



**HAL**  
open science

# Modélisation et optimisation de la consommation énergétique d'un système de communication Wi-Fi

Wissem Benali

► **To cite this version:**

Wissem Benali. Modélisation et optimisation de la consommation énergétique d'un système de communication Wi-Fi. Electronique. Ecole nationale supérieure Mines-Télécom Atlantique, 2017. Français. NNT : 2017IMTA0041 . tel-01781321

**HAL Id: tel-01781321**

**<https://theses.hal.science/tel-01781321>**

Submitted on 30 Apr 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**IMT Atlantique**  
Bretagne-Pays de la Loire  
École Mines-Télécom

**UNIVERSITE  
BRETAGNE  
LOIRE**

## **THÈSE / IMT Atlantique**

*sous le sceau de l'Université Bretagne Loire*

pour obtenir le grade de

**DOCTEUR D'IMT Atlantique**

*Spécialité : Télécommunications*

**École Doctorale Mathématiques et STIC**

Présentée par

**Wissem Benali**

Préparée dans le département Electronique

Laboratoire Labsticc

# Modélisation et optimisation de la consommation énergétique d'un système de communication Wi-Fi

**Thèse soutenue le 17 octobre 2017**

devant le jury composé de :

**Christophe Jegou**

Professeur, IMS - Talence / président

**Maryline Hélard**

Professeure, Insa - Rennes / rapporteur

**Olivier Berder**

Professeur, Enssat / Irista - Lannion / rapporteur

**Olivier Muller**

Maître de conférences, Laboratoire Tima - Grenoble / examinateur

**Marie Le Bot**

Ingénieure de recherche, Orange Labs - Cesson Sévigné / examinatrice

**Charlotte Langlais**

Maître de conférences, IMT Atlantique / examinatrice

**Sylvie Kerouedan**

Directrice d'études, IMT Atlantique / directrice de thèse



# Résumé

La forte augmentation du nombre de terminaux connectés ces dernières années et l'utilisation croissante des technologies de communication impactent de manière non négligeable la facture énergétique. Pour contribuer à diminuer la consommation énergétique, nous proposons de comparer les algorithmes de communications numériques, afin de développer l'architecture de transmission la moins énergivore.

Dans cette thèse, la couche physique des standards Wi-Fi IEEE 802.11ac est analysée sous un angle énergétique. La puissance dissipée dans les circuits pour faire fonctionner les algorithmes de traitement de signal est prise en compte en plus de la puissance d'émission d'antenne classique. La méthodologie mise en œuvre inclut à la fois des simulations et des développements sur plateforme matérielle (FPGA), permettant d'obtenir des évaluations de la consommation plus réalistes.

Nous avons dans un premier temps analysé de façon isolée les éléments composant les chaînes de communications numériques. Puis nous avons intégré les périodes d'activité et d'inactivité de chaque élément dans le calcul de la consommation énergétique globale des chaînes. Nous proposons une méthode pratique et efficace d'estimation de la consommation, incluant une base de données issue de simulations, et une analyse théorique des taux d'activité de chaque élément de la chaîne.

Ces résultats permettent d'analyser la répartition de la consommation en puissance des éléments composant les émetteurs et les récepteurs, et de comparer diverses architectures et jeux de paramètres. En particulier, nous avons évalué l'impact de deux architectures de Transformées de Fourier Rapides sur la consommation globale du système.



# Remerciements

Je voudrais tout d'abord remercier mon encadrante d'Orange, Marie Le Bot ainsi que ma directrice de thèse, Sylvie Kerouédan et mon encadrante à l'IMT Atlantique Charlotte Langlais pour m'avoir donné l'opportunité de réaliser cette thèse et de l'avoir dirigée dans de très bonnes conditions. Vos conseils, remarques et avis m'ont été très importants et constructifs. Vos aides respectives, vos conseils et vos encouragements furent précieux tout au long de ces 3 années. J'ai grandement apprécié de travailler avec vous.

Je voulais également remercier Pierre Pénard qui a encadré cette thèse un an avant de partir en retraite. Ton aide et tes remarques durant cette année m'ont énormément apporté.

Je remercie aussi l'ensemble des membres du jury, Maryline Héléard et Olivier Berder pour leur intérêt porté sur mes travaux en acceptant d'être rapporteurs et à Christophe Jégo et Olivier Muller pour l'évaluation de ma thèse. Leurs remarques vont m'être d'une aide précieuse pour la suite.

Un grand merci à Jean-Christophe Rault, mon responsable durant les deux premières années de thèse grâce à qui j'ai pu réaliser mes travaux dans d'excellentes conditions. Je tiens également à remercier Philippe Blusseau, successeur de Jean-Christophe. Merci à vous deux pour vos disponibilités.

Je remercie également l'ensemble de l'équipe CREM, Redieteb, Marie-Hélène, Jean-Claude, Sandrine, Pierre D, Bruno, Isabelle, Christian, Jean-Philippe, Lin ainsi que les thésards Kaoutar, Alexis et les anciens thésards, Naila et Senad. Sans les nombreux échanges que j'ai pu avoir avec vous, ce travail n'aurait pu être accompli.

Le soutien de ma famille a également été un facteur déterminant dans la réussite de cette thèse. Merci à mes parents Chedia (ohmi) et Sassi (baba), à mes sœurs Aida, Jihene et Sonia, à mon frère Mohamed ainsi qu'à ma grand-mère, mes tantes, oncles, cousin et cousine.

Enfin je ne pourrais jamais assez remercier ma femme, Deborah, pour l'encouragement et l'amour qu'elle m'a apportés pendant les moments difficiles. Merci à toi. J'ai une pensée particulière pour mes enfants, Louane, Zeyneb et Mouhammed. Vous êtes ma source de motivations principales, je vous aime énormément.

Merci également, et mes plus sincères excuses, à tous ceux que j'ai oubliés et qui ont participé de près ou de loin à la réussite de ces travaux.

## Remerciements

---

---

# Table des matières

<b>Résumé .....</b>	<b>3</b>
<b>Remerciements .....</b>	<b>5</b>
<b>Table des matières .....</b>	<b>7</b>
<b>Liste des figures.....</b>	<b>11</b>
<b>Liste des tableaux.....</b>	<b>13</b>
<b>Acronymes .....</b>	<b>15</b>
<b>Introduction.....</b>	<b>17</b>
<b>Chapitre 1 Consommation de puissance dans les circuits.....</b>	<b>21</b>
1.1 Introduction .....	21
1.2 Consommation dans les architectures électroniques pour les transmissions .....	22
1.2.1 Puissance globale.....	22
1.2.2 Puissance dynamique.....	24
1.2.3 Puissance statique .....	27
1.3 Les circuits FPGA.....	30
1.3.1 Définition d'un FPGA .....	30
1.3.2 Structure d'un FPGA .....	31
1.3.3 Choix du circuit FPGA .....	33
1.4 Techniques de réduction de la consommation.....	34
1.4.1 Niveau Circuit.....	35

1.4.2	Niveau Logique .....	36
1.4.3	Niveau RTL et architectural .....	37
1.4.4	Niveau algorithmique .....	40
1.5	Conclusion .....	44
<b>Chapitre 2 Plateforme d'évaluation de la consommation énergétique .....</b>		<b>47</b>
2.1	Introduction .....	47
2.2	Etat de l'art des techniques d'estimation de la consommation .....	47
2.2.1	Evaluation de la consommation au niveau « transistor » .....	49
2.2.2	Estimation de la consommation au niveau « porte logique » .....	51
2.2.3	Estimation de la consommation au niveau « RTL » .....	53
2.2.4	Estimation de la consommation au niveau « système ou algorithmique » .....	54
2.3	Plateforme de test utilisée : mesure et simulation .....	57
2.3.1	Simulation XPower .....	58
2.3.2	Mesure .....	59
2.4	Conclusion .....	61
<b>Chapitre 3 Analyse de la consommation de puissance des blocs Wi-Fi .....</b>		<b>63</b>
3.1	Introduction .....	63
3.2	Contexte Wi-Fi .....	64
3.2.1	Chaîne d'émission/réception .....	64
3.2.2	Description et architecture des blocs .....	65
3.2.3	Validation .....	73
3.2.4	Architecture globale de la chaîne .....	74
3.2.5	Scénarios testés .....	76
3.3	Puissance statique .....	77
3.4	Puissance dynamique des chaînes Tx/Rx .....	78

---

3.5	Puissance dynamique des blocs .....	82
3.6	Conclusion .....	83
<b>Chapitre 4 Méthode semi-analytique d'estimation de la puissance dynamique consommée.....</b>		<b>85</b>
4.1	Introduction .....	85
4.2	Analyse temporelle .....	86
4.2.1	Cas général.....	86
4.2.2	Application à la chaîne Wi-Fi.....	88
4.3	Méthodologie.....	90
4.3.1	Base de données.....	91
4.3.2	Taux d'activité .....	91
4.3.3	Application numérique .....	92
4.4	Exemple .....	92
4.4.1	Chaîne BW 20 MCS 1 .....	93
4.4.2	Chaîne BW 20 MCS 5 .....	96
4.4.3	Taux d'activité des blocs .....	98
4.5	Conclusion.....	99
<b>Chapitre 5 Résultats de mesures et de simulations.....</b>		<b>101</b>
5.1	Introduction .....	101
5.2	Validation des taux d'activité .....	101
5.3	Chaîne Tx/Rx.....	103
5.3.1	Construction de la base de données .....	103
5.3.2	Calcul des puissances consommées réelles à partir des taux d'activité.....	104
5.3.3	Calcul de la puissance consommée de la chaîne par sommation.....	105
5.4	Exploration d'architecture – cas de la FFT.....	107

Table des matières

---

5.4.1	Consommation des blocs FFT isolés .....	108
5.4.2	Analyse temporelle de la chaîne .....	109
5.4.3	Impact de l'architecture de la FFT sur la consommation de la chaîne .....	111
5.5	Conclusion .....	113
<b>Conclusion et Perspectives .....</b>		<b>115</b>
<b>Références .....</b>		<b>121</b>

# Liste des figures

Fig. 1.1 : Chaîne de communication .....	23
Fig. 1.2 : Composantes de la puissance dynamique .....	26
Fig. 1.3 : Courants de fuite d'un transistor Pmos .....	27
Fig. 1.4 : Évolution de la puissance statique et dynamique en fonction de la technologie [15] ....	28
Fig. 1.5 : Nombre de transistors dans les circuits [17] .....	30
Fig. 1.6 : Bloc logique de base .....	32
Fig. 1.7 : Architecture FPGA.....	33
Fig. 1.8 : Principaux niveaux d'abstraction.....	35
Fig. 1.9 : Technique du clock gating.....	36
Fig. 1.10 : Technique DVFS .....	38
Fig. 1.11 : Architecture pipeline .....	39
Fig. 1.12 : Parallélisation d'architectures .....	40
Fig. 2.1 : Analyse de puissance - niveaux d'abstraction.....	48
Fig. 2.2 : Méthodes d'estimation statistique et probabiliste .....	52
Fig. 2.3 : Flot de conception XPower .....	59
Fig. 2.4 : Plateforme de mesure.....	61
Fig. 3.1 : Chaînes d'émission/réception.....	64
Fig. 3.2 : Architecture embrouilleur réalisée par des registres à décalage .....	65
Fig. 3.3 : Codeur.....	66
Fig. 3.4 : Poinçonnage 3/4.....	67
Fig. 3.5 : Fonctionnement du bloc modulation pour $K=6$ .....	69

Fig. 3.6 : <i>Orthogonalité des sous-porteuses [77]</i> .....	70
Fig. 3.7 : <i>Architecture FFT</i> .....	72
Fig. 3.8 : <i>Méthode de validation des blocs/chaînes</i> .....	73
Fig. 3.9 : <i>Méthode de validation du fonctionnement de la chaîne complète</i> .....	74
Fig. 3.10 : <i>Traitement de deux bits en parallèle</i> .....	75
Fig. 3.11 : <i>Interface des blocs</i> .....	76
Fig. 3.12 : <i>Puissances statique et dynamique vs température. 52 Mbps correspond au MCS5 pour une bande de 20 Mhz et 162 Mbps correspond au MCS8 pour une bande de 40 MHz</i> .....	78
Fig. 3.13 : <i>Courbes BER en fonction du SNR</i> .....	80
Fig. 3.14 : <i>Puissance dynamique de la chaîne Tx/Rx pour les différents scénarios. Chaque point représente un MCS donné</i> .....	81
Fig. 3.15 : <i>Puissance dynamique des blocs modulation et IFFT</i> .....	81
Fig. 4.1 : <i>Chronogramme de la chaîne d'émission, BW 20 MHz MCS 1</i> .....	95
Fig. 4.2 : <i>Chronogramme de la chaîne d'émission, BW 20 MHz MCS 5</i> .....	97
Fig. 4.3: <i>Puissance dynamique simulée du bloc IFFT de taille 256 en fonction de la fréquence d'horloge</i> .....	99
Fig. 5.1 : <i>Consommation de puissance des blocs entrelaceur et ifft lors du fonctionnement de la chaîne pour différents scenarios en fonction du débit</i> .....	105
Fig. 5.2 : <i>Chronogramme de la chaîne d'émission, pour une architecture FFT pipelinée, BW 40 MHz MCS 1</i> .....	109
Fig. 5.3 : <i>Chronogramme de la chaîne d'émission, pour une architecture FFT non pipelinée, BW 40M Hz MCS 1</i> .....	110

# Liste des tableaux

Tab. 1.1 : <i>Pire maximale</i> [9].....	24
Tab. 1.2 : <i>Opération "ADD" réorganisée afin d'avoir des chemins séparés de 0 et 1</i> .....	42
Tab. 2.1 : <i>Ressources disponibles sur le FPGA</i> .....	60
Tab. 3.1 : <i>Schéma de modulation (MCS)</i> .....	65
Tab. 3.2 : <i>Paramètres du bloc entrelaceur</i> .....	67
Tab. 3.3 : <i>Complexité et consommation des architectures d'entrelaceur pour une largeur de bande de 80 MHz et une modulation 16-QAM.</i> .....	68
Tab. 3.4 : <i>Nombre de bits par symbole complexe</i> .....	69
Tab. 3.5 : <i>Trame OFDM</i> .....	70
Tab. 3.6 : <i>Paramètres OFDM</i> .....	71
Tab. 3.7 : <i>Ressources utilisées par le bloc IFFT</i> .....	72
Tab. 3.8 : <i>Résultats de synthèse des blocs isolés</i> .....	73
Tab. 3.9 : <i>Scénarios Wi-Fi</i> .....	77
Tab. 3.10 : <i>Variation de la puissance statique</i> .....	78
Tab. 3.11 : <i>Puissance dynamique des chaînes Tx/Rx</i> .....	79
Tab. 3.12 : <i>Débits binaires</i> .....	79
Tab. 3.13 : <i>Consommation en puissance des blocs [mW]</i> .....	82
Tab. 3.14 : <i>Somme des blocs isolés et écart observé par rapport à la chaîne totale</i> .....	83
Tab. 4.1 : <i>Temps de traitement des blocs en nombre de cycles</i> .....	89
Tab. 4.2 : <i>Analyse du flux de données d'information binaire en nombre de cycles</i> .....	90
Tab. 4.3 : <i>Taux d'activité de la chaîne d'émission, BW 20 MHz MCS1</i> .....	95

Tab. 4.4 : Taux d'activité de la chaîne d'émission, BW 20 MHz MCS 5.....	97
Tab. 4.5 : Taux d'activité des blocs à l'émission lors du fonctionnement de la chaîne .....	98
Tab. 4.6 : Taux d'activité des blocs en réception lors du fonctionnement de la chaîne.....	98
Tab. 5.1 : Puissance consommée du bloc modulation [mW] .....	102
Tab. 5.2 : Puissance consommée du bloc IFFT [mW].....	102
Tab. 5.3 : Puissance des blocs [mW] - 100 % d'activité (base de données).....	103
Tab. 5.4 : Puissance des blocs [mW] - Activité réelle .....	104
Tab. 5.5 : Puissance consommée de la chaîne d'émission [mW] .....	106
Tab. 5.6 : Puissance consommée de la chaîne de réception [mW].....	106
Tab. 5.7 : Ressources utilisées par les deux architectures FFT .....	108
Tab. 5.8 : Puissance consommée des deux architectures FFT à 100% d'activité [mW] .....	108
Tab. 5.9 : Puissance consommée de la chaîne d'émission avec l'architecture FFT pipelinée et non pipelinée.....	111
Tab. 5.10 : Consommation et taux d'activité des deux architectures FFT a l'émission .....	112
Tab. 5.11 : Puissance consommée de la chaîne de réception avec l'architecture pipelinée et non pipelinée [mW] .....	113
Tab. 5.12 : Taux d'activité du bloc décodeur.....	113

# Acronymes

**ASIC** : Application Specific Integrated Circuit

**BER** : Bit Error Rate

**BPSK** : Binary Phase Shift Keying

**BW** : Bandwidth

**CAO** : Conception Assistée par Ordinateur

**CLB** : Configurable Logic Block

**CMOS** : Complementary Metal Oxyde Semi-conductor

**DSP** : Digital Signal Processor

**DVFS** : Dynamic Voltage and Frequency Scaling

**FF** : Flip Flop

**FFT** : Fast Fourier Transform

**FPGA** : Field Programmable Gate Array

**IEEE** : Institute of Electrical and Electronics Engineers

**IFFT** : Inverse Fast Fourier Transform

**IP** : Intellectual Property

**JTAG** : Joint Test Action Group

**LDPC** : Low Density Parity Check

**LUT** : Look Up Table

**MCS** : Modulation and Coding Scheme

**MIMO** : Multiple Inputs Multiple Outputs

**MMSE** : Minimum Mean Square Error

**OFDM** : Orthogonal Frequency Division Multiplexing

**PAPR** : Peak to Average Power Ratio

**PHY** : Physical layer

**PIRE** : Puissance Isotrope Rayonnée Equivalente

**QAM** : Quadrature Amplitude Modulation

**QPSK** : Quadrature Phase Shift Keying

**RAM** : Random Access Memory

**RF** : Radio Frequency

**RRAM** : Resistive Random Access Memory

**RTL** : Register Transfer Level

**Rx** : Receiver

**SISO** : Single Input Single Output

**SNR** : Signal to Noise Ratio

**SRAM** : Static Random Access Memory

**TIC** : Technologies de l'Information et de la Communication

**TPC** : Transmission Power Control

**Tx** : Transmitter

**VHDL** : Very high speed integrated circuit Hardware Description Language

**Wi-Fi** : Wireless Fidelity

# Introduction

Du fait de l'expansion considérable des usages multimédia ces dernières années, le nombre d'appareils connectés est en constante augmentation, incluant par exemple les tablettes, smartphones, ordinateurs portables, consoles de jeux vidéo, et plus récemment de nouveaux objets intelligents tels que les stations météo, les dispositifs de surveillance ou encore le contrôle de l'éclairage et du chauffage domestique. Les technologies de l'information et de la communication (TIC) sont donc en pleine croissance, pour répondre à la demande grandissante et diversifiée en services à haut débit. Cependant, ces évolutions représentent un coût énergétique, ainsi qu'un impact environnemental non négligeable. En effet, selon une étude de l'Agence De l'Environnement et de la Maîtrise de l'Energie (ADEME) [1], plus de 30% de la consommation électrique du domicile, hors chauffage et production d'eau chaude, est due au matériel informatique et audiovisuel. Les préoccupations énergétiques des technologies de connectivité sont devenues par conséquent un enjeu sociétal majeur.

Un certain nombre d'initiatives sont prises pour réduire l'impact environnemental de l'industrie des TIC, mais reposent aujourd'hui sur l'engagement volontaire des acteurs, comme par exemple la charte de la Fédération Française des Télécommunications pour la maîtrise des consommations d'énergies [2], ou encore les accords européens « European Code of Conduct on Energy Consumption of Broad Band Equipment » [3], et « Voluntary Industry Agreement to improve the Energy Consumption of Complex Set Top Boxes » [4].

De plus en plus de projets abordent la thématique de l'efficacité énergétique, comme le projet collaboratif français GREENCoMM [5], construit autour d'un consortium d'industriels et de laboratoires et qui rassemble plusieurs partenaires tels que Telecom Bretagne, partenaire académique, deux PME, Siradel et Idil, trois entreprises de taille intermédiaire, Satimo, Turbo Concept et Niji ainsi que les groupes Orange et Technicolor. Ce projet a pris fin en Mars 2016. L'objectif de ce projet était de réduire la consommation d'énergie des technologies de connectivité du réseau local domestique incluant l'accès radio Wi-Fi, les transmissions par courant porteur en ligne, et l'accès par fibre optique. Dans la continuité de ce projet, nous avons analysé la

technologie de communication sans fils Wi-Fi, qui représente un cas particulièrement impactant: largement répandue non seulement dans les réseaux domestiques, mais aussi dans les lieux publics (hôtels, aéroports, centres de conférences, restaurants, parcs, etc.), cette technologie fait aujourd'hui partie intégrante de notre quotidien. En ce sens, Orange Labs à l'origine du projet GREENCoMM, a lancé cette étude qui a été menée dans un de ses laboratoires et dont les résultats vous sont présentés dans ce manuscrit.

Nous proposons d'analyser la chaîne de communications numériques Wi-Fi d'un nouveau point de vue, celui de la consommation énergétique. Nous nous concentrons essentiellement sur la partie bande de base de la couche physique (PHY); les aspects Radio Fréquence (RF) et couches hautes ayant été abordés par ailleurs dans le projet GREENCoMM. Le fait de considérer la chaîne Wi-Fi dans son ensemble, de tenir compte des interactions entre les différents éléments de traitement du signal, et d'évaluer la puissance dissipée par les circuits intégrés en plus de la puissance de transmission de l'émetteur, constituent le principal aspect novateur de la thèse.

L'évaluation de la consommation de chaque élément composant la chaîne de communications numériques Wi-Fi constitue une première étape vers la réduction de l'empreinte énergétique. Ainsi, ma thèse répond dans un premier temps à une problématique importante qui est la difficulté d'évaluer l'énergie dissipée dans des circuits intégrés numériques de plus en plus complexes. De nombreux travaux supposent que la puissance des circuits est négligeable devant la puissance de transmission à l'antenne. Or pour certaines applications cette hypothèse s'avère fautive: en effet, le développement de techniques de communications (encodage, modulation, égalisation, décodage, etc.) de plus en plus sophistiquées entraîne une complexité de traitement plus importante et par la suite une consommation en puissance circuit non négligeable. Il est donc nécessaire de reconsidérer le problème classique en ajoutant une contrainte sur la puissance circuit. D'autre part, les contraintes de qualité de service doivent être maintenues. En effet, l'optimisation de l'efficacité énergétique ne doit pas se faire au détriment de la qualité du signal transmis, et donc du débit réalisé pour un taux d'erreurs acceptable.

Les rares études menées sur le sujet ne fournissent pas de mesures pratiques ou d'informations théoriques qui nous permettraient de lier la quantité d'informations échangées avec la consommation électrique des circuits. La complexité des systèmes de télécommunications augmente constamment au fil des avancées technologiques, ainsi les systèmes électroniques permettant d'embarquer ces nouvelles technologies doivent être de plus en plus performants. Les ASIC (Application Specific Integrated Circuit) sont des circuits intégrés permettant d'obtenir d'excellentes performances dans les systèmes de communication. Leur temps et leur coût de

développement est important et est stratégique pour une entreprise. Il y a peu d'études chiffrées sur la consommation réelle des circuits ASIC dédiés aux communications ; les moyens de comparer des architectures au niveau circuit sont donc limités. C'est pourquoi depuis quelques années les circuits FPGA (Field Programmable Gate Array), sont devenus une véritable alternative aux ASIC dans la phase amont de la production en série des circuits, permettant ainsi de vérifier rapidement la fonctionnalité et d'analyser les performances. Les FPGAs présentent l'avantage d'être reconfigurables dynamiquement, donc extrêmement flexibles; et permettent d'atteindre de bonnes performances dans un temps de développement réduit ce qui est l'idéal lorsque l'on souhaite faire du prototypage rapide. Ainsi il est possible de facilement comparer différentes architectures et d'analyser les ressources à chaque étape du flot de conception; l'objectif de mes travaux est d'utiliser également les FPGAs dans l'analyse de la consommation afin d'évaluer rapidement la puissance dissipée. Nous faisons l'hypothèse que l'ordre de grandeur des gains obtenus sur FPGA à la suite d'une optimisation ayant pour critère la consommation énergétique sera répercuté sur les circuits ASICs.

Les contributions apportées par la thèse sont les suivantes. Tout d'abord, une chaîne de communications numériques Wi-Fi a été développée en langage VHDL (Very high speed integrated circuit Hardware Description Language) et implémentée sur FPGA dans le but de mesurer, sur la carte, la consommation de puissance pour différents paramètres système (largeur de bande, modulation, codage, etc) ainsi que leur impact. Nous avons également utilisé un outil de simulation capable d'évaluer les consommations de puissance des circuits et montré que les résultats de simulation sont proches de ceux obtenus par mesure, avec une erreur moyenne de 6.5%. L'évaluation de la consommation au niveau transistor, après placement-routage, est très précise car toutes les informations d'implémentations sont connues. Néanmoins, l'estimation de la consommation nécessite énormément de temps de simulation ou bien de configuration du circuit, ce qui n'est pas acceptable lorsque l'on souhaite explorer différentes architectures afin d'optimiser l'efficacité énergétique. Pour répondre à cette problématique, nous proposons une méthode d'estimation de la consommation à haut niveau d'abstraction tout en prenant en compte les informations énergétiques issues du niveau matériel. La méthode proposée se base sur l'analyse temporelle du système et permet de calculer mathématiquement l'activité des éléments de la chaîne afin d'estimer la consommation de puissance rapidement. Un point fort de la méthode est qu'elle permet un gain de temps considérable et facilite l'exploration d'architecture. Nous illustrons l'efficacité de cette méthode en comparant deux implémentations d'une transformée de

Fourier rapide (Fast Fourier Transform, FFT) qui ont des complexités différentes avec un effet direct sur l'activité de la chaîne, et donc sur la consommation énergétique.

Ces contributions sont présentées dans ce manuscrit et organisées en cinq chapitres. Le chapitre 1 présente les sources de consommations dans les circuits, une rapide comparaison des avantages et inconvénients des FPGAs versus ASICs et les méthodes de réduction de la consommation à différentes étapes du flot de conception.

Le chapitre 2 décrit l'état de l'art des méthodes d'estimation de la consommation à différents niveaux d'abstraction. Nous verrons notamment que plus l'estimation est réalisée tôt dans le flot de conception, moins elle est précise à cause du manque d'information sur l'implémentation matérielle. Nous présentons également l'outil de simulation que nous avons sélectionné ainsi que la plateforme matérielle permettant de mesurer la consommation sur FPGA.

Le chapitre 3 présente la chaîne de communications numériques Wi-Fi spécifiée par la norme IEEE 802.11ac [6]. Les puissances consommées par les éléments des chaînes d'émission et de réception sont évaluées. Nous montrons notamment que les évaluations issues des mesures et des simulations sont très cohérentes.

Dans le chapitre 4, notre méthode d'estimation de la consommation basée sur l'analyse temporelle des chaînes est présentée. Cette méthode rapide et efficace est applicable à tout système de communications numériques, et tient compte de l'activité temporelle des différents éléments des chaînes.

Le chapitre 5 fournit des résultats de simulation issus des méthodes d'évaluation de la consommation des chaînes Wi-Fi proposées aux chapitres précédents. Nous montrons que la méthode proposée au chapitre 4 fournit un gain significatif en temps de simulation par rapport à une évaluation de la chaîne globale via l'outil de simulation décrit au chapitre 3, ce qui facilite l'exploration d'architecture. Ainsi, nous comparons deux types d'implémentations de transformée de Fourier rapide, l'une est complexe mais permet un temps de traitement court, l'autre est plus simple mais prend plus de temps; et nous évaluons l'impact de ces implémentations sur la consommation globale de la chaîne d'émission.

Enfin, nous concluons par un résumé des principaux résultats et nous proposons des perspectives de travaux futurs.

# Chapitre 1

## Consommation de puissance dans les circuits

### 1.1 Introduction

La forte densité d'intégration sur les circuits électroniques fait que l'énergie est devenue un facteur critique, qui ne peut plus être négligé au détriment de la performance et du coût. A chaque évolution technologique, un circuit électronique intègre plus de transistors d'où l'augmentation de la consommation. Il est également à noter que de nombreux systèmes sont autonomes et ont pour seule alimentation une batterie dont la durée de fonctionnement doit être suffisamment longue pour satisfaire les utilisateurs. C'est pourquoi, la consommation de puissance est devenue un des principaux sujets lorsque l'on s'intéresse aux communications sans fil.

Un des enjeux de la thèse est d'estimer, d'analyser et de réduire la consommation de puissance dissipée dans les circuits de communications numériques. Aujourd'hui, une estimation sur ASIC (Application Specific Integrated Circuit) est compliquée car il est difficile de disposer du temps et des moyens nécessaires pour les phases de conception et d'implémentation. Les FPGAs (Field Programmable Gate Array) sont une alternative réelle aux circuits ASICs de par leur importante puissance de calcul qui permet d'implémenter des algorithmes très complexes. Un des avantages des FPGAs est leur flexibilité qui leur permet de réaliser du prototypage rapide d'architectures, par exemple pour des algorithmes de traitement de signal utilisés dans les communications sans fil.

Aujourd'hui, avec l'augmentation des usages multimédia, les systèmes sont toujours plus performants (puissance de calcul et rapidité) mais consomment également plus de puissance, ce qui représente une source de préoccupation importante pour les concepteurs et développeurs de

circuits intégrés. Il existe de nombreuses méthodes de réduction de la puissance dissipée par les circuits dans l'état de l'art, intervenant à différents niveaux du flot de conception.

L'objectif de ce chapitre est de fournir dans un premier temps les sources de puissance dissipée par un circuit intégré, puis de comparer les circuits FPGAs et ASICs. La structure d'un FPGA sera notamment détaillée. Enfin, nous verrons les principales méthodes de réduction de la consommation à différents niveaux d'abstraction allant du niveau transistor au niveau algorithmique. Nous montrerons les avantages et inconvénients de chacune.

## **1.2 Consommation dans les architectures électroniques pour les transmissions**

### **1.2.1 Puissance globale**

La consommation d'un système de communication inclut la puissance consommée par la chaîne radio fréquence (RF) et la puissance dissipée par les circuits réalisant les fonctions de communication, comme schématisé Fig. 1.1. La consommation des circuits due aux algorithmes de traitement de signal a souvent été négligée devant la puissance consommée pour la transmission. Elle est cependant prise en compte lors de la conception des systèmes de communication, en particulier dans les environnements intérieurs impliquant de courtes distances et donc, de faibles puissances d'émission. On retrouve dans la littérature quelques travaux donnant des chiffres quant au poids de la répartition entre ces deux puissances. Dans [7], les auteurs montrent que 70% de la puissance totale est consommée par la chaîne RF et que 30% de la puissance restante est consommée par la partie circuit dans le cas de la norme 802.11b. Au sein du projet GREENCoMM [8], la question de la répartition de la consommation globale est traitée pour la norme 802.11n. Cette norme Wi-Fi est plus évoluée que la norme 802.11b avec l'utilisation notamment de techniques multi-antennes (Multiple Input Multiple Output, MIMO). Ainsi, ce projet arrive à la conclusion que 50 à 70% de la consommation totale provient de la chaîne RF alors que 30 à 50% de la consommation est due à la partie circuit. Peu de travaux traitent le cas de la répartition de la consommation pour la norme 802.11ac. Néanmoins, avec l'introduction des largeurs de bande plus grandes (80 et 160 MHz) ainsi qu'une modulation plus élevée (256 QAM) par rapport à la norme 802.11n, on peut émettre l'hypothèse que la répartition de consommation s'équilibre.

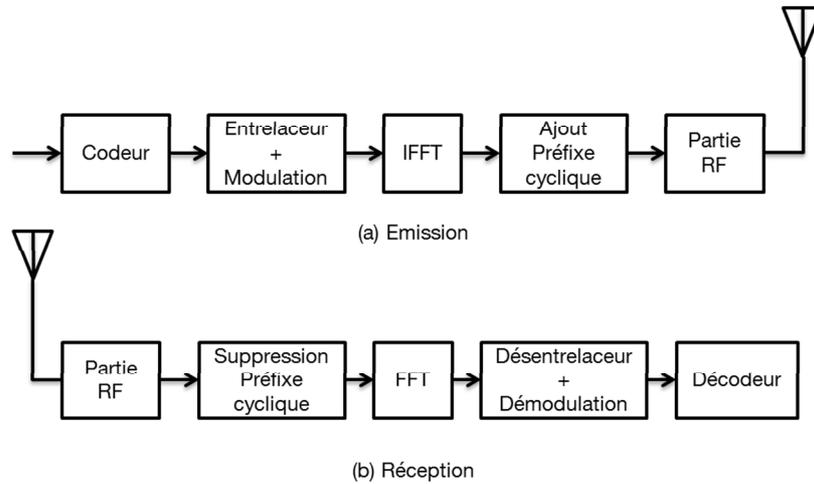


FIG. 1.1 : CHAINE DE COMMUNICATION

### 1.2.1.1 Puissance d'émission en sortie de l'antenne

La puissance totale d'émission d'une carte (Wi-Fi par exemple) est généralement indiquée par la valeur de la puissance isotrope rayonnée équivalente (PIRE). Cette valeur fait référence à la puissance maximale émise par une antenne dans une direction donnée. La PIRE est obtenue en additionnant la puissance d'émission du matériel en dBm et le gain de l'antenne en dBi. Le gain de l'antenne représente la capacité de celle-ci à émettre sur de grandes distances. La PIRE est limitée par les organismes de normalisation des différents pays. En Europe, la puissance d'émission maximale rayonnée par les antennes pour les applications Wi-Fi dans la bande de 5 GHz est de 100 à 1000 mW comme indiqué Tab. 1.1 [9].

Il est à noter que la puissance consommée par l'amplificateur de puissance est très importante et dépend de la puissance en sortie de l'antenne par la relation [10]:

$$P_{PA} = (1 + \alpha)P_{out} \quad (1.1)$$

où  $P_{PA}$  représente la puissance consommée par l'amplificateur de puissance,  $P_{out}$  représente la puissance d'émission,  $\alpha = (\varepsilon/\eta) - 1$  avec  $\varepsilon$  l'efficacité du drain de l'amplificateur de puissance et  $\eta$  le rapport entre puissance maximale et puissance moyenne (Peak to Average Power Ratio, PAPR).

À noter la présence du contrôle de puissance de transmission (Transmission Power Control, TPC) dans certains cas qui influence la valeur de la PIRE maximale. Le rôle du TPC est d'assurer le lien radio entre l'émetteur et le récepteur tout en s'assurant que la puissance d'émission est la plus petite possible [11]. La consommation énergétique des appareils mobiles est alors réduite en ajustant la puissance d'émission au cours du temps à son niveau le plus faible en fonction du canal

et de ses variations. En effet, sans l'utilisation de cette méthode la puissance de transmission est fixée à son maximum sans prendre en compte l'état du canal. Cette méthode peut être bénéfique pour les applications mobiles, car étant en mouvement en permanence, l'état du canal peut osciller entre les conditions favorables et défavorables.

Fréquence [MHz]	PIRE maximale [dBm]	
	Avec TPC	Sans TPC
5150 à 5350	23	20/23
5470 à 5725	30	27

TAB. 1.1 : *PIRE MAXIMALE* [9]

### 1.2.1.2 Puissance circuit (consommation dans les circuits CMOS)

La logique CMOS utilise deux types de transistors, un transistor de type PMOS et un transistor de type NMOS. La fonction de base réalisée avec ces deux types de transistors est un inverseur de signal comme le montre la Fig. 1.2. A partir de cette fonction de base, il est possible de créer des portes logiques plus complexes (ET, NON ET, OU, NON OU, XOR). Ainsi, lorsque les transistors sont en fonctionnement, des pertes de courant apparaissent, caractérisées par une consommation en énergie. La consommation d'énergie d'une fonction logique de type CMOS est la somme de deux composantes indiquées à l'équation (1.2) : une composante dynamique et une statique. La composante dynamique est due à l'activité des transistors alors que la composante statique est due aux courants de fuites engendrés par les transistors. Ces deux composantes sont détaillées dans les sections suivantes.

$$P_{Circuit} = P_{Dynamique} + P_{Statique} \quad (1.2)$$

### 1.2.2 Puissance dynamique

La puissance dynamique est créée par l'activité de commutation des transistors lorsqu'une horloge est appliquée au système. Cette activité cause l'apparition de deux courants: le courant dit de court-circuit et le courant de commutation. Ces deux courants sont décrits plus en détail aux paragraphes suivants.

