

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد- تلمسان -

Université Aboubakr Belkaïd- Tlemcen -

كلية التكنولوجيا

Faculté de TECHNOLOGIE

Département de Génie Electrique et Electronique (GEE)

Filière :Electronique



MASTER INSTRUMENTATION

PROJE DE FIN D'ETUDES

Présentées par : DJEBBAR Nour-El-Houda & HADDOUINE Fatima-Zahra

Intitulé du sujet

Etude simulation et implémentation de la PLL sur FPGA

Soutenu le 11/07/ 2019, devant le jury compose de :

Mr H.BECHAR	MA	Univ. Tlemcen	Président
Mr S.KAMECHE	Professeur	Univ. Tlemcen	Examineur
Mr S.M.H. IRID	MCA	Univ. Tlemcen	Encadreur
Mme A.BABA-HAMED	DOCTUER	Univ. Tlemcen	CO-Encadreur

Année universitaire 2018-2019

DÉDICACES

*Je dédie ce modeste travail aux être qui me sont les plus chers,
A mes parents les plus chers de monde, Papa et maman que dieu les garde
et les protège pour moi et j'espère que vous êtes fière de moi, je suis là
grâce à vous.*

*A mon deuxième père mon oncle « Djebbar Ahmed », merci beaucoup pour
tous de mes premiers pas jusqu'à maintenant.*

*A ma chère sœur Zouaouia, ma plus proche amie, qui m'accompagne
depuis mes premiers pas au cours de mes études, pour leurs
encouragements permanents, et leur soutien moral,
A mes chers frères Mohammed –el-Amine et Farouk pour leur appui et
leur encouragement,
A mon frère « Abdelghani » et sa petite famille
A mon future mari Houcine,*

*A toute ma famille pour leur soutien tout au long de mon parcours
universitaire,
que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de
votre soutien infailible,*

*A tous mes amies en particulier H. Fatima-Zohra, très reconnaissante à
notre amitié, que ses années de mes études sont passé merveilleux avec toi,
aussi*

*Hafsa, Adelhaq, Aderrahmane, B.Fatima, H.Hanen, Youcef, Amina et
Bilel*

Toute le promotion d'ectronique d'instrumentation 2019

Merci d'être toujours là pour moi et avec moi.

Nour-el-Houda

*Je dédie ce modeste travail aux être qui me sont les plus chers,
À mes chers parents, Je vous remercie pour tout le soutien et l'amour que
vous me portez depuis mon enfance et j'espère que votre bénédiction
m'accompagne toujours.*

*À mon petit frère Yassine, qui sait toujours apporter joie et bonheur à toute
la famille.*

A mon frère Mohammed et ma sœur Wafaa.

*À mes professeurs qui doivent voir dans ce travail la fierté des
connaissances bien acquises.*

*Sans oublier mon binôme Houda pour son soutien moral, sa patience et sa
compréhension tout au long de ce projet*

*Mes chers cousins et cousines, spécialement : Fatima, Sara, Mohammed,
Imrane et Karima*

À tous mes amies, en particulier : Houda, Imane, Hayet, Wahiba,

À tous mes collègues de la promo instrumentation

A tous ceux que j'aime

Merci

Fatima zahra

REMERCIEMENTS

On remercie tout d'abord « Allah » de nous avoir donnée le courage d'entamer et de finir ce mémoire dans de bonnes conditions.

On remercie vivement notre encadreur, Monsieur « SIDI MOHAMMED HADJ IRID », Maître de Conférence classe B à l'Université de Tlemcen, d'avoir encadré ce travail avec beaucoup de compétences, ses grandes qualités humaines. Merci pour l'acuité de ses critiques et ses conseils éclairés.

Nos vifs remerciements aussi à notre Co-encadreur Docteur « BABA HAMDE AMEL », il n'y a pas de mots pour exprimer nos remerciements et notre gratitude pour ce qu'elle nous a donné, et fait pour nous, merci pour son indéfectible disponibilité, sa rigueur scientifique et la confiance qu'elle nous 'avez accordée au cours de l'élaboration de ce mémoire, merci pour ses conseils, ses aides, et ses grandes qualités humaines. Merci beaucoup, ce travail a été fini grâce à vous.

Nos remerciements aussi vient à Monsieur « SAMIR KAMECH ». Professeur à l'Université de Tlemcen, merci pour ses conseils et ses aides et pour l'intérêt qu'il a porté à notre travail en acceptant d'examiner ce mémoire.

Aussi Monsieur « BECHAR HASSEN » Maître-Assistant à l'Université de Tlemcen ses aides et merci aussi pour l'honneur qu'il nous 'a fait en acceptant de présider le jury de ce mémoire.

On remercie tous les enseignants qui ont contribué à notre formation durant tous ces années.

Ces quelques lignes ne pourront jamais exprimer la reconnaissance que nous éprouve envers tous ceux qui, de près ou de loin, ont contribué par leurs conseils, leurs aides et leurs encouragements pour l'aboutissement de ce travail.

Merci.

TABLE DES MATIÈRES

Dédicaces	i
Remerciements.....	iii
Tables des matières	iv
Liste des figures.....	viii
Liste des tableaux.....	x
Listes des organigrammes.....	x
Sigles et abréviations	xi
Introduction générale.....	01

Chapitre I : La théorie de la PLL

I.1. Intorduction.....	5
I.2. Les différents types de PLLs.....	5
I.3. Définition et principe de fonctionnement... ..	5
I.4. PLL analogique.....	6
I.4.1. Eléments constitutifs de la PLL analogique.....	7
a. comparateur de phase	7
b. Filtre de boucle	8
c. VCO.....	9
d. Diviseur	10
I.4.2. Fonctionnement d'une PLL à fréquence fixe.....	11
I.4.3. La PLL en régime dynamique : fréquence variable.....	12
I.5. Etude de bruit de phase.....	13
I.6. Fonctions de transfert général de la PLL.....	14
I.7. PLL numérique (digital).....	17
I.8. Conclusion.....	19

Chapitre II : Simulation d'un synthétiseur de fréquence a bas de PLL

II.1. Introduction.....	22
II.2. Le synthétiseur de fréquences.....	22
II.3. Les caractéristiques les plus importantes des synthétiseur de fréquences.....	22
II.3.1. Plage de fréquences et pas de synthèse.....	22
II.3.2. Plage d'accorde.....	23
II.3.3. Pureté spectrale.....	23
II.3.4. Temps d'acquisition.....	23
II.4. Principes de base du synthétiseur PLL.....	23
II.4.1. Synthétiseur de fréquences à division entière.....	23
II.4.2. Synthétiseur de fréquences à division fractionnaire.....	24
II.5. Description les blocs de synthétiseur de fréquence.....	25
II.5.1. La source de référence.....	25
II.5.2. Comparateur phase- fréquence.....	25
II.5.3. La pompe de charge.....	26
II.5.4. Le filtre de boucle.....	26
II.5.5. L'Oscillateur Contrôlé en Tension (VCO).....	27
II.5.6. Diviseur de fréquence.....	27
II.6. Fonctions de transfert... ..	27
II.7. Elaboration et évaluation des performances du filtre de boucle.....	28
II.8. Bruit de phase dans les synthétiseurs de fréquences.....	31
II.9. Présentation générale de logiciel ADIsimPLL.....	32
II.10. La simulation d'un synthétiseur de fréquence.....	33
II.11. Calcul du filtre.....	34
II.12. Les réponses fréquentielles de la boucle.....	35
II.12.1. Gain et phase en boucle ouverte	35

II.12.2. Gain et phase en boucle fermé.....	36
II.12.3. Le bruit de phase à la sortie du synthétiseur de fréquence.....	37
II.12.4. La réponse de la modulation de fréquence	37
II.12.5. Leakagespurs (raies de référence)	38
II.13. La réponse temporelle	38
II.13.1. Comparateur de phase-fréquence.....	39
II.13.2. Erreur de fréquence	40
II.14. Conclusion	40

Chapitre III : Implémentation de la PLL sur FPGA

III.1. Introduction.....	41
III.2. Le circuit FPGA.....	41
III.2.1 Les Les différents types de FPGA.....	43
III.2.1.1. Les FPGA à mémoire statique (SRAM).....	43
III.2.1.2. Les FPGA à mémoire Flash.....	43
III.2.1.3. Les FPGA à anti-fusibles	43
III.2.2. Applications de FPGA.....	43
III.2.2.1. Les domaines d’application des FPGA.....	44
III.3. Méthodes de conception.....	44
III.3.1. La conception des circuits à faibles densités.....	44
III.3.2. La conception des circuits à hautes densités.....	44
III.3.2.1. Simulation et synthèse.....	46
III.3.2.2. Programmation du FPGA.....	46
III.4. Les outils de développement.....	47
III.4.1. Les outils de CAO (Conception Assistée par Ordinateur).....	47

III.4.1.1. Les outils de synthèse et simulation.....	48
III.4.1.2. Les outils de placement et routage.....	48
III.4.1.3. L’outil ISE 14.7.....	48
III.4.2. Les langages de description.....	49
III.4.2.1. Le langage VHDL.....	49
III.5. L’architecture matérielle de la BVPN	50
III.5.1. Comparateur de phase (la porte xor).....	51
III.5.2. Filtre de boucle numérique	51
III.5.3. Oscillateur de commande numérique	53
III.6. L’implémentation matérielle de la BVPN.....	56
III.6.1. Présentation de l'environnement	56
a. Le logiciel de développement.....	57
b. La carte de développement mimas v2 de Xilinx.....	57
III.6.2. La réalisation physique de la BVPN	59
III.7. Conclusion	61
Conclusion général.....	62
Annexes.....	64
Référence.....	75
Résumé	

LISTE DES FIGURES

CHAPITRE I

Figure I.1. Schéma bloc de la PLL.....	6
Figure I.2. Schéma bloc d'une PLL analogique.....	7
Figure I.3. Comparateur de phase analogique.....	8
Figure I.4. Filtre RC passif et sa fonction de transfert.....	9
Figure I.5. Filtre RC actif et sa fonction transfert.....	9
Figure I.6. Circuit oscillant.....	10
Figure I.7. La caractéristique $f_s=f(\text{VCO})$	11
Figure I.8. Plage de capture et plage de verrouillage.....	12
Figure I.9. Modèle linéaire du bruit de phase d'une PLL avec chaque contributeur.....	13
Figure I.10. Bruit de phase typique d'une PLL.....	14
Figure I.11. Schéma bloc d'asservissement e phase de la PLL.....	16
Figure I.12. Filtre de boucle compteur K.....	18
Figure I.13. Diviser par compteur N.....	19

CHAPITRE II

Figure II.1. Plage de fréquences et pas de synthèse.....	23
Figure II.2. Schéma-bloc d'une PLL à division entière.....	24
Figure II.3. Circuit d'un comparateur numérique de phase-fréquence.....	25
Figure II.4. Diagrammes des différents signaux.....	25
Figure II.5. Circuit de la pompe de charge.....	26
Figure II.6. Fonctionnement de la pompe de charge.....	26
Figure II.7. Modèle linéaire de la PLL.....	27
Figure II.8. Schéma de principe d'un système asservi.....	27
Figure II.9. Tracé de Bode de la réponse de boucle ouverte.....	28
Figure II.10. Filtre passif de second ordre.....	29

Figure II.11. Filtre de boucle de troisième ordre.	30
Figure II.12. L'interface graphique du logiciel ADIsimPLL.	33
Figure II.13. Schéma fonctionnel de l'ADF 4107.....	34
Figure II.14. Schéma de l'ADF4106 pour ADIsimPLL.	35
Figure II.15. La réponse de la boucle ouverte.	36
Figure II.16. La réponse de la boucle fermée.	36
Figure II.17. Le bruit de phase de chaque composant dans le PLL.	37
Figure II.18. Réponse de la modulation de fréquence.	37
Figure II.19. Les raies latérales de référence.	38
Figure II.20. La réponse temporelle de la PLL.	39
Figure II.21. Sortie de la pompe de charge.	39
Figure II.22. L'erreur de fréquence de la PLL.	40

Chapitre III

Figure III.1. Structure interne d'un FPGA.....	42
Figure III.2. Schéma bloc d'une cellule dans un bloc CL.....	42
Figure III.3. Les interconnexions entre les blocs d'un FPGA.....	42
Figure III.4. L'utilité d'un outil de développement	47
Figure III.5. L'outil de synthèse	48
Figure III.6. Les outils de placement et de routage.....	48
Figure III.7. Le bloc de BVPN.....	50
Figure III.8. Schéma structurel du BVPN.....	50
Figure III.9. La porte XOR.....	51
Figure III.10. Simulation de la porte XOR.....	51
Figure III.11. Schéma bloc de filtre de la boucle	51
Figure III.12. Structure interne de FBN	52
Figure III.13. Simulation de la sortie de filtre de boucle.....	52
Figure III.14. Schéma bloc de l'OCN.....	53
Figure III.15. Structure interne de l'OCN.	53
Figure III.16. La simulation de l'OCN.....	54

Figure III.17. La structure interne de la BVPN.....	54
Figure III.18. La simulation fonctionnelle de la BVPN.	55
Figure III.19 (a) et (b). La détection des points de verrouillage de la BVPN.....	55
Figure III.22. L’environnement de travail.	57
Figure III.23. Environnement de développement ISE 14.7.....	57
Figure III.24. La carte de développement MIMAS V2.....	58
Figure III.25. Câble USB 2.0.....	59
Figure III.26. L’architecture matérielle globale de la BVPN.....	60
Figure III.27. La simulation de la BVPN avec le diviseur de fréquence.....	60
Figure III.26. La visualisation des signaux s_ref et s_sortie de la BVPN sur l’oscilloscope.....	61

LISTE DES TABLEAUX

Tableau II.1. Valeurs des éléments du filtre de boucle du troisième ordre.....	35
Tableau III.1. Les principaux langages de description.....	49

LISTE DES ORGANIGRAMMES

Organigramme III.1. Conception des circuits à haute densité.....	45
---	----

SIGLES ET ABRÉVIATIONS

A

AC	Circuit Analogique
ACTEL	Agence Commercial de Télécommunication
ADPLL	All Digital Phase Locked Loop
ASIC	Application-Specific Integrated Circuit

B

BVFN	Boucle à Verrouillage de Phase Numérique
------	--

C

CAO	Conception Assistée par Ordinateur
CLB	Configurable Logic Block
CPLD	Complex Programmable Logic Device

D

DC	Circuit Discret
DPLL	Digital Phase Locked Loop
DSP	Digital Signal Processor

E

EEPROM	Electrically Erasable Programmable Read Only Memory
EDA	Electronic Design Automation

F

FBN	Filtre de Boucle Numérique
FPGA	Filed Programmable Gate Array

G

GBF	Générateur de Basse Fréquence
-----	-------------------------------

H

HF	Haut Fréquence
----	----------------

I

IEEE	Institute of Electrical and Electronics Engineers
------	---

L

- LC une bobine (L) et un condensateur (Capacité)
LCA Logic Cell Array
LTE Long Term Evolution

O

- OCN oscillateur à commande numérique

P

- PFD phase frequency detector
PLD Programmable Logic Device
PLICETM Programmable Low Impedance Circuit Element
PLL phase locked loop

R

- RC résistance , capacité

S

- SPLD Simple **P**rogrammable **L**ogic **D**evice
SPLL Software **P**hase **L**ocked **L**oop
SRAM Static **R**andom Access **M**emory

T

- TCXO Temperature **C**ompensated **C**rystal **O**scillator

U

- USB **U**niversal **S**erial **B**us

V

- VCO Voltage **C**ontrolled **O**scillator
VGA **V**ideo **G**raphics **A**dapter
VHDL Very **H**igh speed integrated circuits **H**ardware **D**escription **L**anguage
VLSI Very **L**arge **S**cale **I**ntegration

INTRODUCTION
GENERALE

INTRODUCTION GÉNÉRALE

Henri De Belliscize publia en 1932, un important article intitulé : « la réception synchrone », c'est la naissance de la boucle à verrouillage de phase ou PLL (*Phase Locked Loop*). La PLL est un asservissement qui a pour but à maintenir, en phase deux oscillations distincts en générale, l'un des deux est l'oscillateur de référence. La seconde est alors asservie au premier qui impose une référence.

La mise en œuvre de ce principe était délicate compte tenu des moyens de l'époque, ce qui explique que sa généralisation ait dû attendre les progrès de la technologie des circuit intégrés [1]. En effet c'est qu'en 1965 que les premiers PLL sur circuit intégré ont vu le jour. Depuis, l'utilisation des boucles à verrouillage de phase s'est étendue de façon considérable dans tous les domaines de télécommunications, de mesures, etc...

Les PLLs ont permis des progrès considérables pour le traitement des signaux dans le domaine fréquentiel. La mise au point de la boucle à verrouillage de phase ou PLL (*Phase Locked Loop*) est directement liée à l'histoire de la modulation d'amplitude. Ce dispositif était destiné à améliorer les conditions de réception des signaux radioélectriques noyés dans le bruit en modulation d'amplitude. Les PLL ont envahi tout le domaine de télécommunication. Depuis l'invention de la PLL et ce terme est encours évolue avec la technologie, de la réalisation volumineuse et chère à des circuits intégrés aujourd'hui.

L'enjeu de ce travail est l'analyse, la synthèse et la conception d'un prototype de PLL purement numérique implémenté sur une carte de développement « Mimas V2 » qui contient le circuit programmable FPGA de type Spartan 6 de la société Xilinx.

Dans le premier chapitre, nous allons présenter la théorie de la PLL analogique, son principe de fonctionnement et les blocs constituant ce circuit, suivi d'un résumé d'étude sur le bruit de phase qui influe sur l'erreur et la stabilité la PLL. Ensuite, nous allons étudier la PLL absolument numérique en montrant des exemples.

Dans le deuxième chapitre, nous allons exploiter un logiciel spécialisé pour la simulation et la conception des synthétiseurs de fréquences « ADIsimPLL », afin d'étudier la PLL analogique dans le domaine temporel, fréquentiel, et l'étude du bruit.

Enfin, dans le dernier chapitre, nous allons réaliser une architecture matérielle de la PLL numérique, en utilisant le langage de description matérielle VHDL. Ensuite, nous allons implémenter la PLL numérique sur un circuit programmable FPGA, en réalisant le premier prototype de PLL numérique.

INTRODUCTION GÉNÉRALE

Et finalement une conclusion générale qui résume les résultats de ce travail, en présentant des perspectives.

CHAPITRE I :
LA THEORIE DE
LA
PLL

I.1. Introduction

En 1932, Henri de Bellescize a inventé un montage technique fondamental de l'électronique moderne, la boucle à verrouillage de phase (Phase-Locked Loops ou PLL), pour assurer la détection asynchrone. Seulement la mise en œuvre de ce principe était délicate à cause des moyens disponibles à l'époque. Il a fallu attendre plusieurs années pour que la PLL soit couramment utilisée. En effet, après l'apparition des circuits intégrés, cette invention prit une place majeure dans les systèmes électroniques modernes, tels que les télécommunications, le contrôle, l'automatisme, les mesures...

Les PLLs sont utilisés de principalement soit comme un filtre très étroit, par exemple pour récupérer un signal utile noyé dans du bruit ; soit comme multiplieur de fréquence, c'est le cas de certains synthétiseurs de fréquences. Dans ce chapitre, il est décrit l'essentiel de la théorie des PLLs, la description du comportement des différents blocs des différentes conceptions des PLLs. On va donner une intention particulière pour les PLL numériques.

I.2. Les différents types de PLLs

Selon la technologie choisie et les composants utilisés on trouve plusieurs types de PLL qu'on résume ci-dessous :

- PLL analogique dont tous les circuits des différents blocs sont de type analogique
- PLL semi-numérique est une boucle à verrouillage de phase dont certaines parties sont numériques alors que d'autres sont analogiques.
- PLL purement numérique, elle est exclusivement construite à partir d'un bloc dont les fonctions sont numériques et ne contiennent pas des composants passifs tels que les résistances et les condensateurs, pour les filtres.
- SPLL : programme embarqué qui gère la PLL est normalement mis en œuvre par une plate-forme matérielle telle qu'un microcontrôleur ou un processeur de signal numérique (tel que le DSP)[1].

I.3. Définition et principe de fonctionnement

La PLL est à base d'un montage électronique permettant d'asservir la phase ou la fréquence de sortie d'un système sur la phase ou la fréquence du signal d'entrée. Elle peut aussi asservir une fréquence de sortie sur un multiple de la fréquence d'entrée. Plus précisément, la PLL est un circuit synchronisant un signal de sortie (généré par un oscillateur) avec un signal de référence ou d'entrée aussi bien en fréquence qu'en phase. Le principe de fonctionnement de la PLL est expliqué par l'exemple de la PLL linéaire (PLL). Son schéma bloc est illustré à la figure (I.1).

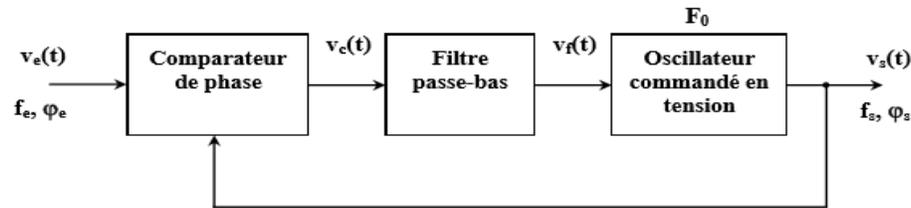


Figure I.1. Schéma bloc de la PLL[2].

Une PLL constitue de 3 composants de base :

1. Un comparateur de phase.
2. Un filtre passe-bas.
3. Un oscillateur commandé en tension (VCO).

Et d'autres composants selon les applications différentes de la PLL, il existe : Les divisions de fréquence, la pompe de charge, des blocs de gain, des convertisseurs de signal de la forme sinusoïdal à la forme carrée,... etc.

On parle de boucle à verrouillage de phase avec le but est d'obtenir des fréquences identiques sur les deux accès du comparateur de phase car dans tout asservissement il existe, à l'équilibre, une différence non nulle entre la grandeur appliquée à l'entrée et celle délivrée en sortie. Ainsi si on réalise une comparaison des fréquences des signaux d'entrée et de sortie, au lieu d'effectuer une comparaison des phases, il subsisterait, à l'équilibre, une différence de fréquence non nulle.

On préfère donc réaliser un asservissement portant sur des grandeurs qui sont les intégrales des fréquences des signaux considérés, c'est-à-dire les phases des signaux, à un coefficient multiplicatif 2π près. A l'équilibre, mais les pulsations sont identiques[3]. Dans ce travail, on s'intéresse à deux types de PLL, les PLL analogiques et les PLL numériques. En effet on implémente une PLL numérique sur carte de développement « MIMAS V2 », et dont l'étude des différentes réponses se font en analogique par le logiciel ADIsimPLL.

I.4. PLL analogique

Une PLL analogique est une boucle à verrouillage de phase dont l'ensemble des éléments internes du dispositif sont des blocs analogiques. Ce genre de PLL généralement coûteux, est réalisé en surface de silicium importante car le filtre et l'oscillateur contrôlé en tension VCO (à base d'inductance) prennent une place importante dans la puce électronique[5]. La boucle permet d'asservir la fréquence d'un VCO à la fréquence d'un signal de référence dans une certaine plage autour de la fréquence centrale F_0 , l'oscillateur délivrant un signal de la forme [4]:

$$u_{osc} = \hat{U}_{osc}(\cos(\omega_{osc})t + \varphi_{osc}) \quad (I.1)$$

I.4.1. Eléments constitutifs de la PLL analogique

Les blocs constituant la PLL analogique sont représentés sur la figure I.2 ci-dessous :

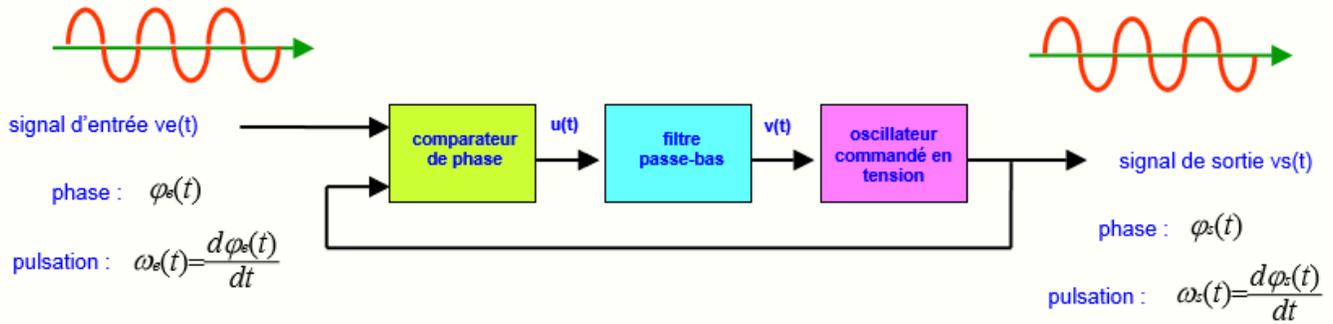


Figure I .2. Schéma bloc d'une PLL analogique[6].

