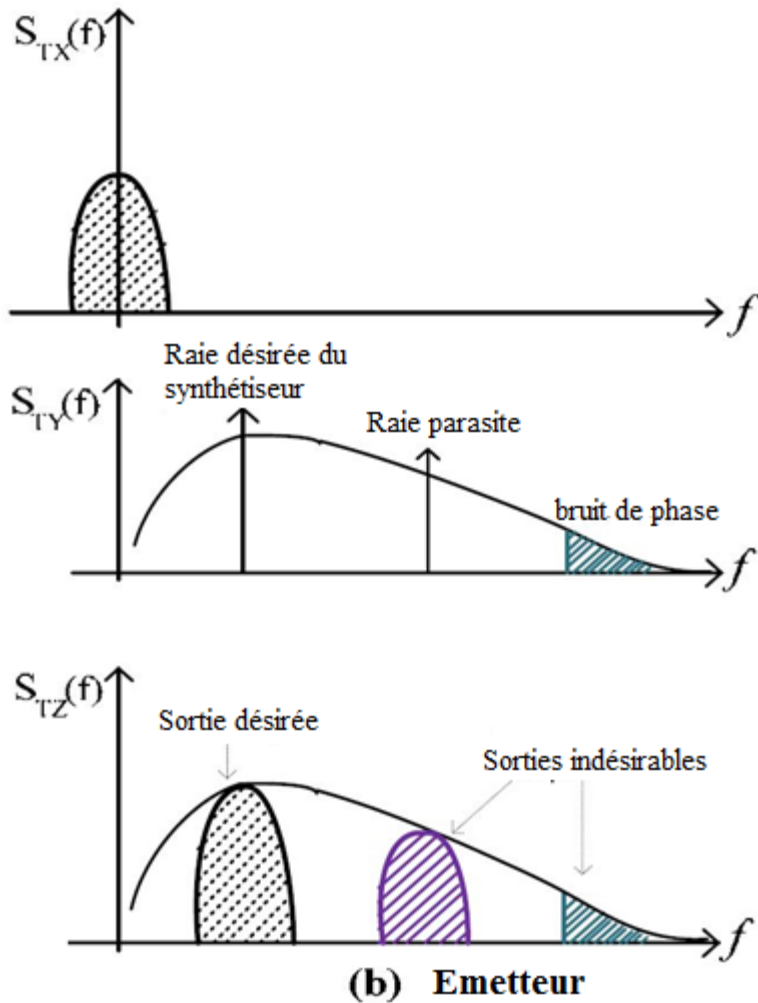


Figure 1. 6 : Impact du bruit de phase sur l'émission [33]

### 1.5.5. Raies parasites

Les raies parasites sont des fréquences présentes dans le spectre du synthétiseur de fréquence et inhérentes au procédés de synthèse. Les raies parasites peuvent avoir comme origine une modulation de phase du signal de référence, dans ce cas le signal peut s'écrire sous la forme [7],

$$V_{osc} = A \cos(2\pi f_{osc} t + \theta \sin(2\pi f_m t)) \quad (1.15)$$

Où  $A$  est l'amplitude du signal,  $f_{osc}$  est la fréquence de la porteuse,  $\theta$  est la déviation maximale de phase et  $f_m$  est la fréquence du signal modulant. Si  $\theta \ll 1$  l'expression (1.15) peut être approximée par,

$$V_{osc} = A \left[ \cos(2\pi f_{osc} t) - \frac{\theta}{2} \cos(2\pi(f_{osc} + f_m)t) + \frac{\theta}{2} \cos(2\pi(f_{osc} - f_m)t) \right] \quad (1.16)$$

Cette relation montre qu'en plus de la composante de la porteuse, il y a des composantes indésirables à  $\pm f_m$  de la porteuse et leur amplitude est  $V_{parasites} = A \frac{\theta}{2}$ . Les raies parasites générées par le synthétiseur de fréquence doivent être les plus faibles pour ne pas polluer les autres canaux (ou le signal transmis) et sont spécifiées par le standard. Par exemple, un niveau de -14 dBc/Hz pour une raie parasite à 5 MHz de la porteuse est requis pour la standard Zigbee [4].

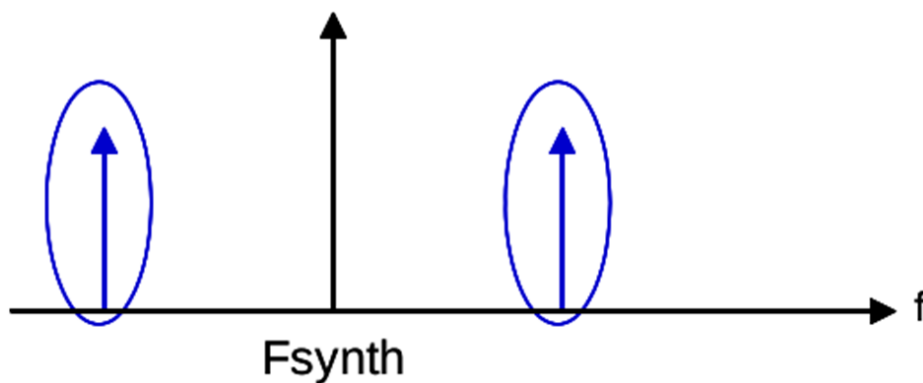


Figure 1. 7 : Spectre de sortie d'un synthétiseur avec des raies parasites [11]

## 1.6. Éléments Constituants les PLLs/Synthétiseurs

Il existe plusieurs architectures de PLL [8], l'architecture la plus utilisée dans les applications de communications RF est la PLL à pompe de charge dont le schéma synoptique est illustré par la figure 1.8. Dans cette architecture, le détecteur de phase est remplacé par un détecteur de phase et de fréquence (*Phase-Frequency Detector PFD*) et une pompe de charge (*Charge Pump – CP*).

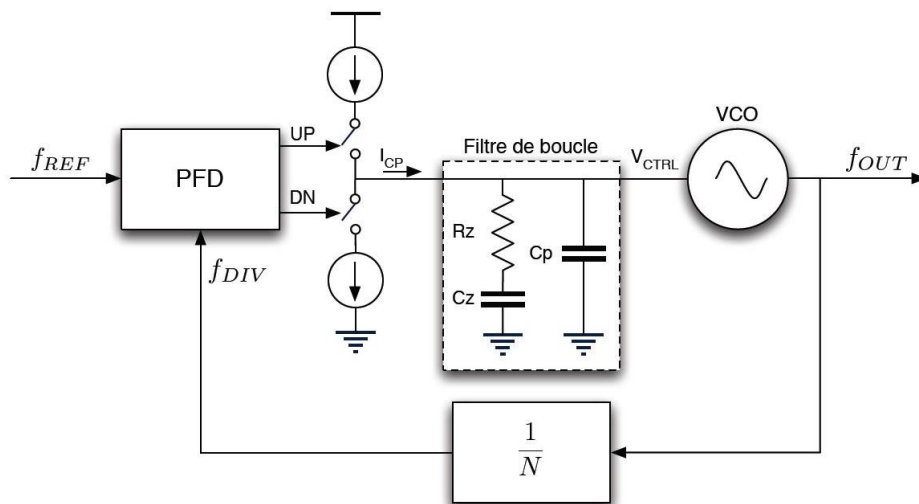


Figure 1. 8 : Architecture d'une PLL à pompe de charge [10]

L'utilisation d'une pompe de charge offre de nombreux avantages par rapport aux autres types de synthétiseurs, où une plage de verrouillage plus améliorée et un processus de capture plus rapide [8]. Aussi, la combinaison du détecteur de phase/fréquence et de la pompe de charges offre un gain suffisamment élevé en DC, et donc une erreur de phase nulle lorsque la PLL est verrouillée [8].

### 1.6.1. Le Comparateur de phase et de fréquence

Le PFD (*Phase-Frequency Detector*) compare la phase du signal à la sortie du diviseur de fréquence (figure 1.8) avec celle du signal de référence et génère un signal UP lorsque  $f_{ref}$  est en avance sur  $f_{fd}$  (respectivement DOWN lorsque  $f_{fd}$  est en avance sur  $f_{ref}$ ).

### 1.6.2. La pompe de charge

Elle consiste en une source et un puits de courant contrôlés par les deux signaux UP et DOWN du PFD. Lorsque le signal UP est activé, la pompe de charge injecte un courant  $I_0$ , et soutire un courant  $I_0$  lorsque le signal DOWN est activé.

### 1.6.3. Le Filtre de boucle

C'est un filtre passe bas qui convertit les impulsions de courant provenant de la pompe de charge en une tension continue qui contrôlera l'oscillateur contrôlé par tension (VCO). Le



filtre joue donc le rôle de régulateur qui asservit la fréquence du VCO dans la boucle de régulation.

#### **1.6.4. L'oscillateur contrôlé par tension**

Comme son nom l'indique, c'est un oscillateur qui génère un signal dont la fréquence varie en fonction de la tension de contrôle à son entrée. Quoique ce circuit peut être analogique ou numérique, il est très répandu d'utiliser des oscillateurs analogiques basés sur des résonateurs LC.

#### **1.6.5. Le diviseur de fréquence**

C'est un module qui divise la fréquence du VCO par un facteur  $N$  afin de comparer sa phase/fréquence avec celles du signal de référence. La fréquence du signal synthétisé est déterminée par la valeur programmable du facteur de division  $N$ . De plus, on dit qu'il s'agit de PLL plutôt que synthétiseurs lorsque la valeur de la division est égale à 1

### **1.7. Modélisation des éléments de la PLL**

L'approche de conception des PLLs adoptée dans notre travail est celle appelée Top-Down. Cette approche exige une compréhension d'un niveau d'abstraction élevée des éléments constituant la PLL. Ainsi, une modélisation précise de ces éléments permet une conception maîtrisée. Nous allons exposer dans ce qui suit cette modélisation.

#### **1.7.1. Modèle du comparateur de phase et de fréquence et de la pompe de charge**

Le PFD est le composant qui caractérise la PLL à pompe de charge, dans ce travail nous nous intéressons à un détecteur de phase/fréquence dit de type 3. C'est une machine à états finie dont le diagramme d'état est présenté dans la figure 1.9.

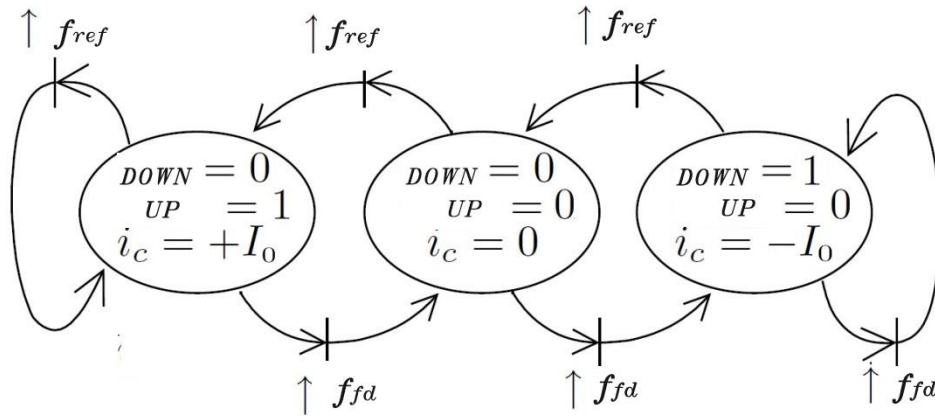


Figure 1. 9 : Diagramme d'état du PFD ( $\uparrow$  signifiant front montant) [9]

D'après ce diagramme d'état, le PFD possède trois états  $-I_0$ ,  $0$  et  $+I_0$  correspondant aux combinaisons permises des signaux de sortie UP et DOWN [0 1], [0 0] et [1 0]. Les transitions de l'état  $-I_0$  à  $+I_0$  sont activées par le front montant du signal de référence tandis que les transitions inverses par celui de la boucle de retour. Une implémentation de cette machine à états finis peut être réalisée à l'aide de bascules D et de porte AND comme le montre la figure 1.10 [8]. Les sorties du détecteur de phase/fréquence, UP et DN, vont commander la pompe de charge comme le montre la figure 1.11.

La pompe de charge consiste en deux sources de courant contrôlées par les signaux du PFD et dont la première s'active par le signal UP, tandis que la deuxième s'active par le signal DOWN.

Pour comprendre le fonctionnement de l'ensemble PFD et CP, nous analysons le diagramme temporel de la figure 1.12.

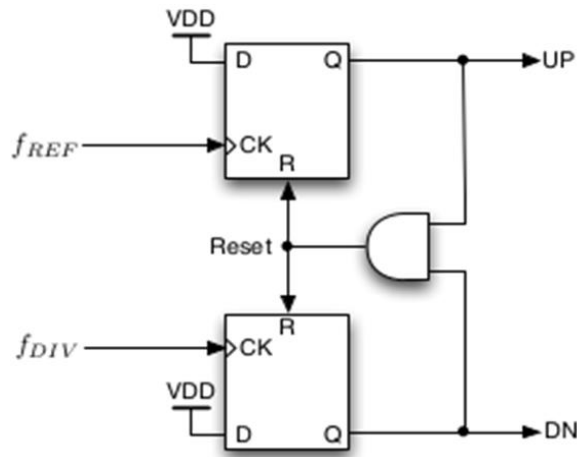


Figure 1. 10: Implémentation d'un PFD à l'aide de circuits logiques

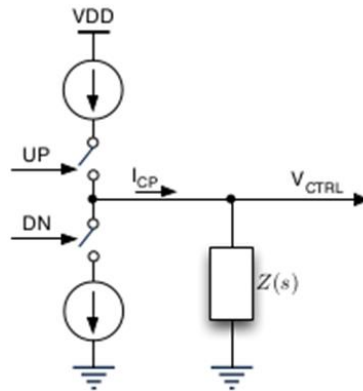


Figure 1. 11 : Modèle de la pompe de charge

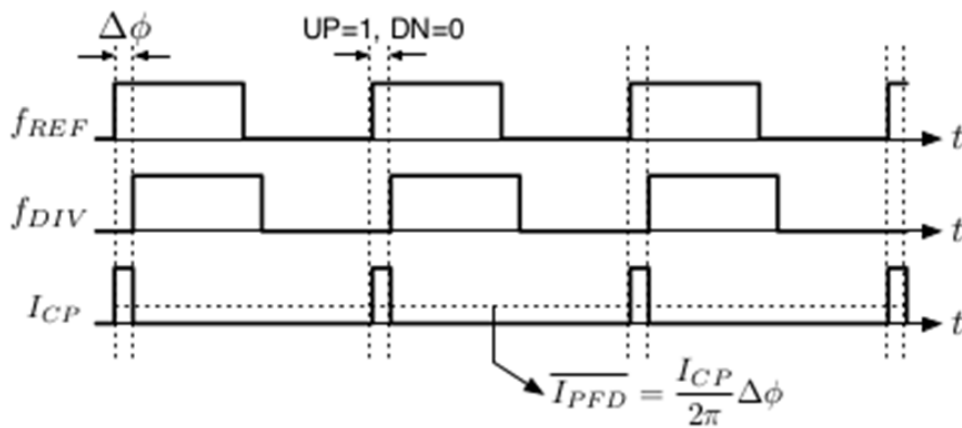


Figure 1. 12: Diagramme temporel du PFD et de la pompe de charge [10]

Lorsque le signal de référence est en avance sur celui du diviseur de fréquence, le signal UP sera activé pendant la durée qui sépare les deux signaux, ce qui est équivalent au pompage d'un courant  $+ I_0$  par la pompe de charge, le rapport cyclique de ce courant est proportionnel à

l'erreur de phase entre les deux signaux. On peut modéliser le comportement de ce système par la relation suivante [9],

$$i_o(t) = e(t) I_0 \quad (1.17)$$

Où  $e(t) \in \{-1, 0, 1\}$

Ce modèle est à caractère discret asynchrone, ce qui le rend difficile à manipuler, une solution est de prendre la moyenne de ce courant le long d'une période du signal de référence en fonction de l'erreur de phase  $\phi_e$  entre le signal de référence et le signal de boucle de retour. En faisant cette approximation, on obtient la caractéristique de la figure 1.13 [9]. Ce qui peut

être exprimé par la relation  $i_{0\text{moy}} = \frac{I_0}{2\pi} \phi_e$ ,  $\phi_e \in [0 \ 2\pi]$ .

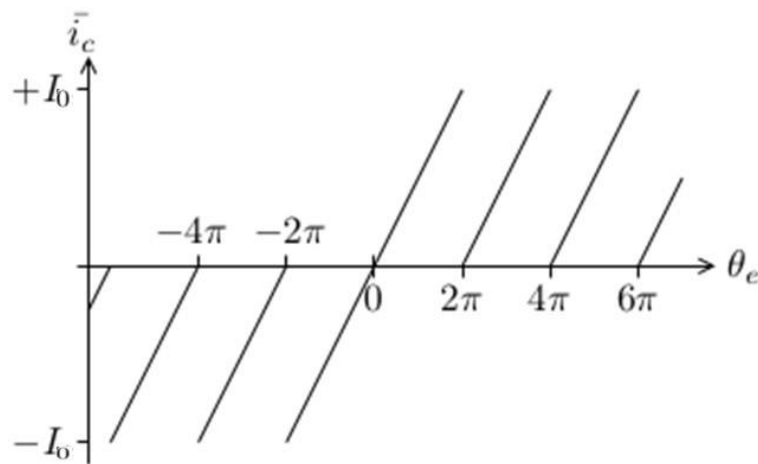


Figure 1. 13: Caractéristique approchée du PFD [9]

### 1.7.2. Modèle du filtre de boucle

Le filtre de boucle peut être passif ou actif, nous considérons seulement les filtres passifs vu qu'ils sont largement suffisants dans une architecture à pompe de charge [8]. L'ordre du filtre déterminera l'ordre de la PLL, une PLL est dite d'ordre deux, comporte un filtre d'ordre un. De même une PLL dite d'ordre trois comporte un filtre d'ordre deux. Le filtre doit fournir à sa sortie une tension la plus stable possible pour commander la fréquence du VCO. La topologie de filtres de premier et de second ordre est présentée dans la figure 1.14 [8].

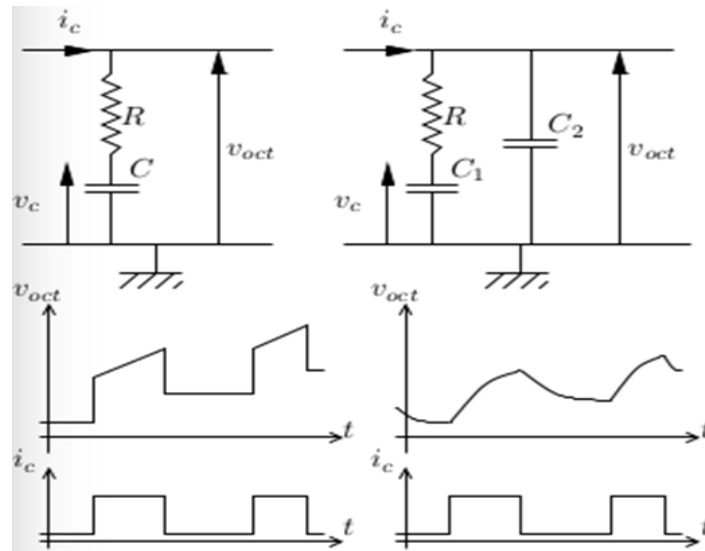


Figure 1. 14: Filtre passif linéaire de premier et de second ordre [9]

On remarque que la tension de sortie du filtre de premier ordre a une forme irrégulière, c'est pour cela qu'on rajoute la capacité  $C_2$  qui applique un lissage à la tension de sortie. En travaillant avec le modèle linéaire du courant de la pompe de charge, on peut écrire la fonction de transfert du filtre pour les deux ordres comme suit,

$$\frac{V_{VCO}(s)}{i_0(s)} = \frac{RCs + 1}{Cs} \quad \text{Pour un filtre d'ordre 1} \quad (1.18)$$

$$\frac{V_{VCO}(s)}{i_0(s)} = \frac{1}{s} \frac{RC_1s + 1}{RC_1C_2s + (C_1 + C_2)} \quad \text{Pour un filtre d'ordre 2} \quad (1.19)$$

La représentation temporelle de la tension de sortie est donnée par,

Pour un filtre d'ordre 1 [9]

$$\begin{aligned} V_{VCO}(t) &= v_c(t) + Ri_0(t) \\ v_c(t) &= \frac{i_0}{C}t \end{aligned} \quad (1.20)$$

Pour un filtre d'ordre 2 [9]

$$\begin{aligned}
V_{VCO}(t) &= v_c(t) + (v_{VCO}(t_0) - v_c(t_0))e^{-\frac{(t-t_0)}{\tau_1}} + \frac{i_0 \tau_1}{C_2} (1 - e^{-\frac{(t-t_0)}{\tau_1}}) \\
v_c(t) &= v_c(t_0) + \left[ \frac{\tau_1}{\tau_2} (v_{VCO}(t_0) - v_c(t_0)) - \frac{i_0 \tau_1}{C_1 + C_2} \right] (1 - e^{-\frac{(t-t_0)}{\tau_1}}) + \frac{i_0(t)}{C_1 + C_2} (t - t_0) \\
\tau_1 &= R \left( \frac{C_1 C_2}{C_1 + C_2} \right) = \frac{1}{\omega_p} \quad \text{et} \quad \tau_2 = RC_1 = \frac{1}{\omega_z}
\end{aligned} \tag{1.21}$$

### 1.7.3. Modèle de l'oscillateur

Le VCO délivre une fréquence instantanée proportionnelle à la tension  $V_{vco}$  présente à son entrée. Sa caractéristique linéarisée est donnée par l'équation (1.22) [11].

$$f_{vco}(t) = K_{vco} V_{vco}(t) + f_{vco,0} \tag{1.22}$$

où  $f_{vco,0}$  est la fréquence d'oscillation libre du VCO et  $K_{vco}$  est le gain du VCO (Hz/V).

On peut obtenir l'expression de la phase de sortie du VCO en fonction de la tension d'entrée  $V_{vco}(t)$  en intégrant l'expression (1.22), ceci mène vers (1.23).

$$\phi_{vco}(t) = \phi_{vco}(t_0) + 2\pi \left[ (t - t_0) f_{vco,0} + K_{vco} \int_{t_0}^t v_{vco}(\tau) d\tau \right] \tag{1.23}$$

Ainsi, la fonction de transfert de l'oscillateur idéal est obtenue en négligeant la contribution de la fréquence d'oscillation libre devant celle du gain, elle s'exprime alors par (1.24).

$$\frac{\phi_{vco}(s)}{v_{vco}(s)} = 2\pi \frac{K_{vco}}{s} \tag{1.24}$$

### 1.7.4. Modèle du diviseur de fréquence

Le diviseur de fréquence est un circuit digital qui fournit à sa sortie un signal de fréquence  $N$  fois inférieure à celle de l'entrée. Il est implémenté numériquement par un compteur qui compte les fronts du signal de sortie du VCO et bascule sa sortie une fois qu'il a atteint la valeur  $N$ . La fonction de transfert du diviseur est simplement son rapport de division.

$$\frac{\phi_{fd}(s)}{\phi_{vco}(s)} = \frac{1}{N} \quad (1.25)$$

## 1.8. Fonctionnement d'une PLL, Modèle et types

### 1.8.1 Modèle continu de la PLL

Le modèle linéaire continu d'une PLL est obtenu en prenant la moyenne du courant fourni par la pompe de charge à travers une période du signal de référence. La condition pour que le modèle reste valide est que la fréquence de référence soit très supérieure à la fréquence naturelle du système. [12]

Le modèle continu est obtenu en assemblant les fonctions de transfert des différents éléments. Ce modèle est représenté dans la figure 1.15. Le couple PFD/CP est modélisé par un gain égal à  $I_0/2\pi$  [12].

La fonction de transfert globale en boucle ouverte s'écrit,

$$G(s) = \frac{K_{vco} I_0 F(s)}{s 2\pi N} \quad (1.26)$$

La fonction de transfert en boucle fermée en considérant le modèle de la figure 1.15, s'écrit,

$$H(s) = \frac{G(s)}{1 + \frac{G(s)}{N}} = \frac{K_{vco} I_0 N}{2\pi} \frac{F(s)}{K_{vco} I_0 F(s) + s} \quad (1.27)$$

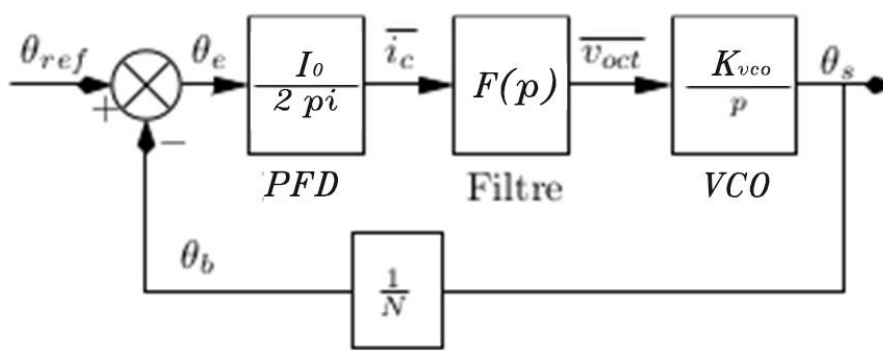


Figure 1. 15: Modèle linéaire continu de la PLL

### 1.8.2. Stabilité, précision d'un système asservi

#### ➤ Stabilité

Un système linéaire est stable s'il tend à revenir à sa position d'équilibre lorsqu'il est écarté de celle-ci. Il existe plusieurs moyens pour étudier la stabilité d'un système linéaire, on peut citer :

- Le lieu de Nyquist de la fonction de transfert en boucle ouverte, si le lieu parcouru suivant les fréquences croissantes passe à gauche ou sur le point «-1», alors le système est instable.
- Si le signe de la partie réelle de l'un des pôles de la fonction de transfert en boucle fermée est positif, alors le système est instable.
- Par le diagramme de Bode de la fonction de transfert de la boucle ouverte, si :  $20\text{Log}|G(j\omega_c)| > 0$  alors le système est instable. Où  $\omega_c$  est la pulsation pour laquelle  $\arg(G(j\omega)) = -180^\circ$

On définit la marge de phase qui est une mesure de la marge de sécurité que l'on prend par rapport à l'instabilité du système. La marge de phase d'un système est définie par :

$$M_\phi = \arg(G(j\omega_c)) + 180^\circ. \text{ Où } \omega_c \text{ est la pulsation pour laquelle } G(j\omega_c) = 1.$$

Les valeurs typiques pour la marge de phase se trouvent entre  $[45^\circ, 70^\circ]$  [13].

La figure 1.16 illustre graphiquement la notion de marge de phase.

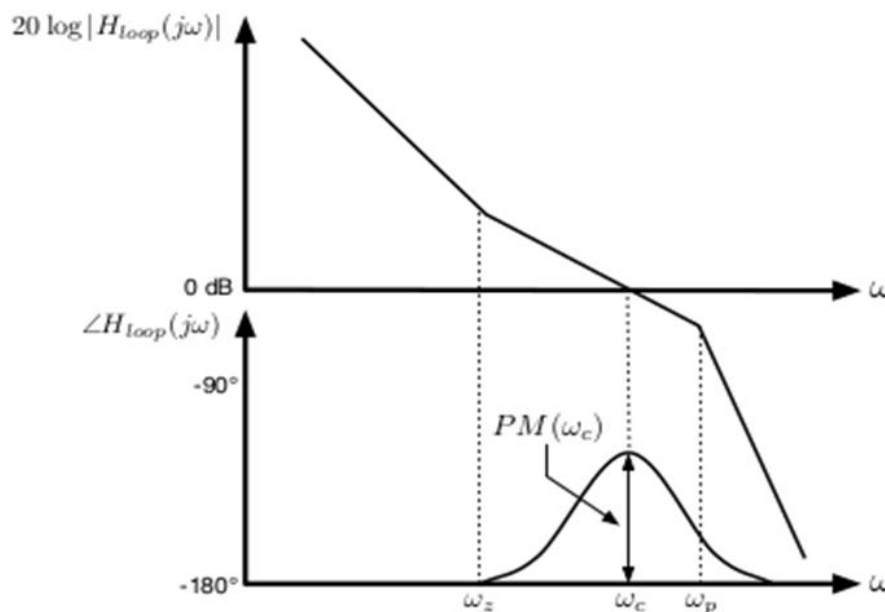


Figure 1. 16 : Diagramme de Bode du gain en boucle ouverte d'un système



➤ **Précision**

Soit le système bouclé de la figure 1.17,

Pour introduire la notion de précision, nous définissons l'erreur du système à une consigne donnée par,

$$x_E = x_c - x_R \quad (1.28)$$



Figure 1. 17 : Etude de la précision d'un système bouclé fermée

Ceci correspond dans le domaine de Laplace à l'expression suivante,

$$\begin{aligned} X_E(s) &= X_c(s) - A(s)B(s)X_E(s) \\ X_E(s) &= X_c(s) / (1 + A(s)B(s)) \end{aligned} \quad (1.29)$$

La précision du système est mesurée dans le régime établi, elle est définie par :

$$\varepsilon = \lim_{t \rightarrow \infty} x_E(t) \quad (1.30)$$

Par application du théorème de la valeur finale, il vient [13],

$$\varepsilon = \lim_{t \rightarrow \infty} x_E(t) = \lim_{s \rightarrow 0} sX_E(s) \quad (1.31)$$

D'après cette formule, l'erreur du système dépend de :

- la forme du signal de commande  $X_c$ .
- la limite de la fonction de transfert en boucle ouverte lorsque  $s$  tend vers 0.

Les entrées typiques auxquelles nous nous intéressons dans ce travail sont l'échelon et la rampe de phase et de fréquence. Dans le domaine de Laplace, l'échelon de phase a la forme,

$$X_E(s) = \frac{\nabla\phi}{s} \quad (1.32)$$

Et l'échelon de fréquence s'exprime par,

$$X_E(s) = \frac{\nabla\omega}{s^2} \quad (1.33)$$

Où  $\nabla\phi$  et  $\nabla\omega$  sont les amplitudes des variations de phase et de fréquence respectivement.

### 1.8.3. PLL d'ordre 2 type 2

Si le filtre est d'ordre 1, les expressions des fonctions de transfert en boucle ouverte et en boucle fermée respectivement sont,

$$G_2(s) = K_{vco} I_0 \frac{RCs+1}{Cs^2} \quad (1.34)$$

$$H_2(s) = N \frac{RCs+1}{\frac{N}{K_{vco} I_0} Cs^2 + RCs+1} = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (1.35)$$

$$\text{Où } \omega_n = \sqrt{\frac{1}{N} \frac{K_{vco} I_0}{2\pi C}} \text{ et } \zeta = \frac{\omega_n \tau}{2}$$

Et  $\omega_n$  et  $\zeta$  sont respectivement la pulsation naturelle et le facteur d'amortissement du système de deuxième ordre.

Les pôles de la boucle fermée sont donnés par :

$$s_{1,2} = \frac{1}{2} \left( \omega_{LF} \pm \sqrt{\left( \omega_{LF}^2 - \frac{2I_0 K_{vco}}{\pi N} \right)} \right) \quad (1.36)$$

$$\omega_{LF} = 2\zeta\omega_n$$

### 1.8.4. PLL d'ordre 3 type 2

De la même façon, si le filtre est d'ordre 2, alors la PLL est d'ordre 3. L'expression de la fonction de transfert en boucle ouverte est régit par,

$$G_3(s) = \frac{K_{vco} I_0}{C_1 + C_2} \frac{1}{Ns^2} \frac{RC_1 s + 1}{R \frac{C_1 C_2}{C_1 + C_2} s + 1} = K \frac{1 + s/\omega_z}{s^2 (1 + s/\omega_p)/\omega_z} \frac{b-1}{b} \quad (1.37)$$

$$\text{Avec } K = \frac{I_0 K_{vco} R_1}{2\pi N} \text{ et } b = \frac{C_2}{C_1 + C_2}$$

La bande passante d'une PLL à charge de pompe d'ordre 3 est [14],

$$\omega_c = K \cdot \frac{b-1}{b} \quad (1.38)$$

Afin de stabiliser le système, il faut choisir les paramètres du filtre d'une façon à avoir le maximum de marge de phase. La marge de phase s'écrit pour ce système comme suit,

$$M_\phi(\omega) = \phi_z - \phi_p = \tan^{-1}\left(\frac{\omega_c}{\omega_z}\right) - \tan^{-1}\left(\frac{\omega_c}{\omega_p}\right) \quad (1.39)$$

En cherchant la pulsation qui maximise la marge de phase, on trouve,

$$\frac{dM_\phi(\omega)}{d\omega} = \frac{1/\omega_z}{1+(\omega/\omega_z)^2} - \frac{1/\omega_p}{1+(\omega/\omega_p)^2} = 0 \Rightarrow \omega_c = \sqrt{\omega_p \omega_z} = \sqrt{b} \omega_z \quad (1.40)$$

Dans ce cas la marge de phase est,

$$M_\phi = \tan^{-1} \sqrt{\frac{\omega_p}{\omega_z}} - \tan^{-1} \sqrt{\frac{\omega_z}{\omega_p}} = \tan^{-1} \frac{b-1}{2\sqrt{b}} \quad (1.41)$$

En ce qui concerne la fonction de transfert en boucle fermée, elle est donnée par :

$$H_3(s) = N \frac{1 + s/\omega_z}{1 + s/\omega_z + s^2/(\omega_z \omega_c) + s^3/(\omega_z \omega_z \omega_p)} \quad (1.42)$$

Si la bande passante de la boucle est choisi tel que la marge de phase soit maximum, (1.42) peut s'écrire sous la forme,

$$H_3(s) = N \frac{\omega_c^3 + \sqrt{b} \omega_c^2 s}{(s + \omega_c) [s^2 + (\sqrt{b} - 1) \omega_c s + \omega_c^2]} \quad (1.43)$$

(1.43) peut s'écrire sous la forme générale suivante [14],

$$H_3(s) = N \frac{\omega_n^3 + (2\xi + 1)\omega_n^2 s}{(s + \omega_n)[s^2 + 2\xi\omega_n s + \omega_n^2]} \quad (1.44)$$

avec  $\xi = \frac{(\sqrt{b} - 1)}{2}$  et  $\omega_n = \omega_c$

### 1.8.5. Etude du bruit de phase de la PLL

Le bruit de phase de la PLL peut être introduit à différents points de la boucle par les différents modules qui la composent comme le montre la figure 1.18.