### 1.2.2.1 Courant de court-circuit

Le courant de court-circuit apparaît lors de la commutation des transistors d'une porte logique. Lors de la commutation de l'inverseur, les transistors PMOS et NMOS vont commuter en même temps, ils seront donc passants au même moment reliant ainsi directement l'alimentation à la masse. Ce chemin direct qui a été créé pendant un temps très court, génère le courant de court-circuit comme schématisé Fig. 1.2. La consommation de court-circuit dépend des paramètres liés à la technologie de réalisation des transistors, notamment à la tension de seuil, mais aussi de l'alimentation et de la fréquence d'horloge. Il est possible de trouver dans l'état de l'art différentes propositions de modélisation de ce courant [12][13][14]. Néanmoins, les courants de court-circuit sont souvent négligeables devant le courant de commutation.

### 1.2.2.2 Courant de commutation

Le courant de commutation représente la charge/décharge de la capacité de chargement présente en sortie des portes logiques. La commutation d'un des deux transistors présents dans une porte logique va provoquer le passage d'un courant  $I_c$  issu de l'alimentation  $V_{dd}$  vers la charge de sortie  $C_l$ . La Fig. 1.2 représente le passage du courant de commutation  $I_c$  allant de l'alimentation vers la capacité.

Ainsi, lors de la charge de la capacité  $C_l$  en sortie, la puissance fournie est égale à :

$$P(t) = V_{dd} \cdot I_c(t) \quad \text{avec } I_c(t) = C_l \frac{dV_c}{dt} \quad (1.3)$$

L'énergie tirée de l'alimentation lors de la charge de la capacité  $C_l$  est quant à elle égale à l'intégrale de la puissance fournie en :

$$\begin{aligned} E_{tot} &= \int_{t_0}^{t_1} P(t) dt = \int_{t_0}^{t_1} V_{dd} \cdot I_c(t) dt \\ &= \int_{t_0}^{t_1} V_{dd} \cdot C_l \frac{dV_c}{dt} dt = \int_{t_0}^{t_1} V_{dd} \cdot C_l dV_c \end{aligned} \quad (1.4)$$

En supposant que les conditions initiales soient les suivantes : à l'instant  $t_0$ ,  $V_c = 0$  et à l'instant  $t_1$ ,  $V_c = V_{dd}$ , l'équation (1.4) devient alors :

$$\begin{aligned}
 E_{tot} &= \int_0^{V_{dd}} V_{dd} C_l dV_c = V_{dd} C_l \int_0^{V_{dd}} dV_c \\
 &= V_{dd}^2 C_l
 \end{aligned} \tag{1.5}$$

Une partie de l'énergie est dissipée dans les transistors, le reste sert à charger ou à décharger la capacité de sortie de la porte logique. Par exemple, une fois la capacité complètement chargée, l'expression de l'énergie à la borne de la capacité  $C_l$  est alors :

$$\begin{aligned}
 E_{C_l} &= \int_0^{V_{dd}} V_c C_l dV_c = C_l \int_0^{V_{dd}} V_c dV_c \\
 &= \frac{1}{2} V_{dd}^2 C_l
 \end{aligned} \tag{1.6}$$

La capacité se chargeant et se déchargeant successivement dans le circuit à une fréquence d'horloge  $f$ , la puissance de commutation dissipée (1.5) est donc proportionnelle à  $f$  et devient :

$$P = E_{tot} \cdot f = V_{dd}^2 \cdot C_l \cdot f \tag{1.7}$$

Enfin, il convient de prendre également en compte l'activité de commutation  $\alpha$  par cycle d'horloge. En effet, l'activité a un rôle prépondérant dans la consommation de puissance dynamique du système, la formule (1.7) finale devient alors :

$$P_{Dynamique} = \alpha V_{dd}^2 C_l f \tag{1.8}$$

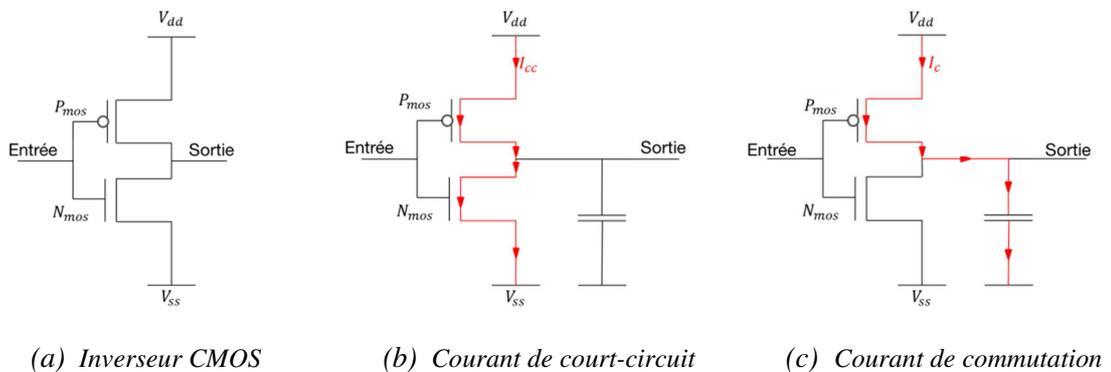


FIG. 1.2 : COMPOSANTES DE LA PUISSANCE DYNAMIQUE

### 1.2.3 Puissance statique

La consommation statique est principalement due aux courants de fuite résultant du courant sous seuil  $I_{ss}$  et du courant inverse  $I_i$ . La Fig. 1.3 schématise les deux principaux courants de fuite.

Le courant sous seuil est le courant de fuite sous la tension de seuil, il apparaît lorsque la tension entre la grille et la source est inférieure à la tension de seuil du transistor. En d'autres termes, plus la tension de seuil est basse, plus la puissance statique augmente. Ce courant est souvent négligé en pratique.

Le courant inverse est la somme des courants de fuite formant des diodes parasites engendrant des courants de polarisation inverses formés principalement entre la grille et le substrat. Ce courant devient de plus en plus important avec l'évolution de la technologie. En effet, avec l'avancée technologique des transistors (sous les 90 nm notamment), l'épaisseur de l'oxyde de grille des transistors a fortement diminué, facilitant le passage d'un courant de fuite. Il est à noter qu'à température ambiante, les fuites liées au courant inverse (de la grille au substrat) sont plus importantes que les fuites liées au courant sous seuil (de la source vers le drain) dans les transistors ayant une technologie inférieure à 90 nm. Cependant, contrairement aux fuites de source à drain, les fuites de grille vers le substrat augmentent faiblement en fonction de la température.

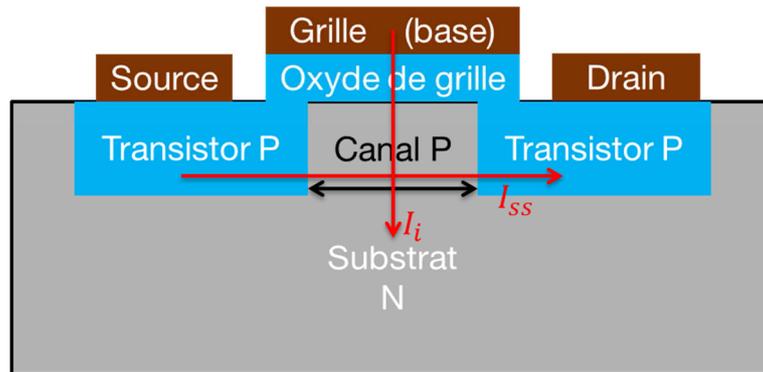


FIG. 1.3 : COURANTS DE FUITE D'UN TRANSISTOR PMOS

Par le passé la puissance statique a souvent été négligée car elle était quantitativement faible par rapport aux autres sources de consommation (puissance dynamique). Cependant avec l'émergence de nouvelles technologies (largeur de gravure plus faible) et la forte densité d'intégration des transistors, la puissance statique ne peut plus être négligée. La Fig. 1.4 schématise l'évolution des

puissances statique et dynamique en fonction de la technologie des transistors: on remarque que la puissance statique n'a cessé d'augmenter et qu'elle devient même plus importante lorsque la technologie est inférieure à 70 nm. Pour cette raison, les courants de fuite deviennent une source de préoccupation majeure.

Lorsque le système est au repos (non-commutation du circuit CMOS) ou que l'activité dynamique du design est faible, la puissance statique représentant les courants de fuite est dissipée. En d'autres termes, on remarque une consommation de puissance même lorsque le circuit n'effectue aucune tâche, donc aucune transition. Cela se produit lorsque la fréquence d'horloge est nulle ou lorsque l'activité est nulle ( $\alpha=0$ ). Un transistor ouvert possède une résistance très élevée mais celle-ci n'étant pas parfaite, des courants de fuite en résultent, reliant directement l'alimentation à la masse. A l'inverse la puissance statique est négligeable dans l'absolu si l'activité du système est grande ( $\alpha$  tend vers 1).

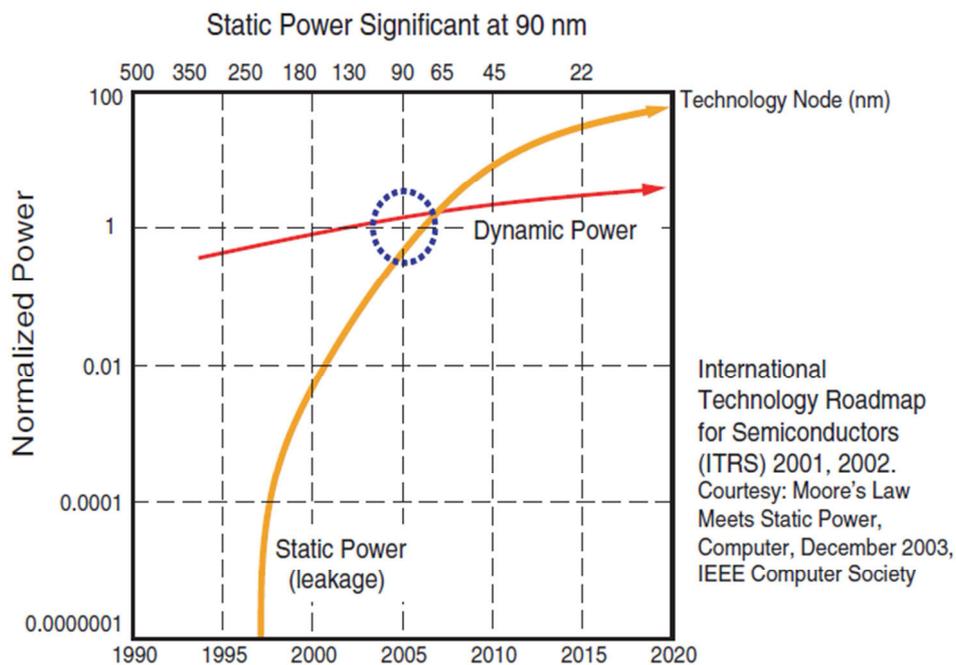


FIG. 1.4 : ÉVOLUTION DE LA PUISSANCE STATIQUE ET DYNAMIQUE EN FONCTION DE LA TECHNOLOGIE

[15]

La consommation statique est principalement due aux courants de fuite  $I_{fuite}$  mais aussi au nombre de transistors  $N_{tr}$ , lié à la surface du circuit, et à la tension d'alimentation  $V_{dd}$  caractérisée par l'équation (1.9). En effet, plus la surface est importante, plus la puissance statique

sera importante. De plus, si l'activité du système est faible alors la puissance statique devient importante devant la puissance dynamique. Par exemple, si les mémoires liées au design sont de grande taille et que l'activité totale du système est faible, la puissance statique devient dès lors très importante.

$$P_{Statique} = V_{dd} I_{fuite} N_{tr} \quad (1.9)$$

Les facteurs impactant la puissance statique sont principalement la tension d'alimentation et la température. De plus, la puissance statique dissipée est déterminée dès la conception de la technologie du composant car fortement dépendante de celle-ci, cela signifie que seuls les constructeurs de circuits peuvent diminuer la puissance statique lors de la conception au plus bas niveau d'abstraction.

D'après la loi de Moore, Fig. 1.5, le nombre de transistors disponibles sur un circuit est doublé tous les deux-trois ans impliquant une croissance exponentielle du nombre de transistors dans les circuits. Cette croissance a pu être réalisée notamment grâce à la réduction de la largeur de gravure des transistors mais aussi aux innovations technologiques en progression constante. Le processeur Intel 4004, lancé en 1972, était gravé en 10  $\mu\text{m}$  alors qu'en 2015 Intel grave des circuits en 14 nm [16]. La réduction de la largeur de gravure année après année a permis d'avoir des transistors de plus en plus petits et donc d'en mettre une quantité plus élevée sur un circuit. Cela a de nombreux avantages comme de diminuer les capacités parasites d'un transistor car la surface de celui-ci est plus petite, et donc de diminuer la puissance dynamique d'un circuit. Cependant, la diminution de la largeur de gravure n'a pas que des avantages, on observe un passage de courant de fuite plus important impliquant une augmentation de la consommation statique.

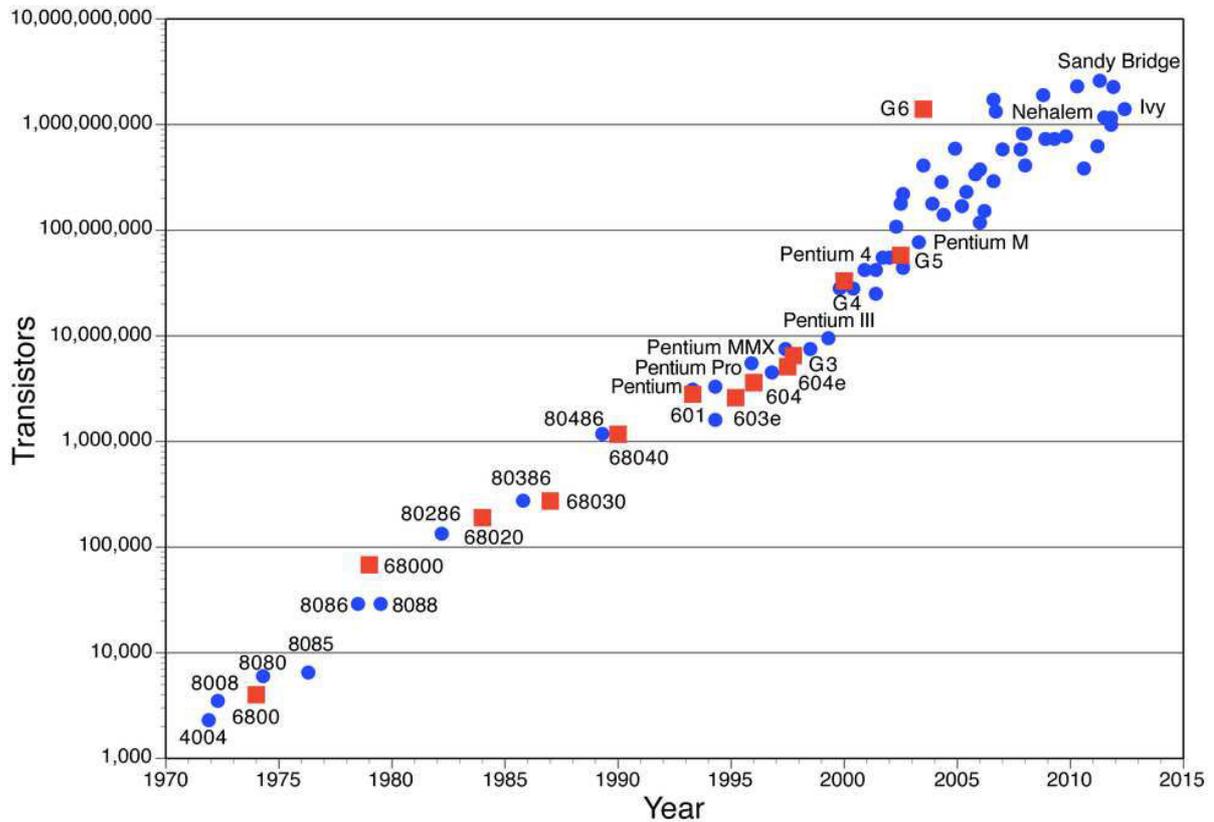


FIG. 1.5 : NOMBRE DE TRANSISTORS DANS LES CIRCUITS [17]

## 1.3 Les circuits FPGA

Nous avons vu à la section précédente les sources de consommation des circuits intégrés. Cette section abordera le problème du choix du circuit intégré le mieux adapté pour notre travail consistant à implémenter une chaîne de communication numérique sans fil et d'en estimer la consommation pour différentes architectures et jeux de paramètres.

### 1.3.1 Définition d'un FPGA

FPGA est l'abréviation de «Field Programmable Gate Array», c'est un circuit intégré reprogrammable. Un FPGA est destiné à réaliser n'importe quelles tâches et peut être reprogrammé à l'infini, ce qui le rend extrêmement flexible. Le premier FPGA fut inventé en 1985 par la société Xilinx. Aujourd'hui, il connaît un franc succès car ce type de circuit est idéal lorsque l'on souhaite faire du prototypage. De plus il peut être vu comme la combinaison d'un

ASIC et d'un processeur. Tout comme un ASIC, un FPGA est de nature complètement parallèle, ainsi les différents traitements des opérations s'exécutent sans concurrence. De plus, cet aspect parallèle permet d'avoir de bonnes performances en matière de latence d'exécution par exemple; et de posséder la même souplesse d'exécution logicielle qu'un processeur sans être limité par le nombre de cœurs de traitement. En effet, chaque tâche de traitement indépendante est affectée à une section spécifique du FPGA et peut donc s'exécuter en parallèle des autres tâches sans dépendre des autres blocs logiques. Cela permet d'augmenter le volume de traitement en parallèle sans que les performances d'une partie de l'application n'en soient affectées [18]. Cependant, l'utilisation d'un FPGA nécessite des compétences spécifiques en conception matérielle, par exemple le langage de description (VHDL, Verilog).

### **1.3.2 Structure d'un FPGA**

Un FPGA est constitué principalement de blocs logiques qui sont des ressources prédéfinies pouvant être reliées entre elles à l'aide d'interconnexions programmables afin de mettre en œuvre un circuit numérique reconfigurable. Des blocs d'entrées/sorties sont également disponibles afin d'interagir avec le monde extérieur.

Pour résumer, tout FPGA est généralement composé de:

1. De blocs logiques configurables (Configurable Logic Block, CLB)
2. D'interconnexions programmables et reconfigurables
3. De blocs d'entrées/sorties
4. De ressources mémoires embarquées
5. De blocs logiques dédiés (multiplieur, DSP (Digital Signal Processing), etc)

L'architecture d'un FPGA regroupant tous les composants matériels décrits précédemment est illustrée Fig. 1.7. La sélection d'un FPGA par rapport à un autre se fait principalement en comparant le nombre de blocs logiques configurables, de blocs logiques dédiés ainsi que la taille des ressources mémoires embarquées.

Les blocs logiques configurables sont l'unité logique élémentaire d'un FPGA, et sont constitués de deux parties, les bascules (Flip-Flops en anglais, FF) et les tables de correspondance ou de vérité (Look-Up Table en anglais, LUT) comme schématisé sur la Fig. 1.6. La manière dont sont reliées les bascules et les tables de correspondance diffère d'un FPGA à un autre. Tous les CLB d'un même FPGA sont identiques et permettent de créer des fonctions logiques complexes. Il est

également possible de les utiliser comme éléments de stockage de signaux. Grâce à l'évolution de la technologie, l'architecture des CLB a énormément évolué en complexité, et leur intégration sur FPGA évolue de manière exponentielle. À titre d'information, le premier FPGA XC2064 inventé par Xilinx en 1985 possédait 64 CLB alors qu'aujourd'hui plus de 2 millions de CLBs sont implantés sur les Virtex 7 de Xilinx.

Contrairement à une bascule qui effectue des traitements séquentiels, la table de correspondance effectue des traitements en combinatoire. Une partie importante de la logique d'un bloc CLB est réalisée par des tables de vérité permettant d'indiquer l'état de sortie en fonction des combinaisons de ses entrées. Un opérateur logique ('NON', 'ET', 'OU', 'XOR', etc.) est réalisé à l'aide d'une table de vérité.

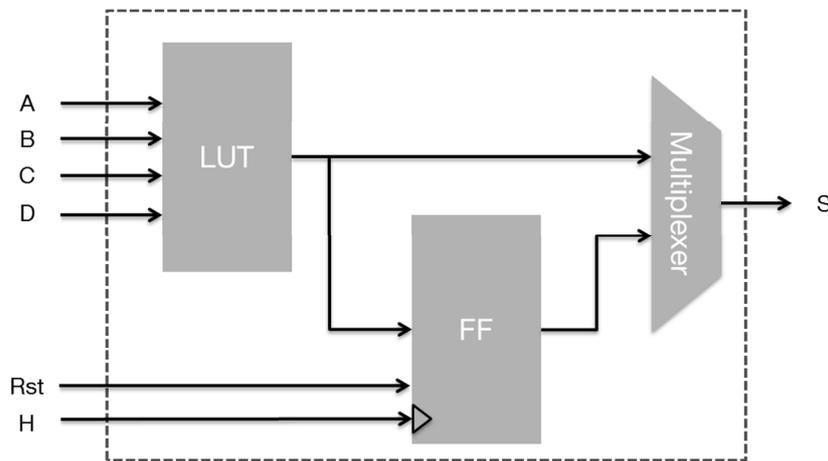


FIG. 1.6 : BLOC LOGIQUE DE BASE

Les blocs logiques dédiés correspondent généralement à des blocs de calculs réalisant des opérations complexes telles que la multiplication. Une multiplication peut s'avérer être extrêmement gourmande en ressources. À titre d'exemple, une multiplication de deux nombres sur 32 bits requiert la réalisation de plus de 2000 opérations. Afin de réduire l'utilisation du nombre de CLBs (FF et LUT) la majorité des FPGAs modernes est munie de multiplicateurs préconstruits. Il existe des circuits dotés de multiplicateurs accumulateurs également appelés DSP qui permettent de réaliser des calculs complexes en utilisant un minimum de ressources.

Enfin les blocs de mémoire embarqués, également appelés mémoire RAM (Random Access Memory) sur FPGA, s'avèrent extrêmement utiles. Ils permettent entre autres de stocker des données et même de pouvoir transférer des informations (valeurs), entre tâches s'exécutant en parallèle.

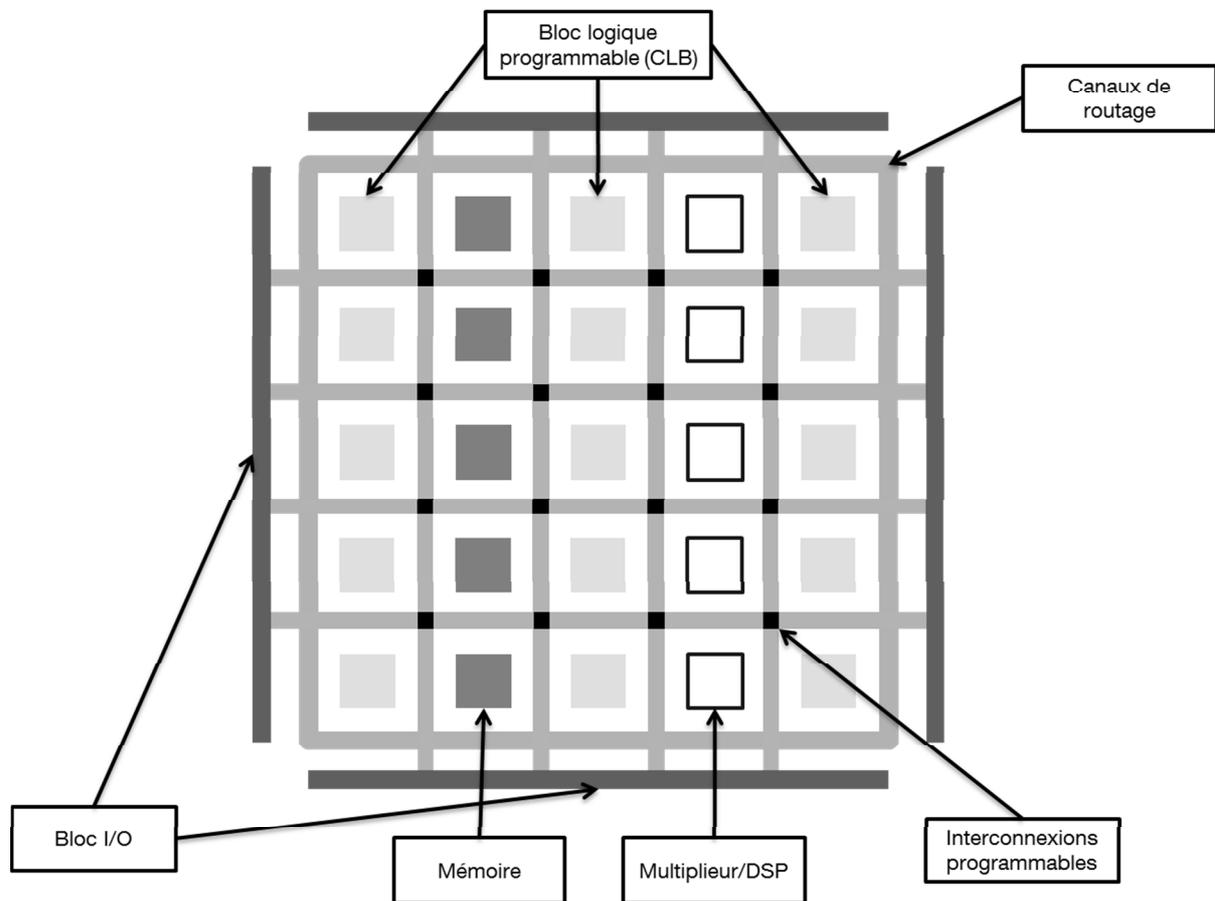


FIG. 1.7 : ARCHITECTURE FPGA

### 1.3.3 Choix du circuit FPGA

Les circuits intégrés commercialisés pour une application particulière sont en général des ASICs permettant de répondre à un cahier des charges précis. L'architecture développée est alors spécifique à une application et est optimale en termes de ressources et de performances. En ce sens, l'évaluation de la consommation d'une application sur ASIC est extrêmement précise pour l'application considérée.

Cependant, des points négatifs existent quant à l'utilisation des ASICs qui sont le temps de fabrication et le prix. Le temps de fabrication d'un ASIC est très long et peut prendre plusieurs mois, à cause de sa complexité. Les deux grandes étapes nécessaires à la production d'un ASIC sont l'étape d'implémentation et l'étape de fabrication/production [19]. L'étape d'implémentation consiste à décrire le design dans un langage matériel (en général VHDL ou Verilog), puis de le vérifier, et enfin de le synthétiser. Quant à l'étape de fabrication elle est regroupée en trois catégories qui sont la fabrication de la puce, la caractérisation et test de la puce, et enfin la

production. La fabrication de la puce consiste à dessiner à la main le design (dessin des transistors ou bien des cellules logiques pré-caractérisées telles que les portes ET, OU, MUX, FF, etc). La caractérisation consiste à définir les caractéristiques physiques de chaque cellule : la longueur d'un transistor, le type, les tensions d'alimentation et de seuil, etc. Le test consiste à vérifier à l'aide des outils de CAO (Conception Assistée par Ordinateur) si le masque dessiné réalise la fonction désirée. Enfin l'étape de production consiste à réaliser le wafer (carte en silicium) à partir du masque. En fonction de la technologie choisie, le prix du masque d'un ASIC varie de quelques centaines de milliers d'euros à 1 million d'euros, ce qui fait qu'il devient vite inaccessible lorsque l'on souhaite faire simplement du prototypage et non de la commercialisation à grande échelle.

La nature flexible d'un FPGA lui permet d'être reconfiguré à l'infini et a pour répercussion un coût significatif en matière de zones d'utilisation, de temps de propagation et de puissance consommée. Un FPGA utilise approximativement 20 à 35 fois plus de place qu'un ASIC, a des performances en termes de temps de propagation 3.5 fois plus faible, et consomme en moyenne 14 fois plus d'énergie (puissance dynamique) [20].

En dépit des avantages d'un ASIC en matière de performances, le temps de fabrication ainsi que le prix et surtout l'aspect non flexible font que les FPGA deviennent une bonne alternative pour l'implémentation des systèmes numériques et en particulier pour le prototypage. En effet, le prix moyen d'un FPGA est d'une centaine d'euros et la programmation est très rapide. De plus, la capacité des FPGAs progresse très rapidement avec le niveau d'intégration offrant une importante puissance de calcul. C'est pourquoi nous nous basons essentiellement sur des FPGA dans cette étude.

## 1.4 Techniques de réduction de la consommation

Comme nous l'avons vu dans la section précédente, les principaux paramètres influant sur la consommation d'un circuit sont l'activité des transistors, la surface, la tension d'alimentation et la fréquence d'horloge. Dans cette partie, nous traiterons principalement des techniques permettant de réduire la consommation dynamique, néanmoins certaines d'entre elles offrent un gain sur la consommation totale (statique et dynamique). De nombreuses techniques permettant de réduire la consommation existent dans la littérature.

Le but de cette section est de fournir une vue d'ensemble des techniques de réduction de la consommation de puissance dans les circuits intégrés intervenant aux différents niveaux d'abstraction allant du niveau circuit (transistor) au niveau algorithmique, comme le montre la

Fig. 1.8. La cible visée pour notre étude étant un FPGA, les références données dans la suite concernent essentiellement des FPGAs.

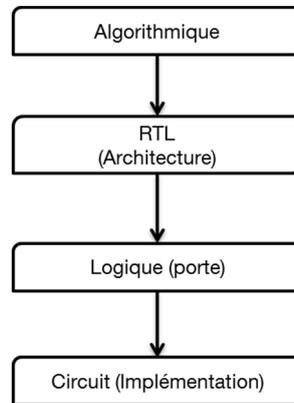


FIG. 1.8 : PRINCIPAUX NIVEAUX D'ABSTRACTION

## 1.4.1 Niveau Circuit

### 1.4.1.1 Voltage scaling (ajustement de la tension)

La tension d'alimentation a un impact significatif sur la consommation de puissance. On remarque dans l'équation (1.8), que la puissance dynamique est proportionnelle au carré de la tension. Ainsi une réduction de la tension d'alimentation du circuit (ou dans certaines zones du circuit) produit un gain conséquent. Cependant, baisser la tension d'alimentation a un impact sur la fréquence de fonctionnement maximale du système car cela entraîne une réduction de la tension de grille et donc une augmentation des temps de propagation (vitesse de commutation des transistors) comme le montre l'équation suivante [21].

$$T_c = C \cdot \frac{V_{dd}}{(V_{dd} - V_{th})^2} \quad (1.10)$$

avec  $T_c$  le temps nécessaire à la commutation d'un transistor,  $C$  la capacité d'un transistor,  $V_{dd}$  la tension d'alimentation et  $V_{th}$  la tension de seuil.

Un compromis doit alors être trouvé entre la fréquence de fonctionnement du système, qui impacte sur les performances, et la consommation, c'est à dire la tension d'alimentation dans ce cas. Cette technique est plus connue sous le nom de « voltage scaling ».

La technique du voltage scaling pose un second problème car en réduisant la tension de seuil, on augmente de fait les courants de fuite et donc la puissance statique. Il convient donc de prendre en

compte le rapport de gain entre la puissance dynamique et la puissance statique lors de l'utilisation de cette technique.

### 1.4.1.2 Surface

Une deuxième solution consiste à diminuer les capacités, en d'autres termes la surface utilisée par le circuit, appelée taille du circuit. Bien choisir le matériel est donc essentiel afin de respecter les budgets énergétiques.

## 1.4.2 Niveau Logique

### 1.4.2.1 Clock gating (ajustement de l'horloge)

La technique de contrôle de l'arbre d'horloge du circuit (clock gating en anglais) consiste à stopper la propagation de l'horloge sur une partie du circuit. Comme nous l'avons vu dans les sections précédentes, le système ou du moins les blocs IP (Intellectual Property) ne sont pas toujours actifs. L'idée est donc de stopper l'activité en stoppant la propagation de l'horloge dans ces blocs afin de ne plus avoir de consommation dynamique.

La mise en œuvre de cette technique est en général assez simple à réaliser en pratique, il suffit de contrôler l'arrivée de l'horloge par un signal logique de contrôle appelé Enable qui a pour rôle d'annuler la propagation de l'horloge et ainsi d'empêcher toute activité. La Fig. 1.9 illustre la mise en œuvre de cette technique. De cette manière l'horloge n'arrive plus directement sur le bloc IP mais est liée au bloc « contrôle » qui décide de la sortie en fonction de la valeur de l'Enable.

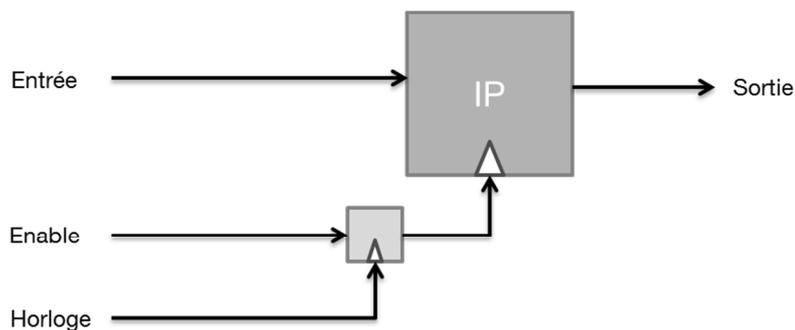


FIG. 1.9 : *TECHNIQUE DU CLOCK GATING*

La technique du clock gating entraîne des gains importants sur la consommation de la puissance dynamique pouvant aller jusqu'à 65% [22]. Cependant, cette technique influence légèrement la puissance statique car l'ajout d'un bloc de contrôle devant chaque bloc IP augmente le nombre de

transistors et donc la puissance statique. La surface est impactée par cette technique, il est donc primordial d'analyser les répercussions sur ce point et d'évaluer le rapport de gain que l'on obtient.

#### 1.4.2.2 Power gating (ajustement de la tension)

La technique du « power gating » consiste à couper l'alimentation de plusieurs parties du circuit lorsqu'elles ne sont plus utilisées. Cette technique agit directement sur la consommation totale du circuit (puissance statique et dynamique). Sa mise en œuvre est assez complexe au niveau logique, une méthode consisterait à placer des interrupteurs sur chaque partie du circuit visée par cette technique. Cependant, cette manière de faire a un coût significatif en matière de délais et surtout de surface. Un second problème apparaît lors de la coupure de l'alimentation si le circuit contient des mémoires volatiles. Toute coupure d'alimentation sans sauvegarde au préalable des mémoires ou même de l'état des bascules entraîne une perte d'information. L'unique solution est de sauvegarder les données puis de les restaurer au moment de la mise en route de l'alimentation. Cependant cette solution engendre un coût en consommation non négligeable. La solution la plus judicieuse est d'utiliser cette technique dans le cas où le traitement est terminé, ainsi la reprise du fonctionnement est réalisée à l'état initial.

### 1.4.3 Niveau RTL et architectural

#### 1.4.3.1 Ajustement dynamique de la tension d'alimentation et de la fréquence

La technique du voltage scaling vue plus haut a permis de mettre en lumière le lien existant entre la tension d'alimentation et la fréquence de fonctionnement. Ainsi, si nous baissions la tension d'alimentation, la fréquence de fonctionnement maximale est automatiquement impactée. En se basant sur ce lien, d'autres techniques ont été réalisées au niveau architectural.