Un oscillateur commandé en tension (VCO) délivre une fréquence \$f_s\$ dont la valeur dépend de la tension appliquée \$v(t)\$ sur son entrée de commande. Cette fréquence est comparée à la fréquence de référence \$f_e\$, en passant par l'intermédiaire des phases instantanées des signaux \$\varphi_e(t)\$ et \$\varphi_s(t)\$. Le comparateur de phase donne en temps réel d'écart (ou d'erreur) de phase entre la source de référence et le VCO. (Le comparateur à la sortie donne deux harmoniques \$f_e + f_s\$ et \$f_e - f_s\$, où l'harmonie \$f_e + f_s\$ va être filtrée, d'où l'importance du choix du filtre qui doit être optimum. Donc, il fournit à sa sortie une tension \$u(t)\$ alternative, dont la valeur moyenne \$v(t)\$ donnée par un filtre passe-bas qui élimine les harmoniques "2. \$f_e\$" ou \$f_e + f_s\$, et cette tension est proportionnelle au déphasage entre \$v_e(t)\$ et \$v_s(t)\$ qui sont les tensions d'entrée et de sortie respectivement, et le comparateur est caractérisé par un coefficient souvent noté \$K_D\$ défini par :

$$K_D = \frac{\text{valeur moyenne de la tension de sortie}}{\text{déphasage entre les signaux d'entrée}} = \frac{U_{\text{moyen}}}{\phi} \text{ en V/radian} \tag{I.2}$$

Ainsi, la fréquence \$f_s\$ est en permanence corrigée pour rester égale à celle de la source. Il est linéarisé autour de \$F_0\$ et caractérisé par sa pente \$K\$[6]:

$$K = \frac{\text{variation de la pulsation du signal de sortie}}{\text{variation de la tension de commande}} \text{ En radian/s.V} \tag{I.3}$$

a. Comparateur de phase

Un détecteur de phase est un circuit capable de délivrer un signal de sortie proportionnel à la différence de phase entre ses deux signaux d'entrée \$U_1\$ et \$U_2\$, de nombreux circuits pourraient être appliqués à l'entrée. Quand il s'agit des PLLs à signaux mixés, le premier détecteur de phase de l'histoire de la PLL était le détecteur de courant linéaire. Lorsque la PLL est entrée sur le territoire numérique, les détecteurs de phase

numériques sont devenus populaires, tels que la porte XOR, la bascule JK, parmi les types de détecteur de phase linéaire le multiplicateur, pour les détecteurs de phase analogique, ce sont les détecteurs les plus utilisés. Ils sont utilisés lorsque le signal d'entrée est sinusoïdal, et en particulier en présence de bruit[4].

Parmi ces types on a :

- **Le comparateur de phase à mélangeur**

Aux fréquences très élevées, on utilise comme comparateur de phase un multiplieur (mélangeur Schottky) suivi d'un filtre passe-bas[6]

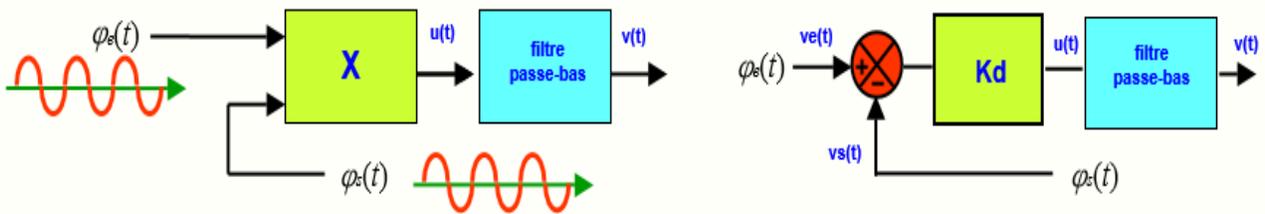


Figure I.3. Comparateur de phase analogique[6].

La tension en sortie du mélangeur s'écrit :

$$\begin{aligned}
 u(t) &= K_D \cdot v_e(t)v_s(t) = K_D \cdot V_e \sin(\omega_0 t + \varphi_e(t)) \cdot V_s \sin(\omega_0 t + \varphi_s(t)) & (I.4) \\
 &= 0,5 \cdot K_D \cdot V_e \cdot V_s \cdot \sin(2\omega_0 t + \varphi_e(t) + \varphi_s(t)) + 0,5 \cdot K_D \cdot V_e \cdot V_s \cdot \sin(\varphi_e(t) - \varphi_s(t))
 \end{aligned}$$

Le filtre passe-bas conserve la partie basse du mélange

$$u(t) = 0,5 \cdot K_D \cdot V_e \cdot V_s \cdot \sin(\varphi_e(t) - \varphi_s(t)) \tag{I.5}$$

b. Filtre de boucle

Le signal de sortie $U_d(t)$ du détecteur de phase est composé d'un certain nombre de termes ; dans l'état verrouillé de la PLL, le premier de ceux-ci est un composant "DC"(composante disctert) et plus ou moins dont l'intensité est proportionnelle à l'erreur de phase ; les termes restants sont des composants "AC"(composante alternative) ayant des fréquences de $2 \cdot f_e, 4f_e, \dots$

Étant donné que les fréquences les plus élevées sont des signaux indésirables, elles sont filtrées par le filtre de boucle. Parce que le filtre de boucle doit supprimer les hautes fréquences, donc il doit s'agir d'un filtre passe-bas, où f_c est la fréquence de coupure du filtre passe bas.

Dans la plupart des conceptions des PLLs, un filtre passe-bas de premier ordre est utilisé. Selon l'application, ce filtre est soit passif ou actif[1].

➤ **Type01 : Filtre passif**

Est un filtre passif, dont le schéma est représenté à la figure I .4 avec un pôle et un zéro, sa fonction de transfert $F(s)$ est donnée par :

$$F(s) = \frac{1+s\tau_2}{1+s(\tau_1+\tau_2)} \tag{I.6}$$

Avec $\tau_1 = R_1C$ et $\tau_2 = R_2C$

τ_1 Caractérise la pulsation de coupure du filtre[4].

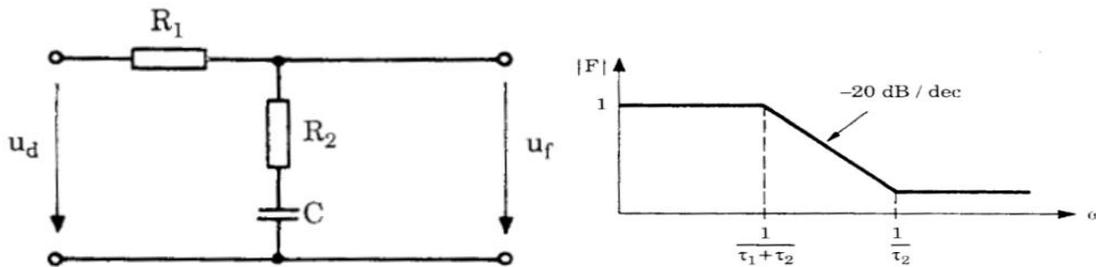


Figure I .4.Filtre RC passif et sa fonction de transfert[4].

➤ **Type 02 : filtre actif**

La plupart du temps est caractérisé à base d'amplificateur opérationnel sa fonction de transfert est très similaire à la fonction passive mais comporte un terme de gain K_0 supplémentaire, que l'on peut choisir supérieur à 1. Sa fonction de transfert $F(s)$ est donnée par [4]:

$$F(s) = K_0 \frac{1+s\tau_2}{1+s\tau_1} \tag{I.7}$$

Avec $\tau_1 = R_1C$ et $\tau_2 = R_2C$, $K_0 = C_1/C_2$

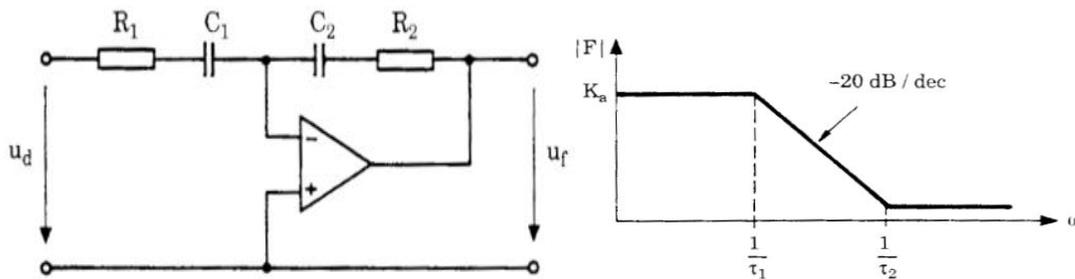


Figure I .5. Filtre RC actif et sa fonction transfert[4].

c. **VCO**

L'oscillateur est un circuit qui délivre un signal périodique de fréquence propre F_0 en absence du signal d'entrée. Le VCO a pour fonction, commeson nom l'indique, de transformer une tension continue appliquée sur l'entrée en un signal modulé en fréquence en sortie. Le VCO est utilisé dans les systèmes de communications, il a plusieurs applications telles que la génération d'une fréquence de référence ou encore la

modulation du signal à émettre. Les VCOs sont peu stables en fréquence, ils ne peuvent être utilisés qu'en boucle fermée. Une variation de tension, lente ou rapide, à leur entrée se traduit par une variation de fréquence en sortie[1]. En appliquant une tension d'entrée $V_e(t)$ (la sortie du filtre passe bas), la fréquence de sortie varie proportionnellement à cette tension, suivant l'expression suivante[4] :

$$f_{VCO} = f_0 + K_{vco} \cdot V_e \tag{I.8}$$

Où K_{vco} est la sensibilité du VCO exprimée en Hz/V, parmi les types d'oscillateurs on a :

• **Oscillateur de type LC (sinusoïdal)**

Un oscillateur à circuit accordé LC est un circuit bouclé auto-oscillant travaillant suivant la condition de Barkhausen. On fait varier la fréquence d'oscillation en agissant sur l'un des deux éléments du circuit bouchon L ou C. Pour commander l'oscillateur avec un signal électrique, on utilise généralement une diode Varicap connectée en parallèle (du point de vue des petits signaux) avec un circuit LC et polarisé en inverse par une tension de commande. Cette tension est généralement appliquée aux bornes de la Varicap par le biais d'une résistance de très grande valeur ou une inductance (de très grande impédance en HF), La connexion de la diode en parallèle avec le circuit LC est réalisée par une capacité de liaison (impédance négligeable en HF et infinie en DC, permettant de séparer les niveaux DC aux bornes de la Varicap et du circuit bouchon). Cette capacité de liaison est souvent (avantageusement) remplacée par une seconde diode Varicap, dont l'impédance en HF n'est pas négligeable, mais également variable avec la tension de commande[4].

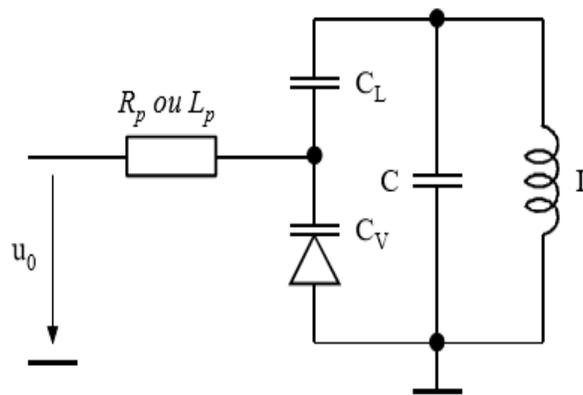


Figure I.6.Circuit oscillant[4].

d. Diviseur

Le diviseur de fréquence permet de diviser la fréquence du signal de sortie du VCO afin de le comparer à la fréquence de référence de la PLL, plus basse. Cette utilisation du diviseur de fréquence sur la

boucle de retour de la PLL réalise la multiplication de fréquence du signal de référence. Les diviseurs de fréquence sont en principe programmables (programmation du rapport de division), ce qui permet de faire varier la fréquence de sortie de la PLL. Ils peuvent être à rapport de division entier ou fractionnaire. Ce circuit est très utilisé dans les synthétiseurs de fréquence à base de PLL[3].

I.4.2. Fonctionnement d'une PLL à fréquence fixe

Le VCO ait la caractéristique tension-fréquence suivante $f_s=f(\text{VCO})$, comme il est caractérisé à la figure I.7, on remarque une plage de fréquence de fonctionnement linéaire du VCO ; où :

F_0 : est appelée la fréquence libre du VCO

K_{vco} : est le gain du VCO

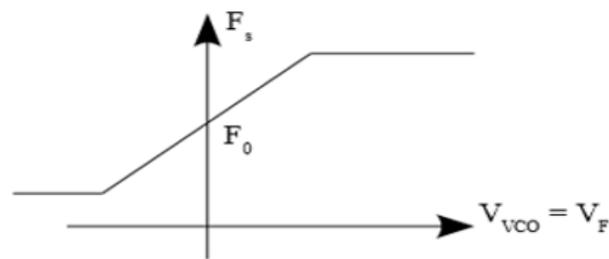


Figure I.7. La caractéristique $f_s=f(\text{VCO})$ [2].

I.4.2.1. PLL décrochée (non verrouillée)

En l'absence de signal injecté à l'entrée de la boucle, ou si la fréquence du signal injecté est en dehors de la plage de fonctionnement du VCO, la boucle est dite non verrouillée et $f_s = F_0$.

I.4.2.2. PLL accrochée (verrouillée)

La plage d'accord est la plage de fréquence que la PLL doit couvrir, elle dépend de l'application

- Si on injecte dans la boucle un signal de fréquence f_e voisin de F_0 , la PLL se verrouille et on aboutit au bout d'un temps bref (1 à 100 ms en général) à un état stable caractérisé par $f_s = f_e$.
- Si la fréquence d'entrée sort de la plage de verrouillage, la boucle décroche et on revient à la situation d'une boucle non verrouillée.
- Pour raccrocher la boucle, il faut alors revenir au voisinage de F_0 et pénétrer dans la plage de capture.

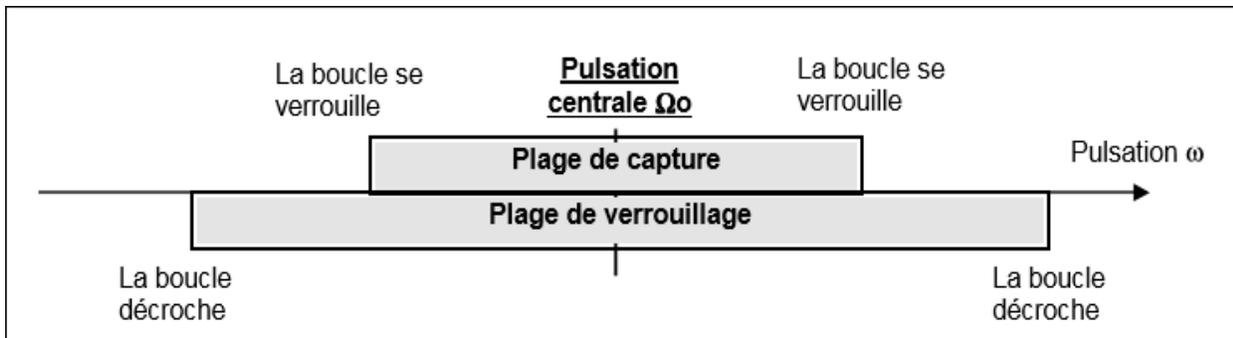


Figure I.8.Plage de capture et plage de verrouillage[2].

I.4.2.3. Plage de capture et plage de maintien

Comme il est montré à la figure I.8, la plage **de capture** est l'intervalle de fréquence pour lequel la PLL est verrouillée mais dont les limites sont fixées par les deux fréquences conduisant à la capture (état non verrouillé vers un état verrouillé). Lorsque la PLL est verrouillée, la fréquence d'entrée peut varier, la fréquence de sortie va suivre. Si la variation est trop grande, la PLL va décrocher. La plage **de maintien** est donc l'intervalle de fréquence pour lequel la PLL est verrouillée mais dont les limites sont fixées par les deux fréquences conduisant au décrochage de la PLL. Systématiquement, la plage de maintien est supérieure ou égale à la plage de capture[2].

I.4.3. La PLL en régime dynamique : fréquence variable

Équation général

$$\varphi(t) = 2. \pi. \int f(t)dt \quad (I.9)$$

I.4.3.1. Point de repos de la PLL

Comme tout asservissement, la PLL doit être modélisée autour d'un point de repos qu'il convient de définir, la PLL est verrouillée (sur sa fréquence centrale f_0), la fréquence d'entrée est constante. La phase croît linéairement (tout comme en asservissement de vitesse, la position croît linéairement à vitesse constante)[2].

- En entrée

$$f_e(t) = f_{e0} = F_0 \quad (I.10)$$

$$\varphi_e(t) = 2. \pi. F_0 t + \varphi_{e0} \quad (I.11)$$

- En sortie

$$f_s(t) = f_{s0} = f_{e0} = F_0 \quad (I.12)$$

$$\varphi_s(t) = 2. \pi. F_0. t + \varphi_{s0} \quad (I.13)$$

I.5. Etude de bruit de phase

L'étude de bruit de phase qui est une technique commune à toutes les PLL, est très importants, car elle permet de connaitre quelles sont les parties de la PLL qui sont les contributeurs majoritaires du bruit total en sortie, Les bruits intrinsèques des blocs suivants sur la figure I.9, influent sur le bruit total de la PLL :

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

À titre de simplification, on va supposer que toutes les sources de bruit sont constantes avec la fréquence (bruit blanc). Si on reprend le schéma-bloc de la PLL (figure I.9) en négligeant le bruit du filtre de boucle, on obtient le schéma de la PLL avec ses sources de bruit, présenté à la figure suivante :

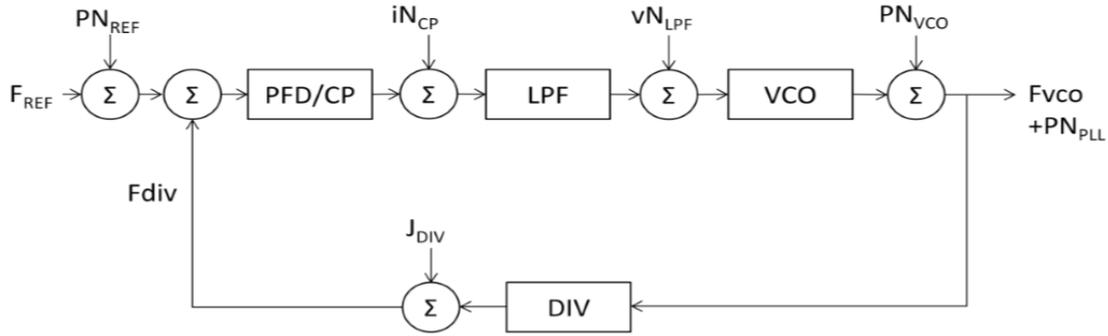


Figure I.9. Modèle linéaire du bruit de phase d'une PLL avec chaque contributeur[3].

Les bruits des diviseurs de fréquences ($\Phi_N Div$) et du VCO ($\Phi_N VCO$) sont modélisés par des bruits de phase. Par contre, le bruit de l'ensemble comparateur de phase - pompe de charges est modélisé par une source de courant (I_N). Le bruit du filtre de boucle est ici négligé, car le filtre est généralement passif et le bruit généré par ses résistances est souvent négligeable. Pour simplifier les calculs, on va déterminer le bruit de phase de sortie de la PLL dans les cas où la pulsation tend vers 0 et vers l'infini.

Lorsque $\omega \rightarrow 0$, le gain de la boucle ouverte est infini, les sources de bruit après le filtre de boucle sont donc masquées. Par contre la PLL recopie fidèlement les bruits présents sur le comparateur de phase. On trouve donc :

$$\varphi_{N\text{ sortie}}(\omega \rightarrow 0) = n \left(\varphi_{N\text{ référence}} + \varphi_{N\text{ div}} + \frac{I_N}{K_D} \right) \text{ en rad} \tag{I.14}$$

A l'inverse, lorsque $\omega \rightarrow \infty$, le gain de boucle est égal à zéro. Les bruits présents sur le comparateur de phase sont filtrés. Seul le bruit du VCO est présent en sortie :

$$\varphi_{N\text{ sortie}}(\rightarrow \infty) = \varphi_{N\text{ VCO}}(\omega) \text{ en rad} \tag{I.15}$$

Ce rapide calcul permet de remarquer qu'en basse fréquence le bruit vient essentiellement de la référence, du comparateur de phase et du diviseur. Par contre en haute fréquence, le bruit du VCO est prépondérant. Les deux asymptotes du bruit de phase sont représentées à la figure I.10, ainsi que la forme typique du bruit de phase d'une PLL. On peut voir, que si la bande passante de la PLL est inférieure à l'abscisse du point de croisement des asymptotes, on a une remontée du bruit de phase. Dans le cas contraire, cette remontée n'a pas lieu. En observant cette courbe, on peut dire que pour minimiser le bruit de phase total, la bande passante de la PLL doit être approximativement égale au point de croisement des deux asymptotes[3].

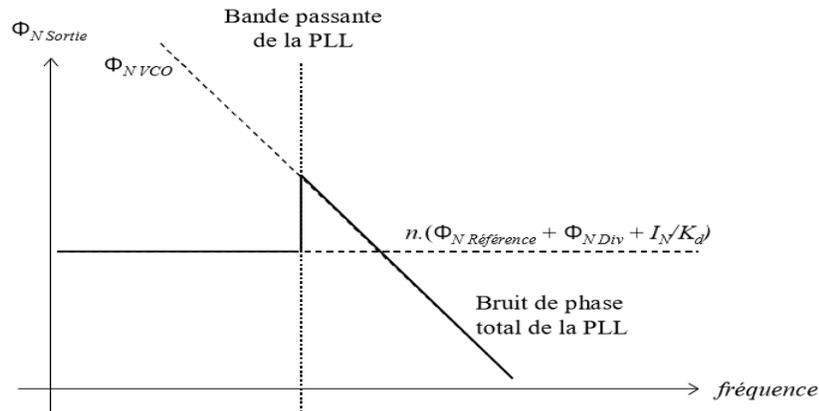


Figure I.10. Bruit de phase typique d'une PLL[3].

I.6. Fonctions de transfert général de la PLL

Pour pouvoir simuler les systèmes à base de PLL il est impératif de déterminer les réponses fréquentielles de la chaîne PLL.

I.6.1. L'ordre du système du PLL

La plupart des PLL considérées jusqu'à présent étaient des PLL de second ordre. Le filtre de boucle a un pôle, et le VCO avait un pôle, de sorte que tout le système avait deux pôles.

Généralement, l'ordre d'une PLL est toujours supérieur à l'ordre du filtre de boucle, ou le filtres de boucle d'ordre 2 est le plus utilisés dans des applications critiques. Alors l'ordre de la PLL est égal au nombre de pôles de la fonction de transfert qui est égal à l'ordre du filtre + 1. A titre d'exemple, pour un filtre d'ordre 1, la PLL est d'ordre 2. Pour un filtre d'ordre 2, la PLL est d'ordre 3[1].

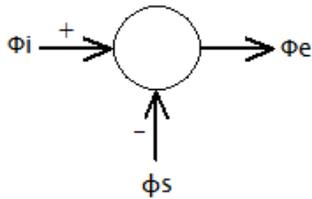
I.6.2. Modélisation de la PLL

La modélisation de la PLL permet de définir les paramètres de chaque bloc, décrit précédemment, nécessaires à la convergence (et donc la stabilité) du système. Lorsque la PLL est verrouillée, le système est considéré comme linéaire pour de faibles variations de fréquence autour de cette position d'équilibre où elle peut être modélisée dans le domaine de Laplace. Toutefois, il est intéressant de noter que cette modélisation

est une approximation qui n'est valable que pour une valeur de BP de la PLL (définie plus bas) faible devant la fréquence de référence f_{ref} et que la modélisation se fait de manière plus rigoureuse, pour le couple PFD/CP. Ceci est lié au fonctionnement discret du couple PFD/CP qui génère des impulsions en courant à la fréquence de référence. Pour assurer la validité du modèle de Laplace ainsi qu'une première condition sur la stabilité du système, la limite de la bande passante est définie par[4] :

$$BP \leq \frac{f_{ref}}{10} \quad (I.16)$$

a. Le détecteur de phase

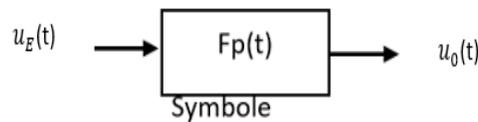


$$U_e(t) = K_D \cdot [\varphi_i(t) - \varphi_s(t)] = K_D \cdot \varphi_e(t) \quad (I.17)$$

Avec φ_e est la phase de signal de sortie de comparateur de phase, et U_e est la tension de sortie de comparateur[4].

b. Le filtre de boucle

Le filtre de boucle de type quelconque, peut être représenté par le schéma bloc de la figure suivante :



La fonction transfert $F_p(t)$ caractérisé d'une façon générale n'importe quel type de filtre. On peut donc écrire la relation comportementale unique : $u_0(t) = F_p(t) \cdot u_E(t)$ [4].

c. Le VCO

$\Omega_{osc} = 2\pi \cdot F_{osc}$: est la pulsation libre de l'oscillateur, pour la quelle $u_0(t) = 0$.

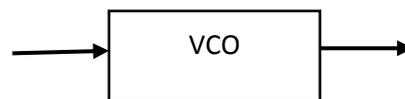
$\Delta\omega_{osc} = K_0 \cdot u_0(t)$ est l'écart de fréquence par rapport à $\Omega_{osc}(t)$.

K_{vco} : est la sensibilité s'exprime en [rad/Vs].

$$\omega_{osc}(t) = \Omega_{osc} + \Delta\omega_{osc}(t)$$

$u_0(t)$

$\omega_{osc}(t)$



Symbole

Relation phase fréquence du signal VCO

La phase d'un signal sinusoïdal est l'intégral par rapport au temps de sa fréquence :

$$\varphi_{osc}(t) = \int_0^t \omega_{osc}(t)dt = \Omega_{osc} \cdot t + \int_0^t \Delta\omega_{osc}(t)dt = \Omega_{osc} \cdot t + \Delta\varphi(t) \quad (I.18)$$

La phase est donc définie par rapport à celle de la fréquence libre avec [4]:

$$\Delta\omega_{osc}(t) = \int_0^t \Delta\varphi(t)dt \quad (I.19)$$

d. Diviseur de fréquence

Le diviseur de fréquence est constitué d'un compteur par N programmable[4].