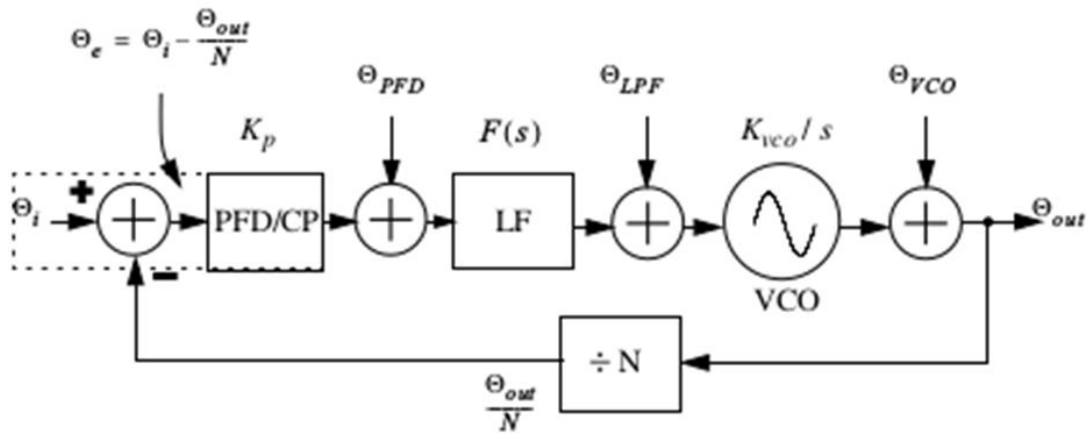


Figure 1. 18 : Modélisation du bruit dans les PLLs

Les fonctions de transfert des bruits provenant de chaque élément de la boucle sont données par les expressions (1.45) [15]. Aussi la figure 1.19 montre le tracé des fonctions de transfert relatives aux bruits liés aux éléments de la PLL.

$$\frac{\Phi_{out}}{\Phi_i} = \frac{K_{vco} \frac{I_0}{2\pi} F(s)}{Ns + K_{vco} \frac{I_0}{2\pi} F(s)} = \frac{NG(s)}{1+G(s)}$$

$$\frac{\Phi_{out}}{\Phi_{PFD}} = F(s) \frac{K_{vco}}{s} \left( \frac{1}{1+G(s)} \right)$$

$$\frac{\Phi_{out}}{\Phi_{LF}} = \frac{K_{vco}}{s} \left( \frac{1}{1+G(s)} \right)$$

$$\frac{\Phi_{out}}{\Phi_{VCO}} = \frac{1}{1+G(s)}$$
(1.45)

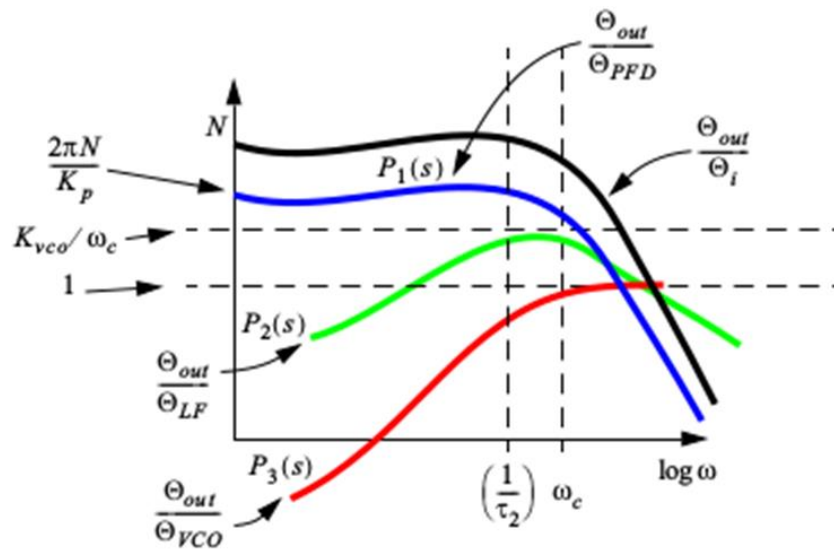


Figure 1. 19: Fonction de transfert des bruits liés au PLLs [2]

## 1.8. Conclusion

L'étude menée dans ce chapitre nous a permis de cerner le rôle des PLLs et synthétiseurs dans les modules RF. Nous avons pu comprendre la différence entre la PLL et le synthétiseur de fréquence, où ce dernier possède comme fonction supplémentaire l'option de modifier la fréquence d'oscillation même après la conception du système complet. Néanmoins, dans un cadre général, on peut utiliser les deux termes pour décrire la fonction de génération de fréquence.

En outre, les éléments constituant une PLL ont été présentés avec leurs modèles ainsi que leur architecture conventionnelle. En effet, les travaux présentés dans ce chapitre vont constituer une base solide sur laquelle reposeront les concepteurs pour réaliser les différentes réalisations que ce soit comportementale ou bien celles avec un niveau d'abstraction plus bas.

## **Chapitre 2**

# **Etude Comportementale d'un Synthétiseur de Fréquence**

## 2.1. Introduction

Une PLL ou synthétiseur de fréquence sont considérés comme étant des circuits complexes qui sont généralement dénommés systèmes plutôt que circuit. De plus, ce qui augmente la complexité d'un tel système est le fait qu'il soit mixte, c'est-à-dire composé de circuits analogiques et de circuits numériques.

La conception d'une PLL/synthétiseur peut se faire à l'aide de deux approches. La première est appelée conception ascendante (Bottom-Up), elle consiste à concevoir les éléments du système à partir de circuits ayant un niveau d'abstraction transistor jusqu'à ce que le système soit complet. La deuxième approche est appelée conception descendante (Top-Down) qui débute par une conception comportementale du système entier puis procéder à la conception des circuits élémentaires.

Dans notre cas nous avons opté pour la seconde approche et nous allons présenter dans ce chapitre la première étape de la conception qui est la plus importante, à savoir, l'étude comportementale du synthétiseur. Cette dernière est effectuée à travers la conception d'un modèle de référence répondant aux exigences du système puis la conception des circuits constituant le synthétiseur en se basant sur leur modèle comportemental moyennant le langage de Description Hardware mixtes verilog-AMS. Ce dernier est utilisé dans le cas de la conception d'un système mixte. Ainsi nous allons exposer dans cette partie toutes les étapes, outils et simulations nécessaires pour la conception comportementale du synthétiseur qui va répondre à la norme du Zigbee.

## 2.2 Flot de conception d'un système mixte

Le flot de conception d'un système mixte commence par une idée et se termine par un layout. Entre ces deux points il y a une série d'étapes de raffinement et de vérification [16]. En première étape, l'idée est transformée en une série de spécifications qui servent à établir une description fonctionnelle du système. Cette description est vérifiée à l'aide d'un simulateur système tel que MatLab et Simulink. Ensuite, la description fonctionnelle est traduite en une architecture qui sera vérifiée par un simulateur qui interprète les langages de description hardware mixte (MS-HDL) comme Verilog-AMS ou VHDL-AMS. Les blocs de l'architecture seront traduits à une représentation de niveau transistor, qui sera vérifiée à l'aide de simulateurs de timing ou SPICE. Cela représente le flow de conception électronique.

Dans le flot de conception physique, l'architecture est convertie à un Floorplan (Plan d'étage) qui sera raffiné jusqu'à ce que les blocs soient disposés et acheminés. La vérification du layout consiste à établir une correspondance avec le schématique (*Layout Vs Schematic*) et vérifie aussi les règles de conception (DRC). La vérification finale consiste à extraire le circuit à partir du layout en incluant ses paramètres parasites et simuler ce nouveau circuit. La figure 2.1 représente ce flot de conception.

## 2.2. Présentation des outils de CAO utilisé pour l'étude

### 2.2.1. Matlab et Simulink

MATLAB (Matrix Laboratory) est un langage de haut niveau et un environnement de programmation interactif pour le calcul numérique et la visualisation. MATLAB peut être utilisé lors des tâches telles que le traitement du signal et les communications, le traitement de l'image et de la vidéo, les finances informatiques et la biologie informatique. Il est le fondement d'un certain nombre d'autres outils, y compris Simulink et diverses boîtes à outils qui étendent ses capacités de base [17].

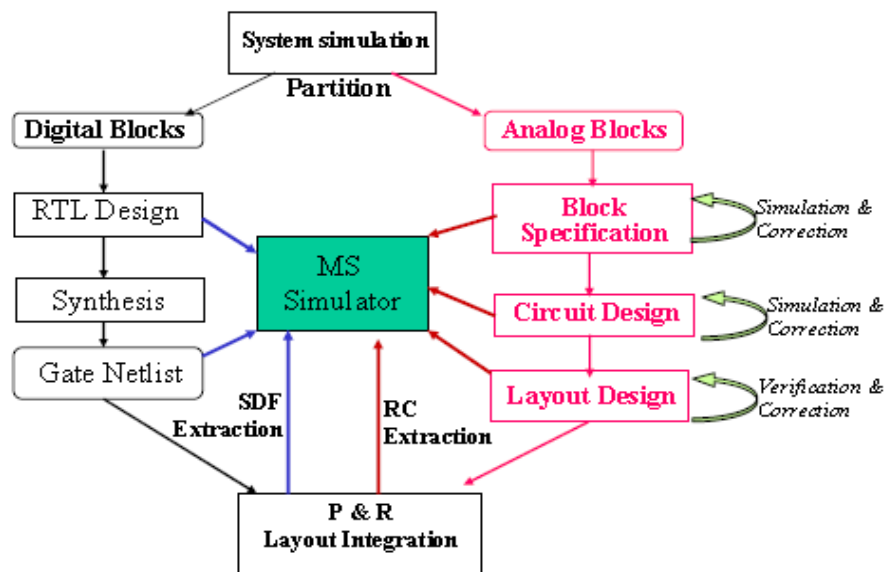


Figure 2. 1 Flot de conception d'un circuit mixte analogique/digital [20]

### 2.2.2. Langages de Description Hardware

Les langages de descriptions hardware (HDL) comme leur nom l'indique sont utilisées pour décrire le comportement d'un circuit, ils ont deux applications fondamentales : la simulation et la synthèse [18].



La simulation permet de comprendre comment les systèmes complexes se comportent avant d'engager le temps et les dépenses de leur mise en œuvre. La synthèse est l'acte de créer une nouvelle description raffinée avec un comportement équivalent aux entrées et sorties qui utilisent des composants qui ont une implémentation physique.

Actuellement, seules les machines à état finis sont synthétisées automatiquement, dans ce cas le comportement désiré est décrit au niveau RTL (Register Transfert Logic) en utilisant un sous-ensemble bien défini d'instructions HDL. La synthèse convertit la description RTL à une description de niveau porte logique. L'implémentation des portes logiques est disponible à partir d'une bibliothèque de cellule standard.

La synthèse automatique de systèmes analogiques et mixtes à partir d'une description du comportement désiré n'a pas pu progresser au point où elle devient pratique sauf pour quelques applications restreintes. La synthèse manuelle est le procédé entrepris par les concepteurs de circuits analogiques et mixtes pour convertir les spécifications de haut niveau à une implémentation qui satisfait ces spécifications.

Cependant, les systèmes mixed-signal sont devenus de plus en plus complexes au point où c'est devenu relativement complexe de les concevoir sans l'utilisation d'abstractions, et c'est là où les langages HDL sont utilisés.

Il existe principalement deux langages HDL pour décrire les systèmes mixtes, le Verilog-AMS et le VHDL-AMS. Ces deux langages permettent en effet d'assurer la modélisation des systèmes analogiques et mixtes. Ces deux langages supportent les analyses en temps continu tel que l'analyse en domaine temporel, analyse DC, AC, petits signaux et l'analyse de bruit. Néanmoins, le Verilog-AMS a une sorte de compatibilité avec le format SPICE utilisé dans la simulation des circuits, tandis que VHDL-AMS n'a pas cette compatibilité.

### 2.2.3. Présentation du langage Verilog-AMS

Verilog-AMS est un langage de modélisation comportementale et structurelle pour les systèmes mixed-signal. Le terme « *mixed-signal* » désigne des systèmes constitués de parties qui traitent des signaux digitaux, et des parties qui traitent des signaux analogiques.

Ainsi le Verilog-AMS constitue un langage et un simulateur unique qui est partagé par les concepteurs de systèmes et les concepteurs de circuits analogiques et digitaux. Son utilisation se trouve très utile pour les cinq raisons majeures suivantes [18],

### **1. Modélisation des composants**

Le Verilog-AMS offre une très grande variété de fonctionnalités qui peuvent être utilisées pour décrire efficacement une large gamme de modèles tels que les modèles de base, composants multidisciplinaires (capteur, actionneurs etc.), et les modèles de haut niveau.

### **2. Création de testbench**

Un testbench est la circuiterie ajoutée à un circuit sous test afin de simuler les dispositifs d'un banc de mesure réel. Il est facile de modéliser les éléments du testbench en Verilog-AMS vu qu'ils sont de nature idéale.

### **3. Accélération de la simulation**

Avec la croissance de la taille et de la complexité des systèmes, il est devenu de plus en plus difficile de les vérifier avec des simulations de niveau transistors. Souvent, il existe des parties critiques particulières de la conception qui sont les plus préoccupantes. Dans ce cas, le temps de simulation peut être réduit si les parties non critiques du circuit sont remplacées par des modèles comportementaux. Un modèle comportemental pour un bloc donné peut être simulé beaucoup plus rapidement qu'un modèle à base de transistors.

### **4. Vérification d'un système mixed-signal**

Lors de la conception de circuits à signaux mixtes, le Verilog-AMS est très utile car il permet de décrire les circuits numériques et analogiques de la manière la plus convenable à chaque type de circuit. Avec les circuits numériques, on utilise soit les portes logiques soit le Verilog-HDL au niveau comportemental, et avec les circuits analogiques, on utilise soit le niveau transistor soit le niveau comportemental.

### **5. La conception descendante**

La conception descendante est une méthodologie de conception utile lors de la conception des grands systèmes complexes. Le principe de base est de concevoir et de vérifier le système à un niveau abstrait avant d'entamer la conception détaillée des blocs individuels. La performance du bloc individuel doit répondre aux exigences globales du système, la performance est soigneusement étudiée et comprise avant que les blocs ne soient développés. La conception descendante sera utilisée à la place de l'approche plus traditionnelle appelée conception ascendante.

### 2.2.4. Environnement de simulation Cadence

La complexité de la conception des circuits intégrés, en particulier ceux dont le nombre de transistors utilisés dépasse les millions, a conduit vers le développement de méthodes et outils de conception assistée par ordinateur (CAO). L'un des outils les plus puissants dédiés à la conception, la simulation et la vérification des circuits intégrés, est l'outil Cadence et ses plateforme Virtuoso et Encounter EDI [19].

La plateforme Virtuoso contient des outils pour la conception des circuits intégrés analogiques, parmi ces outils on peut citer,

- **Virtuoso Schematic Editor** : outil pour le dessin du schéma électrique
- **Virtuoso Analog Design Environment** : Outil qui permet de configurer et d'exécuter des simulations analogiques en utilisant différents simulateurs. Après avoir exécuté une simulation, on peut visualiser et analyser le résultat de la simulation.
- **Spectre** : c'est un simulateur de circuits qui fournit une simulation SPICE de haute précision pour les modèles de signaux RF analogiques et mixtes. Spectre fournit un ensemble complet d'analyses détaillées des circuits de niveau transistor.
- **Virtuoso Verilog Environment** : c'est un outil qui permet de netlister les designs numériques, cela sous-entend la création de lien logique entre les différents blocs numériques. Cet environnement s'intègre également avec d'autres outils Cadence pour simuler, analyser et déboguer des modèles.
- **Virtuoso AMS Designer** : cet outil fournit un cadre pour le développement, la simulation et le débogage de blocs de conception de signaux mixtes

La plate-forme Encounter EDI : elle permet de faire la synthèse des circuits numériques à partir d'une description comportementale (RTL), jusqu'au Layout, parmi les outils que contient cette plateforme :

- **RTL Compiler** : c'est un outil qui permet la synthèse, niveau porte logique des modules décrits au niveau comportementale en Verilog
- **Simvision** : c'est un outil qui permet de simuler les composants et de visualiser les signaux numériques

Les liens existant entre ces différents type d'outils supportés sur Cadence est résumé par la figure 2.2 [20].

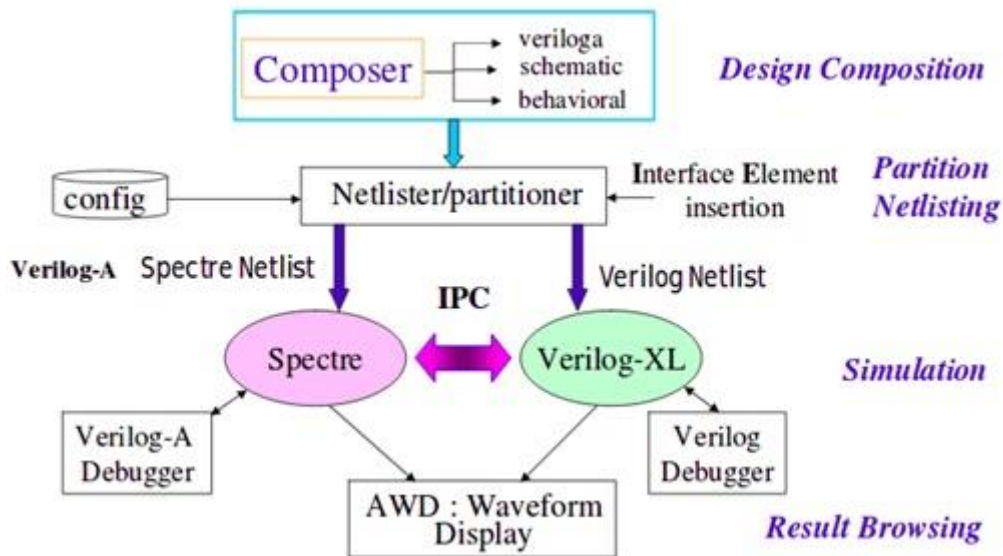


Figure 2. 2 : Flot de simulation des circuits mixtes avec Cadence [20]

## 2.3. Conception système du synthétiseur moyennant MATLAB

### 2.3.1. Spécification du standard Zigbee

Parmi les applications de communication RF, le nouveau concept de l'internet des objets (*Internet of Things : IoT*) est considéré parmi les plus prometteurs. Il s'agit d'un réseau de nœuds de capteurs doté d'une fonction de capture de données, une fonction de traitement et une fonction de transmission sans-fil. Ainsi, dans ce cas spécial le Zigbee est utilisé dans les modules de transmission RF comme norme vu qu'il répond aux exigences de l'*IoT* en terme de portée et quantité de données. C'est pour cette raison que nous nous sommes mis dans le cadre du Zigbee pour la conception de la PLL. Les spécifications du protocole sont résumées dans le tableau 2.1 [4]

Tableau 2. 1 : Spécifications du protocole Zigbee

Paramètre	Min	Typique	Max	Unité
Fréquence de sortie fout	2400		2485	MHz
Espacement de canal		5		MHz
Nombre de canal		16		
Temps d'acquisition				
Temps de verrouillage			192	µs
Temps de démarrage			200	µs
Raies parasites @fout				
@5 MHz			-14	dBc
@10 MHz			-44	dBc
Bruit de phase @fout				
@5MHz			-87	Dbc / Hz
@10 MHz			-117	Dbc / Hz

### 2.3.1.1. Bruit de phase

Dans le protocole Zigbee, la contribution des interférences co-canal vient des canaux adjacents loin du canal concerné de 5MHz et 10 MHz. D'après [21] le bruit de phase requis à un offset de la fréquence porteuse est déterminé par,

$$L_{requi}(\Delta\omega) = ((P_{sig} - P_{int}) - SNR_{requi} - 10\log(BP)) \quad (dBc/Hz) \quad (2.1)$$

Avec  $P_{sig}$  la puissance de la porteuse,  $P_{int}$  est la puissance d'interférence et BP est la bande passante du canal. En prenant  $SNR_{requi}=14$  dB de la littérature [22] et  $BP = 2$  MHz du standard Zigbee [4], donc le bruit de phase requis pour les deux offset 5 et 10 MHz est calculé en utilisant (2.1), les résultats sont donnés ci-après.

$$L_{requi}(5MHz) = ((0 - 0) - 14 - 10 \log(2.10^6)) = -77 \text{ (dBc/Hz)}$$

$$L_{requi}(10MHz) = ((0 - 30) - 14 - 10 \log(2.10^6)) = -107 \text{ (dBc/Hz)}$$

En ajoutant une marge de 10 dB, le bruit de phase requis à 5 MHz et 10 MHz devient, -87 dBc/Hz et - 117 dBc/Hz respectivement, ce qui correspond aux valeurs fournies dans le tableau 2.1.

### 2.3.1.2. Raies parasites

La puissance des raies parasites peut être calculée par l'équation (2.2) exprimée ci-dessous [23].

$$\begin{aligned} (P_{spurs})_{5MHz} &= (P_{sig} - P_{int}) - SNR_{requi} = (0 - 0) - 14 = -14dBc \\ (P_{spurs})_{10MHz} &= (P_{sig} - P_{int}) - SNR_{requi} = (0 - 30) - 14 = -44dBc \end{aligned} \quad (2.2)$$

### 2.3.2. Dérivation du modèle de référence

L'une des étapes primordiales lors de la conception d'une PLL/Synthétiseur est de définir le modèle mathématique de la PLL. Dans ce travail nous avons choisi une architecture de troisième ordre, le modèle général d'une fonction de transfert de troisième ordre est donné par [14],

$$H_3(s) = N \frac{\omega_n^3 + (2\xi + 1)\omega_n^2 s}{(s + \omega_n)[s^2 + 2\xi\omega_n s + \omega_n^2]} \quad (2.3)$$

Avec  $\xi$ ,  $\omega_n$ , sont tel que définis dans le chapitre 1 (1.44)

Pour que le système soit stable, il faut que la marge de phase soit supérieure à 45° [13]. En choisissant une marge de phase 55°, on peut calculer la valeur de b nécessaire comme suit [14],

$$M_\phi = \tan^{-1} \sqrt{\frac{\omega_p}{\omega_z}} - \tan^{-1} \sqrt{\frac{\omega_z}{\omega_p}} = \tan^{-1} \frac{b-1}{2\sqrt{b}} \quad (2.4)$$

Puis on pourra déduire la valeur de  $\xi$

Pour la fréquence de coupure, il faut qu'elle soit inférieure à la fréquence de référence, et supérieure à quatre fois la fréquence maximale requise pour le verrouillage [14]. On choisit

$$f_c = \frac{f_{ref}}{15} .$$

Il faut noter que tous les calculs discutés ci-dessus ont été élaborés avec un programme sous MATLAB.

### 2.3.3 Calcul des paramètres du filtre

Après la définition du modèle de référence, nous allons déduire la fonction de transfert du filtre en fonction de  $K_{vco}$ ,  $I_0$  et la valeur de  $N$ . La fonction de transfert du filtre,  $F(s)$ , apparaît clairement dans l'expression de la fonction de transfert en boucle ouverte,

$$G_3(s) = \frac{K_{vco} I_0}{s 2\pi N} F(s) \quad (2.5)$$

D'autre part, la fonction de transfert en boucle fermée peut s'exprimer aussi de la manière suivante,

$$H_3(s) = \frac{G_3(s)}{1 + N G_3(s)} \Rightarrow G_3(s) = \frac{H_3(s)}{1 - N H_3(s)} \quad (2.6)$$

En remplaçant (2.5) dans (2.6) on peut déduire  $F(s)$  en fonction des paramètres de  $K_{vco}$ ,  $I_0$  et  $N$ . On identifie les pôles et les zéros de cette fonction, à savoir,  $\tau_1$  et  $\tau_2$ . On peut ainsi en déduire les paramètres du filtre comme suit,

$$\begin{aligned} R_1 &= \frac{\omega_c}{\frac{I_0}{2\pi} K_{vco} \frac{1}{N}} \\ C_1 &= \frac{1}{R_1 \tau_2} \\ C_1 &= \frac{1}{R_1 \tau_1} \end{aligned} \quad (2.7)$$

Toutes les formules exposées ci-dessus ont été programmées sous MATLAB, le résultat des paramètres du filtre sont donnés ainsi par ce programme comme on peut le voir sur le code suivant.

Fonction de transfert en boucle fermée:

```
ClosedLoop =
      6.398e15 s + 4.483e21
-----
s^3 + 6.259e06 s^2 + 1.311e13 s + 9.187e18
```

Continuous-time transfer function.

```
R1 =
      1.2776e+05
```

```
C1 =
      1.1169e-11
```

```
C2 =
      1.2505e-12
```

Une fois arrivé à cette étape, nous pouvons vérifier la stabilité de notre système à travers le diagramme de Bode de la fonction de transfert en boucle ouverte (Figure 2.3). On peut constater que le système est stable du moment que la marge de phase est supérieure à 45°. De plus la réponse indicielle du système en boucle fermée, figure 2.4, confirme davantage la stabilité et la rapidité du système qui n'est rien autre que le temps d'acquisition.

En outre, afin de confirmer la validité de l'approche adoptée lors du calcul des paramètres sus-discutés, nous avons comparé le modèle de référence avec le modèle final de la PLL, où on peut noter la convergence entre les deux, comme exprimé dans le code MATLAB suivant.

```
Modele de reference:
ClosedLoop =
      6.398e15 s + 4.483e21
-----
s^3 + 6.259e06 s^2 + 1.311e13 s + 9.187e18
Continuous-time transfer function.
```

```
Modele de la PLL:
ans =
      6.398e15 s + 4.483e21
-----
s^3 + 6.259e06 s^2 + 1.311e13 s + 9.187e18
Continuous-time transfer function.
```



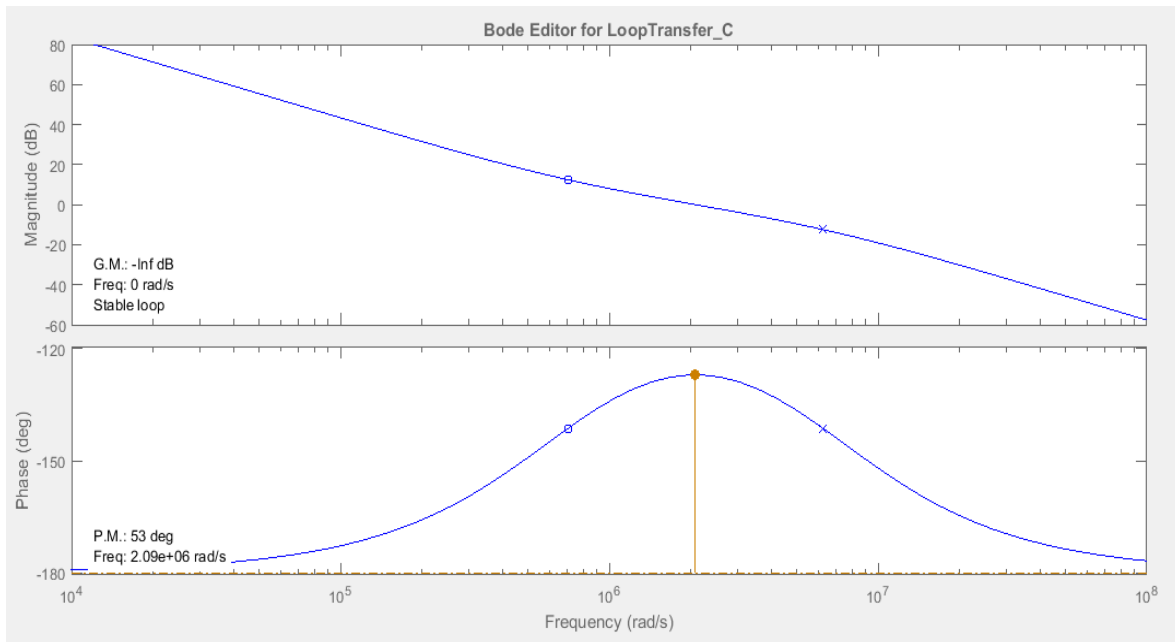


Figure 2. 3 : Réponse de la fonction de transfert en boucle ouverte

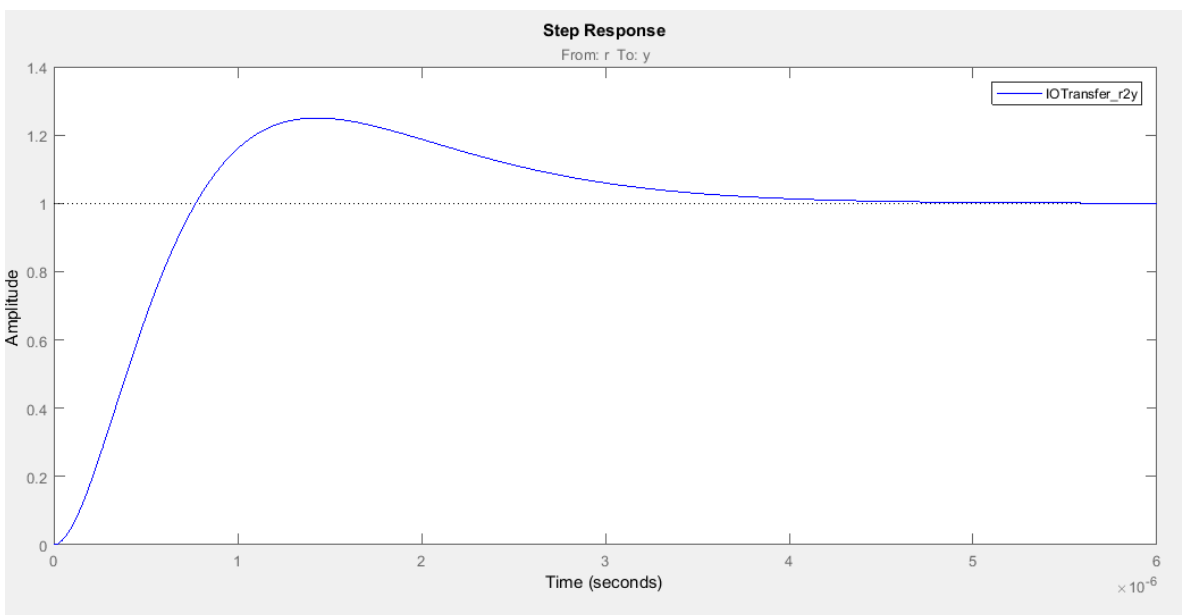


Figure 2. 4 : Réponse indicielle de la PLL en boucle fermée

### 2.3.4 Évaluation du bruit de phase

La dernière étape est l'évaluation du bruit de phase du système. On peut remarquer que la PLL copie avec une certaine atténuation le bruit de phase de la référence à l'intérieur de la bande inférieure à la fréquence de coupure.

Le bruit de phase du VCO est filtré à la sortie de la PLL, elle joue le rôle d'un filtre passe haut dans ce cas. La référence contribue par un bruit égal à  $20\log(N) = 20\log(488) = 53.76\text{dB}$ , mais cela ne pose pas de problème au système [2].

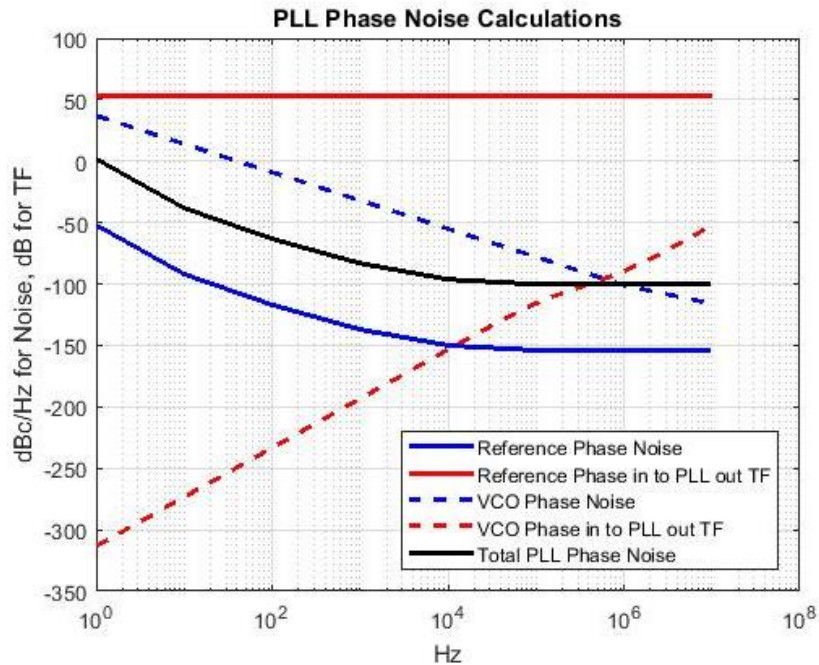


Figure 2. 5: Le bruit de phase de la PLL

## 2.4. Conception comportementale des éléments du synthétiseur

Dans cette section nous allons aborder la conception des différents éléments constituant notre circuit de génération de fréquence. Il s'agit dans ce cas d'une conception de type comportementale moyennant le langage Verilog-AMS. Les conceptions seront en effet validées sur Cadence.

### 2.4.1. Conception de la Source de référence

La source de référence est modélisée par une source d'impulsion de période  $T$ , et de rapport cyclique 50 %. Pour vérifier ce bloc, nous avons effectué une analyse transitoire à l'aide du simulateur AMS pour différentes valeurs de la période d'oscillation.

### 2.4.2. Conception du Comparateur de phase et de fréquence

Une description Verilog du PFD a été élaborée en se basant sur la machine à état fini présentée dans le chapitre 1. Le PFD présente 3 entrées, “fd” et “ref” qui sont les deux signaux à comparer, et un signal “reset”. Et possède aussi deux sorties “up” et “down” (figure 2.6).