La technique du Dynamic Voltage and Frequency Scaling (DVFS en anglais) consiste à ajuster dynamiquement la tension d'alimentation et donc la fréquence de fonctionnement du système. L'idée principale est d'obtenir un gain en énergie en abaissant au maximum la fréquence de fonctionnement. Afin de garantir des performances optimales, ce changement doit être réalisé en cours de fonctionnement afin que le système soit toujours dans de bonnes conditions notamment en respectant les contraintes de temps et de performance. La Fig. 1.10 illustre la technique du DVFS, et schématise le début et la fin d'exécution d'une tâche. On remarque que sans la méthode

DVFS, la tâche est exécutée très rapidement alors qu'il reste un temps conséquent ( $T_{restant}$ ) avant le temps limite d'exécution ( $T_{limite}$ ). Le deuxième cas illustre l'exécution de cette tâche avec la méthode DVFS. Dans ce cas, la tension d'alimentation ( $V_1$ ) et donc la fréquence de fonctionnement ( $F_1$ ) est réduite ( $V_2, F_2$ ) de sorte que la tâche utilise tout le temps mis à disposition. De cette manière les performances du système ne sont pas impactées mais on obtient un gain en énergie significatif ( $P_1 > P_2$ ). Dans [23], les auteurs ont implémenté cette technique sur des opérateurs arithmétiques (multiplieurs et additionneurs). Ils obtiennent des gains en énergie totale de plus de 50% par rapport à la même opération sans utilisation de la méthode DVFS.

La technique du DVFS ne fonctionne que sous certaines conditions. Il faut en effet que les temps de calcul de certaines tâches soient inférieurs à la durée d'un cycle.

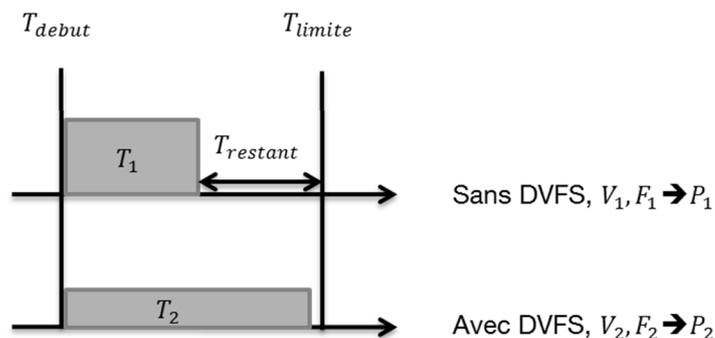


FIG. 1.10 : *TECHNIQUE DVFS*

### 1.4.3.2 Parallélisation et pipeline d'architecture

Comme on vient de le voir, la technique du DVFS, par la diminution de la tension d'alimentation, implique une diminution de la fréquence de fonctionnement du système et donc naturellement une augmentation des temps de calculs, comme le montre la Fig. 1.10 où  $T_2 > T_1$ . Une solution bien connue des concepteurs est donc de paralléliser, ou de pipeliner, les architectures qui nécessiteraient de grands temps de calculs. De cette manière, on peut abaisser la tension d'alimentation et donc la fréquence de fonctionnement du système, tout en respectant les contraintes de temps et de performance.

L'architecture d'un bloc IP peut être pipelinée ou non pipelinée. Une architecture pipelinée consiste à traiter plusieurs données en parallèle, cela signifie qu'il n'est pas nécessaire d'attendre la fin d'un traitement avant d'en commencer un autre, comme le montre la Fig. 1.11; alors qu'une architecture sans pipeline doit traiter les blocs de données les uns après les autres. Dans le cas

d'une architecture pipelinée, le chemin critique est divisé par deux. De cette manière et dans l'optique d'utiliser la technique du DVFS ou plus simplement la technique du voltage scaling, la tension d'alimentation peut être baissée. En effet, si la fréquence maximale du système est diminuée alors on peut diminuer la tension d'alimentation du système. En gardant la fréquence de fonctionnement initiale du système, les temps de calculs restent identiques, donc les performances sont les mêmes, mais la consommation est plus faible du fait de la diminution de la tension.

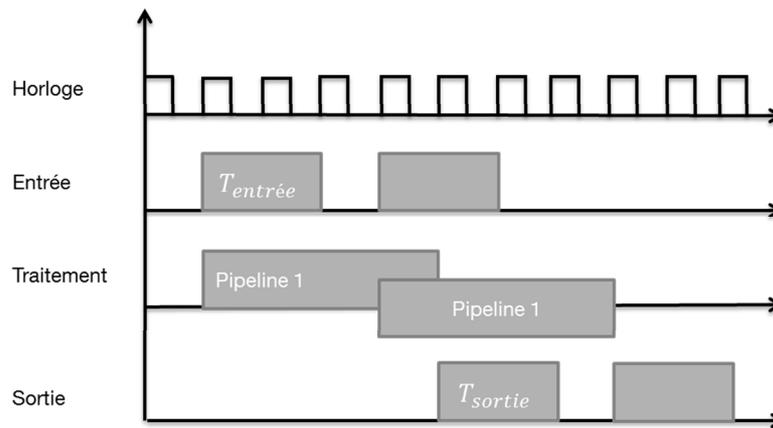


FIG. 1.11 : ARCHITECTURE PIPELINE

La parallélisation des données consiste à simplement dupliquer une architecture, le fonctionnement des deux architectures se réalise en parallèle comme illustré Fig. 1.12. De cette manière, le temps de calculs est divisé par deux. En divisant la tension d'alimentation et la fréquence de fonctionnement par deux, le temps de calculs restera identique à l'architecture initiale.

Cependant ces deux méthodes ont un coût en surface non négligeable. Il est donc primordial de comparer les gains qu'elles procurent avec l'augmentation de la puissance (notamment statique) liée à la surface, c'est-à-dire due à l'ajout des transistors nécessaires à la mise en œuvre ces deux méthodes.

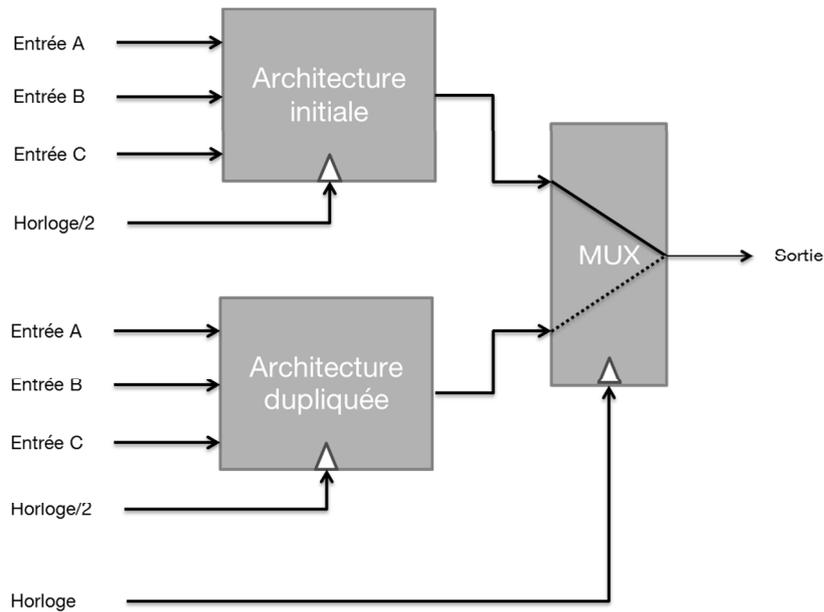


FIG. 1.12 : PARALLELISATION D'ARCHITECTURES

### 1.4.3.3 Précision des données

Une solution assez simple pour diminuer la puissance de consommation est d'optimiser la précision des données pour une tâche, ou bien d'utiliser de faibles précisions de calcul lorsque le cahier des charges le permet. En effet, réduire la précision des données permet de réduire la complexité des calculs et la surface. Dans ce cas un compromis doit être fait entre précisions des données, niveau de performances attendu, et consommation de puissance.

### 1.4.4 Niveau algorithmique

Si les méthodes citées précédemment permettent de réduire la consommation de puissance dans les circuits jusqu'à 40%, les techniques de réduction réalisées au niveau système permettent des gains encore plus importants. Parmi les nombreuses techniques possibles, nous allons ici présenter celles qui sont les plus répandues dans la littérature.

Comme nous l'avons décrit dans les sections précédentes, l'architecture d'un FPGA se compose principalement de CLBs, d'interconnexions programmables, des blocs d'entrées/sorties, des ressources mémoires embarquées et des blocs logiques figés. Si l'on veut diminuer la puissance dissipée par un FPGA à haut niveau alors il est nécessaire de diminuer chacune des puissances dissipées par ses ressources. Dans la suite de cette section, nous allons détailler les optimisations

---

possibles pour les trois principales ressources énergivores présentes dans la structure du FPGA qui sont les CLB, les interconnexions programmables, et les mémoires.

#### 1.4.4.1 CLB

La puissance consommée par les CLBs sur un FPGA varie en fonction de la complexité du système implémenté. En général celle-ci atteint 25% à 35% de la puissance totale [24].

Dans les travaux de [25], les auteurs utilisent une technique de pré-calcul afin d'optimiser la logique en haut niveau qui a des répercussions directes sur la puissance dynamique. L'idée est d'examiner les entrées d'un module du circuit à chaque cycle d'horloge afin de déterminer sous certaines conditions quels modules peuvent être désactivés. Les auteurs prennent l'exemple d'un comparateur 32 bits à deux entrées A et B. Si  $A > B$  la sortie est à 1, sinon elle est à 0. La technique consiste à vérifier préalablement si les entrées A et B ont changé, si oui alors on effectue la comparaison et le module fonctionne normalement, sinon on désactive le module tout en maintenant la sortie. De cette manière les auteurs obtiennent une réduction maximale de la consommation de puissance de la logique de 83%. Cette technique a tout de même un désavantage qui est la complexité d'implémentation et surtout le coût en surface dû à l'ajout devant chaque module d'un bloc de pré-calcul.

Une autre technique dans le même contexte que la technique précédente consiste à limiter le nombre de transitions inutiles. Les auteurs dans [26] considèrent à juste titre que l'écrasement (pour réécriture) d'un registre avec la même valeur se produit fréquemment, générant autant de transitions inutiles impliquant de l'activité, et donc de la consommation. La solution proposée est de ne plus alimenter, via l'horloge, les bascules D (pour données) mais seulement d'alimenter les LUTs. Ainsi, la bascule sera alimentée seulement si une entrée LUT a changé. De cette manière, les auteurs obtiennent une réduction de 75% de la puissance dynamique et de 15% de la puissance totale.

Dans [27], les auteurs ont pour but de réduire l'activité de commutation qui apparaît dans la formule de la puissance dynamique à l'équation (1.8). L'idée est de profiter des propriétés de commutations des opérateurs logiques ('A or B' est égale à 'B or A' par exemple). Ainsi, les données sont réorganisées de sorte qu'il y ait des chemins séparés pour l'état logique 0 et l'état logique 1. Le Tab. 1.2 permet de mieux comprendre la méthode employée. On remarque que dans le cas original, les données  $D_0$ ,  $D_1$  et  $C_{in}$  changent souvent d'état alors que dans le cas réorganisé  $D_0$  et  $C_{in}$  ont souvent la même valeur, 1 et 0 respectivement. Afin de calculer la réorganisation, les auteurs utilisent des formules spécifiant les combinaisons. Un gain maximum de 28% est

obtenu même si la surface du circuit afin d'implémenter cette méthode a augmentée de 11%. De plus un délai a aussi été introduit avoisinant les 18%.

Original			Réorganisé			Sortie	
$D_0$	$D_1$	$C_{in}$	$D_0$ nouveau	$D_1$ nouveau	$C_{in}$ nouveau	Somme	$C_{out}$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0
0	1	0	1	0	0	1	0
0	1	1	1	1	0	0	1
1	0	0	1	0	0	1	0
1	0	1	1	1	0	0	1
1	1	0	1	1	0	0	1
1	1	1	1	1	1	1	1

TAB. 1.2 : OPERATION "ADD" REORGANISEE AFIN D'AVOIR DES CHEMINS SEPARES DE 0 ET 1

#### 1.4.4.2 Interconnexions programmables

La plus grande partie de la puissance dissipée par les FPGAs est due aux interconnexions programmables réalisées afin de relier toutes les ressources entre elles (CLB, entrées/sorties E/S, mémoire, etc.). La puissance consommée due au routage atteint dans certains cas plus de 60% de la puissance consommée totale par le circuit [24][28][29].

La solution la plus répandue est d'améliorer la localisation des données, appelée « clustering » en anglais. Un cluster est un regroupement de modules afin de faciliter les communications. Un travail approfondi a été réalisé par les auteurs dans [30] afin d'évaluer la taille du cluster en fonction de la puissance consommée par les interconnexions, par la logique et par l'horloge du circuit. La taille d'un cluster est définie par le nombre d'éléments logiques de base (LUT et FF) qu'il contient. En comparant des tailles de clusters différents (4, 8 et 12 éléments logiques), les auteurs montrent que les grandes tailles de clusters sont moins énergivores que les petites tailles. Cela est dû au fait que les grandes tailles de clusters aident à réduire l'usage des ressources d'interconnexions entre les différents blocs logiques et ainsi réduire la puissance consommée par les interconnexions. Cependant, quand la taille du cluster augmente, la consommation liée à la

logique augmente aussi. Les auteurs abordent un autre point qui est la taille des entrées de chaque LUT dans les blocs logiques de base (variant de 3 à 7 entrées). Ils montrent que plus la taille des entrées des LUTs augmente, plus le pourcentage de la puissance consommée par les interconnexions ainsi que l'horloge diminue, car moins de routage à l'intérieur du cluster est nécessaire. Cependant, la consommation liée à la logique augmente avec l'augmentation de la taille des LUTs. Néanmoins, puisque la consommation des interconnexions est plus importante que la consommation de la logique, le gain obtenu est tout de même significatif.

Comme on vient de le voir, la technique du clustering a des effets bénéfiques sur la consommation de puissance, de plus cette technique influence aussi les performances. Les auteurs [31] montrent que la technique du clustering peut également améliorer les performances des FPGAs.

La technique du « voltage scaling » sur bus peut aussi être utilisée afin de diminuer la consommation de puissance. Afin de ne pas dégrader les performances du système, il est important de vérifier que la charge des données transitant sur les interconnexions pour réaliser cette technique n'est pas trop élevée. Dans [32], les auteurs utilisent la technique du DVS sur les liens (interconnexions), le but étant d'ajuster dynamiquement la tension et la fréquence des liens en se basant sur l'utilisation passée de ce même lien. L'idée générale simplifiée est de mesurer la charge du lien et de ses entrées. Ainsi en prenant en considération l'utilisation précédente du lien et ce qui s'y passe à l'instant  $t$ , les auteurs ajustent la tension d'alimentation du lien. En moyenne les auteurs obtiennent des résultats permettant de diminuer d'un facteur 4 la puissance consommée par les interconnexions. Une petite réduction du débit est constatée (2.5%). Cependant cet impact sur les performances est modéré comparé aux importants gains en puissance obtenus.

#### 1.4.4.3 Mémoires

La consommation des mémoires dans les circuits modernes est devenue un axe de recherche important. La consommation statique occupe une part significative atteignant les 80% pour certaines applications, notamment lorsque les mémoires sont de grande taille et que l'activité du circuit est faible. Les courants de fuite liés aux mémoires sont dès lors non négligeables du fait du nombre important de transistors utilisés.

Les FPGAs intègrent souvent deux types de mémoires :

- Une mémoire de type SRAM (Static Random Access Memory), cette mémoire est volatile, cela signifie que lorsque l'alimentation est arrêtée, toutes les données contenues dans ce type de mémoire sont perdues.
- Une mémoire externe non-volatile, qui permet de préserver les données même en cas d'arrêt de l'alimentation. Généralement, le bitstream (fichier d'implémentation), contenant toutes les informations liées à l'implémentation du système est stocké sur une mémoire non volatile de type flash.

L'utilisation de deux types de mémoires permet d'utiliser des techniques pouvant stopper la tension d'alimentation telles que le « power gating » vu à la section précédente.

De nombreuses méthodes de réduction d'énergie des mémoires SRAM ont été discutées. Dans [33], les auteurs ont montré qu'en modifiant la configuration des cellules SRAM et en prenant en compte certaines caractéristiques, l'obtention d'un gain conséquent sur la puissance statique pouvait être obtenu (2 fois moins élevée que dans le cas des mémoires SRAM conventionnelles). L'idée consiste à simplement modifier la taille du canal (entre la source et le drain) des transistors présents dans chacune des cellules, alors qu'en général toutes les tailles de canaux dans une même cellule sont égales. Les travaux de [34] proposent de remplacer les mémoires de types SRAM par des mémoires RRAM (Resistive RAM) pour palier au problème de la volatilité. Ainsi, le circuit ne contiendrait plus qu'une mémoire non volatile et contribuerait à réduire la consommation. De plus ce changement permet de gagner en surface car les cellules de configuration des RRAM sont plus petites que les cellules SRAM.

Il est à noter que les mémoires de type RRAM émergent et deviennent des candidats potentiels afin de remplacer les mémoires flash (mémoire non volatile), notamment grâce à leur haute densité d'intégration et leur faible tension [34].

## 1.5 Conclusion

Dans ce chapitre, nous avons présenté les différentes puissances consommées par les circuits intégrés. Premièrement, la puissance statique dépend largement de la technologie employée par les constructeurs, et est due aux courants de fuite qui surviennent lorsque le système est inactif. Deuxièmement, la puissance dynamique est créée par l'activité des transistors et donc dépend de l'architecture développée.

Nous avons vu que l'amélioration de la technologie (surface inférieure à 70 nm) conduit, lorsque l'activité du système est faible, à ce que la puissance statique soit plus importante que la puissance dynamique. Afin d'estimer le rapport entre ces deux puissances, une évaluation de la puissance statique et dynamique sera effectuée sur carte FPGA au chapitre 3. La puissance statique étant dépendante du matériel (nombre et caractéristiques des transistors), nous travaillerons principalement sur la puissance dynamique dans la suite du manuscrit car elle peut être réduite au niveau architectural par le développeur.

Une brève introduction à la structure architecturale et au fonctionnement des circuits FPGA a été faite. Ainsi nous avons décrit la répartition architecturale d'un FPGA qui se décompose en 5 grandes parties (CLB, connexion programmable, bloc d'entrées/sorties, bloc mémoire et bloc de calcul figé) et qui permet de configurer et de reconfigurer le circuit à l'infini.

La différence avec le circuit ASIC a été abordée, ainsi, même si le FPGA est moins performant, la nature flexible de celui-ci fait qu'il devient une très bonne alternative lorsque le but est de faire du prototypage rapide d'architecture. De plus, avec les avancées technologiques, les FPGAs s'imposent de plus en plus dans le monde des circuits car leurs puissances de calcul s'améliorent sans cesse ce qui permet d'avoir des performances tout à fait acceptables.

La puissance de calcul fournie par les FPGAs a un coût au niveau énergétique. Cependant différentes techniques peuvent contribuer à limiter la dissipation de puissance. Pour une réduction de puissance optimale, nous avons vu qu'il faut agir au plus tôt du flot de conception. Parmi les techniques décrites précédemment, nous nous intéresseront principalement à celles qui traitent de l'architecture des blocs dans la suite du manuscrit. Nous comparons notamment deux architectures de FFT au chapitre 5, afin d'optimiser la consommation énergétique.

Avant d'analyser les chaînes de transmission de communications sans fil en détails, nous allons voir dans le chapitre 2 quelles sont les méthodes habituelles d'estimation de la consommation dans les circuits intégrés.



## Chapitre 2

# Plateforme d'évaluation de la consommation énergétique

### 2.1 Introduction

Nous avons vu au chapitre précédent qu'il est préférable d'estimer la consommation de puissance au plus tôt dans la phase de développement d'un système électronique, notamment afin d'éviter les coûts de redéveloppement et les temps de simulations bas niveau très longs.

L'objectif de ce chapitre est de faire un état de l'art des principales techniques actuelles d'estimation de la consommation de puissance dans les circuits. Nous avons vu qu'un système électronique peut être considéré à différents niveaux d'abstraction allant du bas niveau (transistor) au haut niveau (algorithmique). Le concepteur dispose d'informations spécifiques à chaque niveau, qu'il exploite de façons différentes pour évaluer la consommation en puissance.

Nous présentons également la plateforme de tests utilisée pour l'estimation de la consommation qui repose à la fois sur des mesures et des simulations.

### 2.2 Etat de l'art des techniques d'estimation de la consommation

La puissance totale dissipée par un FPGA est la somme de deux composantes, l'une statique et l'autre dynamique, comme indiqué à l'équation (1.2). Les équations (1.8) et (1.9) peuvent être utilisées afin d'estimer la puissance dissipée par le circuit, cependant leur utilisation est assez complexe à mettre en œuvre. Premièrement, des informations sur le matériel sont nécessaires, comme la surface utilisée (nombre de transistors) pour évaluer la puissance statique, ou encore

l'activité temporelle pour la puissance dynamique. Deuxièmement, ces formules sont liées à des modèles faisant intervenir des paramètres complexes à évaluer. La puissance statique est proportionnelle aux courants de fuite qui sont largement dépendants de la température par exemple.

Dans cette section, nous allons détailler les techniques d'estimation de la puissance dissipée par les circuits existantes à différents niveaux d'abstraction, c'est à dire à différentes étapes du flot de conception, en démarrant du niveau le plus bas (transistor) jusqu'au niveau le plus haut (algorithmique). Les estimations à bas niveau sont plus précises car tous les détails d'implémentation sont connus mais elles requièrent plus de temps et/ou de ressources, tandis que les estimations à haut niveau (algorithmique) sont plus rapides mais moins précises comme le montre la Fig. 2.1.

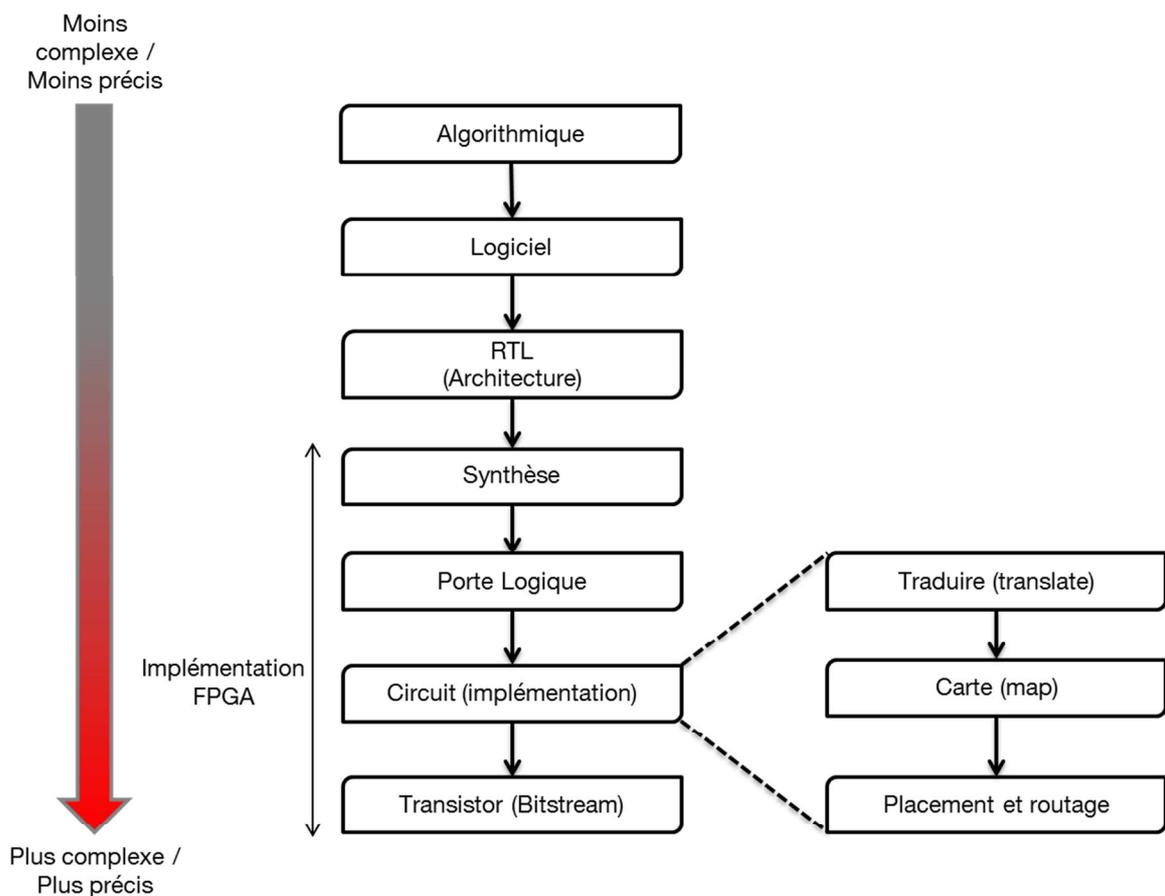


FIG. 2.1 : ANALYSE DE PUISSANCE - NIVEAUX D'ABSTRACTION

## 2.2.1 Evaluation de la consommation au niveau « transistor »

La méthode la plus simple pour estimer la puissance totale dissipée par un FPGA est de se baser sur sa représentation au niveau des transistors (après placement et routage du circuit). Le placement du circuit consiste à allouer les primitives (CLB, bloc mémoire, bloc de calcul figé) aux ressources du circuit. Quant au routage, cela consiste à déterminer l'organisation du réseau d'interconnexions du FPGA permettant de relier les primitives afin de réaliser la fonction désirée. À ce niveau d'abstraction le système est entièrement défini et une estimation très précise de la consommation peut être obtenue. Deux solutions sont alors envisageables, la première est de mesurer directement la consommation sur le circuit, la deuxième est de réaliser la simulation du système complet.

### 2.2.1.1 Mesure

Mesurer directement la consommation sur le circuit est la solution la plus précise possible afin d'obtenir la puissance consommée, car on relève en temps réel les tensions d'alimentation ainsi que les courants aux bornes du circuit. Cependant, pour réaliser cette mesure, une plateforme matérielle est nécessaire. Cela consiste à avoir l'équipement adéquat, à le configurer et le faire fonctionner avec des données d'entrées pertinentes afin d'obtenir la puissance dissipée. On retrouve dans l'état de l'art différentes plateformes permettant de mesurer la consommation d'un circuit. Dans [35], les auteurs implémentent leur système sur un Virtex-5 de chez Xilinx [36]. La puissance consommée est mesurée à l'aide d'une sonde qui récupère la tension et le courant aux bornes du FPGA. Les valeurs sont affichées sur un oscilloscope. Dans les travaux de [37], le système est implémenté sur un Spartan-6 de chez Xilinx. Les auteurs utilisent l'outil LabVIEW [38] qui permet de récupérer les informations d'alimentation à l'aide d'une sonde aux bornes du FPGA.

Dans les travaux de [39], une plateforme de mesure de consommation sur FPGA est décrite et permet de comparer différents algorithmes, ou différentes propositions de réduction de consommation. La plateforme matérielle est composée d'une carte de prototypage contenant un circuit FPGA Virtex-7 [40], d'un contrôleur d'alimentation, un port de programmation JTAG (Joint Test Action Group) ; un port USB/UART et une horloge principale de 200 MHz. Une interface utilisateur permet de contrôler le système et de relever la consommation mesurée. Enfin, des outils de mesure tels qu'une sonde pilotée par l'outil LabVIEW viennent compléter la plateforme. Un bloc BRAM permet de stocker les fichiers de données contenant les vecteurs de

tests fourni en entrée du système à mesurer. La plateforme permet d'avoir la puissance statique et dynamique du FPGA mais également la température.

Toutefois cette solution est compliquée à mettre en œuvre car elle nécessite énormément de temps de développement et de configurations du matériel; elle est en outre coûteuse. Pour remédier à cela, des outils permettant de simuler la puissance dissipée par le circuit existant, basés sur une simulation complète du système. Cette solution sera détaillée dans la prochaine section.

### 2.2.1.2 Simulation

La simulation du système a pour but d'obtenir l'activité temporelle du circuit afin d'estimer la consommation. Comme pour la mesure, le fichier d'activité en entrée permet de balayer les diverses possibilités de valeurs de données afin d'avoir l'activité moyenne du système lors d'un fonctionnement normal.

À ce niveau d'abstraction, des outils de simulation sont disponibles pour évaluer la puissance dissipée par les transistors. L'outil SPICE [41][42] développé par l'université de Berkeley caractérise le comportement des transistors, ce qui le rend très précis. Cet outil simule les performances électriques des transistors en fonction de leurs dimensions, de la tension d'alimentation et des substrats utilisés. L'outil XPower analyser [43] développé par Xilinx [36] permet également d'estimer la consommation moyenne d'un circuit FPGA. Cet outil nécessite un fichier d'activité en entrée, appelé vecteur de test, caractérisant l'activité temporelle du système. IRSIM [44] est également un outil d'estimation de la puissance des transistors. NANOSIM [45] est un logiciel utilisant des modèles simplifiés permettant d'estimer la consommation d'un petit circuit (<100000 portes logiques). Cependant, la précision est légèrement dégradée, de 3%.

L'utilisation d'outils de simulation peut conduire à des temps de simulation prohibitifs lorsqu'il s'agit d'explorer différentes techniques de réduction de consommation ou dans le cas d'exploration d'architectures. En effet, dans le cadre de la chaîne de transmission Wi-Fi étudiée au chapitre 3, nous avons obtenu des temps de simulation de l'ordre de la semaine. Ainsi, la complexité croissante des systèmes et la quantité de paramètres à prendre en compte font qu'il devient nécessaire d'estimer la consommation à plus haut niveau d'abstraction tout en maîtrisant la précision de l'estimation.

### **2.2.2 Estimation de la consommation au niveau « porte logique »**

Les simulations bas niveau ne sont pas toujours envisageables lorsque le système est complexe car les temps de simulation sont trop longs. Le niveau porte logique se situe juste après l'étape de synthèse du système, et contient moins d'informations sur l'implémentation que le niveau transistor. Les techniques d'estimation de puissance à ce niveau d'abstraction peuvent être regroupées en deux grandes catégories : basées sur une méthode statistique impliquant par exemple des tirages de Monte Carlo, ou sur une méthode probabiliste. La Fig. 2.2 présente le fonctionnement des deux méthodes que nous allons détailler dans la suite de cette section.

Les méthodes de simulation consistent à simuler un grand nombre de fois le système en utilisant des vecteurs d'entrées différents, afin de couvrir l'ensemble des valeurs des signaux internes et de sortie. Ces simulations ont pour but de récolter un certain nombre d'informations telles que l'activité temporelle du système. Les résultats de chaque simulation sont par la suite moyennés. Les vecteurs de test utilisés en entrée influencent grandement les résultats obtenus [46] [47], et différentes approches sont proposées dans la littérature afin de les générer. La technique de Monte Carlo, par exemple, les génère aléatoirement. Le problème est alors de savoir combien de simulations réaliser pour obtenir une estimation fiable. Dans [46], les auteurs proposent un nombre minimum de simulations à réaliser respectant l'erreur tolérée en fonction de la puissance moyenne renseignée par l'utilisateur. Ces méthodes nécessitent généralement un grand nombre de simulations, ce qui rend leur mise en œuvre peu efficace, surtout si le système est lui-même complexe.

Il est à noter que la technique de Monte Carlo permettant de générer les vecteurs de tests aléatoires peut aussi être utilisée au niveau transistor afin de générer le vecteur d'entrée qui va permettre de simuler le système complet.

Les méthodes probabilistes utilisent les propriétés probabilistes de type probabilité de transition des entrées, qui sont ensuite propagées à tous les signaux internes jusqu'à la sortie du système. L'avantage est que ces propriétés ne sont propagées qu'une seule fois à travers le système, et permettent de déterminer l'activité temporelle du système en une seule étape de calcul. Cette approche permet d'être beaucoup plus rapide que les techniques basées sur la simulation. Cependant, la précision de l'estimation est plus faible que pour la méthode statistique à cause de l'approximation faite sur les propriétés des entrées.

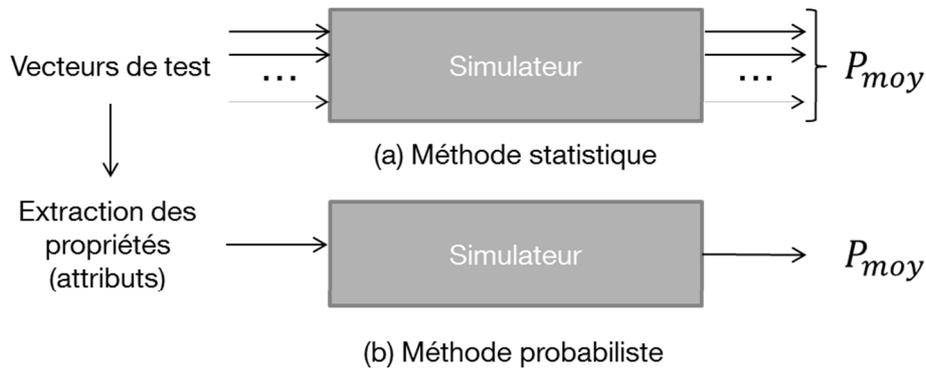


FIG. 2.2 : METHODES D'ESTIMATION STATISTIQUE ET PROBABILISTE

Pour accélérer l'estimation de la puissance, les techniques évoquées précédemment utilisent souvent des simplifications qui peuvent réduire le temps d'exécution au détriment de la précision [48]. Une liste des simplifications possibles est la suivante :

- Tension d'alimentation constante
- « Glitches » ignorés
- Courant de court-circuit et puissance statique ignorés
- Capacité du transistor constante, indépendamment du mode de fonctionnement

La première simplification suppose que la tension d'alimentation est constante sur le circuit, ce qui n'est pas toujours le cas en réalité. En effet, des chutes de tension peuvent se produire, causées par des variations d'intensité des courants sur les rails d'alimentation, elles-mêmes causées par des variations de charges. D'autre part, la commutation simultanée des transistors peut engendrer un bruit, et une réduction de la tension d'alimentation [49].

La deuxième simplification consiste à ignorer les « glitches » surtout lorsque la logique utilisée est combinatoire. Les glitches peuvent contribuer à une partie importante de la puissance totale dissipée par le FPGA et apparaissent lorsque les signaux d'entrée d'un circuit utilisant de la logique combinatoire possèdent des temps d'arrivées différents. Dans [50], les auteurs estiment que le pourcentage de la puissance dynamique dû aux glitches varie de 6 à 45%, avec une moyenne de 26%. De nombreuses techniques d'estimation de puissance ignorent la puissance liée aux glitches, car il est coûteux en matière de calcul de l'évaluer pour tous les signaux internes du circuit.

Certaines techniques d'estimation de puissance simplifient également leurs calculs en ignorant le courant de court-circuit durant la commutation des transistors, ou en supposant qu'il s'agit d'une

constante fixe de la puissance dynamique [51]. Or les auteurs dans [52] et [53] estiment que la puissance liée au courant de court-circuit est comprise entre 10 et 20 % de la puissance dynamique totale. Dans [54], les auteurs vont encore plus loin en annonçant que le courant de court-circuit est souvent sous-estimé et qu'il peut atteindre 30 à 40 % de la puissance dynamique totale.

Ces méthodes d'estimation sont difficiles à mettre en œuvre puisqu'il est nécessaire de disposer de la description précise du circuit au niveau logique. Bien que les simulations soient moins longues qu'au niveau transistor, elles n'en restent pas moins coûteuses en temps car plusieurs simulations doivent être réalisées. De plus, la complexité des architectures font qu'elles prennent de plus en plus de temps. Afin d'accélérer l'estimation de la puissance, d'autres méthodes d'estimation ont été proposées à un niveau d'abstraction plus élevé, le niveau RTL.