Symbole

I.6.3. Fonction transfert de la boucle fermée

Lors du fonctionnement en mode verrouillage de la boucle, les corrections qui sont apportées au VCO ont des amplitudes très faibles. Ceci signifie que les variations des grandeurs autour des points de fonctionnement des différents éléments constitutifs de la boucle ont des amplitudes suffisamment faibles pour que l'on puisse admettre que le fonctionnement de l'asservissement est linéaire. Le schéma bloc de l'asservissement de phase peut être représenté par le schéma suivant [4]:

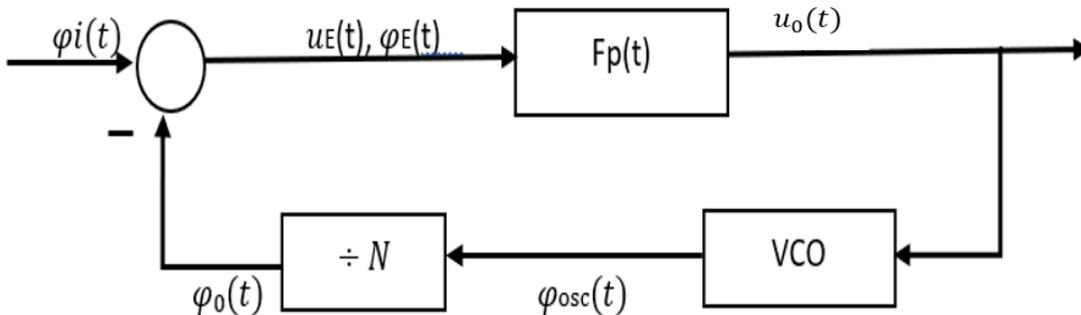


Figure I.11. Schéma bloc d'asservissement e phase de la PLL[4].

La grandeur de sortie est donnée par la relation avec la transformée de Laplace :

$$u_o(s) = K_D \cdot F_p(s) \left(\varphi_i(s) - \frac{K_0}{N} \cdot \frac{1}{s} \cdot u_o(s) \right) \quad (I.20)$$

Qui permet de définir la fonction de transfert du system de la boucle fermée :

$$H(s) = \frac{u_o(s)}{\varphi_i(s)} = \frac{K_D \cdot F_p(s)}{1 + K_D \cdot F_p(s) \frac{K_0}{N} \cdot \frac{1}{s}} = \frac{K_D \cdot F_p(s) \cdot s}{s + K_D \cdot F_p(s) \frac{K_0}{N}} \quad (I.21)$$

C'est la fonction caractéristique d'un système bouclé à réaction négative[4].

I.6.4. Fonction dérivée de la fonction transfert

Sachant que

$$u_0(s) = K_D \cdot F_p(s) \varphi E(s) \quad (I.22)$$

Il vient la fonction d'erreur

$$\frac{\varphi E(s)}{\varphi i(s)} = \frac{s}{s + K_D \cdot F_p(s) \frac{K_0}{N}} \quad (I.23)$$

D'autre part

$$\varphi i(s) = \frac{\omega_i(s)}{s} \quad (I.24)$$

On déduit la fonction de transfert entre la fréquence d'entrée et la tension de sortie :

$$\frac{u_0(s)}{\omega_i(s)} = \frac{K_D \cdot F_p(s)}{s + K_D \cdot F_p(s) \frac{K_0}{N}} \quad (I.25)$$

L'étude de stabilité peut se faire sur n'importe laquelle de ces fonctions de transfert puisque le dénominateur est identique pour chacune d'elles [4].

I.7. PLL numérique (digital)

Le PLL numérique classique est un circuit semi-analogique, parce qu'il a toujours besoin de deux composants externes. Pire encore, la fréquence centrale d'une PLL numérique est influencée par des condensateurs parasites sur la puce PLL numérique, à l'ancienne version PLL numérique, il s'agit d'un système entièrement numérique. Numérique signifie que le système est constitué exclusivement de dispositifs logiques et que les signaux au sein du système sont également numériques.

Alors une PLL numérique est une boucle à verrouillage de phase dont l'ensemble des éléments internes sont à base de circuits numériques, il n'y a pas de composants passifs. Son terme anglophone est ADPLL, un signal dans une BVPN (boucle à verrouillage de phase numérique) peut être un signal binaire (ou signal "bit"), comme ce fait le cas avec la PLL numérique classique, et il peut aussi s'agir d'un signal "mot". Mot de code numérique provenant du registre de données, des sorties parallèles d'un compteur, etc. Les versions numériques du détecteur de phase sont déjà connues, mais nous devons maintenant trouver des circuits numériques pour le filtre de boucle et pour le VCO, le partenaire numérique du VCO est le DCO.

Il existe un nombre presque illimité de blocs de fonction purement numériques pour une BVPN[1].

I.7.1. Caractéristiques de la PLL numérique

Les PLL numériques sont extrême petite taille sur silicium qui la rend spécifiquement compétitive pour la plupart d'applications numériques ; dans le cadre des réductions de taille des circuits intégrés, la PLL numérique est particulièrement facile à synthétiser, au contraire de la PLL mixte ainsi qu'à la PLL analogique[6].

I.7.2. Eléments constitutifs de BVPN

Les blocs qui composent la PLL analogique sont les mêmes qui compose la BVPN mais leurs équivalents en numérique, que ce soit le détecteur de phase, ou le filtre de boucle et le VCO dans ce qui suite on a présenté des exemples numériques des blocs constitutifs de la BVPN.

a. Détecteur de phase numérique

Le comparateur de phase était le seul composant numérisé depuis longtemps. Dans la BVPN on trouve trois implémentations courantes du comparateur de phase numérique sont :

Porte exclusive ou (EXOR), bascule JK, détecteur de phase-fréquence numérique.

La moyenne du signal de sortie de comparateur de phase est proportionnelle au déphasage entre les deux entrées. Par exemple, si le déphasage entre le signal de référence et le signal de VCO est de 90 degrés, le signal de sortie aura un rapport cyclique de 50 %, ce qui implique une moyenne de 0 pour une sortie comprise entre -1 et +1. Si le déphasage s'écarte des 90 degrés, le rapport cyclique de sortie change proportionnellement à la différence de phase en entrée, et ainsi la valeur moyenne du signal de sortie. Par contre, si les signaux d'entrée ne sont pas symétriques, par exemple avec un rapport cyclique différent de 50 %, alors la gamme de détection est réduite. Par exemple, pour un signal d'entrée de rapport cyclique faible, 5 %, la gamme de détection devient seulement $5 \cdot (2\pi) / 100$. Ce système est un détecteur de phase, mais pas un détecteur de fréquence [7].

b. Filtre de boucle numérique

Il existe deux types les plus importants de filtres de boucle numériques.

Filtre de boucle de compteur haut / bas et filtre de boucle de compteur K

L'un des deux filtres de boucle numériques les plus importants est le compteur K illustré à la figure I.12. Ce filtre de boucle fonctionne toujours avec le détecteur de phase EXOR ou bascule JK [7].

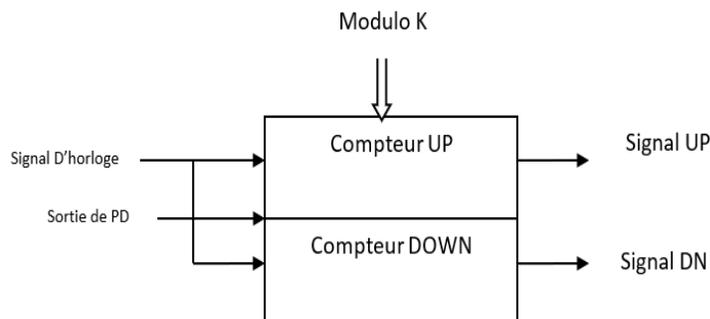


Figure I.12.Filtre de boucle compteur K[7].

Le compteur K est composé de deux compteurs indépendants, généralement appelés « compteur UP » et « compteur DOWN ». En réalité, les deux compteurs comptent toujours vers le haut avec K est le module des deux compteurs ; c'est-à-dire que le contenu des deux compteurs se situe dans une plage allant de 0 à K-1, K peut être contrôlé par l'entrée de commande du module K et correspond toujours à une puissance entière de 2. Le fonctionnement du compteur K est contrôlé par le signal qui vient à partir de comparateur de phase, si ce signal est élevé, le « compteur DOWN » est actif, tandis que le contenu du compteur UP reste inactif, dans le cas contraire, le « compteur UP » compte, mais le compteur DOWN reste inactif. Les deux compteurs sont recyclés à 0 lorsque le contenu dépasse K-1. Le bit le plus significatif du « compteur UP » est utilisé comme sortie « signal UP » et le bit le plus significatif du « compteur DOWN » est utilisé comme sortie « signal DN » [7].

c. Oscillateur à commande numérique

Une variété de DCO peut être conçus, selon l'application de la BVPN on utilise l'un des deux circuits suivants :

Diviser par compteur N ou bien le type d'accumulateur DCO

Dans ce qui suite nous présentons le diviseur par compteur N

d. Diviser par compteur N

La solution la plus simple est le DCO avec Diviser par compteur N, il est utilisé pour réduire le signal généré par un oscillateur haute fréquence fonctionnant à haute fréquence avec Le signal de sortie parallèle à N bits d'un filtre de boucle numérique permet de contrôler le facteur de mise à l'échelle de DCO [7].

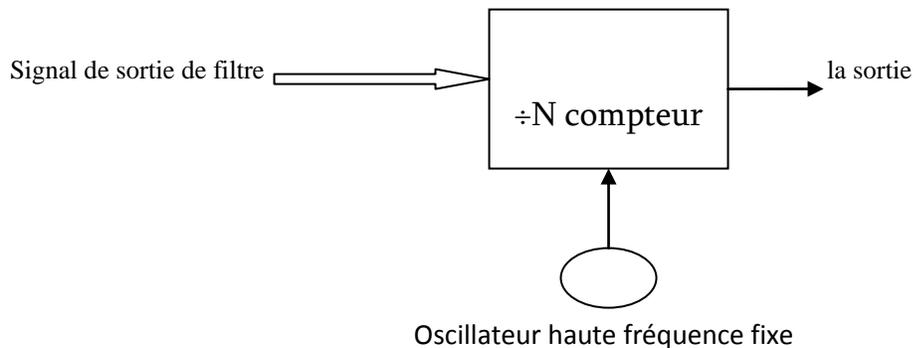


Figure I.13. Diviser par compteur N [7].

I.8. Conclusion

Dans ce chapitre, nous avons présenté deux différents types de PLL et l'essentiel de la théorie qui les décrit. Nous avons donné, dans ce chapitre, une attention particulière aux PLL numérique, car parmi les objectifs de ce travail est l'implémentation de la PLL en FPGA. En plus, nous avons présenté les réponses

des fonctions de transfert des PLL analogique, car elles sont primordiales pour déterminer la bande passante des PLL et optimiser le choix des composants des différents blocs, et les composant du filtre en particulier.

Pour cela, dans le chapitre suivant nous étalons exploiter un logiciel de simulation pour étudier une PLL dans la plage des fréquences dédiés au système de télécommunications LTE.

CHAPITRE II :
SIMULATION
D'UN
SYNTHÉTISEUR
DE FRÉQUENCE
A BASE DE LA
PLL

II.1. Introduction

Au courant de l'utilisation de la PLL on trouve plusieurs applications sont à la base des PLLs, Parmi ces applications on trouve les plus importantes : la démodulation de fréquence et de phase, la reconstruction de porteuse, extraction de signal, conception de radars à effet doppler, réalisation de filtres de poursuite la synthèse de fréquence...etc., cette dernière est un circuit à base de PLL avec une chaîne retour à base de diviseur de fréquences en boucle fermée. Les synthétiseurs de fréquences est la pièce maîtresse dans les circuits de télécommunications pour la sélection des fréquences. Dans ce chapitre on va exploiter un logiciel ADIsimPLL pour étudier notre PLL en simulant une chaîne de synthétiseur de fréquences. Pour cela, en premier temps, on étudie le circuit de synthétiseur de fréquences et son principe de fonctionnement, et en présentant le logiciel ADIsimPLL, et ensuite étudiant un exemple de synthétiseur de fréquence à base de PLL dédiés aux systèmes de télécommunications LTE.

II.2. Le synthétiseur de fréquences

Un synthétiseur de fréquence est un circuit électronique qui génère une plage de fréquences à partir d'une seule fréquence de référence. Les synthétiseurs de fréquence sont utilisés dans de nombreux appareils modernes tels que les récepteurs radio, les téléviseurs, les téléphones mobiles, les récepteurs de télévision par câble, les récepteurs satellite et les systèmes GPS.

Les synthétiseurs de fréquences permettent de générer des signaux de différentes fréquences à partir d'un signal de référence de fréquence fixe.

Un synthétiseur de fréquence est, par extension, un instrument qui permet de générer, dans une gamme de fréquences donnée, un signal dont on peut ajuster la fréquence et l'amplitude et auquel il peut être imposé une modulation de fréquence, de phase ou d'amplitude[8].

II.3. Les caractéristiques les plus importantes des synthétiseurs de fréquences

II.3.1. Plage de fréquences et pas de synthèses

La plage de fréquences est la zone comprise entre la plus petite et la plus grande des fréquences synthétisables.

Le pas de synthèse est la différence de fréquence entre deux fréquences synthétisées consécutives. Habituellement le pas de synthèse est inférieur ou égal à la largeur d'un canal, de façon à pouvoir opérer sur tous les canaux du standard de télécommunications [9]. Comme il est illustré à la figure II.1.

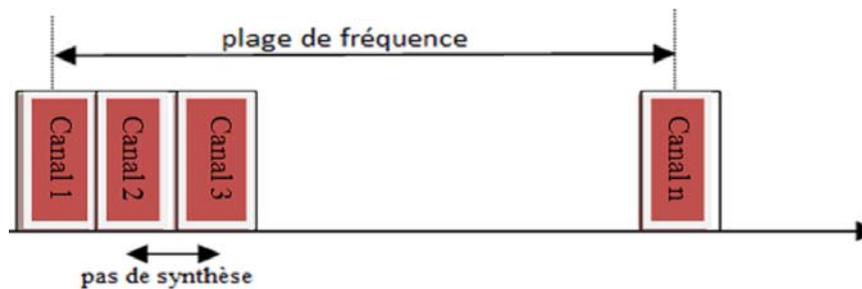


Figure II.1.Plage de fréquences et pas de synthèse.

II.3.2. Plage d'accorde

La plage d'accord est la plage de fréquence que la synthèse de fréquence doit couvrir. Elle dépend de l'application[28], par exemple la plage d'accord en réception pour le LTE est la gamme de fréquence 1710-2170 MHz.

II.3.3. Pureté spectrale

Un synthétiseur de fréquences devrait générer un signal purement sinusoïdal, sans bruit ou parasite. Un premier défaut est la génération de raies parasites dans le spectre du signal synthétisé. Souvent, à la place d'une raie unique à la fréquence synthétisée, on retrouve un ensemble de raies parasites de faibles puissances. Elles sont générées par le synthétiseur [2].

II.3.4. Temps d'acquisition

Représente le temps de stabilisation du synthétiseur lors d'un changement de fréquence. On peut rencontrer des temps d'acquisition de quelques millisecondes à quelques microsecondes[2].

II.4. Principes de base du synthétiseur PLL

Deux types de synthétiseur de fréquence qu'on trouve : à division entière et à division fractionnaire.

II.4.1. Synthétiseur de fréquences à division entière

La boucle à verrouillage de phase à division entière est la plus simple structure de synthétiseur à PLL. Son schéma-bloc est présenté à la figure II.2. Son architecture est composée d'un comparateur phase-fréquence suivi d'une pompe de charge, un filtre de boucle, un oscillateur contrôlé en tension et un diviseur de fréquence. Les diviseurs de fréquences sont en général des diviseurs numériques, qui divisent par un nombre entier, souvent programmable[2].

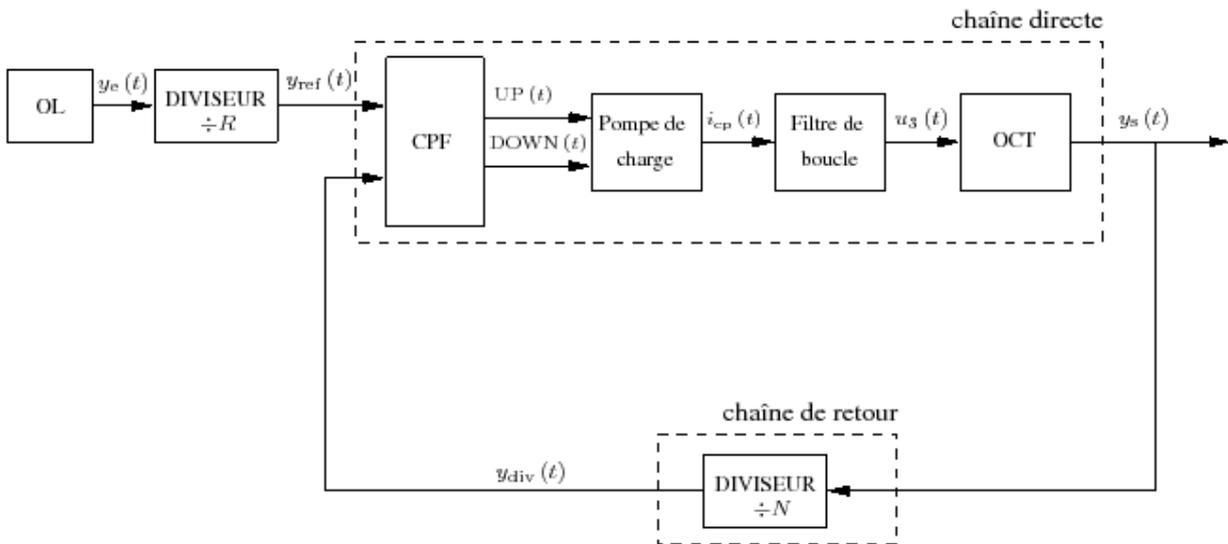


Figure II.2. Schéma-bloc d'une PLL à division entière[2].

Lorsque la boucle est verrouillée, la fréquence de sortie peut s'écrire :

$$y_s = \frac{R}{N} y_{ref} \quad (II.1)$$

y_{ref} , y_s , R et N représentent respectivement, la fréquence générée par l'oscillateur local, la fréquence de sortie de la PLL, le rapport de division programmable de la chaîne de retour et le rapport de division programmable d'entrée [2].

La réalisation de synthétiseur de fréquence se fait par deux possibilités qui sont:

- Soit faire varier la fréquence de référence. La plage de variation de la fréquence de sortie est alors égale à la plage de variation de f_{ref} multipliée par N .
- Soit utiliser une fréquence de référence fixe, généralement générée à partir d'un oscillateur à quartz, et faire varier le rapport de division N en utilisant un compteur programmable dans la boucle de retour.

II.4.2. Synthétiseur de fréquences à division fractionnaire

L'architecture en N fractionnel permet une résolution en fréquence qui représente une fraction du spectre fréquence de référence f_r , par conséquent, cette fréquence peut être supérieur à la taille du pas, dans ce cas la division (par N) peut être réduite.

La division fractionnaire fonctionne sur le principe de la division entière dont le rapport de division est modifié de manière dynamique entre deux ou plusieurs valeurs, le rapport de division fractionnaire est alors égal à la moyenne temporelle des rapports entiers[2].

II.5. Description les blocs de synthétiseur de fréquence

Le synthétiseur de fréquences est conçu des blocs suivants :

II.5.1. La source de référence

Port d'entrée qui transmet la fréquence de référence pour déterminer l'erreur de phase, Le signal de référence f_0 il est délivré par un oscillateur à cristal (quartz).

II.5.2. Comparateur phase- fréquence

Le détecteur ou comparateur de phase est un circuit à deux entrées quidoit fournir, après filtrage, une tension continue ou lentement variable proportionnelle à l'écart de phase existant entre les deux signaux d'entrée et de retour de boucle[29].

Entrée :la fréquence de référence.

Sortie : up (Mesure de la fréquence de référence du bloc PFD).

Down (Mesure de la fréquence de la rétroaction du bloc PFD).

Lesfigures II.3 et II.4 illustrent le principe de fonctionnement du comparateur de phaseainsi que diagramme des différents signaux.

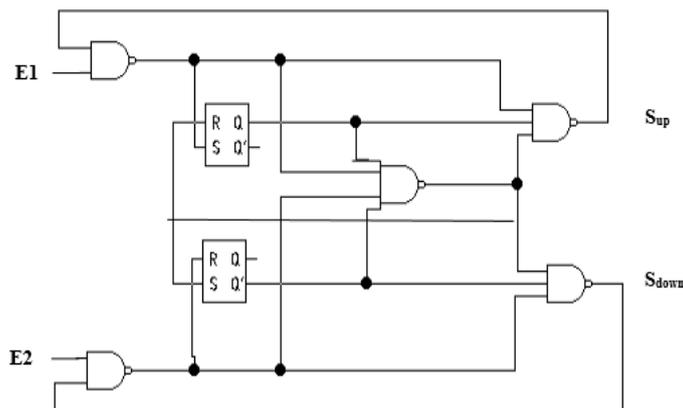


Figure II.3. Circuit d'un comparateur numérique de phase-fréquence[29].

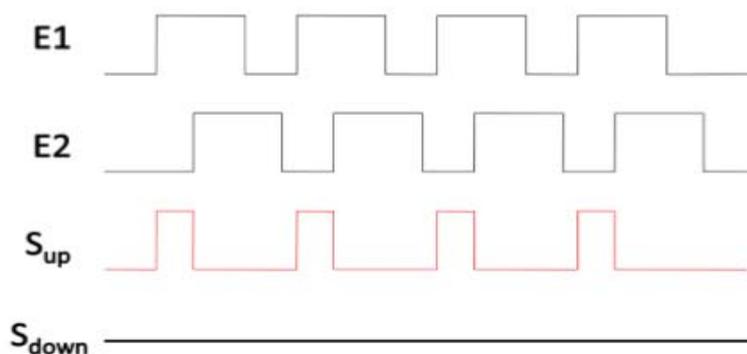


Figure II.4. Diagrammes des différents signaux[29].

Avec : $f_{ref}=E_1$; $f_{out}/N=E_2$

Le comparateur de phase va comparer la phase du signal à la sortie du diviseur de fréquence (f_{out}/N) avec un signal de référence (f_{ref}) généré par un oscillateur à cristal et génère un signal Up lorsque f_{ref} est en phase d'avance par rapport à f_{out}/N alors Sup est au niveau haut et Sdown au niveau bas, et un signal Down lorsque f_{ref} est en retard de phase par rapport à f_{out}/N alors Sup est au niveau bas et Sdown au niveau haut, Lorsque les phases de f_{ref} et f_{out}/N sont identiques : UP = DOWN = 0

II.5.3. La pompe de charge

La pompe de charge permet de convertir les signaux « Up » et « Down » venant du PFD, en courant qui va être injecter dans le filtre de boucle. La fonction de la pompe à charge est de convertir les états logiques du détecteur en signaux analogiques appropriés pour contrôler le VCO. La figure II.5 présente un exemple de circuit de pompe de charge à base de transistor à effet de champs, la figure III.6 présente la sortie de pompe de charge selon les deux entrées Sup et Sdown.

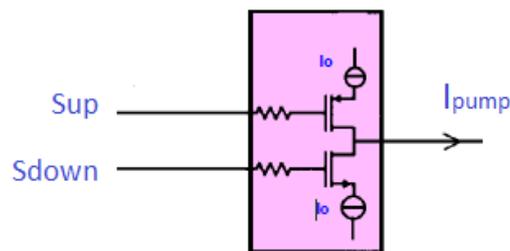


Figure II.5. Circuit de la pompe de charge[29].

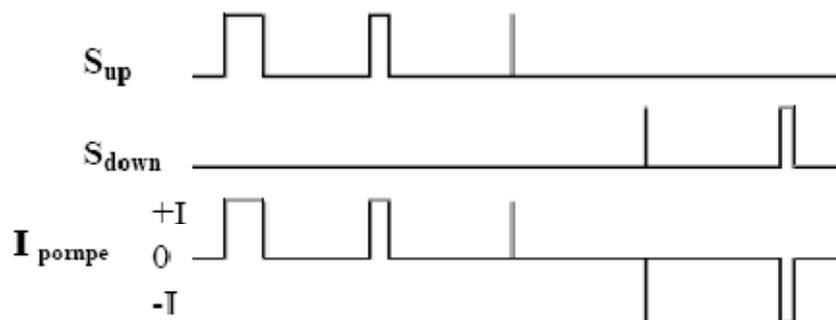


Figure II.6. Fonctionnement de la pompe de charge[29].

Dans cette architecture, deux sources de courant sont pilotées par les signaux UP et DOWN venant du PFD. Pour un délai positif, le signal UP est actif et commande alors l'injection d'une quantité de charges. Pour un délai négatif, l'interrupteur commandé par le signal DOWN devient passant et une quantité de charges est retirée[2].

II.5.4. Le filtre de boucle

Le bloc du sous-système filtre de boucle est un filtre passif dont l'ordre peut varier de deux à quatre. Dans un système de boucle à verrouillage de phase (PLL), le sous-système filtre la sortie du bloc Pompe de Charge et fournit la tension de commande à un bloc VCO pour générer le signal de fréquence requise. Ce filtre permet de filtrer la sortie du comparateur de phase dans la boucle à verrouillage de phase, PLL. Le filtre de boucle est l'élément qui garantit la stabilité de la PLL [29].

II.5.5. L'Oscillateur Contrôlé en Tension (VCO)

L'oscillateur commandé en tension est le bloc de circuit qui génère un signal sinusoïdal dont la fréquence f_{out} varie autour de cette fréquence d'oscillation et le diviseur de fréquence divise le signal de sortie par un facteur N permettant de le comparer au signal de référence de fréquence inférieure [29].

II.5.6. Diviseur de fréquence

Le diviseur de fréquence permet de diviser f_{out} afin de le comparer à f_{ref} . Cette utilisation du diviseur de fréquence sur la boucle de retour de la synthèse de fréquence.

II.6. Fonctions de transfert

La détermination des fonctions de transfert est primordiale pour la simulation des systèmes à base de PLL. La figure II.7 présente un schéma simplifié pour représenter la réponse d'un synthétiseur de fréquences.

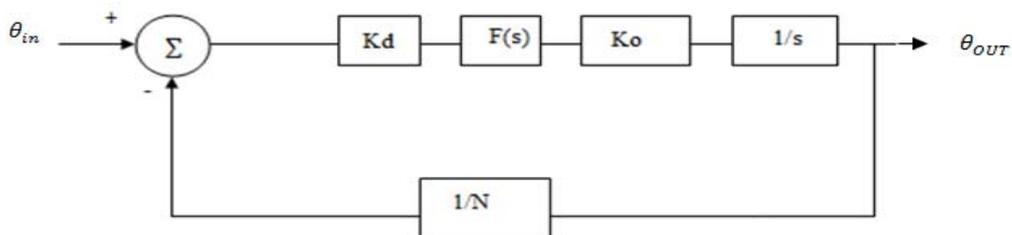


Figure II.7. Modèle linéaire de la PLL [30].