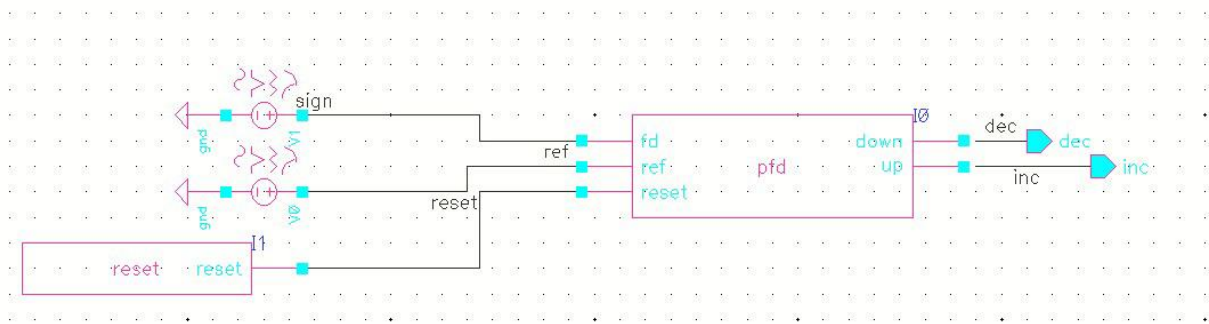


Figure 2. 6: Setup de simulation du PFD

Le code relatif à l'étude comportementale du PFD se présente comme suit,

```

`timescale 10ps / 1ps

module pfd (up, down, fd, ref, reset);
    output up, down;
    input reset, fd, ref;
    wire fv_rst, fr_rst;
    reg q0, q1;

    assign fr_rst = reset | (q0 & q1);
    assign fv_rst = reset | (q0 & q1);

    always @(posedge fd or posedge fv_rst) begin
        if (fv_rst) q0 <= 0; else q0 <= 1;
    end

    always @(posedge ref or posedge fr_rst) begin
        if (fr_rst) q1 <= 0; else q1 <= 1;
    end

    assign up = q1;
    assign down = q0;
endmodule

```

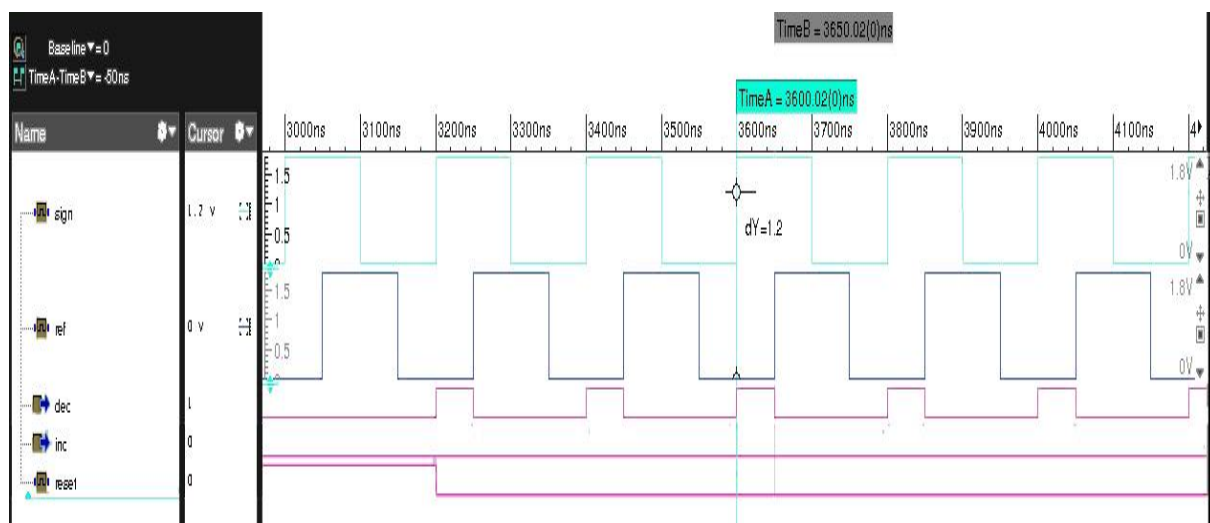
Afin de vérifier le bon fonctionnement du modèle proposé, le testbench de la figure 2.6 a été créé. Les deux entrées ‘fd’ et ‘ref’ sont connectées à deux sources d’impulsions ayant des

périodes T1, T2 et des délais d1, d2 que l'on change pour couvrir tous les états de la PFD. Les cas simulés sont résumés dans le tableau 2.1.

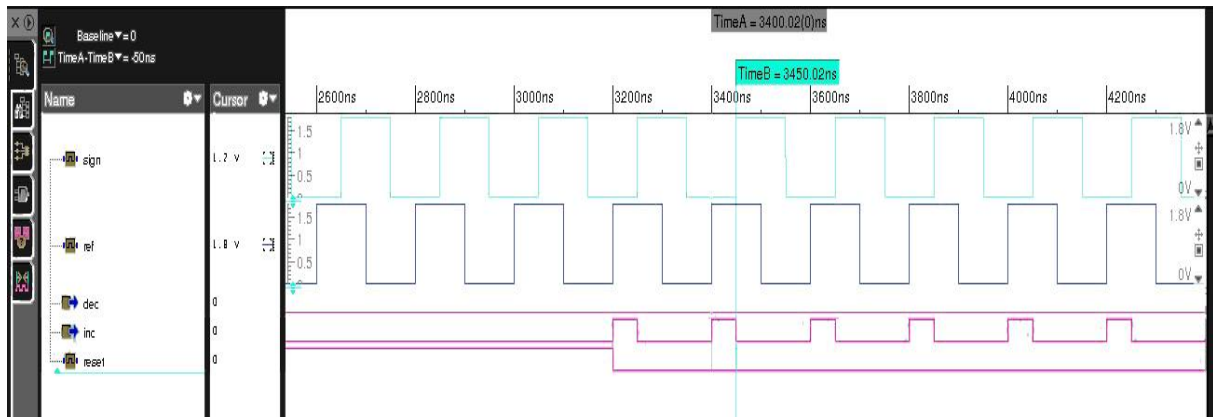
Tableau 2. 2 : valeurs utilisées dans les sources d'impulsions pour la simulation du PFD

	T1 (signal/fd)	T2 (ref)	d1	d2
Premier cas	<b>200 ns</b>	<b>200 ns</b>	<b>0</b>	<b>50ns</b>
Deuxième cas	<b>200 ns</b>	<b>200 ns</b>	<b>50ns</b>	<b>0</b>
Troisième cas	<b>200 ns</b>	<b>200 ns</b>	<b>0</b>	<b>0</b>
Quatrième cas	<b>200ns</b>	<b>300ns</b>	<b>0</b>	<b>0</b>
Cinquième cas	<b>100 ns</b>	<b>200 ns</b>	<b>0</b>	<b>0</b>
Sixième cas	<b>100 ns</b>	<b>200 ns</b>	<b>50 ns</b>	<b>0</b>

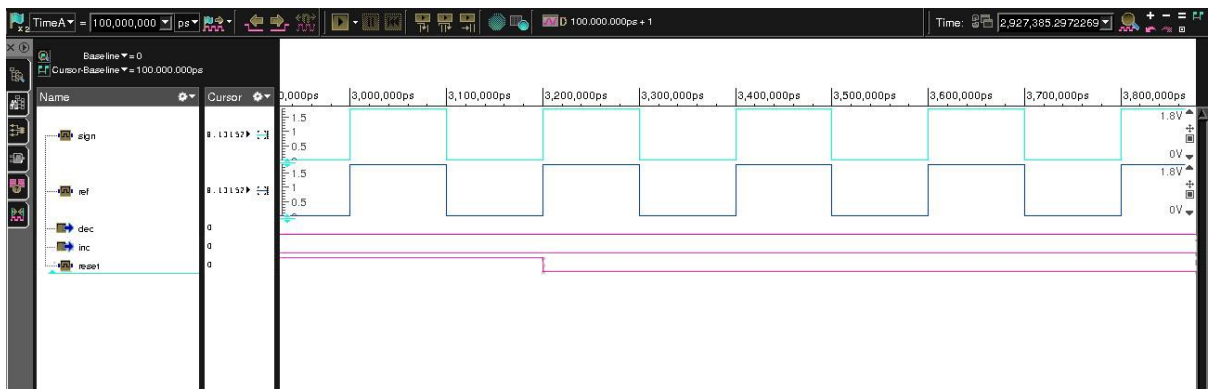
Les résultats de la simulation des différents scénarios sont illustrés dans la figure 2.7. Dans cette figure les signaux présentés sont respectivement de haut en bas, fd, ref, down, up et le reset. On peut constater que les résultats sont en accord avec ce qui était attendu. On observe que l'erreur de phase est exprimée dans le rapport cyclique de l'un des signaux up et down, ce qui valide le modèle. En effet, l'erreur entre les deux sources d'impulsion se manifeste directement par un front montant sur la sortie 'up' ou par un front descendant sur la sortie 'down'.



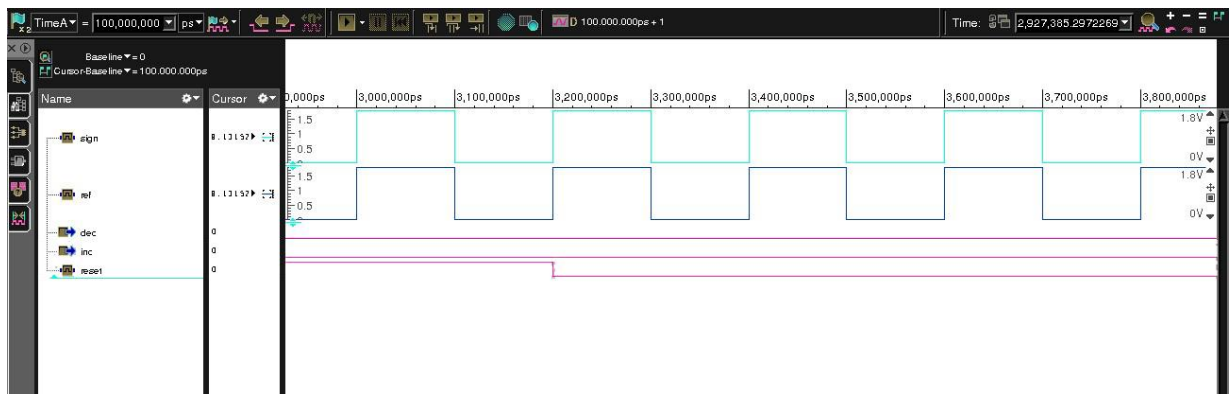
(a) Premier cas : le signal est en avance sur la référence



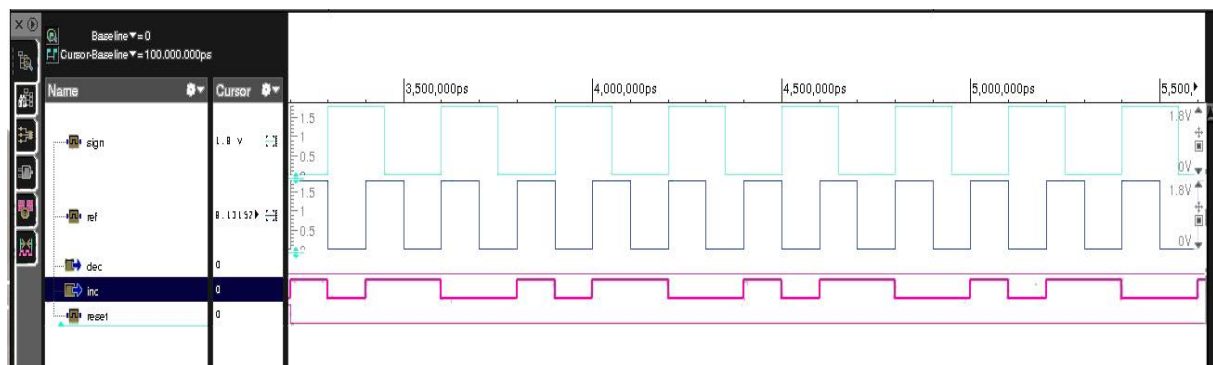
(b) Deuxième cas : la référence est en avance sur le signal



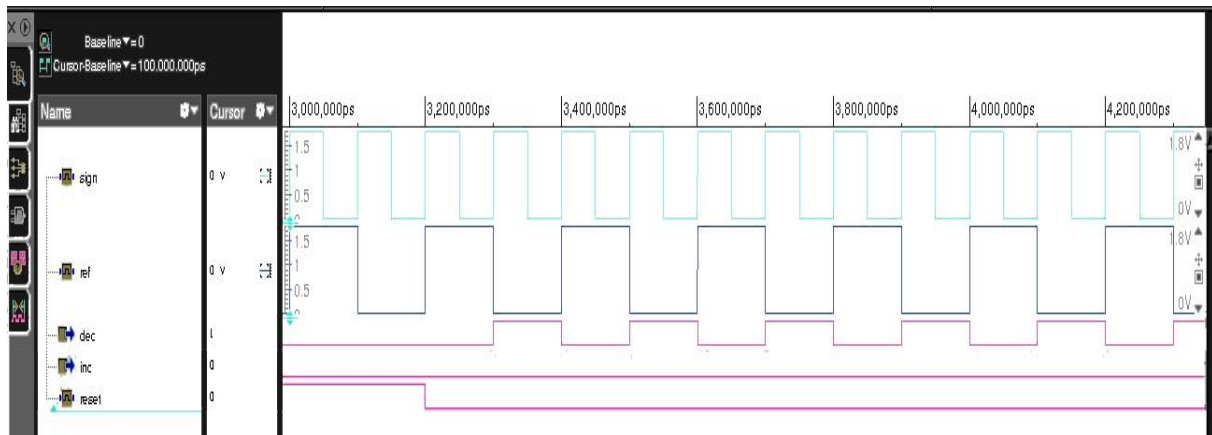
(c) Troisième cas : la référence et le signal ont la même phase



(d) Quatrième cas : la fréquence de la référence est plus grande que celle du signal.



(e) Cinquième cas : la fréquence du signal est plus grande que celle de la référence en phase.



(f) Sixième cas : la fréquence du signal est plus grande que celle de la référence avec déphasage

Figure 2. 7: Résultat de la simulation du modèle comportementale du PFD

### 2.4.3. Conception de la pompe de charge

La pompe de charges est modélisée par un modèle mixte où deux sources de courants qui ont des sorties analogiques sont contrôlées par deux signaux up et down de nature digitale. Le premier signal commande un courant de valeur “  $I_0$  ”, tandis que le deuxième commande un courant “  $- I_0$  ”, la figure 2.8 illustre le bloc en question. Son code Verilog-AMS est donné ci-dessous.

Pour la validation de ce modèle nous avons connecté à ses entrées des sources d’impulsions, de même période mais et en opposition de phase. La figure 2.9 présente les résultats de la simulation, où les signaux en question sont respectivement de haut en bas, down, up, la chute de tension à travers la résistance et le courant. On observe que la forme d’onde du courant est d’amplitude  $I_0$  constante.

```

include "constants.vams"
include "disciplines.vams"
timescale 10ps / 1ps

module cp (pout, nout, up, down);
  parameter real I_0 = 50e-6; // output current (A)
  input up, down;
  electrical pout, nout;
  real out;

  analog begin
    @(initial_step) out = 0.0;

    if (down && !up)
      out = -I_0;
    else if (!down && up)
      out = I_0;
    else out = 0;

    I(pout, nout) <+ -transition(out, 0.0, 1f, 1f);
  end
endmodule

```

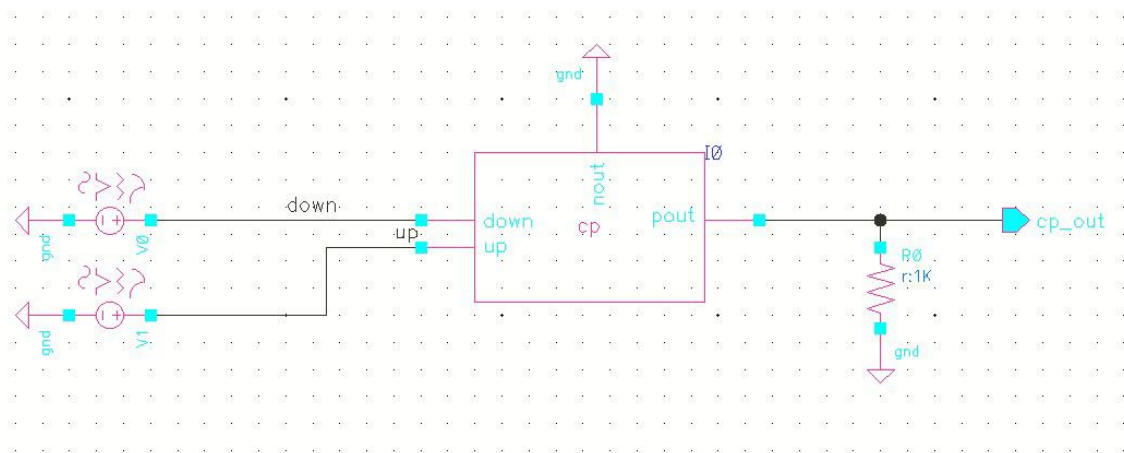


Figure 2. 8 : Setup de simulation de la pompe de charge

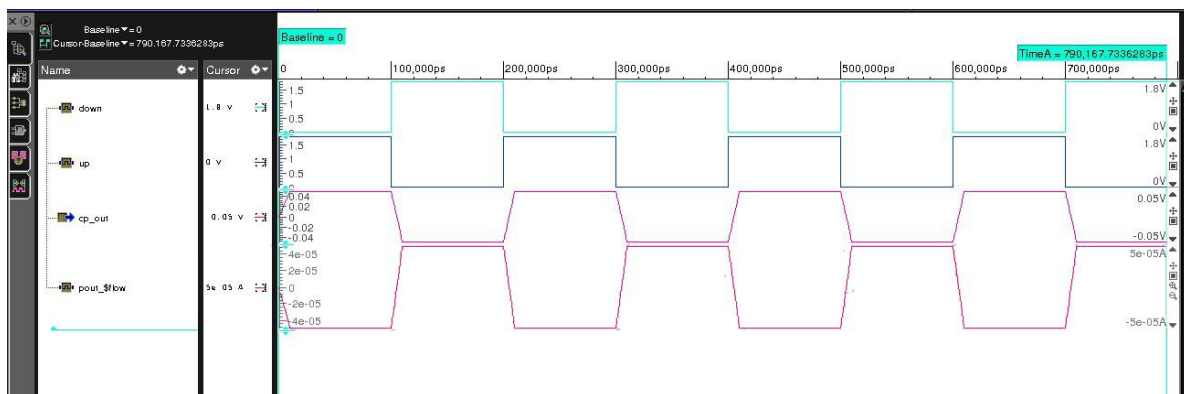


Figure 2. 9: Signaux de simulation de la pompe de charges

#### 2.4.4. Conception du Filtre de boucle

Le filtre est modélisé par des composants parfaits de la bibliothèque analogique, les valeurs des composants sont déduites des spécifications niveau systèmes du synthétiseur que nous avons décrit dans la section 2.3.

#### 2.4.5. Conception de l'oscillateur

Le VCO est modélisé par sa caractéristique tension-fréquence, et par sa sortie sinusoïdale, son code Verilog-AMS est le suivant,

```
`include "constants.vams"
`include "disciplines.vams"
`timescale 10ps/1ps

module vco_me(vtune,out_vco);
    output out_vco;
    input vtune;
    electrical out_vco,vtune;

    parameter real freq=2.38e9;
    parameter real df=160e6;

    real phase=0.0;

    analog begin
        phase = idtmod(freq + df*V(vtune), 0, 1, 0);
        V(out_vco) <+ 0.55*sin(2*'M_PI*phase);
        $bound_step(0.1/freq)
    end
endmodule
```

Pour valider ce modèle, nous avons utilisé la simulation “transient” et “PSS” du simulateur “spectre”. La première a été utilisée pour tracer la représentation temporelle de la tension de sortie du VCO (figure 2.11), et la deuxième pour tracer la fréquence de sortie du VCO en fonction de la variation de la tension de contrôle “vtune” (figure 2.12).





Figure 2. 10 : Setup de simulation du VCO

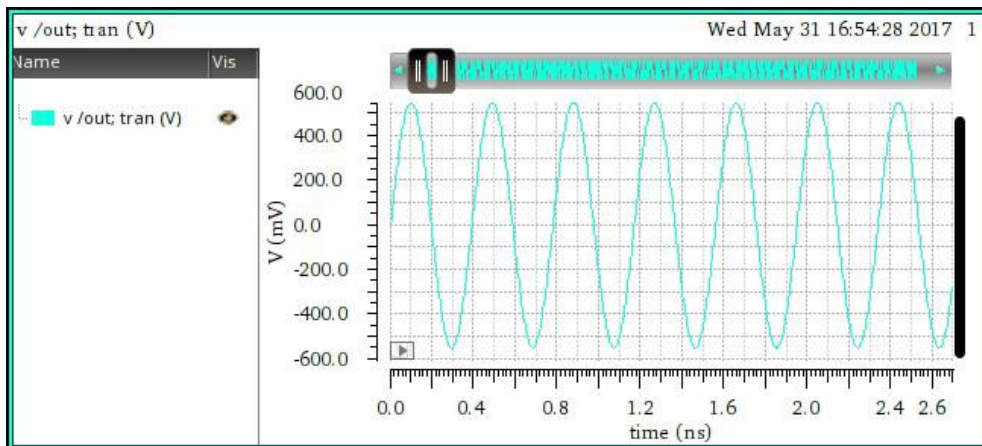


Figure 2. 11 : Simulation comportementale du VCO

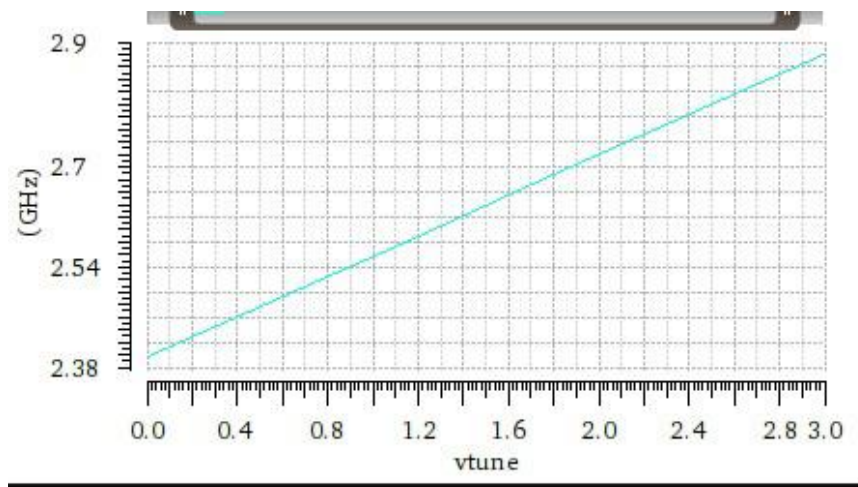


Figure 2. 12: Tension de sortie en fonction de la tension de contrôle

### 2.4.5. Conception du diviseur de fréquence

Le diviseur de fréquence compte les fronts montants et descendants du signal d'entrée et change la valeur de sa sortie à chaque fin de compte, la valeur du compteur est contrôlée par un vecteur de 4 bits. Le code que nous avons élaboré est présenté comme suit.

```

`timescale 10ps / 1ps

module fd_16(out, clk, reset, N);
    input clk, reset;
    input [3:0] N ;
    output out;
    wire out;
    reg q;
    integer i;
    integer t;
    always @(negedge reset) begin
        i = 0;
        q = 0;
    end

    always @(N[3:0]) begin
        case (N[3:0])
            4'b0000 : t = 480;
            4'b0001 : t = 481;
            4'b0010 : t = 482;
            4'b0011 : t = 483;
            4'b0100 : t = 484;
            4'b0101 : t = 485;
            4'b0110 : t = 486;
            4'b0111 : t = 487;
            4'b1000 : t = 488;
            4'b1001 : t = 489;
            4'b1010 : t = 490;
            4'b1011 : t = 491;
            4'b1100 : t = 492;
            4'b1101 : t = 493;
            4'b1110 : t = 494;
        endcase
    end

    always @(posedge clk or negedge clk) begin
        if (~reset) begin
            i = i + 1;
            if (i == t) begin
                q = ~q;
                i = 0;
            end
        end
    end

    assign out = q & ~reset;
endmodule

```

Le testbench utilisé pour la vérification du modèle du diviseur est donné par la figure 2.13. Les signaux dans ce cas sont, un signal “clk” qui représente la fréquence à diviser, un module “N\_const” qui sert à changer le rapport de division, et un module “reset” qui sert à initialiser le diviseur.

La figure 2.14(a) montre la forme d'onde du signal à l'entrée du diviseur où sa période est de 200 ns, tandis que la figure 2.15(b) illustre la forme d'onde à la sortie du diviseur avec une période de 97.2 us. Le rapport entre la période de sortie et de l'entrée du diviseur est de 486 ps, ce qui est égale au rapport de division programmée.

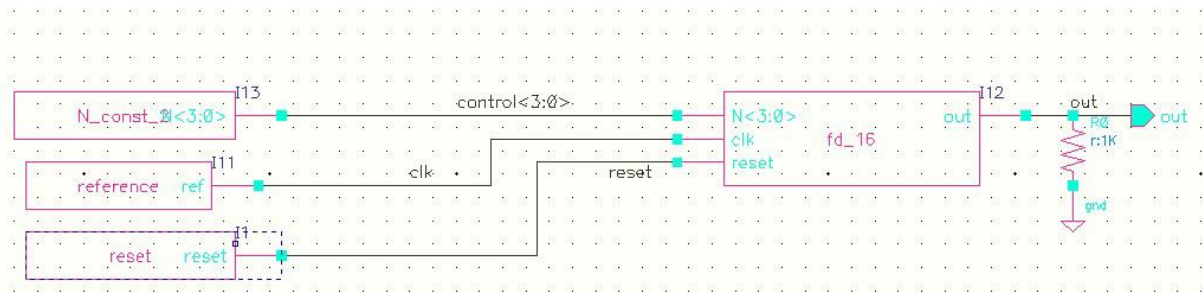
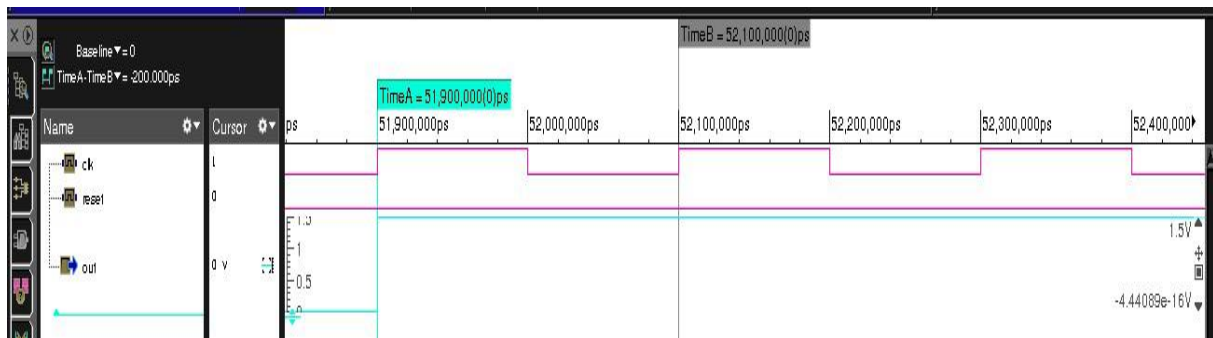
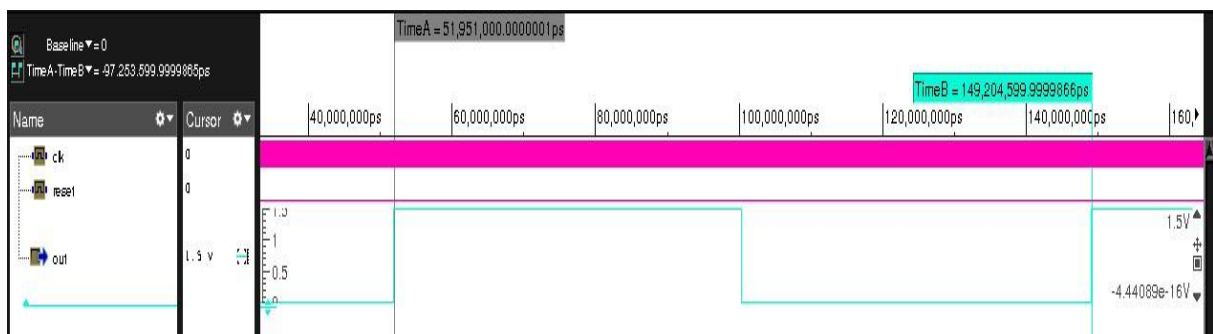


Figure 2. 13: Simulation comportementale du diviseur de fréquence



(a)



(b)

Figure 2. 14: Simulation du modèle comportementale du diviseur. (a) : entrée du diviseur. (b) : sortie du diviseur.

## 2.5. Etude comportementale du système complet

Une fois que les modèles des blocs sont vérifiés individuellement, il faut les connecter pour créer le modèle du système complet, le schéma de la simulation relatif au synthétiseur entier est présenté dans la figure 2.15.

### 2.4.1. Paramétrage de l'outil de simulation

En vue de vérifier le bon fonctionnement du système complet, nous avons effectué une analyse transitoire à l'aide du simulateur AMS. Afin de dérouler le fonctionnement du système nous avons procédé dans le testbench selon les étapes suivantes :

1. Nous avons fixé la période du signal de référence à 200 ns, ce qui est équivalent à une fréquence de 5 MHz. Cette valeur est généralement générée par un oscillateur à quartz.
2. Nous avons fixé la valeur du diviseur de fréquence à 480 pour le démarrage de la PLL,
3. Nous avons changé le rapport de la division d'une manière à avoir la fréquence du canal maximum 2480 MHz, soit  $N = 2480/5 = 496$ .
4. Nous avons commencé à décrémenter le rapport de division, ce qui est équivalent à choisir le canal adjacent.

### 2.4.2. Résultats préliminaires de la simulation

Les résultats de la simulation sont présentés dans la figure 2.16. On peut observer d'une manière générale que les blocs fonctionnent correctement, du fait que le système présente un asservissement adéquat. De plus, la méthode adoptée pour la commande des canaux en changeant le rapport N a permis de voir clairement que tout le système suit les changements apportés. En effet, en gardant la référence constante tout en modifiant le rapport de division N a permis d'obtenir plusieurs paliers qui se sont bien manifestés sur la courbe rouge qui représente l'erreur qui servira de tension de contrôle pour le VCO.

Les résultats de cette simulation se divisent en trois parties, le démarrage de la PLL, l'acquisition de la fréquence maximale, et la commande des canaux. Ces trois phases sont illustrées respectivement dans les figures 2.17 à 2.19. La phase de démarrage est relative au temps nécessaire pour que le VCO commence à osciller, dans ce cas ce dernier commence à osciller avec sa fréquence libre. La phase d'acquisition de la fréquence correspond au démarrage du mécanisme de l'asservissement jusqu'à l'obtention de la fréquence désirée.

On peut noter pour la phase de commande des canaux qu'avec la variation des valeurs de  $N$ , la fréquence à la sortie du VCO reste toujours multiple de la fréquence de référence avec un facteur égale à  $N$  ce qui n'est rien d'autre que le rapport de division du diviseur de fréquence (figure 2.19). Ceci confirme davantage le bon fonctionnement du synthétiseur.