### **2.2.3 Estimation de la consommation au niveau « RTL »**

Au niveau RTL (Register Transfer Level), le système est simplement décrit en langage de description tel que VHDL ou Verilog. Les méthodes d'estimation de la consommation au niveau RTL nécessitent en général moins de temps de simulation qu'au niveau logique, au prix d'une baisse de la précision. Des modèles désignés sous le nom de « macro-modèles » ont été proposés [55] et ont pour but de modéliser la puissance des différents éléments formant le circuit tels que les opérateurs arithmétiques, les mémoires, les machines d'états, les bus d'interconnexions ou encore les blocs IPs [56]. Chaque macro dispose d'un modèle de consommation que l'on place généralement dans des tables de consommations contenant les valeurs de puissance dissipée pour chaque sous-élément. Généralement, l'activité de chaque macro est évaluée à travers des simulations comportementales afin d'affiner les modèles de puissance.

Dans [56], les auteurs proposent de calculer le nombre de ressources utilisées par chaque macro (multiplieur, multiplexeur, mémoire, etc.). Ainsi en utilisant des modèles de puissance et en calculant l'activité des macros en entrée et en sortie à travers une simulation, la puissance dissipée par le circuit peut être obtenue. Dans [57], des modèles analytiques sont utilisés afin de calculer la puissance dissipée par le circuit. Des modèles liant ainsi l'activité aux différents paramètres comme le nombre de ressources utilisées par les macros ou encore la largeur des bits de données sont développés en utilisant des techniques de régression linéaire et de courbes de tendance. L'activité est évaluée en propageant des probabilités de transition à travers le circuit. Les auteurs ont testé leurs méthodes sur 3 familles de FPGA (Virtex-5 et 6, Spartan-6) et obtiennent une erreur de 8% en moyenne sur la puissance totale en comparaison avec une estimation au niveau

logique. Dans [58], les auteurs proposent de calculer la puissance moyenne consommée en utilisant une technique d'estimation de la puissance basée sur l'activité, en spécifiant le nombre de transitions ainsi que la probabilité de tous les signaux. Un outil dédié à l'estimation de puissance RTL est utilisé et permet d'appliquer des paramètres d'activité de commutation aux signaux RTL. De la même manière, le nombre de transitions et les probabilités aux sorties de tous les signaux sont spécifiés par l'outil. Les auteurs obtiennent un écart de 4.4% en comparaison avec une estimation au niveau logique. Dans [59], l'auteur modélise chaque sous-élément du système (opérateur arithmétique, interconnexion, mémoire, etc.) en spécifiant les modèles analytiques préalablement construits. Ainsi l'erreur d'estimation de la puissance obtenue est inférieure à 15%.

On remarque que toutes ces méthodes nécessitent de connaître les modèles de puissance de chaque élément caractérisant le circuit et les ressources utilisées. Une complexité supplémentaire peut être ajoutée si les modèles doivent être créés préalablement en utilisant des techniques de régression linéaire par exemple. Néanmoins, des outils tels que RTLEst [60] permettent de créer des modèles analytiques en utilisant les techniques de régression linéaire de façon automatique.

Le développement des modèles ou encore la caractérisation des systèmes sont des limites conséquentes à l'estimation de la consommation à ce niveau d'abstraction. De plus l'utilisation de tables (LUTs) dans certaines méthodes est également un frein car leur utilisation peut impliquer d'importantes ressources mémoires lorsque le système est complexe. Le léger gain en temps de simulation peut ne pas compenser le manque de précision sur l'estimation de la puissance.

## **2.2.4 Estimation de la consommation au niveau « système ou algorithmique »**

### **2.2.4.1 Estimation de la consommation sans aucune information matérielle**

Au niveau système, la modélisation haut niveau du système est généralement décrite soit sous forme de spécification soit sous forme algorithmique. La spécification d'un système peut être décrite sous forme de tâches ou sous forme graphique (graphe d'état), tandis que la modélisation algorithmique est souvent réalisée sous forme d'instructions à exécuter.

En général à ce niveau d'abstraction, ce n'est pas la précision de l'estimation de puissance qui est le plus important mais plutôt la prise en considération par le concepteur, au plus tôt, de la

consommation du circuit. Cela lui permet d'évaluer rapidement le compromis à faire entre surfaces, vitesse et consommation. Les choix de conception effectués au plus haut niveau ont un impact maximum sur la consommation du circuit. De plus, c'est à ce niveau que les performances théoriques du système sont évaluées, d'où l'importance d'estimer également la puissance afin d'éviter les redéveloppements au cas où les contraintes énergétiques ne sont pas respectées.

On trouve dans la littérature de nombreux travaux se basant sur des modèles de puissance pour estimer la consommation du système. Dans les travaux de [61], des modèles analytiques sont associés à chaque élément d'un système de communication sans fils, tels que la consommation statique et dynamique d'un amplificateur de puissance, la consommation de l'étage RF ou encore la consommation statique et dynamique des blocs de traitement de signal. Ainsi en sommant la consommation de chaque élément de base formant le système, la puissance totale consommée est obtenue. Une approche similaire consiste à approximer la puissance statique en utilisant des expressions polynômiales. La consommation d'énergie est modélisée mathématiquement par des polynômes quadratiques, qui fournissent une formule analytique fonction des variations de la température par exemple. Les coefficients des polynômes sont obtenus en utilisant la méthode des moindres carrés suivant certaines contraintes. Les auteurs obtiennent une précision variant de 5 à 25 % par rapport à des mesures réelles [62].

Il est à noter que de nombreux fournisseurs de FPGA, comme Altera [63] et Xilinx ont développé des outils de CAO propriétaires afin d'estimer la puissance au niveau système à partir de feuilles de calculs. Xilinx a notamment développé Xilinx Power Estimator (XPE) [64] qui est un outil probabiliste d'estimation de puissance basé sur des formules analytiques. L'utilisateur doit néanmoins spécifier des paramètres globaux tels que la technologie cible visée, la température ou encore la tension d'alimentation. Pour une meilleure estimation, l'utilisateur peut également spécifier les différentes ressources utilisées telles que la fréquence, la logique et les entrées/sorties. La consommation d'énergie de ces ressources est directement proportionnelle à la fréquence d'horloge et à l'activité temporelle. Comme nous l'avons déjà mentionné, l'évaluation de l'activité temporelle des transistors peut être difficile à établir en particulier si le système est très complexe, et donc peu efficace si la précision de l'estimation est l'objectif prioritaire. Ces outils sont peu précis et atteignent un écart de 39% par rapport à une estimation au niveau transistor (XPower analyser) [65]. Une partie des erreurs est due au fait qu'aucune information issue d'une implémentation matérielle n'est connue. En ce sens, des techniques d'estimation de la consommation à haut niveau intègrent des informations matérielles afin d'avoir une estimation plus précise. Ces estimations sont détaillées dans la section suivante.

#### 2.2.4.2 Estimation de la consommation haut niveau en prenant en compte des informations issue d'une implémentation matérielle

Dans [66], les auteurs proposent de construire des modèles mathématiques pour chaque composant de la puissance dynamique (horloge, signaux, logique, entrées/sorties, etc.) en prenant en compte les variables des systèmes. Ainsi l'utilisateur peut effectuer des estimations de puissance à un haut niveau d'abstraction. Pour construire leur modèle de puissance, ils suivent 4 étapes, la première consiste à créer les tableaux de puissance pour chaque composant de la puissance dynamique, évalués au niveau transistor à l'aide de l'outil XPower. La deuxième étape consiste à identifier toutes les variables du système incluant les paramètres internes tels que la fréquence, la taille des vecteurs, la surface utilisée et la tension d'alimentation issus des informations d'implémentation. La troisième étape consiste à définir de manière empirique les relations existant entre les différentes variables du système. Enfin, la dernière étape consiste à extraire les coefficients pour chaque composant de la puissance dynamique en effectuant une analyse de régression non linéaire sur les données du tableau de puissance réalisé à l'étape 1. D'autres travaux estiment la puissance en se basant sur la corrélation existant entre l'activité des données en entrée, et la puissance globale du système. Les auteurs définissent la corrélation comme étant un indicateur linéaire de dépendance entre deux variables, ici l'activité et la puissance. Ils observent que la puissance consommée de leur système est corrélée à 96% avec l'activité. Ils définissent donc la formule linéaire suivante :  $P_{dyn} = ax + b$ , où  $x$  représente l'activité de commutation des données en entrée,  $a$  et  $b$  représentent des constantes qui sont fonction des données. Ainsi les auteurs affirment qu'il n'est pas nécessaire d'effectuer la simulation du système au niveau transistor afin d'obtenir la puissance dynamique, mais qu'il suffit de l'estimer à haut niveau. L'erreur moyenne observée est de 13% [66].

Des outils tels que PyGen [67], permettent de récupérer des informations des blocs IP ou des éléments d'opérateurs logiques issus d'une implémentation bas niveau (consommation, ressources utilisées par le système, etc.) et de les transmettre à des outils haut niveau capables d'estimer la puissance, tel que Matlab [68]. Néanmoins, cette manière de faire n'est pas d'une bonne précision et peut conduire à de grandes erreurs d'estimation car l'activité temporelle du système n'est pas prise en compte. Dans [67], les auteurs font l'hypothèse que tous les modules sont actifs, tout au long du fonctionnement du système. Or cette hypothèse est souvent erronée pour des systèmes réels. Dans les travaux de [69], on émet l'hypothèse que l'activité de commutation des transistors

peut être approchée par une moyenne, ce qui n'est pas non plus forcément le cas pour des systèmes réels.

Dans les travaux de [70] et [71], contrairement aux études précédentes, les auteurs simulent le système afin de prendre en compte l'activité réelle lors du fonctionnement normal. Leur méthode est de simuler un système de communications numériques en utilisant l'outil SystemC [72]. Préalablement, les auteurs construisent une bibliothèque contenant les valeurs de consommation de puissance de chaque bloc élémentaire (codeur canal, FFT, décodeur, etc.), les codes RTL associés ainsi que les modèles en haut niveau correspondants. Les valeurs de consommation de puissance sont préalablement évaluées au niveau transistor après placement et routage (bas niveau, en ayant connaissance en détails des informations d'implémentations sur le circuit). L'évaluation de la puissance est réalisée pour chaque bloc élémentaire à une fréquence donnée et pour une activité en entrée de 100% (bloc constamment actif). La technique de régression linéaire est utilisée en haut niveau pour cibler d'autres fréquences d'horloge si nécessaire. Enfin, l'outil SystemC permet de réaliser des simulations qui prennent en compte l'activité temporelle de chaque élément lors du fonctionnement de la chaîne. Les auteurs obtiennent un écart de 5% par rapport à une simulation au niveau transistor effectuée sous XPower Analyzer.

Pour résumer, un compromis doit être fait afin d'estimer la consommation de puissance d'un circuit entre la précision de l'estimation et le niveau d'abstraction. Les simulations bas niveau permettent d'avoir des estimations de puissance très précises mais il n'est pas possible de les réaliser dans la plupart des cas, principalement pour des raisons de temps et de complexité au vu des architectures modernes de plus en plus complexes. L'estimation à haut niveau est moins précise mais facilite la conception des systèmes car la prise en considération de la puissance se fait au plus tôt du flot de conception ce qui permet de respecter des contraintes de budget énergétique fixées au départ.

## **2.3 Plateforme de test utilisée : mesure et simulation**

Notre étude réalise à la fois des mesures réelles sur carte FPGA et des simulations pour évaluer le coût énergétique d'un système de communications numériques. Une chaîne de simulation a été développée dans un langage de description matériel (VHDL) pour évaluer la puissance du circuit. La consommation de chaque bloc est évaluée afin d'obtenir la répartition de la consommation électrique, à l'aide de l'outil XPower Analyzer de Xilinx. Parallèlement, une implémentation matérielle a été effectuée sur carte FPGA pour fournir des mesures réelles de la consommation

d'énergie. L'outil Fusion [73] permet d'obtenir le courant et la tension d'alimentation via une sonde, de sorte que la puissance totale, statique et dynamique, dissipée par le circuit soit calculée.

Nous détaillerons plus précisément dans la section suivante le flot de conception suivi afin d'analyser la consommation de puissance.

### **2.3.1 Simulation XPower**

Comme nous l'avons vu à la section 2.2.1.2, l'évaluation de la consommation par simulation au niveau transistor nécessite de simuler le système complet, et en particulier son activité temporelle. La Fig. 2.3 indique les étapes permettant d'obtenir une estimation de la consommation via l'outil XPower. Le module correspond à la description en VHDL du système à simuler. L'étape de synthèse consiste à passer de la description VHDL au circuit sous forme de portes logiques. Enfin, l'étape de placement et routage permet d'allouer les ressources du FPGA au circuit et de créer les chemins d'interconnexions afin de réaliser le système.

XPower permet d'obtenir la consommation de puissance des ressources du FPGA utilisées qui sont l'horloge, les signaux, la logique, les BRAMs, les DSPs et les entrées/sorties. Ces ressources forment la puissance dynamique. L'outil permet également d'évaluer la puissance statique du FPGA et ainsi d'avoir la puissance totale consommée.

Pour une estimation précise, XPower nécessite 3 fichiers d'entrée qui sont :

- Un fichier (.ncd) contenant les informations d'implémentation du système après placement et routage.
- Un fichier d'activité spécifiant l'activité temporelle du système (.vcd ou .saif). Ce fichier est obtenu à l'issue de la simulation du système à l'aide de l'outil ModelSim [74] et permet d'analyser l'activité de commutation des transistors. Pour une estimation précise de l'activité, il est nécessaire d'effectuer une simulation permettant de tester toutes les combinaisons d'entrée possibles. Ces données d'entrée sont basées sur un modèle statistique de Monte Carlo.
- Un fichier (.ucf) de contrainte spécifiant les contraintes imposées par l'utilisateur telles que la température cible, la fréquence de fonctionnement ou encore l'emplacement des entrées/sorties sur la carte.

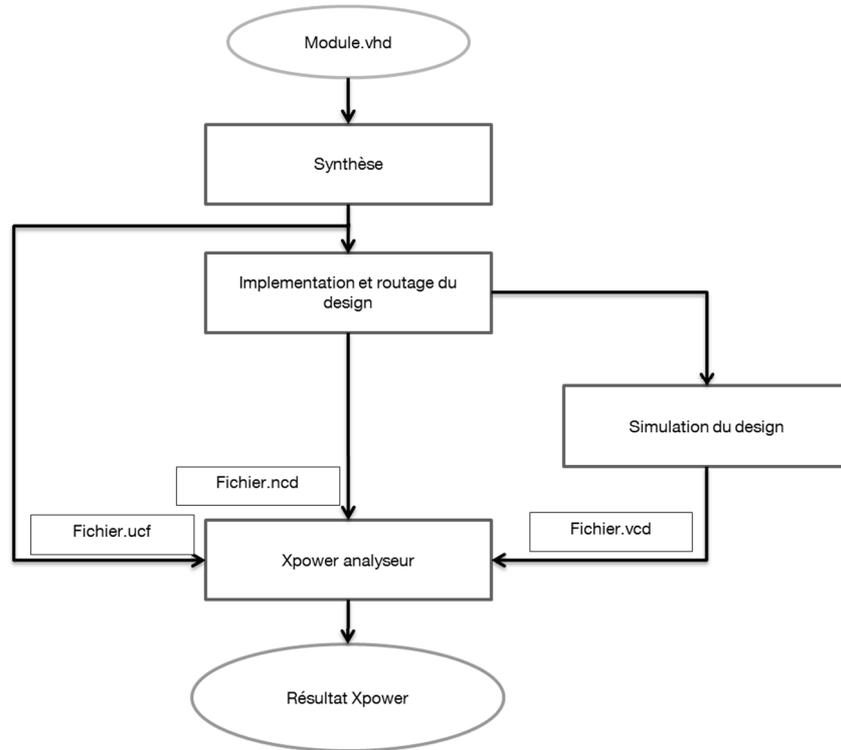


FIG. 2.3 : FLOT DE CONCEPTION XPOWER

### 2.3.2 Mesure

La mesure consiste à relever la puissance dissipée par le circuit directement sur la carte FPGA à l'aide d'une sonde. La carte ainsi que la sonde utilisée dans nos tests sont les mêmes que dans la plateforme [39] présentée à la section 2.2.1.1. La carte FPGA choisie est un Virtex 7, vc707 (xc7vx485r-2ffg1761) de Xilinx. Le Tab. 2.1 montre les ressources totales disponibles sur le FPGA. La plateforme est illustrée Fig. 2.4 montrant la carte FPGA ainsi que la sonde permettant de faire les mesures. Le flot de conception est proche de celui présenté à la Fig. 2.3. Néanmoins, à la suite du routage, il est nécessaire de générer un fichier appelé bitstream contenant toutes les informations d'implémentation (fichier de contrainte, placement et routage du système). Ce fichier est ensuite chargé sur le circuit et permet de configurer le FPGA. L'évaluation de la consommation est réalisée à l'aide de l'outil Fusion sur la plateforme matérielle, contrairement à la plateforme [39] qui utilisait l'outil LabVIEW pour récupérer les informations issues de la sonde telles que le courant, la tension d'alimentation ou encore la température du circuit. Nous avons choisi l'outil Fusion car c'est un outil développé pour travailler avec la sonde utilisée. Cela permet notamment de ne réaliser aucune étape de paramétrage. L'outil détecte et récupère

automatiquement les informations issues de la sonde. Ainsi, une mesure temps réel de la puissance dissipée peut être réalisée.

Les données d'entrée de la chaîne sont issues d'un bloc générateur de séquences aléatoires PRBS (en anglais, Pseudo Random Binary Sequence). Ce bloc permet de générer des données d'entrée pseudo-aléatoires où la probabilité d'avoir un bit à l'état haut est de 50%. Ce point sera abordé plus en détails au chapitre 3.

La mesure effectuée sur la carte permet d'avoir la puissance totale dissipée (puissance dynamique et statique). Afin d'avoir uniquement la puissance statique, nous utilisons le bouton « reset » configuré sur la carte qui permet de stopper le flux de l'horloge. De cette manière, la puissance dynamique est nulle et nous mesurons uniquement la puissance statique du FPGA. Ainsi pour avoir la puissance dynamique, il suffit de soustraire la puissance totale mesurée à la puissance statique. Avec le matériel à notre disposition, le pas de mesure est de 10 mW contrairement à XPower Analyzer qui est de 1 mW. Cela peut entraîner des imprécisions de mesures, notamment pour les faibles niveaux de puissances; il serait envisageable d'obtenir une meilleure précision avec d'autres sondes. Cependant, comme nous le verrons dans la suite de ce document, nous avons recueilli suffisamment de mesures élevées pour valider notre méthodologie.

Tous les résultats de synthèse et de consommation présentés dans la suite du manuscrit seront réalisés par rapport à la cible matérielle présente sur la plateforme.

	Ressources disponible
Nombre de Slice Registers	607200
Nombre de Slice LUTs	303600
Nombre de Flip Flops (FF)	8930
Nombre de IOs	700
Nombre de Blocs RAM	1030

TAB. 2.1 : RESSOURCES DISPONIBLES SUR LE FPGA



FIG. 2.4 : PLATEFORME DE MESURE

## 2.4 Conclusion

Ce chapitre a permis de rappeler quelques techniques d'estimation de la consommation de puissance dissipée par les circuits, allant du niveau transistor au niveau algorithmique.

L'état de l'art est riche en modèles ainsi qu'en techniques de réduction de la consommation. Nous avons vu que c'est à un haut niveau d'abstraction que la prise en compte de la consommation de puissance est la plus efficace, mais aussi qu'à ce niveau de conception les choix ont un impact fort, aussi bien sur la consommation que sur les performances.

Nous avons montré qu'au niveau transistor (bas niveau), des outils tels que SPICE, POWERMIL ou encore XPower Analyser, permettent de calculer la consommation de puissance d'un circuit de manière très précise en prenant en compte toutes les informations d'implémentation. Cependant, avec des architectures toujours plus complexes, ces outils présentent l'inconvénient de demander de longs temps de simulation, de l'ordre du jour voire de la semaine. De plus, avec les multiples configurations à tester dans les cas des communications numériques, ces temps de simulation sont difficilement acceptables.

Au niveau algorithmique (système), les temps de simulation sont beaucoup plus courts, mais le manque d'informations sur l'implémentation pose d'énormes problèmes pour avoir une

estimation précise de la consommation. Nous avons vu qu'une solution crédible est d'amener des informations bas niveau au haut niveau afin d'avoir une meilleure estimation.

Notre objectif est d'évaluer de manière précise la consommation de puissance, en ce sens, une estimation de la consommation bas niveau sera traitée au prochain chapitre. Le développement et l'implémentation d'une chaîne de communications numériques sur FPGA seront réalisés. L'analyse de l'estimation de la puissance ainsi que la méthodologie suivie seront également détaillées.

Néanmoins comme nous l'avons vu, cette méthode d'estimation nécessite beaucoup de temps de simulation ce qui n'est pas acceptable dans le cas de notre étude puisque l'on souhaite comparer différentes architectures afin de diminuer la consommation totale de la chaîne. Nous proposerons donc une méthode d'estimation de la consommation à haut niveau en intégrant des informations issue d'une implémentation matérielle (niveau transistor). Pour élaborer notre méthode nous nous sommes inspirés des idées principales de chaque méthode d'estimation vue précédemment et principalement de ceux vu à la section 2.2.4.2 et principalement à celle décrite en [70]. Les auteurs évaluent la consommation des blocs isolés au niveau transistor et déterminent en simulation à haut niveau l'activité du système. Dans notre cas nous déterminons l'activité théoriquement par calcul.

Notre méthode sera explicitée plus en détails au chapitre 4 et permet d'atteindre un double objectif : une bonne estimation, et un gain significatif en temps de simulation.

## Chapitre 3

# Analyse de la consommation de puissance des blocs Wi-Fi

### 3.1 Introduction

Une problématique soulevée aux deux précédents chapitres est que la chaîne Wi-Fi est composée de blocs impactés au niveau énergétique différemment selon les paramètres système et l'architecture développée pour exécuter les algorithmes de traitement signal. Peu d'études dans l'état de l'art traitent de la consommation de puissance du Wi-Fi et les rares données chiffrées sont des résultats globaux incluant à la fois la puissance circuit et la puissance à l'antenne, et sont peu précis. Le projet GREENCoMM [5], avait ainsi pour objectif de réduire la consommation des technologies de connectivité du réseau local domestique, dont le Wi-Fi.

Dans ce chapitre nous proposons d'analyser la consommation de puissance statique et dynamique de la couche physique de la norme Wi-Fi 802.11ac dans le contexte de la plateforme FPGA présentée au chapitre 2.

L'évaluation de la consommation de la chaîne nécessite le développement et la validation d'une chaîne Wi-Fi d'émission/réception en VHDL, implantable dans la cible FPGA incluse dans la plateforme. Nous présentons ainsi dans la section 3.2 la chaîne Wi-Fi et les blocs de traitement signal la composant ainsi que les architectures développées.

La consommation est évaluée pour la chaîne complète ainsi que pour chaque bloc. Nous pouvons ainsi identifier les blocs les plus gourmands en énergie. Pour cela, l'évaluation de la consommation met en œuvre d'une part la simulation au niveau transistor à l'aide de l'outil XPower Analyser, mais également par la mesure sur la plateforme FPGA. Une comparaison entre les résultats de simulation et les mesures est effectuée afin de valider la fiabilité des résultats

obtenus. De plus, nous évaluons également le temps de simulation nécessaire à l'obtention de l'estimation de la consommation de la chaîne complète.

Dans la suite, nous présentons tout d'abord les architectures développées pour la chaîne Wi-Fi en VHDL ainsi que leur validation. Les résultats de mesure et de simulation de la consommation de puissance statique et dynamique sont ensuite décrits et analysés.

Ces travaux ont fait l'objet de deux publications [75] et [76].

## 3.2 Contexte Wi-Fi

### 3.2.1 Chaîne d'émission/réception

Les principaux composants des chaînes Wi-Fi d'émission (Tx) et de réception (Rx) sont représentés Fig. 3.1, dans le cas où il y a une antenne côté émission, et une antenne côté réception (Single Input, Single Output, SISO). Seule la partie émission est spécifiée par la norme 802.11ac [6]. En effet, les récepteurs ne sont pas standardisés et peuvent contenir des algorithmes propriétaires développés par les constructeurs. Cependant nous pouvons émettre des hypothèses plausibles sur leur mise en œuvre. Le système est basé sur le multiplexage des données par répartition en fréquences orthogonales (Orthogonal Frequency Division Multiplex, OFDM), construit via des transformées de Fourier rapide directe et inverse (Fast Fourier Transform FFT, et Inverse Fast Fourier Transform, IFFT). Chaque bloc est détaillé dans la section suivante.

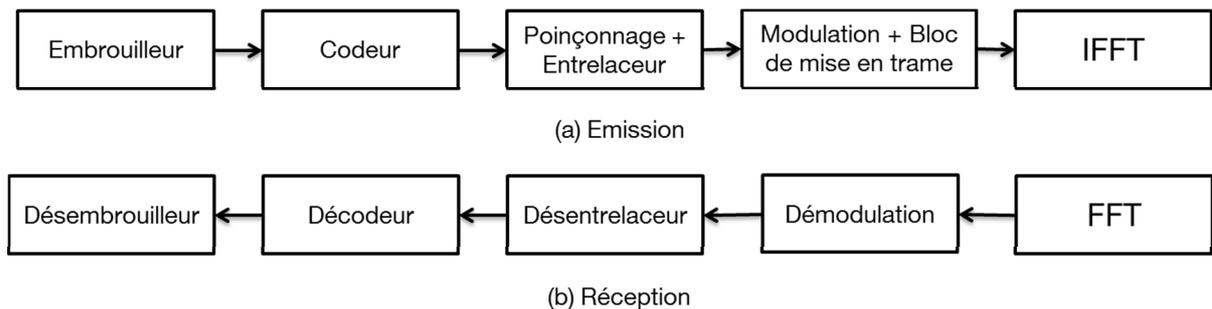


FIG. 3.1 : CHAINES D'EMISSION/RECEPTION

Les largeurs de bandes utilisées dans la norme 802.11ac sont 20, 40, 80 et 160 MHz. Le nombre de flux spatiaux varie de 1 à 8 en fonction de la largeur de bande utilisée. Dans la suite, nous traiterons le cas d'une antenne unique à l'émetteur et au récepteur. Le Tab. 3.1 décrit les différents schémas de modulation et de codage (Modulation and Coding Schemes, MCS) utilisés dans la norme.

MCS	Modulation	Taux de codage	Nb de bits par symbole
0	BPSK	1/2	1
1	QPSK	1/2	2
2	QPSK	3/4	2
3	16-QAM	1/2	4
4	16-QAM	3/4	4
5	64-QAM	2/3	6
6	64-QAM	3/4	6
7	64-QAM	5/6	6
8	256-QAM	3/4	8
9	256-QAM	5/6	8

TAB. 3.1 : SCHEMA DE MODULATION (MCS)

### 3.2.2 Description et architecture des blocs

#### 3.2.2.1 Embrouilleur

Le bloc « embrouilleur » permet d'effectuer une dispersion des données binaires de manière à éviter les longues chaînes de 0 ou de 1. Le but recherché est de produire une répartition uniforme temporellement des 0 et des 1. Pour cela, un polynôme générateur  $S(x) = x^7 + x^4$  est utilisé. L'architecture du bloc embrouilleur est réalisée en implémentant des registres à décalage comme le montre la Fig. 3.2.

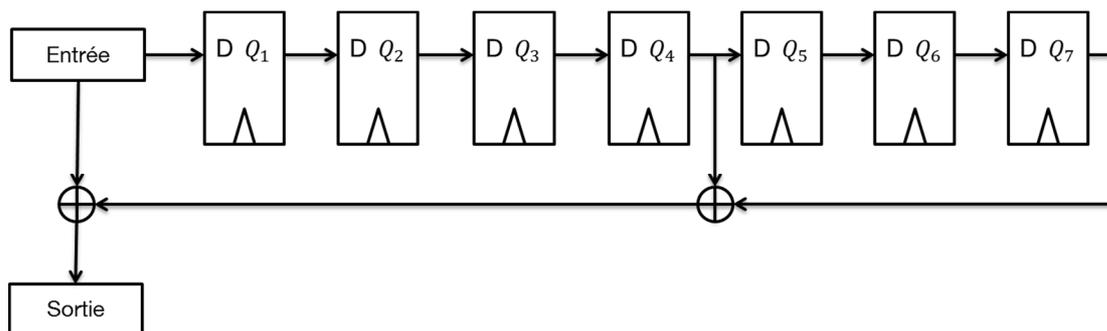


FIG. 3.2 : ARCHITECTURE EMBROUILLEUR REALISEE PAR DES REGISTRES A DECALAGE

#### 3.2.2.2 Codeur

Le bloc « codeur correcteur d'erreurs » permet d'ajouter de la redondance aux données afin d'être plus robuste face aux pertes engendrées par le canal. Le code convolutif utilisé dans la norme

802.11ac dispose de deux sorties déterminées par deux polynômes générateurs qui sont  $G_0 = 1338$  et  $G_1 = 1718$ , schématisés Fig. 3.3.

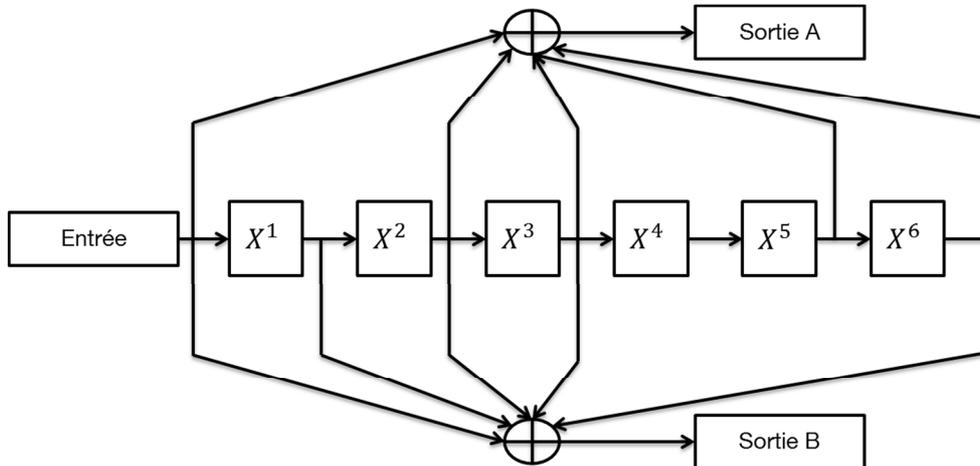


FIG. 3.3 : CODEUR

De la même manière que pour le bloc précédent, l'architecture est réalisée en implémentant des registres à décalage et deux portes OU exclusif.

Ce codeur a initialement un rendement de  $1/2$  appelé code mère, cependant, d'autres rendements sont possibles par le biais du poinçonnage. Comme représenté Fig. 3.1, le bloc poinçonnage a été fusionné avec le bloc entrelaceur. Différents taux de codage sont disponibles qui sont  $2/3$  et  $3/4$ . Le poinçonnage est un procédé qui permet d'omettre plusieurs bits codés. Cela a pour but de réduire le nombre de bits à transmettre et corollairement de transmettre davantage de données lorsque le canal est de bonne qualité. Plus on augmente le rendement plus on omet des bits ; par exemple, pour un rendement  $3/4$ , on omet deux bits pour 6 bits codés comme illustré Fig. 3.4.

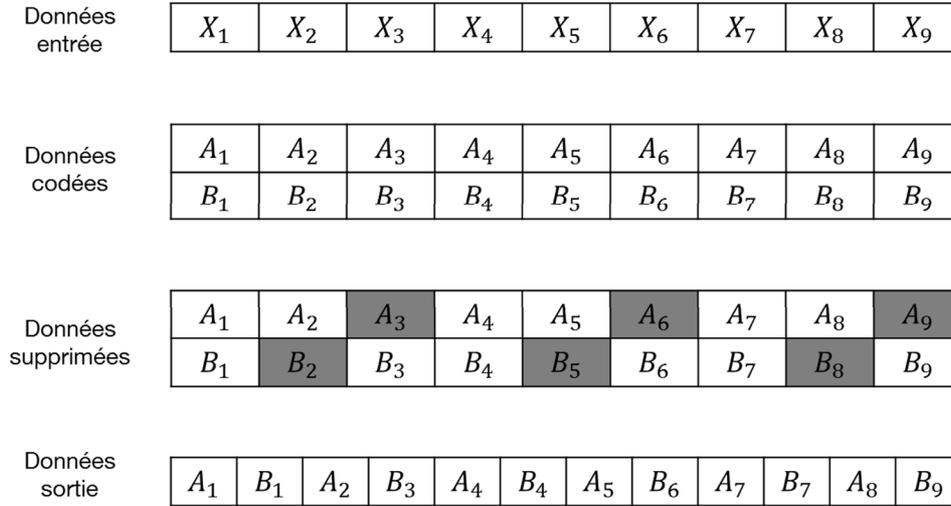


FIG. 3.4 : POINÇONNAGE 3/4

### 3.2.2.3 Entrelaceur

L'entrelaceur a pour but d'éviter les paquets d'erreurs en transmettant les bits consécutifs sur l'ensemble du spectre fréquentiel et non sur une partie limitée du spectre. Dans ce but, les données sont traitées par blocs de taille correspondant au nombre de données codées par symbole OFDM, noté  $N_{CBPS}$ . Les paramètres sont donnés Tab. 3.2 où  $K$  représente le nombre de bits par symbole complexe.

Largeur de bande	20 MHz	40 MHz	80 MHz
$N_{colonne}$	13	18	26
$N_{ligne}$	4 $K$	6 $K$	9 $K$
$N_{rotation}$	11	29	58

TAB. 3.2 : PARAMETRES DU BLOC ENTRELACEUR

L'entrelaceur est défini suivant 2 permutations dans le cas d'une chaîne SISO. Les deux permutations sont les suivantes :

$$i = (N_{CBPS}/16) (k \text{ modulo } 16) + (k/16) \quad \text{où } k = 0, 1, \dots, N_{CBPS} - 1 \quad (3.1)$$

$$j = s \times (i/s) + ((i + N_{CBPS} - (16 \times i/N_{CBPS})) \text{ modulo } s) \quad \text{où } i = 0, 1, \dots, N_{CBPS} - 1 \quad (3.2)$$

où  $k$  est l'indice de départ;  $j$  l'indice en sortie de l'entrelaceur, et  $s = \max(K/2, 1)$ . Pour implémenter ces deux équations, deux solutions architecturales ont été explorées.

La première solution (architecture 1) consiste à implémenter ces formules telles quelles, en réalisant les calculs. Cette solution n'est pas la plus optimale car le chemin critique est très long,

ce qui influence directement la fréquence d'horloge maximale. Une autre solution est alors de segmenter le chemin critique par des registres pour augmenter la fréquence d'horloge mais ce qui augmente la latence. La complexité en matière de calcul reste inchangée.

La deuxième solution (architecture 2) consiste à stocker en mémoire les indices de permutations pré-calculés. Ainsi, aucune phase de calcul n'est nécessaire lors du traitement des données.

La complexité matérielle des deux architectures est présentée Tab. 3.3. On remarque que l'architecture 2 utilise une mémoire supplémentaire par rapport à la première architecture. En revanche, elle n'effectue aucun calcul. Le tableau présente également les résultats de synthèse et l'évaluation de la consommation pour chacune des architectures dans le cas d'une largeur de bande de 80 MHz, pour une modulation 16-QAM. On remarque que l'architecture 1 utilise plus de LUTs et de FFs que l'architecture 2. Cela se traduit sur la consommation de puissance où l'on remarque que l'architecture 2 consomme deux fois moins que l'architecture 1.

Par conséquent, dans la suite de la thèse, l'architecture de l'entrelaceur implémentée sera l'architecture 2, offrant de meilleurs résultats énergétiques.