En se basant sur ce diagramme simplifié (figure II.8), et la théorie de contre-réaction, on peut obtenir les équations des fonctions de transfert suivantes :

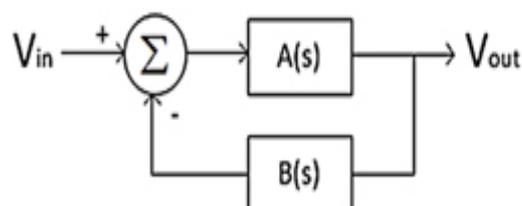


Figure II.8. Schéma de principe d'un système asservi [30].

- Fonction de transfert de la chaîne directe : $A(s) = \frac{\theta_{out}}{\theta_{ref}} = \frac{K_D F(s) K_0}{s}$ (II.2)
- Fonction de transfert de la chaîne de retour : $B(s) = \frac{1}{N}$ (II.3)
- Fonction de transfert de la boucle ouverte : $H(p) = A(s) \cdot B(s)$ (II.4)

$$H(p) = \frac{K_D F(s) K_0}{N \cdot s} \quad (\text{II.5})$$

- Fonction de transfert en boucle fermée : $A(s) = \frac{V_{out}}{V_{in}} = \frac{A(s)}{(1 + A(s)B(s))}$ (II.6)

II.7. Elaboration et évaluation des performances du filtre de boucle

La source de courant du détecteur de phase envoie la charge de la pompe dans le filtre à boucle, qui convertit ensuite la charge en tension de commande du VCO. Le condensateur shunt C1 est recommandé pour éviter des tensions discrètes au niveau du port de commande du VCO en raison des modifications instantanées de la sortie de courant de la pompe à charge. Une section de filtre passe-bas peut être nécessaire pour certaines applications de synthétiseur hautes performances nécessitant une réjection supplémentaire des bandes latérales de référence, appelées Spurs.

Une méthode de conception de filtre utilise la largeur de bande et la marge de phase du gain en boucle ouverte pour déterminer les valeurs des composants. La localisation du point de déphasage minimal à la fréquence de gain unitaire de la réponse en boucle ouverte [30], comme illustré à la figure II.9.

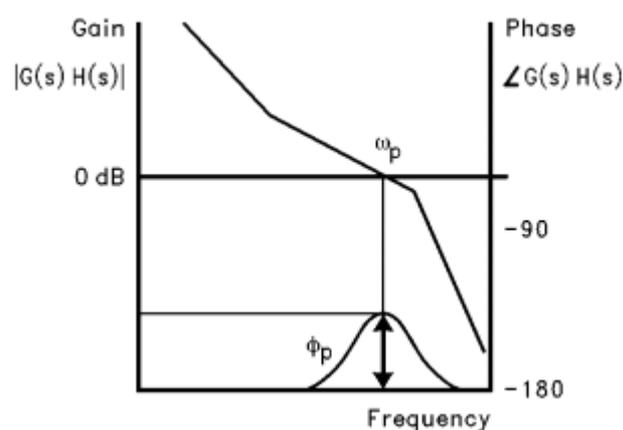


Figure II.9. Tracé de Bode de la réponse de boucle ouverte [30].

La relation de phase entre le pôle et le zéro permet également de déterminer facilement les valeurs des composants du filtre de boucle. La marge de phase, φ_p est définie comme la différence

entre 180° et la phase de la fonction de transfert en boucle ouverte à la fréquence, ω_p correspondant à gain à 0-dB, la marge de phase est choisie entre 30° et 70° .

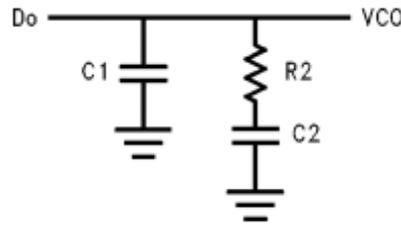


Figure II.10. Filtre passif de second ordre[28].

La fonction de transfert de second ordre de la figureII.10 est donné par :

$$Z(s) = \frac{s(c2.R2)+1}{s^2(c1.c2.R2)+s\tau1+s\tau2} \quad (\text{II.7})$$

Où $\tau1$ et $\tau2$ sont les constantes de temps qui déterminent les fréquences du pôle ω_p et du zéro ω_z on trouve :

$$\omega_p = \frac{1}{\tau1} \quad (\text{II.8})$$

$$\text{avec } \tau1 = R2 \cdot \frac{c1.c2}{c1+c2}$$

$$\omega_z = \frac{1}{\tau2} \quad (\text{II.9})$$

$$\text{Avec } \tau2 = R2.C2$$

Ainsi, le gain en boucle ouverte de PLL du troisième ordre peut être calculé en termes de fréquences ω , les constantes de temps de filtrage $\tau1$ et $\tau2$ et les constantes de conception K_D , K_0 et N .

$$A(s).B(s)|_{s=j.\omega} = \frac{-K_D.K_0(1+j\omega.\tau2)}{\omega^2 C1.N(1+j\omega.\tau1)} \cdot \frac{\tau1}{\tau2} \quad (\text{II.10})$$

D'après l'équation (II.10), on peut voir que le terme de phase dépendra du pôle unique et de zéro, comme la marge de phase est déterminée dans l'équation(II.11). La marge de phase disponible est donc proportionnelle au rapport $C1$ et $C2$.

$$\varphi(\omega) = \tan^{-1}(\omega.\tau2) - \tan^{-1}(\omega.\tau1) + 180 \quad (\text{II.11})$$

En fixant la dérivée de la marge de phase égale à zéro comme indiqué dans l'équation(II.12)

$$\frac{d\varphi}{d\omega} = \frac{\tau2}{1+(\omega.\tau2)^2} - \frac{\tau1}{1+(\omega.\tau1)^2} = 0 \quad (\text{II.12})$$

Le point de fréquence correspondant au point d'inflexion de phase est trouvé en termes de constantes de temps de filtrage $\tau1$ et $\tau2$. Cette relation est donnée dans l'équation (II.13).

$$\omega_p = \frac{1}{\sqrt{\tau_2 \cdot \tau_1}} \quad (\text{II.13})$$

Pour assurer la stabilité de la boucle, nous voulons que la marge de phase soit maximale lorsque l'amplitude du gain de la boucle ouverte est égale à 1. L'équation (II.10) donne alors

$$C1 = \frac{Kd \cdot KO \cdot \tau_1}{\omega_p^2 \cdot N \cdot \tau_2} \cdot \left\| \frac{(1 + j\omega_p \cdot \tau_2)}{(1 + j\omega_p \cdot \tau_1)} \right\| \quad (\text{II.14})$$

Par conséquent, si la bande passante de la boucle, ω_p et la marge de phase φ_p , sont spécifiés, les équations (II.7) et (II.12) permettent de calculer les deux constantes de temps, τ_1 et τ_2 .

Les formules pour τ_1 et τ_2 sont présentées dans les équations (II.9) et (II.10)

$$\tau_1 = \frac{\sec \varphi_p - \tan \varphi_p}{\omega_p} \quad (\text{II.15})$$

$$\tau_2 = \frac{1}{\omega_p^2 \cdot \tau_1} \quad (\text{II.16})$$

A partir des constantes de temps τ_1 et τ_2 et la largeur de bande de la boucle ω_p , les valeurs de $C1$, $R2$ et $C2$ sont obtenues dans les équations.

$$C1 = \frac{\tau_1}{\tau_2} \cdot \frac{Kd \cdot KO}{\omega_p^2 \cdot N} \quad (\text{II.17})$$

$$C2 = C1 \cdot \left(\frac{\tau_2}{\tau_1} - 1 \right) \quad (\text{II.18})$$

$$R2 = \frac{\tau_2}{C2} \quad (\text{II.19})$$

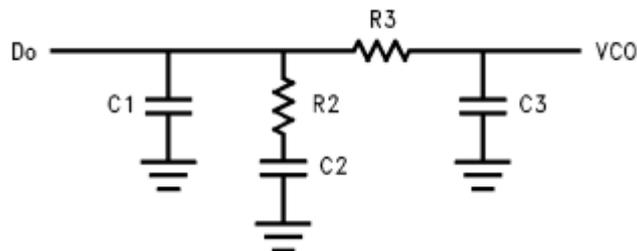


Figure II.11. Filtre de boucle de troisième ordre.

L'atténuation ajoutée par le filtre passe-bas est :

$$A_{ATTEN} = 20 \log[(2 \cdot \pi \cdot Fref \cdot R3 \cdot C3)^2 + 1] \quad (\text{II.20})$$

On finit la constante de temps comme suit :

$$\tau_3 = R3 \cdot C3 \quad (\text{II.21})$$

Ainsi en termes d'atténuation des raies de références ajoutées par le pôle passe-bas, nous aurons:

- $\tau_3 = \sqrt{\frac{10^{\left(\frac{ATTN}{20}\right)} - 1}{(2\pi \cdot Fref)^2}} \quad (\text{II.22})$

- $\tau_2 = 1 / [\omega_c^2 \cdot (\tau_1 + \tau_3)] \quad (\text{II.23})$

$$\bullet \quad \omega_c = \frac{\tan \varphi \cdot (\tau_1 + \tau_3)}{[(\tau_1 + \tau_3)^2 + \tau_1 \tau_3]} \times \left[1 + \frac{(\tau_1 + \tau_3)^2 + \tau_1 \tau_3}{[\tan \varphi \cdot (\tau_1 + \tau_3)]^2} - 1 \right] \quad (\text{II.24})$$

La capacité C1 peut être exprimée au moyen de la relation :

$$C_1 = \frac{\tau_1 \cdot K_D \cdot K_O}{\tau_2 \cdot \omega_c^2 \cdot N} \times \left[\frac{(1 + \omega_c^2 \cdot \tau_2^2)}{(1 + \omega_c^2 \cdot \tau_1^2)(1 + \omega_c^2 \cdot \tau_3^2)} \right]^{1/2} \quad (\text{II.25})$$

Similaire au filtre d'ordre 2 que nous avons

$$C_1 = c_1 \cdot \left(\frac{\tau_2}{\tau_1} - 1 \right) \quad (\text{II.26})$$

$$R_2 = \frac{\tau_2}{c_2} \quad (\text{II.27})$$

II.8. Bruit de phase dans les synthétiseurs de fréquences

Le bruit de phase est un phénomène très important pour les systèmes de télécommunication. Ce bruit minimise des performances du système en réduisant le rapport signal-bruit, en agrandissant la puissance du canal voisin et en réduisant le rejet de ce canal voisin [29]. La définition usuelle du bruit de phase à bande latérale unique SSBPN (Single Side Band Phase Noise) est celle-ci

$$|L(f_c)|_{dBc/Hz} = 10 \log \left(\frac{P(f_c + f_{signal, 1Hz})}{P_{signal}} \right) \quad (\text{II.28})$$

Le bruit de phase se caractérise par une variation aléatoire de la phase du signal qui se traduit par un étalement du spectre de la raie principale, jusqu'à quelques MHz. Il correspond au rapport de la densité spectrale de puissance sur une bande normalisée de 1Hz centrée sur la fréquence $f_{signal} + f_c$ et de la puissance du signal de sortie à la fréquence f_{signal} . Il est noté L (fc) en dBc/Hz [31].

Le bruit de phase du synthétiseur a principalement deux effets sur un émetteur-récepteur. Le premier est la dégradation du rapport signal sur bruit de la transmission, si tout ou partie de l'information est portée par la phase de la porteuse [31]. Le second effet est similaire à celui des raies parasites : avec le bruit de phase, un émetteur va rayonner de l'énergie sur les canaux adjacents, alors qu'un récepteur va recevoir de l'énergie des canaux adjacents [32]. Le phénomène de bruit de phase peut être le résultat du bruit thermique, du bruit dans tous les composants actifs ou passifs.

II.9. Présentation générale de logiciel ADIsimPLL

Cette partie décrit les résultats de simulation et de conception d'un synthétiseur de fréquence.

L'outil de conception ADIsimPLL est un outil complet et facile à utiliser pour la conception et la simulation, de synthétiseur à base de PLL. Tous les effets non linéaires clés qui peuvent avoir une influence sur les performances du PLL peuvent être simulés, y compris le bruit de phase, les Spurs et l'impulsion anti-backlash. La famille de synthétiseurs ADF4xxx est exploitée dans une gamme d'applications telles que les stations de base sans fil, le réseau local, les téléphones mobiles et les PDA (*Personal Digital Assistant*), l'accès sans fil à large bande, l'équipement industriel, l'instrumentation et l'équipement de test, le satellite, le sonar et le CATV. L'outil de conception ADIsimPLL

d'AnalogDevices offre un support pour les synthétiseurs les plus récents de l'entreprise, y compris le nouveau PLL ADF4351 hautement intégré pour la station de base et les applications générales et le récepteur large bande intégré ADRF6850 pour les applications satellitaires. Ce logiciel est toujours en cours de développement par des ingénieurs en vue d'optimiser les conceptions rendant plus rapide et plus facile à atteindre des objectifs visés pour des différentes applications. La Figure II.12 représente l'interface graphique du logiciel ADIsimPLL[13].

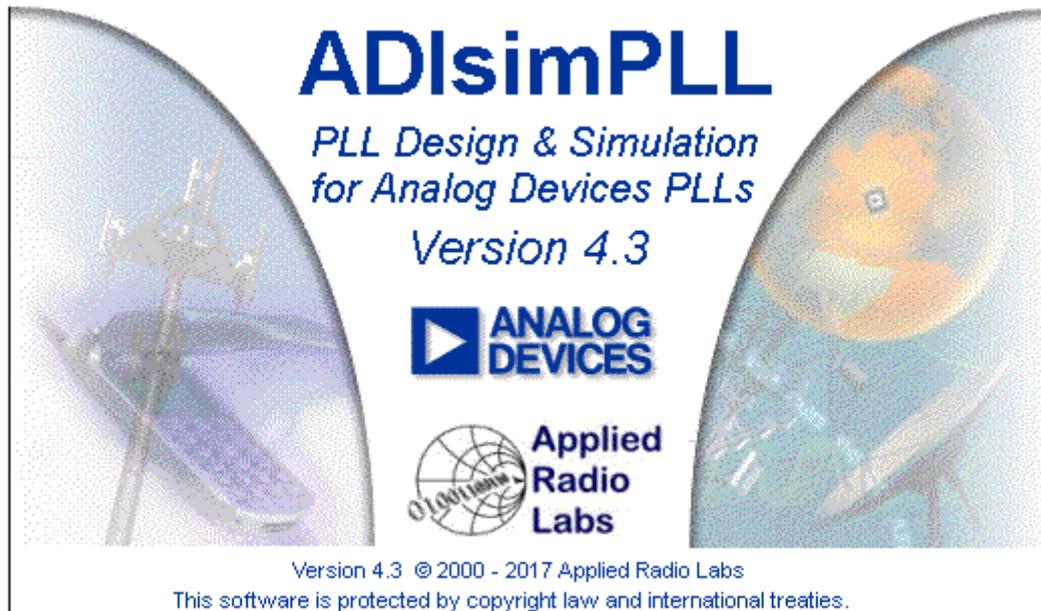


Figure II.12. L'interface graphique du logiciel ADIsimPLL.

II.10. La simulation d'un synthétiseur de fréquence

La bande de fréquence choisie pour la norme LTE/LTE-A, ou la technologie LTE est en pleine expansion depuis le début de l'année en France et dans d'autres pays dans le monde. Elle est utilisée pour transiter de grosses quantités de données jusqu'à une vitesse impressionnante de 1 Gigabit/s en download et 300 mégabit/s en « upload », notamment avec le LTE- Advanced. Qui occupe les bandes [1710 - 1770] MHz pour l'émission et [2100 - 2170] MHz pour la réception, que 60 MHz est la largeur pour chaque bande, répartie en canaux de (5,10,15,20) MHz. Le composant que nous avons choisi pour réaliser la boucle à verrouillage de phase est l'ADF4107. Le synthétiseur de fréquence ADF4107 peut être utilisé pour implémenter oscillateurs locaux dans les sections « upconversion » et « downconversion » des récepteurs et des émetteurs sans fil. Il consiste en un faible bruit. PFD numérique (détecteur de fréquence de phase), une pompe de charge de précision, diviseur de référence programmable, compteurs A et B programmables, et un prescaler à double module ($P / P + 1$). Le A (6 bits) et B compteurs (13 bits), en conjonction avec le module double prescaler ($P / P + 1$), implémenter un diviseur N ($N = BP + A$). Dans outre, le compteur de référence à 14 bits (compteur R) permet fréquences REFIN sélectionnables à l'entrée PFD. Un PLL complet (boucle à verrouillage de phase) peut être implémenté si le synthétiseur est utilisé avec un filtre de boucle externe et VCO

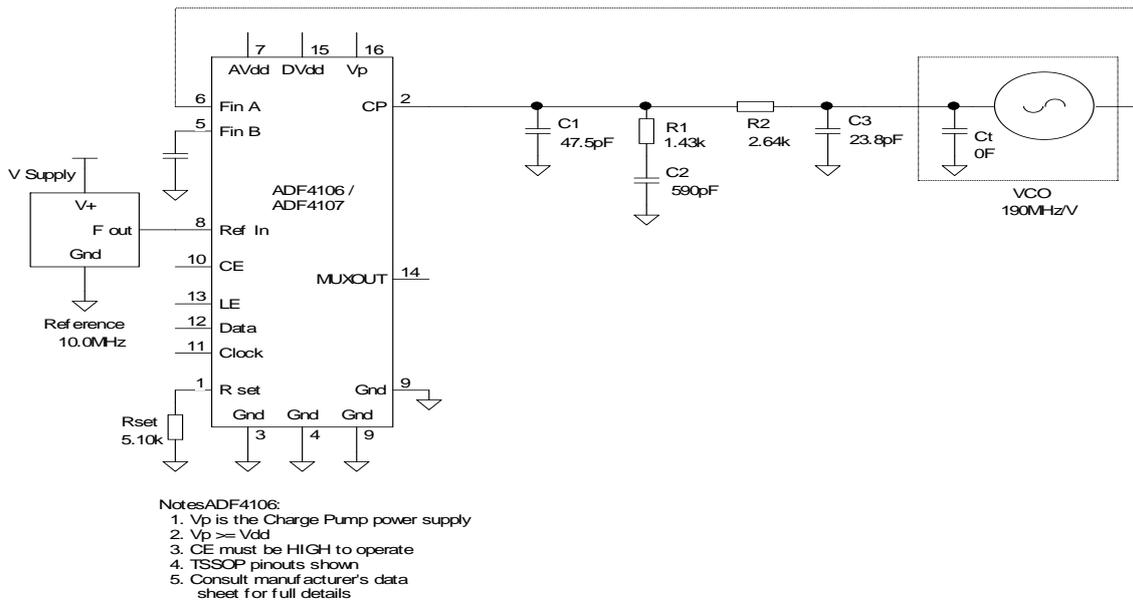


Figure II.14. Schéma de l'ADF4106 pour ADIsimPLL.

Le choix de la fréquence de coupure du filtre résulte d'un compromis entre le bruit de phase du signal produit par le synthétiseur et sa pureté spectrale (niveau des raies parasites espacées de la fréquence de référence). Les valeurs des composants du filtre du 3ème ordre issues de calcul pour une marge de phase de 43° et une largeur de boucle de 500 kHz sont montrées sur le tableau que l'on approchera sans grand inconvénient par les valeurs normalisées[10].

Capacités du filtre	Résistances du filtre
C1=47.5 pF	R1=1.43kΩ
C2=590 pF	R2=2.64kΩ
C3=23.8 pF	

Tableau II.1. Valeurs des éléments du filtre de boucle du troisième ordre.

II.12. Les réponses fréquentielles de la boucle

En introduisant les différentes données pour simuler un synthétiseur de fréquence dédiée aux systèmes LTE sur ADIsimPLL on trouve les différentes réponses.

II.12.1. Gain et phase en boucle ouverte

La figure II.15 dessine l'amplitude et la phase du gain en boucle ouverte de la PLL, La courbe en rouge représente le gain et en bleu représente la phase. Nous avons déterminé la valeur de ω_{pe} (le point où le gain égale à 0dB), correspond à la valeur de la bande passante de boucle ω_p qui est 500kHz, et la phase -137.01°, La marge de phase couramment utilisée est $(180^\circ - 137.01^\circ)$ qui donne 43° (marge de phase).

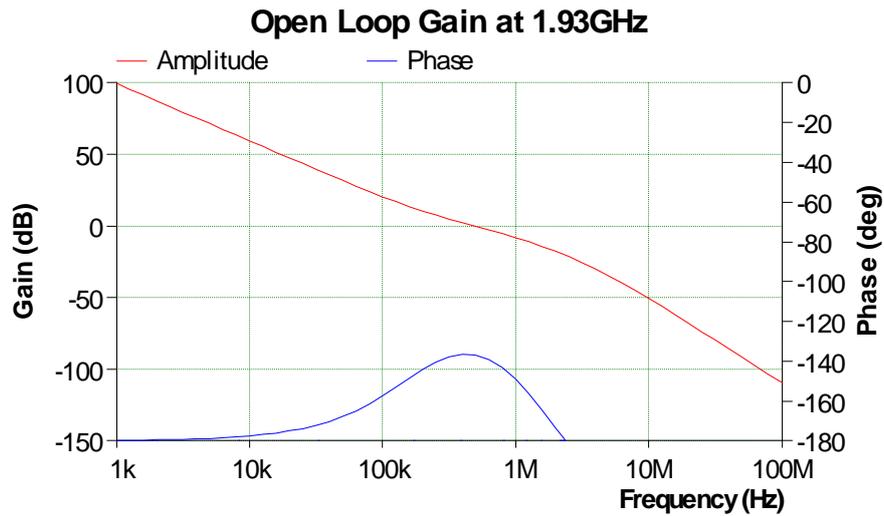


Figure II.15. La réponse de la boucle ouverte.

II.12.2. Gain et phase en boucle fermée

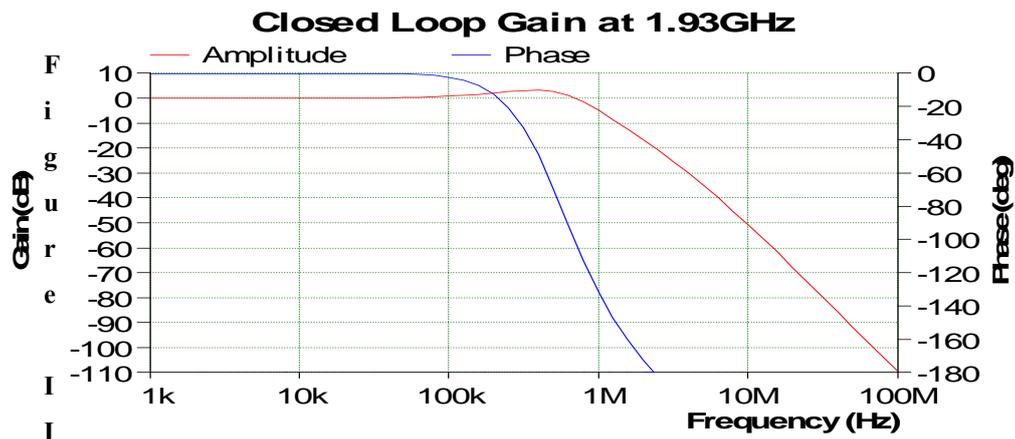


Figure II.16. La réponse de la boucle fermée.

La figure II.16 représente l'amplitude et la phase de la fonction de transfert de phase de l'oscillateur de référence à la sortie du VCO. Nous distinguons que la fonction de transfert de la boucle fermée est très grande en amplitude. Cette valeur de gain reste constante à 0dB jusqu'à il atteint la largeur de bande de la boucle, ensuite il chute rapidement. Cette figure représente aussi le gain de bruit dans la bande passante et l'atténuation de ce bruit au-dessus de cette fréquence (ω_p). Le gain dans la bande passante de la boucle vient en grande partie du rapport de division N de la boucle.

II.12.3. Le bruit de phase à la sortie du synthétiseur de fréquence

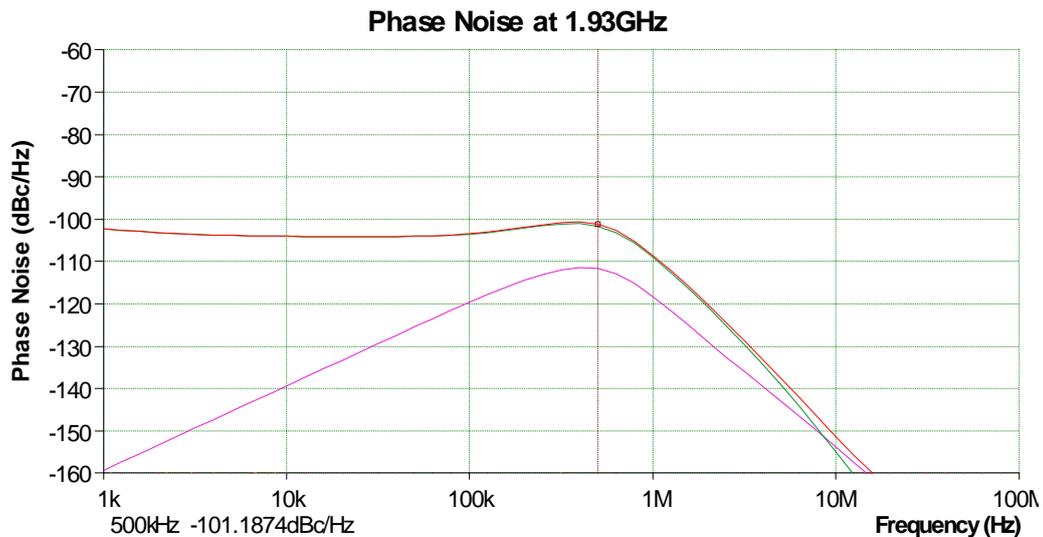


Figure II.17. Le bruit de phase de chaque composant dans le PLL.