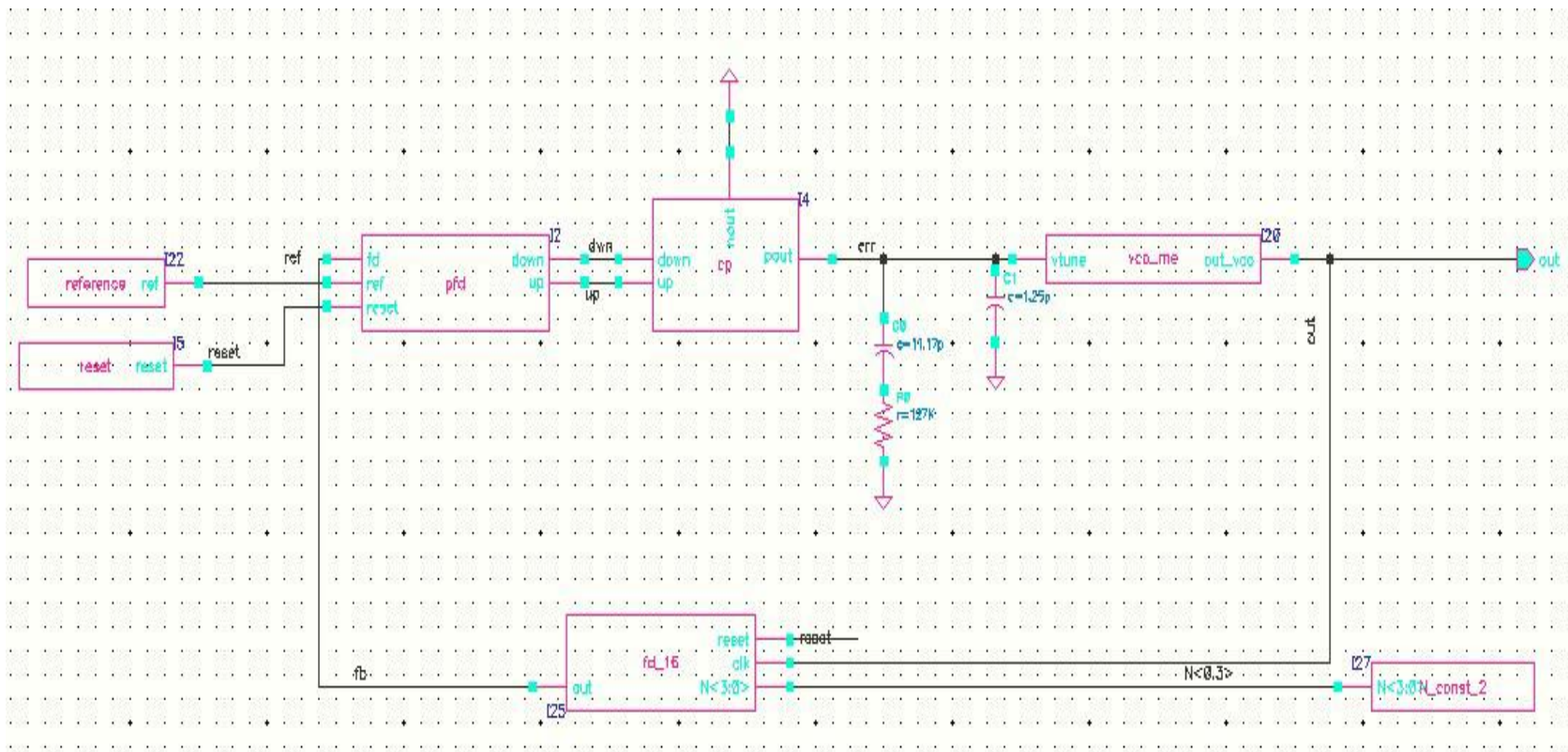


Figure 2. 15 : Setup de simulation du synthétiseur de fréquence

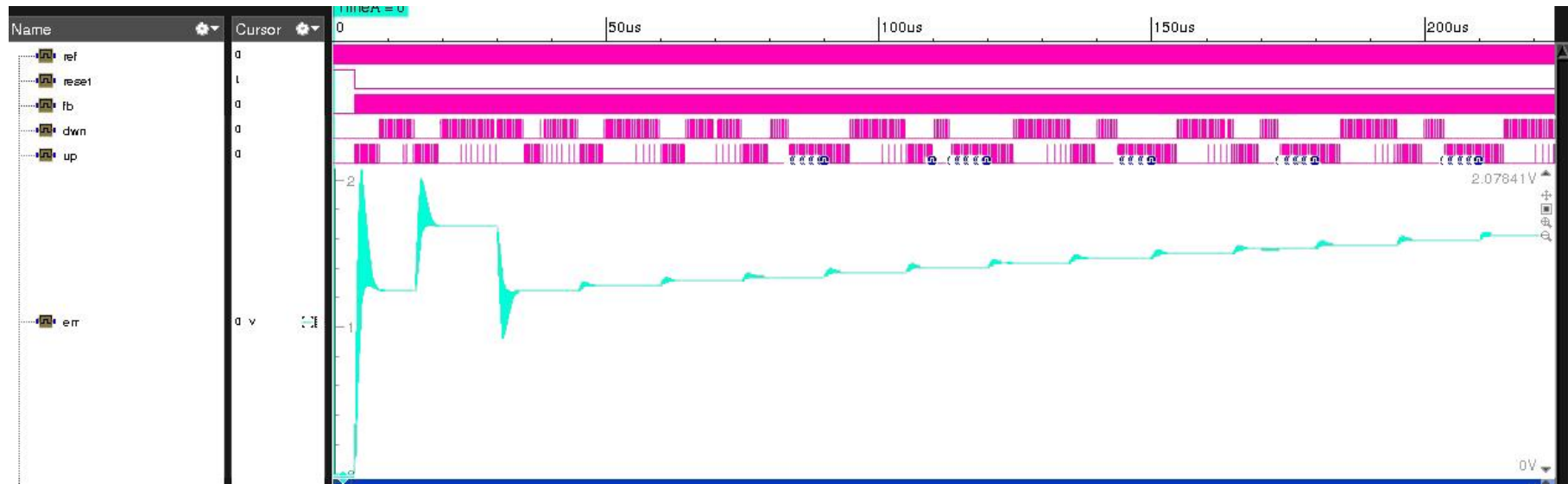
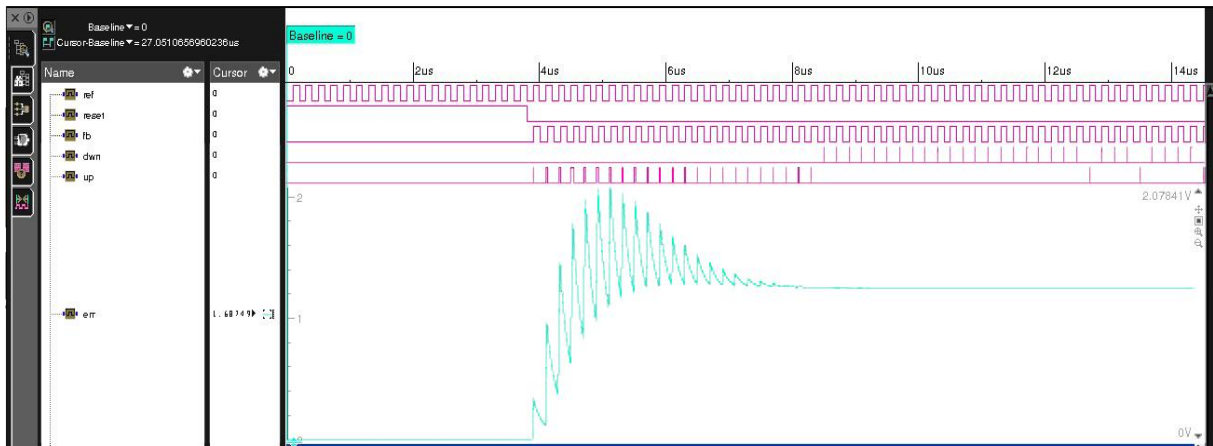
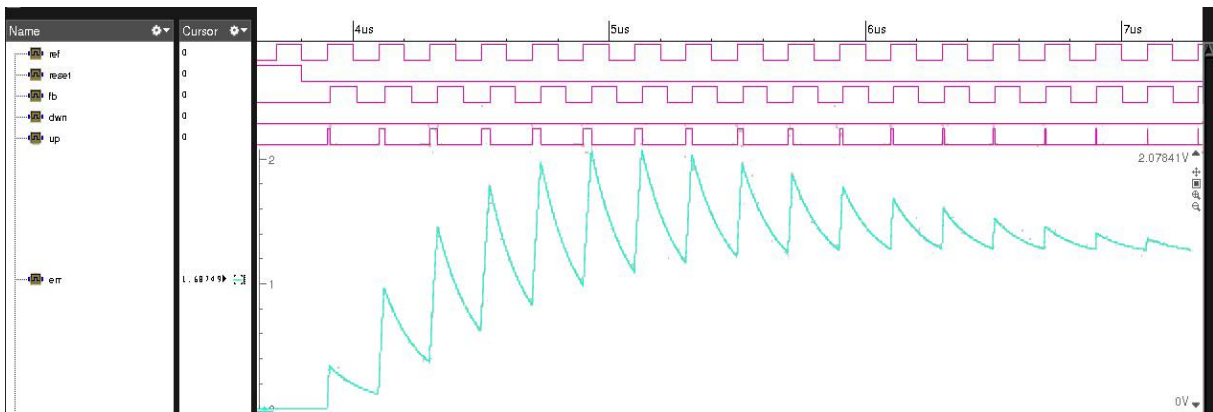


Figure 2. 16: Simulation du Synthétiseur. Les signaux de haut en bas sont : Référence, reset, sortie du diviseur, up et down de la pompe de charge, l'erreur (tension de contrôle)



(a)



(b)

Figure 2. 17 : Phase de démarrage du synthétiseur

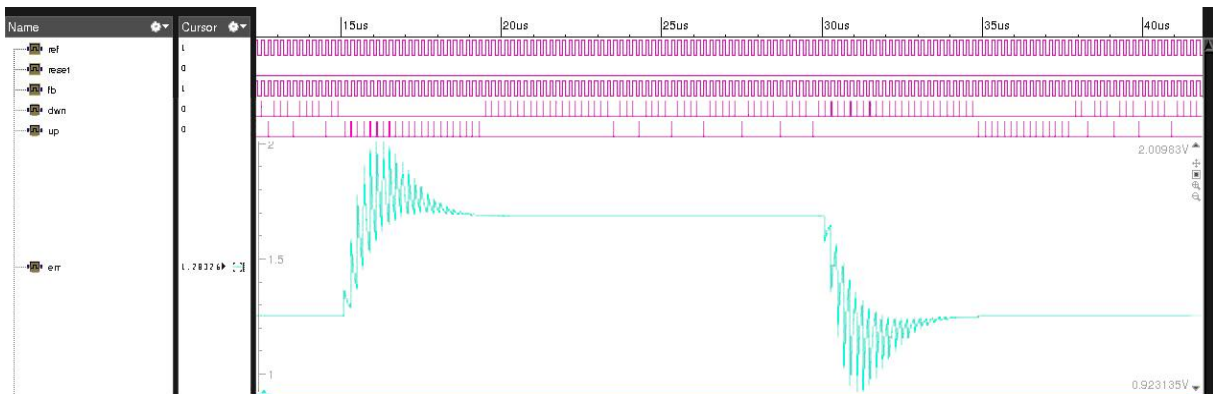


Figure 2. 18: Phase d'acquisition de la fréquence



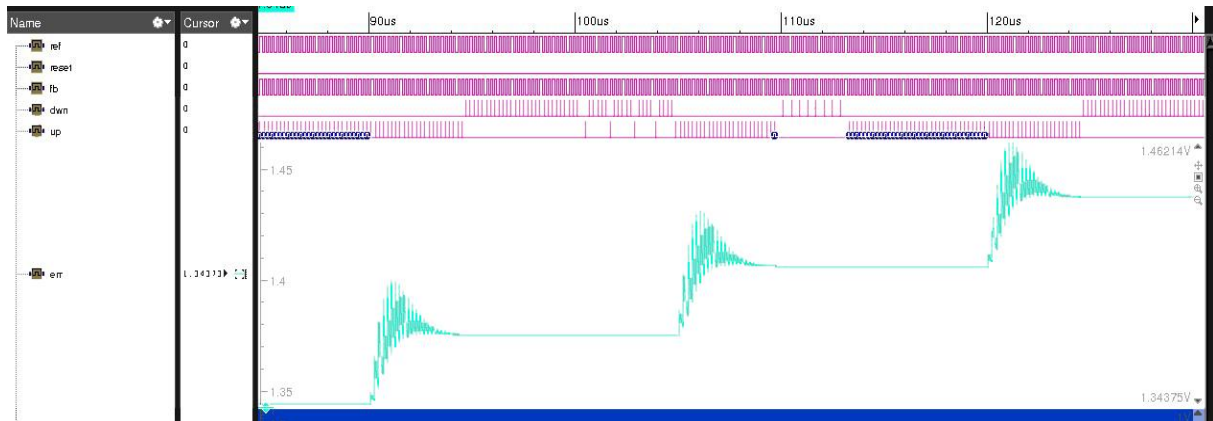


Figure 2. 19: Phase de la commande des canaux (variation du rapport N)

### 2.4.3. Niveau des parasites sur la tension de contrôle

En effectuant un zoom sur l'un des états stables du signal d'erreur (tension de contrôle du VCO) relative à la phase d'acquisition du verrouillage, on constate que la valeur de cette tension n'est pas tout à fait constante mais elle présente quelques fluctuations (figure 2.20). Ces parasites sont dues au fait que les transitions de la pompe de charge ne sont pas idéales (figure 2.9). En effet, les fronts montant et descendant présentent un temps de retard ce qui donne naissance à ce type de parasites.

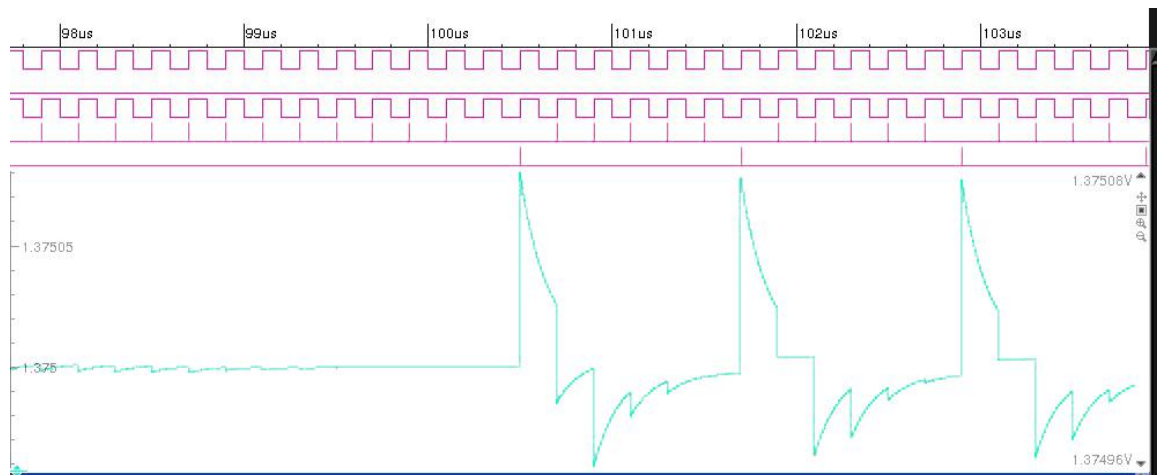


Figure 2. 20: Niveau des parasites sur la tension de contrôle

## 2.6. Conclusion

Dans ce chapitre nous avons abordé l'étude comportementale du synthétiseur de fréquence répondant au standard Zigbee. Nous avons pu constater qu'il est indispensable de déduire le modèle de référence du système en question avant d'entamer sa conception comportementale. Nous avons effectué cette étape à l'aide de la programmation du modèle en question sous MATLAB où nous avons réussi à déduire des paramètres propres aux systèmes tout en considérant les spécifications dictées par le standard ciblé.

Dans un second temps nous avons procédé à la conception comportementale des éléments du synthétiseur en se basant sur le modèle de référence et sur les modèles introduits dans le chapitre 1. La validation de la conception a été effectuée en utilisant l'environnement de simulation sous l'outil de CAO Cadence.

Les résultats de simulation ont montré un bon fonctionnement du synthétiseur, et ce pour différentes valeurs de division  $N$ . Ainsi, la conception niveau système complétée, nous pouvons passer au remplacement des éléments comportementaux par des circuits d'un niveau d'abstraction bas. Cette substitution s'effectue bloc par bloc en commençant par les blocs analogiques puis ceux numériques, cette opération va faire l'objet des deux prochains chapitres.

# **Chapitre 3**

## **Conception d'un synthétiseur Mixte basé sur un VCO en technologie CMOS**

### 3.1. Introduction

Nous avons vu dans le chapitre précédent la conception comportementale du synthétiseur et comment il a été validé avec une simulation moyennant le simulateur AMS sur Cadence. Ainsi, nous pouvons descendre dans niveau d'abstraction plus bas en remplaçant les blocs comportementaux par des blocs à base de circuits implémentés sur une technologie monolithique.

Dans ce chapitre nous allons remplacer l'oscillateur du synthétiseur par un VCO du type LC intégré sur la technologie CMOS 0.18- $\mu\text{m}$ . Après l'insertion de ce dernier dans le synthétiseur nous pouvons dire que nous avons réalisé un système mixte.

Mais avant cette insertion, nous allons exposer avec détail la conception de ce VCO en présentant toutes les étapes de conception nécessaires à sa concrétisation. Néanmoins, dans ce cadre la technologie d'implémentation mérite tout d'abord qu'elle soit présentée.

### 3.2. La technologie CMOS

La technologie CMOS (*Complementary Metal Oxide Semiconductor*) est la technologie prépondérante dans le domaine de la conception des circuits intégrés analogiques, numériques et mixtes.

Selon l'ITRS (*International Technology Roadmap for Semiconductors*), cette suprématie se perpétuera pour les prochaines années vu que les champs d'applications de ladite technologie ne cessent d'augmenter. En effet, cela revient à ses caractéristiques offertes telles qu'un faible coût de fabrication, une évolution continue, des vitesses de fonctionnement très élevées et une puissance dissipée relativement faible [19].

#### 3.2.1. Les dispositifs en technologies CMOS

##### 3.2.1.1. Le MOSFET

Le MOSFET (*Metal-Oxide-Semiconductor Field-Effect-Transistor*) est le composant qui domine l'industrie de la microélectronique. Conçu dans les années 1930s, mais réalisé pour la première fois dans les années 60, le MOSFET offre des propriétés uniques qui ont menées vers la révolution de l'industrie des semi-conducteurs. Cette révolution a abouti à des microprocesseurs ayant 100 millions de transistors, des puces mémoires contenant des

milliards de transistors et des circuits de communication sophistiqués offrant une capacité de traitement de signal énorme [24].

La structure du MOSFET est présentée dans la figure 3.1. Il est constitué de quatre terminaux, grille (G), Drain (D), Source (S) et le Substrat (B). La grille est fabriquée de polysilicium (poly) avec un dopage fort, la couche diélectrique entre la grille et le substrat qui joue un rôle critique dans les performances du transistor est créé par le dioxyde de silicium (oxide) mis sur la surface du silicium.

Le comportement électrique du MOSFET dépend essentiellement de la polarité des tensions grille-source ( $V_{gs}$ ) et drain-source ( $V_{ds}$ ) appliquées. En appliquant une tension  $V_{gs}$  supérieure à la tension de seuil  $V_T$ , des charges positives s'accumulent au-dessus de l'oxyde, et attirent ainsi les électrons minoritaires, dans le substrat p, en créant un canal de conduction entre la source et le drain. Pour un transistor NMOS, lorsqu'une tension  $V_{ds}$  est appliquée, un mouvement d'électrons de S vers D se déclenche générant ainsi un courant  $I_d$  [19]. Il existe deux zones de fonctionnement pour le MOSFET.

- 1. Zone triode** : dans cette zone nous avons  $V_{gs} > V_{th}$  et  $V_{ds} < (V_{gs} - V_{th})$ . L'expression du courant de drain  $I_{ds}$  est de la forme suivante,

$$I_{ds} = k_n [(V_{gs} - V_{th}) V_{ds} - \frac{V_{ds}^2}{2}] \quad (3.1)$$

Avec :  $k_n = \mu_n C_{ox} \frac{W}{L}$

$V_{th}$ ,  $C_{ox}$ ,  $W$ ,  $L$ ,  $\mu_n$  sont respectivement, la tension de seuil, la capacité d'oxyde de grille par unité de surface en ( $F/m^2$ ), la largeur et la longueur de grille du transistor MOSFET en (m) et la mobilité des électrons en ( $m^2/V.s$ )

Pour des valeurs de  $V_{ds} \ll 2(V_{gs} - V_{th})$  on peut écrire  $I_{ds}$  comme exprimé dans (3.2) où on peut assimiler le MOSFET à une admittance dont la valeur est contrôlé par  $V_{gs}$ .

$$I_{ds} = k_n (V_{gs} - V_{th}) V_{ds} \quad (3.2)$$

- 2. Zone de saturation** : dans cette zone nous avons  $V_{gs} > V_{th}$ , et  $V_{ds} > (V_{gs} - V_{th})$ . Dans cette zone, le courant  $I_{ds}$  est maximale, son expression idéale est donnée par la formule (3.3). Le MOSFET dans cette zone est considéré comme une source de courant contrôlée par la tension  $V_{gs}$ .

$$I_{ds} = \frac{k_n}{2} (V_{gs} - V_{th})^2 \quad (3.3)$$

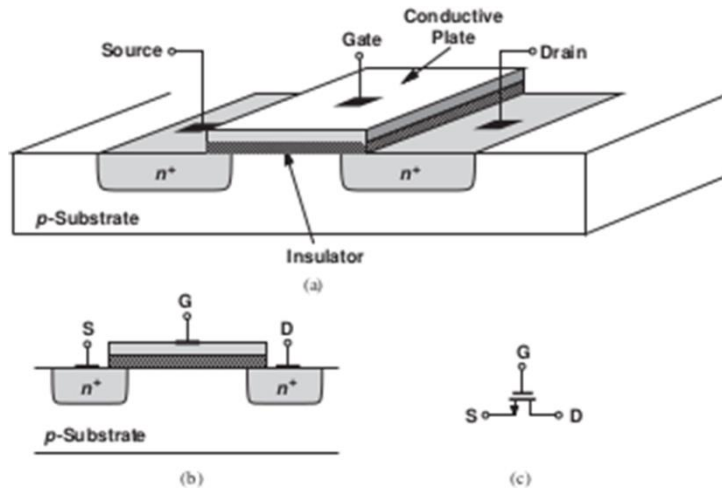


Figure 3. 1 : Physique du transistor MOSFET

Un autre paramètre très important du transistor MOSFET est la transconductance  $g_m$ . C'est la variation du courant de drain par rapport à la tension de grille et s'exprime par (3.4) et (3.5).

$$g_m = \frac{\partial I_d}{\partial V_{gs}} = k_n (V_{gs} - V_{th}) \quad (3.4)$$

$$g_m = \sqrt{2k_n I_d} \quad (3.5)$$

### 3.2.1.2. Résistance

La résistance d'un composant de résistivité  $\rho$ , de longueur  $l$ , d'épaisseur  $d$  et de largeur  $w$  est exprimée par,

$$R = \frac{\rho}{d} \frac{l}{w} \quad (3.6)$$

Le premier facteur est défini par la technologie et le deuxième peut être fixé par le concepteur. Afin de minimiser la capacité parasite,  $w$  est choisi aussi petit que possible par rapport au courant à manipuler, finalement le paramètre libre est le  $l$ . On cite deux implémentations possibles pour la résistance en technologie CMOS [25], à savoir, la résistance au film mince (figure 3.2) et la résistance implantée (figure 3.3).

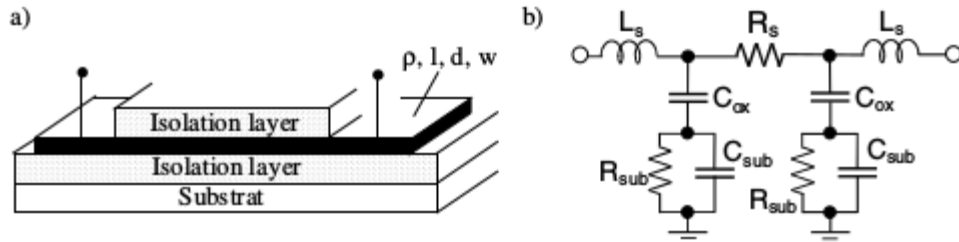


Figure 3. 2 : Résistance à film mince. (a) structure sur silicium. (b) schéma équivalent en technologie CMOS.

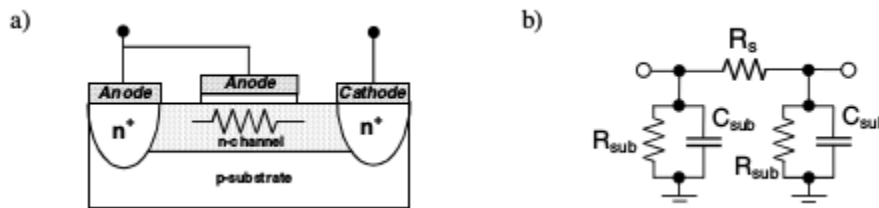


Figure 3. 3: Résistance implantée. (a) structure sur silicium. (b) schéma équivalent.

### 3.2.1.3. Capacité

Parmi les implémentations de capacité disponibles en technologie CMOS on trouve la capacité MOS et la capacité MIM.

- **Capacité MOS**

La capacité MOS n'est qu'un MOSFET dont le drain et la source sont connectés entre eux. Les capacités MOS présentent une capacité de substrat parasite élevée résultant en une perte d'insertion élevée lorsqu'ils sont utilisés pour des connexions en série. Elles ne sont donc pas adaptées aux condensateurs DC-block ou d'adaptation d'impédance, mais ils fournissent une capacité très élevée par unité de surface, ce qui les rend adéquates pour les shunts RF compactes. Une vue en coupe et un circuit équivalent sont présentés dans la figure 3.4.

- **Capacité MIM**

La capacité MIM (Metal Insulator Metal) a de faibles parasites par rapport aux capacités MOS. Elle est composée de deux armatures de conducteurs parallèles séparées par un isolant. La valeur de la capacité est donnée par  $C_s = \frac{\epsilon_0 A}{d} \epsilon_r$ . Où  $A$  représente l'aire de l'armature,  $\epsilon_0$  la permittivité du vide,  $\epsilon_r$  la permittivité relative du matériau et  $d$  l'épaisseur de l'isolant. Une vue en coupe et un circuit équivalent sont présentés dans la figure 3.5.

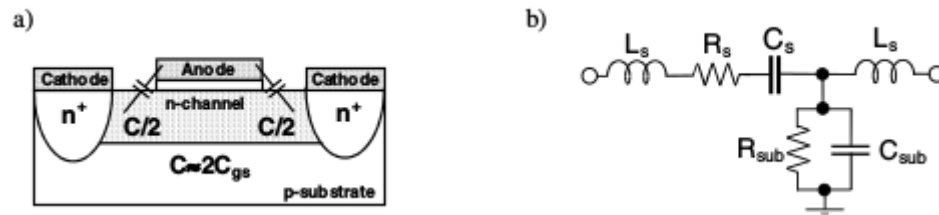


Figure 3. 4: Capacité MOS. (a) structure en technologie CMOS. (b) Schéma équivalent.

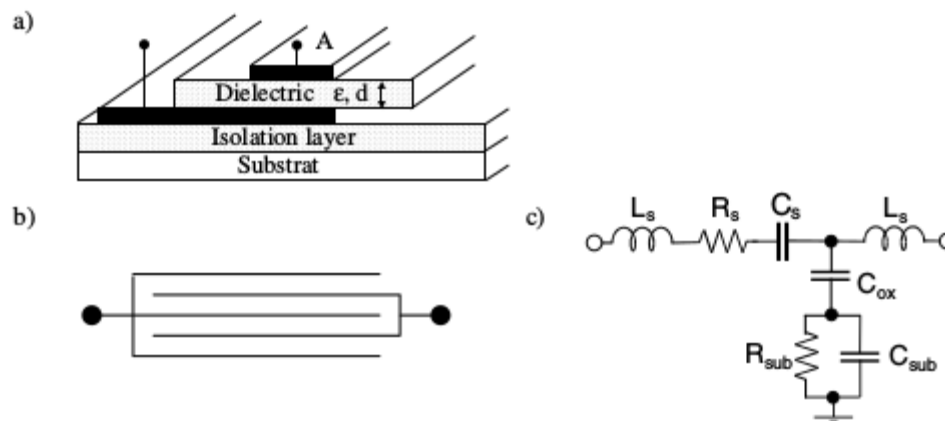


Figure 3. 5 : Capacité MIM. (a) Structure sur Silicium. (b) Symbole. (c) Schéma équivalent.

### 3.2.1.4. Inductance

En technologie CMOS, l'inductance RF, est réalisée à base de ligne de transmission d'impédance caractéristique élevée. Une implémentation de l'inductance et son modèle électrique sont présentés dans la figure 3.6. Les paramètres de l'inductance sont définis comme suit,

- $W$  : la largeur de la piste.
- $S$  : l'espace entre deux pistes métalliques adjacentes.
- $N$  : le nombre de tours
- $R$  : le rayon intérieur de l'inductance.

Dimensionner l'inductance, consiste à déterminer les paramètres géométriques exposés ci-dessus, et ce, à partir des valeurs de l'inductance  $L$  et de son facteur de qualité  $Q_L$ , qui sont déduits à partir de la conception [19]. Ces deux paramètres  $L$  et  $Q_L$  peuvent s'exprimer en fonction des paramètres de répartition  $S_{ij}$  ou des impédances  $Z_{ij}$ .



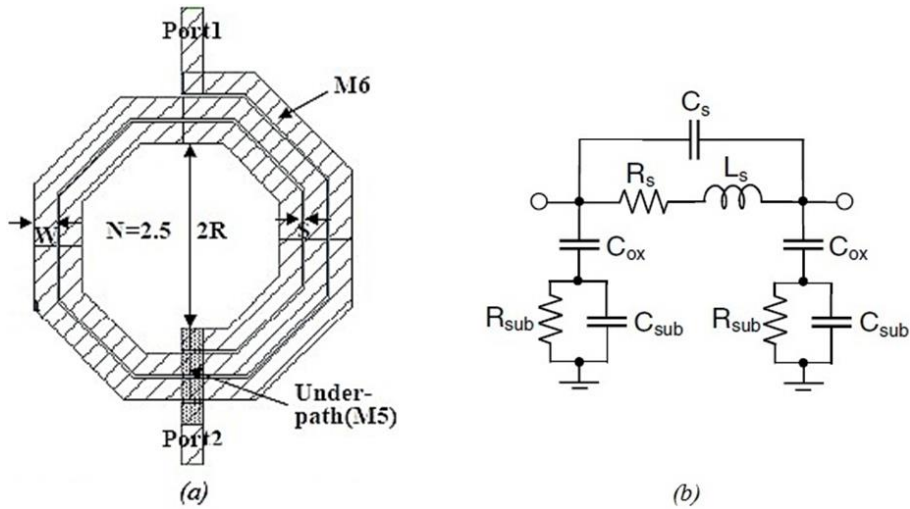


Figure 3. 6 : Layout et schéma équivalent de l'inductance RF en technologie CMOS

### 3.2.1.5 Varactors

Un Varactor est une capacité variable contrôlée par la tension appliquée à ses bornes. Sa valeur varie donc entre sa capacité maximale et minimale  $C_{vmax}$ ,  $C_{vmin}$  respectivement. Il existe deux types de varactors intégrables en technologie CMOS, à savoir, les varactors diode ou jonction P+/N polarisée en inverse et les varactors MOS.

Le varactor diode présente de faible facteur de qualité et rapport de variation, ce qui les rend moins adaptés aux applications RF [26]. Cependant, le varactor MOS est réalisé en connectant la source, le drain et le substrat ensemble, la valeur de la capacité est contrôlée par la tension  $V_{gb}$ , la structure et le symbole de ce varactor sont présentés dans la figure 3.7.

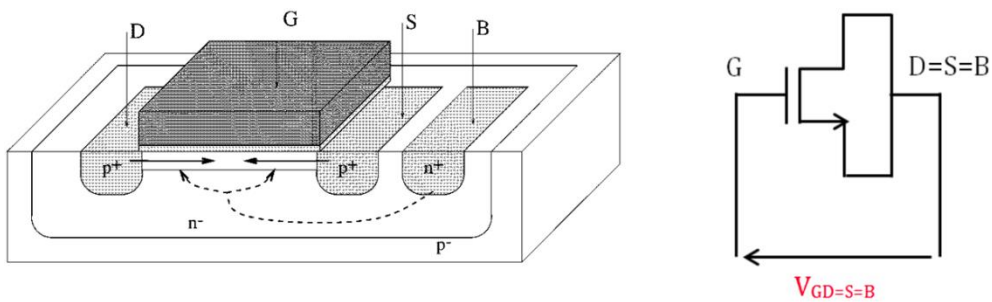


Figure 3. 7: Structure et symbole d'une capacité MOS

### 3.3. Notions sur les oscillateurs

#### 3.3.1. Théorie des oscillateurs

Razavi a défini l'oscillateur comme étant un amplificateur mal conçu [27] qui mène vers un système à boucle négative de gain unitaire, cette définition est présentée à la figure 3.8. En se basant sur cette figure, la fonction de transfert d'un oscillateur peut s'écrire par l'expression ci-dessous.

$$\frac{V_{out}(s)}{V_{in}} = \frac{H(s)}{1 + H(s)} \quad (3.7)$$

Avec  $H(s)$  la fonction de transfert du système en boucle ouverte. Les conditions de Barkhausen pour qu'un tel système produise et maintienne des oscillations sont,

$$\begin{cases} |H(j\omega_0)| \geq 1 \\ \arg(H(j\omega_0)) = 180^\circ \end{cases} \quad (3.8)$$

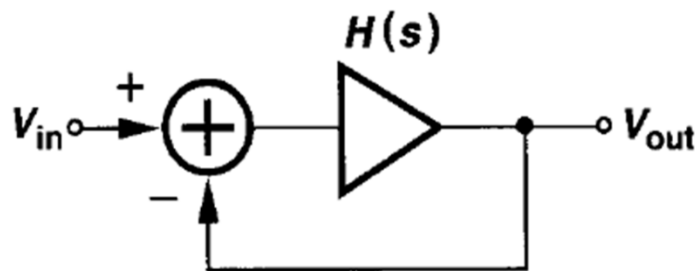


Figure 3. 8 : Modélisation d'un oscillateur

Sous les conditions de (3.9), l'oscillateur amplifie son propre bruit à la fréquence  $\omega_0$  [27], en effet la composante de bruit subit un gain unitaire et un déphasage de  $180^\circ$ . Le signal à la sortie du soustracteur donne une large différence ce qui fait grandir la composante du bruit, ce mécanisme est illustré dans la figure 3.9.

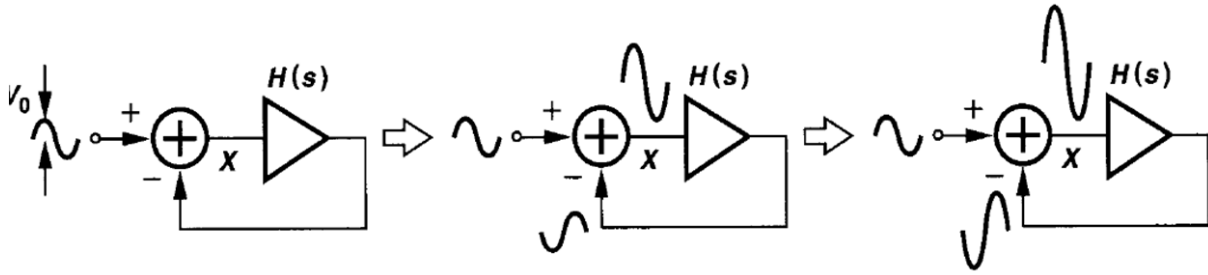


Figure 3. 9: Mécanisme d'amplification et maintien du régime d'oscillation

### 3.3.2. Topologies des oscillateurs

Un oscillateur peut être réalisé de plusieurs manières, le tableau 3.1 résume les topologies existantes ainsi que leurs applications [24]. Les topologies utilisées en RF sont les topologies en anneau et les topologies LC. L'oscillateur en anneau consiste en un multiple d'étages d'inverseurs de gain identique bouclés, ils sont beaucoup utilisés dans les circuits intégrés tel que les microprocesseurs et les mémoires. L'oscillateur LC quant à lui, il se compose d'un circuit RLC et d'un circuit actif qui a pour rôle de présenter à sa sortie une résistance négative

Tableau 3. 1 : Topologies des oscillateurs [24]

Oscillator Topology	LC Oscillators					
	Ring Oscillator	Cross-Coupled Oscillator	Colpitts Oscillator	Phase Shift Oscillator	Wien-Bridge Oscillator	Crystal Oscillator
Implementation	Integrated	Integrated	Discrete or Integrated	Discrete	Discrete	Discrete or Integrated
Typical Frequency Range	Up to Several Gigahertz	Up to Hundreds of Gigahertz	Up to Tens of Gigahertz	Up to a Few Megahertz	Up to a Few Megahertz	Up to About 100 MHz
Application	Microprocessors and Memories	Wireless Transceivers	Stand-Alone oscillators	Prototype Design	Prototype Design	Precise Reference

Les topologies LC présentent de meilleurs performances en termes de bruits et de plus ils ont une grande gamme de fréquence d'opération par rapport aux autres topologies [8], ce qui les rend les plus adaptés aux applications de communication sans fil qui sont très exigeants en

termes de bruits. Cependant, les topologies LC sont plus difficiles à concevoir et elles occupent une large surface de la puce [24].

Dans cette topologie il existe deux architectures principales, l'oscillateur de Colpitts et l'oscillateur à base d'une paire différentielle, ces deux architectures ainsi que l'architecture en anneau sont présentées dans la figure 3.10.

L'oscillateur Colpitts utilise un seul transistor, tandis que l'oscillateur en couplage croisé en utilise deux. La condition d'oscillation pour les deux circuits s'écrit [27] :

$$\begin{cases} g_m R_p \geq 4 & \text{pour Colpitts} \\ g_{m1} R_p g_{m2} R_p \geq 1 & \text{pour la paire différentielle} \end{cases} \quad (3.9)$$

Les équations (3.9) démontrent un désavantage majeur de l'oscillateur de Colpitts par rapport à l'oscillateur à base de la paire différentielle. Le premier demande un gain en tension quatre fois plus grand que le deuxième. Ce qui favorise l'utilisation de la cette dernière.