Complexité	Architecture 1	Architecture 2
Nombre d'additions	4	0
Nombre de multiplications	9	0
Nombre de blocs mémoire (taille $N_{CBPS}$ )	2	3
Nombre de Slice LUTs	821	731
Nombre de Flip Flops	838	739
Puissance dynamique	133 mW	71 mW

TAB. 3.3 : COMPLEXITE ET CONSOMMATION DES ARCHITECTURES D'ENTRELACEUR POUR UNE LARGEUR DE BANDE DE 80 MHZ ET UNE MODULATION 16-QAM.

### 3.2.2.4 Modulation

Les constellations définies dans la norme 802.11ac sont BPSK (Binary Phase Shift Keying), QPSK (Quadratic Phase Shift Keying), 16-QAM (Quadrature Amplitude Modulation), 64-QAM et 256-QAM. La conversion bit à symbole est effectuée selon un code de Gray.

L'architecture développée consiste à transformer  $K$  bits en un symbole complexe,  $K$  étant le nombre de bits par symbole complexe comme indiqué Tab. 3.4. L'ordre de la modulation veut alors  $2^K$ . Le bloc attend les premiers  $K/2$  bits pour générer la partie réelle du symbole complexe, les concaténer afin d'obtenir un entier correspondant à un indice dans un tableau contenant la valeur du symbole complexe correspondant. Le bloc génère la partie imaginaire à partir des  $K/2$

bits suivants de la même manière comme le montre le chronogramme Fig. 3.5 où P.R représente la partie réelle du symbole complexe et P.I la partie imaginaire. Sur ce chronogramme, nous supposons que les bits codés arrivent 2 par 2 à chaque cycle d'horloge. La taille du tableau est égale à  $2^{K/2} \cdot 2$  et permet de réaliser à la fois la partie réelle et imaginaire. L'avantage de cette architecture est de ne tester aucune condition, mais nécessite d'implémenter un tableau en mémoire supplémentaire de petite taille.

Modulation	Nb de bits par symbole $K$	Ordre de la modulation
QPSK	2	4
16-QAM	4	16
64-QAM	6	64
256-QAM	8	256

TAB. 3.4 : NOMBRE DE BITS PAR SYMBOLE COMPLEXE

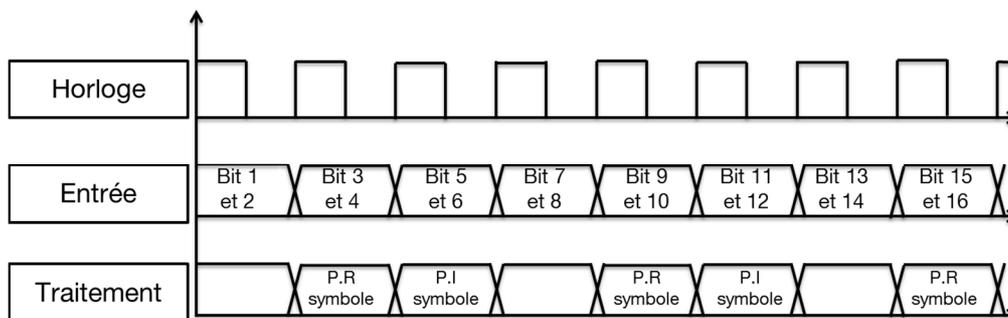


FIG. 3.5 : FONCTIONNEMENT DU BLOC MODULATION POUR  $K=6$

### 3.2.2.5 Mise en trame

Le bloc de mise en trame a pour but de construire le symbole OFDM incluant les données, les symboles pilotes et les symboles nuls. Les pilotes permettent d'estimer le canal afin de corriger les effets de celui-ci sur les symboles de données. Les sous-porteuses nulles sont présentes afin d'éviter les interférences entre bandes et permettent d'avoir un intervalle de garde. Le bloc de mise en trame récupère les symboles de données venant du bloc modulation et les place aux sous-porteuses allouées aux données. De la même manière les pilotes ainsi que les sous-porteuses nulles seront placés aux endroits réservés à cet effet avant envoi au bloc IFFT. La norme 802.11ac spécifie les emplacements réservés comme le montre le Tab. 3.5. On notera que 52 sous-porteuses de données sont disponibles pour la transmission à 20 MHz ; 108 dans le cas 40 MHz et 234 dans le cas 80MHz.

	Indices
Pilotes	-21, -7, 7, 21
Sous-porteuses nulles	-32 à -29, 0, 29 à 31

(a) Trame OFDM pour une bande passante de 20 MHz

	Indices
Pilotes	-53, -25, -11, 11, 25, 53
Sous-porteuses nulles	-64 à -59, -1 à 1, 59 à 63

(b) Trame OFDM pour une bande passante de 40 MHz

	Indices
Pilotes	-103, -75, -39, -11, 11, 39, 75, 103
Sous-porteuses nulles	-64 à -59, -1 à 1, 59 à 63

(c) Trame OFDM pour une bande passante de 80 MHz

TAB. 3.5 : TRAME OFDM

### 3.2.2.6 Modulation OFDM

La modulation OFDM est réalisée à l'aide d'une IFFT. Ce bloc permet de lutter contre les canaux sélectifs en fréquence en divisant la bande de transmission en plusieurs sous-bandes (sous porteuses). L'orthogonalité permet le chevauchement des sous porteuses sans que celles-ci n'interfèrent entre elles comme le montre la Fig. 3.6. L'espacement inter-porteuses  $\Delta f$  est choisi de sorte qu'il n'y ait pas d'interférences entre les sous porteuses.

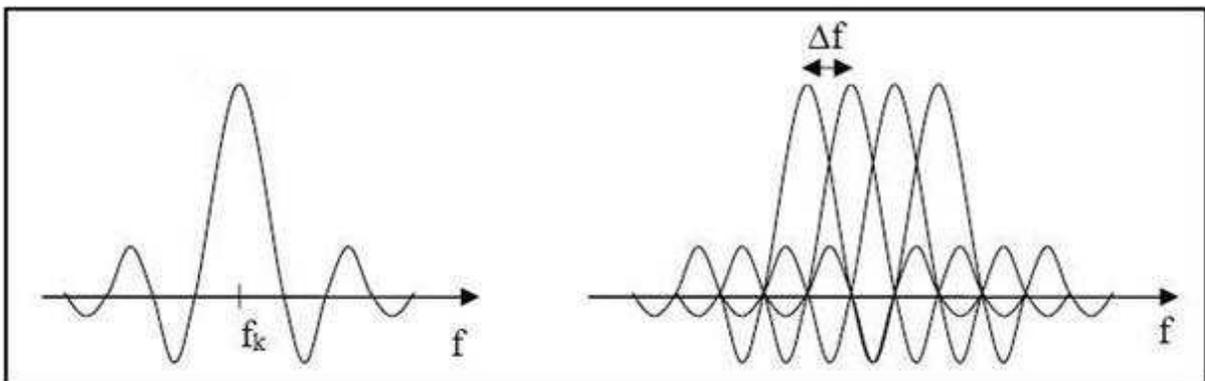


FIG. 3.6 : ORTHOGONALITE DES SOUS-PORTEUSES [77]

Un préfixe cyclique est également ajouté dans le domaine temporel au début de chaque symbole OFDM avant la transmission afin d'éviter l'interférence entre symboles due à l'étalement temporel du canal.

Les principaux paramètres sont présentés Tab. 3.6.

Largeur de bande	20 MHz	40 MHz	80 MHz
Taille de FFT	64	128	256
Nombre total de sous porteuse de données	52	108	234
Nombre de pilotes	4	6	8
Nombre de sous porteuses nulles	8	14	14
Durée du préfixe cyclique	16 (0.8 $\mu$ s)	32 (0.8 $\mu$ s)	64 (0.8 $\mu$ s)
Durée du symbole OFDM avec préfixe cyclique	80 (4 $\mu$ s)	160 (4 $\mu$ s)	320 (4 $\mu$ s)

TAB. 3.6 : *PARAMETRES OFDM*

L'architecture de la IFFT peut être soit pipelinée ou non pipelinée. Un bloc pipeliné est capable de traiter plusieurs trames de données en parallèle comme schématisé Fig. 3.7; alors qu'un bloc sans pipeline traite une trame de données entièrement avant de commencer à en traiter une autre. Il convient de souligner que l'architecture pipelinée est plus rapide, mais nécessite beaucoup plus de ressources matérielles. La comparaison de ces deux architectures est traitée au chapitre 5. Nous avons implémenté dans un premier temps une IFFT avec une architecture pipelinée. Le bloc IFFT utilisé est un IPCore développé par Xilinx. L'architecture utilisée pour pipeliner les données est connue sous le nom de « papillon radix-2 »[78]. Chaque radix possède sa propre mémoire afin de stocker les données d'entrée et les calculs intermédiaires. L'IPCore est capable de traiter en parallèle le calcul de la trame de données, de charger la trame de données suivante et de décharger la trame de données précédente ce qui permet d'avoir des données continues en sortie après l'introduction d'une certaine latence en fonction de la taille de la IFFT. La FFT en réception est également un IPCore ayant les mêmes caractéristiques.

L'objectif de cette étude étant d'évaluer la consommation d'énergie de l'émetteur et du récepteur, le bloc pouvant émuler le canal de transmission n'a pas été développé. Néanmoins, la dynamique d'entrée de la chaîne de réception est égale à la dynamique de sortie de la chaîne de transmission. Par exemple, nous avons choisi de quantifier chaque symbole en entrée de la IFFT sur 10 bits (c'est à dire, 10 bits pour les parties réelles, 10 bits pour les parties imaginaires). Dans le cas d'une IFFT de taille 64, le nombre d'étages est égal à 7 ce qui implique l'ajout d'un bit à chaque étage pour éviter le dépassement de données. En sortie de cette IFFT chaque symbole est donc quantifié sur 17 bits. En entrée de la chaîne de réception, nous utiliserons donc une dynamique de 17 bits pour quantifier les données.

Le Tab. 3.7 montre les ressources utilisées par le bloc IFFT. On remarque que le nombre de ressources augmente non linéairement avec la taille de la IFFT.

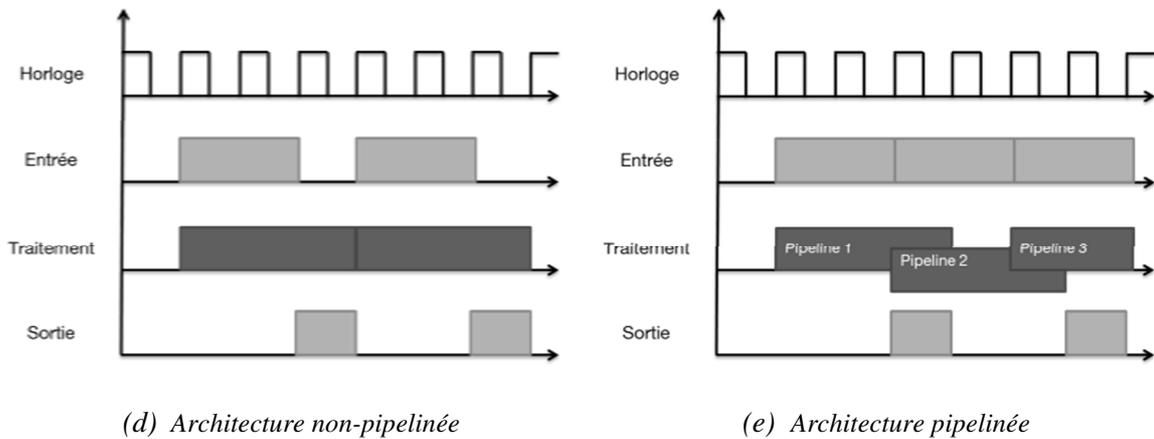


FIG. 3.7 : ARCHITECTURE FFT

	FFT64	FFT128	FFT256	FFT512
Nombre de Slice Registers	2143	2208	2607	3081
Nombre de Slice LUTs	2006	2024	2386	2815
Nombre de Flip Flops (FF)	1395	1401	1621	1863

TAB. 3.7 : RESSOURCES UTILISEES PAR LE BLOC IFFT

### 3.2.2.7 Décodeur de Viterbi

Le décodeur en réception peut être réalisé à l'aide d'un algorithme de Viterbi [79]. Ce bloc a un rôle de correcteur puisqu'il doit retrouver les bits de données originales avant le codage à l'émission. Cela signifie qu'il doit non seulement effectuer les opérations inverses du bloc codeur et poinçonnage, mais également retrouver les bits perdus dans le canal de transmission. La capacité du décodeur à corriger les erreurs aura un effet sur le taux d'erreurs binaires de la chaîne. Les performances d'un décodeur de Viterbi dépendent de la taille de profondeur utilisée pour décoder les bits, ce paramètre est plus connu sous le nom de « *trace-back* ». Plus la taille du *trace-back* est grande, meilleures seront les performances mais plus grande sera la latence. Dans cette étude nous utilisons une longueur de *trace-back* fixe qui est de 35. Il est à noter que le décodeur utilisé a été produit par les membres du consortium GREENCoMM dans le cadre du projet. Le décodeur utilisé est un bloc IP.

### 3.2.2.8 Ressources utilisées par les blocs

Le Tab. 3.8 fournit les résultats de synthèse des différents blocs composant la chaîne d'émission et de réception pour un débit cible de 117 Mbps (80 MHz, MCS 3). On observe que la complexité du décodeur est 3 à 4 fois plus élevée que le bloc (I)FFT. De la même manière, le bloc (I)FFT possède deux fois plus de LUTs et FFs que le bloc entrelaceur. Ces résultats permettent d'avoir un premier aperçu de la consommation de puissance des blocs. Intuitivement, on s'attend à ce que le bloc décodeur consomme plus d'énergie que le bloc (I)FFT par exemple.

	Nombre de Slice LUTs	Nombre de Flip Flops (FFs)
Embrouilleur/Désembrouiller	45	49
codeur	9	9
Entrelaceur	731	739
Désentrelaceur	753	760
Modulation	60	68
Démodulation	82	85
IFFT/FFT	2386	1621
Décodeur	6231	6445

TAB. 3.8 : RESULTATS DE SYNTHESE DES BLOCS ISOLES

### 3.2.3 Validation

La chaîne de communications numériques développée au niveau matériel a été validée par une chaîne « haut niveau » préalablement développée en C++. La validation est, dans un premier temps, réalisée bloc à bloc en générant des entrées identiques pour les deux types de chaînes, haut niveau et bas niveau, et en observant les sorties. Si les sorties sont identiques alors le bloc est validé comme schématisé Fig. 3.8. La validation des chaînes Tx/Rx a été réalisée de la même manière.

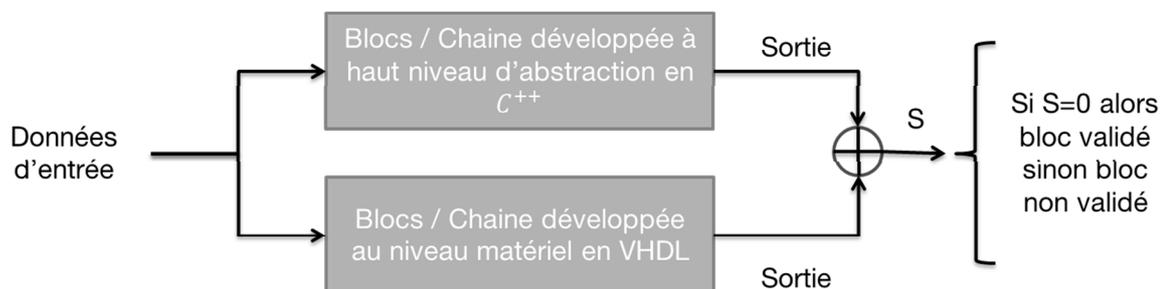


FIG. 3.8 : METHODE DE VALIDATION DES BLOCS/CHAINES

Il est à noter que la chaîne de communication haut niveau permet d'évaluer les performances en termes de taux d'erreurs binaires alors que la chaîne développée au niveau matériel permet d'évaluer la consommation de puissance.

Pour évaluer la consommation d'un bloc, il est nécessaire de fournir des données d'entrée pertinentes afin de tester toutes les combinaisons de trames possibles (méthode statistique). Il existe deux façons pour transmettre les données à l'entrée d'un bloc :

- La première solution consiste à lire les données d'entrée dans un fichier, ce qui permet d'évaluer le bloc seul. Les fichiers utilisés sont issus de la chaîne haut niveau qui permet d'avoir des données pseudo-aléatoires de période longue. Il est à noter que les temps de simulation sont assez longs pour que toutes les combinaisons d'entrées du système soient testées. Cette solution nécessite de sauvegarder les données d'entrée dans une mémoire externe à la carte.
- La deuxième solution consiste à ajouter un bloc générateur de séquences aléatoires PRBS pour générer des données d'entrée pseudo-aléatoires. Cette solution est la plus simple à mettre en œuvre sur du matériel pour mesurer la consommation. Cette solution sera donc retenue dans la suite du chapitre. Le polynôme  $G(x) = x^{15} + x^6$  est utilisé, ce qui permet de générer une séquence de 65535 bits.

Afin de vérifier le fonctionnement des chaînes Tx/Rx, nous avons dupliqué en sortie le bloc PRBS présent en entrée de la chaîne. Cela permet de vérifier que les bits présents en entrée de la chaîne de transmission sont bien retrouvés en sortie de la chaîne de réception comme illustré Fig. 3.9.

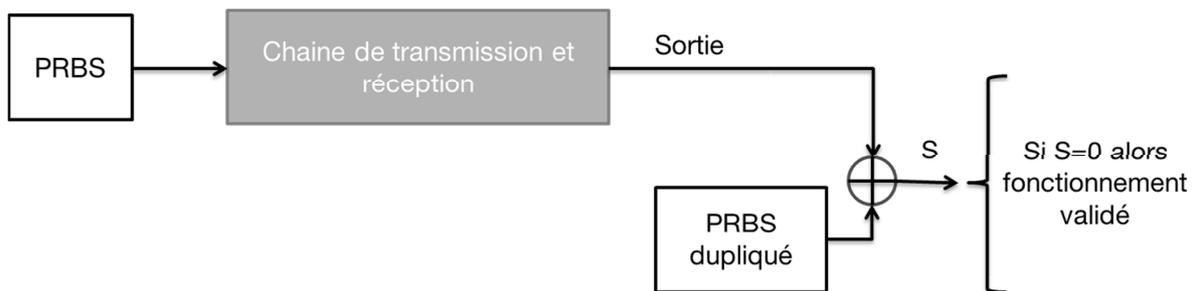


FIG. 3.9 : METHODE DE VALIDATION DU FONCTIONNEMENT DE LA CHAÎNE COMPLETE

### 3.2.4 Architecture globale de la chaîne

L'architecture globale de la chaîne choisie est telle que 2 bits sont transmis par cycle d'horloge par le codeur. Cela signifie qu'à chaque cycle d'horloge, les blocs situés après le codeur traitent deux

bits en entrée et en sortie comme le montre la Fig. 3.10. Par conséquent, chaque symbole complexe nécessite  $K/2$  cycles d'horloge pour être formé,  $K$  étant le nombre de bits codés par symbole complexe. Seul le bloc IFFT impose de recevoir en série et en continu les symboles. Cela conduit à 2 architectures des chaînes, selon le MCS. Une architecture permet d'avoir un flux continu de données binaires en entrée ; et l'autre nécessite d'avoir des arrêts sur le flux d'entrée afin de ne pas perdre des données. Dans le cas du flux continu, la fréquence d'horloge cible est égale au débit ; dans le cas contraire, elle vaut :

$$f = \frac{T_{chaîne}}{N_{CBPS}} \cdot K \cdot R$$

où  $T_{chaîne}$  représente le nombre de cycles d'horloge nécessaires pour transmettre un symbole OFDM,  $R$  représente le débit. Ces deux architectures seront détaillées au chapitre 4, mais nous pouvons d'ores et déjà signaler que le flux de données est continu lorsque  $MCS \geq 3$ , et qu'il est discontinu lorsque  $MCS \leq 1$ .

Le choix de transmettre 2 bits en parallèle après le codeur est dû à l'architecture du codeur convolutif mère dont le rendement est  $1/2$ . Ainsi pour chaque bit en entrée, 2 bits sont placés en sortie. On peut alors envisager de traiter ces bits en série ou en parallèle. La prise en série des bits aurait imposé de doubler la fréquence de fonctionnement ou alors de dupliquer les blocs situés après le codeur convolutif. Cette manière de faire aurait généré un coût significatif sur la puissance énergétique, or le but est d'optimiser la consommation de puissance. Pour cette raison, nous avons opté pour la solution consistant à récupérer les bits en parallèle. Ainsi, la fréquence de fonctionnement de la chaîne est la même pour tous les blocs la composant.

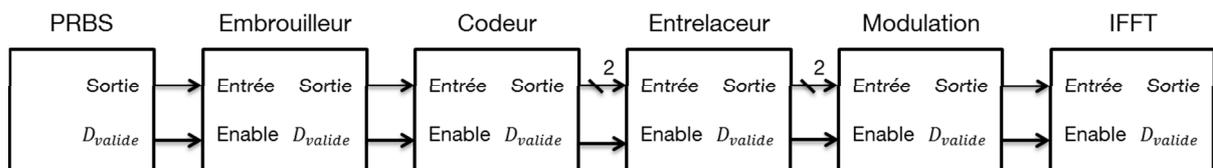


FIG. 3.10 : TRAITEMENT DE DEUX BITS EN PARALLELE

Afin de faciliter la communication entre les blocs, les entrées/sorties de chaque bloc sont identiques. Cela permet de faciliter l'échange, l'ajout ou la suppression d'un bloc du système. La Fig. 3.11 présente le modèle d'interface sélectionné.

L'interface contient une entrée d'horloge (CLOCK), de remise à zéro du système (RESET) et le flux de données en entrée et en sortie (D\_TDATA et Q\_TDATA respectivement). Des signaux de

contrôle sont également présents afin de contrôler les communications. On retrouve en entrée venant du bloc précédent, un signal indiquant la donnée valide un cycle d'horloge avant (D\_TPEVALID), un signal indiquant la donnée valide en entrée (D\_TVALID), un signal indiquant la dernière donnée valide de la trame (D\_TLAST) et un signal venant du bloc suivant indiquant si le bloc est prêt à recevoir des données (Q\_TREADY).

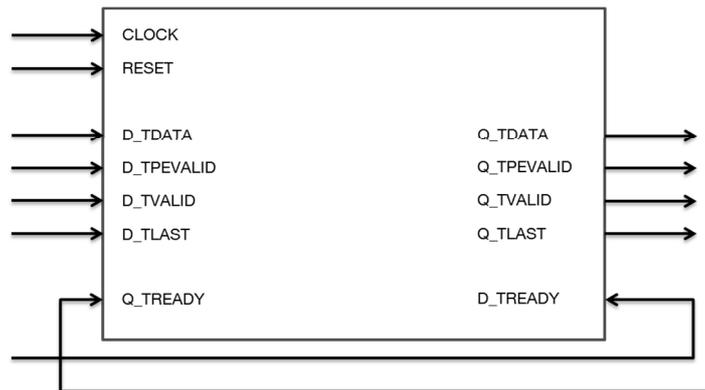


FIG. 3.11 : *INTERFACE DES BLOCS*

### 3.2.5 Scénarios testés

Les chaînes Tx/Rx ont été simulées pour les paramètres suivants : une fréquence porteuse de 5 GHz; une antenne est utilisée du côté de l'émission et de la réception ; la durée du symbole OFDM est de 4  $\mu$ s. Les caractéristiques des scénarios simulés sont résumées dans le Tab. 3.9, ainsi que les débits cibles de la couche physique. Il faut noter qu'avec l'architecture développée, le décodeur ne peut pas supporter des fréquences d'horloge supérieures à 200 MHz, car les chemins critiques sont atteints. Le décodeur étant un IP nous ne pouvons pas agir sur l'architecture afin de casser les chemins critiques. Par conséquent, les scénarios MCS 5 et MCS 8 à 80 MHz (en fond gris dans le tableau) n'ont pas été simulés. Une solution pour remédier à ce problème est de dupliquer le bloc décodeur qui a un chemin critique trop long afin de diviser la fréquence d'horloge par deux.

Les temps de simulations est fixé à 8 ms ce qui permet de simuler un certain nombre de symbole OFDM en fonction de la configuration choisie. Il est à noter que même si les temps de simulation sont fixes, le temps réel nécessaire pour simuler un bloc varie. En effet, en fonction de la complexité du bloc celui-ci prend plus de temps pour terminer la simulation des 8 ms.

Largeur de bande		20 MHz	40 MHz	80 MHz
Taille de FFT		64	128	256
Nombre de données par symbole OFDM		52	108	234
Taille du préfixe cyclique par symbole OFDM		16	32	64
Débits	MCS 1 (QPSK 1/2)	13 Mbps	27 Mbps	58.5 Mbps
	MCS 3 (16-QAM 1/2)	26 Mbps	54 Mbps	117 Mbps
	MCS 5 (64-QAM 2/3)	52 Mbps	108 Mbps	234 Mbps
	MCS 8 (256-QAM 3/4)	78 Mbps	162 Mbps	351 Mbps

TAB. 3.9 : SCENARIOS WI-FI

### 3.3 Puissance statique

Nous avons vu au chapitre 1 que la puissance statique ne peut être négligée devant la puissance dynamique notamment due à la complexité des systèmes. Les principaux paramètres impactant la puissance statique sont la tension d'alimentation, les courants de fuites et la température, comme le montre l'équation (1.9). Dans cette section, nous allons quantifier l'impact de la température sur une chaîne de réception et le poids que cette puissance a sur la consommation totale du circuit.

La Fig. 3.12 montre les puissances statiques et dynamiques de la chaîne de réception Wi-Fi pour deux débits, 52 Mbps et 162 Mbps en simulation. On constate que la puissance statique est affectée par la température, tandis que la puissance dynamique reste constante. A l'inverse, la puissance dynamique est impactée par les paramètres de transmission, alors que la puissance statique reste constante pour un scénario Wi-Fi donné et une température donnée. La puissance statique est dépendante uniquement de la tension d'alimentation et des courants de fuites (températures, tailles des transistors, ...).

Cette analyse nous donne également un ordre de grandeur du poids des parties statiques et dynamiques de la puissance du circuit. Lors de la phase de test, la température mesurée sur la carte était de 46°. La puissance statique à cette température est d'environ 450 mW tandis que la puissance dynamique est de 622 mW pour le plus haut débit simulé (162 Mbps) et de 200 mW pour un débit de 52 Mbps. Ainsi, nous remarquons que la puissance statique représente approximativement 42% de la puissance du circuit dans le cas d'un débit de 162 Mbps et 69% pour un débit de 52 Mbps.

Le Tab. 3.10 montre la puissance statique de la chaîne de réception pour différentes configurations. On observe que la puissance statique est fixe quelle que soit l'architecture utilisée, puisque les variations de tensions et de courants de fuites ne sont pas significatives. Dans la suite

du manuscrit, notre objectif est d'observer l'influence des paramètres de transmission et des architectures choisies sur la consommation de la chaîne. Pour cela nous nous intéressons uniquement à l'étude de la puissance dynamique, la puissance statique étant constante vis à vis de ces paramètres.

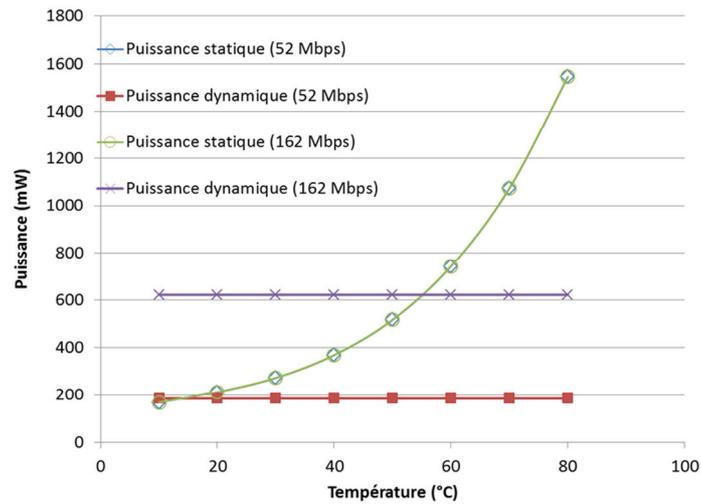


FIG. 3.12 : PUISSANCES STATIQUE ET DYNAMIQUE VS TEMPERATURE. 52 MBPS CORRESPOND AU MCS5 POUR UNE BANDE DE 20 MHZ ET 162 MBPS CORRESPOND AU MCS8 POUR UNE BANDE DE 40 MHZ.

Largeur de bande	MCS	Puissance statique
20 MHz	1	459 (46.6°)
	3	460 (46.6°)
	5	462 (46.7°)
	8	464 (46.9°)
40 MHz	1	461 (46.7°)
	3	462 (46.8°)
	5	467 (47.1°)
	8	470 (47.3°)
80 MHz	1	463 (46.8°)
	3	468 (47.1°)

TAB. 3.10 : VARIATION DE LA PUISSANCE STATIQUE

### 3.4 Puissance dynamique des chaînes Tx/Rx

La consommation de puissance dynamique de la chaîne Tx/Rx incluant tous les blocs décrits à la section 3.2.1 est présentée Tab. 3.11. Le tableau indique à la fois les résultats obtenus en

simulation via l’outil XPower Analyser et les mesures réalisées sur carte FPGA. On observe une faible différence entre les simulations et les mesures, l’erreur moyenne étant de 6.5% normalisée par rapport à la mesure (3.3). Cette erreur est tout à fait acceptable sachant que le pas de mesure de puissance sur FPGA est de 10 mW alors qu’il est de 1 mW en simulation.

$$Erreur\ relative = \frac{Simulation - mesure}{mesure} \cdot 100 \quad (3.3)$$

Bande	MCS	Simulations		Mesures		Erreur relative	
		Puissance Tx [mW]	Puissance Rx [mW]	Puissance Tx [mW]	Puissance Rx [mW]	Puissance Tx [mW]	Puissance Rx [mW]
20 MHz	MCS1	12	48	10	50	20%	-4%
	MCS3	17	87	20	80	-15%	8.75%
	MCS5	39	185	40	190	-2.5%	-2.6%
	MCS8	59	258	60	250	-1.7%	3.2%
40 MHz	MCS1	29	113	30	110	-3.3%	2.7%
	MCS3	50	195	50	170	0%	14.7%
	MCS5	118	453	120	450	-1.6%	0.6%
	MCS8	207	622	200	600	3.5%	3.6%
80 MHz	MCS1	73	257	90	230	-18.9%	11.7%
	MCS3	177	491	160	480	10.6%	2.3%

TAB. 3.11 : PUISSANCE DYNAMIQUE DES CHAINES TX/RX

On remarque que plus le MCS est important plus la consommation est importante, cela est dû au fait que le débit cible augmente avec le MCS ce qui impacte directement la fréquence d'horloge, et donc la consommation.

On remarque qu’il est possible d’avoir un gain important sur la consommation en utilisant les bons jeux de paramètres. En effet, plusieurs scénarios permettent d’avoir des débits très similaires comme le montre le Tab. 3.12.

Largeur de bande	MCS	Débits
20 MHz	3	26
40 MHz	1	27
40 MHz	3	54
80 MHz	1	58,5

TAB. 3.12 : DEBITS BINAIRES

Ainsi si le débit cible est 26 Mbps, on remarque qu’il vaut mieux choisir une bande passante égale à 20 MHz et un MCS 3 plutôt qu’une bande passante de 40 MHz et un MCS 1. Le gain obtenu en termes de puissance dynamique simulé est de l’ordre de 41% pour la chaîne d’émission et de 23% pour la chaîne de réception. Un raisonnement similaire peut être fait pour un débit de 54 Mbps.

Néanmoins, cette optimisation peut être faite seulement si les exigences liées à la qualité de service sont respectées et donc que le taux d'erreurs binaires (Bit Error Rate, BER) cible maximum n'est pas atteint. A noter que dans le contexte du Wi-Fi, la qualité de service est liée au MCS choisi. Afin de vérifier si ces solutions sont envisageables, des simulations haut niveau sur la chaîne de référence (C/C++) ont été menées. Ces études ont permis d'obtenir des courbes de rapport signal à bruit (Signal to Noise Ratio, SNR) en fonction du BER. Le rapport signal à bruit est un indicateur de la qualité de l'information transmise. Ces courbes sont représentées Fig. 3.13. Les paramètres de simulation sont les suivants : la fréquence porteuse est de 5 GHz, le canal simulé est du type B spécifié dans la norme et l'égaliseur de canal est un Zéro Forcing. Ce type de canal permet de simuler un environnement réaliste de type résidentiel.

On remarque que pour un BER cible de  $10^{-3}$ , si le SNR cible est supérieur à 30 dB, il est possible d'utiliser n'importe quelle bande passante pour les MCS 1 et 3, car le BER ciblé sera toujours respecté. Dans l'exemple présenté précédemment, si l'on souhaite utiliser un MCS3+20 MHz au lieu d'un MCS1+40 MHz il faut que le SNR soit supérieur ou égal à 26 dB toujours pour le même BER cible de  $10^{-3}$ .

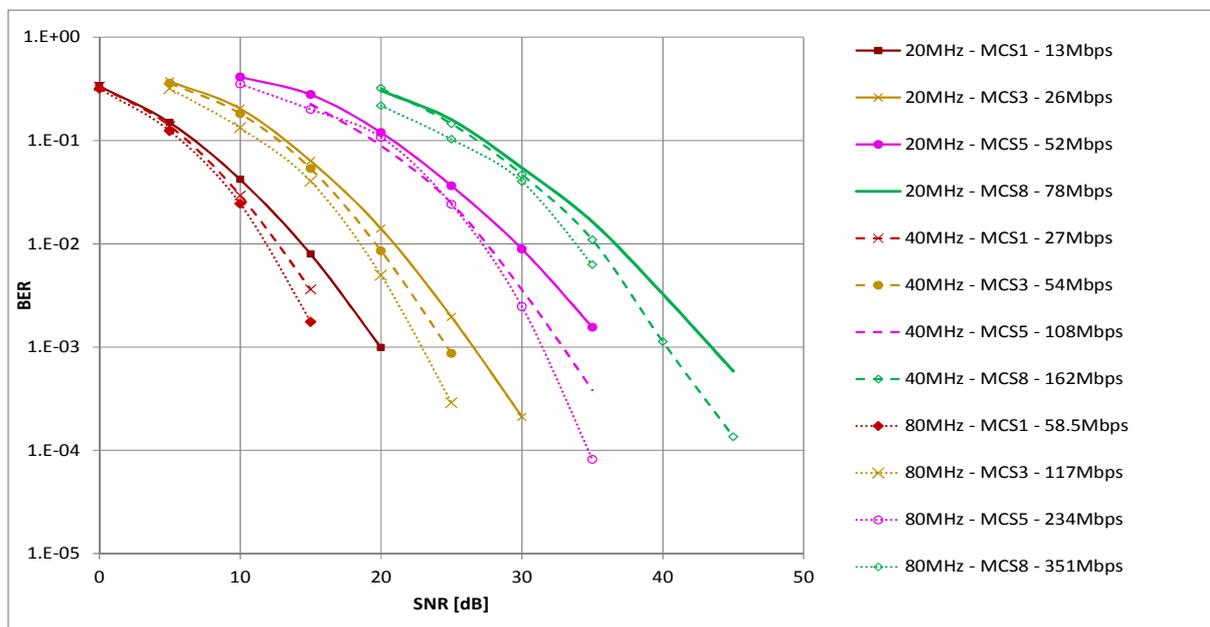


FIG. 3.13 : COURBES BER EN FONCTION DU SNR

On note également que la puissance consommée des chaînes Tx/Rx augmente avec le débit comme le montre la Fig. 3.14 mais de façon non linéaire.

Chaque débit correspond à un MCS et une largeur de bande, qui impacte sur l'architecture de la chaîne et sa fréquence d'horloge cible. Par exemple, le bloc modulation dépend du MCS et de la largeur de bande; la FFT dépend uniquement de la largeur de bande. Donc, pour une largeur de bande donnée (donc une taille de FFT donnée), la consommation de puissance de la FFT augmente linéairement avec la fréquence d'horloge cible, ce qui n'est pas le cas de la modulation comme le montre la Fig. 3.15. En outre, le débit n'est pas proportionnel à la fréquence d'horloge cible, ce qui explique la non linéarité de la puissance des chaînes en fonction du débit.