Cette figure (II.17) illustre le bruit de phase pour chaque composant, le bruit de référence en orange, le bruit du filtre de boucle en rose le bruit de la puce en vert, le bruit de VCO en violet et le bruit total en rouge. Nous avons remarqué que l'intérieur de la largeur de bande de boucle (1kHz-500kHz), le niveau du bruit l'oscillateur de référence est important du fait que le gain de la fonction de transfert en boucle fermée est élevé dans cette bande. Ces amplitudes des différents bruits diminuent au-delà de cette bande. Dans le but de démontrer que le bruit du VCO est hautement filtré par la PLL, en provoquant la réjection du bruit de phase ou erreur de phase dans la bande de boucle (les fréquences inférieures à 500 kHz),

II.12.4. La réponse de la modulation de fréquence

La figure II.18 démontre la réponse à la modulation de fréquence appliquée au VCO alors qu'il est verrouillé dans la PLL. Il s'agit d'une réponse passe-haut car, dans la bande passante de la boucle, la PLL tente de supprimer la modulation.

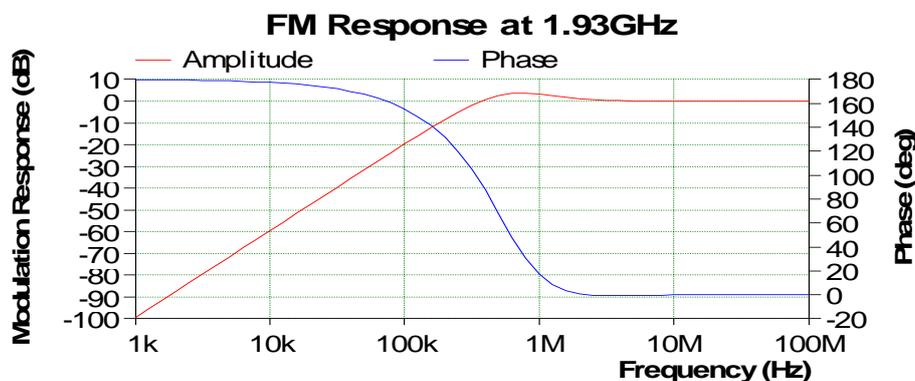


Figure II.18. Réponse de la modulation de fréquence.

II.12.5. Leakagespurs (raies de référence)

La figure II.19 montre les raies de référence générée par une fuite au niveau du détecteur de phase. Le courant de fuite est défini sur le panneau de données sous Puce / Détecteur de phase / Fuite, et peut être réglé pour inclure les fuites du détecteur de phase plus les fuites provenant d'autres sources, telles que les condensateurs de filtrage. ADIsimPLL ajoutera à cela les éventuels courants de polarisation dus à un amplificateur opérationnel dans le filtre en boucle. Il existe d'autres sources des raies de référence qui ne sont pas prises en compte dans le modèle de fuite. La figure III.8 illustre les raies de référence générées à des multiples de la fréquence de comparaison. Le courant de fuite (leakagecurrent) de la pompe de charge que nous avons introduit dans la simulation est de 5uA. Les résultats obtenus montrent les trois premières raies de référence: -27.11dBc, -42.92dBc, -52.93dBc et -60.22dBc situent respectivement à des multiples de la fréquence de comparaison 5MHz, 10 MHz et 15MHz, 20MHz. L'erreur de phase RMS enregistrée est de 0.28 degrés.

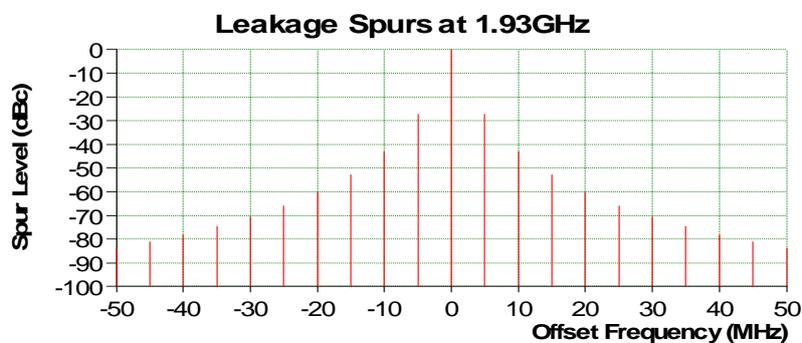


Figure II.19. Les raies latérales de référence.

II.13. La réponse temporelle

Ce graphique montre comment la fréquence de sortie de la PLL change dans des conditions transitoires de fréquence. Il existe trois types de transitoires possibles

1. Etape de fréquence : la PLL est supposée être verrouillée en F1 et doit passer à F2 (non disponible pour les PLL à fréquence unique)
2. Mise sous tension : la sortie du filtre de boucle est supposée être à 0V et la PLL est commandée pour passer à F1.
3. Modulation : la PLL est commutée entre deux fréquences ou un autre mode de modulation.

Dans les systèmes de télécommunications modernes, le synthétiseur a souvent des exigences strictes pour le temps d'établissement, défini comme le temps que prend la PLL pour commuter entre deux fréquences différentes. Ce temps est mesuré depuis le début de l'action de commutation de fréquence au temps de la nouvelle fréquence. Un changement de zone de localisation implique nécessairement un changement de fréquence d'émission (et de réception également). Si la communication est en cours, le système doit pouvoir commuter vers la nouvelle valeur de fréquence allouée par le BTS au terminal sans que l'utilisateur ne sente une interruption de service.

Pratiquement il s'agit d'une atténuation maximale acceptable de puissance correspondant à l'exécution de l'algorithme du Handover. La figure II.20 montre la réponse temporelle de la PLL, le temps de commutation est environ de 4,47 μ s pour un balayage de toutes les porteuses de la bande de réception (60 MHz pour la bande N°10 du LTE/LTE-A constitué de 12 canaux de 5 MHz).



Figure II.20. La réponse temporelle de la PLL.

II.13.1. Comparateur de phase-fréquence

Sortie du détecteur de phase, ce graphique montre la sortie du détecteur de phase de la PLL dans des conditions transitoires. La figure II.21 montre le signal résultant de la pompe de charge (PD out) varie selon l'état des entrées Up et Down. Lorsque la sortie Up est à son niveau haut, le courant de la pompe de charge est positif et lorsque Down est à son niveau haut, la pompe de charge délivre un courant négatif.

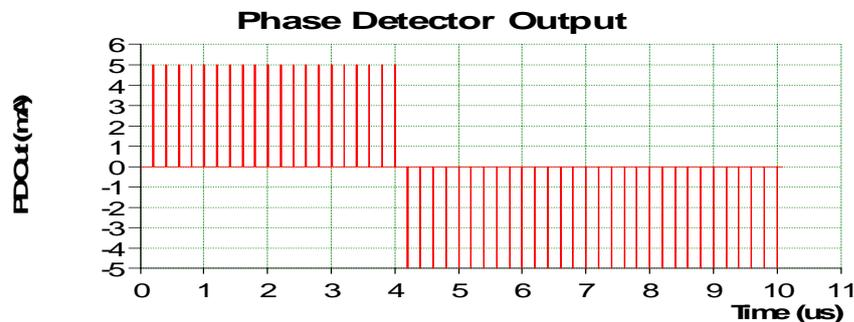


Figure II.21. Sortie de la pompe de charge.

II.13.2. Erreur de fréquence

La figure III.22 montre l'erreur de fréquence de la PLL dans des conditions transitoires. Autrement dit, si la PLL est verrouillée à une fréquence F_1 et est commandé à l'instant ($t=0$) pour accéder à une autre fréquence F_2 , l'erreur de fréquence $|F(t)-F_2|$ peut être ainsi déterminée à travers le graphique ci-dessous.

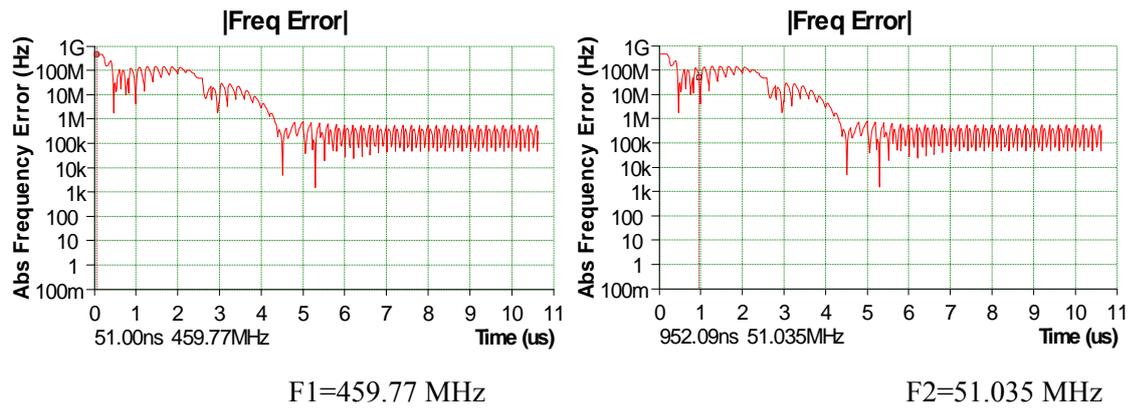


Figure II.22. L'erreur de fréquence de la PLL.

II.14. Conclusion :

Dans ce chapitre, nous avons simulé la PLL en étudiant et en simulant le circuit du synthétiseur de fréquence, opérant dans la bande 10 du réseau LTE avec une fréquence de 10MHz. Nous avons représenté les équations qui régissent ce système ainsi que les différentes réponses des grandeurs de cette chaîne. En plus, nous avons pu prendre en considération l'effet de la marge de phase et la largeur de bande de boucle sur le temps de verrouillage et le niveau des raies de références à des multiples de la fréquence de comparaison. Dans le chapitre suivant nous allons programmer sur VHDL et implémenter une PLL sur une carte FPGA.

CHAPITRE III :
IMPLEMENTATION
DE
LA PLL EN FPGA

III.1 Introduction

Les PLLs sont devenues omniprésentes en raison de leur polyvalence, est un système de contrôle en boucle fermée utilisé pour synchroniser la phase et la fréquence avec celles d'un signal entrant. Entre la PLL analogique et numérique, les PLLs analogiques sont sensibles aux paramètres de processus et, pour chaque nouvelle technologie, il convient de le redéfinir mais tous ces problèmes peuvent être résolus par BVPN (Boucle à Verrouillage de Phase Numérique). Cette dernière est devenue une activité lucrative en raison de son évolutivité, de son verrouillage plus rapide et de sa conception facile avec des modifications de processus. Dans une BVPN, tous les composants utilisés sont de nature numérique ou l'utilisation de composants numérisés dans BVPN confère une immunité vis-à-vis de facteurs tels que la capacité parasite, le bruit et la dépendance à la température qui constituaient un goulot d'étranglement pour les PLL analogiques. Parmi les méthodes de conception de BVPN est l'implémentation du circuit sur un FPGA [7]. Ce dernier, nous a permis de réaliser un prototypage rapide d'une BVPN, afin de présenter son bon fonctionnement. Pour cela, nous avons choisi d'implémenter une PLL purement numérique sur un circuit reconfigurable FPGA

III.2 Le circuit FPGA

Les réseaux logiques programmables FPGA (*Field Programmable GateArrays*) sont des composants VLSI entièrement reconfigurables, ce qui permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs. L'avantage de ce genre de circuit est sa grande souplesse qui permet de les réutiliser à volonté pour effectuer des algorithmes différents en un temps très court [15]. Le progrès de ces technologies permet de fabriquer des composants toujours plus rapides et à haute densité d'intégration, ce qui permet de les utiliser afin de programmer des applications plus importantes. Les circuits FPGA sont constitués d'une matrice de blocs logiques programmables entourés de blocs d'entrée/sortie programmables. L'ensemble est relié par un réseau d'interconnexions programmables. La figure III.1 présente la structure interne d'un FPGA de type matrice symétrique (XC4000 de Xilinx)[16].

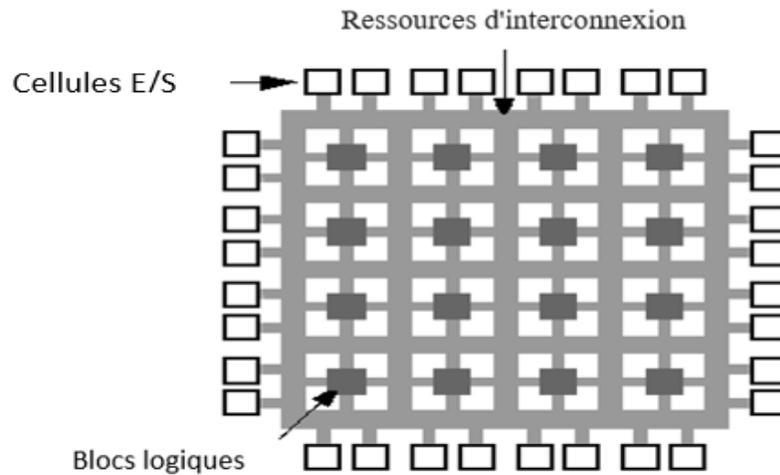


Figure III.1. Structure interne d'un FPGA [17].

L'utilisateur peut programmer la fonction à réaliser dans chaque bloc (CLB):

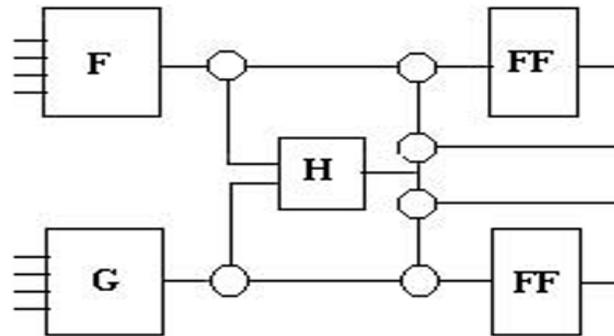


Figure III.2. Schéma bloc d'une cellule dans un bloc CLB [17].

Les interconnexions peuvent être aussi programmées entre les blocs :

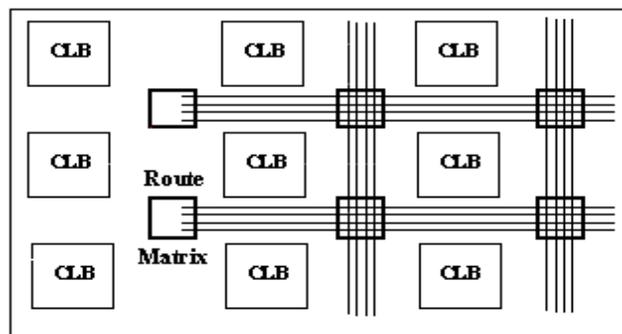


Figure III.3. Les interconnexions entre les blocs d'un FPGA [17].

L'avantage des FPGA est de pouvoir les configurer sur place, sans le passage chez le fondeur, ce qui permet de les utiliser quelques minutes après leurs conceptions. Les FPGA les plus récents sont configurables en une centaine de millisecondes, ils sont utilisés pour un développement rapide et un bon marché des circuits ASIC (*Application Specific Integration Circuit*). Donc c'est des composants standards combinant la densité et les performances d'un pré-diffusé (un circuit ASIC) avec la souplesse due à la reprogrammation des PLD (*Programmable Logic Device*).

III.2.1. Les différents types de FPGA

Nous distinguons trois grands types de FPGA :

III.2.1.1. Les FPGA à mémoire statique (SRAM)

Au début des années 80, la société XILINX a commercialisé pour la première fois un circuit FPGA sous la dénomination de LCA, elle a proposé une grande variété de produits qui utilisent la technologie de la mémoire vive. Dans cette dernière, l'état de chaque interrupteur est commandé par une cellule mémoire classique qui comporte cinq transistors : deux couples de transistors constituent chacun un inverseur logique et l'ensemble de ces deux inverseurs bouclés entre eux réalise un bistable statique, plus un transistor de programmation [16].

III.2.1.2. Les FPGA à mémoire Flash

Lattice Semiconductor et ACTEL offrent des FPGA qui enregistrent leurs configurations dans une mémoire flash, un type de EEPROM. Puisque la mémoire flash est une mémoire non volatile, le FPGA conserve sa configuration même lorsqu'il n'est pas alimenté [18].

III.2.1.3. Les FPGA à anti-fusibles

Les circuits à anti-fusibles partagent, avec ceux à SRAM, le sommet de la gamme des circuits programmables en vitesse et en densité d'intégration. Il est clair que ces circuits ne sont programmables qu'une seule fois. Le procédé des cellules anti-fusibles a été élaboré par la société ACTEL en 1986 sous l'appellation PLICETM [18].

III.2.2. Applications de FPGA

Les FPGA sont utilisés dans diverses applications nécessitant de l'électronique numérique (télécommunications, aéronautique, transports...). Ils sont également utilisés pour le prototypage d'ASIC. Les FPGA sont généralement plus lents, plus chers à l'unité et consomment d'avantage d'énergie que leur équivalent les ASIC. Cependant, ils ont plusieurs avantages :

- Délai de mise sur le marché plus court, car ce sont des composants standards,

- Temps de conception plus court, car les fonctions de base sont réutilisées, dont la reconfiguration autorise une validation préalable moins stricte,
- Coût inférieur pour de petites séries (moins de 10 000 unités). Mais avec l'évolution technologique, cette quantité tend à augmenter. En effet, le prix d'une puce est proportionnel à sa surface, qui diminue avec la finesse de gravure, tandis que les coûts initiaux pour fabriquer un ASIC (conception, tests, masques de gravure) sont en forte augmentation [15].

III.2.2.1. Les domaines d'application des FPGA

Nombreux domaines favorisent l'utilisation de ces circuits comme les systèmes embarqués, la télécommunication, le traitement du signal et de l'image, l'imagerie médicale, les capteurs biomédicaux, les réseaux sans fil, le cryptage, les dispositifs de sécurité et le prototypage des circuits etc..., Dans notre mémoire, nous nous intéressons plus au domaine de la télécommunication car il est le domaine où l'application de la BVP est très utilisée. Par exemple, nous trouvons l'application des FPGA dans le domaine de télécommunications très vaste. Par conséquent, selon *TransparencyMarketResearch* (le marché mondial des FPGA), le déploiement des technologies 3G et LTE dans le monde, avec des demandes en bande passante élevée, va tirer ce marché, avec des taux de croissance moyens annuels supérieurs à 9% entre 2013 à 2019, rien que pour ce domaine particulier. Le secteur des écrans pour les SmartPhone, Tablettes et Phablettes, terminaux dotés de fonctions tactiles de plus en plus évoluées, sera également très favorable à la progression du marché des FPGA [19].

III.3 Méthodes de conception

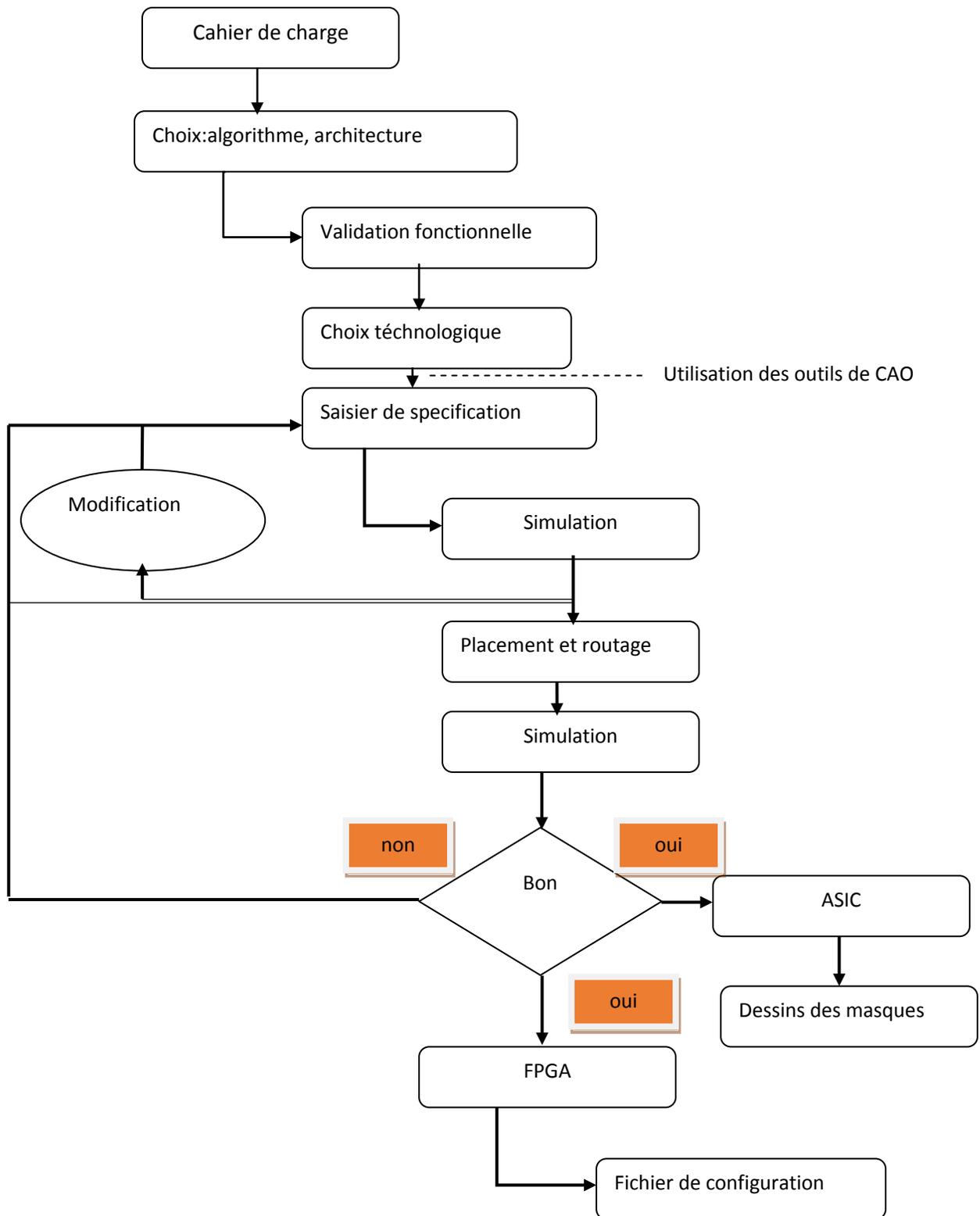
Dans une première étape ; il s'agit de définir et d'évaluer les différentes phases permettant la mise en place d'un flot de conception des circuits logiques programmables. Pour cela, nous distinguons deux méthodes de conception pour leurs développements [20] :

III.3.1. La conception des circuits à faibles densités

Le principe de programmation d'un SPLD est relativement simple, car il faut déterminer les fusibles à claquer. Mais ce travail peut devenir long et fastidieux dans le cas d'utilisation de réseaux logiques programmables complexes ou de fonctions à réaliser complexes. La programmation des SPLD nécessite un logiciel adapté pour le développement du projet et un programmeur permettant de programmer le circuit [20].

III.3.2. La conception des circuits à hautes densités

Le flot de conception classique des circuits de hautes densités (CPLD, FPGA, ASIC) est présenté par l'organigramme suivant :



Organigramme III.1. Conception des circuits à haute densité [20].

III.3.2.1. Simulation et synthèse

La synthèse est indépendante de la technologie du circuit cible, contient la saisie de l'application dans un outil de développement, en tenant compte de la nature de la fonction à réaliser et du composant qui va accueillir cette fonction, car c'est des critères importants pour faire un choix du mode de description qui facilite la conception de ce circuit. On peut définir trois descriptions : Une description schématique utilise des composants de base (hard macro, composants optimisés) fournis pour un circuit donné ou du moins une famille de ce dernier. On utilisera ce type de description lorsque la nature de la fonction à implanter est très structurée. Une description HDL utilise un langage de type comportemental permet de rendre le code source indépendant de la cible. Un code HDL normé (VHDL, VERILOG, ...) sera accepté par la plupart des outils (simulateurs, synthétiseurs...). Et une description mixte consiste à définir certains éléments hiérarchiques en schématique et d'autres de manière syntaxique. Il pourra par exemple s'avérer judicieux de définir schématiquement la structure globale d'une conception et de définir de manière comportementale chaque bloc fonctionnel [18]. La synthèse aboutit à une description du circuit au niveau logique, couramment appelée : "netlist". Cette étape prend en compte les contraintes imposées par le cahier des charges (surface, temps). La synthèse est validée toujours par une simulation logique qui utilise les caractéristiques temporelles fournies par la bibliothèque [20].

La simulation permet de charger le composant à simuler dans un simulateur (par exemple : MODELSIM) directement et faire varier entrée par entrée à chaque pas de simulation en définissant le temps qui dépendra de la fréquence du signal d'horloge. La simulation est réalisée par un fichier de test (test bench) consiste à créer un composant sans entrées –sorties en déclarant le composant à simuler et les signaux internes (autant de signaux que les entrées-sorties du composant). Ce banc d'essai est un module écrit en VHDL qui permet de faire varier les signaux internes connectés aux entrées afin de visualiser les résultats sur une fenêtre des courbes [24].

III.3.2.2. Programmation du FPGA

L'implémentation d'un circuit sur un FPGA est la création d'un fichier de configuration, contenant l'ensemble des informations relatives à ce circuit. Celui-ci peut alors être utilisé pour configurer le circuit programmable, soit à l'aide d'une mémoire non-volatile associée au circuit, soit directement à partir d'une interface externe (processeur, bus, liaison parallèle, etc...) [20]. Autrement dit, l'implémentation est l'étape où un fichier *.bit est envoyé dans le FPGA. Pour cela, il faut utiliser un langage approprié à une description matérielle et un compilateur qui contient les outils de CAO[21].

III.4 Les outils de développement

L'électronicien a toujours utilisé des outils de description pour représenter des structures logiques ou analogiques. Le schéma structurel que l'on utilise depuis si longtemps et si souvent n'est en fait qu'un outil de description graphique [20]. Aujourd'hui, l'électronique numérique est de plus en plus présente et tend bien souvent à remplacer les structures analogiques utilisées jusqu'à présent. Ainsi, l'ampleur des fonctions numériques à réaliser nous impose l'utilisation d'un autre outil de description [22]. Avec l'apparition des circuits logiques programmables de type SPLD, CPLD ou FPGA, l'utilisateur peut créer dans ces derniers toutes les fonctions logiques en utilisant des outils de développement mis à sa disposition par leurs fabricants ; ces outils doivent permettre de passer de la description du comportement d'une fonction logique à son câblage dans le circuit et cela de la manière la plus simple possible.