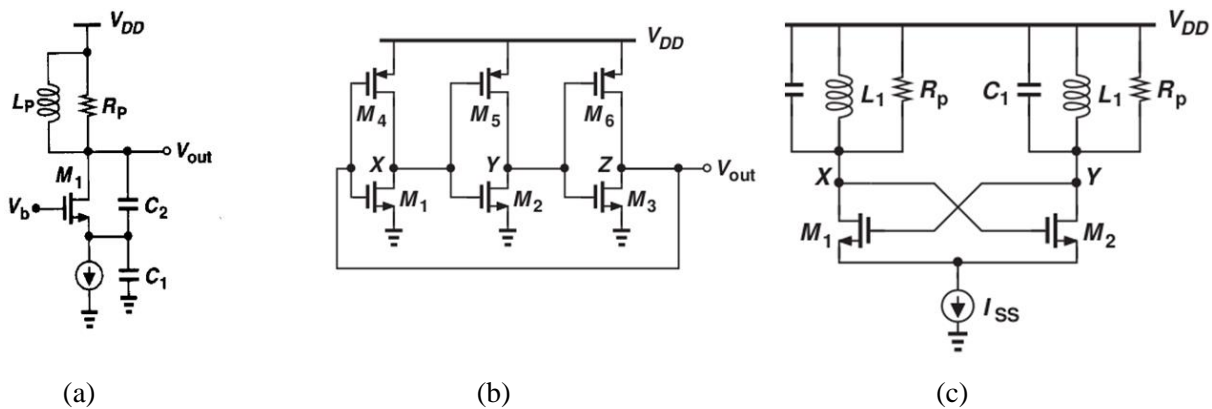


Figure 3. 10: Topologies d'oscillateur. (a) Colpitts. (b) Anneau. (c) paire différentielle

### 3.3.3. Etude de l'oscillateur LC à base d'une paire différentielle

La manière la plus simple pour comprendre le fonctionnement d'un oscillateur LC est d'étudier le modèle de la résistance négative. Considérons un circuit LC en parallèle, supposons qu'à l'état initial la capacité porte une tension  $V_0$ , la capacité commence à se décharger à travers l'inductance, dans ce cas l'énergie électrique est transformée en une énergie magnétique. A un moment donné, seule l'inductance porte de l'énergie sous forme de courant, ce courant va continuer de charger la capacité vers  $-V_0$ . Ce transfert d'énergie entre

$C_1$  et  $L_1$  (figure 3.11) se répète et le réservoir LC oscille indéfiniment. La fréquence des oscillations est donnée par,

$$f_0 = \frac{1}{2\pi\sqrt{C_1 L_1}} \quad (3.10)$$

En réalité les inductances et capacités présentent des résistances interne, ce qui entraîne une dissipation de l'énergie chaque cycle, ce qui produit une oscillation évanescence. La figure 3.11 représente la forme des oscillations dans ces deux cas.

Afin de maintenir ces oscillations, on connecte un circuit actif qui présente une résistance négative au résonateur LC de sorte que:  $(-R_1) \parallel R_p = \infty$ . La forme de la tension de sortie dans ce cas est représentée qualitativement dans la figure 3.12.

La résistance négative dans notre cas est réalisée par une paire différentielle comme le montre la figure 3.13. En modèle petits signaux, dans le cas où on applique une tension à l'entrée de la paire, le courant qui traverse les MOSFET et la tension peuvent s'écrire par :

$$\begin{aligned} I_x &= g_{m2} \cdot V_1 = -g_{m1} \cdot V_2 \\ V_x = V_2 - V_1 &= -I_x \left( \frac{1}{g_{m1}} + \frac{1}{g_{m2}} \right) \end{aligned} \quad (3.11)$$

Avec  $g_{m1}$  et  $g_{m2}$  sont les transconductances des transistors  $M_1$  et  $M_2$  respectivement. De ces deux équations on peut tirer l'expression de la résistance équivalente qui est de la forme exprimée dans (3.12) dans le cas où les deux MOSFETs sont identiques.

$$R_{négative} = \frac{V_x}{I_x} = - \left( \frac{1}{g_{m1}} + \frac{1}{g_{m2}} \right) = - \frac{2}{g_m} \quad (3.12)$$

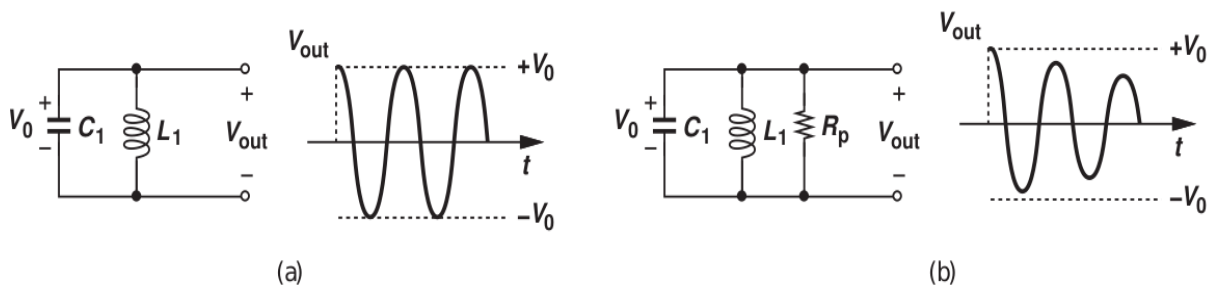


Figure 3. 11: Oscillation à travers le modèle de la résistance négative. (a) idéal. (b) Réel.

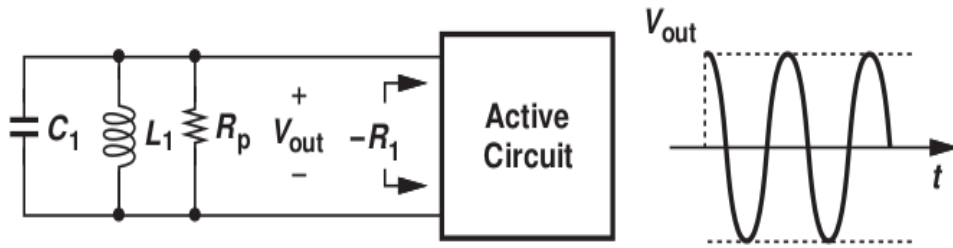


Figure 3. 12: Entretien des oscillateurs à travers un circuit de résistance négative

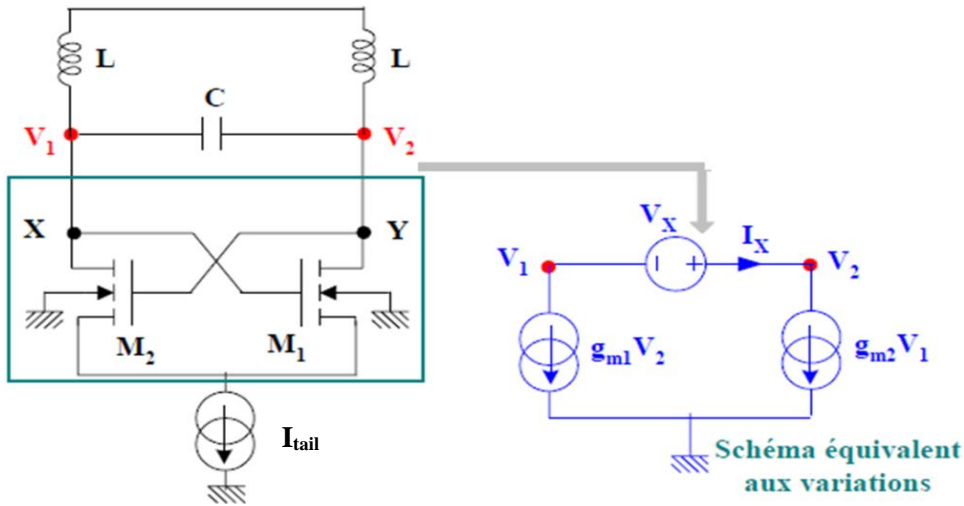


Figure 3. 13 : Modèle petits signaux de la paire différentielle.

### 3.4. Conception d'un VCO LC en technologie CMOS 0.18µm

#### 3.4.1. Conception du résonateur LC

Nous allons commencer notre conception par l'élaboration du résonateur LC. C'est ce dernier qui détermine la fréquence d'oscillation en se basant sur la fréquence de résonance définie dans (3.10).

Donc, il suffit de dimensionner l'inductance de telle sorte à avoir un bon facteur de qualité, ensuite choisir la valeur de la capacité nécessaire. Néanmoins, il faut juste respecter les limites imposées par la technologie CMOS 0.18-µm.

Pour les inductances, elles sont réalisées sur le métal M6 pour minimiser les effets des capacités du couplage au silicium. La largeur et l'espacement entre les rubans ont été imposés par le procédé utilisé, soit  $S = 2\mu\text{m}$ . Lors de la synthèse des inductances, il faut faire un compromis entre sa taille et son facteur de qualité à la fréquence de résonance, l'inductance utilisée dans ce travail présente les paramètres résumés dans le tableau 3.2.

Tableau 3. 2: paramètres de l'inductance du résonateur LC

Métal	W/S	diamètre	N° tours	Q	L
M6	15 $\mu\text{m}$ / 2 $\mu\text{m}$	90 $\mu\text{m}$	3	11.53	3.10 nH

En imposant la valeur de l'inductance, on peut déduire la valeur de la capacité moyennant la formule (3.10). Néanmoins, la valeur de cette capacité n'est pas finale mais elle doit être ajustée une fois le résonateur sera connecté à la paire différentielle. Car cette dernière en effet, tend à décaler la fréquence de résonance sous l'influence des capacités  $C_{gs}$  et  $C_{gd}$  des transistors.

### 3.4.2. Résonateur LC accordé en fréquence

Une fois le résonateur dimensionné, on doit l'octroyer avec la fonction de variation de fréquence. Pour cela nous avons placé en parallèle avec le tank LC une paire de capacités MOS basée sur des transistors NMOS (figure 3.14).

Comme la plage de variation est 140 MHz, on choisit une fréquence minimale de 2.36 GHz et une fréquence maximale de 2.5 GHz

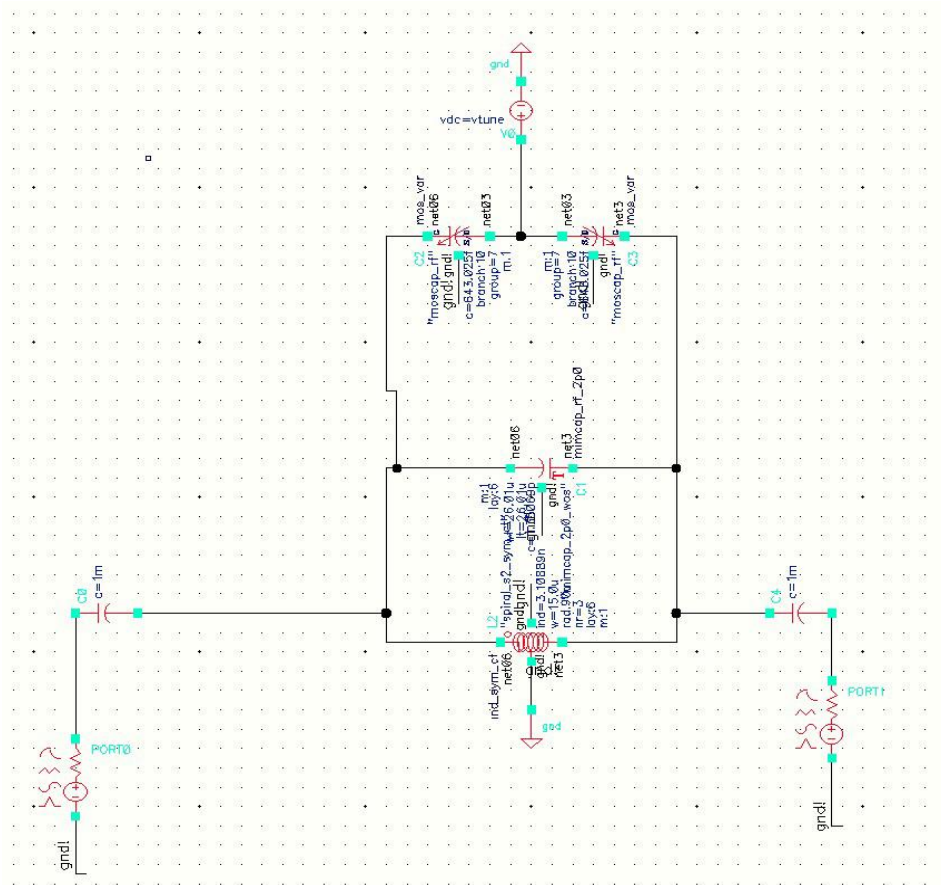
$$\text{Pour } f_{min} = 2.36 \text{ GHz on doit avoir, } C_{max} = \frac{1}{(2\pi f_{min})^2 L} = 2.579 \text{ pF}$$

$$\text{Pour } f_{max} = 2.5 \text{ GHz on doit avoir, } C_{min} = \frac{1}{(2\pi f_{max})^2 L} = 2.125 \text{ pF}$$

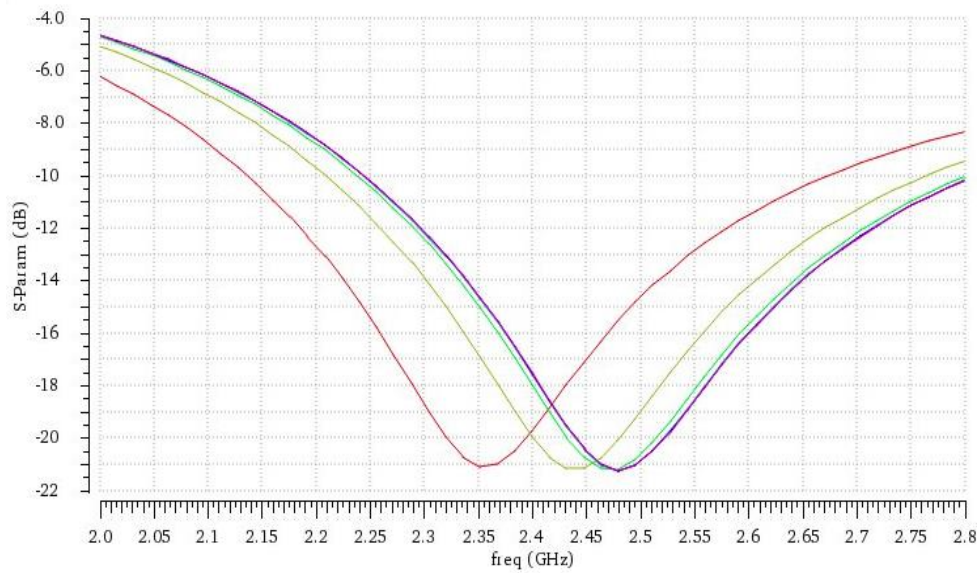
Ainsi on doit dimensionner la paire des capacités MOS pour répondre aux deux valeurs susmentionnées. Le tableau 3.3 présente les valeurs des dimensions de ces dispositifs. Comme on peut le constater la valeur de la capacité varie en fonction de la tension appliquée. La figure 3.14, illustre le circuit final du résonateur accordé en fréquence en technologie CMOS, ainsi que sa réponse variable reflétée par le module du paramètre  $S_{11}$ .

Tableau 3. 3 : Dimension du circuit d'accord de fréquence

$C_{min}$ @-vdd	$C_0$ @ 0	$C_{max}$ @vdd	finger/groupe	groupe
321.416f	643.025f	846.526f	10	7



(a)



(b)

Figure 3. 14 : Résonateur accordé en fréquence. (a) Circuit sous Virtuoso Cadence. (b) Réponse  $|S_{11}|$  pour différentes tension de contrôle



### 3.4.3. Détermination de la résistance équivalente du résonateur

Pour le calcul de la résistance équivalente du tank  $R_p$  nous allons utiliser le montage de figure 3.15, où le port de caractérisation a été placé à l'entrée du circuit du résonateur pour pouvoir déduire l'impédance vue à ce point. Cette configuration permet de tracer le coefficient  $|S_{11}|$  sur l'abaque de Smith et d'en déduire l'impédance d'entrée du circuit.

A partir de la figure 3.16 on peut déduire la valeur de  $R_p$  qui n'est rien d'autre que la partie réelle de l'impédance d'entrée à 2.4 GHz,  $Z = 237.055 - j229.226\Omega$ , ainsi  $R_p = 237.055 \Omega$ .

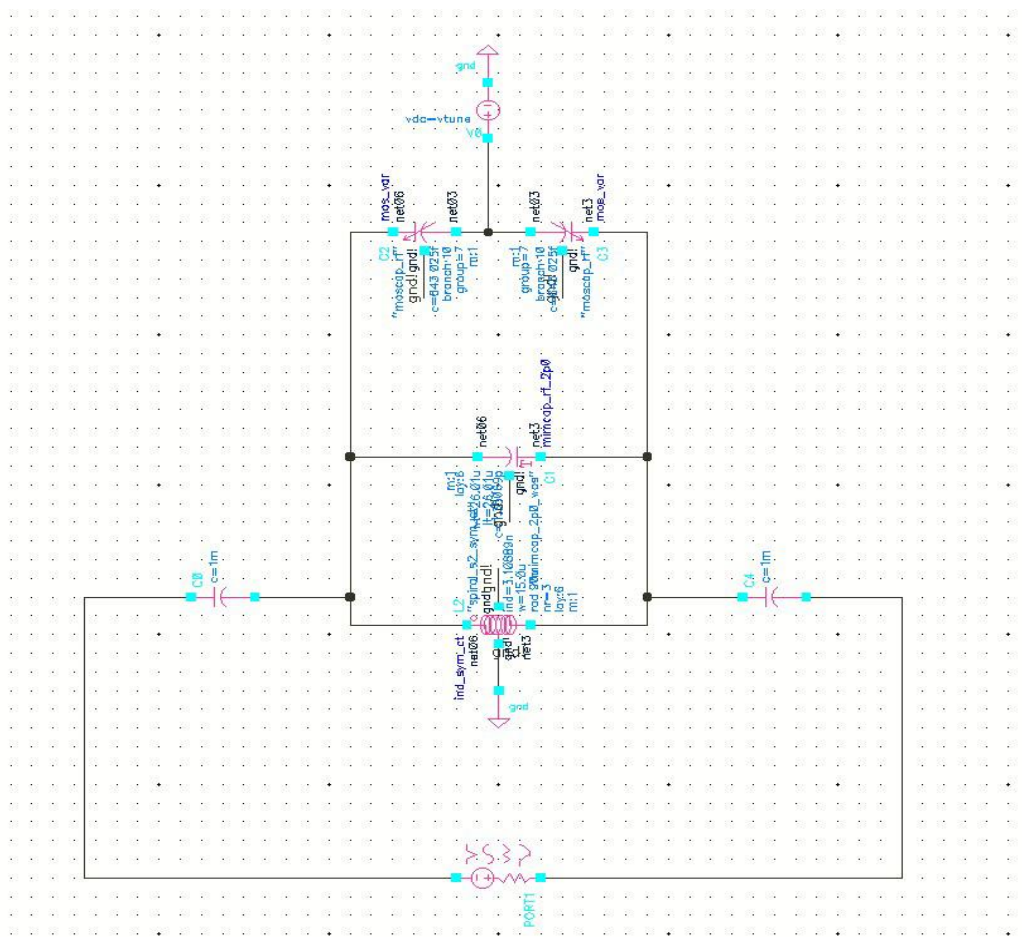


Figure 3. 15: Setup de simulation pour la déduction de la résistance équivalente du résonateur accordé en fréquence

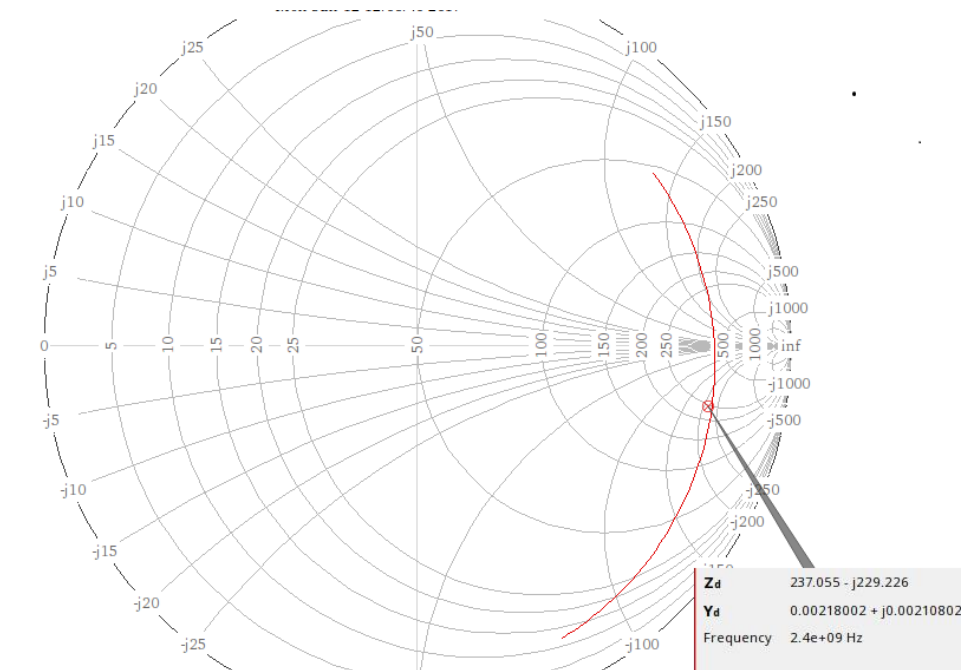


Figure 3. 16 : Dédution de l'impédance d'entrée du résonateur accordé en fréquence

### 3.4.3. Conception du circuit actif

Le circuit actif doit présenter à son entrée une résistance négative ( $-R_p$ ) et doit se mettre en parallèle avec le résonateur pour justement compenser la résistance  $R_p$  de ce dernier. A cet effet, il faut trouver les dimensions nécessaires des transistors pour que la paire différentielle présente une impédance d'entrée égale à  $R_p$  pour satisfaire l'égalité définie dans (3.12). L'impédance d'entrée de la paire différentielle est définie comme suit,

$$Z_{in}(paire\_diff.) = \frac{-2}{g_m} \tag{3.13}$$

Ainsi, on aura  $R_p = \frac{-2}{g_m}$ , en conséquence, la valeur de la transconductance des deux transistors

NMOSFET utilisés pour la paire différentielle est égale à 8.42 mS.

Les transistors M1 et M2 (figure 3.13) sont supposés être en régime de saturation, dans ce cas, pour déduire le rapport W/L des deux transistors, on peut utiliser la formule (3.14) qui exprime la  $g_m$  en fonction des paramètres du transistor en technologie CMOS.

$$g_m = \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_{th}) = \sqrt{2\mu_n C_{ox} \frac{W}{L} I_d} \tag{3.14}$$

Le courant de drain des deux transistors  $I_d$  dans notre cas est égale  $I_{tail} / 2$  (figure 3.13). Le courant  $I_{tail}$  de la source de courant est l'élément qui détermine la consommation du VCO. En supposant une valeur donnée pour  $I_{tail}$  cela revient en réalité à supposer la consommation du VCO. En se basant sur des travaux cités dans la littérature on suppose  $I_{tail}=2.77$  mA, l'objectif est d'avoir une consommation totale de 3mA, en conséquence la différence de 0.23mA sera réservée à la source de courant qui va constituer le tail du VCO (voir plus loin figure 3.18).

Dans ce cas le rapport est  $W/L=712$ . Etant donné que la longueur du canal est  $L=0.18\mu\text{m}$  qui correspond au nœud technologique utilisée, on déduit que  $W=128.16$   $\mu\text{m}$ .

### 3.4.3. Dimensionnement de la source de courant

L'implémentation de la source de courant dans notre conception a été faite par le montage miroir de courant présenté dans la figure 3.17

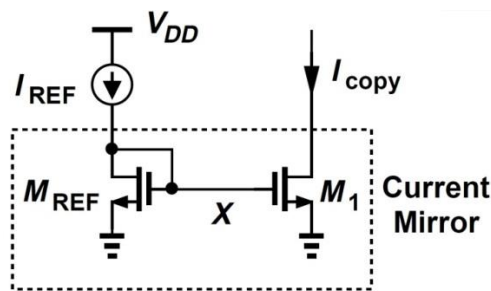


Figure 3. 17: Montage miroir de courant qui va jouer le rôle de la source de courant du VCO

Les courants de référence, et copie  $I_{ref}$ ,  $I_{copy}$  sont donnés par les équations (3.15) et (3.16) sachant que  $I_{copy}$  représente dans notre cas le courant  $I_{tail}$ .

$$I_{REF} = \frac{1}{2} \mu_n C_{ox} \left( \frac{W}{L} \right)_{REF} (V_X - V_{th})^2 \quad (3.15)$$

$$I_{copy} = \frac{1}{2} \mu_n C_{ox} \left( \frac{W}{L} \right)_1 (V_X - V_{th})^2 \quad (3.16)$$

De ces deux équations on peut déduire  $I_{copy}$  comme suit,

$$I_{copy} = \frac{\left( \frac{W}{L} \right)_1}{\left( \frac{W}{L} \right)_{REF}} I_{REF} \quad (3.17)$$

Comme nous l'avons mentionné ci-dessus, le courant consommé par la source de courant (miroir de courant) doit être égale à  $0.23\text{mA}$ , donc  $I_{REF}$  prend cette valeur.

De plus on a choisi  $L_1$  et  $L_{REF}$  de même valeur pour simplifier la conception, et pour diminuer l'effet de modulation de canal, on a choisi des longueurs suffisamment grandes à savoir  $L = L_1 = L_{REF} = 400\text{ nm}$ .

Ainsi en connaissant  $I_{copy}$  ( $I_{tail}=2.77\text{mA}$ ) et  $I_{REF}$  ( $0.23\text{mA}$ ) on peut déduire le rapport des largeurs des deux transistors. Le montage de la paire différentielle avec le miroir de courant, est présenté dans la figure 3.18.

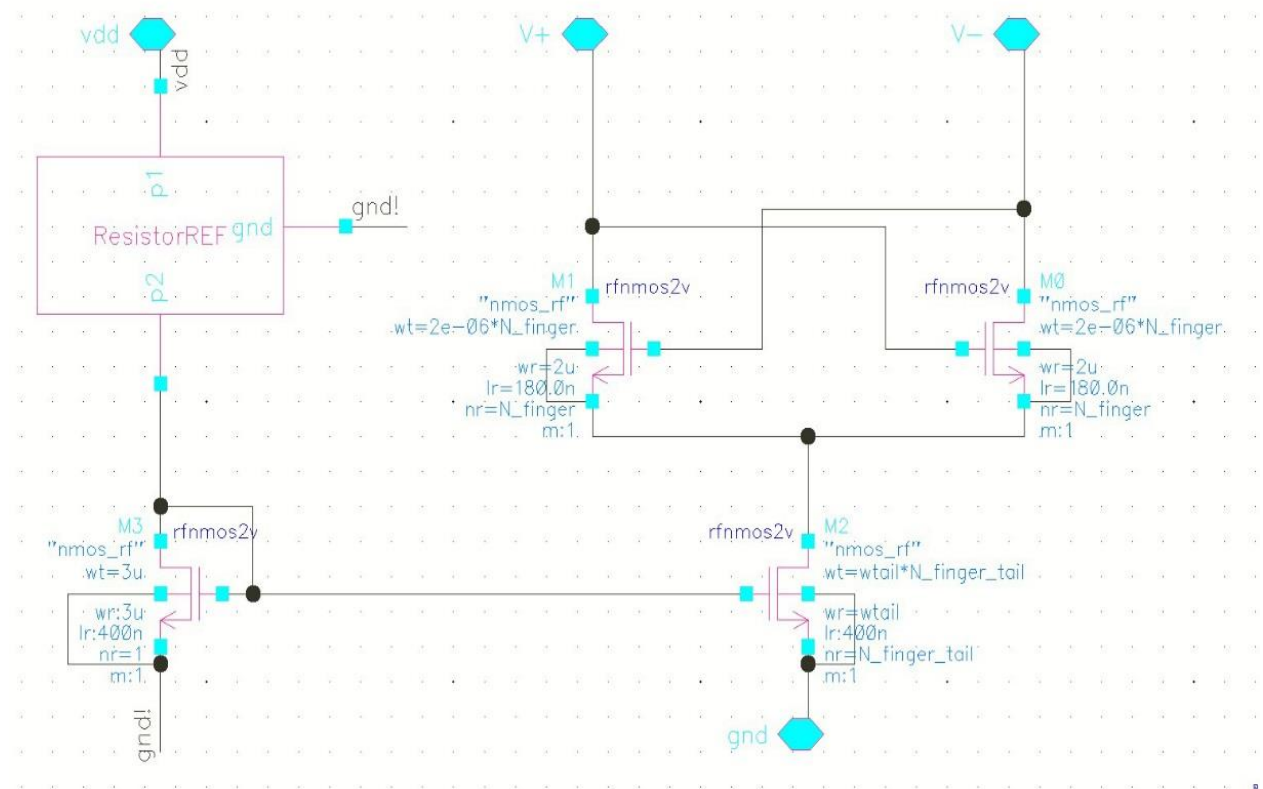


Figure 3. 18: Montage du circuit actif à résistance négative : paire différentielle et miroir de courant

### 3.5. Simulation du VCO LC

Une fois toutes les parties du VCO conçues nous pouvons les interconnecter afin d'aboutir au circuit final du VCO illustré à la figure 3.19. Le circuit possède deux sorties qui sont les sorties différentielles notées  $v_{plus}$  et  $v_{moins}$ . Il est doté aussi de trois pins, à savoir,

vtune pour l'application de la tension de contrôle, vdd pour appliquer la tension de polarisation et gnd pour la masse.

Afin de vérifier le bon fonctionnement du VCO, nous avons utilisé deux simulations, une avec le module '*PSS*' qui permet de voir les oscillations, et l'autre avec '*psnoise*' qui permet d'évaluer le bruit de phase du VCO.

En outre, la simulation *PSS* permet de résoudre le régime établi de circuits complexes avec un temps de calcul réduit [28]. Le nombre d'harmoniques considérées est choisi par l'utilisateur, il doit être assez faible pour minimiser le temps de calcul tout en étant assez important pour obtenir la bonne solution. Avec la simulation *PSS* nous pouvons analyser la réponse temporelle dans les deux régimes transitoire et établi, aussi, nous pouvons évaluer la pureté spectral à travers le tracé de la distribution spectrale de l'une des sorties du VCO, à savoir, vplus ou vminus.

Un symbole a été créé pour le VCO afin de procéder à la simulation, le tesbench utilisé est illustré à la figure 3.20.

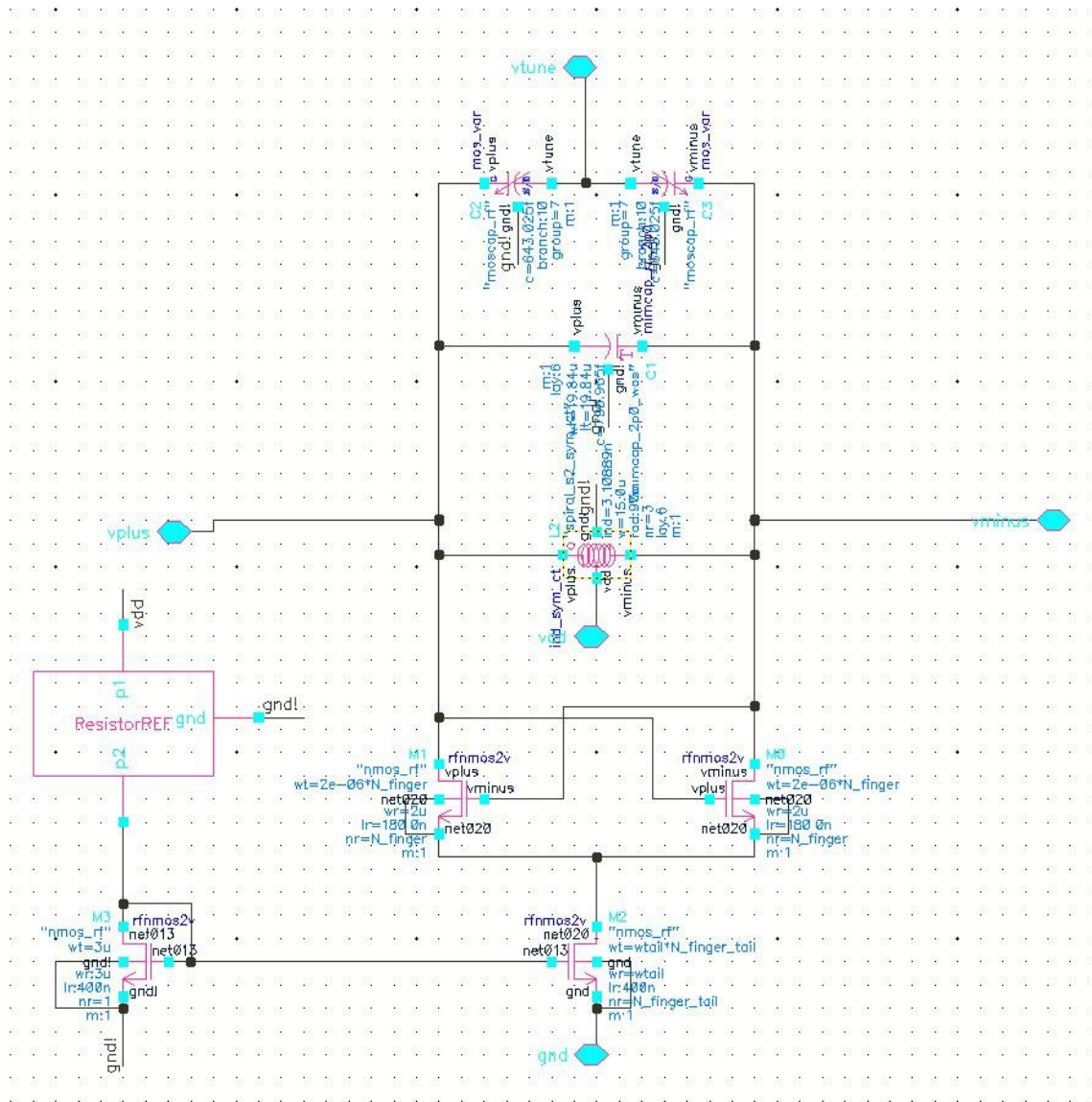


Figure 3. 19: Circuit du VCO LC en technologie CMOS 0.18- $\mu\text{m}$

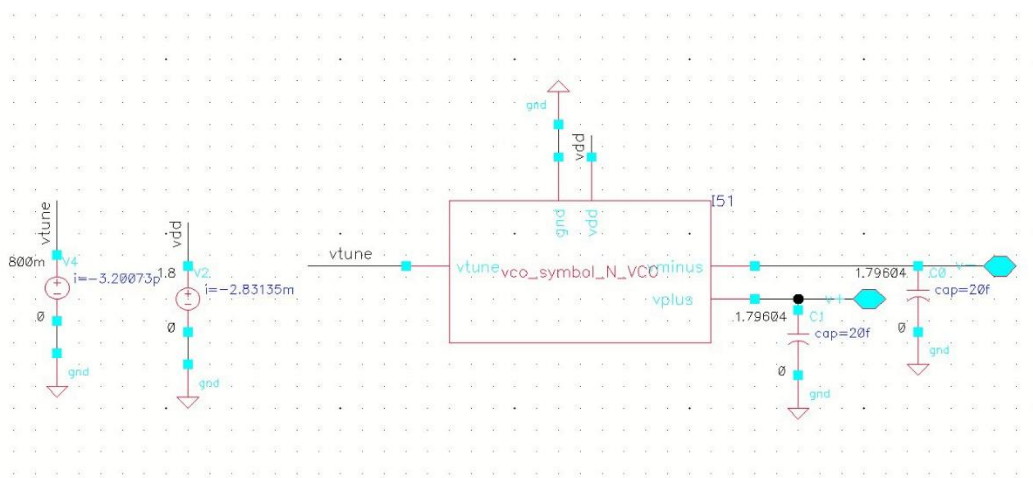


Figure 3. 20 : Setup de simulation du circuit VCO-LC CMOS 0.18- $\mu\text{m}$

### 3.5.1. Résultats de simulation

Les résultats de la simulation sont donnés par les figures 3.21 à 3.26 qui illustrent respectivement, le régime transitoire du VCO, la forme d'onde du signal de sortie, la distribution spectrale du signal de sortie, la variation de fréquence en fonction de la tension de control et le bruit de phase.