Dans la section suivante, nous détaillons les consommations simulées pour l'ensemble des blocs de la chaîne.

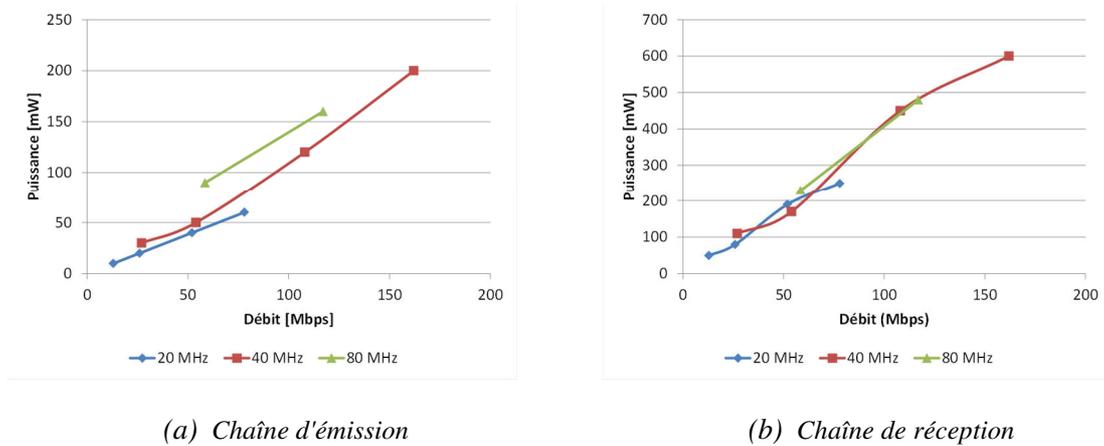


FIG. 3.14 : PUISSANCE DYNAMIQUE DE LA CHAÎNE TX/RX POUR LES DIFFÉRENTS SCÉNARIOS. CHAQUE POINT REPRÉSENTE UN MCS DONNÉ.

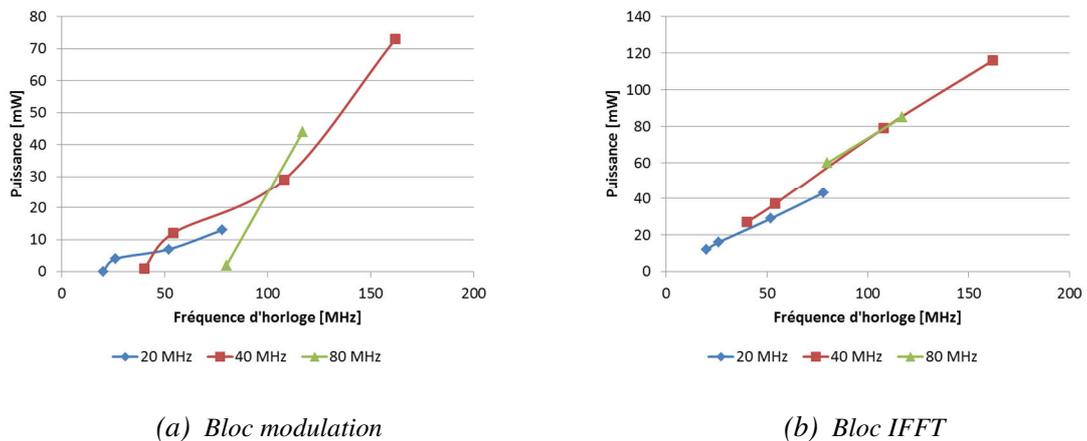


FIG. 3.15 : PUISSANCE DYNAMIQUE DES BLOCS MODULATION ET IFFT

### 3.5 Puissance dynamique des blocs

Un des objectifs de la thèse est d'obtenir la consommation de puissance des blocs composant les chaînes d'émission et de réception du Wi-Fi afin d'identifier les blocs les plus énergivores.

Le Tab. 3.13 présente la consommation en puissance des blocs lors d'une activité continue en simulation, c'est-à-dire lorsque les blocs sont constamment actifs. Il est à noter que l'énergie nécessaire pour exécuter le bloc embrouilleur et IFFT du côté de l'émission est identique à celle des blocs désembrouilleur et FFT respectivement, car ce sont exactement les mêmes architectures utilisées. Pour les autres blocs, des différences entre l'émission et la réception doivent être prises en compte :

- A l'émission le bloc poinçonnage a été fusionné avec le bloc entrelaceur. Sa puissance est donc différente du bloc désentrelaceur en réception.
- A l'émission le bloc de mise en trames a été fusionné avec le bloc modulation.
- En réception, l'étape de dépoinçonnage est incluse dans le décodeur de Viterbi.

Largeur de bande	MCS	embrouilleur	codeur	entrelaceur	modulation	IFFT	démodulation	désentrelaceur	décodeur
20 MHz	MCS1	1	0	2	0	12	1	4	57
	MCS3	1	0	4	4	16	4	4	74
	MCS5	1	1	16	7	29	17	17	147
	MCS8	1	2	31	13	43	22	20	221
40 MHz	MCS1	1	1	6	1	27	3	11	112
	MCS3	1	1	16	12	37	9	28	152
	MCS5	1	2	62	29	79	55	80	302
	MCS8	2	3	91	73	116	62	98	454
80 MHz	MCS1	2	2	27	2	60	4	52	225
	MCS3	2	3	71	44	85	36	68	331

TAB. 3.13 : CONSOMMATION EN PUISSANCE DES BLOCS [mW]

On observe qu'à l'émission le bloc IFFT est le plus gourmand en énergie, cependant à haut débit les blocs entrelaceur et modulation se rapprochent sensiblement du bloc IFFT. Du côté de la réception, le décodeur de Viterbi est de loin le plus consommateur. Cela montre clairement le gain en énergie que l'on peut obtenir en optimisant l'architecture de ces blocs.

A titre d'information, le Tab. 3.14 montre la somme des puissances dynamiques de tous les blocs des chaînes Tx et Rx, ainsi que l'écart observé par rapport à la chaîne totale simulée fournie Tab. 3.11. On observe que la somme des blocs est supérieure à la puissance totale des chaînes. Ceci est

dû en partie à l'optimisation et au partage de mémoires et de registres lorsque les chaînes complètes sont synthétisées, mais aussi et principalement en raison des taux d'activité temporels des blocs. En effet, une fois intégrés dans la chaîne, ces blocs ne sont pas constamment actifs ou n'utilisent pas 100% de leur capacité en ressources. Cet aspect sera traité au chapitre suivant.

Bande	MCS	Puissance Tx [mW]	Puissance Rx [mW]	Ecart Tx	Ecart Rx
20 MHz	MCS1	15	75	25%	56.2%
	MCS3	25	99	47%	13.8%
	MCS5	54	211	38.5%	14.1%
	MCS8	90	307	52.5%	20%
40 MHz	MCS1	36	154	24.1%	36.3%
	MCS3	67	227	34%	16.4%
	MCS5	173	517	46.6%	14.1%
	MCS8	285	732	37.7%	17.7%
80 MHz	MCS1	93	343	27.4%	33.5%
	MCS3	285	522	61%	6.3%

TAB. 3.14 : SOMME DES BLOCS ISOLEES ET ECART OBSERVE PAR RAPPORT A LA CHAINE TOTALE

### 3.6 Conclusion

Nous avons vu au chapitre 1 que la puissance totale sur carte est composée d'une partie statique et d'une partie dynamique. Dans ce chapitre, nous avons observé que la puissance statique augmente avec la température alors que la puissance dynamique reste constante. Inversement, la puissance dynamique augmente avec la fréquence d'horloge tandis que la puissance statique reste constante. La puissance statique dépendant en grande partie de la technologie utilisée, nos moyens d'action pour réduire cette puissance sont très limités. Nous n'avons donc traité que la puissance dynamique lors de l'évaluation de la consommation des blocs et des chaînes Tx/Rx.

L'évaluation de la consommation des chaînes Wi-Fi a été traitée dans ce chapitre. Les développements au niveau matériel des chaînes ont permis d'implémenter puis d'évaluer la consommation de puissance aussi bien en simulation qu'en mesure sur carte FPGA. Nous avons comparé les résultats de consommation de puissance obtenus en simulation à ceux obtenus en mesure. Il s'avère que les résultats sont proches avec une erreur moyenne de 6.5%.

De plus, nous avons identifié les blocs les plus gourmands en énergie et quantifié le poids de chaque bloc dans la chaîne lorsqu'ils sont évalués seuls. Il en résulte que le décodeur est le bloc le plus consommateur du côté de la réception. La IFFT/FFT est le bloc le plus gourmand en énergie du côté de l'émission.

Enfin, nous avons vu que la somme de la consommation des blocs composant la chaîne n'est pas égale à la consommation de cette dernière. Cela est dû en grande partie au fait que l'activité temporelle des blocs n'a pas été prise en compte à ce stade de notre étude. Ils sont supposés être constamment actifs, mais ce n'est pas le cas lors du fonctionnement de la chaîne. Cette problématique sera traitée dans le prochain chapitre où l'étude d'une méthodologie permettant de calculer les taux d'activité des blocs sera présentée. Ces taux d'activités par bloc seront ensuite pris en compte dans le calcul de la consommation totale de la chaîne.

## Chapitre 4

# Méthode semi-analytique d'estimation de la puissance dynamique consommée

### 4.1 Introduction

A l'aide des outils et de la plateforme décrits au chapitre 2, nous avons au chapitre précédent évalué la consommation de puissance d'une chaîne de communication Wi-Fi pour différents paramètres système. Nous avons remarqué que les temps de simulation de l'ordre de la semaine sont trop longs et inappropriés lorsque l'on souhaite optimiser l'architecture ou tester des techniques de réduction de la consommation. Nous proposons dans cette section une méthode à haut niveau d'abstraction rapide et fiable dans le but d'estimer la consommation en puissance, permettant de tester diverses architectures, et applicable à tout système de communications numériques.

Afin d'estimer la consommation de puissance de manière précise, il est nécessaire de connaître l'activité temporelle de chaque bloc. Dans les systèmes de communications, l'activité varie et est influencée par exemple par les entrées/sorties des blocs, l'architecture globale du système ou encore les multiples configurations disponibles dans les normes (largeur de bande, MCS). Nous proposons dans ce chapitre une analyse temporelle d'une chaîne de communication, déterminée suivant 3 paramètres qui sont les entrées, les sorties et les temps de traitement des blocs composant le système.

La méthode proposée utilise une base de données contenant les consommations dynamiques des blocs pris individuellement lors d'une activité continue. L'analyse temporelle de la chaîne permet d'obtenir les taux d'activité temporelle des blocs lorsqu'ils sont intégrés dans la chaîne, puis de

pondérer les puissances des blocs évaluées individuellement par ces taux. La méthode se décompose en trois étapes:

- Création de la base de données : évaluation de la consommation des blocs pris individuellement par simulation,
- Calcul analytique de l'activité des blocs composant la chaîne,
- Calcul de la consommation énergétique de la chaîne.

## 4.2 Analyse temporelle

Dans cette section, nous détaillerons l'architecture de la chaîne en spécifiant les transferts de bits ou de symboles entre blocs. Nous proposons dans un premier temps une analyse temporelle des chaînes de communications numériques, puis nous l'appliquons au cas du Wi-Fi.

### 4.2.1 Cas général

L'analyse temporelle d'un bloc peut être représentée selon trois paramètres qui sont les temps d'entrées, de sorties et de traitement d'une trame de données. Il est à noter que lorsque le système est sous tension, les entrées-sorties d'un bloc sont tout le temps actives, cela signifie qu'une donnée est toujours présente. Ainsi il est nécessaire de différencier les données valides utiles au traitement de la trame, des données invalides.

L'architecture développée prend en compte ces données en affectant un signal à l'entrée de chaque bloc permettant de vérifier si la donnée en entrée est valide; ainsi qu'un signal en sortie permettant d'indiquer au bloc suivant si la donnée transmise est valide ou non.

Comme nous l'avons vu au chapitre précédent, on peut définir deux grandes catégories d'architectures d'un bloc, l'architecture pipelinée et l'architecture non pipelinée représentées Fig. 3.7. Le choix de l'architecture influence le nombre de cycles d'horloge nécessaires au bloc pour traiter une trame de données lorsque celui-ci est isolé, nous les traiterons donc séparément. Dans la suite de ce chapitre nous définissons  $T_m$  comme étant le nombre de cycles nécessaire au  $m$ -ième bloc pour traiter une trame de données, lorsque ce bloc est isolé. Pour faciliter la lecture, nous définissons dans la suite du chapitre le temps comme étant le nombre de cycles d'horloge.

Dans le cas d'une architecture non pipelinée, le temps nécessaire au bloc pour traiter une trame de données lorsque le bloc est isolé est le nombre de cycles le plus grand entre  $T_{in,m}$ ,  $T_{out,m}$  et  $T_{proc,m}$ ; où

- $T_{in,m}$  représente le temps durant lequel des données valides sont présentes en entrée du bloc afin de traiter une trame de données complète
- $T_{out,m}$  représente le temps durant lequel des données valides sont présentes en sortie du bloc afin de traiter une trame de données complète
- $T_{proc,m}$  représente le temps nécessaire afin de traiter une trame de données complète.

Cela signifie que  $T_m$  est déterminé par le temps le plus contraignant et est défini par l'équation (4.1).

$$T_m = \max\{T_{in,m}; T_{out,m}; T_{proc,m}\} \quad (4.1)$$

Dans le cas d'une architecture pipelinée, le temps de traitement  $T_{proc,m}$  étant parallélisé, il n'intervient plus dans le calcul de  $T_m$ . En effet dans une architecture pipelinée, le but recherché est que le traitement des données du bloc n'influence pas l'entrée ou la sortie, c'est-à-dire que les données en entrée ou en sortie soient continues si l'activité du bloc est continue. L'équation devient alors la suivante :

$$T_m = \max\{T_{in,m}; T_{out,m}\} \quad (4.2)$$

Afin de déterminer si le flux de données en entrée d'une chaîne constituée de  $M$  blocs est continu, il convient de définir le bloc le plus contraignant de la chaîne, c'est-à-dire le bloc nécessitant le temps de traitement d'une trame de données le plus long. Dans la suite de ce chapitre, nous définissons  $T_{chaîne}$  comme étant la période de traitement d'une trame par la chaîne complète.

Ainsi  $T_{chaîne}$  est défini par l'équation (4.3) où  $T_m$  est défini par l'équation (4.1) dans le cas d'une architecture non pipelinée et par (4.2) dans le cas d'une architecture pipelinée.

$$T_{chaîne} = \max_{1 \leq m \leq M} \{T_m\} \quad (4.3)$$

Pour savoir si le flux de données en entrée d'une chaîne est continu il faut comparer  $T_{chaîne}$  au temps de traitement du 1<sup>er</sup> bloc,  $T_{in,1}$ . Si  $T_{chaîne} > T_{in,1}$  alors le flux de données d'entrée n'est pas continu, à l'inverse si  $T_{chaîne} \leq T_{in,1}$  alors le flux de données est continu.

Dans de nombreux systèmes, notamment dans les systèmes de communication, un débit cible est imposé par le standard en fonction des paramètres systèmes utilisés. Comme nous l'avons vu, selon l'architecture développée il est possible que le débit binaire d'information en entrée du système ne soit pas continu. Cette discontinuité en entrée est due au traitement des blocs qui imposent l'arrêt du flux de données en entrée de la chaîne pour ne pas perdre d'informations. Pour

satisfaire le débit cible  $R$  (en bit/s), la fréquence de fonctionnement  $f$  doit être ajustée et est donnée par l'équation suivante :

$$f = \beta \cdot R \geq R \quad (4.4)$$

- où  $\beta = \frac{T_{chaîne}}{T_{in,1}}$  si  $T_{chaîne} > T_{in,1}$
- sinon,  $\beta = 1$

## 4.2.2 Application à la chaîne Wi-Fi

Dans cette section, nous appliquons l'analyse des flux de données de la section précédente au cas particulier d'une chaîne d'émission Wi-Fi. Dans ce cas, une trame correspond aux données nécessaires à la construction d'un symbole OFDM.

Nous utilisons les notations suivantes:

- $N_{info}$  : nombre de bits de données par trame, en entrée de la chaîne
- $r$  : rendement de code
- $K$  : nombre de bits par symbole complexe
- $N_{data}$  : nombre de symboles complexes de données par symbole OFDM
- $N_{FFT}$  : taille de la FFT/IFFT
- $N_{CP}$  : taille du préfixe cyclique

Les valeurs de  $N_{data}$ ,  $N_{FFT}$  et  $N_{CP}$  sont données au Tab. 3.6.

En entrée de la chaîne, nous avons  $N_{info}$  bits de données, correspondant à  $N_{info}$  cycles d'horloge. En sortie du premier bloc (embrouilleur), nous avons le même nombre de cycles d'horloge. En sortie du bloc codeur de rendement-mère 1/2, nous avons  $2N_{info}$  bits, qui nécessitent  $N_{info}$  cycles d'horloge, car l'architecture choisie permet d'avoir 2 bits par cycle en sortie du codeur. L'étape de poinçonnage fournit  $N_{info}/r$  bits, correspondant à  $N_{info}/2r$  cycles. L'étape d'entrelacement, réalisée dans le même bloc que le poinçonnage, ne modifie pas cette durée. Or,  $N_{info}$  est égal à  $r \cdot K \cdot N_{data}$ . Ainsi, en entrée du bloc de modulation et mise en trame, le nombre de cycles par trame est donc égal à

$$T_{in,A} = \frac{N_{info}}{2r} = \frac{K}{2} \cdot N_{data}. \quad (4.5)$$

L'architecture d'IFFT choisie impose de prendre un symbole complexe par cycle. Nous avons donc, en sortie du bloc de modulation et de mise en trame,  $N_{FFT}$  cycles d'horloge. L'écart entre  $K \cdot N_{data}/2$  et  $N_{FFT}$  correspond à l'ajout des symboles pilotes et des symboles nuls.

Le Tab. 4.1 récapitule, pour chaque bloc de la chaîne d'émission, les temps d'entrée, de sortie et de process, en nombre de cycles tels que définis à la section précédente. A l'aide de ce tableau, il est aisé de déterminer les temps  $T_m$ , définis par (4.1) pour chaque bloc non pipeliné, et par (4.2) pour le bloc IFFT, qui est pipeliné. La variable  $x$  représente la taille du pipeline, c'est-à-dire le temps de calcul d'une trame OFDM par le bloc IFFT. Etant donné que le bloc est pipeliné ce traitement n'entre pas en compte dans le calcul du taux d'activité.

	$T_{in,m}$	$T_{proc,m}$	$T_{out,m}$	$T_m$
Embrouilleur	$r.K.N_{data}$	$r.K.N_{data}$	$r.K.N_{data}$	$r.K.N_{data}$
Codeur 1/2	$r.K.N_{data}$	$r.K.N_{data}$	$r.K.N_{data}$	$r.K.N_{data}$
Poinçonnage + Entrelaceur	$r.K.N_{data}$	$r.K.N_{data}$	$K.N_{data}/2$	$r.K.N_{data}$
Modulation + Mise en trame	$K.N_{data}/2$	$K.N_{data}/2$	$N_{FFT}$	$K.N_{data}/2$
IFFT	$N_{FFT}$	$x$	$N_{FFT} + N_{CP}$	$N_{FFT} + N_{CP}$

TAB. 4.1 : TEMPS DE TRAITEMENT DES BLOCS EN NOMBRE DE CYCLES

En précisant que le rendement de code  $r$  est toujours supérieur ou égal à 1/2, la période de traitement d'un symbole OFDM par la chaîne complète est alors nécessairement :

$$T_{chaîne} = \max_{1 \leq m \leq M} \{T_m\} = \max\{r.K.N_{data}; N_{FFT} + N_{CP}\} \quad (4.6)$$

Le temps de traitement du 1er bloc vaut  $T_{in,1} = N_{info}$ , donc en appliquant le raisonnement de la section précédente, nous pouvons conclure que:

- si  $N_{info} \geq N_{FFT} + N_{CP}$  alors le flux de données d'entrée de la chaîne est continu
- sinon, le flux est discontinu

Le Tab. 4.2 résume les valeurs  $T_{in,1}$  et  $T_{chaîne}$  pour les MCS allant de 0 à 8 de la norme IEEE 802.11ac dans le cas d'une largeur de bande égale à 20, 40 et 80 MHz. Ces valeurs sont obtenues à partir des tableaux Tab. 3.1 et Tab. 3.6. Ainsi on remarque que pour les MCS 0 et MCS 1, le flux d'entrée n'est pas continu; alors que dans le cas où le MCS est supérieur ou égal à 3, le flux est continu. Pour le cas du MCS 2, en fonction de la largeur de bande utilisée, deux cas sont

possibles. Si l'on est dans la bande des 20 MHz alors le flux de données n'est pas continu mais si l'on se trouve dans les bandes 40 et 80 MHz alors le flux est continu.

		Flux de données $T_{in,1}$	$T_{chaîne}$	Flux de données continu
20MHz	MCS0	26	80	non
	MCS1	52	80	non
	MCS2	78	80	non
	MCS3	104	80	oui
	MCS4	156	80	oui
	MCS5	208	80	oui
	MCS6	234	80	oui
	MCS7	260	80	oui
40MHz	MCS8	312	80	oui
	MCS0	54	160	non
	MCS1	108	160	non
	MCS2	162	160	oui
	MCS3	216	160	oui
	MCS4	324	160	oui
	MCS5	432	160	oui
	MCS6	486	160	oui
80MHz	MCS7	540	160	oui
	MCS8	648	160	oui
	MCS0	117	320	non
	MCS1	234	320	non
	MCS2	351	320	oui
	MCS3	468	320	oui
	MCS4	702	320	oui
	MCS5	936	320	oui
MCS6	1053	320	oui	
MCS7	1170	320	oui	
MCS8	1404	320	oui	

TAB. 4.2 : ANALYSE DU FLUX DE DONNEES D'INFORMATION BINAIRE EN NOMBRE DE CYCLES.

### 4.3 Méthodologie

L'estimation de la puissance consommée par la chaîne requiert un nombre important de simulations, typiquement au moins une pour chaque jeu de paramètres système. De la même manière si nous changeons l'architecture d'un bloc, il est alors nécessaire de simuler une nouvelle fois toute la chaîne afin d'avoir l'activité temporelle et la consommation de puissance totale du système. Les temps de simulation étant trop longs, cette manière de faire n'est pas appropriée. En ce sens, nous avons mis au point une méthode d'estimation rapide et précise de la consommation à haut niveau d'abstraction. Cette méthode repose sur le calcul mathématique de l'activité temporelle du système au niveau algorithmique. Elle est divisée en 3 étapes:

- 
- Création d'une base de données contenant la puissance consommée de chaque bloc élémentaire composant la chaîne d'émission/réception. Cette puissance est obtenue par simulation au niveau transistor.
  - Calcul mathématique de l'activité temporelle des blocs de la chaîne au niveau algorithmique.
  - Calcul de la puissance consommée de la chaîne en prenant en compte les activités temporelles de chaque bloc et en sommant les puissances individuelles.

Chacune de ces étapes sera détaillée dans les sections suivantes.

#### **4.3.1 Base de données**

La première étape de notre méthode consiste à évaluer la puissance consommée de chaque bloc élémentaire de la chaîne de communication à l'aide de XPower Analyzer. L'évaluation du bloc seul est réalisée lors d'une activité continue, cela signifie que le bloc fonctionne à 100% d'activité.

La création de la base de données nécessite de réaliser la simulation de chaque bloc pour chaque paramètre, mais une fois cet effort réalisé, aucune simulation supplémentaire n'est nécessaire pour évaluer la consommation de la chaîne globale pour n'importe quelle association de blocs élémentaires.

La fréquence d'horloge utilisée pour construire la base de données est fixée arbitrairement. En effet, elle est proportionnelle à la puissance dynamique comme le montre l'équation (1.8). Cela signifie que nous pouvons par la suite atteindre la fréquence souhaitée en exploitant la proportionnalité. Nous avons montré au chapitre 3 que l'estimation de puissance obtenue est précise puisque la simulation est effectuée au niveau transistor.

#### **4.3.2 Taux d'activité**

Nous avons vu au chapitre précédent que la somme des puissances dynamiques des blocs isolés à 100% d'activité n'était pas égale à la puissance de la chaîne composée par ces blocs. En effet, lorsqu'un bloc est inséré dans une chaîne, celui-ci peut ne pas fonctionner en continu du fait de l'impact des autres blocs ou bien de l'architecture globale de la chaîne. Cela signifie que son taux d'activité au niveau algorithmique peut être inférieur à 100%.

Le taux d'activité est calculé à l'aide d'un diagramme temporel qui permet de représenter les trois paramètres présentés à la section précédente qui sont pour rappel, les temps d'entrées, de sorties

et de traitement d'une trame de données. Ainsi le taux d'activité du  $m^{ième}$  bloc de la chaîne est défini par l'équation (4.7).

$$A_m = T_m / T_{chaîne} \quad (4.7)$$

### 4.3.3 Application numérique

La dernière étape de notre méthode consiste simplement à exploiter les deux premières étapes afin de calculer la puissance dynamique de la chaîne. Nous avons vu à la section précédente qu'il est possible de calculer le taux d'activité réel des blocs une fois insérés dans la chaîne ainsi que la fréquence d'horloge permettant d'atteindre le débit cible, comme précisé à l'équation (4.4). En appliquant l'activité de chaque bloc contenu dans la base de données créée préalablement, nous pouvons avoir la puissance totale de la chaîne définie par l'expression suivante:

$$P_f = \sum_{m=1}^M A_m P_m(f) \quad (4.8)$$

où  $P_f$  représente la puissance hiérarchique dynamique de la chaîne et  $P_m(f)$  représente la puissance dynamique du  $m^{ième}$  bloc présente dans la base de donnée à la fréquence d'horloge  $f$  permettant d'atteindre le débit cible.

Il est à noter que les puissances dynamiques présentées dans la suite du chapitre ne représentent que la partie hiérarchique (logique, signaux, BRAM et DSP), cela signifie que la contribution de l'horloge et des entrées/sorties a été volontairement soustraite de la puissance dynamique totale. En ce qui concerne les entrées/sorties, cette ressource a été supprimée car notre objectif est d'étudier la consommation d'énergie de chaque bloc, sans tenir compte de ses interactions externes. Quant à l'horloge, cette ressource est constamment active que le bloc soit actif ou non, sa consommation est donc perçue comme une valeur constante quelle que soit l'activité du bloc. Pour cette raison elle n'entre pas en considération pour la comparaison entre deux blocs ou pour l'étude des taux d'activité.

## 4.4 Exemple

Nous avons vu précédemment que l'architecture globale de la chaîne est dépendante du MCS. Dans le cas d'un  $MCS \leq 1$ , le flux de données en entrée n'est pas continu alors que pour un  $MCS \geq 3$ , le flux de données est continu. Dans cette section nous allons illustrer ces deux cas par des exemples, dans la bande (bandwidth, BW) de 20 MHz, MCS 1 et MCS 5.

Pour calculer les taux d'activité des blocs lors du fonctionnement de la chaîne, nous avons vu aux sections précédentes qu'il fallait suivre plusieurs étapes qui sont :

1. Détermination des paramètres systèmes
2. Calcul de  $T_{in}$ ,  $T_{proc}$ ,  $T_{out}$  et détermination de  $T_m$
3. Détermination du bloc le plus contraignant  $T_{chain}$
4. Calcul du taux d'activité de chaque bloc
5. Calcul de la fréquence d'horloge

#### 4.4.1 Chaîne BW 20 MCS 1

1. Détermination des paramètres systèmes

Le diagramme temporel Fig. 4.1 montre pour chaque bloc de la chaîne d'émission: en bleu les entrées/sorties valides, et en vert le temps nécessaire pour traiter les données contenues dans un symbole OFDM. Dans cet exemple, la bande passante est égale à 20 MHz et le MCS est égal à 1, ce qui correspond à une modulation QPSK 1/2. L'évaluation du taux d'activité est réalisée lorsque tous les blocs sont actifs en même temps, c'est-à-dire en cours de fonctionnement, de sorte que les latences ne sont pas prises en compte. Cependant, tous les chronogrammes de cette section montrent les latences afin de mieux comprendre le fonctionnement des blocs les uns par rapport aux autres.

Les paramètres système sont donc les suivants:  $r = 1/2$ ;  $K = 2$ ;  $N_{data} = 52$ ;  $N_{FFT} = 64$ ;  $N_{CP} = 16$ .

2. Calcul de  $T_{in}$ ,  $T_{proc}$ ,  $T_{out}$  et détermination de  $T_m$

En appliquant les formules fournies Tab. 4.1, il est possible de déterminer les valeurs de  $T_{in}$ ,  $T_{proc}$  et  $T_{out}$ . Ainsi, en utilisant les équations (4.1) pour un bloc non pipeliné et (4.2) lorsque le bloc est pipeliné, il est aisé de déterminer  $T_m$ . Les résultats sont montrés Tab. 4.3.

3. Détermination du bloc le plus contraignant  $T_{chain}$

On observe que dans cette configuration, le bloc le plus contraignant est la IFFT puisque sa sortie nécessite 80 cycles d'horloge,  $T_{chain}$  est donc égal à 80, tel que montré dans le Tab. 4.2. Il est à noter que le nombre de cycles d'horloge pour effectuer le traitement d'un symbole OFDM par le bloc IFFT d'une taille 64 est de 205 cycles d'horloge. Cette valeur est fournie dans la documentation de l'IPCore de la FFT [78].

4. Calcul du taux d'activité de chaque bloc

Maintenant que nous avons déterminé  $T_{chain}$  ainsi que le  $T_m$  de chaque bloc, nous pouvons déterminer l'activité de chaque bloc en utilisant l'équation (4.7). Les taux d'activités sont donnés Tab. 4.3.

#### 5. Calcul de la fréquence d'horloge

On observe que le bloc de mise en trame est tout le temps alimenté par des données, mais qu'il est nécessaire d'arrêter ce flux durant l'insertion des pilotes et des symboles nuls pour éviter de perdre des données.

Cette modification impacte tous les blocs précédant le bloc de mise en trame. En effet si celui-ci n'est pas en mesure de traiter les bits en entrée alors il va envoyer un signal d'arrêt à tous les blocs le précédant afin de stopper le flux et figer les registres et les mémoires. Ainsi, tant que le bloc de mise en trame ne peut pas traiter les bits entrants, tous les blocs précédents restent figés jusqu'à ce qu'il envoie un signal débloquant le flux de la chaîne. De la même manière, en entrée de la IFFT, le flux de données binaires doit être stoppé afin d'avoir le temps d'insérer le préfixe cyclique. Ainsi, comme le flux de bits d'information en entrée n'est pas transmis en continu, la fréquence de fonctionnement du système doit être ajustée afin de respecter les contraintes liées au débit en utilisant l'équation (4.4).

Dans ce cas de figure, la fréquence d'horloge est égale à  $f = \beta.R = 20 \text{ MHz}$  où  $\beta$  est égal à  $80/52$  et  $R$  est égal à  $13 \text{ Mbps}$ .

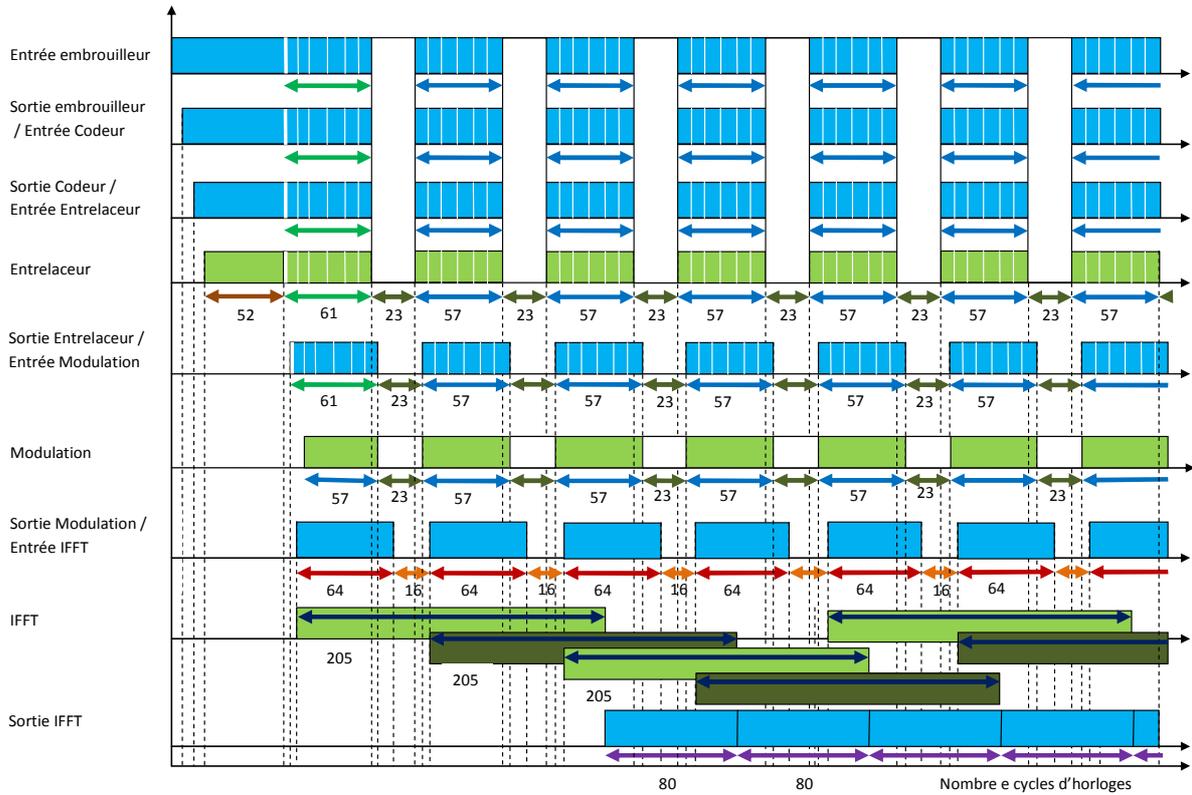


FIG. 4.1 : CHRONOGRAMME DE LA CHAINE D'EMISSION, BW 20 MHz MCS1

	$T_{in}, T_{proc}, T_{out}$	$T_m(4.2)$	Taux d'activite (4.7)
Embrouilleur	$T_{in} = 52$	52	65%
	$T_{proc} = 52$		
	$T_{out} = 52$		
Codeur	$T_{in} = 52$	52	65%
	$T_{proc} = 52$		
	$T_{out} = 52$		
Entrelaceur	$T_{in} = 52$	52	65%
	$T_{proc} = 52$		
	$T_{out} = 52$		
Modulation	$T_{in} = 52$	64	80%
	$T_{proc} = 52$		
	$T_{out} = 64$		
IFFT	$T_{in} = 64$	80	100%
	$T_{proc} = 205$		
	$T_{out} = 80$		

TAB. 4.3 : TAUX D'ACTIVITE DE LA CHAINE D'EMISSION, BW 20 MHz MCS1

#### 4.4.2 Chaîne BW 20 MCS 5

1. Détermination des paramètres systèmes

Le diagramme temporel Fig. 4.2 correspond à une bande passante égale à 20 MHz et un MCS égal à 5. Dans cet exemple, les paramètres système sont les suivants:  $r = 2/3$ ;  $K = 6$ ;  $N_{data} = 52$ ;  $N_{FFT} = 64$ ;  $N_{CP} = 16$ .

2. Calcul de  $T_{in}$ ,  $T_{proc}$ ,  $T_{out}$  et détermination de  $T_m$

En utilisant les informations les formules fournies Tab. 4.1, ainsi que les équations (4.1) et (4.2), il est possible de déterminer  $T_m$ . Les résultats sont montrés Tab. 4.4.