Figure III.4. L'utilité d'un outil de développement [20].

III.4.1. Les outils de CAO (Conception Assistée par Ordinateur) :

La conception assistée par ordinateur CAO (*Electronic Design Automation* : EDA) est la catégorie des outils servant à la production des systèmes électroniques allant des circuits imprimés jusqu'aux circuits intégrés [23]. Avec la miniaturisation continue de la technologie des semi-conducteurs, les circuits électroniques sont devenus trop complexes pour être conçus à la main ce qui a résulté la nécessité des outils informatiques adéquats et c'était la naissance des outils de CAO. Les concepteurs réalisent et testent les circuits sur ordinateur avant de lancer la fabrication. Nous pouvons distinguer trois types d'outils de CAO selon les modes de descriptions trouvés dans ces derniers :

- Un outil de CAO utilisé que pour la synthèse et la simulation (comme MODELSIM).
- Un outil de CAO qui utilise le mode schématique et textuel (comme ISE de XILINX).
- Un outil de CAO qui peut utiliser le mode schématique, textuel et permet même le dessin du masque (comme CADENCE).

III.4.1.1. Les outils de synthèse et simulation

L'outil de synthèse (Figure III.5) a pour objectifs : minimiser la surface de silicium, le temps de propagation ainsi que la consommation.

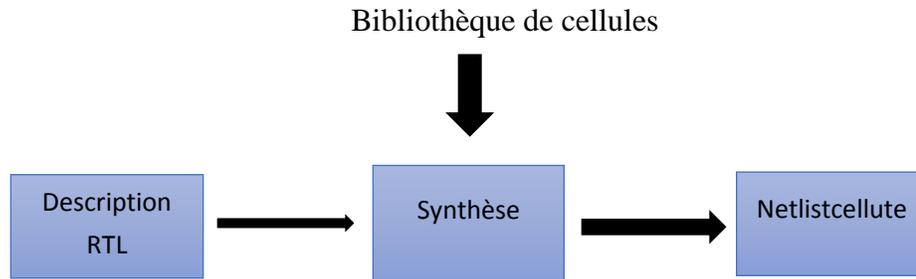


Figure III.5. L'outil de synthèse [20].

III.4.1.2. Les outils de placement et routage

Les outils de placement et routage (Figure III.6) permettent de minimiser la surface globale du silicium et les longueurs d'interconnexions.

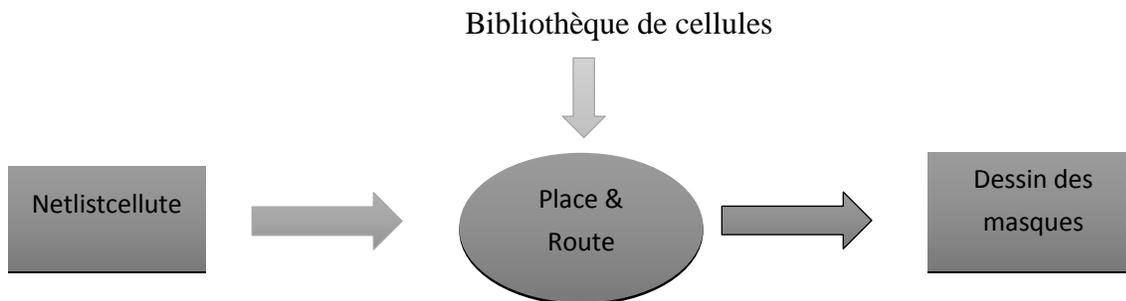


Figure III.6. Les outils de placement et de routage [20].

Nous avons utilisé dans notre travail le compilateur ISE 14.7, qui englobe les différents outils de CAO.

III.4.1.3. L'outil ISE 14.7

L'outil ISE 14.7 de la société Xilinx est un compilateur qui permet la description matérielle, la simulation fonctionnelle et l'implémentation des circuits réalisés sur les composants programmables. L'outil ISE permet :

- La description de circuits numériques sous forme de schémas logiques, de machines à états finis ou en langages de description matérielle (VHDL, Verilog, ABEL).
- La compilation, la simulation comportementale.
- La synthèse, le placement, le routage et l'implémentation.

- La simulation temporelle et l'analyse de timing.
- La programmation sur les circuits programmables de Xilinx (CPLD et FPGA) [25].

III.4.2. Les langages de description

Pour implémenter un circuit sur un FPGA, il est nécessaire d'utiliser un langage de description matérielle ou bien un outil de saisie graphique, après la compilation de cette description, le placement et le routage du circuit, un fichier de configuration est généré afin d'implémenter le circuit sur un FPGA choisi. Les principaux langages de description matérielle sont mentionnés dans le tableau suivant [20]:

Langage	Société
ABEL	Xilinx (Data I/O corporation)
CUPL	Logic Devices, Inc
PALASM	Lattice (MMI/AMD)
VERILOG	Verilog et Norme IEEE
VHDL	Norme IEEE
HANDEL-C	CELOXICA
L'outil Stream-C	Los Alamos National Laboratories
Système C	Synospys-CoWare

Tableau III.1. Les principaux langages de description [20].

Dans ce mémoire, nous avons utilisé le langage VHDL pour décrire le circuit.

III.4.2.1. Le langage VHDL

Développé dans les années 80 aux États-Unis, le langage VHDL est ensuite devenu une norme IEEE numéro 1076 en 1987. Révisée en 1993 pour supprimer quelques ambiguïtés et améliorer la portabilité du langage, cette norme est vite devenue un standard en matière d'outils de description de fonctions logiques [22]. Aujourd'hui, le langage VHDL permet :

- Concevoir des ASIC, programmer des composants programmables du type PLD, CPLD et FPGA,
- Concevoir des modèles de simulations numériques ou des bancs de tests,
- Il assure la portabilité des conceptions,
- De synthétiser des fonctions logiques complexes à partir d'une description concise.

Dans la pratique, une fois que les modules sont développés, ils sont intégrés dans une bibliothèque dont ils sont réutilisés en cas de besoin ultérieur.

III.5 L'architecture matérielle de la BVPN

L'implémentation de notre architecture de la boucle à verrouillage de phase numérique sur FPGA consiste à réaliser le circuit à l'aide du langage de description de haut niveau VHDL, nous allons donc décrire le circuit matériellement. La seconde étape consiste à effectuer une simulation fonctionnelle du circuit afin de vérifier son bon fonctionnement.

La boucle à verrouillage de phase numérique BVPN (Figure III.7), est un circuit qui rassemble des composants logiques en permettant le fonctionnement de la boucle.



Figure III.7. Le bloc de BVPN.

Le schéma structurel du BVPN (Figure III.8) est composé des trois blocs suivants : un comparateur de phase, un filtre de boucle et un oscillateur de commande numérique, en sachant que tous les blocs utilisés fonctionnent numériquement.

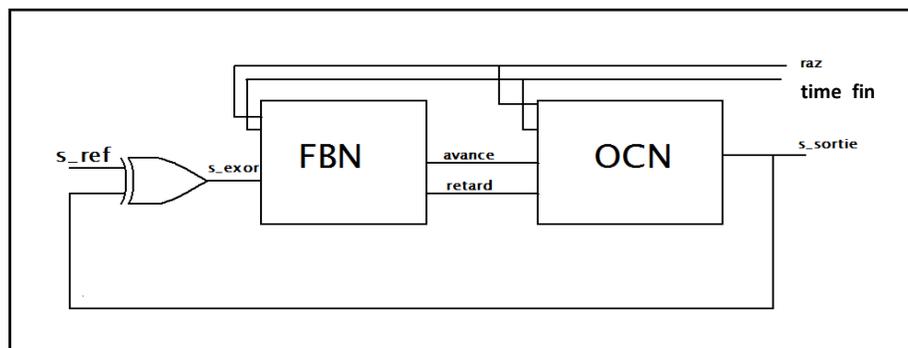


Figure III.8. Schéma structurel du BVPN.

III.5.1. Comparateur de phase (la porte XOR)

Comme nous avons déjà présenté dans le premier chapitre, le processus utilisé est simple et fiable, dont nous avons mis un comparateur simple (la porte XOR). Ce dernier permet la comparaison entre le signal d'entrée (le signal de référence) et le signal de sortie de l'oscillateur OCN (Oscillateur à Commande Numérique).

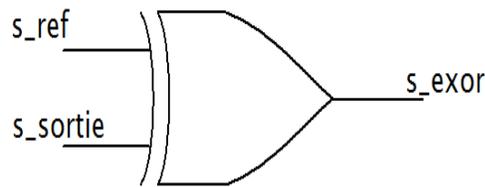


Figure III.9. La porte XOR.

La simulation fonctionnelle de la porte XOR sous ISE est présentée sur la figure suivante :

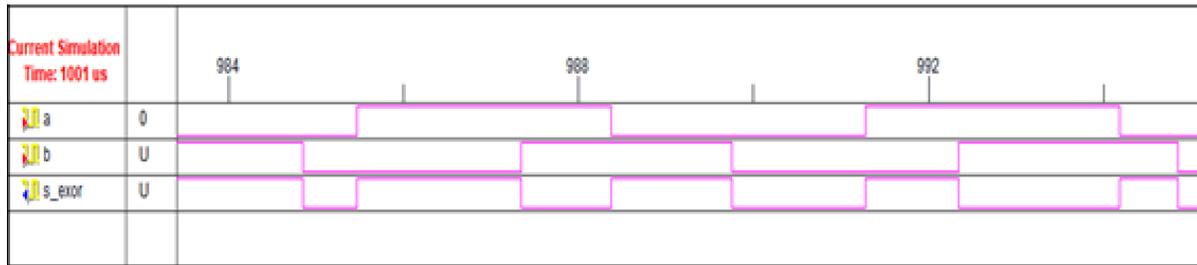


Figure III.10. Simulation de la porte XOR.

Où (a) représente le signal de référence " s_ref", (b) représente le signal de " s_sortie "

III.5.2. Filtre de boucle numérique

Le filtre de boucle numérique (FBN) est un circuit qui peut générer un signal de sortie dépendant de la sortie du comparateur de phase.

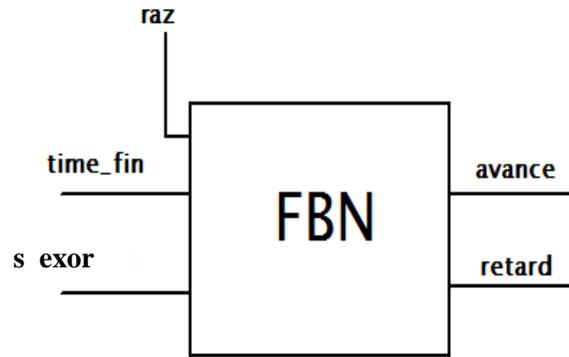


Figure III.11. Schéma bloc de filtre de la boucle.

L'architecture matérielle du filtre de la boucle est présentée par la figure III.12. Le circuit est composé de portes logiques, de bascules et de deux compteurs « UP counter » et « DOWN counter ». Ces derniers sont de même modulo E , dont l'opération du comptage se déroule dans un intervalle de 0 à $E-1$.

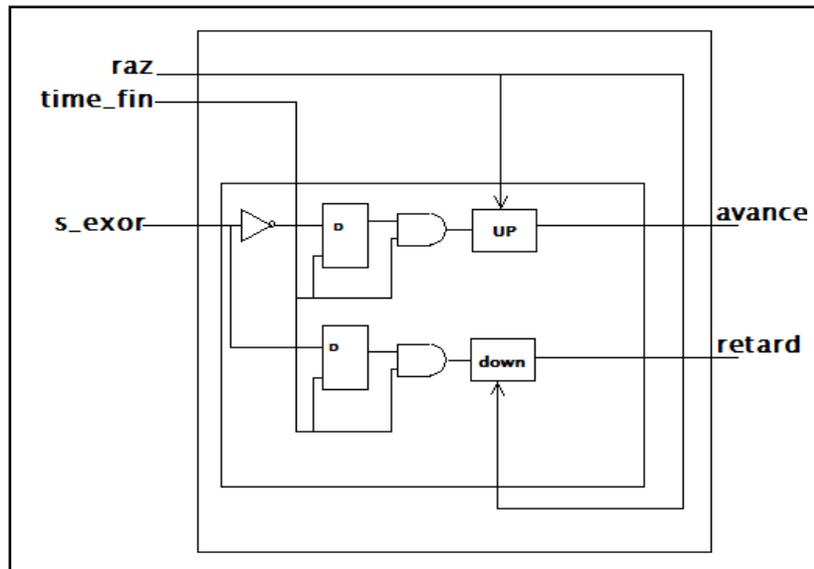


Figure III.12. Structure interne de FBN [26].

Le signal de commande s_cmd (qui est la sortie de CPN) va sélectionner le fonctionnement de l'un des deux compteurs, dont le premier commence à compter lorsque le signal s_cmd est à l'état haut ($s_exor=1$) et lorsqu'il est à l'état bas ($s_exor=0$), le deuxième compteur fonctionne. Les deux bascules sont utilisées afin de synchroniser le même signal d'horloge $time_fin$ aux deux compteurs.

Les deux sorties de filtre sont les signaux $avance$ et $retard$ des deux compteurs « UP counter » et « DOWN counter » respectivement. Ces derniers commencent à compter séparément et lorsqu'ils arrivent à la moitié du comptage (à $E/2$), chaque compteur active sa sortie ($avance$ et $retard$) séparément. Ces deux signaux vont commander le circuit suivant (OCN).

La simulation du filtre de boucle est présentée dans la figure III.13 :

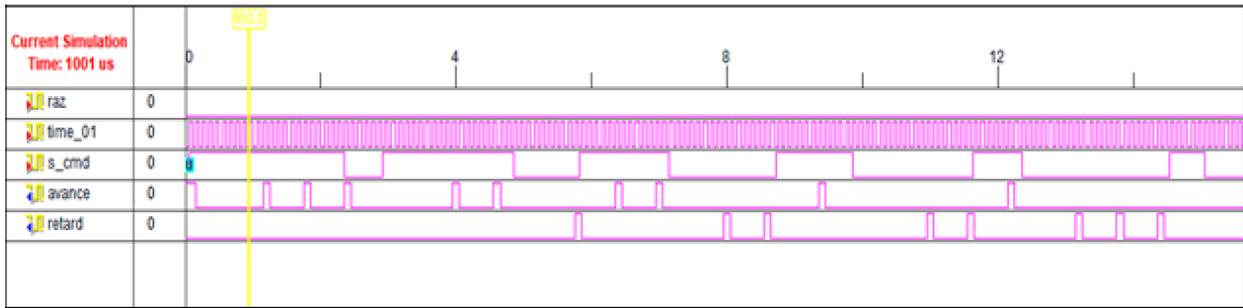


Figure III.13. Simulation de la sortie de filtre de boucle.

“time_01” représente le signal d’horloge” time_fin’, s_cmd est le signal de sortie de la porte xor (s_exor).

III.5.3. tOscillateur de commande numérique

L’OCN produit le signal de sortie dont la fréquence et la phase sont asservies sur celles de s_ref, via la commande issue du filtre. Cet élément occupe une place particulière au sein de la BVPN[27].

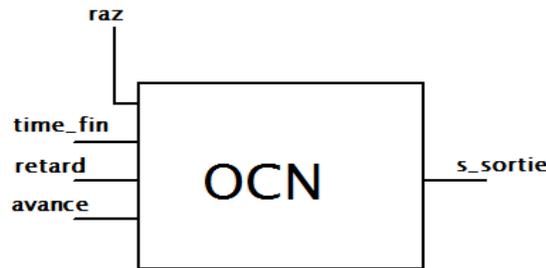


Figure III.14. Schéma bloc de l’OCN.

Plusieurs modèles de l’OCN peuvent être conçus, parmi eux, nous avons choisi le principe de fonctionnement du circuit « incréments-décément ». Par conséquent, notre OCN est réalisé à partir de deux blocs « divf » et « delay », illustré dans la figure III.15.

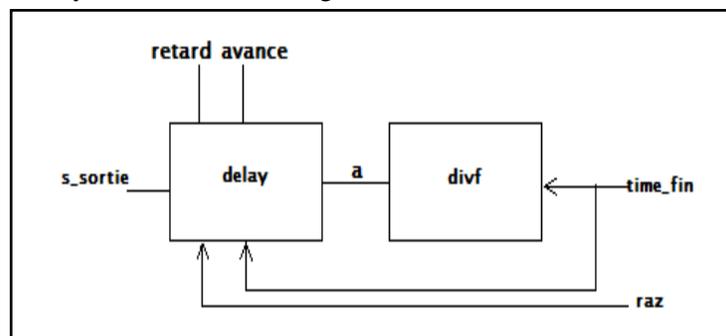


Figure III.15. Structure interne de l’OCN.

L'OCN est sensible aux fronts montants des signaux avance et retard, où la sortie de l'OCN est déduite par la fonction logique « $s_sortie = \text{not}(a) \text{ and } \text{not}(\text{time_fin})$ ». Nous avons obtenu ces trois cas suivants :

1^{ère} cas : A l'absence des deux signaux de commande (le retard et l'avance), l'OCN provoque une impulsion de sortie suivant la relation $s_sortie = \text{not}(\text{time_fin}) \text{ and } \text{not}(a)$ et cela pour chaque détection d'un front montant de l'horloge (time_fin) de l'OCN.

2^{ème} cas : Le signal avance (x) est activé à 1 si la fréquence divisée est à l'état haut ($a = 1$). Cette dernière va basculer à l'état bas pour deux périodes successives. Cela signifie que le signal de sortie « s_sortie » sera avancer par rapport à l'horloge d'une période.

3^{ème} cas : le signal retard (y) est activé à 1 si la fréquence divisée est à l'état bas ($a = 0$). Cette dernière va basculer à l'état haut pour les deux périodes successives, donc le signal de sortie « s_sortie » sera retarder d'une période par rapport à l'horloge.

La simulation de l'OCN a donné les résultats suivants (figure III.16).

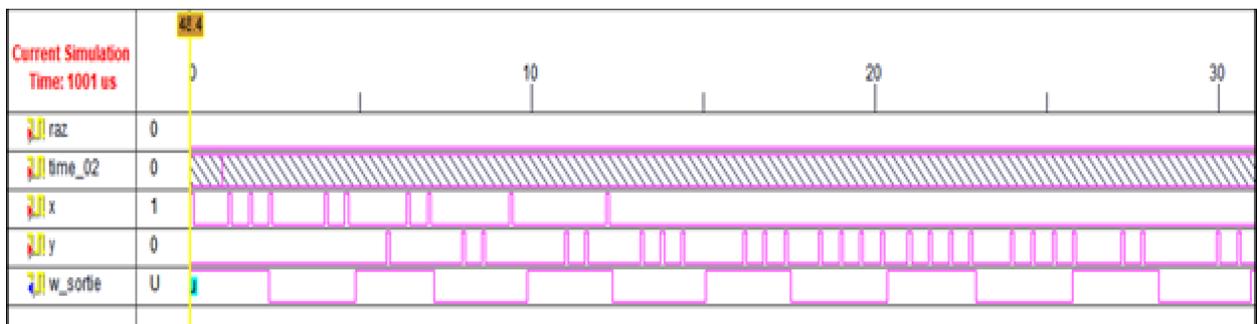


Figure III.16. La simulation de l'OCN.

Dans sa forme générale, la BVPN comprend principalement d'un générateur de fréquence (OCN), d'un comparateur phase-fréquence (porte XOR) et d'un filtre de boucle (FBN) (figure III.17).

Sachant que "time_02" représente le signal d'horloge "time_fin", x représente le signal de commande "avance", y représente le signal de commande "retard", w_sortie représente le signal de sortie de l'oscillateur s_sortie .

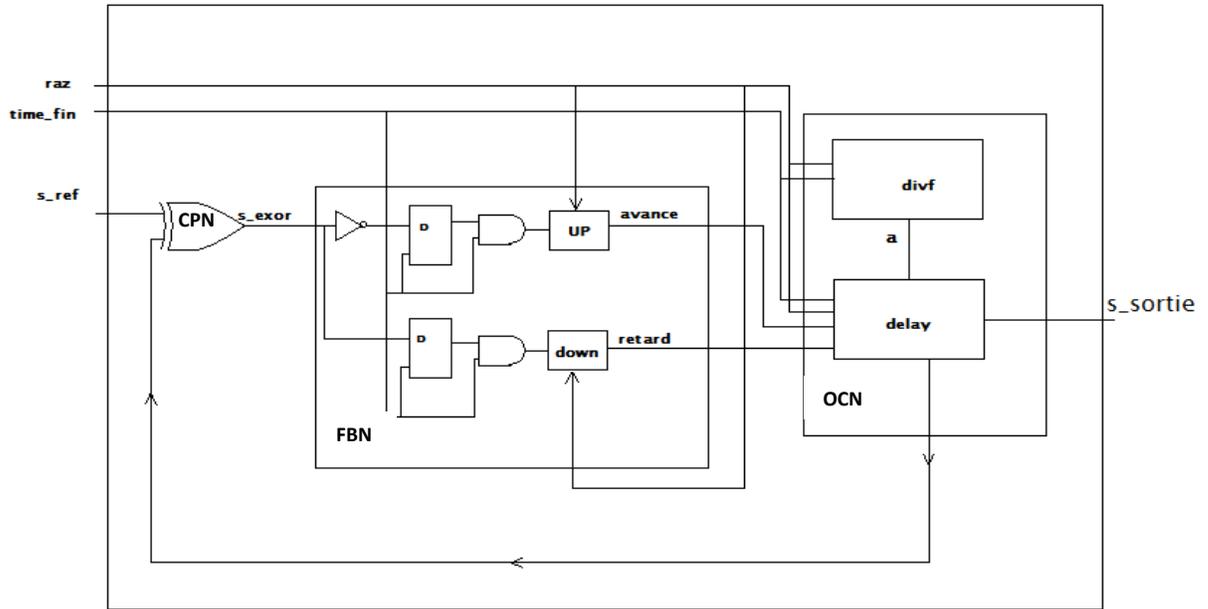


Figure III.17. La structure interne de la BVPN.

La simulation de la BVPN est présentée dans la figure III.18, avec un signal de remise à zéro (raz), « s_ref » représente le signal de référence, « s_sortie » le signal de sortie de l’OCN, « avance » et « retard » sont les deux signaux de commande de l’OCN.

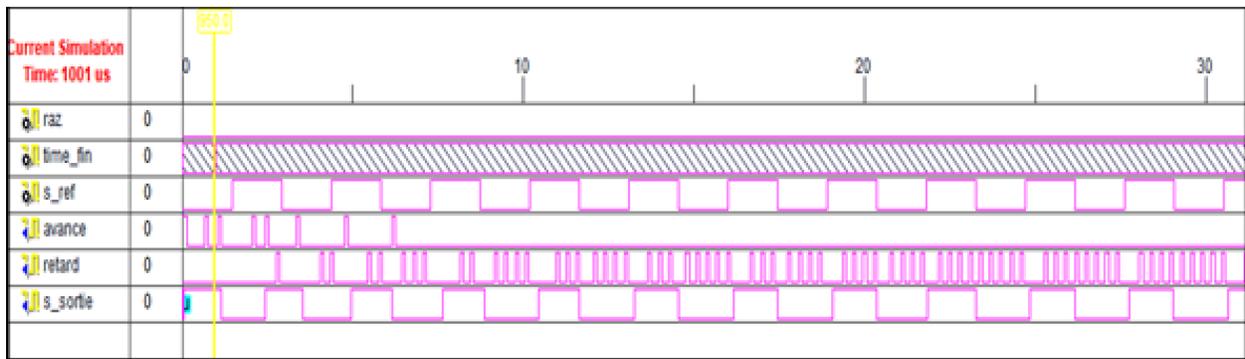
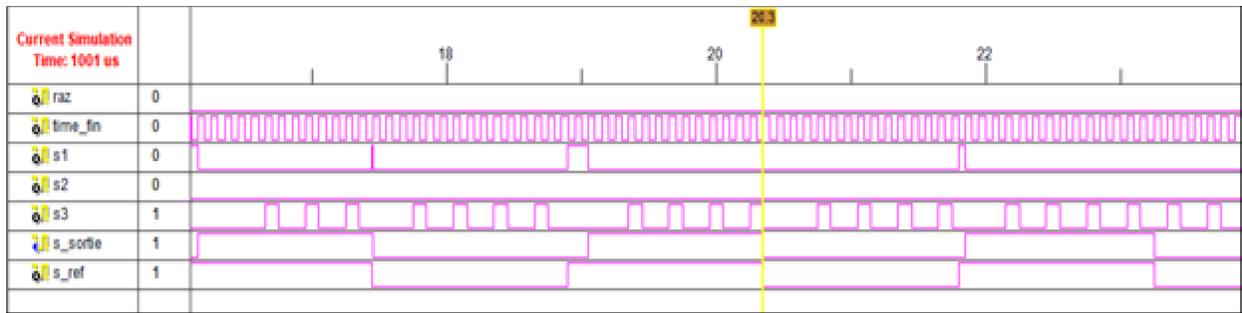
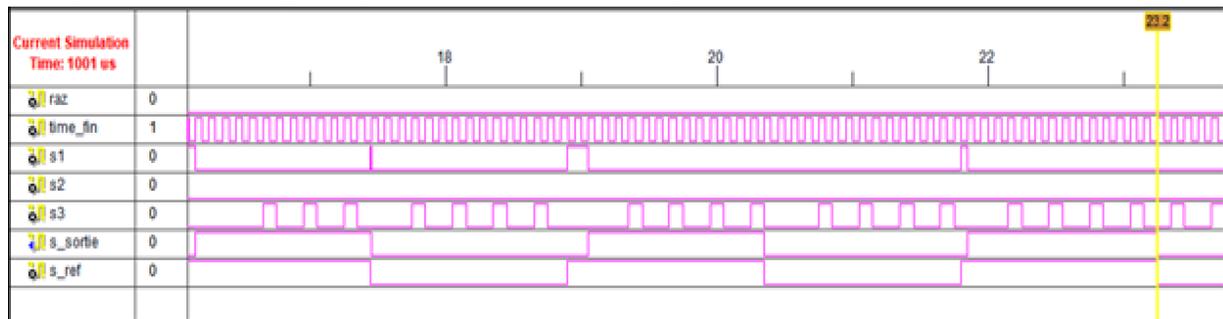


Figure III.18. La simulation fonctionnelle de la BVPN.