La figure 3.21 montre la réponse transitoire du signal de sortie du VCO (vplus ou vminus). Cette réponse nous informe sur le temps d'amorçage du transistor  $T_1$  ainsi que le temps  $T_2$  nécessaire pour atteindre le régime permanent et maintenir les oscillations, il se trouve que ces deux valeurs sont en accord avec ce qui a été reporté dans l'état de l'art [29] [30].

La figure 3.22 démontre bien que la fréquence désirée a été obtenue en présentant une harmonique qui varie en fonction de la tension de contrôle dans l'intervalle 2.36 GHz à 2.6 GHz avec une puissance d'environ 2.9 dBm. La seconde harmonique est suffisamment atténuée par rapport à la fondamentale, avec une valeur d'atténuation d'environ 16.9 dB. De même, la troisième harmonique présente une atténuation de 31.9 dB par rapport à la fondamentale.

En outre, la pureté spectrale du signal de sortie peut être confirmée davantage par la forme d'onde illustrée à la figure 3.23. On peut noter qu'une forme sinusoïdale a été réalisée et ce pour toutes les tensions d'accord  $V_{\text{tune}}$ .

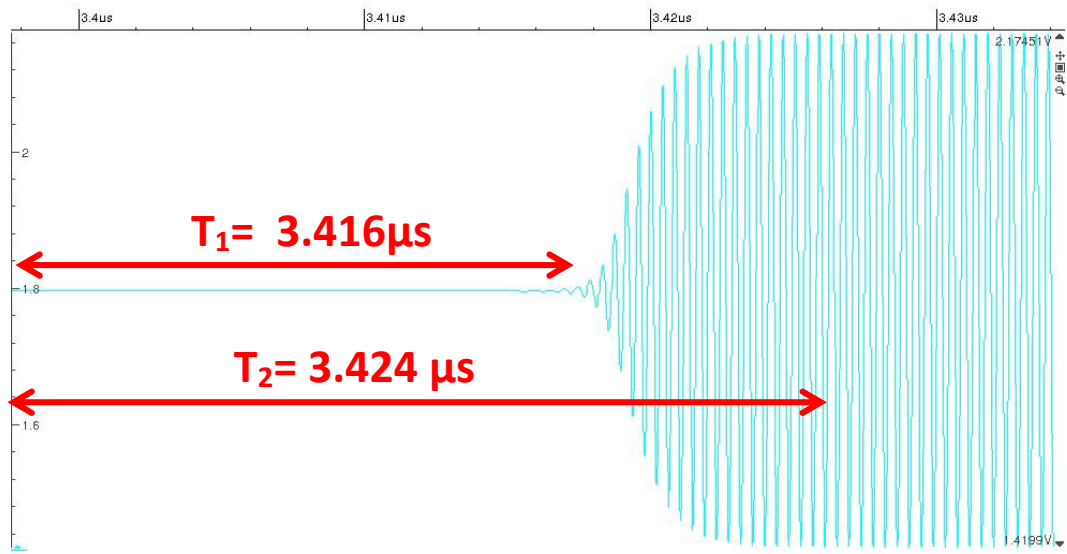


Figure 3. 21 : Régimes transitoire et établi du VCO

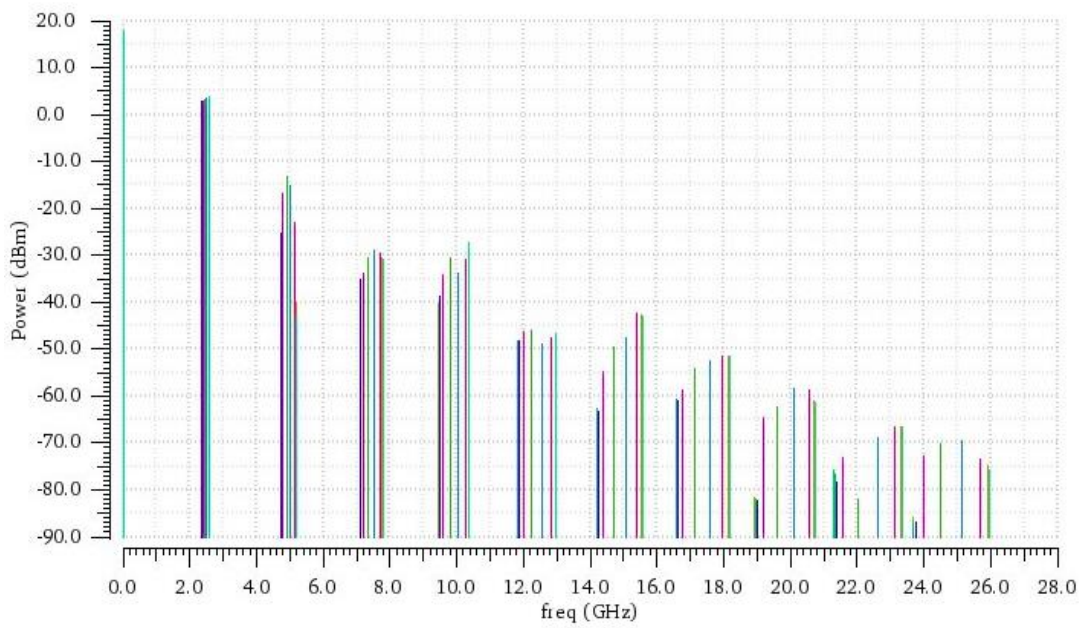


Figure 3. 22 : la distribution spectrale du signal de sortie (Puissance de sortie) pour différentes valeurs de la tension de contrôle



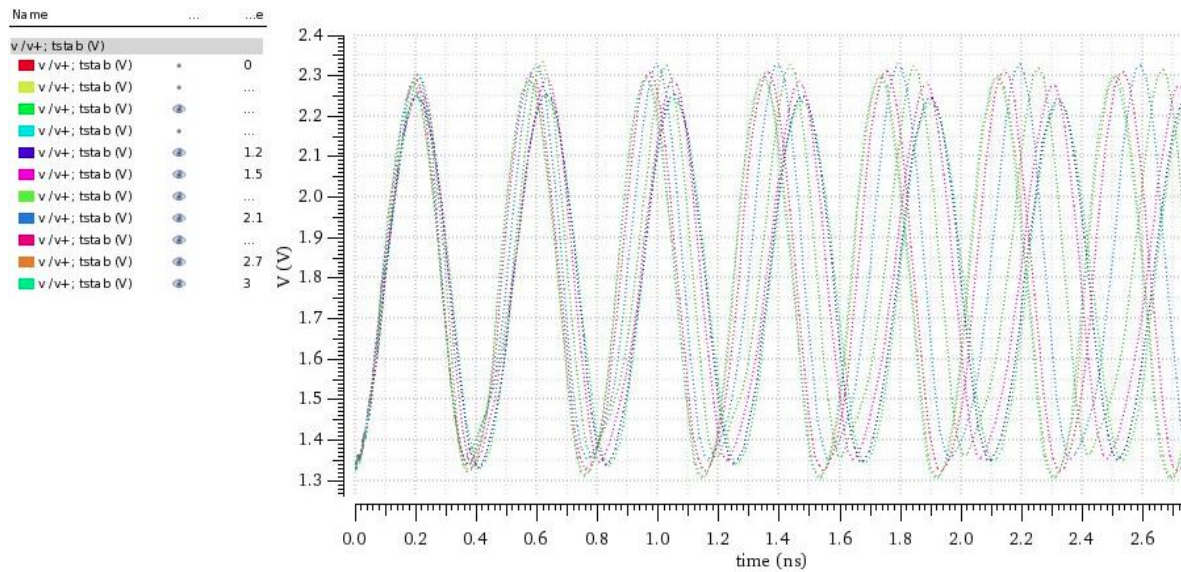


Figure 3. 23 : Forme d'onde du signal de sortie du VCO pour différentes valeur de la tension de contrôle

Une autre caractéristique intéressante est celle qui permet de tracer la variation de la fréquence d'oscillation en fonction de la tension de contrôle  $V_{\text{tune}}$  (figure 3.24). On peut observer que l'allure résultante n'est pas linéaire contrairement à la conception comportementale du VCO présentée au chapitre 2. Cependant, à l'intérieur de l'intervalle de fréquence qui nous intéresse, à savoir, 2.4 GHz à 2.485 GHz (points M3 et M4 de la figure 3.24), on peut assimiler cette variation à une droite. Ainsi, la pente de cette droite représente notre fameux gain du VCO  $K_{\text{vco}}$ . Ce dernier a été tracé en fonction de la tension de contrôle comme illustré à la figure 3.25.

Enfin, la figure 3.26 présente le bruit de phase du circuit proposé, on peut remarquer qu'il a une valeur d'environ -92.91 dBc/Hz, -111.472 dBc/Hz, -118.613 dBc/Hz aux fréquences 1MHz, 5MHz et 10 MHz respectivement. Ces valeurs sont un peu élevées comparativement aux travaux présentés dans la littérature mais qui restent quand même dans les marges tolérées par l'application ciblée, en l'occurrence, le Zigbee.

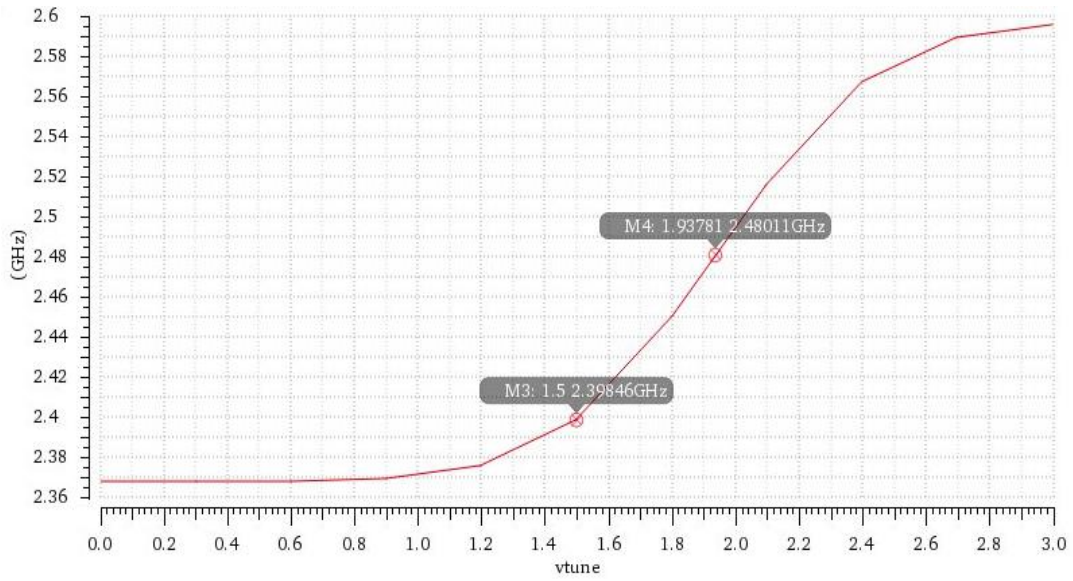


Figure 3. 24 : Variation de la fréquence du signal de sortie en fonction de la tension d'accord ( $V_{tune}$ )

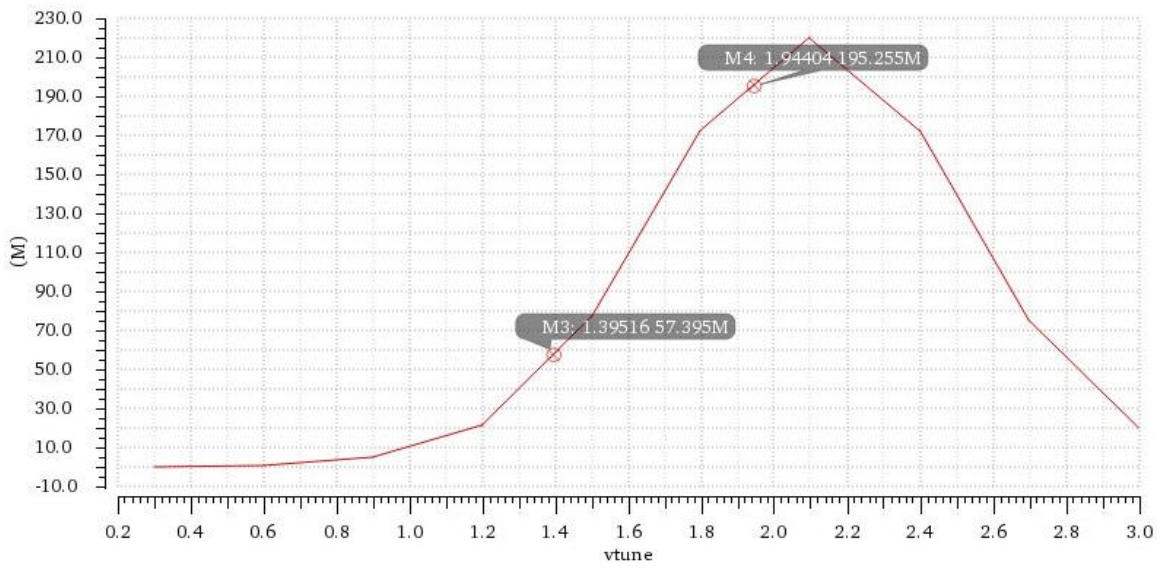


Figure 3. 25 : le gain du VCO  $K_{vco}$  en fonction de la tension d'accord  $V_{tune}$

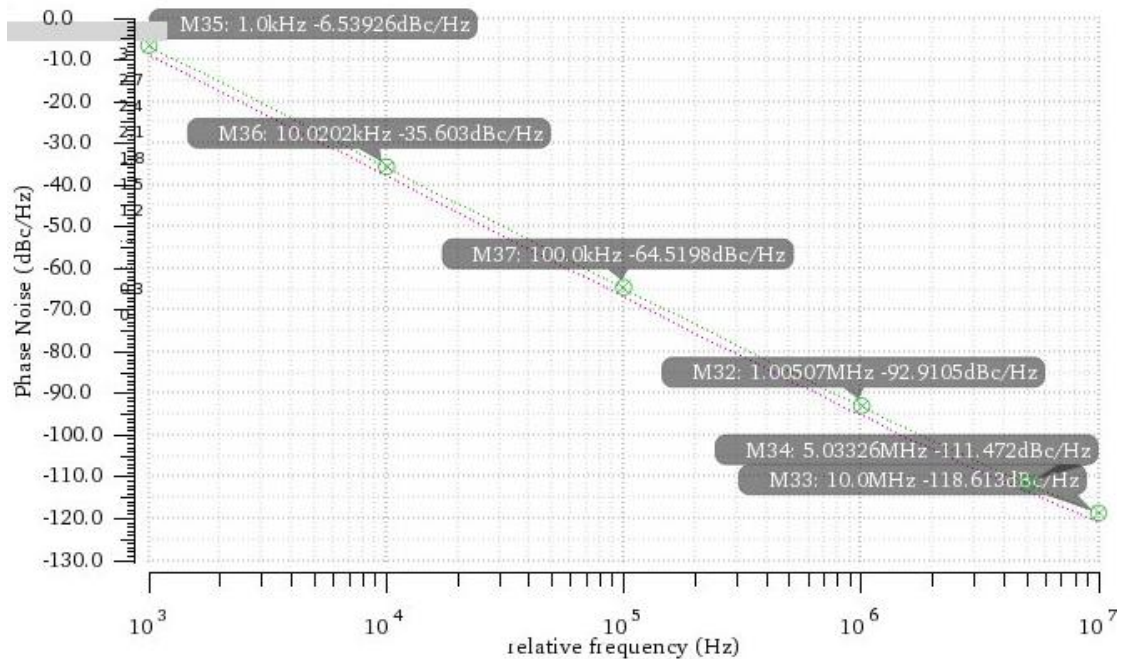


Figure 3. 26 : Bruit de phase du VCO

### 3.6. Insertion du CMOS VCO LC dans le Synthétiseur

Dans cette partie nous allons reprendre le schématique du synthétiseur conçu dans le chapitre 2, mais dans ce cas nous allons remplacer le bloc relatif au VCO à base de Verilog-AMS par le VCO CMOS dont la conception a été exposée ci-dessus. Dans ce cas on se retrouve avec un synthétiseur mixte, c'est-à-dire, composé de circuit analogique qui est le VCO CMOS et de circuits digitaux qui sont le reste des blocs.

La méthode et le setup de simulation que nous allons utiliser sont les mêmes que celles utilisés dans le chapitre 2, à savoir, le simulateur AMS qui prend en charge la simulation des systèmes mixtes.

En outre, nous avons exposé trois phases du fonctionnement du synthétiseur. La première est la phase de démarrage présentée dans la figure 3.27 où on peut noter que l'erreur, qui est la sortie du filtre de boucle, augmente afin de contrôler le VCO CMOS pour atteindre la valeur de la fréquence qui est égale à  $N$  fois la fréquence de référence. On peut remarquer que l'erreur se stabilise après un temps égal à  $4\mu\text{s}$  ce qui signifie que le verrouillage a été atteint, cette valeur est considérée comme très intéressante comparativement à l'état de l'art [31], [32].

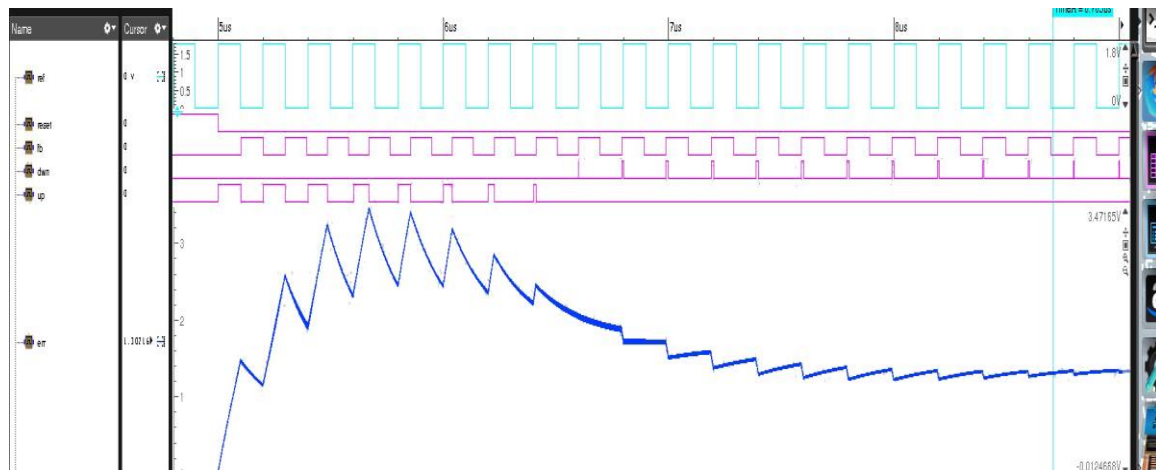


Figure 3. 27: La réponse du synthétiseur dans la phase de démarrage. Les signaux sont de haut en bas : référence, reset, sortie diviseur, down, up, erreur.

La deuxième phase est celle du saut en fréquence de 2.4GHz à 2.475 GHz, elle est représentée dans la figure 3.28. Sachant que ce saut en fréquence a été apporté au système en modifiant la valeur de la division N. On peut noter que le système répond parfaitement à ce changement brutal dans un délai remarquable de 5  $\mu$ s ce qui est largement inférieure à la limite maximale exigée par le standard Zigbee qui est de 192  $\mu$ s [4].

Enfin, la figure 2.29 présente la phase de commande de 15 canaux où la fréquence est incrémentée avec un pas de 5 MHz à chaque fois. On constate de même que le système suit le changement à chaque saut avec un temps de d'acquisition égale à 5 $\mu$ s.

En conséquence le synthétiseur mixte intégrant le VCO CMOS assure parfaitement la fonction de génération de fréquence.

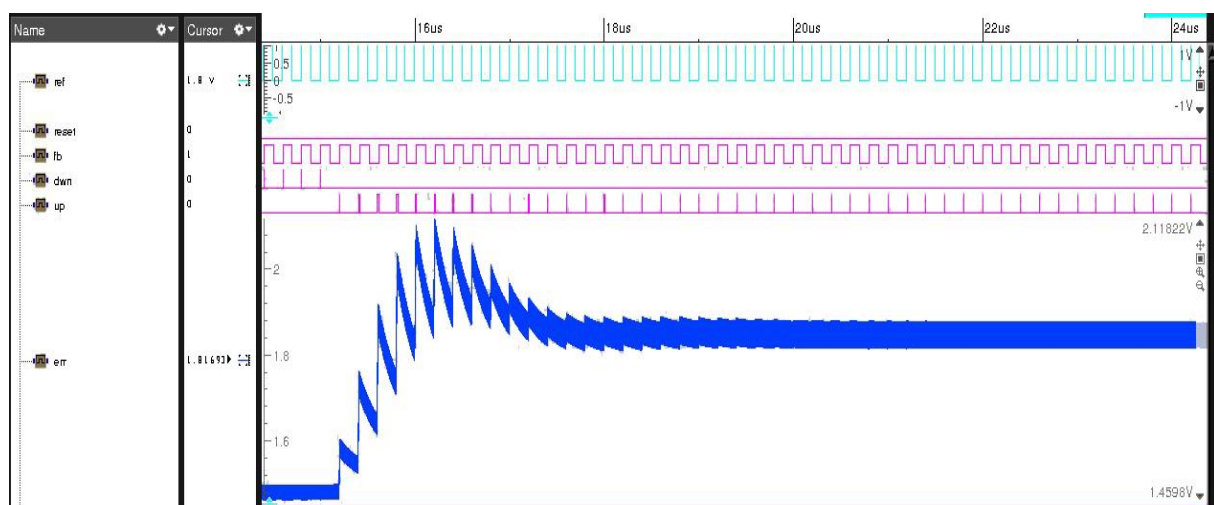
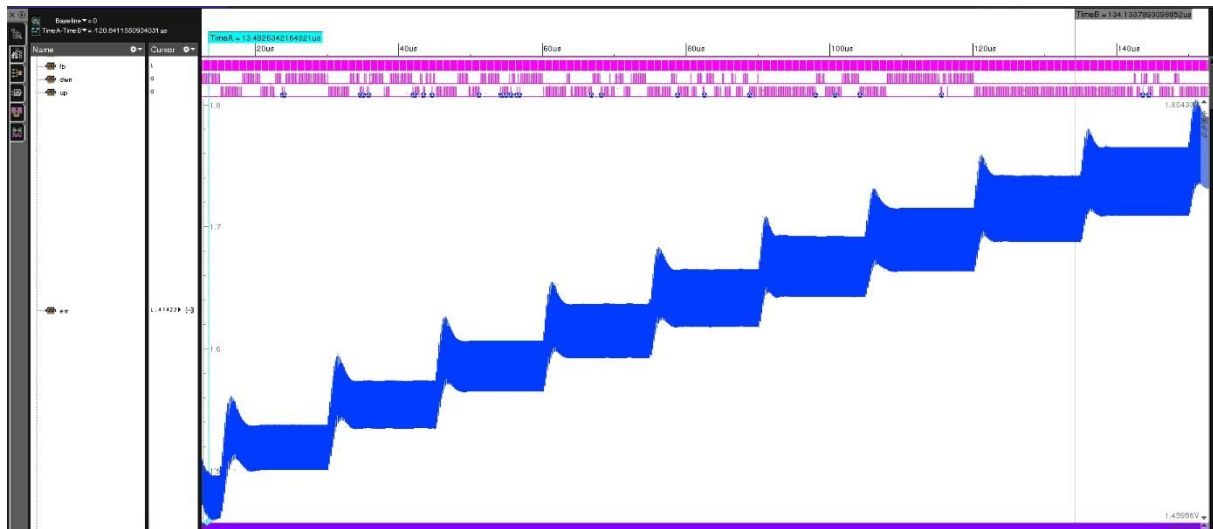
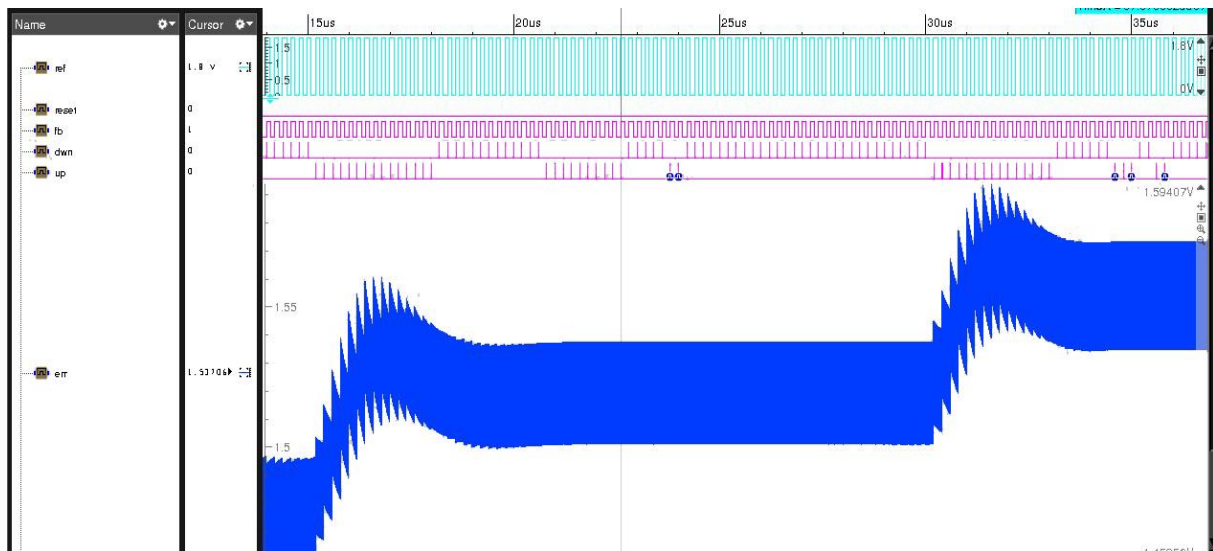


Figure 3. 28 : La réponse du synthétiseur à un saut de fréquence de 2.4 à 2.475 GHz. Les signaux sont de haut en bas : référence, reset, sortie diviseur, down, up, erreur.



(a)



(b)

Figure 3. 29 : Réponse du synthétiseur à la phase de commande des canaux. (a) vue d'ensemble. (b) zoom de la vue

### 3.7. Conclusion

Ce chapitre a traité la conception d'un circuit analogique qui est considéré comme le cœur du synthétiseur, en l'occurrence, l'oscillateur. Ce dernier a été réalisé par un VCO LC implémenté en technologie CMOS 0.18- $\mu\text{m}$ . Le circuit conçu a présenté de bonnes performances en accord avec le standard ciblé, tel que, une pureté spectrale remarquable qui s'est manifestée par une forme d'onde sinusoïdale parfaite, une atténuation de plus de 20 dB des harmoniques par rapport à la fondamentale, une excursion de plus de 1 V du signal de

sortie. De plus toutes ces performances restent valables pour une plage d'accord en fréquence qui s'étale de 2.36 à 2.6 GHz.

Le VCO en question a été par la suite inséré dans le synthétiseur conçu dans le chapitre 2. La simulation du nouveau synthétiseur mixte a donné des performances similaires à celles observées lors de l'étude comportementale avec un asservissement parfait de la fréquence ce qui confirme davantage la validité du VCO conçu ainsi que la réussite de son insertion.

# **Chapitre 4**

## **Conception des Circuits Digitaux du Synthétiseur**

## 4.1. Introduction

Comme nous l'avons introduit au début de ce manuscrit, les synthétiseurs sont des systèmes mixtes composés de circuits analogiques et digitaux. Dans le chapitre précédent nous avons traité la conception du circuit analogique le plus important, à savoir, un VCO LC implémenté en technologie CMOS 0.18- $\mu\text{m}$ . Dans ce chapitre nous allons aborder la conception d'un autre circuit numérique important, il s'agit du diviseur de fréquence. Nous avons retenu une architecture intelligente qui permet de changer le rapport de division  $N$  par simple commande numérique. Cette topologie est constituée de sous-circuits, nous allons présenter la conception de ces derniers et leurs simulations ainsi que le diviseur complet. Il faut noter que l'implémentation de ce diviseur a été effectuée sur la technologie CMOS 0.18- $\mu\text{m}$ .

A la fin du chapitre nous allons aussi exposer la conception du PFD à travers la synthèse de son code Verilog à travers l'outil Encounter où le circuit niveau porte logique sera présenté.

## 4.2. Conception du diviseur

La plupart des synthétiseurs à  $N$  entier sont implémentés avec le diviseur dont la topologie est présentée dans la Figure 4.1 [7]. Il se compose d'un prédiviseur (prescaler) à double module  $N/N+1$  (Dual Modulus Prescaler), d'un compteur programmable  $P$  et d'un compteur programmable  $S$  (Swallow counter). Le rapport de division de ce diviseur est donné par [7]

$$f_{out} = ((N+1)S + N(P-S))f_{in} = (NP + S)f_{in} = N' f_{in} \quad (4.1)$$

Le prédiviseur commence à diviser par le rapport  $N+1$ , jusqu'à ce que le compteur  $S$  arrive à la fin de compte, il change la valeur du bit de contrôle forçant ainsi le prédiviseur à changer le module de division vers  $N$ . A ce point le diviseur a compté  $(N+1).S$ , le compteur  $P$  quant à lui continue à compter les  $(P-S)$  fronts montant restants du prédiviseur et émet un signal reset au compteur  $S$ , et le système recommence à nouveau. Pour que cette architecture fonctionne, il faut que  $P > S$ .



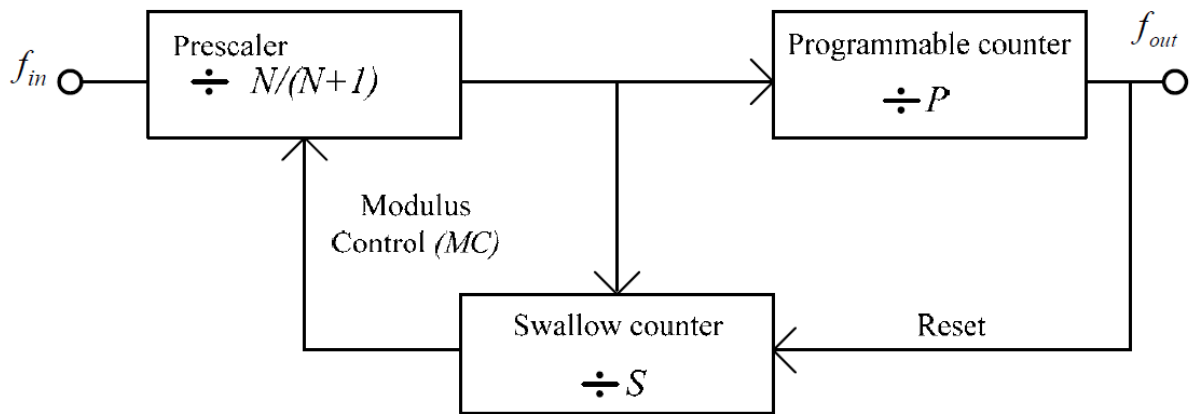


Figure 4. 1 : Architecture du diviseur de fréquence [31]

Pour notre cas, nous avons besoin d'un rapport de divisions entre les deux valeurs présentées dans (4.2) et qui correspondent aux deux fréquences de coupure de la plage d'accord,

$$\begin{aligned}
 N'_{\max} &= \frac{f_{\max}}{f_{REF}} = \frac{2475}{5} = 495 \\
 N'_{\min} &= \frac{f_{\min}}{f_{REF}} = \frac{2400}{5} = 480
 \end{aligned}
 \tag{4.2}$$

En choisissant un prédiviseur 32/33, la valeur nécessaire pour P est 15, et S varie donc entre 0 à 14. Dans le mode S=14, le rapport de division sera  $N' = (32 \times 15 + 14) \times 5 = 494$ . Pour avoir le dernier rapport de division, nous ajoutons un mode où le compteur S sera désactivé, dans ce cas le rapport de division sera  $(N + 1)P = 33 * 15 = 495$

#### 4.2.1 Conception du prédiviseur

Une implémentation du prédiviseur peut être réalisée à l'aide de l'architecture de la figure 4.2 fournie dans [31]. Une autre forme simplifiée d'un diviseur par deux est implémentée à l'aide d'une bascule D comme il est montré dans la figure 4.3 [34].