3. Détermination du bloc le plus contraignant  $T_{chain}$

Le bloc le plus contraignant est le premier bloc de la chaîne, qui nécessite 208 cycles d'horloge pour traiter les bits de données contenus dans un symbole OFDM, ainsi  $T_{chain} = T_{in,1}$ ; le flux d'entrée est donc continu. Il est intéressant d'observer que le nombre de cycles à l'entrée de l'entrelaceur est supérieur à celui de la sortie, cela est dû au taux de codage qui est de  $2/3$  dans ce cas de figure.

4. Calcul du taux d'activité de chaque bloc

En utilisant l'équation (4.7), il est possible de calculer les taux d'activités de chaque bloc. Ceux-ci sont illustrés Tab. 4.4.

5. Calcul de la fréquence d'horloge

Puisque  $T_{chain} = T_{in,1}$ , la fréquence d'horloge est égale au débit imposé par la norme. Dans ce cas de figure,  $R$  est égal à 52 Mbps.

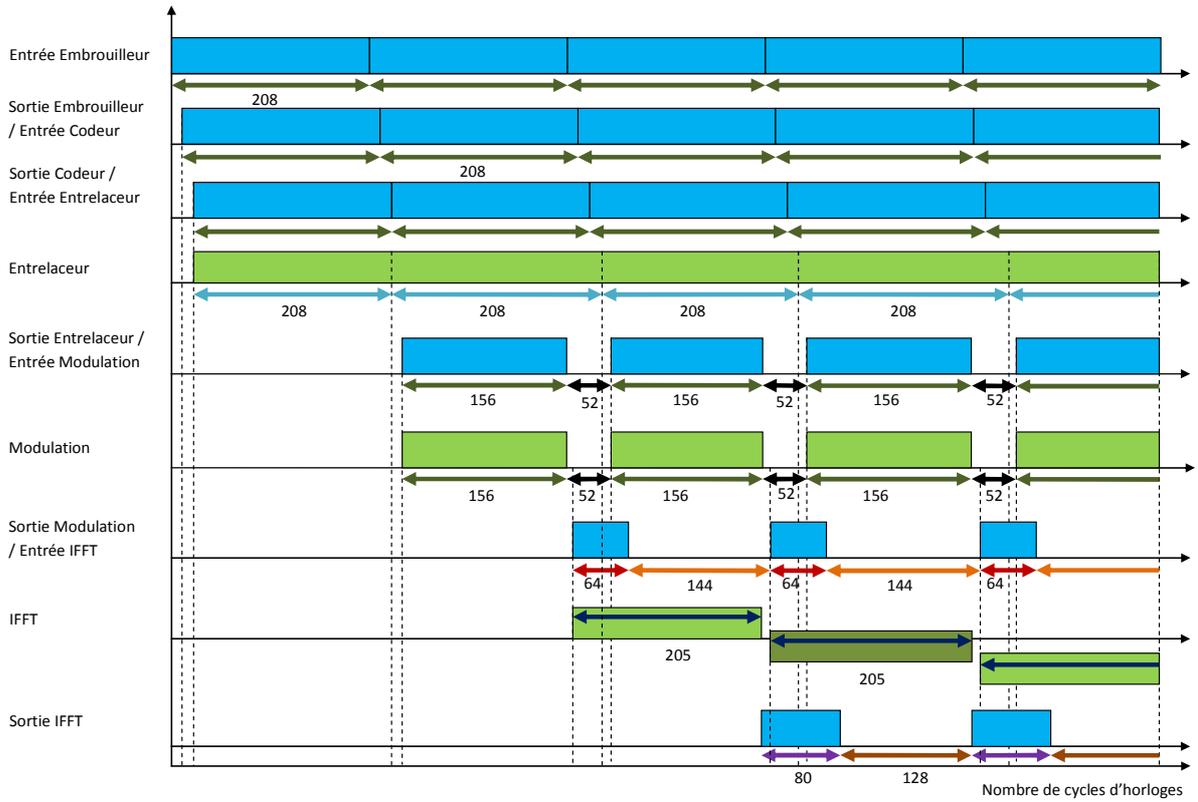


FIG. 4.2 : CHRONOGRAMME DE LA CHAINE D'EMISSION, BW 20 MHz MCS 5

	$T_{in}, T_{proc}, T_{out}$	$T_m$ (4.2)	Taux d'activité (4.7)
Embrouilleur	$T_{in} = 208$	208	100%
	$T_{proc} = 208$		
	$T_{out} = 208$		
Codeur	$T_{in} = 208$	208	100%
	$T_{proc} = 208$		
	$T_{out} = 208$		
Entrelaceur	$T_{in} = 208$	208	100%
	$T_{proc} = 208$		
	$T_{out} = 156$		
Modulation	$T_{in} = 156$	156	75%
	$T_{proc} = 156$		
	$T_{out} = 64$		
IFFT	$T_{in} = 64$	80	38.4%
	$T_{proc} = 205$		
	$T_{out} = 80$		

TAB. 4.4 : TAUX D'ACTIVITE DE LA CHAINE D'EMISSION, BW 20 MHz MCS 5

### 4.4.3 Taux d'activité des blocs

Les taux d'activité (au niveau algorithmique) de chaque bloc sont calculés comme décrit à la section 4.3.2 à l'aide des chronogrammes et sont fournis Tab. 4.5 et Tab. 4.6. Il est intéressant d'observer que les blocs en réception permettant d'inverser les calculs faits à l'émission ont le même taux d'activité. Par exemple, les blocs embrouilleur/désembrouiller, codeur/décodeur ou encore IFFT/FFT ont le même taux d'activité. Néanmoins cela n'est pas vérifié pour les blocs entrelaceur et modulation. Cela est dû au fait qu'à l'émission le bloc entrelaceur est fusionné avec le bloc poinçonnage et que le bloc modulation est fusionné avec le bloc mise en trame. En effet dans le cas d'un rendement 1/2, le bloc poinçonnage n'a aucun rôle donc l'entrelaceur et le désentrelaceur ont la même activité. Lorsque le rendement de code est différent de 1/2, l'activité n'est plus la même. Le même raisonnement est effectué pour le bloc modulation.

La fréquence de fonctionnement est calculée afin de respecter les débits cibles fixés par la norme. La Fig. 4.3 montre l'évolution de la puissance dynamique du bloc IFFT de taille 256 en fonction de la fréquence d'horloge. On observe que la courbe est linéaire et que la puissance dynamique est bien proportionnelle à la fréquence d'horloge.

Largeur de bande	MCS	Fréquence d'horloge	embrouilleur	codeur	entrelaceur	modulation	IFFT
20 MHz	MCS1	20MHz	65%	65%	65%	80%	100%
	MCS3	26MHz	100%	100%	100%	100%	77%
	MCS5	52 MHz	100%	100%	100%	75%	38.4%
	MCS8	78 MHz	100%	100%	100%	66.7%	25.6%
40 MHz	MCS1	40 MHz	67.5%	67.5%	67.5%	80%	100%
	MCS3	54 MHz	100%	100%	100%	100%	74%
	MCS5	108 MHz	100%	100%	100%	75%	37%
	MCS8	162 MHz	100%	100%	100%	66.7%	24.6%
80 MHz	MCS1	80 MHz	73%	73%	73%	80%	100%
	MCS3	117 MHz	100%	100%	100%	100%	68%

TAB. 4.5 : TAUX D'ACTIVITE DES BLOCS A L'EMISSION LORS DU FONCTIONNEMENT DE LA CHAINE

Largeur de bande	MCS	Fréquence d'horloge	Désembrouilleur	Décodeur	Désentrelaceur	Démodulation	FFT
20 MHz	MCS1	20MHz	65%	65%	65%	65%	100%
	MCS3	26MHz	100%	100%	100%	100%	77%
	MCS5	52 MHz	100%	100%	75%	75%	38.4%
	MCS8	78 MHz	100%	100%	66.7%	66.7%	25.6%
40 MHz	MCS1	40 MHz	67.5%	67.5%	67.5%	67.5%	100%
	MCS3	54 MHz	100%	100%	100%	100%	74%
	MCS5	108 MHz	100%	100%	75%	75%	37%
	MCS8	162 MHz	100%	100%	66.7%	66.7%	24.6%
80 MHz	MCS1	80 MHz	73%	73%	73%	73%	100%
	MCS3	117 MHz	100%	100%	100%	100%	68%

TAB. 4.6 : TAUX D'ACTIVITE DES BLOCS EN RECEPTION LORS DU FONCTIONNEMENT DE LA CHAINE

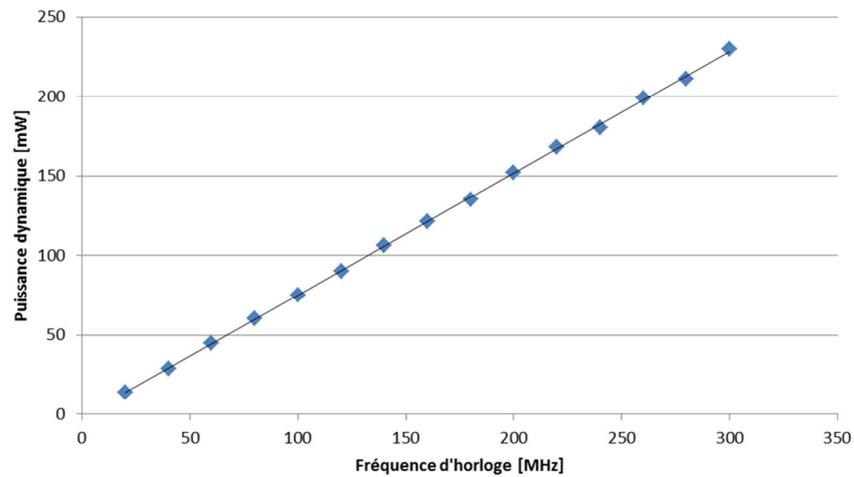


FIG. 4.3: *PUISSANCE DYNAMIQUE SIMULEE DU BLOC IFFT DE TAILLE 256 EN FONCTION DE LA FREQUENCE D'HORLOGE*

## 4.5 Conclusion

Nous avons vu que les estimations de puissance bas niveau (niveau transistor) sont précises mais nécessitent un temps de simulation important ce qui est problématique lorsque l'on souhaite faire de l'exploration d'architecture avec critère la consommation de puissance. Les estimations faites à haut niveau d'abstraction sont beaucoup plus rapides mais sont peu précises pour évaluer la consommation de puissance. Pour remédier à ces difficultés, nous proposons une estimation de puissance combinant le haut niveau et le bas niveau.

Dans ce chapitre, une analyse temporelle du système Wi-Fi est réalisée et permet d'analyser le comportement de chaque bloc en fonction de la taille de ses entrées/sorties, ainsi que des temps nécessaires au traitement des données. Basée sur cette analyse, une méthode d'estimation de la consommation rapide et précise est proposée. Cette méthode inclut la construction préalable d'une base de données contenant les consommations des blocs dans une configuration de référence. La consommation est réalisée au niveau transistor sous XPower. Puis, nous calculons le taux d'activité de chaque bloc dans la configuration souhaitée en observant les entrées, sorties et temps de traitement d'une trame pour chaque bloc. Dans certains cas, nous avons vu que le flux de données en entrée doit être arrêté pour ne pas perdre de données. Dans ce cas, nous devons augmenter la fréquence d'horloge afin de satisfaire les contraintes de débits fixées par la norme. Enfin, nous sommes la consommation des blocs présente dans la base de données en prenant en compte l'activité réelle de chaque bloc calculée analytiquement.

De cette manière, nous pouvons évaluer rapidement la consommation du système. Cette méthode peut également être adaptée à n'importe quel système de communication numérique.

Dans le chapitre suivant nous validons et montrons les avantages de cette méthode notamment à travers l'estimation de la puissance des chaînes Tx/Rx. Nous verrons que la méthode proposée permet également de faciliter l'exploration d'architecture à l'aide d'un exemple sur le bloc IFFT.

## Chapitre 5

# Résultats de mesures et de simulations

### 5.1 Introduction

La méthode proposée au chapitre précédent permet d'estimer la consommation de puissance d'une chaîne de communications numériques rapidement et précisément, en se basant sur l'analyse temporelle du système. La rapidité d'estimation est liée au fait qu'il n'est pas nécessaire d'évaluer l'ensemble de la chaîne lorsque l'on cherche à optimiser un bloc. La précision est obtenue grâce à des simulations de puissance bas niveau via XPower.

Dans ce chapitre, nous appliquons cette méthode au contexte Wi-Fi. En particulier, nous comparons l'estimation de la puissance totale consommée par la chaîne avec la somme des puissances évaluées individuellement pour chaque bloc, ce qui nous permet de valider la méthode proposée.

Par la suite, nous illustrons l'utilisation de la méthode dans le cadre d'une exploration d'architecture avec comme critère la minimisation de la consommation de puissance. Pour cela, nous comparons deux architectures IFFT : une architecture pipelinée et une non pipelinée, sachant que l'architecture pipelinée nécessite plus de ressources mais permet d'optimiser l'activité de la chaîne. La question que l'on se pose dans ce chapitre est donc « comment le choix d'une architecture pipelinée impacte-t-elle la consommation de puissance globale de la chaîne par rapport à une architecture non pipelinée ? »

### 5.2 Validation des taux d'activité

Afin de valider les taux d'activité trouvés à la section précédente par calcul, nous les avons comparés à des résultats obtenus en simulation. Nous avons dans un premier temps simulé toute la chaîne dans le but de sauvegarder les sorties de chaque bloc dans des fichiers, appelés fichiers

d'activité. Ces fichiers représentent les données d'entrée réelles (activité) des blocs. Puis, nous avons simulé chaque bloc séparément en plaçant en entrée les fichiers d'activité. Chaque bloc est placé en « top module », cela signifie qu'il est seul dans le design et que ses entrées/sorties sont contrôlées uniquement à travers les fichiers d'activité. De cette façon, nous avons pu évaluer la consommation de puissance de chaque bloc en prenant en compte leurs taux d'activité réels.

Le Tab. 5.1 et Tab. 5.2 donnent les puissances consommées des blocs modulation et IFFT, obtenues via les deux stratégies proposées suivantes:

- 1ere stratégie - méthode proposée au chapitre précédent : la puissance du bloc fonctionnant à 100% d'activité est multipliée par le taux d'activité théorique (4.8)
- 2eme stratégie - validation: le bloc est simulé en utilisant un fichier d'activité en entrée. La puissance est évaluée par simulation (XPower).

On remarque que les différences entre ces deux méthodes sont très faibles ce qui conforte les expressions théoriques du calcul du taux d'activité présentées au chapitre précédent.

Nous rappelons que les puissances consommées des blocs et des chaînes présentes dans ce chapitre ne prennent pas en compte la puissance des entrées/sorties et de la fréquence d'horloge. Ces résultats sont donc différents de ceux présentés au chapitre 3 qui prenaient en compte la puissance liée à la fréquence d'horloge mais pas la puissance des entrées/sorties.

Largeur de bande	20 MHz				40 MHz				80 MHz	
	MCS	1	3	5	8	1	3	5	8	1
Méthode proposée (4.7)	0	3	4	6	0	7	14	38	1	29
Validation par simulation	0	2	4	6	0	7	13	38	1	29

TAB. 5.1 : PUISSANCE CONSOMMEE DU BLOC MODULATION [mW]

Largeur de bande	20 MHz				40 MHz				80 MHz	
	MCS	1	3	5	8	1	3	5	8	1
Méthode proposée (4.7)	7	8	9	9	18	21	21	22	43	46
Validation par simulation	7	8	8	9	18	20	20	22	43	47

TAB. 5.2 : PUISSANCE CONSOMMEE DU BLOC IFFT [mW]

## 5.3 Chaîne Tx/Rx

Afin de compléter la validation vue à la section précédente, nous allons comparer la puissance consommée des chaînes Tx et Rx issue de la méthode semi-analytique développée au chapitre 4 avec les résultats de simulation fournis par l’outil XPower Analyzer.

### 5.3.1 Construction de la base de données

Nous avons dans un premier temps créé une base de données contenant la consommation de puissance des blocs à 100% d'activité, à l’aide de l’outil XPower Analyser selon la méthodologie décrite à la section 4.3.1. La fréquence d’horloge choisie permet de respecter les débits fixés par la norme. Ces fréquences d’horloge seront exposées au chapitre suivant à la suite de l’analyse temporelle. Le Tab. 5.3 fournit les résultats pour chaque scénario. Nous remarquons à nouveau que le bloc IFFT et le bloc décodeur sont les plus gourmands en énergie, respectivement à l’émission et à la réception. Le bloc entrelaceur/désentrelaceur est également consommateur d’énergie, notamment pour les MCS élevés.

Largeur de bande	MCS	embrouilleur	codeur	entrelaceur	modulation	IFFT	démodulation	désentrelaceur	décodeur
20 MHz	MCS1	0	0	1	0	9	1	3	49
	MCS3	0	0	3	2	11	3	3	64
	MCS5	0	0	13	5	23	15	15	129
	MCS8	0	0	25	9	34	18	16	194
40 MHz	MCS1	0	0	4	0	18	3	8	99
	MCS3	0	0	12	7	28	7	19	134
	MCS5	0	0	51	19	58	45	70	266
	MCS8	0	0	72	57	89	49	82	399
80 MHz	MCS1	0	0	22	1	43	3	37	197
	MCS3	0	0	58	29	67	25	57	291

TAB. 5.3 : PUISSANCE DES BLOCS [mW] - 100 % D'ACTIVITE (BASE DE DONNEES)

### 5.3.2 Calcul des puissances consommées réelles à partir des taux d'activité

Nous avons multiplié les puissances du Tab. 5.3 par les taux d'activité théoriques des blocs (Tab. 4.5 et Tab. 4.6) pour obtenir les puissances réelles lorsque les blocs sont inclus dans la chaîne. Ces puissances sont données au Tab. 5.4.

Largeur de bande	MCS	embrouilleur	codeur	entrelaceur	modulation	IFFT	démodulation	désentrelaceur	décodeur
20 MHz	MCS1	0	0	0.8	0	9	0.6	1.9	31.9
	MCS3	0	0	3	2	8.5	3	3	64
	MCS5	0	0	13	4	9	11.2	11.2	129
	MCS8	0	0	25	6	8.7	12	10.7	194
40 MHz	MCS1	0	0	3.2	0	18	2	5.4	66.8
	MCS3	0	0	12	7	20.7	7	19	134
	MCS5	0	0	51	14.2	21.5	33.7	52.5	266
	MCS8	0	0	72	38	22	32.7	54.7	399
80 MHz	MCS1	0	0	17.6	0.7	43	2.2	27	143.8
	MCS3	0	0	58	29	46	25	57	291

TAB. 5.4 : PUISSANCE DES BLOCS [mW] - ACTIVITE REELLE

Le taux d'activité impacte fortement la répartition de la puissance consommée des blocs. Au chapitre précédent, nous avons vu que lorsque les blocs sont évalués seuls, la IFFT est le bloc le plus gourmand en énergie du côté de l'émission. Une fois que tous les blocs sont insérés dans la chaîne, cette affirmation n'est pas toujours vraie. La Fig. 5.1 montre l'évolution de la puissance des blocs entrelaceur et IFFT en fonction du débit. On remarque qu'à haut débit le bloc entrelaceur est plus gourmand en énergie que le bloc IFFT. Cela est dû au fait que l'activité du bloc IFFT n'est pas continue contrairement au bloc entrelaceur comme le montre le Tab. 4.5.

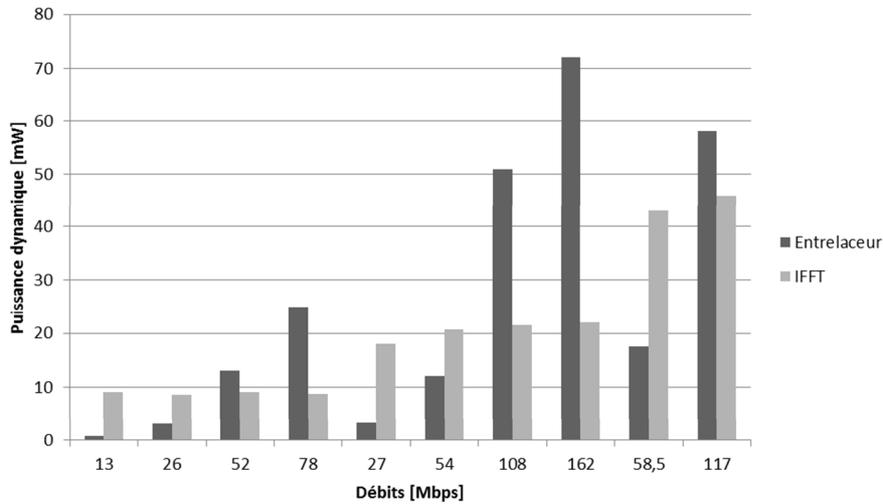


FIG. 5.1 : CONSOMMATION DE PUISSANCE DES BLOCS ENTRELACEUR ET IFFT LORS DU FONCTIONNEMENT DE LA CHAÎNE POUR DIFFERENTS SCENARIOS EN FONCTION DU DEBIT

### 5.3.3 Calcul de la puissance consommée de la chaîne par sommation

Nous avons sommé la puissance consommée des blocs suivant l'équation (4.8) pour avoir la consommation de puissance de la chaîne. Nous avons comparé ces résultats à ceux obtenus en simulant la chaîne dans son ensemble. L'écart est fourni Tab. 5.5 pour la chaîne d'émission et Tab. 5.6 pour la chaîne de réception. Nous observons que l'écart en moyenne est inférieur à 4.1%, ce qui peut être considéré comme acceptable et montre l'efficacité de notre méthode.

L'écart peut être dû à plusieurs facteurs, le premier étant que l'outil utilisé en simulation (XPower) a un pas d'analyse de 1 mW. On peut voir, par exemple dans le Tab. 5.5, que pour le premier cas (BW 20 MHz, MCS 1), l'écart est de 8,8 % alors que la différence entre notre méthode semi-analytique et la simulation de la chaîne est inférieure à 1 mW. Le deuxième facteur est la précision des valeurs des taux d'activité fournis Tab. 4.5 car les erreurs de précision se cumulent lorsqu'on effectue la somme (4.9).

Largeur de bande	MCS	Ecart	Méthode proposée (4.8)	Simulation XPower
20 MHz	MCS1	8.8%	9.8	9
	MCS3	3.8%	13.5	13
	MCS5	-7%	26	28
	MCS8	-3.2%	39.7	41
40 MHz	MCS1	0.9%	21.2	21
	MCS3	7.3%	39.7	37
	MCS5	-1.4%	86.7	88
	MCS8	-2.2%	133	136
80 MHz	MCS1	5.7%	61.3	58
	MCS3	-0.8%	133	132

TAB. 5.5 : PUISSANCE CONSOMMEE DE LA CHAINE D'EMISSION [mW]

Largeur de bande	MCS	Ecart	Méthode proposée (4.8)	Simulation XPower
20 MHz	MCS1	0.9%	43.4	43
	MCS3	4.6%	78.5	75
	MCS5	2.2%	160.4	157
	MCS8	-3.3%	225.4	233
40 MHz	MCS1	-1.9%	92.2	94
	MCS3	4%	181	174
	MCS5	-1.1%	373.7	378
	MCS8	-0.3%	508.4	510
80 MHz	MCS1	0.9%	216	214
	MCS3	0.7%	419	416

TAB. 5.6 : PUISSANCE CONSOMMEE DE LA CHAINE DE RECEPTION [mW]

Les simulations des chaînes effectuées sous XPower conduisent à d'énormes temps de simulation, de l'ordre de plusieurs jours. Afin d'avoir une évaluation fiable, nous avons simulé les blocs et les chaînes sur une transmission de 8 ms. Il est à noter que pour simuler ces 8 ms, le temps réel nécessaire à l'outil est supérieur, et varie en fonction de la complexité des blocs. Le bloc qui a nécessité le plus de temps de simulation est le décodeur du fait de sa complexité importante, les simulations pouvant durer de 5 heures à 4 jours suivant les scénarios.

Les temps de simulation de la chaîne de réception, qui inclut d'autres blocs, atteignent pour des hauts débits 7 jours de simulation pour une seule configuration. A l'aide de la méthode semi-analytique proposée, ce temps est largement réduit puisqu'il sera nécessaire de simuler uniquement le bloc impacté par le changement d'architecture. Il est également à noter que certains blocs ne dépendent pas de paramètres système ou utilisent la même architecture dans la même largeur de bande. Cela est le cas du bloc embrouilleur et du bloc FFT. Dans le cas de la FFT, pour une largeur de bande fixe, l'architecture est la même quel que soit le MCS utilisé. En ce sens, si l'on effectue une modification d'architecture du bloc FFT, avec notre méthode, une seule simulation du bloc est nécessaire alors qu'avec la simulation de la chaîne complète, il aurait fallu simuler la chaîne autant de fois qu'il y a de MCS.

Ainsi, un des avantages de cette méthode semi-analytique est le gain important en temps réalisé sur les simulations, car elle est basée sur un modèle mathématique ne nécessitant aucune autre simulation, une fois la base de données de référence créée.

Par conséquent, cette méthode peut être appliquée efficacement à l'exploration d'architecture d'un bloc spécifique ou pour évaluer les conséquences de l'insertion d'un nouveau bloc dans la chaîne en termes de consommation d'énergie, sans nécessiter de re-simuler la chaîne en entier.

## **5.4 Exploration d'architecture – cas de la FFT**

Dans cette section, nous allons mettre à profit les gains en temps de simulation fournis par la méthode semi-analytique d'évaluation de la puissance développée au chapitre 4, pour faire de l'exploration d'architecture rapide.

Nous allons comparer deux architectures de FFT, l'une pipelinée et l'autre non pipelinée. L'objectif est de minimiser la consommation de puissance de la chaîne de communication pour un débit cible fixé par la norme Wi-Fi 802.11ac. Comme le montre le Tab. 3.8, le bloc FFT est le plus complexe en terme de ressources utilisées et de calcul du côté de l'émission, et nous avons vu au chapitre 3 qu'il est l'un des blocs le plus consommateurs; il est par conséquent avantageux de connaître l'architecture la plus économe en énergie. En outre, ce bloc a un rôle prépondérant sur le débit de données, et son architecture influence grandement l'activité des autres blocs, et donc la consommation totale.

Une architecture pipelinée parallélise les traitements de plusieurs trames, ce qui permet d'augmenter le débit pour une fréquence d'horloge donnée. Cependant, cela implique plus

d'opérations simultanées qu'une architecture non pipelinée, impliquant plus de complexité, donc une augmentation de la consommation.

Dans la suite de cette section, pour simplifier les expressions, nous parlerons indifféremment de FFT, qu'il s'agisse de l'IFFT à l'émission ou de FFT en réception.

### 5.4.1 Consommation des blocs FFT isolés

Le Tab. 5.7 montre les ressources utilisées par les deux architectures de FFT. On rappelle que le bloc est un IPCore de Xilinx. L'architecture non pipelinée utilisée est une architecture radix-4 consistant en  $\log_4(N_{FFT})$  étages, où  $N_{FFT}$  correspond à la taille de la FFT. Chaque étage contient  $N_{FFT}/4$  radix-4 papillons. On remarque que l'architecture pipelinée est beaucoup plus complexe et utilise 3 fois plus de ressource.

	FFT 64 – Radix 4	FFT 64 – Pipeline
Nombre de slice registers	786	2143
Nombre de slice LUTs	660	2006
Nombre de FF	464	1395

TAB. 5.7 : RESSOURCES UTILISEES PAR LES DEUX ARCHITECTURES FFT

Le Tab. 5.8 montre la consommation des deux architectures FFT simulées seules, lors d'une activité continue. Sans surprise l'architecture pipelinée consomme plus de puissance que l'architecture non pipelinée car elle nécessite plus de ressources.

BW [MHz]	MCS	Fréquence d'horloge	Puissance dynamique de l'architecture FFT pipelinée [mW]	Puissance dynamique de l'architecture FFT non pipelinée [mW]
20	1	20MHz	9	6
	3	26MHz	11	8
	5	52 MHz	23	16
	8	78 MHz	34	23
40	1	40 MHz	20	13
	3	54 MHz	28	18
	5	108 MHz	58	37
	8	162 MHz	89	54
80	1	80 MHz	43	35
	3	117 MHz	64	52

TAB. 5.8 : PUISSANCE CONSOMMEE DES DEUX ARCHITECTURES FFT A 100% D'ACTIVITE [mW]

## 5.4.2 Analyse temporelle de la chaîne

Nous avons montré que la consommation d'un bloc intégré dans une chaîne dépend de son activité dans cette chaîne, ainsi que de l'activité des autres blocs. L'analyse temporelle de l'activité du bloc FFT intégré dans la chaîne est donc nécessaire pour comparer les deux architectures. On sait que la consommation augmente à la fois avec le taux d'activité et avec la complexité d'un bloc.

Les Fig. 5.2 et Fig. 5.3 montrent les chronogrammes d'une chaîne d'émission pour une bande passante de 40 MHz et un MCS égal à 1 avec une architecture pipelinée et une architecture non pipelinée, respectivement.

On remarque sur la Fig. 5.2 que dans le cas d'une architecture pipelinée, l'élément le plus contraignant de la chaîne est imposé par la sortie du bloc IFFT où  $T_{chaîne}$  est égal à 80 cycles d'horloge. Dans ce cas de figure, le bloc IFFT impose l'arrêt du flux de données valides durant 28 cycles d'horloge correspondant aux 4 pilotes et aux 12 sous-porteuses nulles à insérer dans la trame en entrée de la IFFT, en plus des 16 cycles correspondant à l'insertion du préfixe cyclique.

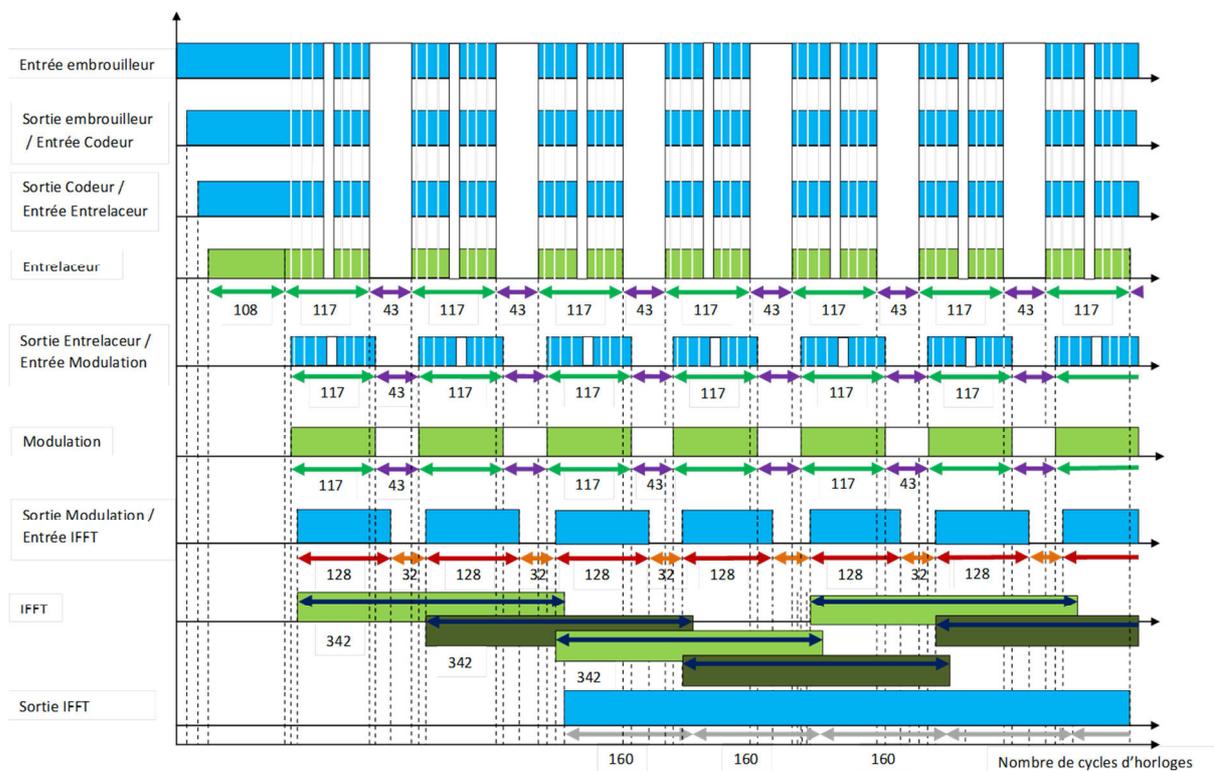


FIG. 5.2 : CHRONOGRAMME DE LA CHAÎNE D'ÉMISSION, POUR UNE ARCHITECTURE FFT PIPELINÉE, BW 40 MHz MCS 1

La Fig. 5.3 montre que dans le cas d'une architecture non pipelinée, l'élément le plus contraignant de la chaîne est sans aucun doute le bloc IFFT, puisque le fait de traiter les trames de données l'une après l'autre impose de longues périodes d'arrêt de la chaîne: 388 cycles d'horloge par trame. Ces arrêts imposent d'augmenter la fréquence de fonctionnement pour satisfaire les contraintes de débit fixées par la norme, et donc d'augmenter la consommation de puissance. Néanmoins, l'architecture non pipelinée nécessite beaucoup moins de ressources (notamment de mémoire) comme le montre le Tab. 5.7, ce qui a pour effet de diminuer la puissance consommée de la chaîne.

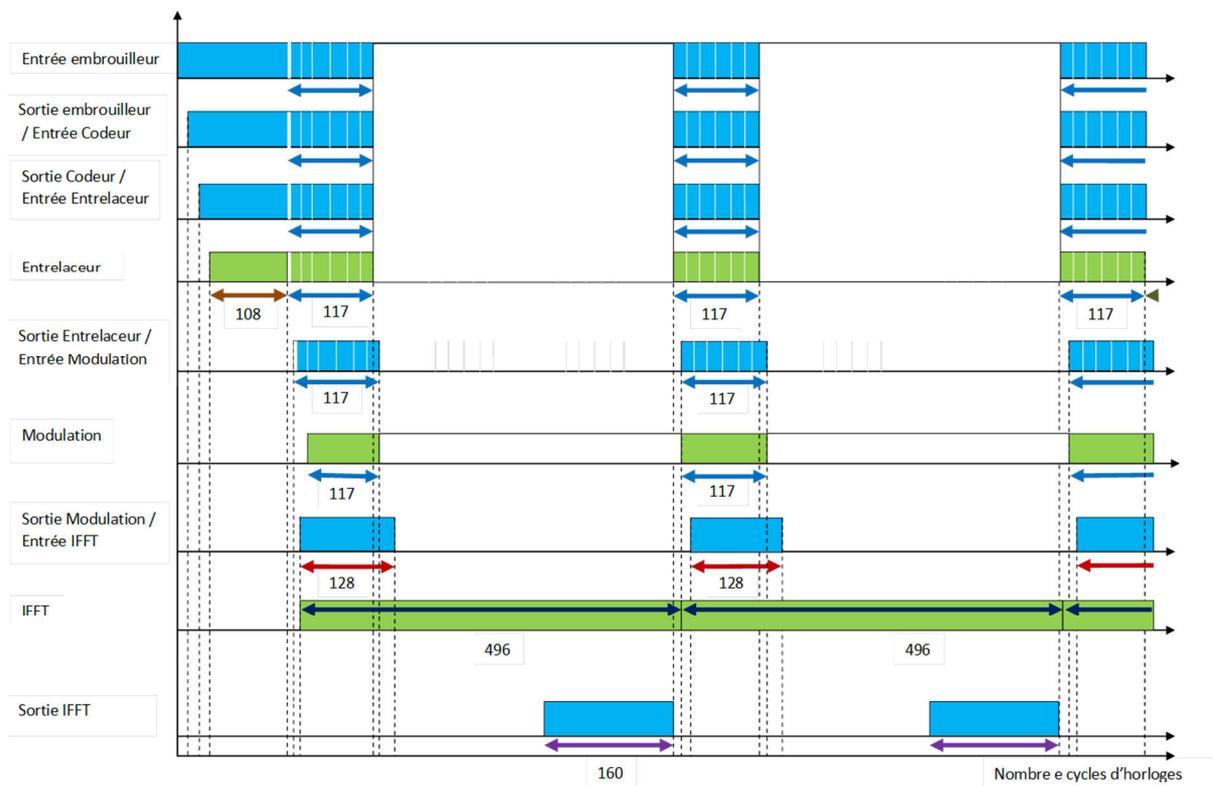


FIG. 5.3 : CHRONOGRAMME DE LA CHAÎNE D'ÉMISSION, POUR UNE ARCHITECTURE FFT NON PIPELINÉE, BW 40M HZ MCS 1

Nous confirmons à travers ces deux chronogrammes que la FFT pipelinée, bien que plus complexe que la FFT non pipelinée, permet un temps de traitement moins long, et donc une fréquence d'horloge moins élevée, pour un débit donné. Ces caractéristiques ont un effet contradictoire sur la consommation de la chaîne globale ; les simulations permettent de déterminer l'architecture la plus économe en énergie.