Après quelques périodes de l’horloge (time_fin), nous observons que la période du signal s_sortie s’approche à la période du signal s_ref dans plusieurs instants, et cela montre que la boucle vérifie le principe de la BVPN($f_s=f_e$).



(a)



(b)

Figure III.19 (a) et (b). La détection des points de verrouillage de la BVPN.

S1 est le signal de sortie de comparateur de phase s_exor, s2 est le signal avance, s3 est le signal retard .

D’après les figures (a) et (b) de III.19, nous observons deux points de verrouillage de la BVPN à deux instants différents ($t_0 = 20,3 \text{ us}$ et $t_1 = 23,2 \text{ us}$), dont la fréquence de référence et de la sortie ont la même période tous en vérifiant la caractéristique de la BVPN ($f_s=f_e$). En plus, nous remarquons que les signaux sont en phase.

III.6 L’implémentation matérielle de la BVPN

Pour réaliser notre circuit BVPN, nous avons décidé d’implémenter ce dernier sur un circuit reconfigurable FPGA. Le cahier des charges du projet consiste à concevoir puis implémenter une architecture matérielle sur une carte numérique à base de FPGA. Pour cela, nous avons besoin de l’environnement suivant :

III.6.1. Présentation de l’environnement

L’implémentation de notre circuit sur un circuit FPGA afin de s’assurer du bon fonctionnement de la BVPN nécessite généralement d’un logiciel de conception, d’une carte de développement qui contient le circuit FPGA, et un micro-ordinateur hôte pour la gestion de l’ensemble. Particulièrement, nous avons

besoin aussi d'un générateur de fonction et d'un oscilloscope. L'environnement préparé pour l'implémentation (Figure III.22) est composé :

- D'un micro-ordinateur doté du logiciel de développement (ISE 14.7 de XILINX), permettant la conception de l'architecture matérielle de la BVPN et de son implantation sur le FPGA.
- Une carte de développement (MIMAS V2) qui contient le circuit reconfigurable FPGA (SPARTAN6).
- Un câble USB pour connecter notre PC avec la carte de développement.
- Un GBF pour générer l'entrée du circuit (la fréquence de référence).
- Un oscilloscope pour visualiser la sortie du circuit (la fréquence de sortie).

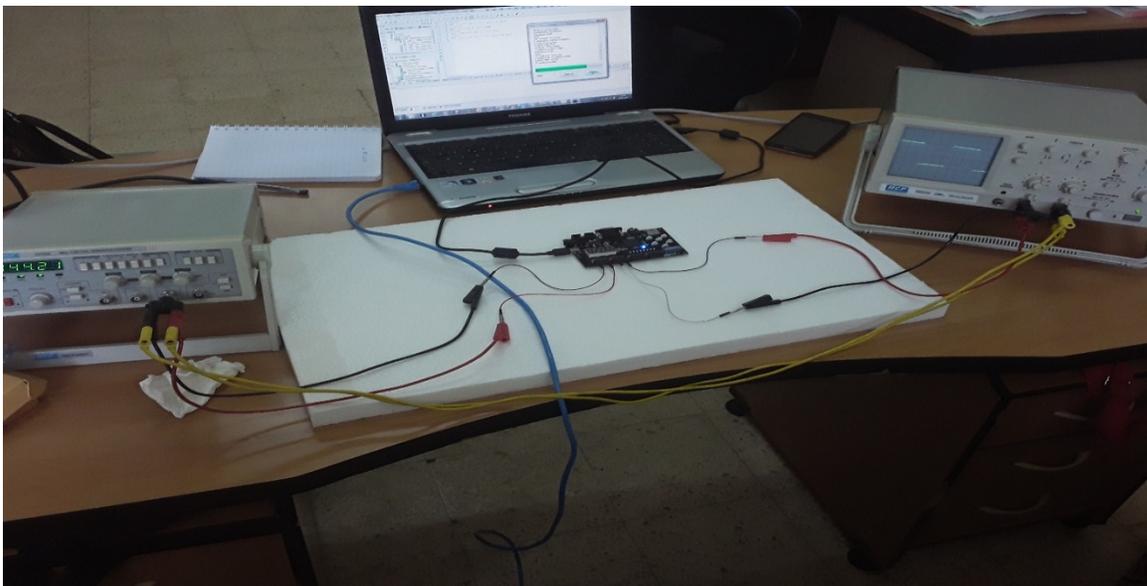


Figure III.22. L'environnement de travail.

a. Le logiciel de développement

Le compilateur ISE 14.7 a permis la gestion complète du flot de conception et l'implémentation du circuit BVPN sur le circuit FPGA. L'interface graphique d'ISE 14.7 est présentée par la figure ci-dessous (figure III.23).

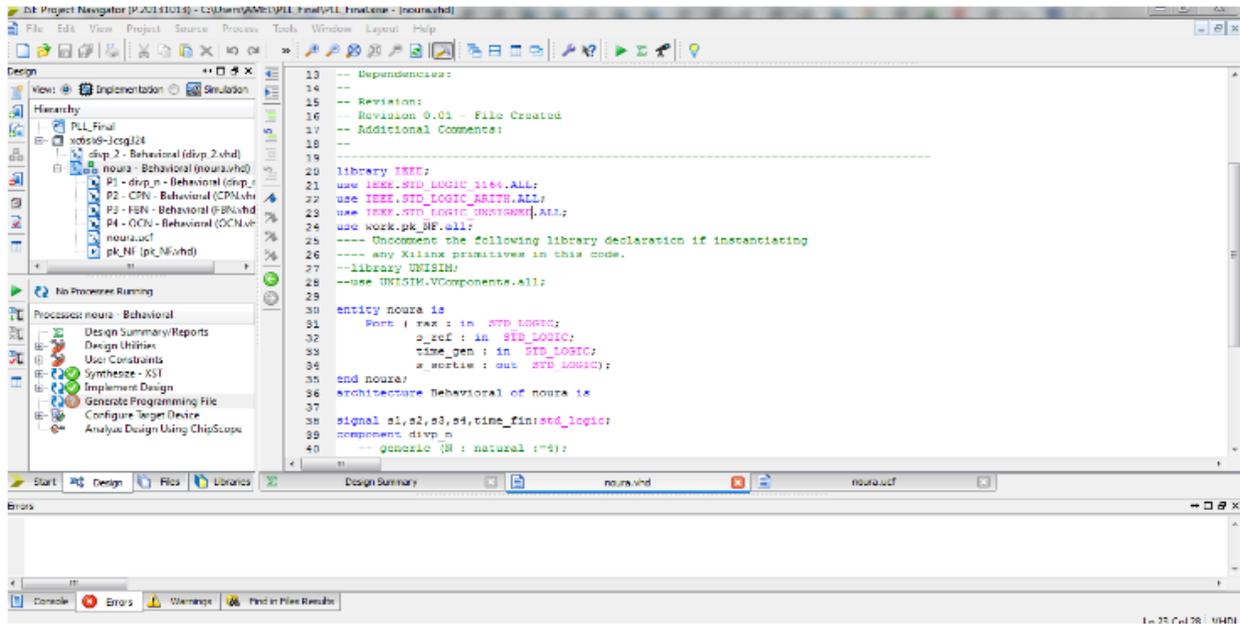


Figure III.23. Environnement de développement ISE 14.7

b. La carte de développement Mimas V2 de Xilinx

Pour notre projet, nous avons utilisé la carte de développement « MIMAS V2 » (figure III.20), cette dernière permet l'implémentation de notre circuit matériellement.



Figure III.24. La carte de développement MIMAS V2

« MIMAS V2 » est une carte de développement FPGA riche en fonctionnalités et à prix réduit, elle est spécialement conçue pour expérimenter et apprendre la conception de systèmes avec des FPGA, cette carte de développement comprend un FPGA SPARTAN XC6SLX9 CSG324 avec SDRAM DDR 512

Mo intégrée. L'interface USB 2.0 permet un téléchargement rapide et facile de la configuration (le fichier *.bin) sur la mémoire flash SPI intégrée. La carte est composée de[27]:

- FPGA : Spartan XC6SLX9 dans le package CSG324.
- Mémoire DDR : 166 MHz, 512 Mo LPDDR (MT46H32M16LF / W949D6CBHX6E).
- Mémoire flash : Mémoire flash de 16 Mo SPI (M25P16).
- Interface USB 2.0 pour la programmation du mémoire flash intégrée.
- Configuration du FPGA via JTAG et USB.
- Connecteur VGA.
- 6 boutons poussoirs.
- Adaptateur de carte micro SD 3.
- 3 Afficheurs sept segments.
- 32 pins d'entrées / sorties définies par l'utilisateur.
- Régulateurs de tension intégrés pour un fonctionnement sur un seul support d'alimentation

✓ **Le câble USB**

Ce câble permet la liaison entre le micro-ordinateur et la carte de développement (figure III.25).



Figure III.25. Câble USB 2.0

✓ **Le générateur de fréquence GBF**

Nous avons utilisé un GBF afin de générer le signal d'entrée (s_ref) de notre circuit BVPN.

✓ **L'oscilloscope**

Cet appareil est utilisé pour visualiser le signal de sortie (s_sortie) et même notre signal d'entrée (s_ref), afin de distinguer les points de verrouillages de la BVPN.

III.6.2. Implémentation de la BVPN

Les paramètres d'entrée de notre BVPN sont :

- Le signal (raz) : est utilisé pour la remise à zéro du circuit en appuyant sur un bouton poussoir de la carte de développement MIMAS V2.
- Le signal (s_ref) : est un signal de fréquence dans une plage de fréquence de (200 à 344 KHZ), qui sera délivré par le GBF.Ce dernier doit être connecté avec la carte de développement MIMAS V2 à travers des pins.
- Le signal (time_fin) : est utilisé pour le fonctionnement des deux blocs FBN et OCN, il est d'une fréquence de 10 MHz, cette dernière sera émise par l'horloge globale du circuit FPGA (100 MHz).

Nous avons qu'un seul paramètre de sortie de la BVPN, le signal (s_sortie) sera connecté avec l'oscilloscope. Ce dernier doit être connecté avec la carte de développement MIMAS V2 à travers des pins.

Comme nous avons présenté ci-dessus, la fréquence entrante dans les deux blocs FBN et OCN est de 10 MHz et la fréquence du circuit FPGA est de 100 MHz (time_gene), nous avons besoin d'un diviseur de fréquence par 10 (figure III.26).

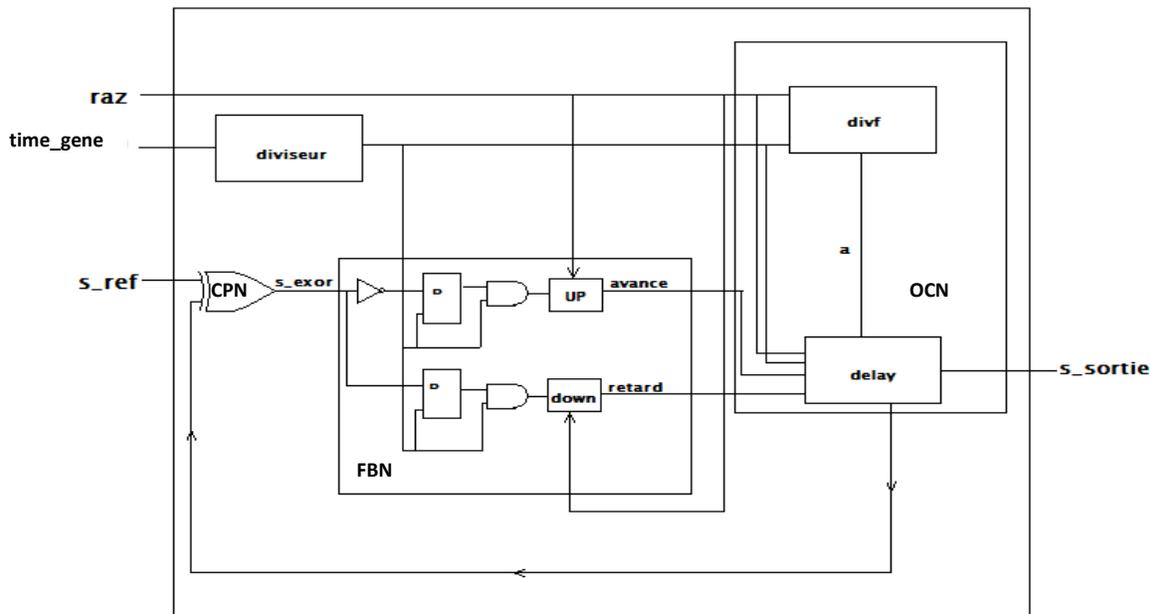


Figure III.26. L'architecture matérielle globale de la BVPN.

La figure III.27 présente la simulation fonctionnelle de la BVPN après l'ajoute d'un bloc de diviseur de fréquence.

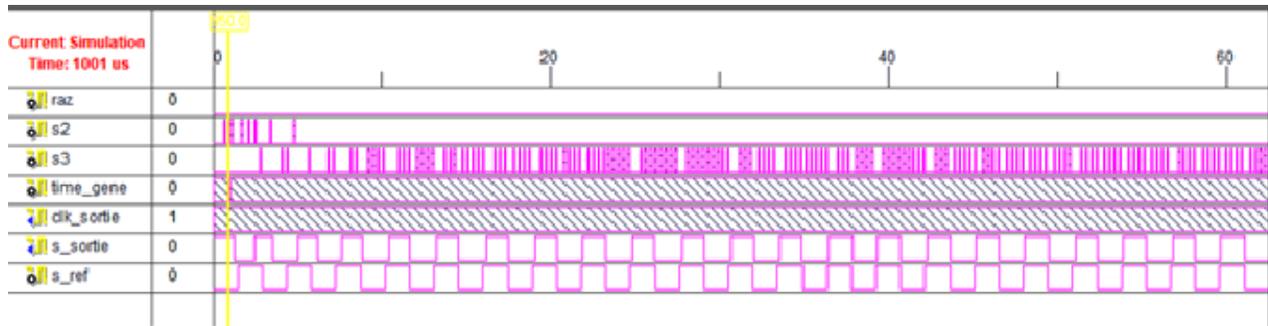


Figure III.27. La simulation de la BVPN avec le diviseur de fréquence.

Pour une réalisation physique, l'architecture doit être menée sur le dispositif FPGA, les signaux d'entrées-sorties doivent affectés aux différentes broches de ce dernier. Ceci peut être fait, avant la synthèse, en éditant un fichier de contraintes (*.ucf : *User Constraints File*) ou bien graphiquement à l'aide de l'un des outils (*PACE*) du logiciel ISE (Voir Annexe). Le fichier de contraintes nécessaires pour notre projet est présenté en ci-dessous.

```

-----
-----
# Switch
NET "raz" LOC = "L18"; #SW2

#HPin6
NET "s_ref" LOC = "U7"; #pin1 de Hpin6

#Clock
NET "time_gen" LOC = "V10"; #Clk

#HPin7
NET "s_sortie" LOC = "U8"; #pin2 de Hpin7
-----
-----

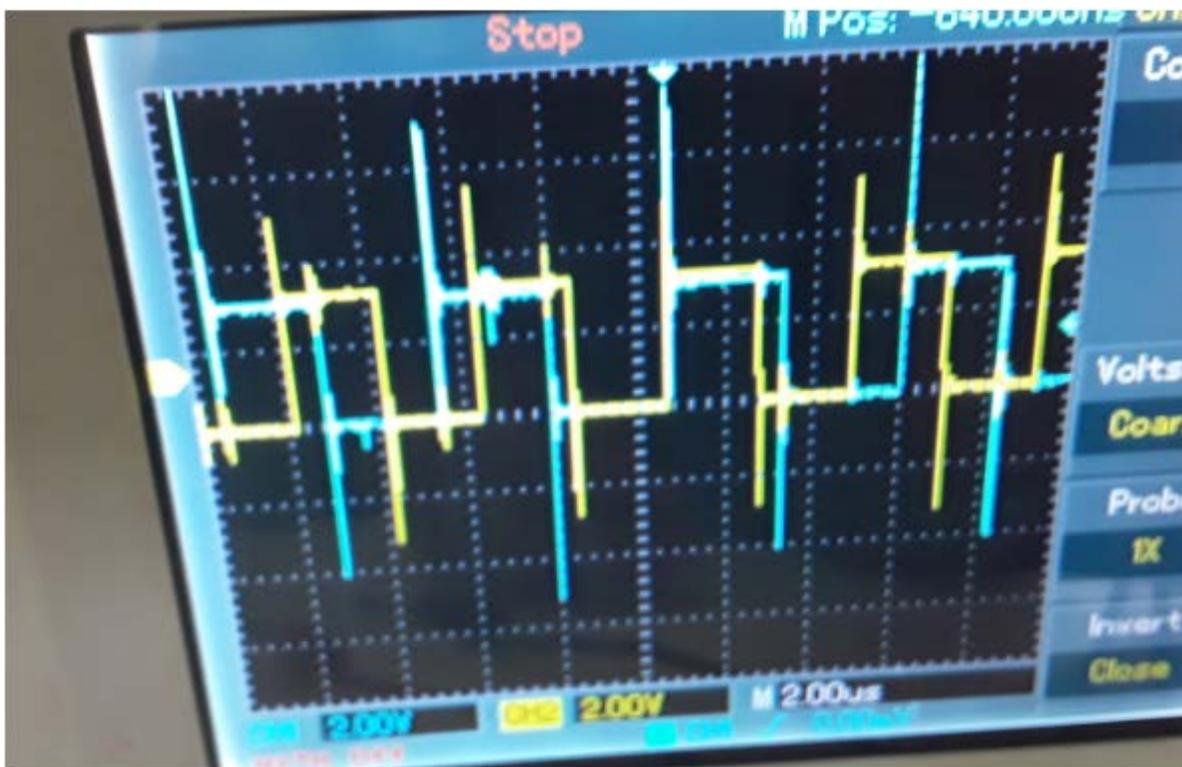
```

L'étape suivante dans le processus est de générer le fichier de configuration (*.bin) en utilisant l'outil « *Generate Programming File* » du logiciel ISE 14.7 de XILINX. Ce fichier devra être téléchargé vers le composant reconfigurable « FPGA » en utilisant le câble USB2.

Enfin, nous avons établi la conception, la synthèse et l'implémentation de notre circuit BVPN sur la carte MIMAS V2 à base du composants reconfigurable FPGA SPARTAN XC6SLX9 CSG324Le résultat est visualisé sur l'oscilloscope (figure III.28), dont le signal en couleur bleu est le signal de sortie (s_sortie), en couleur jaune est le signal de référence (s_ref) de notre circuit BVPN.



(a)



(b)

Figure III.26 (a) et (b). La visualisation des signaux s_refet s_sortie de la BVPN sur l'oscilloscope.

III.7. Conclusion

Enfin, nous avons réussi à décrire notre boucle à verrouillage de phase numérique (BVPN), composée d'un comparateur de phase numérique (CPN), d'un filtre de boucle numérique (FBN) et d'un oscillateur de commande numérique (OCN) en VHDL, comme nous avons réussi à les simuler bloc par bloc, en utilisant le compilateur ISE 14.7. Cette première étape nous a permis de vérifier le bon fonctionnement de notre circuit avant de l'implémenter où nous avons obtenus des bons résultats de simulation.

La deuxième étape concerne l'implémentation, dont nous avons pu implémenter notre BVPN matériellement sur le circuit programmable FPGA. En commençant par la configuration des entrées et des sorties du circuit sur les blocs d'entrée/sortie programmables du FPGA. Donc l'implémentation de notre BVPN a été bien effectuée sur le circuit FPGA, en arrivant à détecter les points de verrouillage de la boucle.

CONCLUSION
GENERALE

CONCLUSION GÉNÉRALE

Dans ce travail, nous avons présenté la boucle à verrouillage de phase, qui asservit la fréquence d'un oscillateur commandé en tension VCO à un signal injecté à l'entrée ; nous trouvons que le cœur de la PLL est le VCO fournit en sortie un signal sinusoïdal ou carré, dont la fréquence instantanée de sortie dépend de la tension de commande, qui sort d'un filtre passe-bas. Le comparateur de phase élabore une tension dépendant du déphasage entre les signaux d'entrée. La moyenne de tension de sortie commande le VCO.

Nous avons présenté une étude théorique des différents types de PLL. Pour renforcer notre analyse, nous avons exploité le logiciel de simulation des synthétiseurs de fréquences ADsimPLL, afin d'étudier les réponses temporelles et fréquentielles de ce système. L'exploitation de ce logiciel est faite en concevant un système de synthétiseur de fréquence.

En dernier, nous avons décrit le circuit de la PLL numérique matériellement en utilisant le langage VHDL et le compilateur ISE 14.7 de la société Xilinx. Ensuite, nous avons pu implémenter ce circuit sur la carte de développement « Mimas V2 » qui contient le circuit programmable FPGA de type Spartan 6 de la société Xilinx. Les résultats des signaux de sorties ont été vérifiés par mesure réelle sur un oscilloscope pour une bande de fréquence de 10 MHz.

Ce travail n'est qu'une plateforme de départ pour d'autre projet d'aspect pratique et qui ont un impact socio-économique direct. A partir de ce prototype réalisé, nous pouvons concevoir un autre prototype mais avec réinitialisation synchrone. Il est aussi favorable d'implémenter un synthétiseur de fréquence, dédié aux systèmes de télécommunications, dont la fréquence est conditionnée par la bande passante des systèmes utilisés. Un autre défi qu'on doit faire face, qui est l'utilisation d'une partie du circuit analogique. Pour cela, il faut se disposer des circuits FPGA spécialisés mixtes, et dont le langage de description matérielle sera différent du VHDL comme le VHDL-AMS.

ANNEXE

ANNEXE

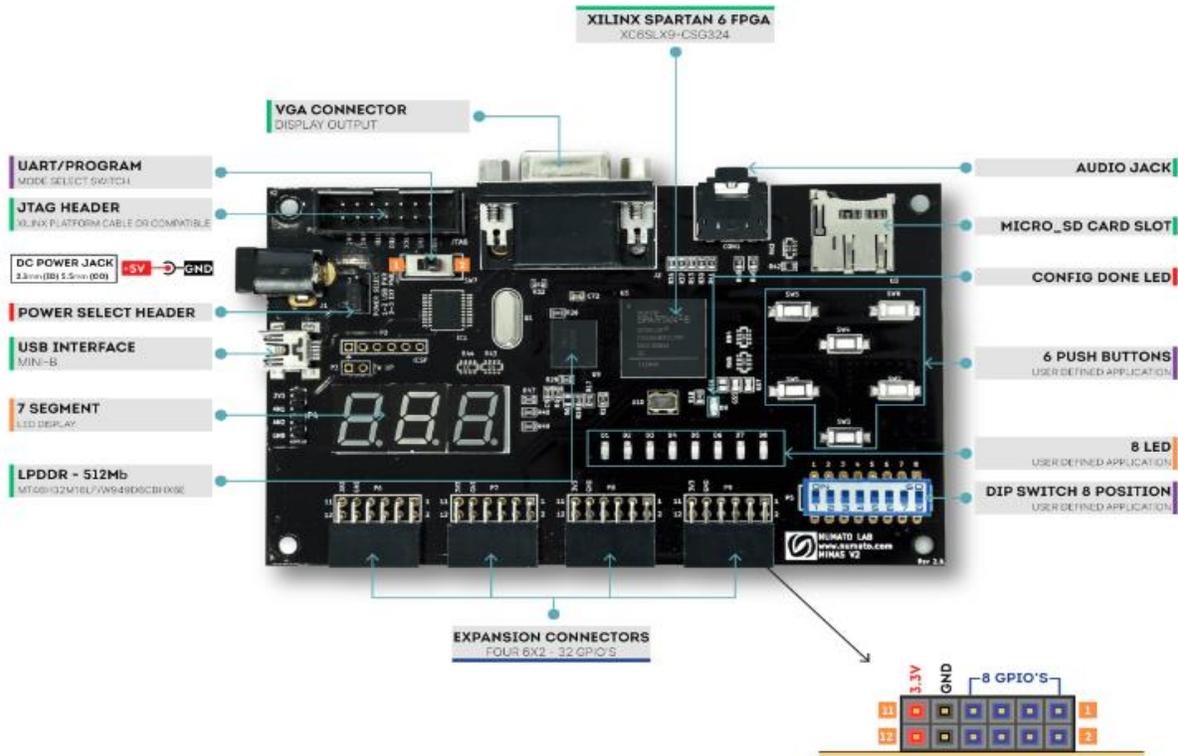


Figure A.1. La carte de développement MIMAS V2

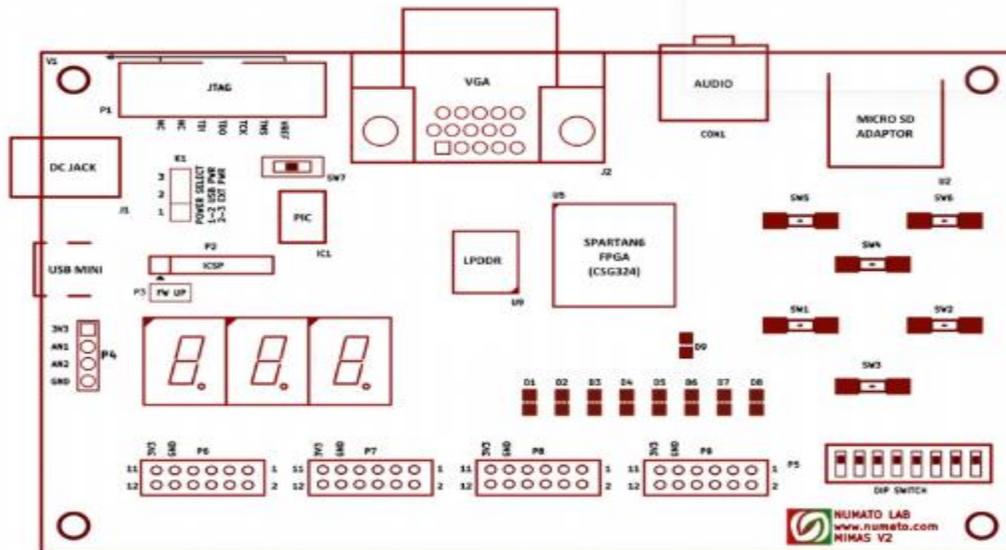


Figure A. 2. Architecture globale de MIMAS V2.