On peut citer une autre architecture, celle d'un diviseur par trois qui peut être implémentée à l'aide du compteur de Johnson comme il est montré dans la figure 4.4 [34].

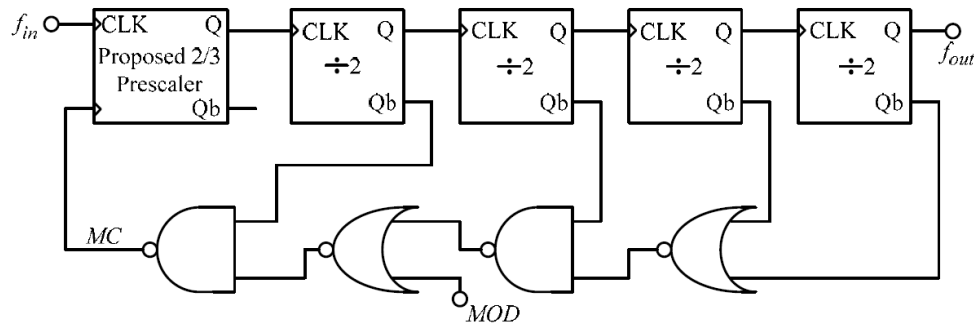


Figure 4. 2 : Architecture du prédiviseur

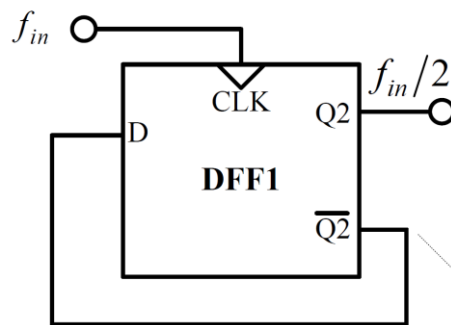


Figure 4. 3 : Architecture du diviseur par deux

Finalement le prédiviseur 2/3 est la combinaison des deux diviseurs précédents, son architecture est proposée dans la figure 4.5.

Le diviseur fonctionne à des fréquences d'environ 2.4 GHz, ce qui ne peut être atteint par des cellules standard d'une bibliothèque fournie par une technologie CMOS donnée [33]. Il faut donc personnaliser ces composants, ceci est fait en optimisant la bascule D utilisée comme bloc unitaire. Il existe plusieurs solutions pour optimiser ce bloc, parmi ces méthodes, il y a la logique CML (Current Mode Logique) et la logique dynamique. Les diviseurs à base de CML ont une plus grande fréquence de travail mais une plus grande consommation de puissance que les diviseurs à logique dynamique. La logique dynamique optimise les bascules D par l'utilisation des capacités parasites comme mémoire de la bascule, la figure 4.6 en démontre un exemple.

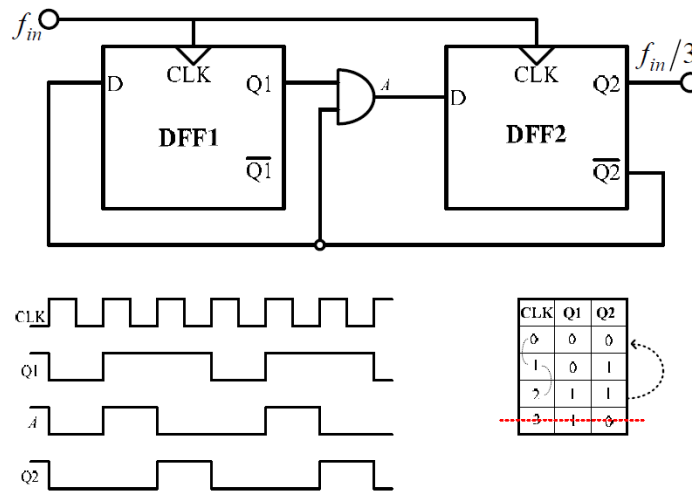


Figure 4. 4 : Architecture du diviseur par 3

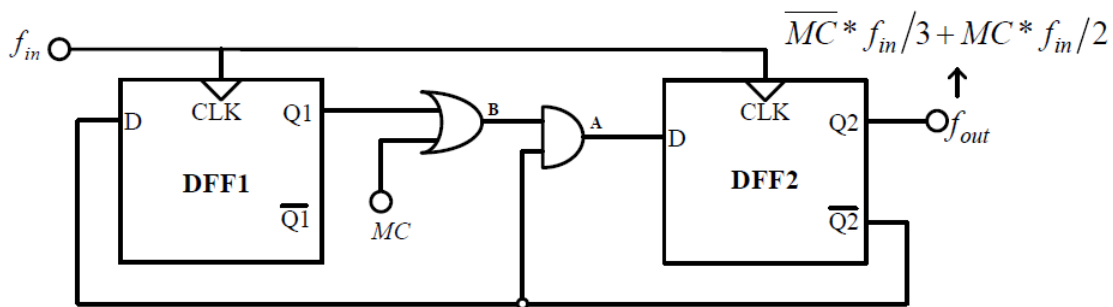


Figure 4. 5 : Architecture du prédiviseur 2/3

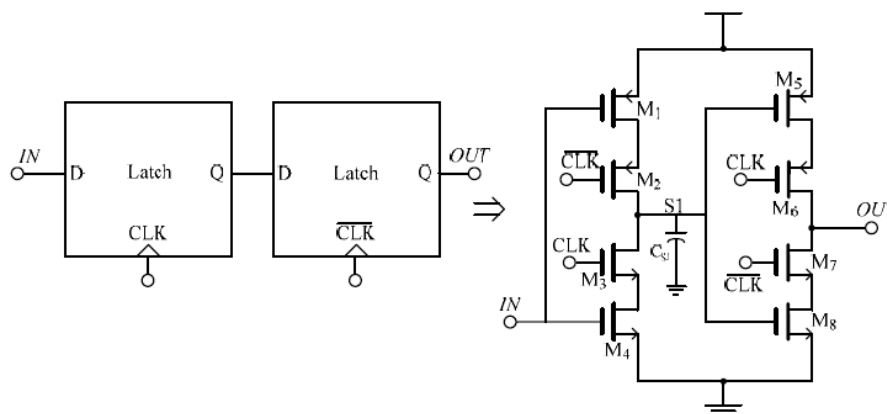


Figure 4. 6 : Bascule logique Clocked-CMOS

Le problème de cette solution est la nécessité d'utiliser une horloge complémentaire, en effet l'utilisation de cette dernière engendre des déphasages avec l'horloge primaire, ceci va constituer une source d'erreurs importante. A cet effet, une autre solution est envisageable, il

s'agit d'utiliser une seule phase du signal d'horloge, la concrétisation de cette alternative est présentée dans la figure 4.7 [33]. Nous avons utilisé cette configuration dans notre cas, nous l'avons reproduit sur Cadence [35] (figure 4.8) et la simulation de cette bascule est présentée en annexe.

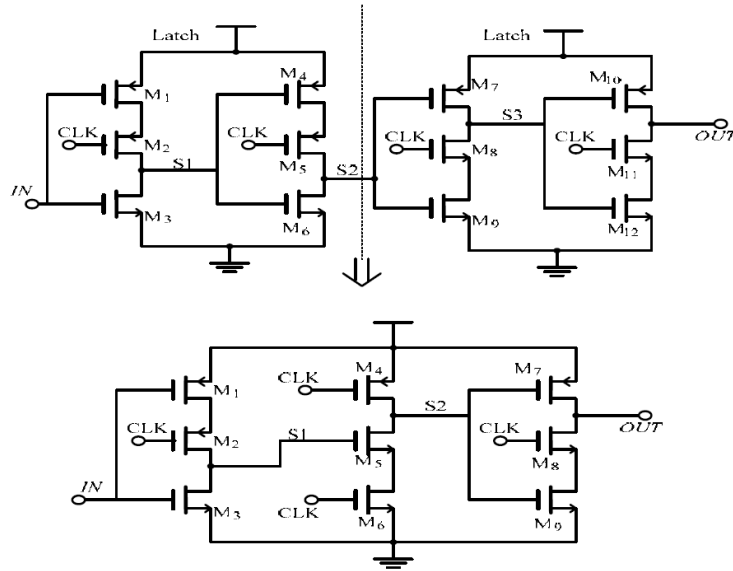


Figure 4. 7 : Obtention d'une bascule TSPC à partir d'une bascule Clocked-CMOS

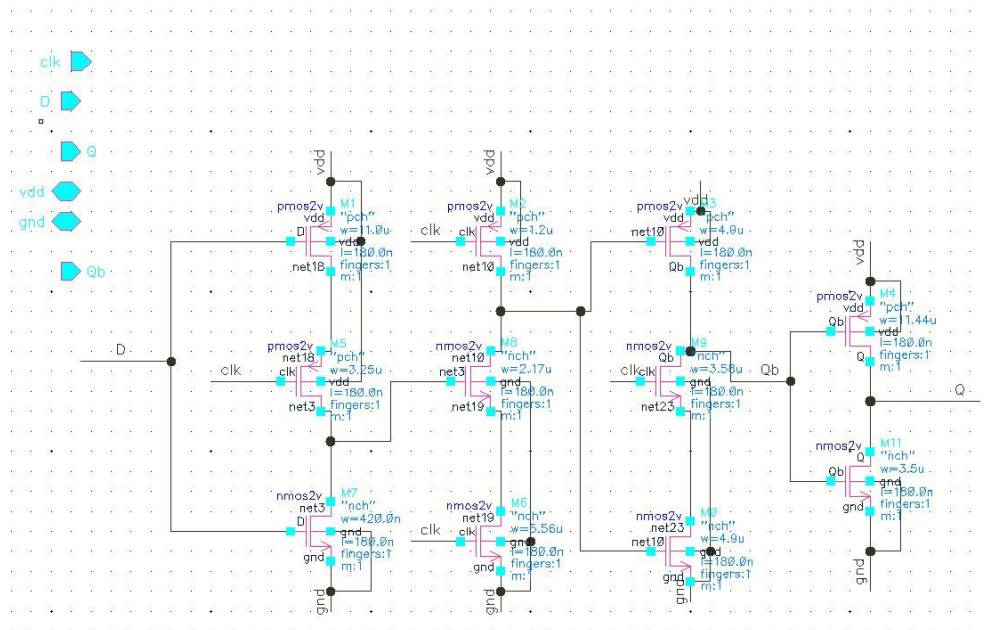


Figure 4. 8 : Schématique de la bascule TSPC utilisé dans ce travail

### 4.2.2. Simulation du diviseur par 2

Nous avons utilisé la bascule de la figure 4.8 dans le diviseur par 2 exposée dans la figure 4.3, l'ensemble a été reproduit sur Cadence. Pour vérifier le fonctionnement de ce montage, nous avons injecté à son entrée un signal de période 410ps, le résultat de la simulation est présenté dans la figure 4.9. On peut vérifier facilement que le signal à la sortie (rouge) du diviseur a la moitié de la fréquence du signal d'entrée (bleu).

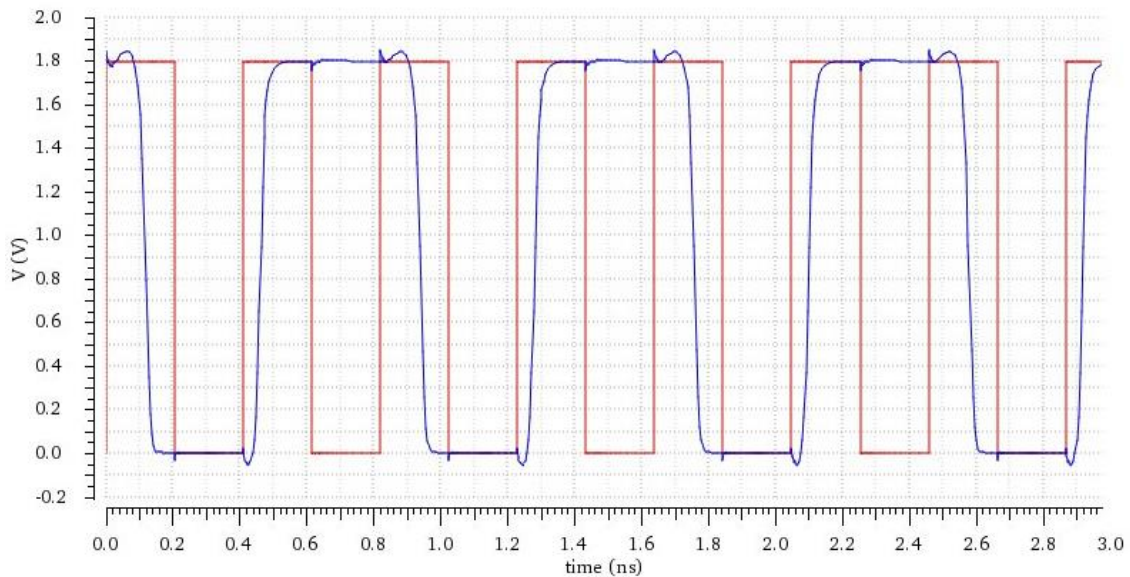


Figure 4. 9 : Réponse du diviseur par 2 moyennant une simulation sur Cadence, signal d'entrée bleu.

Signal de sortie rouge

### 4.2.3. Simulation du prédiviseur 2/3

De même nous avons reproduit l'architecture illustrée dans la figure 4.5. Pour simuler ce prédiviseur nous avons injecté un signal de période  $t = 385$  ps à son entrée clk, et nous l'avons simulé pour les deux valeurs du bit de control MC, le résultat de la simulation est présenté dans la figure 4.10.

Lorsque  $MC = 1$ , le prédiviseur fonctionne en mode de division par 2, et lorsque  $MC=0$ , le prédiviseur fonctionne en mode de division par 3, on peut très bien noter ce changement au moment où le signal en vert change d'état de 1 à 0.

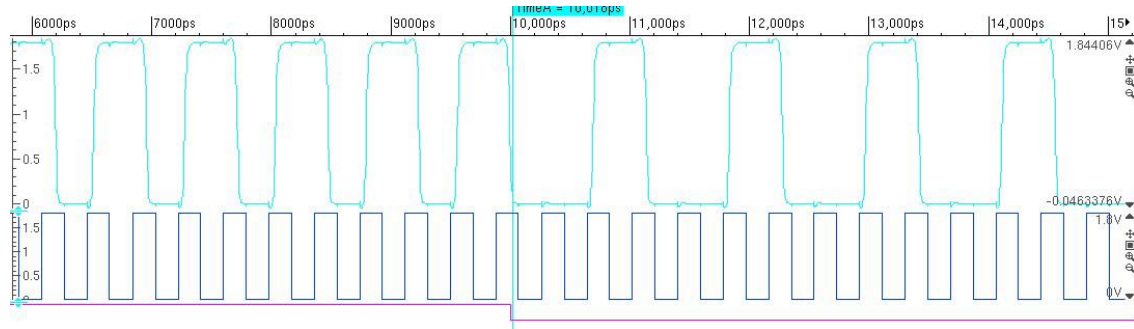


Figure 4. 10 : Réponse du prédiviseur 2/3 pour MC=0 et MC = 1. Signal injecté en rouge. Signal de sortie en jaune. MC en vert.

#### 4.2.4. Simulation du prédiviseur 32/33 ( $t=410\text{ps}$ )

Finalement nous avons construit à l'aide du diviseur par deux, et du prédiviseur 2/3, un prédiviseur 32/33 en suivant l'architecture de la figure 4.2. Le setup de la simulation est présenté dans la figure 4.11 où on injecte aux entrées du prédiviseur les signaux suivants : N\_value qui est la commande MC, un signal sinusoïdal pour simuler le VCO, et le signal reset. Le résultat de la simulation est présenté dans la figure 4.12 où la sortie du prédiviseur est exposée.

La période dans le premier mode est donnée par la distance entre le marqueur A et B (figure 4.13). La période dans le deuxième mode est donnée par la distance entre le marqueur C et D (figure 4.13). Afin de vérifier les rapports de division du prédiviseur dans les deux modes, nous pouvons effectuer le calcul suivant :

$$\begin{aligned} T_{out}^1 &= 32T_{in} = 0.410 * 32 = 13.12ns \\ T_{out}^0 &= 33T_{in} = 0.410 * 33 = 13.53ns \end{aligned} \quad (4.3)$$

Ces deux valeurs sont proches des valeurs de la simulation présentée dans la figure 4.13, cependant on peut expliquer cette légère différence par le fait que les diviseurs par 2 et 2/3 ne fournissent pas une division idéale à cause des retards de montée dans les signaux digitaux. La figure 4.14 fournit un zoom du signal de sortie pour le mode MC=0 nous pouvons donc calculer le nombre d'impulsion que le circuit a compté, on vérifie bien qu'il a divisé par 33.

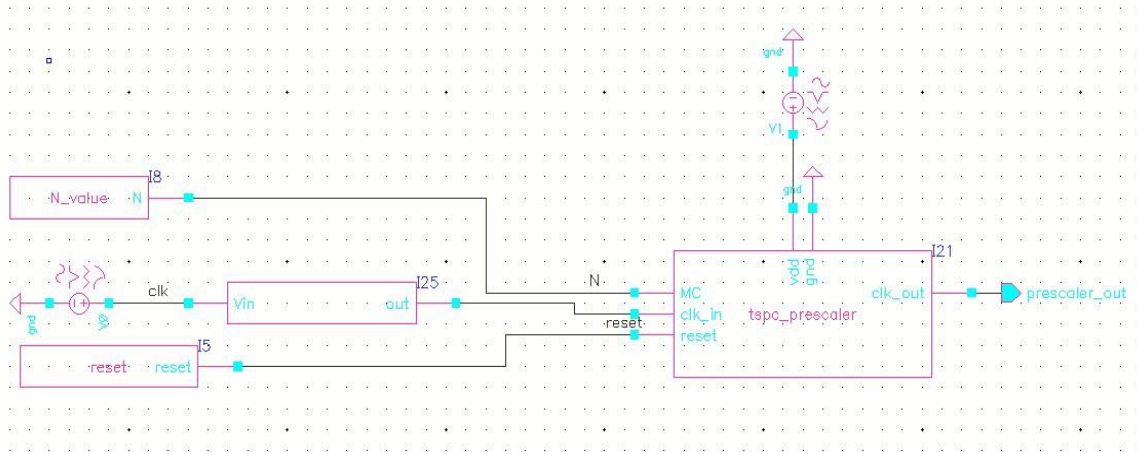


Figure 4. 11 Setup de la simulation du prédiviseur 32/33

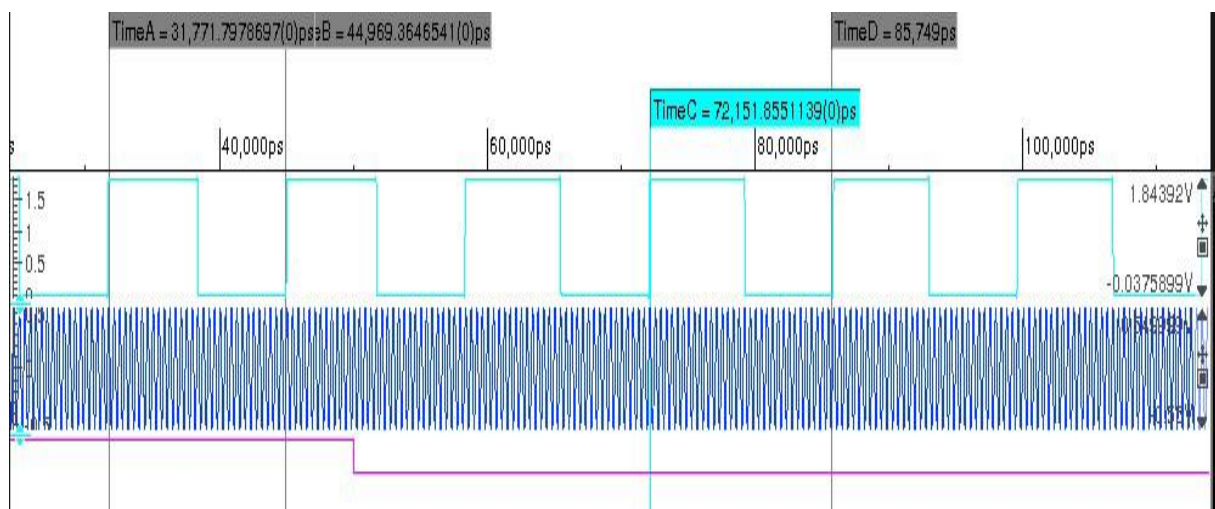


Figure 4. 12 : Réponse du prédiviseur 32/33 pour MC = 0 et MC=1. Signal injecté en rouge. Signal de sortie en jaune. MC en vert.

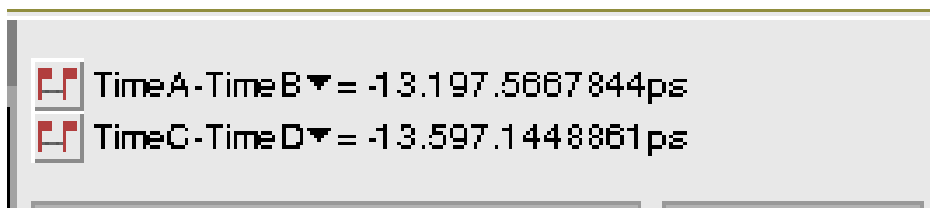


Figure 4. 13 : Périodes de la sortie du mode MC=0 et MC=1

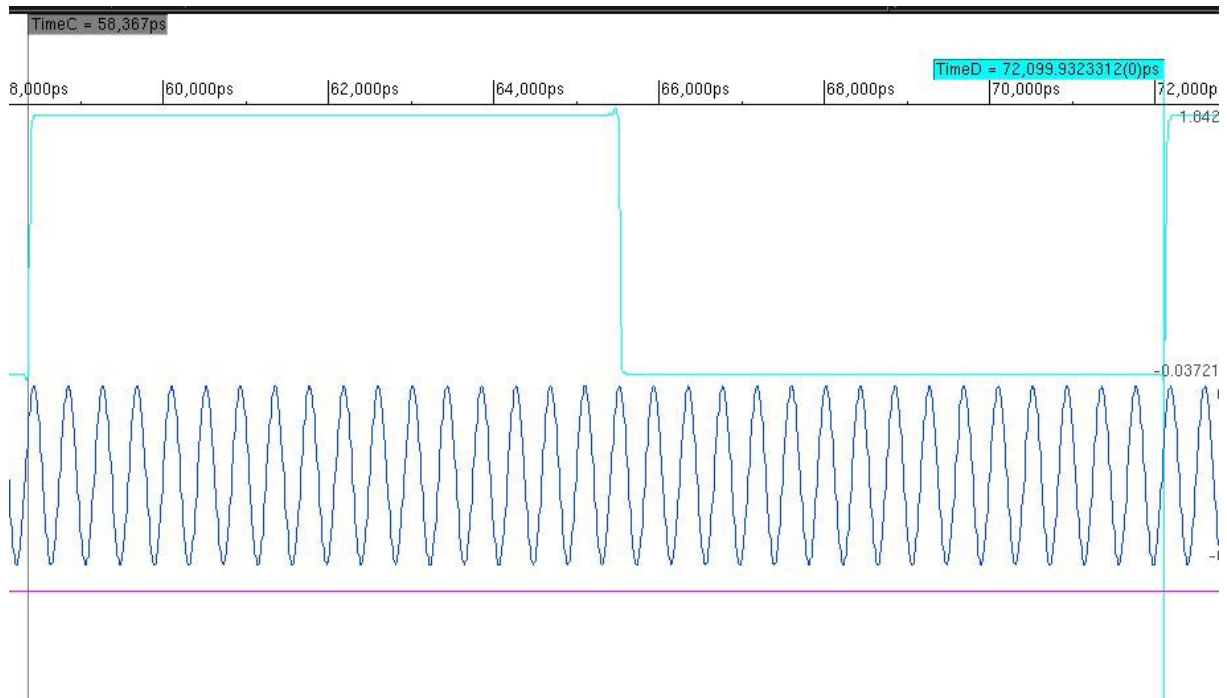


Figure 4. 14 : zoom sur la réponse en mode MC=0

#### 4.2.4 Conception des compteurs P et S

Pour les compteurs P et S, leur fréquence de fonctionnement est de 75 MHz. Cette dernière peut être supportée par les cellules standard de la technologie CMOS 0.18- $\mu\text{m}$  que nous allons utiliser. Nous avons donc opté pour une synthèse RTL à l'aide du synthétiseur RTL de l'outil Encounter de Cadence.

Le compteur nécessaire que l'on a besoin doit avoir 4 digits, et ce, en se basant sur le calcul suivant,

$$n = \frac{\ln 15}{\ln 2} = 3.9 \approx 4 \quad (4.4)$$

La description comportementale du compteur est donnée en annexe ainsi que la description niveau porte logique synthétisée. La description niveau porte logique est obtenue après l'opération de la synthèse effectuée avec Encounter.

##### 4.2.4.1. Résultat de simulation du compteur (P/S)

Les résultats de la simulation fonctionnelle du diviseur sont présentés dans les figures 4.15 à 4.17. Les signaux de haut en bas sont : le signal est le signal de référence, le signal reset, la sortie du compteur, et les quatre derniers sont le vecteur de commande N qui



représente la valeur de la division. Dans chaque cas de ces combinaisons, on voit bien que le compteur compte la bonne valeur. Les combinaisons sont les suivantes

- Premier cas :  $N = (0010)_2 = (2)_{10}$
- Deuxième cas :  $N = (0011)_2 = (3)_{10}$
- Troisième cas :  $N = (0101)_2 = (5)_{10}$

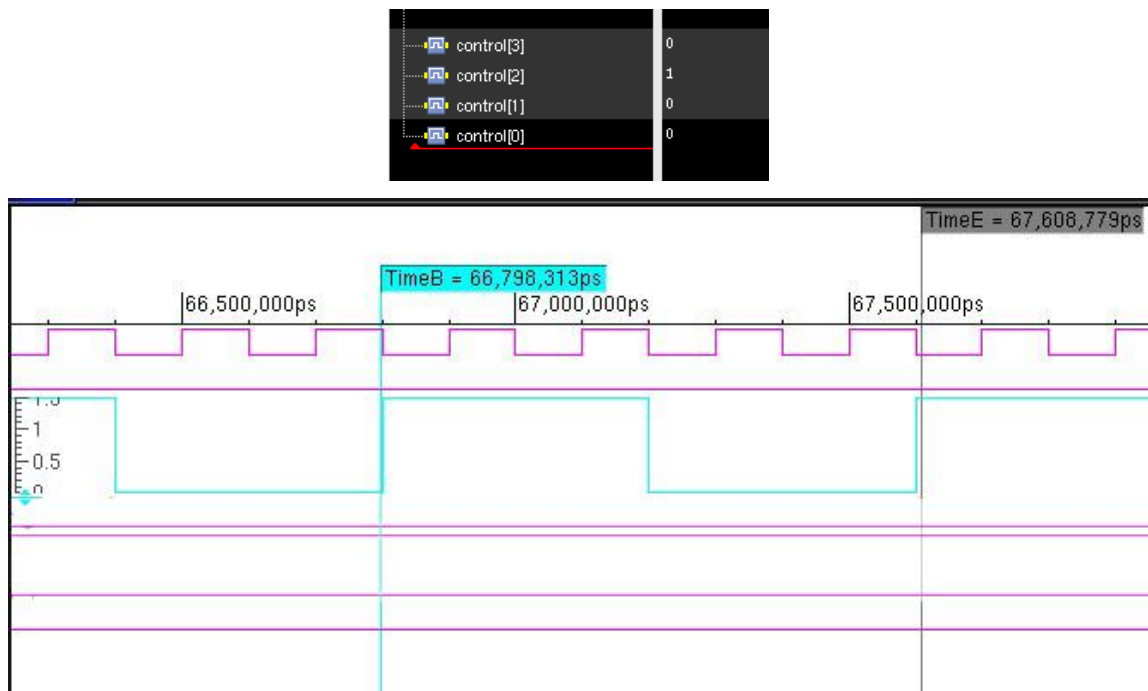


Figure 4. 15 : Simulation du compteur avec  $N = 2$

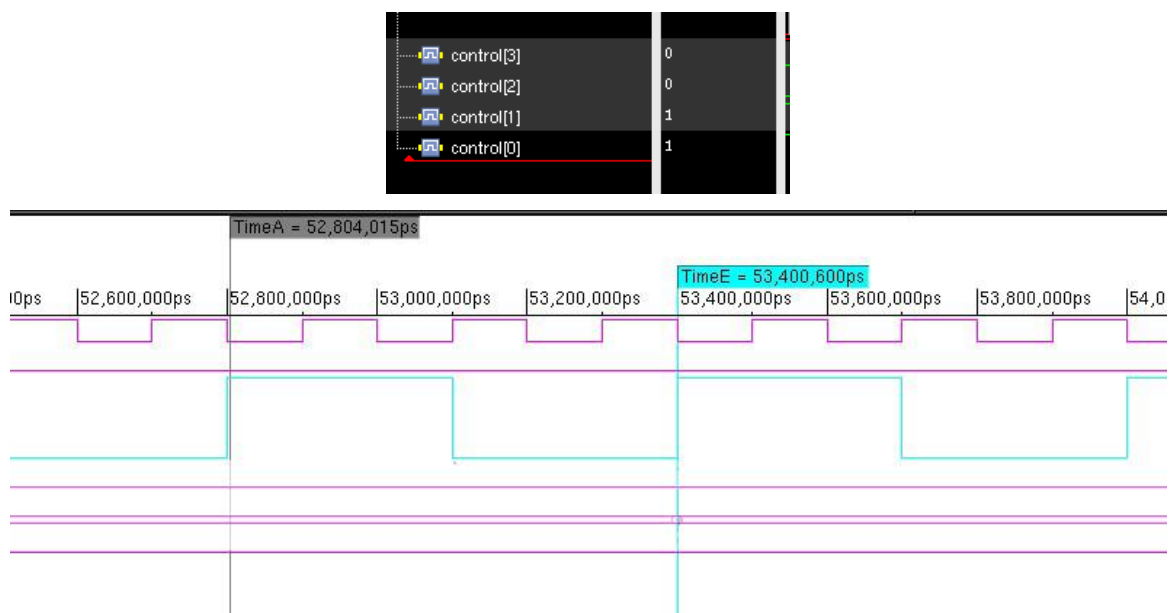


Figure 4. 16 : Simulation du compteur avec  $N = 3$

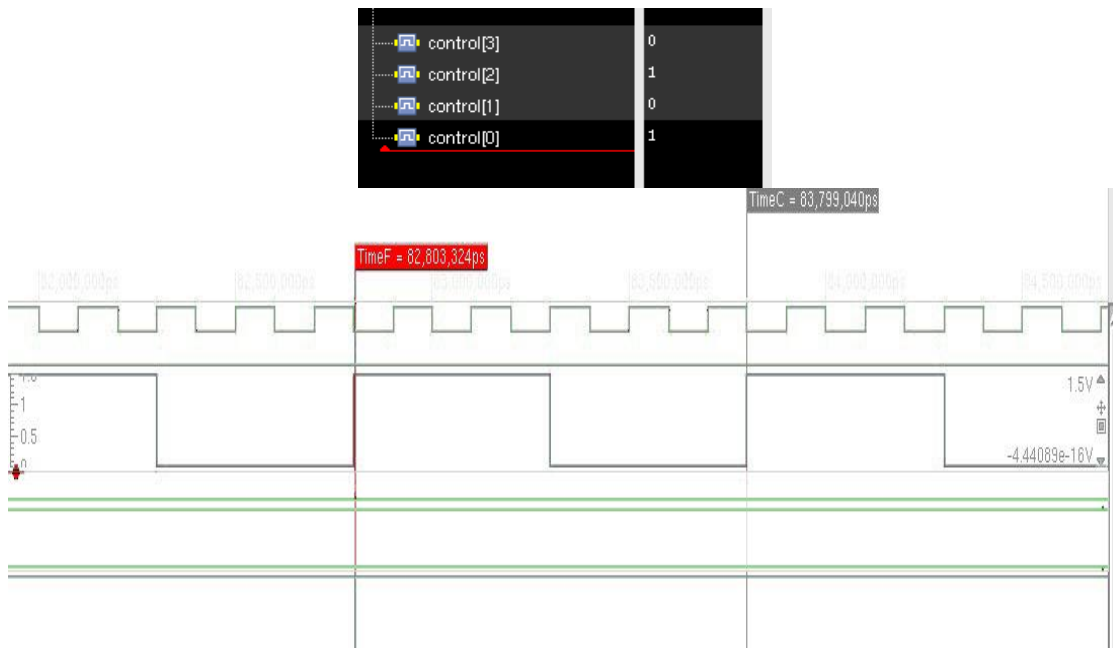


Figure 4. 17 : Simulation du compteur avec N = 5

#### 4.2.5. Simulation du diviseur complet

Une fois que les compteurs et le prédiviseur sont vérifiés, nous les rassemblons selon l'architecture présentée dans la figure 4.1 pour concevoir le diviseur complet.

Afin de vérifier son fonctionnement, nous avons injecté à son entrée ' clk\_in' un signal sinusoïdal de période (t = 410ps), la valeur du compteur P est fixée à 15 ('1111'), et la valeur du compteur S est varié de 0 à 14, la figure 4.18 montre cette configuration.

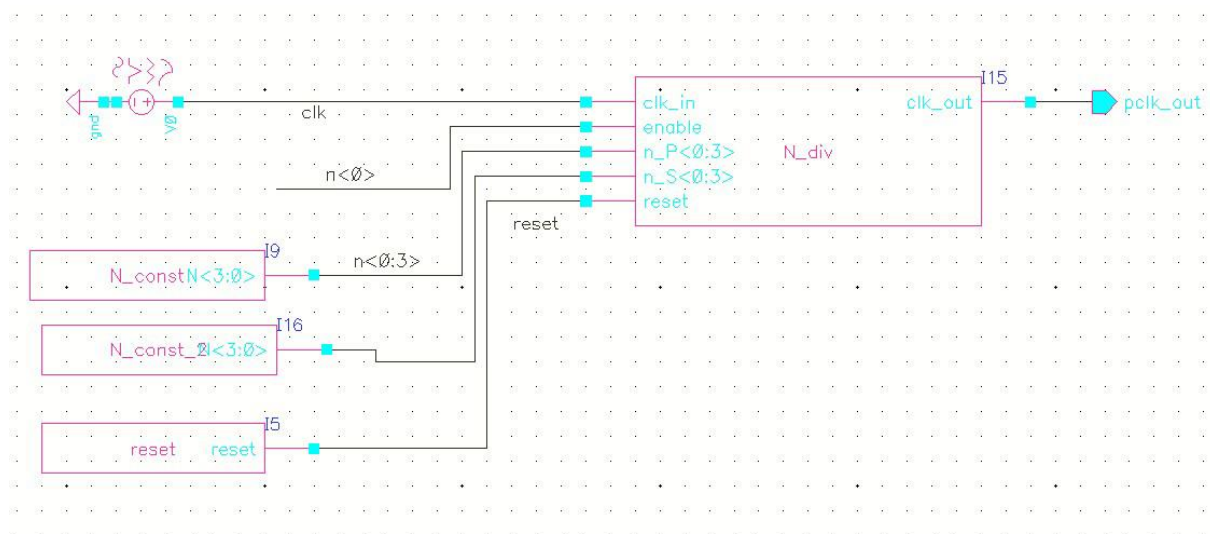


Figure 4. 18 : Setup de la simulation du diviseur de fréquence

La figure 4.19 montre le résultat de la simulation. Les signaux représentés sont de haut en bas, l'entrée 'clk\_in', la sortie 'pclk\_out', le reset, la sortie du prédiviseur, et le vecteur de commande. Le vecteur de commande est arrangé de haut en bas du bit le moins significatif au bit le plus significatif.