### 5.4.3 Impact de l'architecture de la FFT sur la consommation de la chaîne

Afin d'estimer la consommation de la chaîne en émission et en réception pour les 2 architectures de FFT, nous appliquons la méthode semi-analytique présentée au chapitre 4. Ainsi, l'estimation de la puissance via XPower n'est réalisée que sur l'architecture de FFT non pipelinée puisqu'elle n'est pas présente dans la base de données, pour une fréquence de fonctionnement arbitraire. La consommation est ensuite réajustée en fonction des taux d'activité et de la fréquence de fonctionnement cible.

Le Tab. 5.9 montre la puissance consommée de la chaîne d'émission dans le cas d'une architecture IFFT pipelinée et dans le cas d'une architecture IFFT non pipelinée. Le tableau montre également les fréquences de fonctionnement de la chaîne nécessaires pour atteindre les débits fixés par la norme. On constate que la chaîne d'émission incluant l'architecture IFFT pipelinée consomme moins d'énergie que la chaîne incluant l'architecture non pipelinée. Par exemple, dans le cas d'une largeur de bande égale à 20 MHz et d'un MCS 3, l'utilisation d'une architecture pipelinée fournit un gain de 46% sur la puissance consommée de la chaîne d'émission par rapport à une architecture non pipelinée. Les temps de traitement de l'architecture pipelinée sont beaucoup moins longs que ceux de l'architecture non pipelinée, ce qui permet d'avoir une fréquence de fonctionnement moins importante. Ainsi, nous pouvons observer que la fréquence de fonctionnement de la chaîne d'émission incluant la IFFT non pipelinée est largement plus importante que dans l'autre cas, excepté pour le MCS égal à 8.

BW [MHz]	MCS	Architecture FFT pipelinée		Architecture FFT non pipelinée	
		Puissance dynamique [mW]	Fréquence d'horloge [MHz]	Puissance dynamique [mW]	Fréquence d'horloge [MHz]
20	1	9	20	18	63.7
	3	13	26	24	63.7
	5	28	52	39	63.7
	8	41	78	52	78
40	1	21	40	47	124
	3	37	54	60	124
	5	88	108	107	124
	8	136	162	164	162

TAB. 5.9 : PUISSANCE CONSOMMEE DE LA CHAINE D'EMISSION AVEC L'ARCHITECTURE FFT PIPELINEE ET NON PIPELINEE

Dans le cas du MCS 8, la fréquence de fonctionnement des chaînes d'émission est la même quelle que soit l'architecture, néanmoins la chaîne d'émission incluant l'architecture pipelinée

consomme toujours moins d'énergie. Cela est dû au taux d'activité du bloc qui est 3 fois plus important dans le cas de l'architecture non pipelinée comme le montre le Tab. 5.10. Dans ce cas, même si l'architecture pipelinée consomme plus d'énergie lors d'une activité continue, une fois insérée dans la chaîne le taux d'activité diminue et donc la consommation de puissance diminue.

Cet exemple permet de souligner que la connaissance seule de la fréquence d'horloge de la chaîne n'est pas suffisante pour évaluer l'impact d'une architecture sur la puissance consommée, mais qu'il faut également connaître les taux d'activité.

BW [MHz]	MCS	Puissance dynamique de l'architecture FFT pipelinée [mW]	Taux d'activité de l'architecture FFT pipelinée	Puissance dynamique de l'architecture FFT non pipelinée [mW]	Taux d'activité de l'architecture FFT non pipelinée
20	1	9	100%	17	100%
	3	8.5	77%	18	100%
	5	9	38.4%	18	100%
	8	8.7	25.6%	18.8	81.7%
40	1	18	100%	42	100%
	3	20.7	74%	42	100%
	5	21.5	37%	42	100%
	8	22	24.6%	41.3	76.5%

TAB. 5.10 : CONSOMMATION ET TAUX D'ACTIVITE DES DEUX ARCHITECTURES FFT A L'EMISSION

Au vu de ces résultats, nous allons à présent analyser l'impact de l'architecture de la FFT sur la chaîne de réception. En effet, dans ce cas, le bloc le plus gourmand en énergie est le décodeur, et non la FFT, lorsque les blocs sont évalués seuls. Le Tab. 5.11 montre la consommation des chaînes de réception dans le cas de l'utilisation d'une architecture FFT pipelinée et non pipelinée. Bien que les résultats soient proches, la chaîne de réception incluant l'architecture FFT pipelinée consomme toujours moins d'énergie.

Le Tab. 5.12 montre les taux d'activité du bloc décodeur dans le cas de l'utilisation d'une architecture FFT pipelinée et dans le cas d'une architecture non pipelinée. Il est intéressant de noter que le taux d'activité du bloc décodeur est plus faible dans le cas de l'architecture non pipelinée alors que c'est la chaîne qui est la plus gourmande en énergie. Cela est dû à la fréquence de fonctionnement beaucoup plus importante dans le cas de l'architecture non pipelinée. Ce résultat montre à nouveau que la complexité n'est pas le seul paramètre déterminant sur la consommation de puissance mais qu'il est nécessaire de prendre en compte également les taux d'activité et la fréquence de fonctionnement.

BW [MHz]	MCS	Architecture FFT pipelinée	Architecture FFT non pipelinée
20	1	43.6	49
	3	78.5	90
	5	160	186
	8	225.4	236
40	1	92	119
	3	181	206
	5	373.5	427
	8	509	528

TAB. 5.11 : PUISSANCE CONSOMMEE DE LA CHAÎNE DE RECEPTION AVEC L'ARCHITECTURE PIPELINEE ET NON PIPELINEE [mW]

BW [MHz]	MCS	Architecture FFT pipelinée		Architecture FFT non pipelinée	
		Taux d'activité	Fréquence d'horloge [MHz]	Taux d'activité	Fréquence d'horloge [MHz]
20	1	65%	20	20.4%	63.7
	3	100%	26	40.8%	63.7
	5	100%	52	82.7%	63.7
	8	100%	78	100%	78
40	1	67.5%	40	21.8%	124
	3	100%	54	43.5%	124
	5	100%	108	87%	124
	8	100%	162	100%	162

TAB. 5.12 : TAUX D'ACTIVITE DU BLOC DECODEUR

## 5.5 Conclusion

Dans ce chapitre, nous avons validé la méthode semi-analytique présentée au chapitre 4, qui permet une estimation de la consommation de puissance rapide et précise, sur la chaîne Wi-Fi développée. L'erreur d'estimation de la méthode est en moyenne inférieure à 4%, ce qui est acceptable.

Nous avons également validé le calcul du taux d'activité par deux stratégies :

- 1ere stratégie - méthode proposée en (4.9): la puissance du bloc fonctionnant à 100% d'activité est multipliée par le taux d'activité théorique (4.8)
- 2eme stratégie - le bloc est simulé en utilisant un fichier d'activité en entrée préalablement construit en ayant simulé toute la chaîne.

Un des principaux avantages de cette méthode est qu'elle facilite l'exploration d'architecture. En effet, si l'on change l'architecture d'un bloc ou bien si l'on ajoute un bloc au système, il suffit

uniquement d'évaluer le nouveau bloc, de calculer les nouveaux taux d'activités ainsi que la fréquence de fonctionnement associée afin d'obtenir la puissance dynamique du nouveau système. Nous avons illustré ce principe en comparant deux architectures FFT dans le but de montrer l'influence de la fréquence d'horloge et de l'activité de la chaîne. Ainsi, à l'aide de la méthode proposée, il a fallu évaluer uniquement la nouvelle architecture de FFT au lieu de simuler toute la chaîne, ce qui permet de diminuer considérablement les temps de simulation.

Cette étude a confirmé que la complexité n'était pas le seul facteur déterminant dans l'évaluation de la consommation énergétique mais qu'il est important de prendre en compte également la fréquence de fonctionnement ainsi que l'activité de tous les éléments du système.

# Conclusion et Perspectives

## Conclusion

Cette étude a permis d'évaluer et d'analyser la consommation de puissance des éléments composant la chaîne de communications numériques Wi-Fi; puis d'étudier l'impact de différents paramètres système et de différentes architectures sur la consommation de la chaîne dans son ensemble. Ce cas d'étude nous a permis de développer une méthode fiable et rapide permettant de comparer des architectures en termes de consommation.

Au chapitre 1, nous avons vu que la consommation totale dans les circuits est composée d'une composante dynamique et d'une composante statique. La puissance statique dépend de la technologie alors que la puissance dynamique est impactée par l'activité du système (commutations des transistors), la technologie, la fréquence d'horloge et la tension d'alimentation. Nous avons également comparé les circuits ASICs aux FPGAs. Nous avons conclu que les circuits FPGAs sont une bonne alternative aux ASICs pour faire du prototypage rapide et donc de l'exploration d'architecture. Enfin, nous avons détaillé quelques techniques de réduction de la consommation présentes dans la littérature et qui peuvent être appliquées à différents niveaux d'abstraction. Ces techniques permettent d'optimiser des paramètres comme la tension d'alimentation, la fréquence d'horloge ou encore l'activité du système afin de gagner en consommation.

Au chapitre 2, nous avons cité quelques techniques d'estimation de la puissance à différents niveaux du flot de conception. Nous avons montré qu'au niveau transistor, les estimations sont très précises mais nécessitent des temps de simulation très longs. A l'inverse au niveau algorithmique les temps de simulations sont négligeables mais la précision de l'estimation est faible. L'objectif de cette étude est d'évaluer de manière précise la consommation de puissance, nous avons donc développé une méthode semi-analytique combinant des informations haut et bas niveau en nous inspirant des points forts de chacune des techniques présentées dans ce chapitre. La méthode proposée vient ainsi enrichir les travaux du domaine. Notre méthode est inspirée des travaux menés dans [70], qui consistent à évaluer la consommation de puissance de la chaîne en simulation à haut niveau, en ayant préalablement évalué au niveau matériel la consommation de

puissance de chaque bloc composant la chaîne. La principale différence est que nous calculons les taux d'activité théoriquement à l'aide d'un modèle semi-analytique alors que dans [70] ces taux d'activité sont pris en compte via des simulations haut niveau.

Dans le chapitre 3, nous avons développé au niveau matériel (VHDL) une chaîne de communications numériques bande de base Wi-Fi complète comprenant l'émission et la réception, afin d'évaluer les consommations en puissance via l'outil XPower Analyser. Les scénarios spécifiés dans la norme IEEE 802.11ac ont été développés et validés par comparaison avec une chaîne logicielle de référence. La chaîne a été implémentée sur une plateforme FPGA dédiée à la mesure de la puissance dissipée par les circuits. Ce démonstrateur a donc permis de valider les résultats obtenus avec XPower dans cette thèse, les résultats issus des mesures étant très proches des valeurs obtenues par simulations. Ainsi, nous avons pu analyser les différents blocs constituant la partie bande de base de la couche physique du Wi-Fi, et évaluer leur consommation de puissance dans de nombreuses configurations. Les temps de simulations se sont cependant avérés trop longs, de l'ordre de la semaine, ce qui ne facilite pas la comparaison de différentes configurations et architectures. Notre étude a également permis d'explorer les perspectives des travaux de thèse suggérées par [71]. Ainsi, nous avons notamment comparé la consommation de puissance des chaînes pour différents MCS. Nous avons ainsi mis en évidence des scénarios plus économes en énergie, pour une même qualité de service, à partir d'un certain niveau de rapport signal-à-bruit. De plus, notre étude a permis de mettre au point une plateforme de mesure qui a permis de quantifier l'erreur produite par XPower.

Au chapitre 4, nous avons mis au point une méthode semi-analytique, rapide et précise, d'estimation de la consommation de puissance dynamique de toute chaîne de communications numériques, basée sur une analyse temporelle, et qui se décompose en trois étapes :

- La première étape consiste à créer une base de données contenant les puissances dynamiques de chaque bloc pris individuellement, et pour un taux d'activité de 100%. Les puissances dynamiques sont évaluées en simulation au niveau transistor après placement-routage. Il est nécessaire de simuler la consommation du bloc pour chaque paramètre système pouvant impacter sa consommation. Cependant, le temps de simulation reste acceptable car les blocs sont évalués individuellement et il n'est pas nécessaire d'évaluer la consommation de la chaîne complète.
- La seconde étape consiste à calculer les taux d'activité des blocs une fois insérés dans la chaîne ainsi que la fréquence de fonctionnement de la chaîne pour atteindre le débit défini dans la norme. Les taux d'activités peuvent varier selon les choix d'architecture des blocs et de la chaîne. De plus, suivant les configurations, le flux de données entrant peut ne pas

être continu ce qui implique un ajustement de la fréquence d'horloge. Le calcul des taux d'activités est réalisé à partir de l'analyse temporelle de la chaîne. Cette étape nous permet également de mettre en exergue le lien entre la fréquence de fonctionnement du système, le débit cible et les puissances des blocs, variables selon les scénarios.

- Enfin la dernière étape consiste à calculer la puissance consommée de la chaîne en sommant les consommations de chaque bloc, pondérées par les taux d'activité et corrigées par la fréquence de fonctionnement cible.

Cette méthode est dite semi-analytique car elle combine des estimations de puissance consommée réalisées à bas niveau avec des calculs analytiques de taux d'activités au niveau algorithmique. Elle fournit un gain significatif en temps de simulation, et facilite la comparaison des différentes configurations puisqu'il n'est pas nécessaire de simuler la chaîne complète pour chaque nouveau scénario, mais seulement les blocs impactés.

Cette méthode peut être appliquée à différents systèmes et pas uniquement au système de communications numériques Wi-Fi. En effet, la méthode proposée est décrite de manière générique et peut donc être généralisée en calculant par exemple les taux d'activité de n'importe quel système.

Enfin au chapitre 5, nous avons mis en pratique notre méthode et établi des modèles de consommation en fonction des différents paramètres système. Il a notamment été montré que la consommation de la FFT est proportionnelle à la fréquence d'horloge pour une bande passante donnée, alors que d'autres éléments de la chaîne, tels que l'entrelaceur ou la modulation, dépendent également de l'ordre de la modulation et du taux de codage. En outre, nous avons évalué la distribution des puissances des éléments composant la couche PHY du Wi-Fi, ce qui nous a permis d'identifier les éléments les plus consommateurs. Ainsi, nous pouvons conclure que l'entrelaceur, la FFT et le décodeur sont les blocs qui impactent le plus l'empreinte énergétique de la partie bande de base des émetteurs et récepteurs Wi-Fi, et nous recommandons d'améliorer ces blocs afin d'optimiser la consommation de puissance. Dans cette optique, nous avons comparé deux implémentations de FFT, l'une étant pipelinée et l'autre non. Nos résultats montrent que les chaînes qui intègrent une FFT pipelinée consomment moins de puissance qu'avec une FFT non pipelinée. Cela est dû aux taux d'activité et aux fréquences de fonctionnement, ces paramètres étant des facteurs déterminants de la consommation de puissance, en plus de la complexité.

## Perspectives

La méthode d'évaluation de la consommation en puissance des circuits d'une chaîne de communications numériques ouvre la voie à plusieurs perspectives, la première étant l'exploration d'architecture afin d'optimiser l'efficacité énergétique. Concernant le Wi-Fi, une première idée est de remplacer en réception le décodeur de Viterbi, associé au codage convolutif, par un codeur/décodeur LDPC (Low Density Parity Check). En effet, les codes LDPC sont spécifiés dans la norme en plus des codes convolutifs, et de nombreux travaux ont montré que les performances des décodeurs LDPC en termes de taux d'erreurs binaires sont meilleures que celles du décodeur de Viterbi. Concernant sa consommation en puissance, nous savons seulement que le décodeur LDPC est plus complexe qu'un décodeur de Viterbi, mais nous avons vu que la complexité n'est pas le seul critère à prendre en compte dans le bilan énergétique. Il serait par conséquent intéressant de comparer les consommations en puissance des récepteurs incluant les décodeurs LDPC et Viterbi.

Nous avons étudié le cas des émetteurs et récepteurs à une seule antenne (Single Input Single Output, SISO). Nous pourrions envisager de procéder à la même analyse avec des émetteurs et récepteurs multi-antennaires (MIMO). En utilisant la méthode décrite au chapitre 4, l'analyse pourrait être rapide puisque l'ensemble des blocs est déjà présent dans la base de données; pour du multiplexage spatial, il suffirait dans un premier temps de dupliquer les chaînes.

Une troisième perspective pourrait être de compléter les connaissances actuelles sur les algorithmes de réduction des pics de puissance d'émission (Peak to Average Power Ratio, PAPR). Ces algorithmes permettent de diminuer les pics de puissance non souhaités du signal émis, en contrepartie d'une complexité de traitement supplémentaire, et donc d'un coût énergétique ajouté au niveau circuit. Il serait donc pertinent de comparer l'augmentation de la consommation des circuits avec le gain obtenu au niveau de l'amplificateur de puissance. Cette étude nécessiterait une analyse approfondie de la consommation en puissance induite par la partie radio-fréquence des émetteurs/récepteurs.

Enfin une dernière perspective pourrait être de mettre en œuvre les techniques de réduction de puissance de consommation des circuits décrites au chapitre 1. Certaines de ces techniques consistent à modifier l'architecture des blocs déjà présents ou bien à en ajouter de nouveaux. Nous avons vu notamment que la technique du contrôle de l'arbre d'horloge (clock gating), qui consiste à ajouter un bloc contrôle en entrée de chaque bloc, peut réduire dans certains cas la puissance dynamique mais peut éventuellement augmenter la puissance statique. Nous avons

également vu qu'il était possible d'utiliser la méthode du DVFS combinée avec des implémentations dupliquées ou bien pipelinées. De cette manière, nous pourrions vérifier que l'on obtient bien un gain significatif sur la puissance totale (dynamique et statique) avec la réduction de la tension d'alimentation.

La méthode développée ainsi que la plateforme, peuvent être appliquées aux générations futures de systèmes Wi-Fi ainsi qu'à tout autre système de communication. Ainsi, les futurs travaux réalisés à Orange Labs bénéficieront de la plateforme de mesure et des études exposées dans cette thèse.



---

# Références

- [1] ADEME ([www.ademe.fr](http://www.ademe.fr)), Guide : Gérer ses équipements électriques, Mars 2012.
- [2] Fédération Française des Télécommunications, "Charte d'engagement volontaire du secteur des télécoms pour le développement durable", 2010.
- [3] Commission Européenne, "European Code of Conduct on Energy Consumption of Broad Band Equipment", 2011.
- [4] Commission Européenne, "Voluntary Industry Agreement to improve the energy consumption of Complex Set Top Boxes", 2010.
- [5] GreenCOMM web page. [Online]. Available : <http://greencomm-projet.org/fr/>.
- [6] "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification", IEEE 802.11ac, 2013.
- [7] J.H Choi, J. Bok and H.G. Ryu, "Power analysis and communication capacity of 160GHz wireless communication in multipath environment", IEEE International Conference on information Networkink (ICOIN), February 2012.
- [8] GreenCOMM deliverable, « Livrable 2.1, Modèle de répartition », 2014
- [9] "Broadband Radio Access Networks (BRAN); 5 GHz high performance RLAN; Harmonized EN covering the essential requirements of article 3.2 of the R&TTE Directive", ETSI EN 301 893 V1.7, June 2012.
- [10] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks", IEEE Journal on Selected Areas in Communications, vol. 22, no. 6, pp. 1089–1098, 2004.
- [11] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic and T. He, "ATPC: adaptive transmission power control for wireless sensor networks", IEEE International conference on Embedded networked sensor systems, pp. 223-236, November 2006.

- [12] S-H. Jung, J-H. Baek and S-Y. Kim, "Short circuit power estimation of static CMOS circuits", IEEE International Proceedings of the ASP-Design Automation Conference, February 2001.
- [13] A. Chatterjee, M. Nandakumar and I-C. Chen, "An investigation of the impact of technology scaling on power wasted as short-circuit current in low voltage static CMOS circuits", IEEE International Symposium on Low Power Electronics and Design, August 1996.
- [14] Q. Wang and S.B.K. Vrudhula, "On short circuit power estimation of CMOS inverters", IEEE International Conference on Computer Design: VLSI in Computers and Processors, October 1998.
- [15] M. Klein, "Static Power and the Importance of Realistic Junction Temperature Analysis", Xilinx, WP221, v1.0, March 23, 2005.
- [16] Procédé de gravure en 14 nm - Intel, web page. [Online]. Available : <http://www.intel.fr/content/www/fr/fr/silicon-innovations/intel-14nm-technology.html>
- [17] web page. [Online]. Available : <http://theinnovationandstrategyblog.com/2014/07/09/linnovation-et-le-progres-est-ce-la-meme-chose/>
- [18] Principes fondamentaux du FPGA, web page. [Online]. Available : <http://www.ni.com/white-paper/6983/fr/>
- [19] web page. [Online]. Available : [http://cleroux.vvv.enseirb-matmeca.fr/EN219/Cours/Cours\\_EN219.pdf](http://cleroux.vvv.enseirb-matmeca.fr/EN219/Cours/Cours_EN219.pdf)
- [20] I. Kuon and J Rose, "Measuring the gap between FPGAs and ASICs", IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 2, pp. 203-215, 2007.
- [21] K.S. Khouri and N.K. Jha, "Leakage power analysis and reduction during behavioral synthesis", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp. 876-885, December 2002.
- [22] P.C. Bhaskar and V.K. Jathar, "Development of processor engine for FPGA based clock gating and performing power analysis", IEEE International Conference on Computing Communication Control and Automation (ICCUBEA), August 2016.
- [23] D.Garg and R. Sharma, "Low power multiplier using dynamic voltage and frequency Scaling (DVFS)", IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), September 2014.

- 
- [24] J.B. Goeders and S.J.E. Wilton, "VersaPower : Power estimation for diverse FPGA architectures", IEEE International Conference on Field-Programmable Technology (FPT), December 2012.
- [25] C.C. Tsang and H.K.H. So, "Reducing dynamic power consumption in FPGAs using precomputation", IEEE International Conference on Field-Programmable Technology, December 2009.
- [26] T. Marconi, D. Theodoropoulos, K. Bertels and G. Gaydadjiev, "A novel HDL coding style to reduce power consumption for reconfigurable devices", IEEE International Conference on Field-Programmable Technology (FPT), December 2010.
- [27] M. Nakkar and P. Franzon, "Low power logical element for FPGA fabric", IEEE International Conference on Microelectronics (ICM), December 2002.
- [28] S. Deo, "Power consumption calculation of AP-DCD Algorithm using FPGA Platform", IEEE International Conference on Reconfigurable computing and FPGA, Dec 2010.
- [29] A. Garcia, W. Burleson, and J.L. Danger, "Power modeling in field programmable gate array (FPGA) ", In journées francophone d'études faibles tension faible consommation, 2001.
- [30] F. Li, D. Chen, L. He and J. Cong, "Architecture evaluation for power efficient FPGAs", IEEE International symposium on Field programmable gate array, pp. 175-184, February 2003.
- [31] V. Betz, J. Rose and A. Marquardt, "Architecture and CAD for deep-submicron FPGAs", The Springer International Series in Engineering and Computer Science, February 1999.
- [32] L. Shang, L.S. Peth and N.K. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks", IEEE International Symposium on High-Performance Computer Architecture, February 2003.
- [33] D.V. Matyushin and S.I. Kurganskii, "Power consumption reduction for configuration SRAM of field programmable gate arrays", IEEE International Conference and Seminar of Young Specialists on Micro/Nanotechnologies and Electron Devices (EDM), July 2011.
- [34] E. Vianello, O. Thomas, G. Molas, O. Turkyilmaz, N. Jovanovic, D. Garbin, G. Palma, M. Alayan, C. Nguyen, J. Coignus, B. Giraud, T. Benoist, M. Reyboz, A. Toffoli, C. Charpin, F. Clermidy, L. Perniol, "Resistive memories for ultra-low-power embedded computing design", IEEE International Electron Devices Meeting (IEDM), December 2014.
- [35] A. Nafkha and Y. Louet, "Accurate measurement of power consumption overhead during FPGA dynamic partial reconfiguration", IEEE International Symposium on Wireless Communication System (ISWCS), September 2016.

- [36] Xilinx, web page. [Online]. Available : <https://www.xilinx.com/>
- [37] N. Lawal, F. Lateef and M. Usman, "Power consumption measurement & configuration time of FPGA", IEEE International Conference on Power Generation System and Renewable Energy Technologies (PGSRET), June 2015.
- [38] LabVIEW, web page. [Online]. Available : <http://www.ni.com/labview/f/>
- [39] GreenCOMM deliverable, "Livrable 5.1b, Démonstrateur FPGA du récepteur pour le système WiFi", 2016.
- [40] "VC707 Evaluation board for the Virtex-7 FPGA, USER guide," vol. v 1.7, March 2016.
- [41] SPICE consortium. (2006) The spice page. web page. [Online]. Available : <http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE/>
- [42] X. Tang, P. E. Gaillardon, and G. D. Micheli, "FPGA-SPICE: a simulation-based power estimation framework for FPGAs", IEEE International Conference on Computer Design (ICCD), pp. 696-703, 2015.
- [43] Xilinx Power Tools Tutorial, Spartan-6 and Virtex-6 FPGA, v14.4, Xilinx. Dec. 2012.
- [44] A. Salz and M. Horowitz, "IRSIM : An incremental mos switch-level simulator", IEEE International Conference In Design Automation, 1989, pp. 173-178, June 1989.
- [45] Nanosim User Guide, TLD-2001.06, June 2001, web page. [Online]. Available : [http://frank.harvard.edu/~howard/pulsenet/docs/nanosim\\_usersguide.pdf](http://frank.harvard.edu/~howard/pulsenet/docs/nanosim_usersguide.pdf)
- [46] R. Burch, F. Najm, P. Yang and T. Trick, "McPower: a Monte Carlo approach to power estimation", IEEE International Conference on Computer Aided Design, pp. 90-97, November 1992.
- [47] N.C.K. Choy and S.J.E. Wilton, "Activity based power estimation and characterization of DSP and multiplier blocks in FPGAs", IEEE International Conference on Field Programmable Technology, pp. 253-256, Dec 2006.
- [48] S. Dueck, "A power evaluation framework for FPGA Applications and CAD experimentation", PhD thesis, University of British Columbia, June 2013.
- [49] D. Hodges, H. Jackson and R. Salah, "Analysis and Design of Digital Integrated Circuits", ed. McGraw-Hill, 2003.
- [50] W. Shum and J.H Anderson, "FPGA glitch power analysis and reduction", IEEE International Symposium on Low Power Electronics and Design (ISLPED), pp. 27-32, August 2011.

- 
- [51] S.H. Jung, J.H. Baek and S.Y. Kim, "Short circuit power estimation of static CMOS circuits", IEEE Proceedings of the ASP- Design Automation Conference, February 2001.
- [52] T. Lin, K.S. Chong, B.H. Gwee and J.S. Chang, "Fine-grained power gating for leakage and short-circuit power reduction by using asynchronous logic", IEEE International Symposium on Circuit and Systems, May 2009.
- [53] G. Chen and E.G. Friedman, "Effective capacitance of inductive interconnects for short-circuit power analysis", IEEE Transactions on Circuit and Systems II, vol. 55, pp. 26-30, January 2008.
- [54] D. Eckerbert and P. Larsson-Edefors, "Interconnect-driven short-circuit power modeling", IEEE Proceedings Euromicro Symposium on Digital Systems Design, September 2001.
- [55] H. Kawauchi, M. Tsuzuki, I. Taniguchi and M. Fukui, "An accurate RTL power estimation considering library unevenness", IEEE International Symposium on Circuits and Systems (ISCAS), June 2010.
- [56] N. Bombieri, F. Fummi, V. Guarnieri and A. Acquaviva, "Energy aware TLM platform simulation via RTL abstraction", IEEE International High Level Design Validation and Test Workshop (HLDVT), November 2012.
- [57] P. Schumacher, P. Jha, S. Kuntur, T. Burke and A. Frost, "Fast RTL power estimation for FPGA design", IEEE International Conference on Field Programmable Logic and Applications (FPL), September 2011.
- [58] A. Singh, M. Shetty, S. Ravi and R. Nibandhe, "Methodology for early and accurate test power estimation at RTL", IEEE International Test Conference (ITC), November 2010
- [59] M. Denoual, "Estimation au niveau architectural de la consommation des circuits dédiée au traitement numérique du signal", PhD thesis, Université de Nice-Sophia Antipolis, 2001.
- [60] A. Bogliolo, I. Colonescu, E. Macii, and M. Poncino, "An RTL power estimation tool with on-line model building capabilities", IEEE International Workshop in Power and Timing Modeling, Optimization and Simulation, pp. 391–396, January 2001.
- [61] O. Arnold, F. Richter, G. Fettweis, and O. Blume, "Power consumption modeling of different base station types in heterogeneous cellular networks", IEEE in Future Network Mobile Summit, pp. 1-8, 2010.
- [62] S. Shim, J.W. Lee and Y. Shin, "An analytical approach to thermal design and optimization with a temperature-dependent power model", IEEE Transaction on Circuits and Systems I (TCAS), vol. 62, no. 3, pp. 816-324, Mar. 2015.
- [63] Altera, web page. [Online]. Available : <https://www.altera.com/>

- [64] Xilinx, Xilinx Power Estimator User Guide, December 2012, web page. [Online]. Available :  
[http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2012\\_4/ug440-xilinx-power-estimator.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx2012_4/ug440-xilinx-power-estimator.pdf)
- [65] C. Kalaycioglu and I. Hamzaoglu, "Dynamic power estimation for motion estimation hardware", IEEE Euromicro Conference on Digital System Design (DSD), pp. 279-282, 2011.
- [66] A. Amira and S. Chandrasekaran, "Power modeling and efficient FPGA implementation of FHT for signal processing", IEEE Transaction on Very Large Scale Integration (VLSI) Systems, vol. 15, no. 3, pp. 286-295, Mar. 2007.
- [67] J. Ou and V.K. Prasanna, "PyGen : a MATLAB/Simulink based tool for synthesizing parameterized and energy efficient designs using FPGAs", IEEE Symposium on Field-Programmable Custom Computing Machines, April 2004.
- [68] MATLAB, web page. [Online]. Available: <https://fr.mathworks.com/products/matlab.html>
- [69] S. Choi, J.W. Jang, S. Mohanty and V.K. Prasanna, "Domain-specific modeling for rapid energy estimation of reconfigurable architectures", The journal of Supercomputing, v 26, issue 3, pp 259-281, November 2003.
- [70] J. Lorandel, J. C. Prévotet, and M. H elard, "Fast power and performance evaluation of FPGA-based wireless communication systems", IEEE Access, vol. 4, pp. 2005-2018, Apr. 2016.
- [71] J. Lorandel, "Etude de la consommation  nerg tique de syst me de communications num riques sans fil implant s sur cible FPGA", PhD thesis, Universit  de l'INSA de Rennes, 2016.
- [72] Accellera Systems Initiative. System-C, web page. [Online]. Available: <http://accellera.org/>
- [73] Fusion Digital Power Designer, web page. [Online]. Available :  
[https://www.ti.com/tool/fusion\\_digital\\_power\\_designer](https://www.ti.com/tool/fusion_digital_power_designer)
- [74] ModelSim, web page. [Online]. Available : <https://www.altera.com/products/design-software/model---simulation/modelsim-altera-software.html>
- [75] W. Benali, M. Le Bot, C. Langlais, S. Kerouedan, "Power consumption of Wi-Fi transceivers", IEEE International Symposium on Wireless Communication System (ISWCS) 2016.
- [76] W. Benali, M. Le Bot, C. Langlais, S. Kerouedan, "Consommation des  metteurs/r cepteurs Wi-Fi", Conf rence GRETSI, Septembre 2017.

- [77] web page. [Online]. Available : [http://adslproject.free.fr/prefixe\\_cyclique.htm](http://adslproject.free.fr/prefixe_cyclique.htm)
- [78] Data Sheet IPCore FFT Xilinx, web page. [Online]. Available :  
[https://www.xilinx.com/support/documentation/ip\\_documentation/xfft\\_ds260.pdf](https://www.xilinx.com/support/documentation/ip_documentation/xfft_ds260.pdf)
- [79] G. D. Forney, "The Viterbi algorithm", IEEE Proceedings, vol 61, pp. 268-276, 1973.



La forte augmentation du nombre de terminaux connectés ces dernières années et l'utilisation croissante des technologies de communication impacte de manière non négligeable la facture énergétique. Pour enrayer cette augmentation de la consommation énergétique, il devient primordial de pouvoir comparer en termes de consommation les algorithmes de communications numériques, afin de développer l'architecture de transmission la moins énergivore.

Dans cette thèse, la couche physique des standards Wi-Fi IEEE 802.11ac est analysée sous un angle énergétique. La puissance dissipée dans les circuits pour faire fonctionner les algorithmes de traitement de signal est prise en compte en plus de la puissance d'émission d'antenne classique. La méthodologie mise en œuvre inclut à la fois des simulations et des développements sur plateforme matérielle (FPGA), permettant d'obtenir des évaluations de la consommation plus réalistes.

Nous avons dans un premier temps analysé de façon isolée les éléments composant les chaînes de communications numériques. Puis nous avons intégré les périodes d'activité et d'inactivité de chaque élément dans le calcul de la consommation énergétique globale des chaînes. Nous proposons une méthode pratique et efficace d'estimation de la consommation, incluant une base de données issue de simulations, et une analyse théorique des taux d'activité de chaque élément de la chaîne.

Ces résultats permettent d'analyser la répartition de la consommation en puissance des éléments composant les émetteurs et les récepteurs, et de comparer diverses architectures et jeux de paramètres. En particulier, nous avons évalué l'impact de deux architectures de Transformées de Fourier Rapides sur la consommation globale du système.

**Mots clef** : FPGA, Consommation énergétique, Estimation de puissance, Couche physique, Wi-Fi, Exploration d'architecture, Prototypage, Mesure de consommation d'un circuit

The strong increase of the number of connected devices in recent years and the increasing use of communication technologies has a significant impact on the energy bill. To stop the increase in energy consumption, it is essential to be able to compare the digital communication algorithms in terms of consumption, in order to develop the most energy-efficient transmission architecture.

In this thesis, the IEEE 802.11ac Wi-Fi standard of physical layer is analyzed at an energy point of view. The power dissipated in the circuits for operating the signal processing algorithms is taken into account in addition to the antenna transmission power. The implemented methodology includes both simulations and developments on a hardware platform (FPGAs), resulting in more realistic consumption assessments.

First, we analyzed separately the components of the digital communications chains. Then we integrated the periods of activity and inactivity of each element in the calculation of the global energy consumption of the chains. We propose a practical and efficient method of estimating consumption, including a database derived from simulations, and a theoretical analysis of the activity rates of each element of the chain.

These results make it possible to analyze the distribution of the power consumption of the elements composing transmitters and receivers, and to compare various architectures and sets of parameters. In particular, we evaluated the impact of two Fast Fourier Transform architectures on overall system consumption.

**Keywords:** FPGA, Power consumption, Power estimation, Physical layer, Wi-Fi, Architectural exploration, Prototyping, Consumption measurement of a circuit