ANNEXE

❖ Composants / outils requis

En utilisant la carte, nous avons besoin d'un élément présenté ci-dessous pour une installation facile et rapide du fichier de configuration. C'est le câble USB A à Mini B. 2.

1. Interface USB



Figure A.3. Interface USB

❖ Sélection du mode de configuration

Le commutateur à glisser SW7 permet de basculer entre deux modes de configuration USB ou le mode UART. Pour notre projet, Nous choisissons la position une du commutateur SW7 pour télécharger le fichier(.bit) via l'outil de configuration USB.



Figure A.4. Le commutateur à glissière SW7

❖ GPIOs

Cette carte est équipée de 32 broches IO d'utilisateur pouvant être utilisées pour diverses applications personnalisées. Les affectations des broches sur les connecteurs sont disponibles dans les tableaux ci-dessous. Les broches utilisées dans notre projet ont été pris des HEADER P6 et HEADER P7.

ANNEXE

HEADER P6

Header Pin No.	Pin description	Spartan-6 (CSG324) Pin No.
1	IO_L43P_2	U7
2	IO_L43N_2	V7
3	IO_L63P_2	T4
4	IO_L63N_2	V4
5	IO_L49P_D3_2	U5
6	IO_L49N_D4_2	V5
7	IO_L62P_D5_2	R3
8	IO_L62N_D6_2	T3
9	GND	NA
10	GND	NA

ANNEXE

Header Pin No.	Pin description	Spartan-6 (CSG324) Pin No.
11	VCCAUX	NA
12	VCCAUX	NA

HEADER P7

Header Pin No.	Pin description	Spartan-6 (CSG324) Pin No.
1	IO_L41P_2	U8
2	IO_L41N_VREF_2	V8
3	IO_L31P_GCLK31 _D14_2	R8
4	IO_L31N_GCLK3 0_D15_2	T8
5	IO_L48P_D7_2	R5
6	IO_L48N_RDWR_ B_VREF_2	T5
7	IO_L32P_GCLK29	T9

ANNEXE

Header Pin No.	Pin description	Spartan-6 (CSG324) Pin No.
	_2	
8	IO_L32N_GCLK2 8_2	V9
9	GND	NA
10	GND	NA
11	VCCAUX	NA
12	VCCAUX	NA

❖ Installation du pilote

• Installation sous Windows

Ce produit nécessite l'installation d'un pilote pour fonctionner correctement sous Windows. Le package de pilote peut être téléchargé à partir de la page du produit. Pour l'installation du pilote, nous décompressons le contenu du package de pilote téléchargé dans un dossier. Nous avons connecté un câble USB au PC et, à la demande de l'assistant d'installation de périphérique Windows, puis pointé sur le dossier contenant les fichiers du pilote. Lorsque l'installation du pilote est terminée, le module doit apparaître dans le Gestionnaire de périphériques Windows en tant que port série (figure A.5). Le nom du port série (COM1, COM2 etc.). Ces informations sont nécessaires lors de la programmation du module avec l'outil de configuration.

ANNEXE

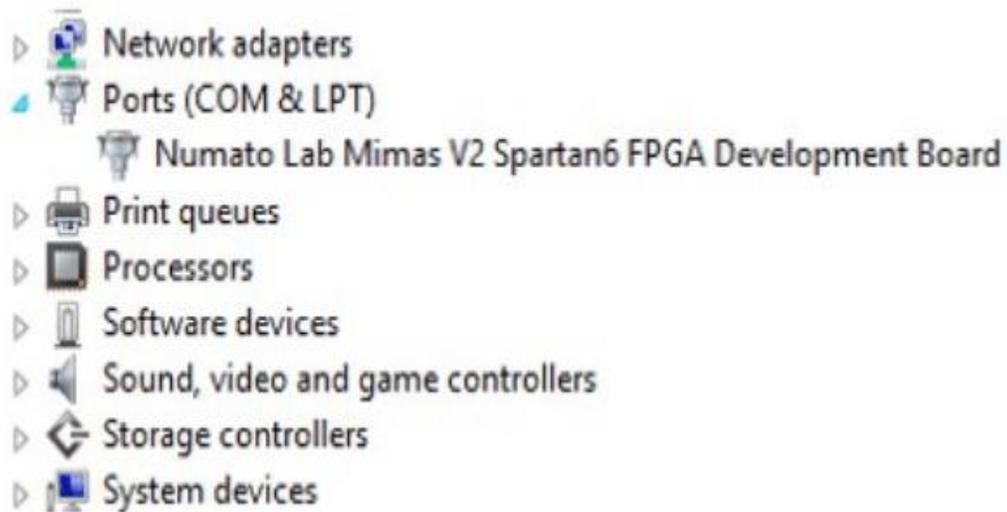


Figure A.5. L'installation de la carte MIMAS V2

- **Génération de fichier de bits pour MIMAS**

La conception HDL doit être convertie en fichier (.bit) avant de pouvoir être programmée en FPGA. L'outil de configuration MIMAS V2 n'accepte actuellement que le fichier binaire (.bin) créé par XILINX ISE. Une fois le HDL synthétisé, il est facile de créer un fichier binaire. Nous avons suivi les étapes ci-dessous pour générer un fichier binaire à partir de la conception en utilisant ISE Web Pack.

Étape 1: En cliquant avec le bouton droit de la souris sur l'option «*Generate Programming File*» dans la fenêtre «*Process*».

ANNEXE

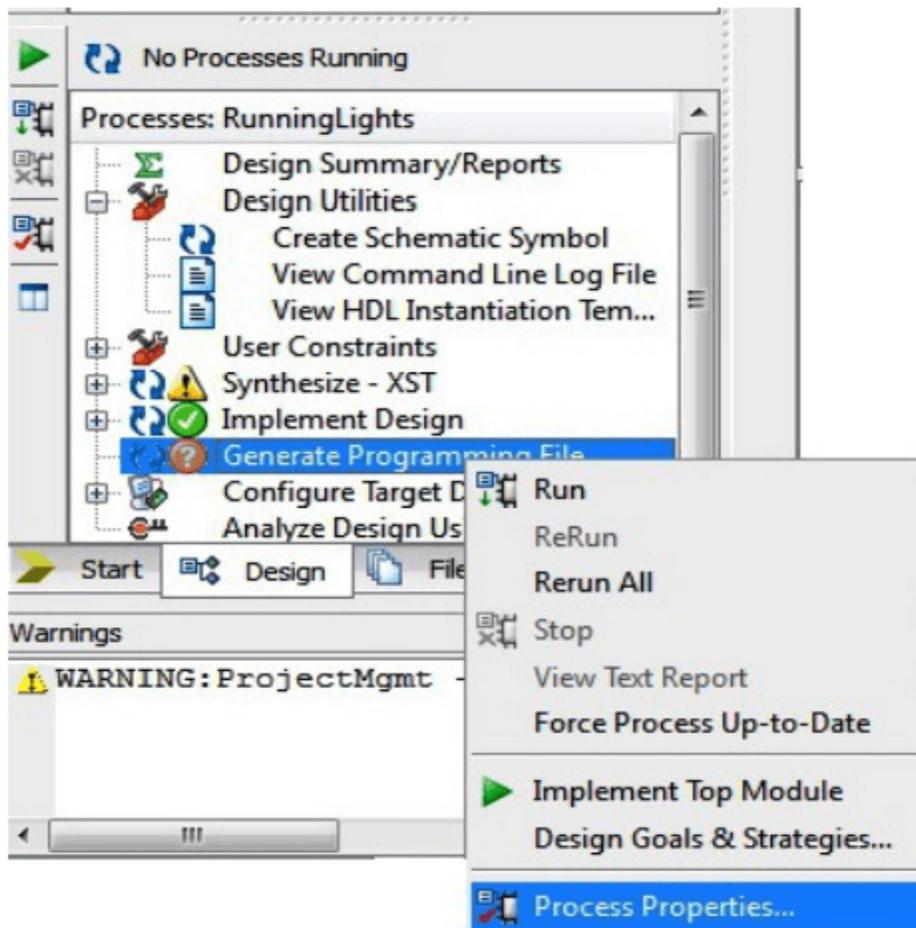


Figure A.6. Génération du fichier binaire (1).

Étape 2 : En Sélectionnant «*Process Properties*» dans le menu . Dans la boîte de dialogue, on coche la case «*Create Binary Configuration File*» et on clique sur «*Apply*».

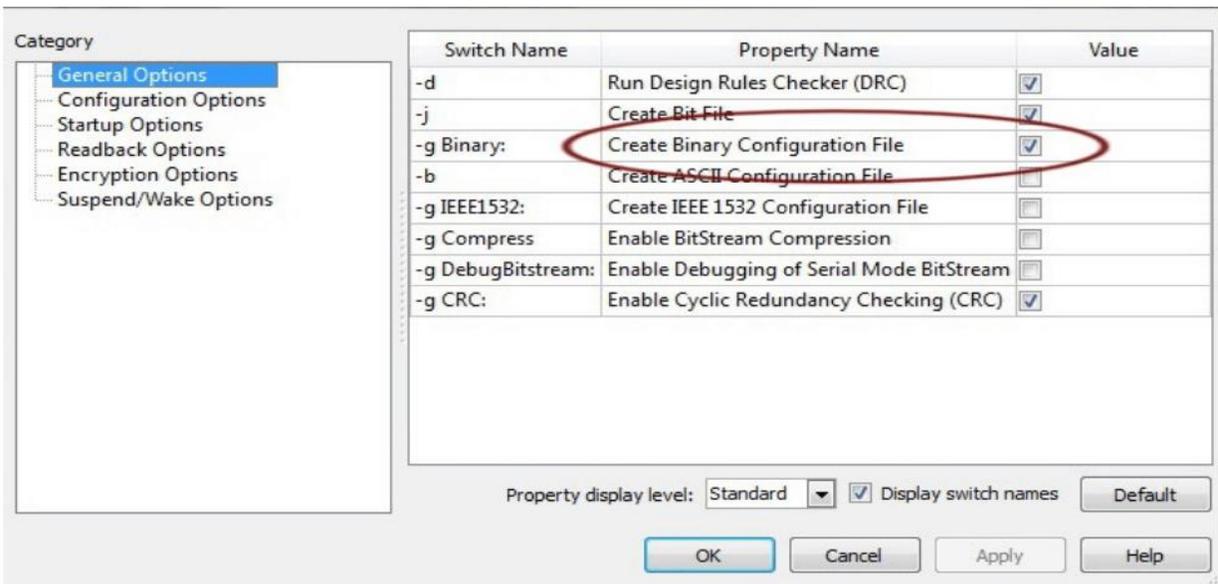


Figure A.7. Génération du fichier binaire (2).

Étape 3: En cliquant sur «OK» pour fermer la boîte de dialogue. Nous cliquons à nouveau avec le bouton droit de la souris sur l'option «*Generate Programming File*» et on sélectionne «*Run*». Nous pourrions maintenant trouver un fichier .bin dans le répertoire du projet et ce fichier pourra être utilisé pour la configuration de MIMAS V2.

Configuration de MIMAS V2

Le module MIMAS V2 Spartan6 peut être configuré selon deux méthodes:

- Utilisation de l'outil de configuration MIMAS V2 via USB.
- Utilisation du câble de programmation Xilinx.

❖ Configuration de MIMAS V2 à l'aide de l'outil de configuration

MIMAS V2 est doté d'un microcontrôleur intégré facilitant la reprogrammation de la mémoire flash SPI intégrée via une interface USB. Le microcontrôleur reçoit le fichier de bits de l'application hôte et le programme dans le flash SPI et permet au FPGA de démarrer à partir du flash. Lorsque MIMAS V2 est connecté à un PC, il apparaît comme un port COM dans le Gestionnaire de périphériques. En Exécutant l'application de configuration, et sélectionnant le port COM approprié avant de télécharger le fichier binaire. En cliquant sur «*Open File*» pour sélectionner le fichier de fichier de bits (.bin) et en appuyant sur le bouton «*Program*» pour télécharger le flux de bits. En attendant que le téléchargement

ANNEXE

soit terminé. Une fois que le processus de téléchargement est terminé, le contrôleur de configuration essaiera de démarrer automatiquement le FPGA à partir du flash SPI. Suivant les étapes ci-dessous.

Étape 1: En Vérifiant que nous avons sélectionné le mode de configuration USB (réglant SW7)sur la position 1.nous reportons la section «*Configuration Mode Selection*».En exécutant l'outil de configuration MIMAS V2 et sélectionnant le bon port nous reportons à la section «*Driver installation*». En cliquant sur le bouton Ouvrir le fichier et sélectionnant le fichier .bin

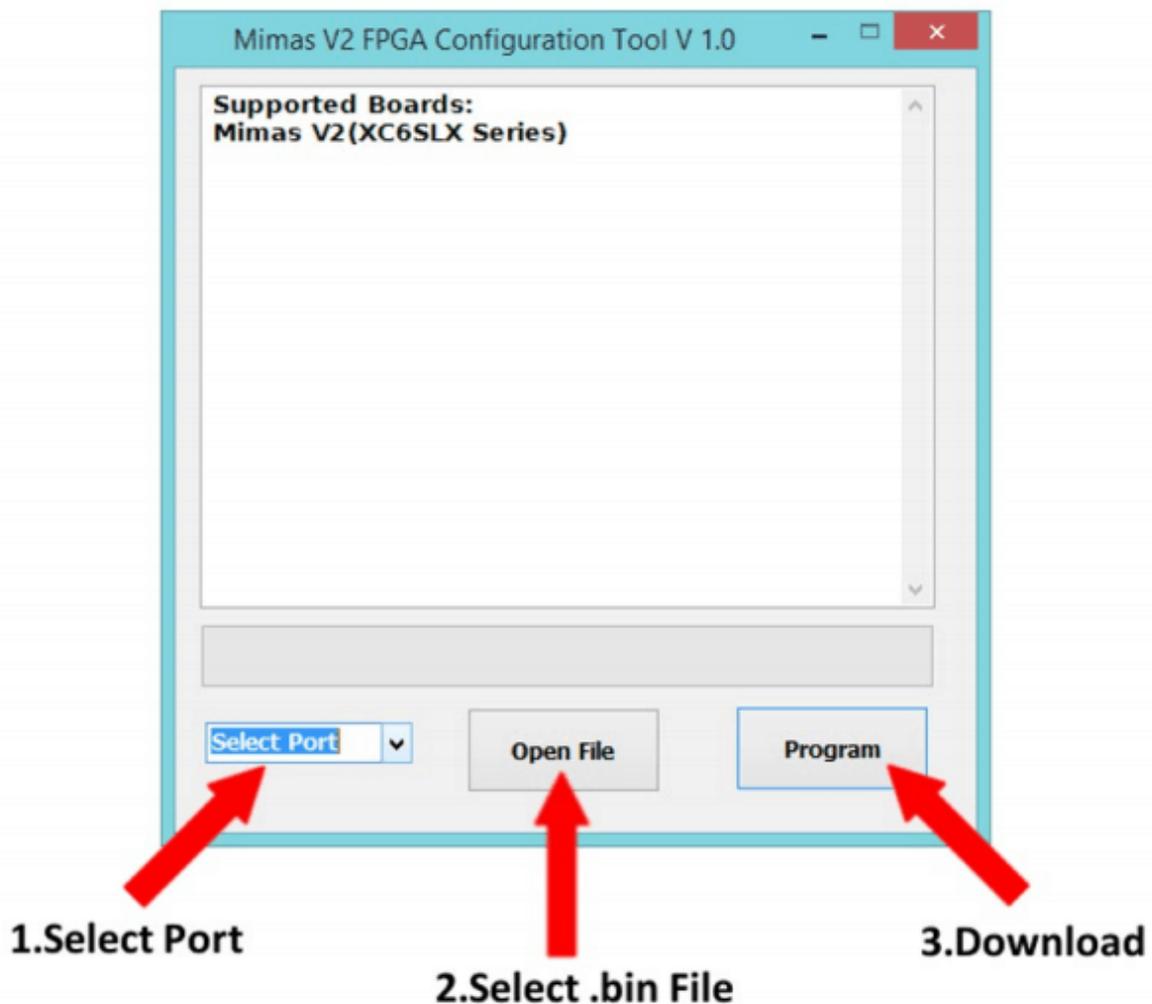


Figure A.8.La sélection du port et du fichier bin

Étape 2: En cliquant sur le bouton “*Program*”, attendant que “*Done*” apparaisse à l’écran.

ANNEXE

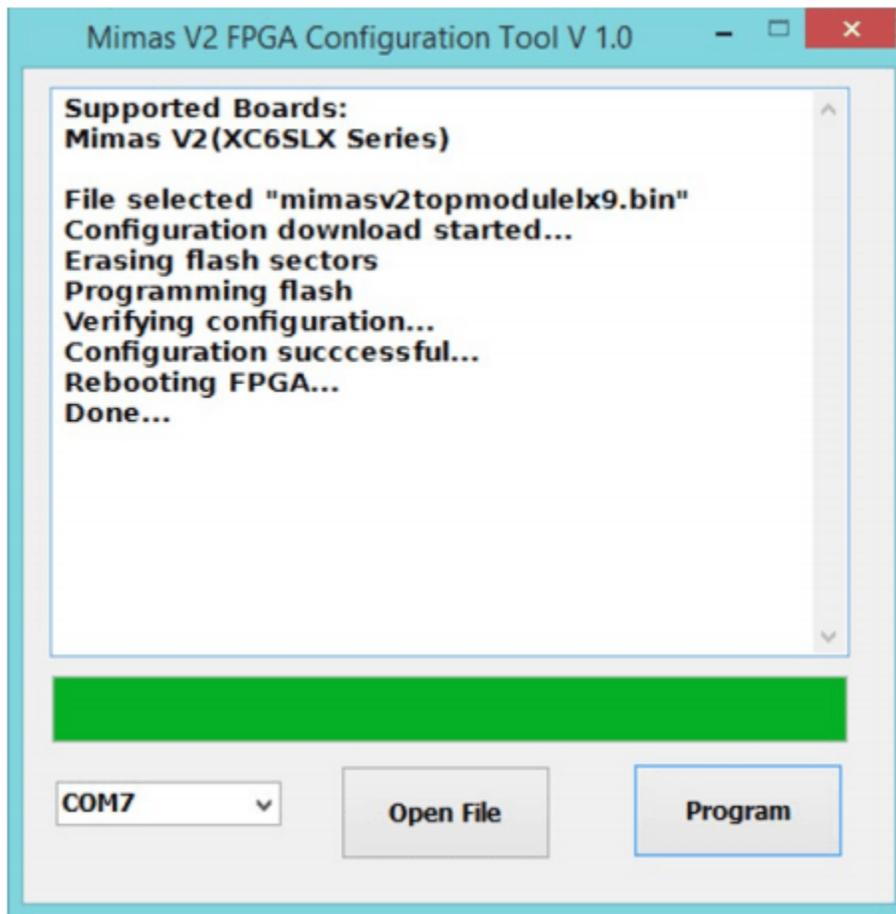


Figure A.9. La programmation du fichier sur FPGA.

*REFERENCE &
BIBLIOGRAPHIE*

Référence & Bibliographie

- [1] Roland E. Best, « Phase Locked Loops Design Simulation and Applications », McGraw-Hill, 5th Edition, « Boucle à verrouillage de phase », <http://www.ta-formation.com/acrobat-modules/pll.pdf>.
- [2] HIERRY ROCACHER « Résumé de cours sur la PLL & la synthèse de fréquence » https://www.leselectroniciens.com/sites/default/files/cours/pll_synthese_de_frequence.pdf.
- [3] Julien KIEFFER ,Optique et Radiofréquence,thèse de doctorat,«Contribution au dimensionnement des PLL pour des modulations polaires larges bandes », août 2006.
- [4] M. Correvon« Systèmes électroniques, BOUCLES A VERROUILLAGE DE PHASE PHASE-LOCKED LOOPS (PLL) »,<https://www.leselectroniciens.com/sites/default/>
- [5] JeanPhilippeMuller « La_boucle_à_verrouillage_de_phase », https://www.academia.edu/36786557/Physique_appliquée
- [6]wikipedia ,« boucle à phase asservie »,https://fr.wikipedia.org/wiki/Boucle_%C3%A0_phase_asservie
- [7] Vallabhaneni^a, Dr. Sanjay Attri^b, N. Krishnan, Sanjay Sharma, R. C. Chauhan^c ,” DESIGN OF AN ALL-DIGITAL PLL (ADPLL) CORE ON FPGA Sandeep”https://www.researchgate.net/publication/322001805_Design_and_implementation , April 2017.
- [8] Vincent GIORDANO, Enrico RUBIOLA, " Synthèse de fréquence", <https://www.techniques-ingenieur.fr/>,10 nov. 2002.
- [9] MAZOUFFRE Olivier "Conception de synthèses de fréquences à 24 GHz" ,<https://www.theses.fr/>,2008.
- [10] Samir KAMECHE "Conception et optimisation d'un synthétiseur de fréquence pour les communications mobiles sur GSM" ,2008

Référence & Bibliographie

- [11] J. JUYON, "Contribution à la conception de synthèses de fréquence pour liaison satellite embarquée: montée en résolution et réduction de raies parasites ", *Thèse de Doctorat*, Université Toulouse, décembre 2013.
- [12] <https://hal.archives-ouvertes.fr/>
- [13] "ADIsimPLL REQUEST FOR SOFTWARE " <https://form.analog.com/>
- [14] Analog Devices, "PLL Frequency Synthesizer", Data Sheet ADF4107, WWW.analog.com.
- [15] wikipedia "circuit logique programmable" <https://fr.wikipedia.org/>
- [16] Catherine Douillard, Gérald Ouvradou, Michel Jezequel "Logique séquentielle Techniques d'intégration", (2005).
- [17] Litayem Nabil, « Architecture interne d'un circuit FPGA » <https://www.researchgate.net/figure/Architecture-interne-dun-circuitFPGA>.
- [18] Laurent Dutrieux, Didier Demigny "Logique Programmable, Architecture des FPGA et CPLD", (1997).
- [19] François Gauthier "Le Marché des FPGA pourrait croître de 8.5 % par an en 2019 pour atteindre les 9 MD\$", 2013.
- [20] BABA HAMED Amel "Vers Un Modèle De Classification Neuronale Des Données Médicales A Base De La Technologie FPGA", 2017
- [21] Johanna Mariani "Programmation et Utilisation du FPGA pour la validation et la vérification de circuits électroniques", 2011.
- [22] T. BLOTIN "Le Langage De Description VHDL", (2005).
- [23] wikipedia "cartes auto adaptatives" <https://fr.wikipedia.org/>.
- [24] siMahmoud KARABERNOU "Méthodologie De Conception Et Langages De Description De Matériel", Septembre 2009.
- [25] "Introduction au logiciel Xilinx ISE 9.2i" <https://docplayer.fr/>
- [26] KRISHNA KUMAR "design and simulation of power efficient all digital phase locked loops(adpll)", mai 2017

ملخص

في عام 1932، اخترع De Bellescize حلقات قفل الطور PLL لإنشاء كشف متزامن. اكتسب هذا التدخل زخماً مع ظهور الدوائر المتكاملة وتطويرها. في الواقع، أصبحت PLL أنظمة أساسية في جميع التخصصات: الاتصالات، والأوتوماتيكية، والبيوطبية والطيران... وفقاً للتطبيقات، نجد PLL تماثلية أو رقمية أو هجينة (الاثين معا) في هذا العمل، تم تقديم دراسة نظرية لـ PLL. بعد ذلك، من خلال الإجابة على أحد المواصفات، قمنا بمحاكاة PLL على برنامج ADIsimPLL، أثناء تصميم دائرتها الإلكترونية ودراساتها على مختلف الاستجابات الزمنية والترددية وتأثير الضوضاء. من أجل أن تكون قادرة على تنفيذ هذه الدائرة على FPGA، وصفناها في لغة VHDL. يتم التحقق من كفاءة وأداء دارة PLL لدينا من خلال القياسات الإلكترونية المختلفة في مختبر القياس.

الكلمات المفتاحية:

PLL, FPGA, BVPN, synthèse de fréquence, VHDL, les outils de CAO.

Résumé

En 1932, De Bellescize a inventé les boucles de verrouillage de phase PLL pour établir la detection synchrone. Cette intervention a pris l'ampleur avec l'apparition et le développement des circuits intégrés. En effet, les PLL sont devenus des systèmes essentiels dans toutes les disciplines : télécommunications, automatiques, biomédicales et aéronautiques... Selon les applications, nous trouvons des PLL analogiques, numériques ou mixte.

Dans ce travail, une étude théorique des PLL est présenté. Ensuite, en répondant à un cahier des charges nous avons simulé la PLL sur le logiciel ADIsimPLL, tout en concevant son circuit électronique et des études sur les différentes réponses temporelles, fréquentielles et l'influence du bruit. Afin de pouvoir implémenter ce circuit sur un FPGA, nous l'avons décrit en langage VHDL. L'efficacité et les performances de notre circuit PLL sont vérifiés par les différentes mesures électroniques au laboratoire de mesure.

Mots clés : PLL, FPGA, BVPN, synthèse de fréquence, VHDL, les outils de CAO.

Abstarct

In 1932, De Bellescize invented PLL phase lock loops to establish synchronous detection. This intervention has gained momentum with the appearance and development of integrated circuits. Indeed, PLL have become essential systems in all disciplines: telecommunications, automation, biomedical and aeronautical... Depending on the applications, we find PLL analog, digital or mixed.

In this work, a theoretical study of PLL is presented. Then, by answering a specification we simulated the PLL on the ADIsimPLL software, while designing its electronic circuit and studies on the different temporal, frequency responses and the influence of the noise. In order to be able to implement this circuit on an FPGA, we have described it in VHDL language. The efficiency and performance of our PLL circuit are verified by the various electronic measurements at the measuring laboratory.

Key words: PLL, FPGA, BVPN, frequency synthesizer, VHDL.