La position des marqueurs correspond aux intervalles de temps où la valeur du vecteur de contrôle du compteur S prend les combinaisons ci-dessous et leurs périodes respectives sont données dans la figure 4.20. Ces périodes doivent être en accord avec les valeurs théoriques exprimées dans (4.6)

$$\begin{aligned} A, B &\rightarrow (0000)_2 = (0)_{10} \\ C, D &\rightarrow (0100)_2 = (4)_{10} \\ E, F &\rightarrow (0101)_2 = (5)_{10} \end{aligned} \quad (4.5)$$

$$\begin{aligned} T_{out}^0 &= (32 \times 15 + 0)T_{in} = 196.8ns \\ T_{out}^4 &= (32 \times 15 + 4)T_{in} = 198.44ns \\ T_{out}^5 &= (32 \times 15 + 5)T_{in} = 198.85ns \end{aligned} \quad (4.6)$$

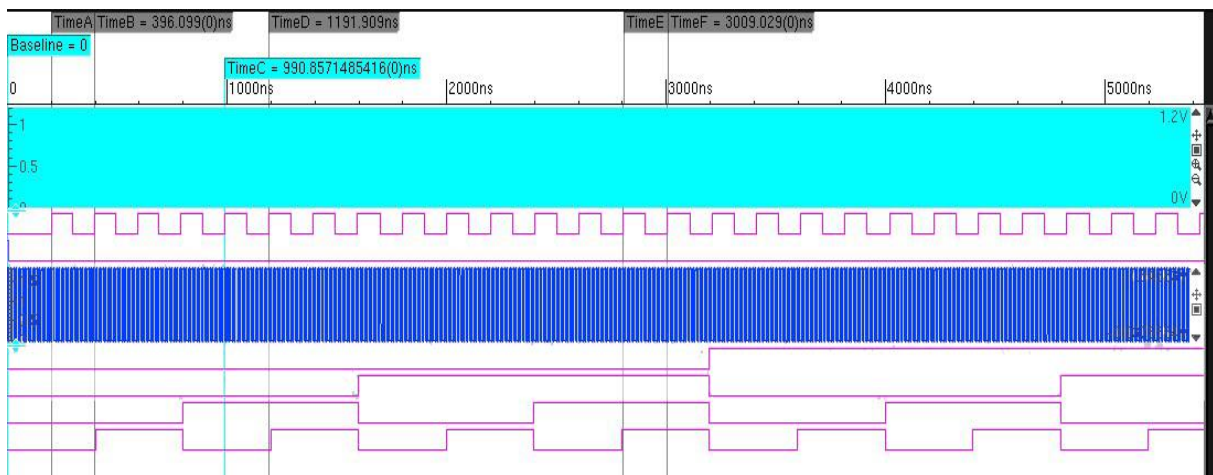


Figure 4. 19 : Résultats de la simulation du diviseur de fréquence

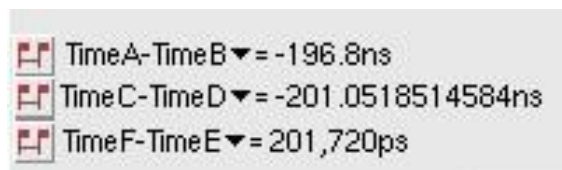


Figure 4. 20 : Périodes du signal pour S=0, S= 4 et S=5

On remarque qu'il y a une différence de 2.7 ns entre les périodes des divisions simulées et celle de la théorie (4.6), et ce, pour le mode 4 et 5. En effectuant une analyse inverse, les périodes observées dans ces deux modes correspondent à un rapport de division par 10 et par 12 respectivement. Ce problème a besoin de plus d'investigation.

### 4.3. Conception de la PFD

La PFD fonctionne à une fréquence d'horloge qui est la fréquence de référence 5MHz, nous pouvons la synthétiser selon le flot de conception digitale à l'aide des cellules standards. Le code Verilog de la PFD conçu dans le chapitre 2 est synthétisé en utilisant le compilateur RTL de l'outil Encounter.

La synthèse donne l'architecture de la figure 4.21 qui est constituée de deux bascules D, d'une porte AND et d'une porte OR. Le code verilog issu de la synthèse est présenté dans la figure 4.22. Le résultat de simulation de cette synthèse donne exactement le même comportement que le PFD exposé dans le chapitre 2, ainsi nous n'avons pas présenté le résultat dans cette section.

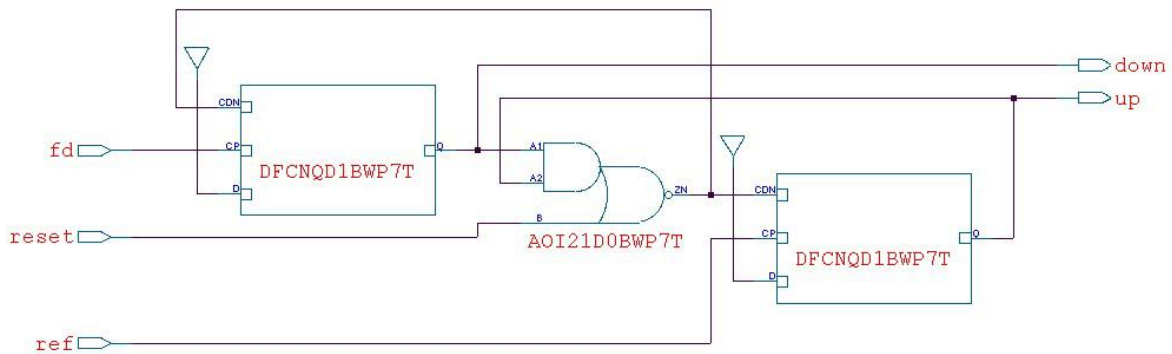


Figure 4. 21 : Schématique de la PFD synthétisé

```
1
2 // Generated by Cadence Encounter(R) RTL Compiler v14.10-s022_1
3
4 // Verification Directory fv/pfd
5
6 module pfd(up, down, fd, ref, reset);
7   input fd, ref, reset;
8   output up, down;
9   wire fd, ref, reset;
10  wire up, down;
11  wire n_1;
12  DFCNQD1BWP7T q0_reg(.CDN (n_1), .CP (fd), .D (1'b1), .Q (down));
13  DFCNQD1BWP7T q1_reg(.CDN (n_1), .CP (ref), .D (1'b1), .Q (up));
14  AOI21D0BWP7T g48(.A1 (down), .A2 (up), .B (reset), .ZN (n_1));
15 endmodule
16 |
```

Figure 4. 22 Code verilog niveau porte logique synthétisé

#### 4.4. Conclusion

Dans ce chapitre un diviseur de fréquence a été conçu en technologie CMOS. L'architecture retenue a mené vers un diviseur de fréquence reconfigurable vue que son rapport de division peut être modifié à travers un vecteur numérique externe. La conception a été validée par simulation sous Cadence, en commençant par les circuits élémentaires du diviseur jusqu'à sa réalisation complète.

Une autre conception a été présentée portant sur la synthèse du code Verilog du PFD pour obtenir son circuit ayant un niveau d'abstraction niveau portes logiques. Ces dernières ont été implémentées moyennant les cellules standards fournies par le design kit de la technologie CMOS 0.18- $\mu\text{m}$ . Nous avons noté que la simulation à ce niveau d'abstraction a donné les mêmes résultats que la simulation comportementale du chapitre 2, confirmant ainsi la conception ascendante du PFD.

Bien que nous avons envisagé d'insérer ces deux circuits (PFD et diviseur) dans le synthétiseur, comme nous l'avons fait avec le VCO LC dans le chapitre 3, ceci n'a pas pu se produire par faute de temps. En conséquence nous allons le projeté comme perspectives.

# *Conclusion Générale*

## *Conclusion générale*

Les travaux menés pour la conception de cette PLL nous ont permis d'acquérir un background remarquable relevant de la thématique de la conception RFIC. Nous avons pu aborder la conception de circuits analogiques, la conception de circuits numériques, et aussi de la conception de circuits mixtes vu que la PLL fait partie de cette dernière catégorie. En outre, ce mémoire nous a permis de travailler à plusieurs niveaux d'abstraction, à savoir, le niveau système à travers une conception comportementale de la PLL, un niveau d'abstraction transistor en manipulant des dispositifs en technologie CMOS et aussi un niveau d'abstraction porte logique. De plus, une maîtrise notable a été acquise des outils de CAO et de langages de programmation, sachant que nous avons effectué durant le présent travail trois types de simulations qui sont : une simulation de circuit analogique pour le VCO, un autre type de simulation pour les circuits digitaux comme le PFD et le diviseur de fréquence, et enfin une simulation mixte pour tout le système (PLL).

Ainsi lors de la conception nous avons ciblé le standard du Zigbee IEEE 802.15.4, et nous avons procédé selon les étapes suivantes :

Dans un premier temps, nous avons passé en revue la théorie relative aux circuits de génération de fréquence. Un constat très important a été noté, c'est que dans le cas des applications RF on parle de synthétiseur de fréquence qui est une PLL qui contient dans sa boucle de retour un diviseur de fréquence qui permet d'offrir une variété de fréquences mais qui sont toutes multiples de la fréquence de référence. Aussi, la modélisation des éléments constituant le synthétiseur a été présentée et qui a fait une base pour les travaux élaborés dans les chapitres qui suivent.

Sachant que nous avons opté pour une approche de conception ascendante, dans le deuxième chapitre nous avons abordé la conception niveau système en effectuant une étude comportementale du synthétiseur. Tous les éléments du système ont été modélisés avec des codes en Verilog, puis une simulation AMS (Analog Mixed Signal) a été effectuée pour évaluer le comportement de tout le système. Des résultats satisfaisants ont été obtenus vu qu'ils répondent aux spécifications exigées par le standard ciblé.

Une fois la conception niveau système validée, nous avons passé à une deuxième étape qui consiste à remplacer les blocs comportementaux par des circuits en technologie CMOS. A cet effet, nous avons présenté dans le chapitre 3 une conception complète d'un VCO LC en

technologie CMOS 0.18- $\mu\text{m}$ . Les performances du VCO conçu sont notables comparativement à l'état de l'art, à savoir, une consommation d'environ 5.4mW, une excursion du signal de sortie de plus de 1V, une pureté spectrale remarquable avec une forme d'onde sinusoïdale quasi-parfaite, et aussi une plage d'accord allant de 2.36 à 2.6 GHz. L'insertion du VCO dans la PLL a donné un fonctionnement parfait de cette dernière et similaire au fonctionnement noté dans le chapitre 2, ce qui confirme l'adéquation de la conception comportementale et du VCO aussi.

Enfin, comme dernière étape, nous avons essayé de concevoir les blocs digitaux du synthétiseur. Nous avons conçu un diviseur de fréquence programmable à partir d'une commande logique. Ce dernier a été conçu en commençant d'abord par leurs circuits élémentaires puis le diviseur complet a été validé par simulation où nous avons testé plusieurs rapports de division en variant la commande logique. Aussi, le PFD a été conçu à l'aide de la synthèse de son code verilog qui a été élaboré dans le chapitre 2. La circuiterie niveau porte logique du PFD a été obtenue en suivant le flot de synthèse supporté sur Cadence. La simulation a donné des résultats similaires à ceux exposés lors de l'étude comportementale. Néanmoins, nous ne sommes pas parvenues à insérer ces deux circuits (diviseur et PFD) dans la PLL faute de temps car il fallait maîtriser une autre étape dans le flot de conception sous Cadence.



**Références bibliographiques**

- [1] B. Razavi, Design of Monolithic Phase Locked Loops and Clock Recovery Circuits : A Tutorial, Wiley-IEEE Press, 1996
- [2] S. I. Ahmed “Submicron CMOS Components for PLL Based Frequency Synthesis”, these de Master, Carleton University, Ottawa, Canada, 2002
- [3] V. Reinhardt, K. Gould, K. McNab, and M. Bustamante “A Short Survey of Frequency Synthesis Techniques”, 40th Annual Frequency Control Symposium, pp. 355-365, 1986
- [4] IEEE Standard Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPAN)
- [5] J. Rutman, “Characterization of Phase and Frequency Instabilities in Precision Frequency Sources: Fifteen Years of Progress” Proc. IEEE, vol. 66, pp. 1048-1174, Sept. 1978
- [6] A. Hajimiri, “A General Theory of Phase Noise in Electrical Oscillators”, IEEE Journal of Solid-State Circuits, Vol. 33, pp. 179-194, Feb. 1998
- [7] D. Banerjee ,PLL Performance, Simulation and Design , 2<sup>nd</sup> Edition, 2001
- [8] F. M. Gardner, Phase-lock Techniques, Wiley, New York, 3<sup>rd</sup> edition 2005.
- [9] P. ACCO, "Etudes de la boucle à verrouillage de phase par impulsions de charge", Thèse doctorat, INSA, Toulouse, France, 2003.
- [10] L. F. Tanguay, “ Synthétiseur de fréquence RF destiné aux dispositifs Médicaux Implantable “, thèse de doctorat, Université de Montreal, Montreal, Canada, 2010.
- [11] V. Lagareste “ Contribution à l’étude de nouvelles architectures de synthétiseur de fréquence», thèse de doctorat, Université de Bordeaux 1, Bordeaux, France, 2006
- [12] F. M. Gardner, “Charge-Pump Phase-Lock Loops”, IEEE Tans. Communications, vol. 28, pp. 1849-1858, Nov. 1980
- [13] F. Manneville ; J. Esquieu , Systèmes Bouclés Linéaires, de Communication et de Filtrage , Dunod, 1998
- [14] k. Shu; E. Sinchez-Sinencio, CMOS PLL Synthesizers: Analysis and Design, Springer, 2005
- [15] W. F. Egan, Frequency Synthesis by Phase Lock, 2<sup>nd</sup> edition, J. Wiley & Sons, 1998

- [16] K. Kundert ; H. Chang; D.Jefferies; G. Lamant ; E. Malavasi ; F. Sendig “Design of Mixed-Signal Systems on Chip” IEEE Trans Computer-Aided Design of Integrated Circuits and Systems, vol. 19, no. 12, pp. 1561-1571, Dec 2000
- [17] Site internet : <https://stackoverflow.com/tags/matlab/info>
- [18] K. S. Kundret ; O. Zinke, The Designer’s Guide to Verilog-AMS, 1<sup>st</sup> edition, Kluwer Academic Publishers, 2004
- [19] A. Taibi “ Conception d’Amplificateurs RF Faible Bruit en Technologie CMOS”, thèse de doctorat, Ecole Nationale Polytechnique, Alger, Algérie, 2017
- [20] Site internet:[http://www2.ece.ohio-state.edu/~bibyk/ece822/MSDK\\_July\\_2003.pdf](http://www2.ece.ohio-state.edu/~bibyk/ece822/MSDK_July_2003.pdf)
- [21] B. Razavi, RF Microelectronics, Prentice Hall, 1998
- [22] P. Gorday, “802.15.4 Multipath”, Motorola, July 2004.
- [23] N.J. Oh, and S-G. Lee, “A CMOS 868/915 MHz Direct Conversion Zigbee Single-Chip Radio”, IEEE Communications Magazine, vol. 43, issue 12, pp.100-109, Dec. 2005.
- [24] B. Razavi, Fundamentals of Microelectronics, 2<sup>nd</sup> edition, J. Wiley & Sons, 2013
- [25] F. Ellinger, Radio Frequency Integrated Circuits and Technologies, Springer, 2007
- [26] P. Andreani; S. Mattisson, “ On the Use of MOS Varactors in RF VCO’s”, IEEE Journal of Solid-State Circuits, Vol. 35, pp. 905-910, June. 2000
- [27] B. Razavi, Design of Analog CMOS Integrated Circuits, McGraw Hill ,2001
- [28] M. Sié, “ Synthétiseurs de fréquence monolithiques micro-ondes à 10 et 20 GHz en technologies BiCMOS SiGe 0.25 et 0.35  $\mu\text{m}$  ”, thèse de doctorat, Université de Toulouse, Toulouse, France, 2004
- [29] S. Lee and al., “ A Low Power Injection Locked LC-tanks with Current Reused Topology”, IEEE Microwave Wireless Components Letters, vol 17, no.3, pp. 220-222, March 2007
- [30] B. Perumana ; S. Chakraborty, C. H. Lee ; J. Laskar, “ A Low-Power Fully Monolithic Subthreshold CMOS Receiver with Integrated LO Generation for 2.4 GHz Wireless PAN Applications” IEEE, Journal of Solid-State Circuits, vol. 43 , no. 10, pp. 2229-2238, Oct. 2008
- [31] T. C. Lee ; B. Razavi, “ A stabilization Technique for Phase-Locked Frequency Synthesizers”, IEEE Journal of Solid-State Circuits, vol. 38, pp. 888-894, June 2003.
- [32] M. Debashis ; T. K. Bhattacharyya, “ Implementation of CMOS low power Integer N Frequency Synthesizers for SOC Design”, Journal of Computers, vol.3, no. 4, pp. 31-38, April 2008

- [33] M. Vamshi Krishna, “ Ultra Low Power CMOS Phase-Locked Loop Frequency Synthesizers”, thèse de doctorat, Université Technologique de Nanyang, Nanyang, Singapore, 2011
  
- [34] H. Joshi ; S. M. Ranjan ; V. Nath, « Design of High speed Flip-Flop Based Frequency Dividers for GHz PLL System : Theory and design Techniques in 250 nm CMOS Technology », IJECSE
  
- [35] Site internet : <http://sugandhasharma.weebly.com/computer-architecture--digital-vlsi.html>

# **Annexe**

## Code Matlab de l'analyse/conception système

```
1
2 %% initialisation de l'environnement Matlab
3 close all;
4 clear all;
5 clc;
6 %% Specification niveau systeme
7
8 % frequence de reference:
9 Fxtal = 10e6;
10 Fref = 5e6; % hertz
11 R = Fxtal / Fref ;
12 % frequence de sortie:
13 Fmax = 2480e6; % Fout maximale
14 Fmin = 2400e6; % Fout minimale
15 Nmoy = round((Fmax + Fmin)/(2 *Fref )) ; %rapport de division moyen
16
17 % bande passante de la pll:
18 F3dB = Fref /15; % hertz
19 omega_3dB = 2*pi*F3dB; % radians / second
20
21 % Marge de phase:
22 PM = 53; % degrees
23
24 %%
25 LW = 2; %line width
26
27 %Modele du bruit de phase de la reference
28 Ref_PN_Level = [-52 -92 -117 -137 -150 -154 -154 -154]; % dBc/Hz
29 Ref_PN_Freq = [1 10 100 1e3 10e3 100e3 1e6 10e6]; % Hz
30 Ref_Filter_Length = 2^12; % FIR filter for phase noise generation
31 %modele du bruit de phase du VCO
32 VCO_PN_Level = [22 -1 -24 -47 -70 -93 -116 -131 ]+15; % dBc/Hz
33 VCO_PN_Freq = [1 10 100 1e3 10e3 100e3 1e6 10e6]; % Hz
34 VCO_Filter_Length = 2^14; % FIR filter for phase noise generation
35
36 %Affichage du bruit de phase du VCO et de la Reference
37 noise_fig = figure;
38 semilogx(Ref_PN_Freq,Ref_PN_Level+20*log10(Nmoy)-20*log10(R),'Color','r','LineWidth',LW);
39 hold;
40 semilogx(VCO_PN_Freq,VCO_PN_Level,'Color','b','LineWidth',LW);
41 legend('Amplified Ref Noise','VCO Noise');
42 xlabel('Frequency (Hz)');
43 ylabel('Noise Spectra (dB)');
44 grid;
45
46
47 %% parametres systeme
48
49 % calculer b de la marge de phase souhaitée
50 PHI = [1 -2*(2*tand(PM)^2+1) 1];
51 b = max(roots(PHI));
52 % facteur d'amortissement :
53 zeta = (sqrt(b)-1)/2;
54 % la frequence naturelle:
55 omega_0 = omega_3dB ;
56
57 %%
58 % Modele de reference - fonction de transfert en boucle fermee:
59 ClosedLoop = Nmoy * tf(1,[1 omega_0]) * tf([(2*zeta+1)*omega_0^2 omega_0^3],[1 2*zeta*omega_0 omega_0^2]);
60
61 %% affichage le la fonction de transfert
62 fprintf('Fonction de transfert en boucle fermée:\n');
63 display(ClosedLoop);
64
```

```

64
65 %% Definir les parametres blocs
66
67 % Gain du PFD:
68 Kn = 1/Nmoy;
69
70 % VCO:
71
72 Fv = 160e6; % hertz / volt
73 Kv = 2*pi*Fv; % radias / volt-second
74
75 % PFD:
76 I_0 = 50e-6;
77 Kp = 0.8*I_0/pi; % Amper/radian
78
79 %% Calcul de la fonction de transfert du filtre
80 %
81 % Filtre de boucle:
82 Filtre = minreal(tf([1 0],Kp*Kv*Kn)*feedback(Kn*ClosedLoop,-1));
83
84 %% Analyse du filtre de boucle
85
86 %Zeros, Poles et gain du compensateur
87 [Z,P,K_1f] = zpndata(zpk(Filtre));
88 Kf = K_1f/P{1}(2);
89
90 % circuit equivalenteNt
91 R1 = (omega_3dB) / (Kv * Kp * Kn )
92 C1 = 1 / (R1 * -Z{1})
93 C2 = 1 / (R1 * -P{1}(2))
94
95 %% Modele du PLL
96
97 % Diviseur:
98 Diviseur = tf(Kn,1);
99
100 % VCO:
101 VCO = tf(Kv,[1 0]);
102
103 % PFD:
104 PFD = tf(Kp,1);
105
106 PLL.OpenLoop = tf(PFD*Filtre*VCO*Diviseur,1);
107 sisofig = sisotool('bode',PLL.OpenLoop);
108
109 PLL.ClosedLoop = minreal(feedback(PFD*Filtre*VCO,Diviseur));
110
111 %% comparaison du modele de la PLL avec le modele de reference
112
113 % Reference:
114 fprintf('\n\n');
115 fprintf('Modele de reference: \n');
116 display(ClosedLoop);
117
118 % modele de la PLL:
119 fprintf('\n\n');
120 fprintf('Modele de la PLL: \n');
121 display(PLL.ClosedLoop);
122
123 %% Prediction du bruit de phase
124 % Creation des fonctions de transfert de la reference et du VCO
125 CL_phase_in_vco_out = feedback((Filtre*VCO),Diviseur);
126 CL_phase_disturbance_in_vco_out = feedback(1,(Filtre*VCO*Diviseur));

```

```

126 CL_phase_disturbance_in_vco_out = feedback(1,(Filtre*VCO*Diviseur));
127
128 % Definir la plage de frequence
129 fmin = min([VCO_PN_Freq,Ref_PN_Freq]);
130 fmax = max([VCO_PN_Freq,Ref_PN_Freq]);
131 Npts = 1000; % nombre de points de frequence
132 freqv = logspace(log10(fmin),log10(fmax),Npts); % vecteur de frequence
133
134 % Compute frequency response of PLL ref input to PLL output
135 H_ph2VCO = squeeze(freqresp(CL_phase_in_vco_out,2*pi*freqv));
136 H_dB = 20*log10(abs(H_ph2VCO)); % note the transpose
137 Interp_Type = 'linear';
138
139 % Interpolate measured reference phase noise over same set of frequencies
140 Ref_Phase_Noise_dB = interp1(log10(Ref_PN_Freq),Ref_PN_Level,log10(freqv),Interp_Type);
141 G_dist2VCO = squeeze(freqresp(CL_phase_disturbance_in_vco_out,2*pi*freqv));
142 G_dB = 20*log10(abs(G_dist2VCO)); % note the transpose
143 VCO_Phase_Noise_dB = interp1(log10(VCO_PN_Freq),VCO_PN_Level,log10(freqv),Interp_Type);
144
145 %Gardner's fig
146 TF_figure=figure;
147 semilogx(freqv,Ref_Phase_Noise_dB,'color','blue','LineWidth',LW); hold on;
148 semilogx(freqv,H_dB,'color','red','LineWidth',LW);
149
150 semilogx(freqv,VCO_Phase_Noise_dB,'color','blue','LineWidth',LW,'LineStyle','--'); hold on;
151 semilogx(freqv,G_dB,'color','red','LineWidth',LW,'LineStyle','--');
152
153 % Noise power from both sources must be added and then total power
154 % converted back to dBc
155 Ptotal_1 = 10.^((VCO_Phase_Noise_dB+G_dB)/10)+10.^((Ref_Phase_Noise_dB+H_dB)/10);
156 semilogx(freqv,(10*log10(Ptotal_1)),'color',[0 0 0],'LineWidth',LW);
157 legend({'Reference Phase Noise',...
158         'Reference Phase in to PLL out TF',...
159         'VCO Phase Noise',...
160         'VCO Phase in to PLL out TF',...
161         'Total PLL Phase Noise'},'Location','SouthEast');
162 ylabel('dBc/Hz for Noise, dB for TF'); xlabel('Hz'); title('PLL Phase Noise Calculations');
163 hold off;
164 grid

```

## Description comportementale du compteur 4 bits

```
counter_4b.v X
1 //Verilog HDL for "mixed_pll", "prog_div" "functional"
2
3
4 module counter_4b ( reset, clk, enable, n, clk_out );
5
6 input clk;
7 input reset;
8 input enable;
9 input [3:0] n;
10 output clk_out;
11
12 wire [3:0] m;
13 wire dbn_en;
14 reg [3:0] count;
15 reg out1;
16 reg out2;
17 wire out;
18 wire clk_out;
19
20 assign dbn_en = n[3] | n[2] | n[1];
21
22 always @(posedge clk or posedge reset) begin
23 if (reset==1) begin
24 out1<=1'b0;
25 count<=4'h00;
26 end
27 else if (dbn_en==1 && enable==1) begin
28 if (n[0]==0) begin // even count
29 if (count==m-1)begin
30 count<=4'h00;
31 out1<=~out1;
32 end
33 else
34 count<=count+1;
35 end
36 else if (count==n-1)begin // odd count
37 count<=4'h00;
38 out1<=~out1;
39 end
40 else
41 count<=count+1;
42 end
43 end
44
45 assign m=n>>1;
46
47 always @(negedge clk or posedge reset) begin
48 if (reset==1) begin
49 out2<=1'b0;
50 end
51 else if (count==m && enable==1)
52 out2<=out1;
53 end
54
55 assign out=(enable==1)?((dbn_en==0)? clk :(n[0]==1)? out1^out2 : out1):1'b0;
56
57 assign clk_out = out;
58
59 endmodule
```



## Code Verilog après synthèse du Compteur niveau porte logique

```
counter_4b.v counter_4b.v
1
2 // Generated by Cadence Encounter(R) RTL Compiler v14.10-s022_1
3
4 // Verification Directory fv/counter_4b
5
6 module counter_4b(reset, clk, enable, n, clk_out);
7   input reset, clk, enable;
8   input [3:0] n;
9   output clk_out;
10  wire reset, clk, enable;
11  wire [3:0] n;
12  wire clk_out;
13  wire [3:0] count;
14  wire UNCONNECTED, n_0, n_1, n_2, n_3, n_4, n_5, n_6;
15  wire n_7, n_8, n_9, n_10, n_11, n_12, n_13, n_14;
16  wire n_15, n_16, n_17, n_18, n_19, n_20, n_21, n_22;
17  wire n_23, n_24, n_25, n_26, n_27, n_28, n_30, n_31;
18  wire n_32, n_33, n_34, n_35, n_36, n_37, n_38, out1;
19  wire out2;
20  OAI32D0BWP7T g879(.A1 (count[2]), .A2 (n_33), .A3 (n_34), .B1 (n_6),
21    .B2 (n_35), .ZN (n_37));
22  OAI22D0BWP7T g880(.A1 (n_35), .A2 (n_10), .B1 (n_34), .B2 (n_7), .ZN
23    (n_36));
24  DFCNQD1BWP7T \count_reg[0] (.CDN (n_38), .CP (clk), .D (n_32), .Q
25    (count[0]));
26  AOI21D0BWP7T g884(.A1 (n_27), .A2 (n_33), .B (n_31), .ZN (n_35));
27  SDFCNQD1BWP7T out1_reg(.CDN (n_38), .CP (clk), .D (out1), .SI (n_28),
28    .SE (n_26), .Q (out1));
29  MOAI22D0BWP7T g885(.A1 (n_34), .A2 (count[0]), .B1 (count[0]), .B2
30    (n_31), .ZN (n_32));
31  MOAI22D0BWP7T g886(.A1 (n_34), .A2 (n_1), .B1 (count[1]), .B2 (n_31),
32    .ZN (n_30));
33  OAI221D0BWP7T g891(.A1 (n_28), .A2 (n_18), .B1 (n_0), .B2 (n_23), .C
34    (n_24), .ZN (clk_out));
35  INV00BWP7T g888(.I (n_34), .ZN (n_27));
36  NR2D0BWP7T g887(.A1 (n_25), .A2 (n_31), .ZN (n_26));
37  IND2D0BWP7T g889(.A1 (n_31), .B1 (n_25), .ZN (n_34));
38  ND3D0BWP7T g893(.A1 (n_22), .A2 (n_23), .A3 (enable), .ZN (n_24));
39  OAI21D0BWP7T g892(.A1 (n_20), .A2 (n_0), .B (n_21), .ZN (n_25));
40  OAI22D0BWP7T g896(.A1 (n_19), .A2 (out1), .B1 (out2), .B2 (n_28), .ZN
41    (n_22));
42  SDFNCND1BWP7T out2_reg(.CDN (n_38), .CPN (clk), .D (out2), .SI
43    (out1), .SE (n_17), .Q (out2), .QN (UNCONNECTED));
44  OAI31D0BWP7T g897(.A1 (count[0]), .A2 (n_2), .A3 (n_11), .B (n_0),
45    .ZN (n_21));
46  NR4D0BWP7T g894(.A1 (n_13), .A2 (n_16), .A3 (n_8), .A4 (count[3]),
47    .ZN (n_20));
48  ND2D0BWP7T g900(.A1 (out2), .A2 (n_18), .ZN (n_19));
49  NR2D0BWP7T g895(.A1 (n_9), .A2 (count[3]), .ZN (n_17));
50  OR2D0BWP7T g903(.A1 (n_31), .A2 (n_0), .Z (n_18));
51  MOAI22D0BWP7T g901(.A1 (n_15), .A2 (n_14), .B1 (n_15), .B2 (n_14),
52    .ZN (n_16));
53  MOAI22D0BWP7T g902(.A1 (n_12), .A2 (count[0]), .B1 (n_12), .B2
54    (count[0]), .ZN (n_13));
55  OAI221D0BWP7T g906(.A1 (n_3), .A2 (n_1), .B1 (n_3), .B2 (n_10), .C
56    (n_4), .ZN (n_11));
57  ND4D0BWP7T g898(.A1 (n_12), .A2 (n_15), .A3 (n_8), .A4 (enable), .ZN
58    (n_9));
59  OA32D0BWP7T g899(.A1 (count[3]), .A2 (n_6), .A3 (n_33), .B1
60    (count[2]), .B2 (n_10), .Z (n_7));
61  IND2D0BWP7T g904(.A1 (n_5), .B1 (enable), .ZN (n_31));
62  ND2D0BWP7T g905(.A1 (n_5), .A2 (enable), .ZN (n_23));
63  AOI22D0BWP7T g908(.A1 (n_10), .A2 (n_3), .B1 (n_3), .B2 (n_1), .ZN
64    (n_4));
65  MOAI22D0BWP7T g912(.A1 (n_6), .A2 (n_2), .B1 (n_6), .B2 (n_2), .ZN
66    (n_3));
```

```

65 MOAI22D0BWP7T g912(.A1 (n_6), .A2 (n[2]), .B1 (n_6), .B2 (n[2]), .ZN
66 (n_2));
67 MAOI22D0BWP7T g913(.A1 (n_3), .A2 (count[0]), .B1 (n_3), .B2
68 (count[0]), .ZN (n_1));
69 NR3D0BWP7T g907(.A1 (n[1]), .A2 (n[3]), .A3 (n[2]), .ZN (n_5));
70 NR2D0BWP7T g914(.A1 (n_3), .A2 (n[2]), .ZN (n_14));
71 MOAI22D0BWP7T g909(.A1 (count[2]), .A2 (n[3]), .B1 (count[2]), .B2
72 (n[3]), .ZN (n_15));
73 MOAI22D0BWP7T g910(.A1 (count[0]), .A2 (n[1]), .B1 (count[0]), .B2
74 (n[1]), .ZN (n_8));
75 MOAI22D0BWP7T g911(.A1 (count[1]), .A2 (n[2]), .B1 (count[1]), .B2
76 (n[2]), .ZN (n_12));
77 ND2D0BWP7T g915(.A1 (count[0]), .A2 (count[1]), .ZN (n_33));
78 INV0BWP7T g920(.I (clk), .ZN (n_0));
79 INV0BWP7T g921(.I (reset), .ZN (n_38));
80 INV0BWP7T g918(.I (out1), .ZN (n_28));
81 DFCND1BWP7T \count_reg[1] (.CDN (n_38), .CP (clk), .D (n_30), .Q
82 (count[1]), .QN (n_3));
83 DFCND1BWP7T \count_reg[2] (.CDN (n_38), .CP (clk), .D (n_37), .Q
84 (count[2]), .QN (n_6));
85 DFCND1BWP7T \count_reg[3] (.CDN (n_38), .CP (clk), .D (n_36), .Q
86 (count[3]), .QN (n_10));
87 endmodule
88

```

### Simulation de la bascule D- TSPC

D est l'entrée (T= 600 ps), clk est l'horloge (T= 416 ps) et Q est la sortie

